Les 5 Schatten en simuleren

5.1 Maximum likelihood schatting

Tot nu toe hebben we meestal naar voorbeelden gekeken waar we van een kansverdeling zijn uitgegaan en dan voorspellingen hebben gemaakt. In de praktijk komen we echter vaak een iets andere situatie tegen. We weten dat er iets volgens een zekere kansverdeling zal gebeuren, maar deze hangt van een parameter af die we niet kennen. Bijvoorbeeld kunnen we aannemen dat de kans p waarmee een machine defecte stukken produceert constant is, maar dat we de waarde van p niet kennen. Als we nu in een steekproef defecte stukken tellen, kunnen we het aantal defecte stukken door de binomiale (of hypergeometrische) verdeling beschrijven. Wat we nu nodig hebben is een *schatting* voor de kans p, gegeven de aantallen van defecte stukken in een paar steekproeven. Neem aan dat we altijd een steekproef van m stukken nemen, dan vinden we in de verschillende steekproeven $k_1, k_2, \ldots, k_n$ defecte stukken. We kunnen nu op verschillende manieren een waarde voor p schatten, bijvoorbeeld:

- simplistisch: We schatten $p = \frac{k_1}{m}$, dus we nemen aan dat de eerste steekproef typisch was en negeren de anderen (dit kunnen we natuurlijk ook met k_3 of k_n in plaats van k_1 doen).
- optimistisch: We schatten $p = \frac{k_{min}}{m}$, waarbij k_{min} de minimale waarde van de k_i is.
- pessimistisch: We schatten $p = \frac{k_{max}}{m}$, waarbij k_{max} de maximale waarde van de k_i is.
- pragmatisch: We schatten $p = \frac{1}{n} \cdot \frac{\sum_{i=1}^{n} k_i}{m}$, dus we nemen het gemiddelde van de relatieve frequenties in de enkele steekproeven.

Een algemene methode om parameters van kansverdelingen te schatten is gebaseerd op het volgende argument: Voor elke keuze van een parameter (of meerdere parameters) heb je een kansverdeling, die aan een waargenomen resultaat een zekere kans geeft. In het voorbeeld is dit $P_p(X = k) = b(m, p; k) = \binom{m}{k} p^k (1-p)^{m-k}$. Bij onafhankelijke herhaling kunnen we de kans voor een rij observaties als product van de kansen voor de aparte observaties berekenen, in het voorbeeld hebben we dus

$$P_p(k_1, \ldots, k_n) = \prod_{i=1}^{n} \binom{m}{k_i} p^{k_i} (1-p)^{m-k_i}.$$

De kans voor de observaties is nu een functie van de parameter p en we noemen deze functie de *aannemelijkheidsfunctie* of *likelihood* functie. We maken nu een schatting voor p door te zeggen, dat we p zo kiezen dat de aannemelijkheidsfunctie aan maximale waarde heeft, dus dat de kans voor onze observatie maximaal wordt. Deze methode van schatting noemt men de *meest aannemelijke* of *maximum likelihood* schatting van de parameter.

Om een maximum likelihood schatting uit te werken, moeten we in principe de functie $P_p(k_1, \ldots, k_n)$ naar p afleiden en de nulpunten van de afgeleide bepalen. Omdat de kans een product van de enkele kansen is, zal het afleiden een hele hoop termen opleveren, want we moeten altijd de productregel toepassen. Hier is het volgende trucje vaak erg handig: In plaats van het maximum van $P_p(k_1, \ldots, k_n)$ te berekenen, bepalen we het maximum van $\log(P_p(k_1, \ldots, k_n))$. Dit zit namelijk op dezelfde plek, omdat de logaritme een monotone functie is. De (negatieve) logaritme van de kans noemt men soms ook de *score* van een uitkomst.

