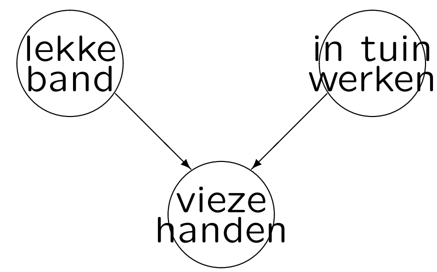
Kunstmatige Intelligentie (AI)

Hoofdstuk 12 en 13 van Russell/Norvig = [RN]
Bayesiaanse netwerken



voorjaar 2021
College 13, 10 mei 2021

www.liacs.leidenuniv.nl/~kosterswa/AI/bayesnet.pdf

We gaan nu eenvoudige “systemen” bekijken waarin kansen een rol spelen, in het bijzonder Bayesiaanse netwerken waarin vele (on)afhankelijkheden gelden.

Basisbouwsteen is de uit de kansrekening bekende regel van Bayes.

We gebruiken dus veelvuldig de regel van Bayes (te bewijzen via $P(A,B) = P(A \text{ en } B) = P(A \wedge B) = P(A)P(B|A)$):

$$P(B|A) \stackrel{\text{def}}{=} \text{kans op } B \text{ gegeven } A = \frac{P(A|B)P(B)}{P(A)}$$

Een eenvoudige toepassing is als volgt. Stel, met $S \stackrel{\text{def}}{=}$ stijve nek en $M \stackrel{\text{def}}{=}$ meningitis:

$$P(S|M) = 0.5 \quad P(M) = 1/50000 \quad P(S) = 1/20$$

(die laatste twee zijn de "prior probability"). Dan:

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002 \; .$$

Diagnostische kennis $P(M|S)$ gedraagt zich minder stabiel dan causale kennis $P(S|M)$ (denk maar aan M-epidemie!).
NB: als A en B onafhankelijk zijn: $P(A,B) = P(A)P(B)$.

Met $W \stackrel{\text{def}}{=}$ whiplash kunnen we de relatieve "likelihood" van meningitis en whiplash, gegeven een stijve nek, berekenen:

$$\frac{P(M|S)}{P(W|S)} = \frac{P(S|M)P(M)}{P(S|W)P(W)} = \frac{0.5 \times 1/50000}{0.8 \times 1/1000} = \frac{1}{80} \, ,$$

wetende dat $P(S|W) = 0.8$ en $P(W) = 1/1000$.

En soms hoef je niet alles te weten:

$$P(M|S) = \alpha P(S|M)P(M) \quad P(\neg M|S) = \alpha P(S|\neg M)P(\neg M) \, ,$$

terwijl $P(M|S)+P(\neg M|S) = 1$: normalisatie, met uiteraard

$$\alpha = 1/P(S) = 1/\{P(S|M)P(M) + P(S|\neg M)P(\neg M)\} \, .$$

Bij de tandarts geldt, met $G \stackrel{\text{def}}{=}$ gaatje; $K \stackrel{\text{def}}{=}$ kiespijn:

$$P(G|K) = P(G)\ \frac{P(K|G)}{P(K)}\ .$$

En met $H \stackrel{\text{def}}{=}$ "haakje hangt" erbij:

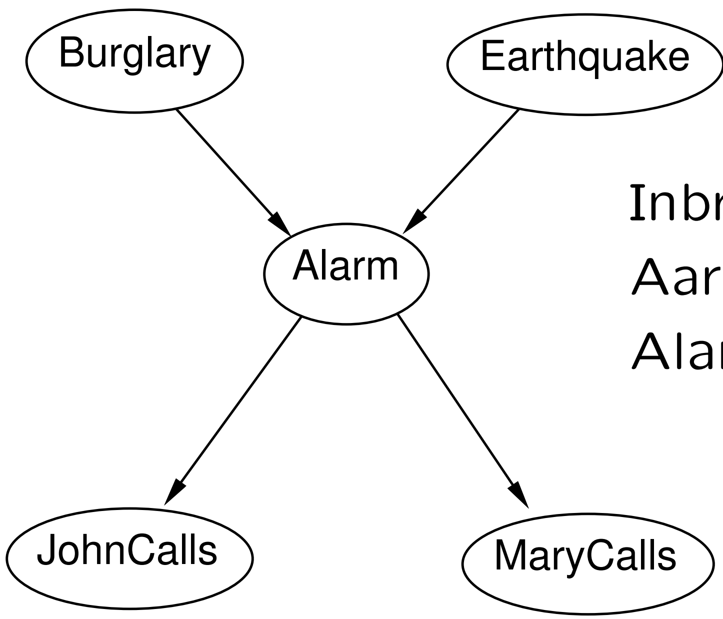
$$P(G|K \wedge H) = P(G|K)\ \frac{P(H|K \wedge G)}{P(H|K)} = P(G)\ \frac{P(K|G)}{P(K)}\ \frac{P(H|K \wedge G)}{P(H|K)}\ .$$

