Danmarks Tekniske Universitet

Kursusnavn	Introduktion til Intelligente Systemer
Kursusnummer	02461
Hjælpemidler	Alle hjælpemidler er tilladt
Varighed	2 timer
Vægtning	Alle delopgaver vægtes ens

Indhold

Besvarelse af opgaverne

Alle spørgsmål skal besvares med et entydigt resultat, som skal angives med understregning sidst i hver besvarelse. Besvarelsen skal altid underbygges med relevante betragtninger og/eller beregninger. Det skal klart fremgå hvilke teorier og formler der tages udgangspunkt i, og alle relevante mellemregninger skal medtages.

Opgave A Clairvoyance

En ven hævder at være synsk. For at teste hans evner, foreslår du at lave et videnskabeligt forsøg hvor du tænker på en af jeres fem venner: Anna, Bettina, Christian, Dennis, eller Elinor. Din ven skal gætte hvem du tænker på, hvorefter du kan afsløre om det er korrekt, og forsøget gentages n gange.

Din ven påstår at han i hvert fald kan gætte 30% korrekt. Hvis man gætter tilfældigt vil man kun gætte 20% korrekt i gennemsnit. Efter at have udført forsøget vil I beregne et 95% konfidensinterval for proportionen af korrekte gæt, og hvis intervallet ikke indeholder 20% vil I konkludere at han er synsk.

Spørgsmål A.1: Beregn et estimat af den nødvendige stikprøve-størrelse n.

Svar A.1: Vi benytter formlen for beregning af stikprøve-størrelse

$$n = 1.96^2 \frac{p(1-p)}{e^2}$$

Her anvender vi den synske vens påståede proportion af korrekte gæt, $p = 0.3$, og da vi vil sikre at samplestørrelsen er stor nok til at kunne udelukke 20% skal bredden af intervallet være højst 10 procentpoint, $e = 0.1$. Det giver os følgende stikprøve-størrelse

$$n = 1.96^2 \frac{0.3(1-0.3)}{0.1^2} \approx \underline{81}$$

I ender med at benytte en stikprøvestørrelse på $n = 100$, og din ven gætter rigtigt i $n_x = 29$ tilfælde.

Spørgsmål A.2: Beregn et 95% konfidensinterval for proportionen, og angiv svaret som $0.29 \pm e$ hvor e er bredden af intervallet.

Svar A.2: Vi benytter formlen for et 95% konfidensinterval for en proportion og beregner

$$e = 1.96\sqrt{\frac{p(1-p)}{n}} = 1.96\sqrt{\frac{0.29(1-0.29)}{100}} \approx 0.089,$$

hvilket giver os intervallet $\underline{0.29 \pm 0.089}$.

Spørgsmål A.3: Hvis din ven havde gættet korrekt i færre end $n_x = 29$ tilfælde, ville det beregnede konfidensinterval da blive bredere, smallere, eller forblive uændret?

Svar A.3: Konfidensintervallet for en proportion er breddest for $p = 0.5$. Det kan vi huske fra formlen for beregning af sample størrelse, hvor $p = 0.5$ kan benyttes som et worst-case estimat. Det kan også ses fra formlen for bredden af intervallet, eller ved at indsætte et taleksempel, fx. $p = 0.28$ hvilket giver en interval-bredde på

$$1.96\sqrt{\frac{0.28(1-0.28)}{100}} \approx 0.088,$$

hvilket er lidt mindre end de 0.089 vi havde før. Med andre ord, hvis proportionen falder til mindre end 0.29, vil konfidensintervallet blive <u>smallere</u>.

Det beregnede interval indeholder ikke 20%, så derfor erklærer din ven, at det nu er endegyldigt bevist, at han er synsk. Men du mener dog at en så ekstra-ordinær påstand kræver ekstra-ordinær bevisførelse.

Spørgsmål A.4: Angiv mindst to måder hvorpå forsøgs-metoden kunne ændres, således at vi med større statistisk sikkerhed vil kunne fastslå om din ven er synsk.