We gaan nu de maximum likelihood schatting voor een aantal kansverdelingen uitwerken:

Binomiale verdeling: In n steekproeven van grootte $m_1, \ldots, m_n$ vinden we $k_1, \ldots, k_n$ gunstige uitkomsten. We hebben

$$P_p(k_1, \ldots, k_n) = \prod_{i=1}^{n} \binom{m_i}{k_i} p^{k_i} (1-p)^{m_i - k_i}.$$

We definiëren nu

$$L(p) = \log(P_p(k_1, \ldots, k_n)) = \sum_{i=1}^{n} \left(\log(\binom{m_i}{k_i}) + \log(p^{k_i}) + \log((1-p)^{m_i - k_i}) \right)$$

$$= \sum_{i=1}^{n} \log(\binom{m_i}{k_i}) + \sum_{i=1}^{n} k_i \log(p) + \sum_{i=1}^{n} (m_i - k_i) \log(1-p).$$

De afgeleide (met betrekking tot p) hiervan is

$$L'(p) = \frac{1}{p}(\sum_{i=1}^{n} k_i) - \frac{1}{1-p}(\sum_{i=1}^{n} (m_i - k_i)).$$

We hebben

$$L'(p) = 0 \Leftrightarrow (1-p)(\sum_{i=1}^{n} k_i) = p(\sum_{i=1}^{n} (m_i - k_i)) \Leftrightarrow \sum_{i=1}^{n} k_i = p(\sum_{i=1}^{n} m_i)$$

$$\Leftrightarrow p = \frac{\sum_{i=1}^{n} k_i}{\sum_{i=1}^{n} m_i}.$$

Dit betekent dat we de parameter als de relatieve frequentie van gunstige uitkomsten in alle steekproeven bij elkaar kiezen. Dit komt op de pragmatische keuze neer, maar we hebben nu een betere onderbouwing voor onze keuze. Het is namelijk de parameter die de observaties het beste verklaart.

Poisson-verdeling: Een zeldzaam gebeurtenis zien we $k_1, \ldots, k_n$ keer gebeuren. We hebben

$$P_\lambda(k_1, \ldots, k_n) = \prod_{i=1}^{n} \frac{\lambda^{k_i}}{k_i!} e^{-\lambda}.$$

We definiëren nu

$$L(\lambda) = \log(P_\lambda(k_1, \ldots, k_n)) = \sum_{i=1}^{n} (\log(\lambda^{k_i}) - \log(k_i!) - \lambda)$$

$$= \sum_{i=1}^{n} k_i \log(\lambda) - \sum_{i=1}^{n} \log(k_i!) - n\lambda.$$

De afgeleide hiervan is

$$L'(\lambda) = \frac{1}{\lambda}(\sum_{i=1}^{n} k_i) - n$$

en we hebben

$$L'(\lambda) = 0 \Leftrightarrow \lambda = \frac{1}{n}(\sum_{i=1}^{n} k_i).$$

De schatting voor de verwachtingswaarde λ van de Poisson-verdeling is dus het rekenkundig gemiddelde van de aantallen geobserveerde zeldzame gebeurtenissen. Ook dit klopt met onze intuïtie, dat we na een aantal pogingen aannemen, dat we vervolgens ook weer gebeurtenissen met ongeveer hetzelfde gemiddelde zullen krijgen.

Normaalverdeling: De normaalverdeling wordt in de opgaven behandeld.

Exponentiële verdeling: Voor een gebeurtenis dat volgens een exponentiële verdeling optreedt maken we de observaties $x_1, \ldots, x_n$. Er geldt

$$f_\lambda(x_1, \ldots, x_n) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i} = \lambda^n e^{-\lambda(\sum_{i=1}^{n} x_i)}.$$

Merk op dat we het hier met een dichtheidsfunctie voor de kansverdeling te maken hebben. Maar we kunnen aannemen, dat we steeds een klein interval rond een geobserveerde waarde bekijken, dan is de kans voor een observatie in het interval $[x, x+\delta]$ gegeven door $P_\lambda(X \in [x, x+\delta]) = \lambda e^{-\lambda x}\delta$. Maar δ heeft als constante factor geen invloed op het maximum van de functie, dus kunnen we meteen naar de dichtheidsfunctie kijken.

