

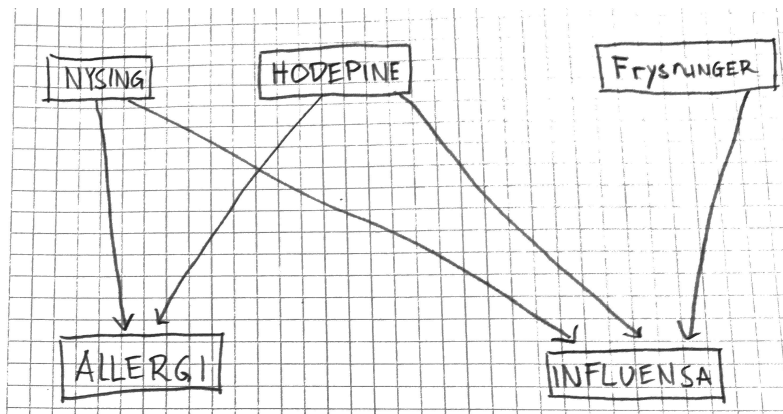
# TDT4171: Exam

10016

May 2020

## 1 Sannsynlighetsregning

1.1 Tegn Bayesiansk nettverk som inkluderer 2 sykdommer og minst 3 symptomer.



$$P(Nys) = 0,1$$

$$P(Frys) = 0,05$$

$$P(Hodepine) = 0,02$$

$$P(Nys|Allergi) = 0.75$$

$$P(Frys|Allergi) = P(Frys) = 0.01$$

$$P(Hodepine|Allergi) = 0.75$$

$$P(Nys|Influenza) = 0.5$$

$$P(Frys|Influenza) = P(Frys) = 0.75$$

$$P(Hodepine|Influenza) = 0.75$$

$$P(Allergi) = 0.002$$

$$P(Influenza) = 0.001$$

## 1.2 Bruk nettverket og sannsynligheter for å vise hvordan å finne mest sannsynlig sykdom for ett test eksempel. Ta med beregninger.

TestCase:

Nysing: True

Frying: True

Hodepine: True

We assume Naïve Bayes.

$$\begin{aligned}
P(\text{Allergi}|\text{Nys}, \text{Frys}, \text{HP}) &= \frac{P(\text{HP}, \text{Nys}, \text{Frys}|\text{Allergi}) * P(\text{Allergi})}{P(\text{HP}, \text{Nys}, \text{Frys})} \\
&= \frac{P(\text{HP}|\text{Allergi}) * P(\text{Nys}|\text{Allergi}) * P(\text{Frys}|\text{Allergi}) * P(\text{Allergi})}{P(\text{HP}) * P(\text{Nys}) * P(\text{Frys})} \\
&= \frac{P(\text{HP}|\text{Allergi}) * P(\text{Nys}|\text{Allergi}) * P(\text{Frys}) * P(\text{Allergi})}{P(\text{HP}) * P(\text{Nys}) * P(\text{Frys})} \\
&= \frac{P(\text{HP}|\text{Allergi}) * P(\text{Nys}|\text{Allergi}) * P(\text{Allergi})}{P(\text{HP}) * P(\text{Nys})} \\
&= \frac{0.75 * 0.75 * 0.05}{0.02 * 0.1} \\
&= 14
\end{aligned}$$

$$\begin{aligned}
P(\text{flu}|\text{Nys}, \text{Frys}, \text{HP}) &= \frac{P(\text{HP}, \text{Nys}, \text{Frys}|\text{flu}) * P(\text{flu})}{P(\text{HP}, \text{Nys}, \text{Frys})} \\
&= \frac{P(\text{HP}|\text{flu}) * P(\text{Nys}|\text{flu}) * P(\text{Frys}|\text{flu}) * P(\text{flu})}{P(\text{HP}) * P(\text{Nys}) * P(\text{Frys})} \\
&= \frac{0.75 * 0.5 * 0.75 * 0.001}{0.02 * 0.1 * 0.05} \\
&= 3
\end{aligned}$$

I have obviously screwed up the starting probabilities but nevertheless we find here that of the two, allergies is the most likely explanation.

### 1.3 Diskuter om Bayesiansk nettverk passer til å representere den type kunnskap? Er det en alternativ modell som kunne passe her? Hva er fordeler, ulemper med den?

Bayesiansk nettverk fungerer bra med denne typen kunnskap der vi har en kausal historie. her har vi sannsynligheter ved hver av symptomene og sykdommene og kondisjonelle sannsynligheter. Derfor har vi en meget god basis for et bayes net.