Svar A.4: Da vi har benyttet et 95% konfidensinterval, vil den sande proportion kun i 95% af de gange vi benytter proceduren ligge inden for intervallet. Desuden er det beregnede interval ganske bredt, på grund af den forholdsvis lille sample størrelse. For at få større sikkerhed for konklusionen kan vi benytte et højere konfidens-niveau og/eller en større stikprøve-størrelse.

Opgave B Afstand mellem to sæt af tal

Et *talsæt* er en liste af tal, hvor rækkefølgen ikke har nogen betydning. En måde at måle hvor "ens" to talsæt X og Y er, kan være at beregne summen af den kvadratiske afstand fra hvert tal i sættet X til det nærmeste tal i sættet Y. Matematisk kan det skrives som

$$d(X,Y) = \sum_i \min_j \left((X_i - Y_j)^2 \right),$$

hvilket kan implementeres i Python med følgende funktion:

```python
def set_distance(X, Y):
    d = 0
    for x in X:
        d += min([(x-y)**2 for y in Y])
    return d
```

Spørgsmål B.1: Hvis de to sæt er givet ved

$$X = \{5,8,6,9,6\}, \quad Y = \{3,5,5,4,7\}$$

hvad er da afstanden $d(X,Y)$?

Svar B.1: Afstanden kan beregnes ved at finde det nærmeste tal i sættet Y for hvert tal i X og summe de kvadrerede forskelle:

$$d(X,Y) = (5-5)^2 + (8-7)^2 + (6-5)^2 + (9-7)^2 + (6-5)^2 = \underline{7}$$

Vi antager i det følgende at sættene X og Y begge indeholder n tal, og at n kan antage en vilkårlig værdi.

Spørgsmål B.2: Hvis vi tæller anvendelsen af plus (`+=`), minus (`-`) eller opløftet i anden potens (`**2`) som én operation hver, hvad er da tidskompleksiteten, $T(n)$, af algoritmen i Python-koden?

Svar B.2: For hvert tal i X udføres 1 plus-operation når tallene lægges sammen i variablen `d`, dvs. n operationer. Derudover udføres én minus-operation samt én potens-operation for hvert par af tal i X og Y, dvs. $2n^2$ operationer. Samlet får vi $\underline{T(n) = 2n^2 + n}$.

Spørgsmål B.3: Hvad er køretidskompleksiteten af Python-algoritmen i store-O notation?

Svar B.3: Da samtlige tal i de to sæt af størrelse n skal sammenlignes parvis, er kompleksiteten $\underline{O(n) = n^2}$.

En kollega foreslår at omskrive Python-koden ved hjælp af en list-comprehension, og når frem til følgende funktion, som er mere kort og kompakt, men måske også lidt sværere at læse og forstå:

