Programmering og Modellering (PoM)
Ugeseddel 13 — Uge 50 — Deadline 12/12

Kim Steenstrup Pedersen, Katrine Hommelhoff Jensen, Knud Henriksen,
Mossa Merhi og Hans Jacob T. Stephensen

November 28, 2014

1 Plan for ugen

Denne uge tager vi fat på et nyt emne: Statistisk modellering og regression. I sidste uge tog vi fat på modellering med differentialligninger. I matematisk modellering formulerer vi vores ide om et fænomen i den virkelige verden ved hjælp af matematiske udtryk. Differentialligningerne kan i denne sammenhæng anvendes til at formulere vækstmodeller og andre modeller, hvor tilstanden (f.eks. bestanden) på et bestemt tidspunkt estimeres udfra bestanden på, samt vækstraten i, estimatet af et tidligere tidspunkt. I denne uge tager vi udgangspunkt i *eksperimentiel data*, d.v.s. data indhentet fra den virkelige verden, som typisk er behæftet med *støj* og andre unøjagtigheder, og ser på, hvordan vi kan foretage matematisk modellering og statistisk analyse af sådan data. *Regression* er en form for matematisk modellering, hvor vi forsøger at tilpasse et matematisk udtryk til vores data. Hvis dette udtryk er en lineær ligning betegnes metoden *lineær regression*. Med statistisk analyse kan vi beskrive bestemte egenskaber ved data relateret til dens *middelværdi* og *spredning*, og derved forudse værdien af nye målinger samt evaluere hvor meget et datapunkt *afviger* fra resten af data. Med andre ord anvender vi statistiske egenskaber ved data til at modellere fænomenet.

Til tirsdag:

Læs Guttag kap. 15 (Understanding experimental data)
Til forelæsningen gennemgås:

- Matematisk modellering
- Eksperimentiel data
- Regressionsanalyse

Til torsdag:

Læs: Guttag kap. 12 (Stochastic programs, probability, and statistics), Massemord.pdf (noter)
Til forelæsningen gennemgås:

- Statistisk analyse og modellering
- Fordelinger, standard afvigelse, middelværdi

1.1 Gruppeopgave

Den 12/12 senest klokken 15:00 skal besvarelse af følgende opgave afleveres elektronisk via Absalon. Opgaven skal besvares i grupper bestående af 1 til 3 personer og skal godkendes, for at gruppedeltagerne kan kvalificere sig til den afsluttende tag-hjem eksamen. Opgavebesvarelsen skal oploades via kursushjemmesiden på Absalon (find underpunktet `ugeseddel13` under punktet `Ugesedler og opgaver`). Kildekodefiler ("script"-filer) skal afleveres som "ren tekst", sådan som den dannes af `emacs`, `gedit`, `Notepad`, `TextEdit` eller hvilket redigeringsprogram man nu

bruger (*ikke* PDF eller HTML eller RTF eller MS Word-format). Filen skal navngives *efternavn1.efternavn2.efternavn3*.50.py, mens andre filer skal afleveres som en PDF-fil med navnet *efternavn1.efternavn2.efternavn3*.50.pdf.

13g1 Denne opgave drejer sig om at tilnærme noget data med et matematisk udtryk, for på den måde at beskrive hvordan data "opfører" sig.

Lineær regressionsanalyse

Antag, at der foreligger en række observationer $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ af sammenhørende værdier af en såkaldt *uafhængig variabel* x og den *afhængige variable* y. Variablen y formodes at være en *lineær funktion* af x, men på grund af støj i observationerne, begrundet som *naturens tilfældigheder*, ligger de observerede punkter ikke helt på en ret linje. Antages denne støj at følge en statistisk normalfordeling, kan man med *lineær regressionsanalyse* bestemme den lineære sammenhæng, der "bedst" stemmer overens med observationerne. I denne forbindelse kan "bedst" forstås som den lineære hypotese, der gør de faktiske observationer mest sandsynlige. Statistikere kalder metoden for "maximum likelihood"-princippet.