Een redelijke *aanname* (voorwaardelijke onafhankelijkheid) zoals $P(H|K \wedge G) = P(H|G)$ (of equivalent: $P(K|G \wedge H) = P(K|G)$) maakt het Bayesiaans updaten wat "eenvoudiger" voor de tandarts:

$$P(G|K \wedge H) = P(G)\ \frac{P(K|G)}{P(K)}\ \frac{P(H|G)}{P(H|K)}\ ,$$

waarbij je de noemers nog kunt "weg-normaliseren".

Ons standaardvoorbeeld van een Bayesiaans netwerk (oftewel probabilistisch netwerk = belief netwerk = knowledge map), afkomstig van Judea Pearl, is:

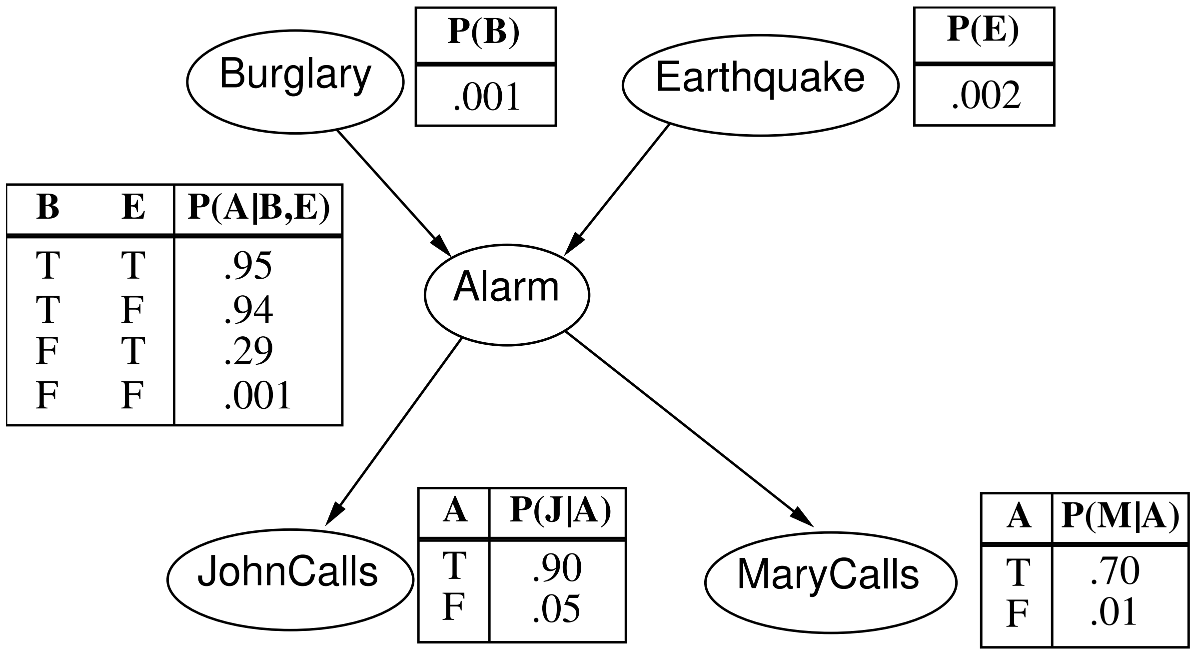


Inbreker kan Alarm af laten gaan
Aardbeving kan Alarm af laten gaan
Alarm kan Mary en/of John
aanzetten tot bellen

Een Bayesiaans netwerk is een graaf zonder cykels met:
1. random variabelen als knopen,
2. pijlen tussen knopen (die directe invloed aangeven),
3. conditional probability tables (CPT's) die het gezamenlijk effect van al hun ouders aangeven op de kinderen.