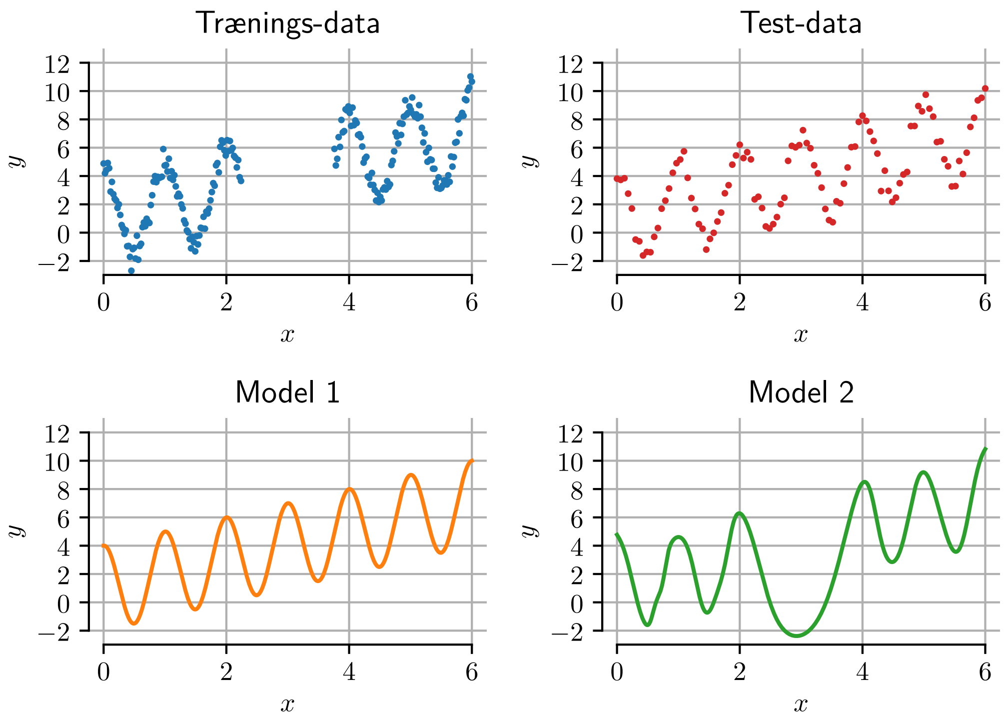
```python
def set_distance_new(X, Y):
    return sum([min([(x-y)**2 for y in Y]) for x in X])
```

Spørgsmål B.4: Sammenlignet med funktionen `set_distance`, er køretidskompleksiteten $O(n)$ af den nye funktion `set_distance_new` da større, mindre, eller uændret?

Svar B.4: Hvorvidt funktionen implementeres med en for-løkke eller en list-comprehension har ingen indflydelse på den asymptotiske køretidskompleksitet, som er <u>uændret</u>.

Opgave C Regression med en periodisk og stigende funktion

Vi ønsker at finde en model $f(x)$ der kan forudsige værdien af y for et givet punkt x. I plottet øverst til venstre ses trænings-data som vi vil benytte til at fitte modellen. Plottet øverst til højre viser test data som vi efterfølgende vil benytte til at estimere modellens generaliseringsfejl.



Da data ser ud til at være vel beskrevet ved et bias, en lineær komponent, samt en cosinus-komponent med perioden 2π, vil vi fitte følgende funktion, som vi kalder *Model 1*:

$$f(x) = w_0 + w_1 \cdot x + w_2 \cdot \cos(2\pi x)$$

Spørgsmål C.1: Hvilken konkret metode kan passende benyttes til effektivt at estimere parametrene w_0, w_1 og w_2?

Svar C.1: Da Model 1 er lineær i parametrene, kan vi passende benytte en ordinær lineær regression. Vi kan anvende den kvadratiske fejl som cost-funktion og finde parametrene ved den gængse metode med at sætte den afledede af cost-funktionen lig nul og løse de tre ligninger med tre ubekendte der kan opstilles.

Et fit af Model 1 er vist i plottet nederst til venstre, og det oplyses at amplituden af cosinus-komponenten er fundet til $w_2 = 3$.

Spørgsmål C.2: Giv ud fra figuren et estimat af parametren w_0.

Svar C.2: I punktet $x = 0$ aflæser vi $f(0) = 4$. Vi har dermed

$$f(0) = w_0 + w_1 \cdot 0 + w_2 \cdot \cos(2\pi 0) = w_0 + w_2 = 4 \Rightarrow w_0 = 4 - 3 = \underline{1}.$$

Som et alternativ har vi også fittet et neuralt netværk bestående af ét skjult lag med 50 neuroner og en hyperbolsk tangens aktiveringsfunktion, defineret med følgende Pytorch kode:

```
model2 = torch.nn.Sequential(
    torch.nn.Linear(1, 50),
    torch.nn.Tanh(),
    torch.nn.Linear(50, 1))
```

Denne model, som vi kalder *Model 2*, er fittet til data og vist i plottet nederst til højre. Model 2 er væsentligt mere fleksibel end Model 1, da den har mange flere parametre.