We definiëren nu

$$L(\lambda) = \log(f_\lambda(x_1, \ldots, x_n)) = \log(\lambda^n) - \lambda(\sum_{i=1}^{n} x_i) = n\log(\lambda) - \lambda(\sum_{i=1}^{n} x_i).$$

De afgeleide hiervan is

$$L'(\lambda) = \frac{n}{\lambda} - (\sum_{i=1}^{n} x_i)$$

en we hebben

$$L'(\lambda) = 0 \Leftrightarrow \frac{1}{\lambda} = \frac{1}{n}(\sum_{i=1}^{n} x_i).$$

De schatting voor de verwachtingswaarde $\frac{1}{\lambda}$ van de exponentiële verdeling is dus weer het rekenkundig gemiddelde van de observaties.

Hypergeometrische verdeling: In de eerste les hebben we al het voorbeeld bekeken dat we het aantal vissen in een vijver willen bepalen. Het idee hiervoor is, dat we s vissen markeren en dan kijken hoeveel gemarkeerde vissen we in een (latere) steekproef van m vissen vinden. De kans dat we er k gemarkeerde vissen in vinden is gegeven door de hypergeometrische verdeling

$$h(n, m, s; k) = \frac{\binom{s}{k}\binom{n-s}{m-k}}{\binom{n}{m}}$$

waarbij n het onbekende aantal vissen in de vijver is. In dit voorbeeld gaan we niet de logaritme gebruiken, maar bepalen we het maximum van $h(n, m, s; k)$ als een functie van n op een andere manier. We kijken naar de quotiënt

$$q(n) := \frac{h(n, m, s; k)}{h(n-1, m, s; k)}.$$

Als $q(n) \geq 1$ is $h(n, m, s; k)$ stijgend, als $q(n) \leq 1$ is $h(n, m, s; k)$ dalend. We hebben

$$q(n) = \frac{\binom{s}{k}\binom{n-s}{m-k}}{\binom{n}{m}} \cdot \frac{\binom{n-1}{m}}{\binom{s}{k}\binom{n-1-s}{m-k}} = \frac{\frac{(n-s)!}{(m-k)!(n-s-m+k)!} \cdot \frac{(n-1)!}{m!(n-1-m)!}}{\frac{n!}{m!(n-m)!} \cdot \frac{(n-1-s)!}{(m-k)!(n-1-s-m+k)!}}$$

$$= \frac{(n-s)!(n-1)!(n-m)!(n-1-s-m+k)!}{(n-s-m+k)!(n-1-m)!n!(n-1-s)!} = \frac{(n-s)(n-m)}{(n-s-m+k)n}$$

$$= \frac{n^2 - sn - nm + sm}{n^2 - sn - mn + kn}.$$

We zien dus dat $q(n) \geq 1$ als $sm \geq kn$ en $q(n) \leq 1$ als $sm \leq kn$. Het maximum wordt dus bereikt voor $n = \frac{sm}{k}$, d.w.z. voor $\frac{k}{m} = \frac{s}{n}$. Dit betekent dat we de grootte van de populatie zo schatten dat het relatieve aantal gemarkeerde vissen in onze vangst hetzelfde is als het relatieve aantal in de hele vijver.

Merk op: In de voorbeelden die we hier hebben behandeld, kunnen we de maximum likelihood schatting expliciet uitrekenen en krijgen meestal een resultaat dat we ook intuïtief hadden verwacht. Voor ingewikkeldere kansverdelingen (bijvoorbeeld met veel parameters) is het vaak niet mogelijk de nulpunten van de partiële afgeleiden expliciet te bepalen. Hier worden dan iteratieve benaderingsmethoden toegepast, bijvoorbeeld het *EM-algoritme* (hierbij staat *EM* voor *expectation maximization*).

Er zijn ook andere schatters dan de maximum likelihood schatter, bijvoorbeeld de *momentenschatters*. Het k-de moment van een stochast X is de verwachtingswaarde $E(X^k)$ van de k-de macht van de stochast. Bij een momentenschatter wordt geprobeerd de parameters van een kansverdeling zo te bepalen dat de momenten van de kansverdeling gelijk zijn aan de momenten die in een steekproef waargenomen zijn.

We zullen ons hier niet verder in verdiepen omdat het probleem van het schatten van parameters van een kansverdeling meer in de statistiek thuis hoort.