In ons voorbeeld is er *geen directe invloed* van Earthquake op MaryCalls — en dus geen pijl (andersom ook niet). Die invloed is er op zich wel, maar loopt geheel via Alarm: $P(M|A,E)(= P(M|A \wedge E)) = P(M|A)$: MaryCalls is voorwaardelijk onafhankelijk van Earthquake, gegeven Alarm. Als Mary ook rechtstreeks zou reageren op Earthquake, moest er een pijl bij.

Opnieuw ons Bayesiaanse netwerk, nu met CPT's:



Let er op dat de pijlen als het ware van oorzaak naar gevolg lopen. We hebben 10 getallen nodig om dit systeem te beschrijven, in plaats van $32 - 1 = 31$.

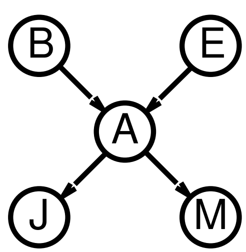
In een Bayesiaans netwerk moet altijd gelden dat

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \,|\, \text{Parents}(X_i)) \ ,$$

als we definiëren $P(x_1, \ldots, x_n) = P(X_1 = x_1, \ldots, X_n = x_n)$ en Parents(X_i) = de verzameling van knopen die een pijl naar X_i hebben.

Een voorbeeld (met $j = (\text{JohnCalls} = \text{true}), \ldots$):

$$\begin{aligned} P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) &= P(j|a)P(m|a)P(a|\neg b \wedge \neg e)P(\neg b)P(\neg e) \\ &= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 \\ &= 0.00063 \ . \end{aligned}$$



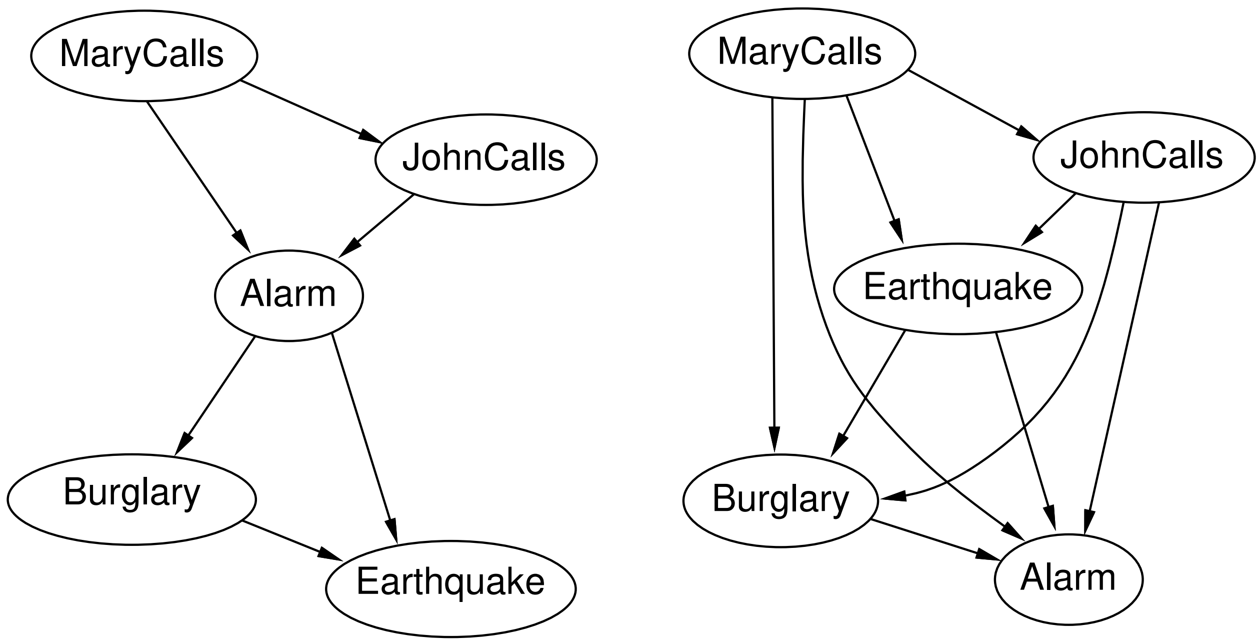
Hoe construeer je zo'n Bayesiaans netwerk?

