STK-IN1050 - Statistikk for informatikere

11. Dynamiske prosesser

Matematisk institutt

2026-02-25

Læringsmål

- Forstå hva vi mener med *dynamiske prosesser*
- Se hvordan vi kan beskrive slike prosesser med enkle modeller
- Bruke simulering (Monte Carlo) for å:
  - approksimere sannsynligheter
  - beregne forventningsverdier og standardavvik
  - utforske usikkerhet i modeller

Hva er en dynamisk prosess?

- Et system som endrer seg over tid
  - Eksempler:
    - antall biler som passerer et punkt på en vei
    - størrelsen på dyrebestander (rovdyr og byttedyr)
    - antall smittede i en epidemi

- Mange systemer har tilfeldighet:
  - vi kan ikke forutsi nøyaktig hva som skjer
  - men vi kan beskrive *typisk* oppførsel statistisk

Hvorfor simulere?

- Noen ganger er det mulig å regne ut alt med sannsynlighetsteori
- Men ofte:
  - modellen er for komplisert
  - formlene blir for tunge
- Da kan vi:
  1. lage kode som simulerer prosessen
  2. gjenta simuleringen mange ganger
  3. bruke resultatene til å anslå
     - sannsynligheter
     - forventningsverdier
     - standardavvik, maks/min, osv.

Monte Carlo – idéen

Anta at vi vil beregne sannsynligheten

$$P(Y > c)$$

men det er vanskelig å gjøre analytisk.

Monte Carlo-tilnærming:

1. Simuler M uavhengige kopier $y_1, \ldots, y_M$
2. Tell hvor mange som er større enn c
3. Estimere:

$$\widehat{\Pr}(Y > c) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{1}\{y_m > c\}$$

Når M blir stor, blir estimatet typisk mer presist.

Problem: antall biler som passerer

Vi ser på en énfelts vei, og en linje tvers over veien.

- Antakelse: tiden mellom to biler er eksponensiellfordelt
- Vi vil beskrive **fordelingen til antall biler** som passerer i løpet av én time

Dette er et klassisk eksempel:

- Antakelsene leder til at antall biler er **Poisson-fordelt**, men her vil vi bruke simulering for å se det.

Simulere ankomsttidspunkter

Ide: start på tid $t = 0$ og:

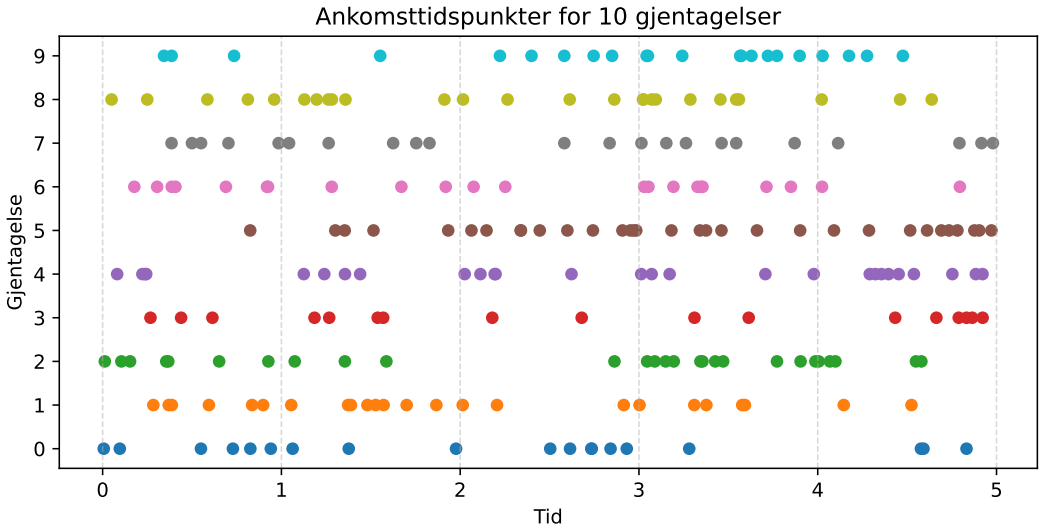
- simuler et mellomrom fra eksponensiellfordeling
- legg det til t
- hvis $t > T$ (maks tid), stopp
- ellers, lagre ankomst-tidspunktet og fortsett

```python
import numpy as np

def simulate_arrival_times(lam, T):
    """Return all arrival times up to time T."""
    t = 0.0
    arrivals = []

    while True:
        interarrival = np.random.exponential(lam)
        t += interarrival
        if t > T:
            break
        arrivals.append(t)

    return arrivals
```