5.2 Simulatie

Soms heb je bij experimenten na een aantal observaties een idee erover wat er gebeurt en maak je er een model om de resultaten te beschrijven. De kwaliteit van een model ligt in het vermogen om toekomstige resultaten te kunnen voorspellen en dit is ook de manier hoe een model getoetst wordt. Vaak zijn experimenten zo ingewikkeld of kostbaar dat je bij een aanpassing van het beschrijvende model niet meteen weer veel experimenten kunt of wilt doen. Dan is het handig om het nieuwe model met een simulatie te testen, waarbij je zekere parameters volgens een kansverdeling kiest.

Een andere motivatie voor het simuleren van kansverdeling is dat sommige effecten pas naar heel veel herhalingen van een experiment naar voren komen. Voor een computer is het veel makkelijker om iets 10000 keer te herhalen dan dit in de realiteit te doen, bijvoorbeeld een munt 10000 keer te werpen.

We gaan daarom in deze paragraaf bekijken hoe we voor een aantal kansverdelingen een stochast met gegeven verdelingsfunctie kunnen simuleren.

Randomgenerator

Het startpunt voor alle soorten simulaties is een *toevalsgenerator* of *randomgenerator*. Dit is een procedure (een soort orakel) die een rij getallen $U_1, U_2, \ldots$ tussen 0 en 1 produceert die aan de volgende eisen voldoet:

(1) De kansverdeling op de i-de plek in de rij is de uniforme verdeling op het interval $[0, 1)$, d.w.z. er geldt $P(U_i \leq u) = u$ voor elke i.

(2) De stochasten $U_1, U_2, \ldots$ zijn onafhankelijk, d.w.z. voor elke keuze van indices $i_1 < i_2 < \ldots < i_k$ hebben we $P(U_{i_1} \leq u_1, U_{i_2} \leq u_2, \ldots, U_{i_k} \leq u_k) = P(U_{i_1} \leq u_1) \cdot P(U_{i_2} \leq u_2) \cdot \ldots \cdot P(U_{i_k} \leq u_k) = u_1 \cdot u_2 \cdot \ldots \cdot u_k$.

Als de rij $U_1, U_2, \ldots$ van getallen aan deze eisen voldoet, noemt men de U_i *toevalsgetallen*. Helaas kan een praktische implementatie van een toevalsgenerator nooit perfect aan deze eisen voldoen, men spreekt daarom strikt genomen beter ervan dat een randomgenerator *pseudo-toevalsgetallen* en geen 'echte' toevalsgetallen produceert.

Een veel gebruikte type van randomgeneratoren zijn de *lineaire congruentie modellen*: Kies een getal $m \in \mathbb{N}$, constanten $a, c \in \mathbb{Z}$ en een *zaad* (Engels: seed) I_0. Vervolgens bereken je iteratief

$$I_{n+1} := (aI_n + c) \bmod m$$

waarbij $x \bmod m$ de rest bij het delen van x door m is. De waarden van de getallen I_n liggen tussen 0 en $m - 1$, hieruit krijgt men toevalsgetallen U_n in het interval $[0, 1)$ door $U_n := \frac{I_n}{m}$ te definiëren.

Omdat I_n alleen maar de waarden $0, 1, \ldots m-1$ kan hebben, is deze randomgenerator altijd periodiek met een periode van lengte hoogstens m. Maar behalve voor speciale (slechte) waarden van m, a, c en I_0 wordt deze lengte van de periode ook bereikt en levert deze methode een redelijk goede randomgenerator. Vaak wordt voor m een macht van 2 zoals 2^{32} gekozen, omdat dit op

een computer met 32-bit of 64-bit getallen de *modulo* operatie heel eenvoudig maakt. In dit geval laat zich aantonen, dat een lineaire congruentie model met $a \equiv 1 \bmod 4$ en een oneven c altijd een periode van maximale lengte oplevert.

Voordat een randomgenerator voor simulaties wordt gebruikt, is het verstandig om te toetsen of de pseudo-toevalsgetallen die hij oplevert inderdaad redelijk goed gelijkverdeeld en onafhankelijk zijn. Hiervoor zijn er een aantal tests, die op methoden uit de statistiek gebaseerd zijn.

Om een eerste indruk te krijgen, kan men de punten (U_{2i-1}, U_{2i}) in het 2-dimensionale vlak plotten en kijken of dit er redelijk toevallig uitziet. Als er hier al een soort structuur of patroon opvalt, is er zeker iets mis met de randomgenerator.