Spørgsmål C.3: Hvor mange parametre har Model 2?

Svar C.3: Det er oplyst at Model 2 er et neuralt netværk med ét skjult lag med 50 neuroner. Da input er en skalar, har hver neuron 2 parametre (1 vægt og 1 bias). I output-laget kombineres de 50 neuroner i en output-neuron med 50 input. Den har dermed 51 parametre (50 vægte og 1 bias) kombineres. Det samlede antal parametre er dermed

$$(1+1)\cdot 50 + (50+1) = \underline{151}$$

Spørgsmål C.4: Hvilken model har den laveste generaliseringsfejl (dvs. fejl på test-data)?

Svar C.4: Modellerne ser begge ud til at fitte trænings-data ganske fint, men de er meget forskellige i intervallet omkring $x = 3$, hvor der ikke er trænings-data til rådighed. I dette interval ses det at Model 1 stemmer godt overens med test-data hvorimod Model 2 ikke er så god. Det kan derfor konkluderes at Model 1 har den laveste generaliseringsfejl.

C

Opgave D Programmerings-prøve

Vi forestiller os en undersøgelse foretaget blandt studerende på et stort universitet: I alt 100 tilfældigt udvalgte studerende medbringer deres egen computer og tager en svær prøve i programmering. Det noteres hvilken computer-type de har medbragt og om de består prøven. Resultatet er følgende:

	Ikke bestået	Bestået	Total
Mac	40	10	50
Windows	24	24	48
Linux	0	2	2
Total	64	36	100

Resultatet viser at $\frac{10}{50} = 20\%$ af Mac-brugerne, $\frac{24}{48} = 50\%$ af Windows-brugerne og $\frac{2}{2} = 100\%$ af Linux-brugerne bestod prøven.

Spørgsmål D.1: Selv om undersøgelsen viser, at 100% af Linux-brugerne bestod, kan vi *ikke konkludere* at alle ville bestå, hvis universitetet pålagde alle at benytte Linux. Giv *to* argumenter for hvorfor vi ikke kan konkludere dette.

Svar D.1: Med kun to Linux-brugere i undersøgelsen har vi en for lille sample størrelse til at vi kan sige noget klart om beståelses-procenten blandt alle Linux-brugere i populationen.

Derudover er der ikke påvist en kausal sammenhæng mellem brug af Linux og beståelse af prøven.

Spørgsmål D.2: Er der statistisk belæg for at konkludere, at mindre end 50% af Mac-brugere blandt hele universitetes population vil kunne bestå prøven?

Svar D.2: Vi kan beregne 95% konfidens-intervallet for proportionen som

$$p_{\text{Mac}} = 0.2 \pm 1.96\sqrt{\frac{0.2\cdot(1-0.2)}{48}} \approx 0.2 \pm 0.11$$

Da konfidensintervallet ikke indeholder 50% er det klart, at der er statistisk belæg for konklusionen.

Universitetes administration laver en ny undersøgelse hvor 1000 studerende bliver bedt om at tage prøven på en computer som stilles til rådighed af universitetet. Halvdelen af de studerende udvælges tilfældigt til at tage prøven med en Mac, og den anden halvdel med en Windows computer. Resultatet af den nye undersøgelse er følgende:

	Ikke bestået	Bestået	Total
Mac	324	176	500
Windows	329	171	500
Total	650	350	1000

Spørgsmål D.3: Hvad kan du, ud fra de to undersøgelser, konkludere om den kausale sammenhæng mellem computer type og beståelse af prøven?

Svar D.3: I undersøgelsen hvor computer-typen tildeles tilfældigt er beståelsesprocenten i de to grupper næsten ens. Der kan ikke påvises nogen statistisk signifikant forskel, og sample størrelsen $n = 1000$ er forholdsvis stor. Det konkluderes, at der er ingen signifikant kausal sammenhæng mellem computer type og beståelse af prøven.