1. Kies relevante variabelen die het domein beschrijven.
2. Orden ze "verstandig": $X_1, \ldots, X_n$.
3. Kies zo lang het kan de volgende overgebleven variabele X_i, maak daarvoor een knoop in het netwerk, en maak Parents(X_i) $\subseteq \{X_{i-1}, \ldots, X_1\}$ zo klein mogelijk met

$$P(X_i \,|\, X_{i-1}, \ldots, X_1) = P(X_i \,|\, \text{Parents}(X_i)) ,$$

de voorwaardelijke onafhankelijkheids eigenschap; definieer tot slot de CPT (conditional probability table) voor X_i.

De volgorde maakt veel uit voor het uiteindelijke netwerk:

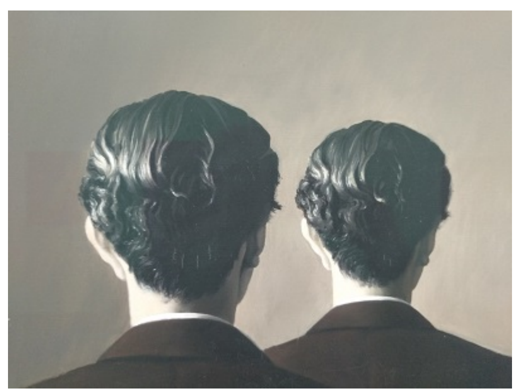


Volgorde van toevoegen links: M, J, A, B, E, rechts: M, J, E, B, A (met 31 "parameters"!). Begin dus liever met oorzaken ...

Algemeen: neem aan dat elke Booleaanse variabele rechtstreeks wordt beinvloed door maximaal k andere, dan heb je — als er n knopen zijn — aan $n \cdot 2^k$ getallen voldoende voor de CPT's.
De volledige "joint" gebruikt er $2^n - 1$ (-1 omdat de getallen tot 1 sommeren).

Met $n = 20$ en $k = 5$ is dat 640 respectievelijk 1 miljoen.

Als je *meer* weet kun je ze soms efficiënter opslaan, zeker in het geval van deterministische knopen.



In "gewone" logica geldt:

$$\text{Koorts} \Leftrightarrow \text{Verkouden} \vee \text{Griep} \vee \text{Malaria}$$

Neem nu aan dat:

(1) elke oorzaak heeft een onafhankelijke kans het effect Koorts te veroorzaken;

(2) alle oorzaken zijn gegeven (voeg eventueel een "leak node" toe);

(3) dat wat (bijvoorbeeld) Verkouden ervan weerhoudt Koorts te veroorzaken is onafhankelijk van dat wat Griep verbiedt Koorts te veroorzaken.

Samen heet dit wel een Noisy-OR relatie.

We krijgen dan een volgende tabel:

Verkouden	Griep	Malaria	P(Koorts)	P(¬Koorts)
F	F	F	0.0 (2)	1.0
F	F	T	0.9 (1)	0.1
F	T	F	0.8 (1)	0.2
F	T	T	0.98	$0.02 = 0.2 \cdot 0.1$ (3)
T	F	F	0.4 (1)	0.6
T	F	T	0.94	$0.06 = 0.6 \cdot 0.1$ (3)
T	T	F	0.88	$0.12 = 0.6 \cdot 0.2$ (3)
T	T	T	0.988	$0.012 = 0.6 \cdot 0.2 \cdot 0.1$ (3)

Alleen de drie rode getallen hoef je te onthouden, de rest volgt hier uit.

Bij de Naive Bayes classifier (liever: model; zie Data mining) neem je iets soortgelijks aan.