Biblioteksmodulet `scipy` indeholder funktioner til lineær regressionsanalyse, men den her stillede opgave går ud på at programmere en sådan funktion selv, uden brug af `scipy` eller andre tilsvarende biblioteker.

Formlerne er ganske simple. Foretag disse beregninger:

$$\begin{array}{lrcl} x\text{-gennemsnittet} & \overline{x} & = & \dfrac{\sum_{i=1}^{n} x_i}{n} \\ y\text{-gennemsnittet} & \overline{y} & = & \dfrac{\sum_{i=1}^{n} y_i}{n} \\ \text{Summen af afvigelsernes kvadrater} & \text{SAK} & = & \sum_{i=1}^{n}(x_i - \overline{x})^2 \\ \text{Summen af afvigelsernes produkter} & \text{SAP} & = & \sum_{i=1}^{n}(x_i - \overline{x})y_i \\ & a & = & \dfrac{\text{SAP}}{\text{SAK}} \end{array} \tag{1}$$

Så kan den linje, som stemmer bedst overens med observationerne, beskrives med ligningen

$$y = f(x) = a(x - \overline{x}) + \overline{y} \tag{2}$$

Opgave: Udklækningstid for flueæg

I et forsøg undersøges en mulig sammenhæng mellem luftfugtigheden L (målt i %) og udklækningstiden T (målt i timer) for en bestemt type flueæg. Data findes i tabellen nedenfor:

L	100	94	88	82	76	70	64	58	52	46
T	16.6	17.4	18.3	18.2	19.6	20.2	21.7	22.0	22.7	23.2

Denne data er lagret i filen "flueaeg.txt", som kan findes på kursushjemmesiden i underkataloget "Datamateriale" under "Undervisningsmateriale'. I skal nu udvikle funktionaliteten til at foretage lineær regressionsanalyse på dataen og samtidig opfylde visse designkrav til koden:

(a) Skriv en klasse `Dataset` der administrerer en variabel med observationsdata, `dataPoints`. Den interne repræsentation af `dataPoints` er valgfri (f.eks. hashtabel, liste). Klassen skal indeholde metoder til at sætte og udlæse `dataPoints`, samt metoder til at opslå

enkelte datapunkter (hint: gør det før valg af datastruktur / opslagsmetode klart, hvordan dataen skal bruges i regressionsanalysen). Klassen `Dataset` skal ydermere indeholde en metode `readDataPoints(filePath)`, som tager et absoult stinavn `filePath` til en fil som "flueaeg.txt", der indeholder komma-separerede data-punktsæt x_i, y_i, indlæser disse punktsæt og lagrer dem i `dataPoints`. Der skal håndteres fejl i metoderne - f.eks. skal `readDataPoints(filePath)` kunne detektere og give den rette respons på et forkert dataformat.

(b) Skriv en klasse `Regression` der administrerer en variabel `data` af typen `Dataset` og som tilgår dataen i `data` v.h.a. `Dataset` klassens metoder. Klassen `Regression` skal indeholde en metode `linearAnalysis()`, der foretager en lineær regresionsanalyse på punktsættene i `data`. `linearAnalysis()` skal beregne og returnere listeparret `([`$x_{\min}$`,` $x_{\max}$`], [`$f(x_{\min})$`,` $f(x_{\max})$`])`, idet $x_{\min}$ og $x_{\max}$ betegner henholdsvis den mindste og den største af værdierne $x_1, \ldots, x_n$ i `data`, og f er beregnet i henhold til (1) og (2). Skriv formlerne i (1) som hjælpemetoder til `linearAnalysis()` i klassen `Regression`. De skal skrives på en sådan form, at kode genbruges til flere formler. Der skal håndteres fejl i metoderne - f.eks. skal `linearAnalysis()` kunne håndtere hvis `data` ikke er blevet initialiseret, eller ikke indeholder nok punkter til en regressionsanalyse.