Ankomsttidspunkter for 10 gjentagelser
Gjentagelse
Tid
0
1
2
3
4
5
6
7
8
9

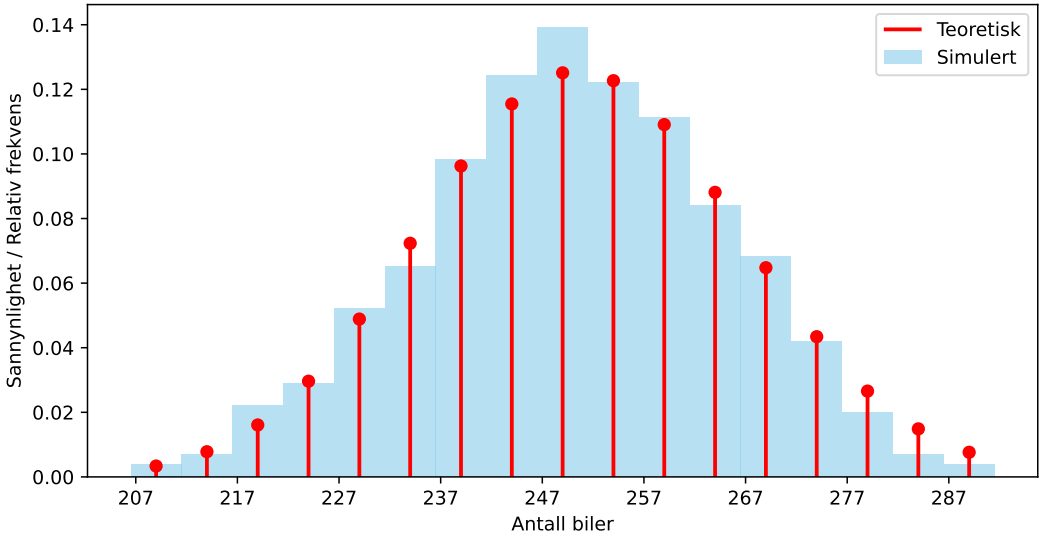
Vi trenger egentlig bare antallet ikke alle tidene.

Vi kan definere en enkel funksjon som returnerer antall biler og korteste tid mellom to biler:

```
import numpy as np

def simulate_car_count(lam, T):
    count = 0
    t = 0.0
    min_interarrival = float('inf')

    while True:
        interarrival = np.random.exponential(lam)
        t += interarrival
        if t > T:
            break
        count += 1
        if interarrival < min_interarrival:
            min_interarrival = interarrival
    return count, min_interarrival
```


Teoretisk
Simulert
Sannynlighet / Relativ frekvens
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
207
217
227
237
247
257
267
277
287
Antall biler

Rovdyr-byttedyr modell: Intuisjon

Vi ser på to populasjoner:

- $r(t)$ – antall byttedyr
- $f(t)$ – antall rovdyr

Typisk dynamikk:

- Mange byttedyr $\Rightarrow$ mye mat for rovdyr $\Rightarrow$ rovdyr øker
- Mange rovdyr $\Rightarrow$ mange byttedyr blir spist $\Rightarrow$ antall byttedyr synker
- Få byttedyr $\Rightarrow$ vanskelig for rovdyr å overleve $\Rightarrow$ antall rovdyr synker

Dette gir ofte svingende populasjoner.

En enkel stokastisk modell

La $X(t) = (r(t), f(t))$.

I et veldig lite tidsintervall $[t, t + h]$ kan det skje et av følgende:

- et byttedyr fødes
- et byttedyr dør
- et rovdyr fødes
- et rovdyr dør
- ingenting

I tillegg er der en liten mulighet for innvandring (betegnet med ε i modellen)

En enkel stokastisk modell

Vi antar altså (for små h):

$$\Pr(X(t+h) = (r',f') \mid X(t) = (r,f)) \approx \begin{cases} \alpha r + \varepsilon & (r',f') = (r+1,f) \\ \beta rf & (r',f') = (r-1,f) \\ \gamma rf + \varepsilon & (r',f') = (r,f+1) \\ \delta f & (r',f') = (r,f-1) \\ 1 - \alpha r - \beta rf - \gamma rf - \delta f & (r',f') = (r,f) \end{cases}$$

