Probability

$P(x,y)$ — 'x and y both occur'

- $\sum_x P(X=x) = 1.0$
- $\sum_{x,y} P(X=x, Y=y) = 1.0$
- Marginal Distribution of x over y: $\boxed{\sum_x P(X=x, Y=y) = P(y)}$

$P(x|y)$ — 'x occurs given y occured'

- $\sum_y P(x|y) = P(x)$
- $\boxed{\sum_x P(x|y) = 1.0}$

chain rule

$$\boxed{P(x,y) = P(x|y)P(y) = P(y|x)P(x)}$$

Bayes Rule

$$\boxed{P(x) = \frac{P(x|y)P(y)}{P(y|x)}}$$

Independence

$P(x) = P(x|y)$

$P(y) = P(y|x)$

Learning Y doesn't influence belief about X

... given you know z

Conditional Independence

$P(x|z) = P(x|y,z)$ &

$P(y|z) = P(y|x,z)$ &

$P(x,y|z) = P(x|z)P(y|z)$

Expected Value

$$\mathbb{E}[X] = \sum_x x P(x)$$

Bayes Net

Example

Problem Variables

A=Jesse's alarm doesn't go off
C=Jesse doesn't get coffee
G=Jesse is grumpy
L=lecture doesn't go smoothly
S=students are sad

Dependencies

$A \longrightarrow C \longrightarrow G \longrightarrow L \longrightarrow S$

conditionally independent

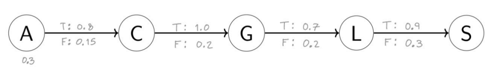
given L

PDF:

$P(A,C,G,L,S) = ?$

$P(A)P(C|A)P(G|C)P(L|G)P(S|L)$

Bayes-Net



$\mathbf{P}(\mathbf{A} = \mathbf{true}) = 0.3$

A	$P(C = true \mid A)$
t	0.8
f	0.15

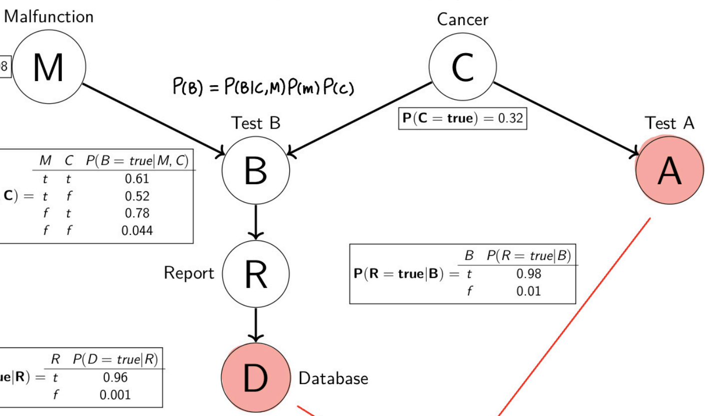
C	$P(G = true \mid C)$
t	1.0
f	0.2

G	$P(L = true \mid G)$
t	0.7
f	0.2

L	$P(S = true \mid L)$
t	0.9
f	0.3

Example

$P(C,M,B,R,D,A) = P(M)P(C)P(B|C,M)P(R|B)P(D|R)P(A|C)$



$\mathbf{P}(\mathbf{M} = \mathbf{true}) = 0.08$

$P(B) = P(B|C,M)P(M)P(C)$

$\mathbf{P}(\mathbf{C} = \mathbf{true}) = 0.32$

$P(A) = P(A|C)P(C)$

$\mathbf{P}(\mathbf{B} = \mathbf{true} \mid \mathbf{M}, \mathbf{C}) =$

M	C	$P(B = true \mid M, C)$
t	t	0.61
t	f	0.52
f	t	0.78
f	f	0.044

$\mathbf{P}(\mathbf{A} = \mathbf{true} \mid \mathbf{C}) =$

C	$P(A = true \mid C)$
t	0.80
f	0.15

Shorthand for ...