In een iets systematischere test deelt men het interval $[0, 1)$ in d (even grote) deelintervallen, telt hoe veel van $U_1, U_2, \ldots, U_n$ in elk van die deelintervallen ligt en toetst deze verdeling met een χ^2-test tegen de gelijkverdeling. (De χ^2-test is een standaardtest uit de statistiek die toetst of de gevonden verdeling te veel of te weinig van de gelijkverdeling afwijkt, want het is ook heel onwaarschijnlijk, dat in elk deelinterval precies d/n getallen terecht komen.) Een soortgelijke test kan men in plaats van de enkele toevalsgetallen ook op paren of in het algemeen op k-dimensionale vectoren $(U_1, \ldots, U_k)$, $(U_{k+1}, \ldots, U_{2k}), \ldots, (U_{(n-1)k+1}, \ldots, U_{nk})$ toepassen, die gelijkverdeeld in de k-dimensionale kubus $[0, 1)^k$ moeten zijn.

Met andere tests wordt de onafhankelijkheid getoetst. Bijvoorbeeld wordt in de *gap test* een deelinterval $[a, b]$ van $[0, 1)$ gekozen en vervolgens gekeken, hoe lang de stukken van de rij (U_i) zijn die niet in $[a, b]$ liggen. Als we $p := |b - a|$ definiëren, dan is de kans op een stuk van lengte k tussen twee getallen die wel in $[a, b]$ liggen, gelijk aan $p(1 - p)^k$ (dit noemt men een geometrische verdeling met parameter p). De gevonden verdeling van lengtes van stukken kunnen we nu ook weer tegen de verwachte geometrische verdeling toetsen (bijvoorbeeld met een χ^2-toets).

We gaan er vanaf nu van uit dat we een (wel getoetste) randomgenerator ter beschikking hebben, die elke keer dat we hem gebruiken een toevalsgetal $U_i \in [0, 1)$ oplevert zo dat deze getallen gelijk verdeeld en onafhankelijk zijn.

Er zijn een aantal algemene principes, hoe we een gewenste kansverdeling met behulp van een randomgenerator kunnen simuleren. De meest belangrijke zijn de methode van de *inverse verdelingsfunctie* en de *wegwerp* (rejection) methode. In principe zijn deze methoden op discrete en continue kansverdelingen toepasbaar, omdat we ook voor discrete kansverdelingen vaak eenvoudig een verdelingsfunctie $F(x)$ en dichtheidsfunctie $f(x)$ kunnen aangeven. Maar voor zekere discrete kansverdelingen zullen we later nog andere (meer directe) methoden aangeven.

Simulatie met behulp van de inverse verdelingsfunctie

Voor een algemene (continue) kansverdeling met dichtheidsfunctie $f(x)$ en verdelingsfunctie $F(x) = \int_{-\infty}^{x} f(t)\, dt$ passen we de inverse F^{-1} van de verdelings-

functie op de uniforme verdeling toe: Zij U een stochast met uniforme verdeling op $[0,1)$, dan definiëren we een nieuwe stochast X door $X := F^{-1}(U)$. Voor de kans $P(X \leq a)$ geldt nu $P(X \leq a) = P(F(X) \leq F(a)) = P(U \leq F(a)) = F(a)$ omdat U uniform verdeeld is. De stochast X heeft dus de verdelingsfunctie $F(x)$.

Voorbeeld 1: We willen een algemene rechthoekverdeling op het interval $[a,b]$ simuleren. De verdelingsfunctie voor deze verdeling is $F(x) = \frac{1}{b-a}(x-a)$ en uit $y = \frac{1}{b-a}(x-a) \Leftrightarrow (b-a)y = (x-a) \Leftrightarrow x = a + (b-a)y$ volgt $F^{-1}(y) = a + (b-a)y$.

We krijgen dus een toevalsrij (V_i) met waarden in het interval $[a,b]$ door $V_i := a + (b-a)U_i$. Dit hadden we natuurlijk ook zonder de inverse van de verdelingsfunctie kunnen bedenken.