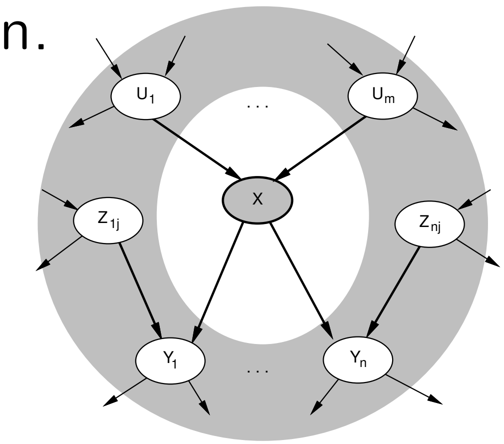
In ons Tennis-voorbeeld zou je kunnen veronderstellen:

$$\mathrm{P}(\mathrm{Weer} = \mathrm{zonnig} \wedge \mathrm{Temp} = \mathrm{koud} \mid \mathrm{Tennis} = \mathrm{Ja}) =$$
$$\mathrm{P}(\mathrm{Weer} = \mathrm{zonnig} \mid \mathrm{Tennis} = \mathrm{Ja})$$
$$\times\ \mathrm{P}(\mathrm{Temp} = \mathrm{koud} \mid \mathrm{Tennis} = \mathrm{Ja})$$

Hiermee kun je “eenvoudig” classificeren. Als namelijk het weer zonnig, temperatuur koud, vochtigheid hoog en wind sterk is, kun je een hogere kans voor het effect Tennis = Nee (benaderend) uitrekenen dan voor Tennis = Ja.

Het berekenen van voorwaardelijke kansen in een Bayesiaans netwerk (inferentie) is in het algemeen niet eenvoudig. We willen "P(Query | Evidence)" weten: we hebben informatie over zekere "Evidence" variabelen, en zijn geinteresseerd in zekere "Query" variabelen.

In het algemeen heb je te maken met de zogeheten Markov blanket: ouders, kinderen en co-ouders van kinderen.



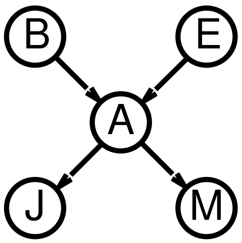
Er worden vier soorten inferentie onderscheiden:

diagnostisch van effect naar oorzaak: $P(b|j) = 0.016$

causaal van oorzaak naar effect (met de pijlen mee):
$P(j|b) = 0.86$
(via $P(j|b) = P(j|a)P(a|b) + P(j|\neg a)P(\neg a|b)$ en
$P(a|b) = P(a|b,e)P(e) + P(a|b,\neg e)P(\neg e)$)

intercausaal ("explaining away") tussen oorzaken van gemeenschappelijk effect: $P(b|a \wedge e) = 0.003$

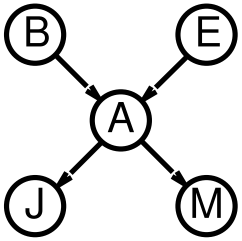
mixed overig: $P(a|j \wedge \neg e) = 0.03$



Waarbij weer $j =$ (JohnCalls $=$ true), ...

We zullen als voorbeeld van een diagnostische afleiding $P(b|j)$ berekenen:

Stap 1: $P(b|j) = P(b|a)P(a|j) + P(b|\neg a)P(\neg a|j)$
Stap 2: $P(b|a) = P(a|b)P(b)/P(a)$ (Bayes)
Stap 3: $P(a) = P(a|b,e)P(b)P(e) + P(a|\neg b,e)P(\neg b)P(e) +$
$P(a|b,\neg e)P(b)P(\neg e) + P(a|\neg b,\neg e)P(\neg b)P(\neg e) = 0.0025$
Stap 4: $P(a|b) = P(a|b,e)P(e) + P(a|b,\neg e)P(\neg e) = 0.94$
Stap 5: $P(b|a) = 0.376$ en $P(b|\neg a) = 0.00006$
Stap 6: $P(j) = P(j|a)P(a) + P(j|\neg a)P(\neg a) = 0.052$
Stap 7: $P(a|j) = P(j|a)P(a)/P(j) = 0.043$ en
$P(\neg a|j) = 0.957$
Stap 8: $P(b|j) = 0.016$



We kunnen $P(b|j)$ ook op een andere manier berekenen.