A	C	P(A|C)
T	T	0.8
T	F	0.15
F	T	0.20
F	F	0.85

$P(A|C=f)$ sums to 1

$P(A|C=t)$ Sums to 1

$\mathbf{P}(\mathbf{R} = \mathbf{true} \mid \mathbf{B}) =$

B	$P(R = true \mid B)$
t	0.98
f	0.01

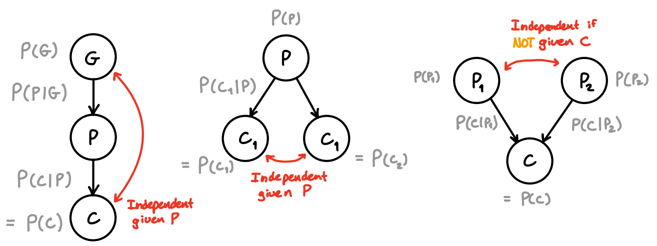
$\mathbf{P}(\mathbf{D} = \mathbf{true} \mid \mathbf{R}) =$

R	$P(D = true \mid R)$
t	0.96
f	0.001

$P(D) = P(D|R)P(R|B)P(B|C,M)P(m)P(c)$

directly observable

Building Blocks



observed variables

$P(\text{Query} \mid \text{evidence})$

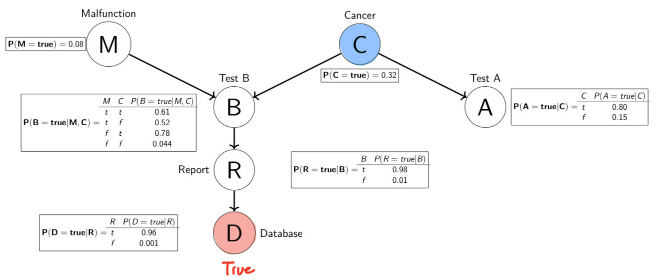
> h hypothesis
>
> Prior belief, $P(h)$
>
> Likelihood, $P(e \mid h)$
>
> Posterior Belief
>
> $$P(h \mid e) = \frac{P(h)P(e \mid h)}{P(e)} \leftarrow \sum_h P(h)P(e \mid h)$$
>
> Bayes Rule
>
> Sum Rule
>
> hard to find

---

① $\boxed{P(C = \text{true} \mid D = \text{true})}$

$$\left.\begin{aligned} P(h) &= 0.32 \\ P(e \mid h) &= P(D{=}t \mid C{=}t) \end{aligned}\right\} P(h \mid e) = \frac{P(h)P(e \mid h)}{P(e)} = 0.8$$

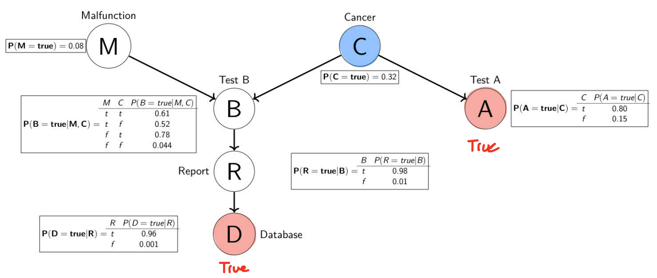
Increased



② $\boxed{P(C = \text{true} \mid D = \text{true}, A = \text{true})}$

$$\left.\begin{aligned} P(h) &= 0.32 \\ P(e \mid h) &= P(D{=}t, A{=}t \mid C{=}t) \end{aligned}\right\} P(h \mid e) = \frac{P(h)P(e \mid h)}{P(e)} = 0.95$$

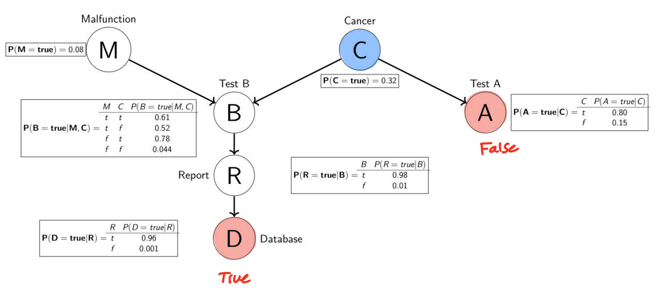
Increased

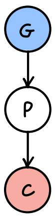


② $\boxed{P(C = \text{true} \mid D = \text{true}, A = \text{False})}$

$$\left.\begin{aligned} P(h) &= 0.32 \\ P(e \mid h) &= P(D{=}t, A{=}F \mid C{=}t) \end{aligned}\right\} P(h \mid e) = \frac{P(h)P(e \mid h)}{P(e)} = 0.48$$

Decreased



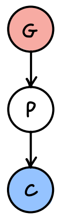


Downstream e directly affects Q

- Q influences descendants.
- observing descendant updates our belief of Q.
- need <u>bayes rule</u> to infer Q

inverts the direction of dependency

$$\underbrace{P(h|e)}_{\text{we want}} = \frac{P(h)\overbrace{P(e|h)}^{\text{we know}}}{P(e)}$$