Voorbeeld 2: Ook voor de exponentiële verdeling krijgen op deze manier een simulatie. Na een mogelijke verschuiving op de x-as heeft de exponentiële verdeling de dichtheidsfunctie $f(x) = \lambda e^{-\lambda x}$ en de verdelingsfunctie $F(x) = 1 - e^{-\lambda x}$. Omdat $y = 1 - e^{-\lambda x} \Leftrightarrow -\lambda x = \log(1-y) \Leftrightarrow x = -\frac{1}{\lambda}\log(1-y)$, hebben we $F^{-1}(y) = -\frac{1}{\lambda}\log(1-y)$. Voor een uniform verdeelde stochast U is dus de stochast $X := -\frac{1}{\lambda}\log(1-U)$ exponentieel verdeeld met parameter λ. Maar omdat met U ook $1-U$ gelijkverdeeld op $[0,1)$ is, kunnen we net zo goed ook $X := -\frac{1}{\lambda}\log(U)$ definiëren.

Simulatie met behulp van de wegwerp methode

Soms is de inverse F^{-1} van de verdelingsfunctie $F(x)$ van een kansverdeling niet makkelijk te bepalen of zelfs onmogelijk expliciet op te schrijven. Het meest prominente voorbeeld hiervoor is de normaalverdeling.

Maar we kunnen een kansverdeling met dichtheidsfunctie $f(x)$ op een eindig interval $[a,b]$ als volgt simuleren: Stel de dichtheidsfunctie is op het interval $[a,b]$ door een waarde c begrensd, d.w.z. $f(x) \leq c$ voor alle $x \in [a,b]$. Dan produceren we een rij toevalsgetallen (X_i) volgens een gelijkverdeling op $[a,b]$ en een rij (Y_i) volgens een gelijkverdeling op $[0,c]$. We accepteren nu alleen maar de X_i voor de indices i waarvoor geldt dat $Y_i \leq f(X_i)$ en werpen de andere X_i weg. Het is niet moeilijk om in te zien dat de geaccepteerde toevalsgetallen X_i de dichtheidsfunctie $f(x)$ hebben, want een waarde $X_i = x$ wordt juist met kans $\frac{f(x)}{c}$ geaccepteerd.

Simulatie van speciale verdelingen

Voor een aantal belangrijke kansverdelingen geven we nu aan hoe we met behulp van een randomgenerator die toevalsgetallen U_i op het interval $[0,1)$ produceert een stochast X met deze kansverdeling kunnen simuleren.

Discrete gelijkverdeling: Voor een eindige uitkomstenruimte Ω met $|\Omega| = n$ kunnen we aannemen dat $\Omega = \{0, \ldots, n-1\}$. We krijgen een gelijkverdeling op Ω door $X := \lfloor n \cdot U_i \rfloor$, waarbij $\lfloor x \rfloor$ het grootste gehele getal is dat $\leq x$ is.

Binomiale verdeling: We kunnen algemeen een uitkomst met kans p simuleren door $X := \lfloor p+U_i \rfloor$, want $p+U_i$ is een gelijkverdeling op het verschoven interval $[p, 1+p]$ en we hebben een waarde ≥ 1 met kans p.

Voor de binomiale verdeling $b(m, p; k)$ herhalen we m keer een simulatie met kans p en krijgen: $X := \sum_{i=1}^{m} \lfloor p + U_i \rfloor$.

Hypergeometrische verdeling: Om de hypergeometrische verdeling met parameters n, m en s te simuleren volgen we in principe de procedure van een echte proef. We noemen s_i het aantal slechte stukken die voor de i-de greep nog in de verzameling zitten en $p_i = \frac{s_i}{n}$ de kans dat we in de i-de greep een slecht stuk kiezen. Onze stochast X is het aantal slechte stukken die we grijpen. We beginnen dus met $X := 0$, $s_1 := s$ en $p_1 := \frac{s_1}{n} = \frac{s}{n}$ en voeren de volgende procedure voor $i = 1, 2, \ldots, m$ uit:

> Laat $A_i := \lfloor p_i + U_i \rfloor$ dan geeft $A_i = 1$ aan dat een slecht stuk werd getrokken, en $A_i = 0$ dat geen slecht stuk werd getrokken. We zetten nu $X := X + A_i$, $s_{i+1} := s_i - A_i$ en $p_{i+1} := \frac{s_{i+1}}{n}$.