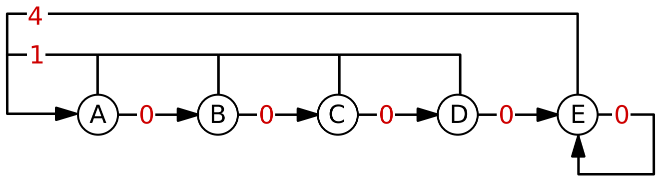
Spørgsmål D.4: Giv et eksempel på en mulig confounder, som kan forklare forskellen mellem de to undersøgelser.

Svar D.4: En confounder vil være en ikke-observeret variabel, som både er medvirkende årsag til de studerendes valg af computer-type og deres chance for at bestå. Det kunne fx. være at studerende som på forhånd har større erfaring med programmering er mere tilbøjelige til at købe en Windows computer. En mulig confounder kunne således være forudgående programmeringserfaring.

D

Opgave E Reinforcement learning

Vi betragter følgende miljø som indeholder fem tilstande (states), A–E. I hver tilstand er der to mulige actions: at gå `op` eller til `højre`. Pilene angiver de mulige transitioner mellem tilstandene, og tallene på pilene angiver de tilhørende belønninger (rewards). At gå `op` fører altid til tilsgand A.



At gå `op` fra tilstand A–D giver en belønning på 1, og at gå `op` fra tilstand E giver en belønning på 4. Alle andre belønninger er nul.

Spørgsmål E.1: Hvad er den optimale action, når man står i tilstand E?

Svar E.1: Hvis man går til `højre` i tilstand E får man en reward på nul og ender tilbage i tilstand E. Dermed vil den kumulative discountede reward være nul, uanset hvilken discount faktor der benyttes. Hvis man derimod går `op` får man en reward på 4, og ender i tilstand A, hvorfra man kan opnå yderligere positiv reward. Da der ikke er nogen negativ reward i miljøet, er det klart at der er større kumulativ discounted reward forbundet med at gå `op`. Dermed må den optimale action fra tilstand E være <u>`op`</u>.

I de følgende tre delopgaver anvender vi en discount factor på $\gamma = 0.9$. Det oplyses at værdien af tilstandene A og E er henholdsvis $v(A) = 10$ og $v(E) = 13$. Det oplyses endvidere at den optimale action, når man står i tilstand B, er at gå `op`.

Spørgsmål E.2: Hvad er værdien af tilstand B, $v(B)$?

Svar E.2: Da vi kender den optimale action, kan værdien af tilstanden kan beregnes som den kumulative discountede reward:

$$v(B) = r(B, \texttt{op}) + \gamma \cdot v(A) = 1 + 0.9 \cdot 10 = 1 + 9 = \underline{10}.$$

Spørgsmål E.3: Hvad er værdien af tilstand D, $v(D)$?

Svar E.3: Værdien kan beregnes ud fra formlen

$$v(s) = \max_a (r(s,a) + \gamma v(s'))$$

Vi indsætter for de to mulige actions, `op` og `højre`, hvilket giver os

$$\begin{aligned} v(D) &= \max\Big(r(D, \texttt{op}) + \gamma \cdot v(A), \quad r(D, \texttt{højre}) + \gamma \cdot v(E)\Big) \\ &= \max\Big(1 + 0.9 \cdot 10, \quad 0 + 0.9 \cdot 13\Big) = \max(10, \quad 11.7) = \underline{11.7} \end{aligned}$$

Spørgsmål E.4: Beskriv den optimale policy, hvis man i stedet anvender en discount faktor på $\gamma = 0$.

Svar E.4: Ved $\gamma = 0$ vil den kumulative discountede reward kun vægte den reward man får i det næste skridt, og slet ikke vægte fremtidig reward. Dermed bliver den optimale policy at tage den action der giver højest umiddelbar reward. Af figuren kan det ses at denne optimale policy er at <u>altid gå `op`</u>.