TMA4245 – Parallell 1

Geir-Arne Fuglstad, IMF

3. februar

Negativ binomisk fordeling

Definisjon:
La k være et positivt heltall og la $0 < p \leq 1$. En negativ binomisk fordelt stokastisk variabel oppstår når vi teller i hvilket forsøk suksess nummer k inntreffer i en Bernoulli forsøksrekke der suksessannsynlighet er p. Vi skriver f.eks $X \sim \mathrm{NBinom}(k, p)$ for å betegne en slik stokastisk variabel.

NB: Denne stokastiske variablen (X) tar verdier $\boldsymbol{x = k, k+1}$, …, men det er også veldig vanlig å se på antall fiaskoer før suksess $\boldsymbol{r}$ inntreffer og i det tilfellet tar den stokastiske variabelen (Y) verdier $\mathbf{y = 0, 1}$, …. I dette kurset bruker vi alltid "Antall forsøk" slik som beskrevet i formelsamlingen.

Definisjon:
Vi kaller $\mathrm{NBinom}(1, p)$ en geometrisk fordeling og skriver f.eks. $X \sim \mathrm{Geo}(p)$ for en geometrisk fordelt stokastisk variabel.

Egenskaper

Punktsannsynlighet:

$$f(x;k,p) = \binom{x-1}{k-1} p^k (1-p)^{x-k}, \qquad x = k, k+1, \ldots$$

Hvor $k \in \mathbb{N}$ er antall suksess og $0 < p \leq 1$ er suksessannsynlighet. Finnes i formelsamling.

Kumulativ fordeling:

$$F(x;k,p) = \sum_{i \leq x} \binom{x-1}{k-1} p^k (1-p)^{x-k}, \qquad x \in \mathbb{R}$$

har ingen forenkling, og finnes **ikke** i tabellform i formelsamling.

Forventning og varians:

$$\mathrm{E}[X] = \frac{k}{p} \qquad \text{og} \qquad \mathrm{Var}[X] = k\frac{(1-p)}{p^2}$$

VIKTIG angående Python

I Python finner dere ikke direkte $X \sim \mathrm{NBinom}(k, p)$, men i stedet

$$Y = \text{“Antall fiasko”} = X - k$$

`scipy.stats.nbinom.pmf` gir punktsannsynlighet:

$$f(y; k, p) = \binom{y+k-1}{k-1} p^k (1-p)^y, \qquad y = 0,1,\ldots$$

`scipy.stats.nbinom.cdf` gir kumulativ fordeling:

$$F(y; k, p) = \sum_{i \leq y} \binom{y+k-1}{k-1} p^k (1-p)^y, \qquad y \in \mathbb{R}$$

`scipy.stats.nbinom.rvs` eller `numpy.random.negative_binomial` gir simulering av Y. Eventuelt må man simulere basert på Bernoulli forsøksrekke eller fra cdf hvis man blir bedt om å implementere en egen metode.

Alle disse funksjonene kan selvfølgelig brukes også for vår definisjon av negativ binomisk fordeling, men man må være forsiktig med input/output. F.eks. kan man simulere $y_1, \ldots, y_n$ og så sette $x_1 = y_1 + k, \ldots, x_n = y_n + k$.

Poissonprosess og poissonfordeling

Poissonprosess:
Hvis man kan argumentere for at en gitt situasjon tilfredstiller betingelsene:

1. Antall hendelser i disjunkte intervaller er uavhengige
2. Sannsynligheten for at én (1) hendelse skjer i et lite intervall er proposjonal med lengden av intervallet
3. Sannsynligheten for to eller flere hendelser i et lite intervall er neglisjerbar

Kan vi anta at

$$N(t) = \text{“Antall hendelser opptil og inkludert tid } t\text{”}, \qquad \mathrm{t} \geq 0,$$

er en poissonprosess. Denne prosessen har en parameter $\lambda > 0$ som beskriver raten (forventet antall hendelser per tidsenhet).

NB: Samme prinsipp for områder osv. Se introvideoer for en mer matematisk tilnærming.

Definisjon:
La $\lambda > 0$ og $t \geq 0$. En poissonfordelt stokastisk variabel oppstår når vi teller antall hendelser i et tidsinterval av lengde t for en poissonprosess med rate λ. Vi skriver f.eks. $X \sim \text{Poisson}(\lambda t)$.

NB: Her er $X = N(t)$.

Egenskaper

Punktsannsynlighet:

$$f(x;\mu) = \frac{\mu^x}{x!}\,\mathrm{e}^{-\mu}, \qquad x = 0,1,\ldots$$

hvor forventningsverdi $\mu \geq 0$, og finnes i formelsamling og Python (f.eks. `scipy.stats.poisson.pmf`)

Kumulativ fordeling:

$$F(x;\mu) = \sum_{i\leq x} \frac{\mu^x}{x!}\,\mathrm{e}^{-\mu}, \qquad x \in \mathbb{R}$$

Har ingen forenkling, men finnes i tabellform i formelsamling. Se også f.eks. `scipy.stats.poisson.cdf`.

Forventning og varians:

$$\mathrm{E}[X] = \mathrm{Var}[X] = \mu$$

Simulering:
Flere muligheter (`scipy.stats.poisson.rvs` eller `numpy.random.poisson`). Se introvideo for måter å implementere selv.

Tilnærming til binomisk fordeling:
Fordelingen $\mathrm{Binom}(n, p)$ kan tilnærmes med $\mathrm{Poisson}(np)$ når n stor og p liten.

Normalfordeling

Sannsynlighetstetthet:

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right), \qquad x \in \mathbb{R},$$

hvor forventningsverdi $\mu \in \mathbb{R}$ og standardavvik $\sigma > 0$. Vi skriver f.eks. $X \sim \mathrm{N}(\mu, \sigma^2)$. Finnes i formelsamling og Python (f.eks. `scipy.stats.norm.pdf`). **NB: Noen skriver $\mathbf{N(0, \boldsymbol{\sigma})}$. Dette er typisk tilfellet i Python!!!**

Kumulativ fordelingsfunksjon:

$$F(x; \mu, \sigma^2) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(t-\mu)^2\right) \mathrm{d}t, \qquad x \in \mathbb{R},$$

har ingen analytisk løsning. Vi diskuterer tabell i formelsamling på neste side eller man kan bruke f.eks. `scipy.stats.norm.cdf`.

Forventning og varians:

$$\mathrm{E}[X] = \mu \qquad \text{og} \qquad \mathrm{Var}[X] = \sigma^2$$

Simulering:
I Python f.eks. `scipy.stats.norm.rvs` eller `numpy.random.normal`.

Utregning av sannsynligheter gjennom standardnormalfordeling

Definisjon:
Fordelingen $\mathrm{N}(0,1)$ kalles en standardnormalfordeling.

Notasjon:
For en standardnormalfordeling, la $\phi(x),\ x \in \mathbb{R}$, betegne sannsynlighetstettheten og $\Phi(x), x \in \mathbb{R}$, betegne kumulativ fordelingsfunksjon.

Utgregning av sannsynlighet:

La $X \sim \mathrm{N}(\mu, \sigma^2)$. Da kan vi regne ut sannsynligheter ved

$$P(X \leq x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

derfor finner man bare $\Phi(x) = F(x; 0,1)$ i tabellen i formelsamlingen.