Poisson-verdeling: Als m groot is kunnen we met behulp van de simulatie van de binomiale verdeling ook de Poisson-verdeling met parameter $\lambda = m \cdot p$ simuleren. Maar hiervoor maken we beter gebruik van *Poisson-processen* die we in de volgende les uitgebreid gaan behandelen. Een Poisson-proces beschrijft gewoon de tijdstippen van gebeurtenissen die volgens een Poisson-verdeling optreden. Het cruciale punt is dat de tussentijden tussen twee gebeurtenissen van een Poisson-proces exponentieel verdeeld zijn en de parameter van deze exponentiele verdeling noemen we de *intensiteit* van het Poisson-proces. We zullen zien dat voor een Poisson-proces met intensiteit 1 het aantal waarnemingen in het tijdsinterval $[0, \lambda]$ een Poisson-verdeling met parameter λ heeft. We moeten dus het tijdsinterval $[0, \lambda]$ met exponentieel verdeelde tussentijden overdekken en tellen hoeveel tussentijden er nodig zijn. Hiervoor nemen we onafhankelijke stochasten $Y_1, Y_2, \ldots$ die exponentieel verdeeld zijn met parameter 1. Als we nu een stochast X definiëren door de eigenschap

$$\sum_{i=1}^{X} Y_i \leq \lambda < \sum_{i=1}^{X+1} Y_i$$

dan heeft X een Poisson-verdeling met parameter λ. Maar de Y_i kunnen we zo als boven gezien met behulp van een randomgenerator U_i simuleren door $Y_i := -\log(U_i)$ (de parameter van de exponentiële verdeling is 1), dus is $-\sum_{i=1}^{X} \log(U_i) \leq \lambda < -\sum_{i=1}^{X+1} \log(U_i)$ en dit is equivalent met

$$\prod_{i=1}^{X} U_i \geq e^{-\lambda} > \prod_{i=1}^{X+1} U_i.$$

We vermenigvuldigen dus exponentieel verdeelde toevalsgetallen tussen 0 en 1 tot dat het product kleiner is dan $e^{-\lambda}$, het aantal X van benodigde getallen is dan een stochast met Poisson-verdeling met parameter λ.

Normaalverdeling: Voor de normaalverdeling bestaat er behalve de wegwerpmethode nog een andere mogelijkheid om tot een efficiënte simulatie te komen. Deze methode berust op de

Centrale limietstelling: Als $X_1, X_2, \ldots$ onafhankelijke stochasten zijn met verwachtingswaarde $E(X_i)$ en variantie $Var(X_i)$, dan is de limiet

$$X := \lim_{n\to\infty} \frac{\sum_{i=1}^{n}(X_i - E(X_i))}{\sqrt{\sum_{i=1}^{n} Var(X_i)}}$$

onder zwakke verdere voorwaarden aan de X_i een stochast met standaardnormaalverdeling. In het bijzonder wordt aan de voorwaarden voldaan als alle X_i dezelfde standaardafwijking σ hebben, in dit geval convergeert $\sum_{i=1}^{n} \frac{X_i - E(X_i)}{\sqrt{n}\cdot\sigma}$ tegen de standaardnormaalverdeling.

Voor de door de randomgenerator (U_i) gesimuleerde uniforme verdeling op $[0,1)$ hebben we $E(X) = \frac{1}{2}$ en $Var(X) = \int_0^1 (x - \frac{1}{2})^2 dx = \frac{1}{12}$. Als we nu n waarden van de rij (U_i) optellen hebben we $S_n = \sum_{i=1} U_i$ en er geldt $E(S_n) = \frac{n}{2}$ en $Var(S_n) = \frac{n}{12}$. Als benadering van de standaardnormaalverdeling krijgen we dus

$$X := \sqrt{\frac{12}{n}}((\sum_{i=1}^{n} U_i) - \frac{n}{2}).$$

Deze benadering is al voor $n = 10$ heel goed en voor de meeste toepassingen voldoende.