Omdat $P(b|j) = P(b,j)/P(j)$ en $P(\neg b|j) = P(\neg b,j)/P(j)$ is het voldoende $P(b,j)$ en $P(\neg b,j)$ te bepalen, en daarna te "normaliseren" — of $P(j)$ te gebruiken.

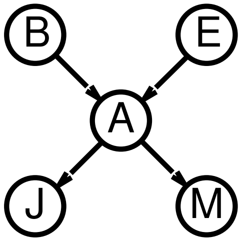
$$P(b,j) = P(b)\Big\{P(e)\{P(a|b,e)P(j|a) + P(\neg a|b,e)P(j|\neg a)\} + P(\neg e)\{P(a|b,\neg e)P(j|a) + P(\neg a|b,\neg e)P(j|\neg a)\}\Big\}\ ,$$

wat na invullen gelijk blijkt aan 0.00085.

Analoog: $P(\neg b,j) = \ldots = 0.05129$.

En dus $P(j) = P(b,j) + P(\neg b,j) = 0.05214$.

Tot slot: $P(b|j) = 0.00085/0.05214 = 0.016$.



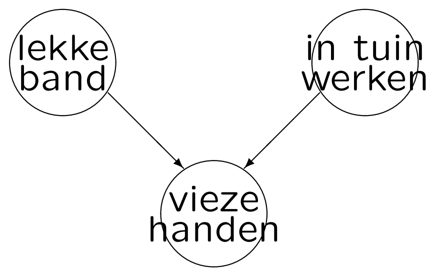
Het is blijkbaar lastig! Enkele vuistregels:

- werk toe naar "causaal"

- gebruik de uitgebreide regel van Bayes, met overal $|C$:

$$P(A|B,C) = P(B|A,C)P(A|C) \,/\, P(B|C)$$

Voorbeeld:



$$P(\text{lek}\,|\,\text{vies},\text{tuin}) = \frac{P(\text{vies}\,|\,\text{lek},\text{tuin})P(\text{lek}\,|\,\text{tuin})}{P(\text{vies}\,|\,\text{tuin})}$$

waarbij $P(\text{lek}\,|\,\text{tuin}) = P(\text{lek})$ en
$P(\text{vies}\,|\,\text{tuin}) = P(\text{vies}\,|\,\text{tuin},\text{lek})P(\text{lek}) + P(\text{vies}\,|\,\text{tuin},\neg\text{lek})P(\neg\text{lek})$

In netwerken waarbij knopen met meerdere paden verbonden zijn, is exacte inferentie nog moeilijker. In het algemeen is inferentie in een Bayesiaans netwerk zelfs NP-volledig!

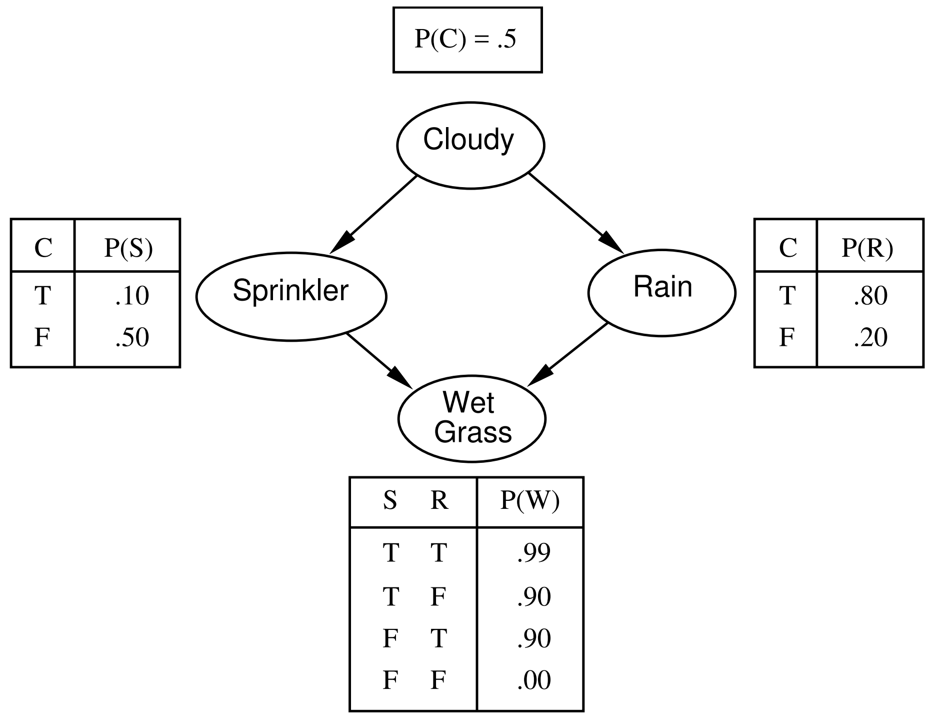
Je kunt verschillende methoden gebruiken:

clustering = join tree methoden: bouw het netwerk om naar één zonder verschillende paden: een “polytree”

conditioning methoden: maak kopieën van het netwerk door vaste waardes te kiezen voor lastige variabelen

sampling methoden: doe simulaties om een benaderend antwoord te krijgen — veel statistiek dus

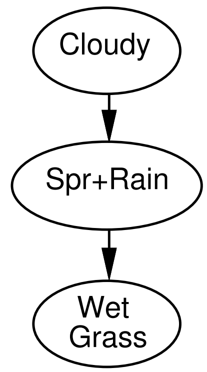
We bekijken het volgende "multiply connected" netwerk:



Hier zijn 9 getallen nodig. De onderste knoop voldoet overigens aan de "Noisy-OR" relatie.

Bij de join tree methode voegen we nu de knopen Sprinkler en Rain samen tot één mega-knoop met een grote CPT, en krijgen dan een “polytree”:

P(C) = .5



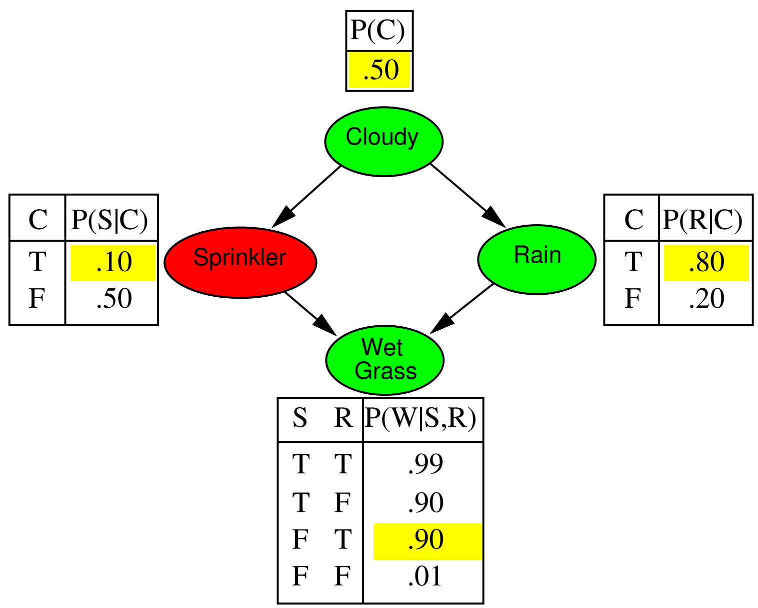
C	P(S+R=x) TT	TF	FT	FF
T	.08	.02	.72	.18
F	.10	.40	.10	.40

S+R	P(W)
T T	.99
T F	.90
F T	.90
F F	.00

Hier zijn 11 getallen nodig (de 12e en 13e volgen hieruit). In het algemeen explodeert dit aantal.

Je kunt ook (conditioning) twee netwerken maken, één voor Cloudy = true en één voor Cloudy = false: "polytrees" gelabeld met kans 0.5. Cloudy vormt de "cutset".

Of gaan "samplen":



In de laatste week, op maandag 17 mei 2021, kijken we naar het geheel en het oude tentamen van 2 juni 2014.

Denk aan de opgaven:

www.liacs.leidenuniv.nl/~kosterswa/AI/opgaven2.pdf

Doe in het bijzonder: 30, 31, 33, 13 en 14. Zie ook de video's met uitwerkingen op

www.liacs.leidenuniv.nl/~kosterswa/AI/

Werk aan de vierde opgave: Neurale netwerken; deadline: dinsdag 18 mei 2021.