Vi kunne alternativt brukt CBR om vi har en viss domenekunnskap. Dette krever også at våre innstanser er gitt som caser. Hvis input er gitt som rådata er derimot ikke CBR passende, og vi må bruke noen form for ML. I en slik situasjon er nevralt nett lite passende da innblikk i prosessen er etisk sett nødvendig. Mer passende gren av ML er derimot decision trees. Disse er lett forståelig og transparent. Ettersom det ikke er noen sammeligning involvert, og det generelt betraktes få attributter i dette, er decision trees godt passende i en slik situasjon.

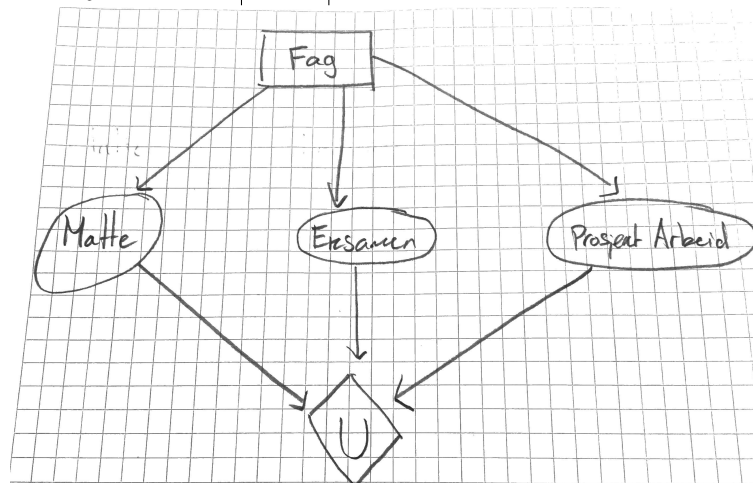
## 2 Rasjonell Beslutningstaking

Attributter og deres nytteverdidbidrag:

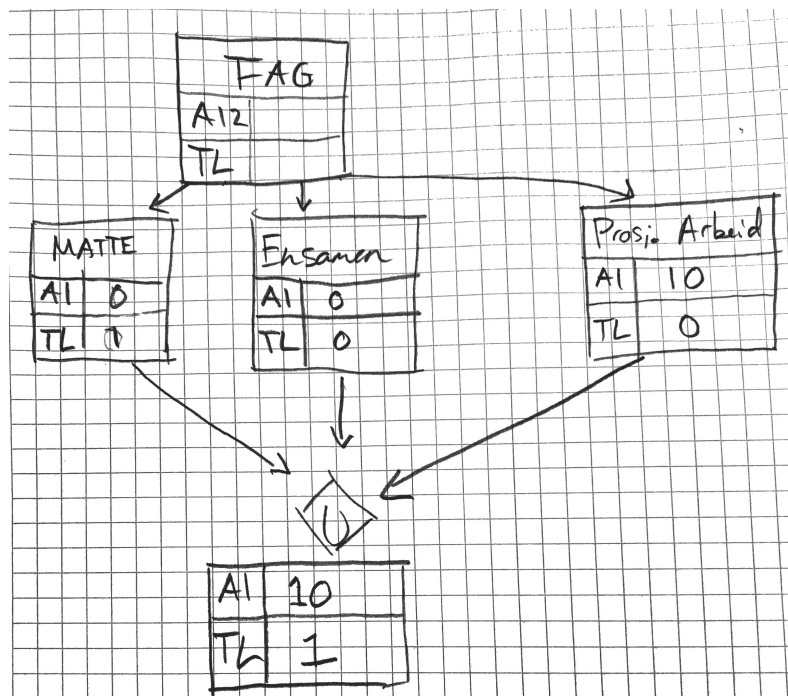
	T	F
Mattetung	0	1
Eksamen	0	3
Prosjektarbeid	0	10

Eksempelfag og deres sannsynlighet for True på hver av attributtene:

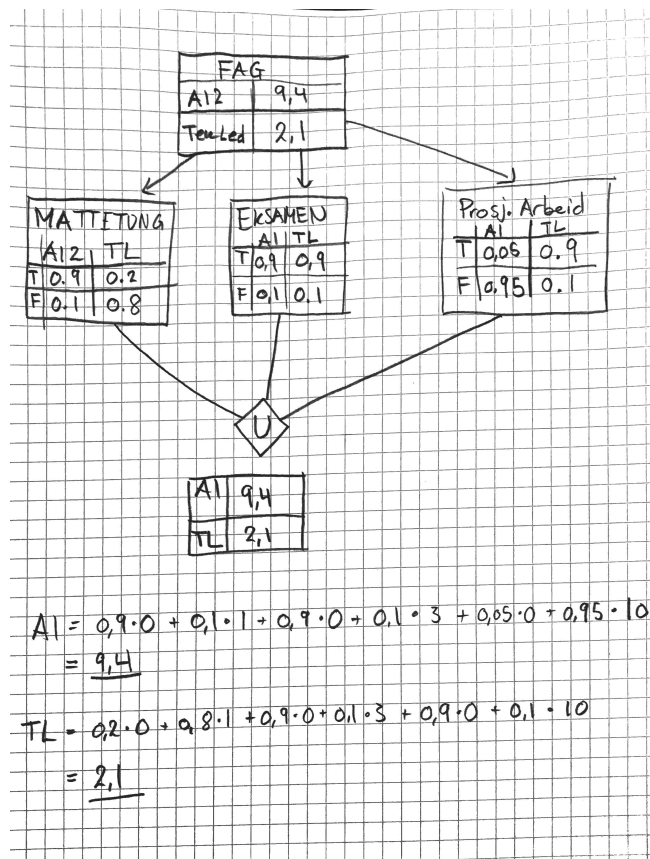
	AI2	TekLed
Mattetung	0.9	0.2
Eksamen	0.9	0.9
Prosjektarbeid	0.05	0.9



Man gjør resonnering ved å bruke modellen på fagene og anser hvilket fag har høyest nytteverdi (utility) i nytteverdinoden (utility node).



Her resonnerer vi oss fram ved bruk av den kalkulererte nytteverdien ved å se hvilket fag som gir oss mest nytte basert på hvordan vi vekter nytte i attributtene. Ved bruk av vektingen gitt tidligere i oppgaven ser vi at det er mer nytteverdi i å velge AI2 over Tekled.



Ved manglende attributter kan vi bruke domenekunnskap for å vekte sannsynlighet for en attributt. Dette gjør vi for alle og får en lik anbefalt beslutning.

## 2.1 Verdier og antagelser

Verdiene brukt er gitt ovenfor både for nytteverdi av attributtene og sannsynligheter.

## 2.2 Notes

NOTE: Jeg har her brukt en ikke-deterministisk tilnærming til fagenes attributter da jeg anser det ikke til å være binært hverken et fag kommer til å være mattetungt men heller en sannsynlighet som baserer seg på mange aspekter som f.eks. lærer, øvingslager og om det oppstår en pandemi i løpet

av semesteret.

NOTE(1,5): Det ble litt kluss i at jeg ikke helt forsto om den skulle være deterministisk eller ikke. I utgangspunktet gikk jeg for ikke-deterministisk som er den nederste av figurene, men når spørsmålet om manglende verdier kom ble jeg usikker og lagde en binær versjon i tillegg. Hvis jeg skal stå ved en blir det den ikke-deterministiske.

NOTE2: Jeg har brukt AI2 og AI om hverandre, jeg har også brukt TL og Tekled om hverandre.



### 3 Sekvensielle Beslutninger

#### 3.1 Representer problemet som Markov beslutningsprosess. Bruk tabeller for å definere sannsynligheter og belønninger.

*NOTE: En andre ordens markov modell ble for tungvint å lage for den tiden jeg hadde igjen. Så her er det samme som om det skulle være en 1. ordens.*

Her må vi definere en Transisjonsmodell både for søk om jobb til Svak og sterk, states, rewards.

The states are defined by the persons jobb, where 0: no job, 1: SvakAI, 2: SterkAI

	Job	Reward				
	0	0				
Rewards are defined as follows:	1	1				
	2	5				
				0	1	2
	0	0.2	0	0.01		
Transition model for applying to job 1:	1	0.8	1	0.95		
	2	0	0	0.04		
				0	1	2
	0	0.95	0.1	0		
Transition model for applying to job 2:	1	0	0.4	0		
	2	0.05	0.5	1		

#### 3.2 Vis hvordan å beregne optimal policy for å skaffe seg jobb i SterkAI. Vis beregninger for første iterasjon.

Her skal vi bruke Value iteration policy.