Voorbeeld: We kijken tot slot naar een simulatie van het Monty-Hall probleem, om mensen die de theoretische argumenten niet accepteren door een experiment te kunnen overtuigen. De simulatie volgt de stappen in de show:

(1) Kies een deur A waar de auto achter staat: $A := \lfloor 3 \cdot U_i \rfloor$ (we noemen de deuren 0, 1 en 2).

(2) De kandidaat kiest een deur K: $K := \lfloor 3 \cdot U_i \rfloor$.

(3) De moderator opent een deur M. Hier zijn twee gevallen mogelijk:

   (i) $A = K$: in dit geval heeft de moderator de keuze tussen $A + 1$ en $A + 2$ (als we nummers van de deuren modulo 3 nemen) we nemen dus $M := A + \lfloor 2 \cdot U_i \rfloor + 1 \bmod 3$.

   (ii) $A \neq K$: in dit geval heeft de moderator geen keuze, hij moet de deur M openen met $M \neq A$ en $M \neq K$.

(4) Hier zijn er twee versies:

   (A) De kandidaat blijft bij zijn keuze, dus $K' = K$.

   (B) De kandidaat wisselt van keuze, dus K' zo dat $K' \neq K$ en $K' \neq M$.

(5) Als $K' = A$ krijgt de kandidaat de auto, anders alleen maar de geit.

Dit kunnen we voor de versies A en B in stap (4) op een computer heel makkelijk 10000 keer doorspelen. Na drie herhalingen voor beide versies krijgen we bijvoorbeeld 3319, 3400 en 3327 successen voor versie A en 6583, 6655 en 6675 successen voor versie B.

Het blijkt dus ook uit het experiment dat het verstandig voor de kandidaat is om van keuze te wisselen.

Belangrijke begrippen in deze les

- maximum likelihood schatting
- simulatie
- randomgenerator, toevalsgetallen
- methode van de inverse verdelingsfunctie
- wegwerp methode
- centrale limietstelling

Opgaven

21. Voor een gebeurtenis dat volgens een normaalverdeling met dichtheidsfunctie

$$f(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

optreedt, zijn de observaties $x_1,\ldots,x_n$ gemaakt.

(i) Bepaal de maximum-likelihood schatting voor de verwachtingswaarde μ als de variantie σ^2 bekend is.

(ii) Bepaal de maximum-likelihood schatting voor de variantie σ^2 als de verwachtingswaarde μ bekend is.

(Opmerking: Als de verwachtingswaarde μ en de variantie σ^2 onbekend zijn, zijn de waarden uit (i) en (ii) de nulpunten van de partiële afgeleiden van de likelihood-functie en geven dus noodzakelijke voorwaarden voor een maximum van de likelihood-functie. Er laat zich aantonen dat men zo inderdaad een maximum vindt, dus laten zich μ en σ simultaan schatten.)

22. Laten U_1 en U_2 twee uniform verdeelde stochasten op $[0,1)$ zijn. Laat zien dat $\sqrt{U_1}$ en $\max(U_1,U_2)$ dezelfde verdeling hebben. (Dit geeft een zuinige manier om het maximum van twee uniforme kansverdeling te simuleren.)

23. Een symmetrische *driehoeksverdeling* op het interval $[-1,1]$ heeft de dichtheidsfunctie $f(x) = 1-|x| = \begin{cases} 1+x & \text{als } x<0 \\ 1-x & \text{als } x\geq 0 \end{cases}$.

Laten U en V twee stochasten zijn die uniform verdeeld zijn op het interval $[0,1)$.

(i) Laat zien dat de stochast $X_1 := U+V-1$ de boven aangegeven driehoeksverdeling als dichtheidsfunctie heeft.

(ii) Ga na dat de stochast $X_2 := U - V$ dezelfde kansverdeling als X_1 heeft. (We hebben dus twee manieren om de driehoeksverdeling met behulp van een randomgenerator te simuleren.)

(iii) Laat zien dat X_1 en X_2 covariantie 0 hebben, d.w.z. dat $E(X_1 \cdot X_2) = E(X_1) \cdot E(X_2)$ is.

(iv) Toon aan dat X_1 en X_2 *niet* onafhankelijk zijn.

24. Bedenk en beschrijf een efficiënte simulatie voor het trekken van de lottogetallen.