Vi antar at sannsynligheten for hver type endring er proporsjonal med

- antall individer
- eller produktet av antall individer (interaksjon)

```python
import numpy as np

def step(r, f, alpha, beta, gamma, delta, eps, rng=None):
    if rng is None:
        rng = np.random.default_rng()

    # Overgangssannsynligheter
    p_birth_r = alpha * r + eps     # byttedyr fødes
    p_death_r = beta * r * f        # byttedyr dør (blir spist)
    p_birth_f = gamma * r * f + eps # rovdyr fødes
    p_death_f = delta * f           # rovdyr dør

    # Sannsynlighet for ingen endring
    p_stay = 1 - (p_birth_r + p_death_r + p_birth_f + p_death_f)

    probs = np.array([p_birth_r, p_death_r, p_birth_f, p_death_f, p_stay])
    probs = np.clip(probs, 0, 1)
    probs = probs / probs.sum()  # normaliser
    event = rng.choice(5, p=probs)

    # Oppdater tilstand
    if event == 0:      # (r+1, f)
        r += 1
    elif event == 1:    # (r-1, f)
        r = max(r - 1, 0)
    elif event == 2: # (r, f+1)
        f += 1
    elif event == 3:    # (r, f-1)
        f = max(f - 1, 0)
    # event == 4 → ingen endring

    return r, f
```

Simulere mange steg

Vi lager en funksjon som kjører mange tidssteg og lagrer hele tidsserien.

```
def simulate(r0, f0, alpha, beta, gamma, delta, eps, steps, seed=None):
    rng = np.random.default_rng(seed)
    r, f = r0, f0

    Rs = [r]
    Fs = [f]

    for _ in range(steps):
        r, f = step(r, f, alpha, beta, gamma, delta, eps, rng)
        Rs.append(r)
        Fs.append(f)

    return np.array(Rs), np.array(Fs)
```

Eksempel: typisk dynamikk

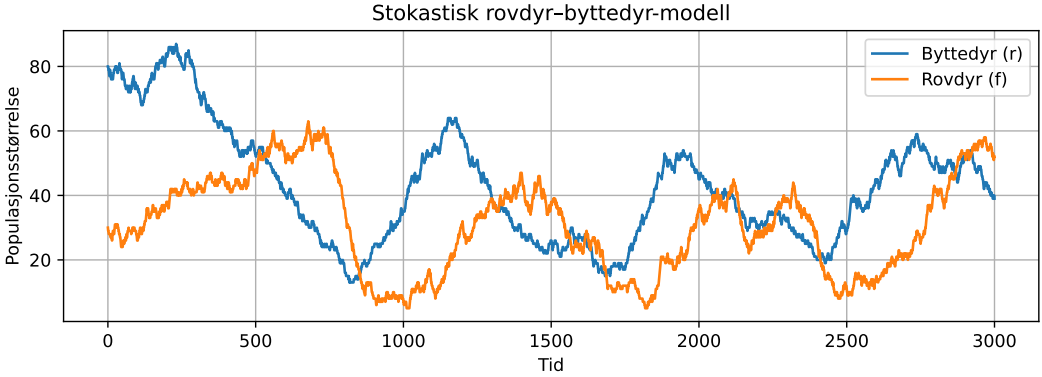
```
import matplotlib.pyplot as plt

# Eksempelparametre
alpha = 0.01
beta  = 0.0004
gamma = 0.0006
delta = 0.02
eps = 0.0   # ingen innvandring
seed = 1234

r0, f0 = 80, 30
steps = 3000

Rs, Fs = simulate(r0, f0, alpha, beta, gamma, delta, eps, steps, seed)
```

Eksempel: typisk dynamikk



- Typisk ser vi at:
  - økning i byttedyr etterfølges av økning i rovdyr
  - så faller byttedyr, deretter rovdyr, osv.