(c) Skriv funktionalitet til at plotte en linie i det format der returneres af `linearAnalysis()`, ovenpå punksæt som lagret i klassen `Dataset`. Hvordan denne funktionalitet interagerer med klasserne `Dataset` og `Regression`, og i hvilken form den implementeres, er valgfrit. Forklar jeres designvalg (hint: overvej generalitet og genbrug af funktionalitet).

(d) Overfør filen "flueaeg.txt" til jeres egen computer. Anvend koden fra (a - c) til at indlæse punkterne fra "flueaeg.txt" samt beregne og plotte regressionslinien for disse punkter. Vis plottet i afleveringen og kommenter: Hvad viser regressionslinien?

(e) Foretag en grundig afprøvning af koden og dokumenter afprøvningen.

1.2 Tirsdagsøvelser

Besvarelser af disse opgaver skal ikke afleveres, men opgaverne forventes løst inden torsdag.

13ti1 Når man analyserer data er det ofte praktisk at lave et *histogram*. Et histogram er et plot af antallet af data-punkter som falder indenfor et bestemt interval. Hvis der f.eks. analyseres et datasæt som består af vægten af insekter fundet på en mark, vil man måske gerne vide hvor mange insekter som har en vægt mellem 0.1 gram og 0.2 gram, mellem 0.2 gram og 0.3 gram og så videre. Skriv en funktion `valueIntervCount` som tager en liste `data` af insektvægte og en liste `interv` af værdi-intervaller, og returnerer en liste som indeholder antallet af insektvægte i hvert interval. Hvis vi f.eks. har

```
data = [.01,.14,.15,.21,.01,.11,.25,.32,.35,.37,.39,.45]
interv = [.1,.2,.3,.4,.5]
```

så skal `valueIntervCount(data, interv)` returnere listen `[3, 2, 4, 1]`. Et histogram kan også afbildes direkte fra data, i Python. Plot histogrammet ved at kalde

```
import matplotlib.pyplot as plt
plt.hist(data, interv)
plt.show()
```

Stemmer plottet overens med resultatet af `valueIntervCount`?

13ti2 Vi skal se på en matematisk model for væksten af to bestande som har en direkte indvirkning på hinanden. På en eng findes en stor bestand af kaniner; kaninerne spiser græsset og reproducerer hurtigt. Den samme eng forsørger også en bestand af ulve, som lever af kaninerne. Som kaninbestanden vokser, vokser ulvebestanden ligeledes indtil der er så mange ulve at de overspiser kaninerne og ulvebestanden igen reduceres. Idet ulvebestanden er reduceret, kan kaninbestanden igen vokse, og igen ligeledes med ulvebestanden. Dette fænomen er blevet

studeret af økologer som konkluderer, at bestandene er cykliske; der er aldrig en stabilisering i bestandenes størrelse. Man formulerede to differentialligninger som varetager størrelsen af de to interagerende bestande:

$$\frac{dR}{dt} = R_g R - R_d RW \tag{3}$$

$$\frac{dW}{dt} = W_g RW - W_d W \tag{4}$$

hvor R er antal kaniner ved starttidspunktet, R_g kaninernes vækstrate og R_d kaninernes dødsrate, og W er antal ulve ved starttidspunktet, W_g ulvenes vækstrate og W_d ulvenes dødsrate. Det vil sige at $\frac{dR}{dt}$ beskriver ændringen i kaninbestanden og $\frac{dW}{dt}$ ændringen i ulvebestanden på tiden t. Skriv nu en funktion `population(R, W, Rg, Wg, Rd, Wd, maxt)` der som inddata tager startbestandene `R` og `W`, vækstraterne `Rg` og `Wg`, dødsraterne `Rd` og `Wd` samt en maksimaltid `maxt`, og som returnerer to lister der beskriver populationen af h.h.v. kaniner og ulve for hver tid $1, 2, \ldots,$ `maxt`. Populationerne i hvert tidsskridt beregnes v.h.a. den afledte fra forrige tidsskridt d.v.s. $f(t+\Delta t) = f(t) + \frac{df}{dt}(t)\Delta t$. Plot graferne for resultatet af `population(40, 15, 0.1, 0.005, 0.01, 0.1)` og diskuter resultatet.