Value iteration policy assuming  $\gamma = 0.9$  :

$$U_0^0 = 0, U_1^0 = 1, U_2^0 = 5$$

This gives:

$$U_0^1 = 0 + 0.9 * (\max((0.8 * 1, 0.05 * 5))) = 0.72$$

$$U_1^1 = 1 + 0.9 * (\max((1 * 1, 0.4 * 1 + 0.5 * 5))) = 3.61$$

$$U_2^1 = 5 + 0.9 * (\max((0.95 * 1 + 0.04 * 5, 1 * 5))) = 9.5$$

### 3.3 Definer antagelser som er nødvendig for å løse problemet.

**2. ordens Markov Antagelse:** Fremtiden er uavhengig fortiden gitt nåværende state og forrige state.

**Stasjonærhet:** Transition modellen er lik for alle verdier  $t$

**Effektivt helt observerbar verden:** Vi må kjenne til hele "verdens" tilstand til enhver tid.

## 4 Case-Basert Resonnering – Anbefalingssystem for AI Metoder

*Lag et system som bruker case-basert resonnering for å anbefale AI metoder fra faget for å løse spesifikke virkelige problemer.*

### 4.1 Definer case struktur med minst 5 attributter.

Case structure:

Need for transparency
Data amount
Computational allowance
Available causality
Markovian Assumption

There must also be a solution included in the case.

### 4.2 Lag 5 eksempel caser.

case 1:	Need for transparency	No
	Data amount	Small
	Computational allowance	Large
	Available causality	No
	Markovian Assumption	Yes
	Solution	MDP
case 2:	Need for transparency	Yes
	Data amount	Small
	Computational allowance	Large
	Available causality	No
	Markovian Assumption	No
	Solution	Decision Tree
case 3:	Need for transparency	Yes
	Data amount	Medium
	Computational allowance	Medium
	Available causality	Yes
	Markovian Assumption	No
	Solution	BN

case 4:	Need for transparency	Yes
	Data amount	Medium
	Computational allowance	Medium
	Available causality	No
	Markovian Assumption	No
	Solution	CBR
case 5:	Need for transparency	No
	Data amount	Large
	Computational allowance	Large
	Available causality	No
	Markovian Assumption	No
	Solution	ANN

### 4.3 Lag ett problem for å teste systemet

Need for transparency	Yes
Data amount	Medium
Computational allowance	Large
Available causality	Yes
Markovian Assumption	No

This will give Decision Tree as result despite not linking up on all accounts.

### 4.4 Definer domenekunnskap som kan brukes i en av de fire steg av CBR syklus

I Retrieve kan vi bruke domenekunnskap for å definere en god similarity modell. Dette betyr at ved å vite at markov assumption er kritisk til MDP kan vi sette en slik parameters koeffisient i similaritetsmodellen settes høyt slik at for å koble til markov assumption må denne være lik til sin solution.

Dette kan også brukes i Revise delen der vi skal revidere funnet løsning til å gjelde det nye feltet.

## 4.5 Beskriv hvert steg i CBR syklus som løser test problemet. Bruk eksempel case og domenekunnskap definert tidligere

### 4.5.1 Retrieve:

Vi henter ut den casen som matcher mest med vårt problem ved bruk av vår valgte likhetsmodell.

Vi bruker vektingen:

Need for transparency	7
Data amount	3
Computational allowance	1
Available causality	5
Markovian Assumption	10

Dette gir den beste casen med 18 poeng:

Need for transparency	Yes
Data amount	Small
Computational allowance	High
Available causality	No
Markovian Assumption	No
Solution	Decision Tree

### 4.5.2 Reuse:

Her skal vi bruke løsningen gitt i den hentede casen. Denne løsningen må egentlig ”adapt”-es til vår situasjon men i den gitte problemstillingen er det sjeldent dette skal gjøres om vi bare returnerer et navn på en modell som løsning.

Vi bruker da løsningen av Decision tree i vår instans.

### 4.5.3 Revise:

Her skal vi Revidere vår løsning til å passe vårt nye problem.

Decision tree-løsningen må ikke revideres i dette tilfellet da det passer godt allerede.

### 4.5.4 Retain:

Her lærer vi den gitte casen inn i vår CaseBase om det var en suksess.

Vi setter vår case med den Decision tree som løsningen tilbake inn i CaseBase.

## **4.6 Definer antagelser og komponenter i CBR systemet som er nødvendig for å løse problemet**

### **4.6.1 Antagelser:**

- Det antas en del domenekunnskap.
- Det antas at like problemer har like løsninger.
- Det antas at om en modell er en løsning for et problem i et tidspunkt vil modellen alltid være en løsning for problemet.

### **4.6.2 Komponenter:**

- Løsningen
- Instruksjoner om hvordan lage modellen tilbudt.
- Kanskje en eksempelmodell