Utryddelse

Med $\varepsilon = 0$ kan begge populasjoner dø ut:

- Ved noen simuleringer vil begge arter forsvinne innenfor tidsintervallet
- Vi kan anslå sannsynligheten for utryddelse ved å:
  1. Simulere prosessen mange ganger
  2. Sjekke om $r(t)$ og $f(t)$ blir 0 i løpet av perioden
  3. Telle andelen ganger det skjer

Kompartmentmodeller

Kompartmentmodeller deler en populasjon inn i grupper:

- S – **Susceptible** (mottakelige)
- I – **Infectious** (smittsomme)
- R – **Recovered** (frisk igjen og immune)

Tilstand ved tid *t*:

$$X_t = (S_t, I_t, R_t)$$

Vi antar fast befolkningsstørrelse *N*:

$$R_t = N - S_t - I_t$$

Det holder derfor å modellere (S_t, I_t).

Kontinuerlig tid: idé

La $\Delta_t = (S(t+h) - S_t, I(t+h) - I_t)$. I et lite intervall $[t, t+h)$ kan det skje at:

- én mottakelig blir smittet ($\Delta_t = (-1, +1)$),
- én smittet blir frisk ($\Delta_t = (0, -1)$),
- eller ingenting ($\Delta_t = (0, 0)$)

La β være smittehastighet og γ være friskningshastighet. Da blir modellen:

$$\Pr(\Delta_t \mid (S_t, I_t) = (s, i)) = \begin{cases} \frac{\beta}{N} si\, h + o(h) & \Delta_t = (-1, +1) \\ \gamma i\, h + o(h) & \Delta_t = (0, -1) \\ 1 - \left[\frac{\beta}{N} si + \gamma i\right] h + o(h) & \Delta_t = (0, 0) \\ o(h) & \text{ellers} \end{cases}$$

Reproduksjonstallet R_0

En viktig størrelse er

$$R_0 = \frac{\beta}{\gamma}$$

Tolkning:

- $R_0 > 1$: sykdommen har (med høy sannsynlighet) et utbrudd
- $R_0 < 1$: sykdommen dør ut

Modellen er en **Markov-prosess**:

- framtiden avhenger bare av nåværende tilstand (S_t, I_t), ikke hele historien.

Diskret tid – en praktisk tilnærming

For simulering approksimerer vi med diskret tid:

$$
\begin{aligned}
S_{t+h} &= S_t - I_t^{\text{new}}, \quad I_t^{\text{new}} \sim \text{Binom}(S_t, p_t^I) \\
I_{t+h} &= I_t + I_t^{\text{new}} - R_t^{\text{new}}, \quad R_t^{\text{new}} \sim \text{Binom}(I_t, p_t^R) \\
R_{t+h} &= R_t + R_t^{\text{new}}
\end{aligned}
$$

med

$$
\begin{aligned}
p_t^I &= 1 - \exp(-\beta I_t h / N) \\
p_t^R &= 1 - \exp(-\gamma h)
\end{aligned}
$$

For små h er $p_t^I \approx \frac{\beta I_t}{N} h$ og $p_t^R \approx \gamma h$.

```python
import numpy as np

def simulate_epidemic(N=1000, T=500, gamma=1/7, beta=1/3, h=0.25, seed=None):
    """
    Simulerer diskret SIR-modell.

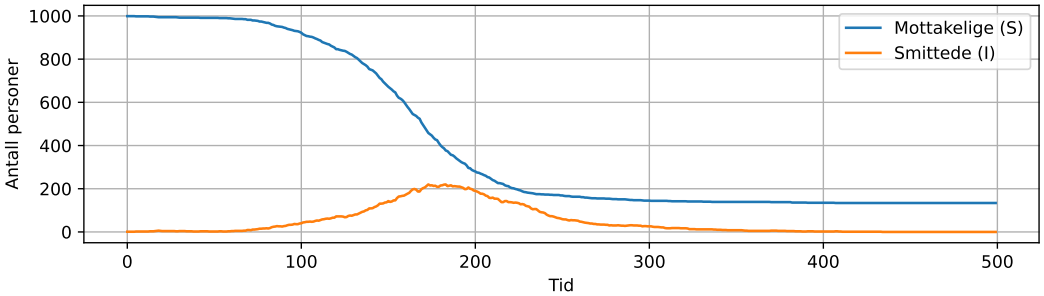
    N     = populasjonsstørrelse
    T     = antall tidssteg
    gamma = friskningsrate
    beta  = smitterate
    h     = tidssteg-lengde
    """
    if seed is not None:
        np.random.seed(seed)

    X = np.zeros((T, 2), dtype=int)
    # X[t, 0] = S_t, X[t, 1] = I_t
    X[0] = [N-1, 1]   # start med 1 smittet

    for t in range(1, T):
        S_prev, I_prev = X[t-1]
        # nye smittede og nye friske
        Inew = np.random.binomial(S_prev, 1 - np.exp(-beta * I_prev * h / N))
        Rnew = np.random.binomial(I_prev, 1 - np.exp(-gamma * h))
        S_new = S_prev - Inew
        I_new = I_prev + Inew - Rnew
        X[t] = [S_new, I_new]

    return X
```

```
X = simulate_epidemic(N=1000, T=500, seed=1234)
S = X[:, 0]
I = X[:, 1]
```