1.3 Torsdagsøvelser

Besvarelser af disse opgaver skal ikke afleveres, men opgaverne forventes løst inden tirsdag i efterfølgende uge.

13to1 En kemiker vil analysere fejlraten af en mekanisme, som skulle returnere præcis 4 gram af et bestemt stof. I nedenstående tabel har hun indsamlet en mængde målinger af den faktiske vægt:

Måling	1	2	3	4	5	6	7	8	9	10
Vægt/gram	4.080	3.991	4.094	4.107	4.056	3.978	4.112	4.174	4.198	3.967

I denne opgave skal I implementere funktionalitet til at foretage en statistisk analyse af 1-dimensionel data, som i tabellen ovenfor.

(a) *Middelværdien* af data giver et estimat af dataens *forventede værdi*. Middelværdien af n datapunkter x_i er givet ved $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. Skriv en funktion `meanVal` der beregner middelværdien og brug den til at beregne middelværdien af målingerne i tabellen ovenfor.

(b) *Spredningen* af data giver et estimat af forskellen på målingerne, også beskrevet som *usikkerheden* af målingerne. Det mest anvendte spredningsmål er *standard afvigelsen* fra middelværdien, givet ved

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{5}$$

Skriv en funktion `stdDev` der beregner standard afvigelsen og brug den til at beregne standard afvigelsen af målingerne i tabellen ovenfor.

(c) Med middelværdien og standard afvigelsen kan vi beskrive dataens *normalfordeling*. Dette er en funktion som fortæller hvad sandsynligheden er for, at en (ny) måling har en bestemt værdi. Ligningen for funktionen for normalfordelingen kan udtrykkes som

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \tag{6}$$

Skriv dette udtryk som en funktion `normalDist` (hint: brug `meanVal` og `stdDev`). Anvend `normalDist` til at plotte normalfordelingen af målingerne i tabellen ovenfor. Hvad er sandsynligheden for at en måling antager værdien 4?

13to2 Middelværdien af data er den mest anvendte estimator af dataens *central tendency*, men det er ikke en såkaldt robust estimator, idet blot een meget fejlbehæftet værdi vil påvirke middelværdien meget. Istedet anvendes sommetider *medianen* $\tilde{x}$, der er givet ved den midterste værdi i rækken af n måliner, sorteret efter størrelse. Hvis n er et lige antal, så beregnes medianen som gennemsnittet af de to midterste værdier. Observer målingerne i tabellen nedenfor:

Måling	1	2	3	4	5	6	7	8	9	10
Vægt/gram	4.080	3.991	4.094	8.107	4.056	3.978	4.112	4.174	4.198	3.967

(a) Brug funktionen `meanVal` fra første opgave til at beregne middelværdien af målingerne i tabellen ovenfor.

(b) Skriv en funktion `medianVal` der beregner medianen og brug den til at beregne medianen af målingerne i tabellen ovenfor.

13to3 I en *standard normalfordeling* er middelværdien af data 0 og standard afvigelsen 1. Funktionen `normal` fra modulet `numpy.random` returnerer en tilfældig værdi fra denne distribution. Anvend `normal` til at generere 1000 tilfældige værdier.

(a) Med samme strategi som i første tirsdagsøvelse, inddel de 1000 samplede værdier i diskrete værdiintervaller (f.eks. af størrelsen 0.1) og plot histogrammet.

(b) Anvend funktionen `normalDist` fra torsdagsøvelse 1(c) til at plotte normalfordelingen af de 1000 samplede værdier. Hvordan passer det med histogrammet?