Typisk ser vi:

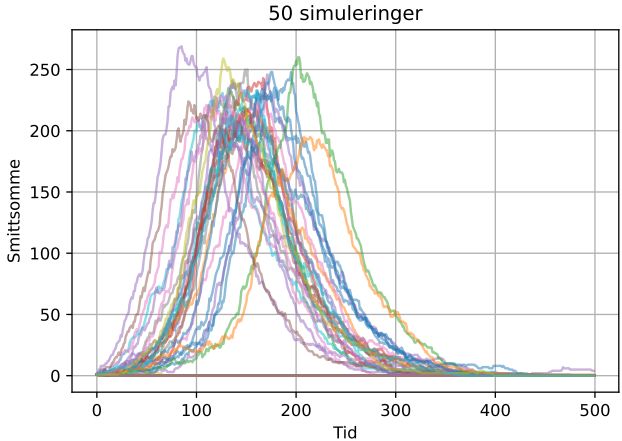
- S_t synker over tid (folk blir smittet og deretter immune)
- I_t har en topp og går så mot 0

Variasjon mellom simuleringer

Hvis vi kjører simuleringen mange ganger:

- selve formen på kurven likner
- men topphøyde og tidspunkt for toppen varierer

Modellen er stokastisk, ikke deterministisk.



Eksempel: kapasitet i helsevesenet

Tenk at:

- alle smittede trenger sykehusbehandling
- lokalt sykehus har 275 senger

Vi spør:

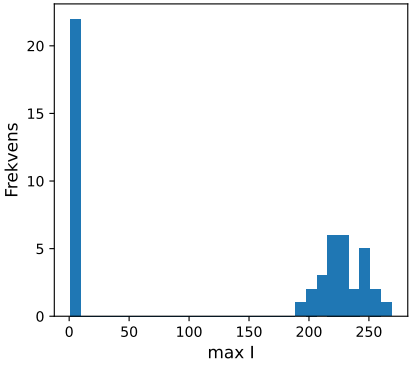
- Hva er sannsynligheten for at antall smittede noen gang overstiger 275?

Fremgangsmåte (Monte Carlo):

1. Simuler epidemien M ganger (her f.eks. 10 000)
2. For hver simulering, finn maksimum av I_t
3. Tell andelen ganger maksimum > 275

Da får vi en estimert sannsynlighet på ca. 0.0099.

Fordelingen til maksimal antall smittede, $\max_t I_t$



Vi får et histogram med en stor topp ved 0

Grunnen: vi starter med bare én smittet

- denne vil ikke smitte noen med en viss sannsynlighet
- da dør epidemien ut nesten umiddelbart

Dette illustrerer:

- simulering kan føre til kompliserte fordelinger
- «ingenting skjer» kan ha høy sannsynlighet

Oppsummering

Vi har sett på tre eksempler på dynamiske prosesser:

1. **Trafikkflyt**
   - eksponensiellfordelte ventetider
   - antall biler i et intervall $\Rightarrow$ Poisson
   - verifisert via simulering
2. **Rovdyr–byttedyr**
   - enkel stokastisk modell med fødsler/dødsfall
   - tidsserier med svingende populasjoner
   - sannsynlighet for utryddelse via simulering
3. **SIR-epidemi**
   - personer flyttes mellom S, I, R
   - enkle sannsynligheter i små tidsintervall
   - Monte Carlo for å vurdere toppbelastning og risiko

Konklusjon

- Simulering er et kraftfullt verktøy når analytiske løsninger er:
  - vanskelige
  - upraktiske
- Ved å kjøre mange simuleringer kan vi:
  - anslå sannsynligheter og forventningsverdier
  - forstå variasjon og usikkerhet