Upstream e does not directly affect Q

- Q is cond. indep. to e given P
- don't need <u>bayes rule</u> to infer Q

$$P(C|G) = \frac{P(G|C)\,P(C)}{P(G)} = \frac{\cancel{P(G)}P(C)}{\cancel{P(G)}} = P(C)$$

Observing C doesn't change G

Below is just repeating what you should already know (from above pages)

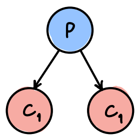
$$P(G|C) = \frac{P(C|G)\,P(G)}{P(C)}$$

marg. sum given G

$$\sum_P P(C, P|G) = \sum_P P(C|P)\,P(P|G)$$

marg. sum

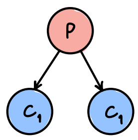
$$\sum_{P,G} P(C, P, G) = \sum_{P,G} P(C|P)\,P(P|G)\,P(G)$$



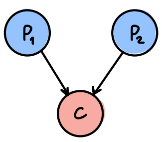
$$P(p|c_1,c_2) = \frac{P(c_1,c_2|P)\,P(p)}{P(c_1,c_2)}$$

$$P(c_1|c_2,P)\,P(c_2|p) = P(c_1|P)\,P(c_2|p)$$

$$\sum_p P(c_1,c_2\,p) = \sum_p P(c_1|p)\,P(c_2|p)\,P(p)$$



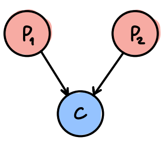
$$P(c_1,c_2|p) = P(c_1|p)\,P(c_2|p)$$



$$P(P_1,P_2|C) = \frac{P(C|P_1,P_2)\,P(P_1,P_2)}{P(C)}$$

Given Given

$$\sum_{P_1,P_2} P(C,P_1,P_2) = \sum_{P_1,P_2} P(C|P_1,P_2)\,P(P_1)\,P(P_2)$$



...

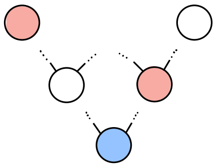
Forward Inference

descendants

$$P(Q) = \sum_{ances} P(\text{Query}, \text{ances})$$

If evidence

$$P(Q \mid e) = \sum_{ances} P(\text{Query}, \text{ances} \mid \text{Evidence})$$



Why?

Independent

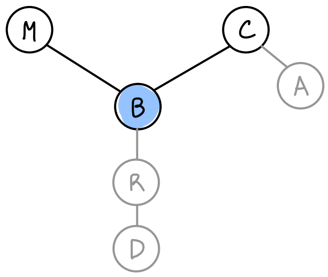
$$P(h \mid e) = \frac{P(e \mid h)P(h)}{P(e)} = \frac{P(h)\cancel{P(e)}}{\cancel{P(e)}} = P(h)$$

$$= \sum_{other,\, e} P(h, other, e)$$

$$=$$

* Works w/ or w/out evidence

1.



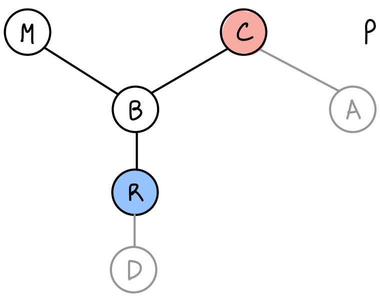
$$P(B) = \sum_{m,c} P(B, m, c)$$ marginalization - sum

$$= \sum_{m,c} P(B \mid m,c)P(m \mid c)P(c)$$ chain rule ×2

$$= \sum_{m,c} P(B \mid m,c)P(m)P(c)$$ independence

$$= \sum_{m} \left[P(m) \sum_{c} P(B \mid m,c)P(c) \right]$$

2.



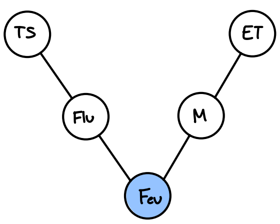
$$P(R \mid c) = \sum_{b,m} P(R, m, b \mid c)$$ marginalization - sum

$$= \sum_{b,m} P(R \mid m,b,c)\, P(b \mid m,c)\, P(m \mid c)$$ chain rule ×2

cond. indep   indep.

$$= \sum_{b,m} P(R \mid b)\, P(b \mid m,c)\, P(m)$$

* 3.



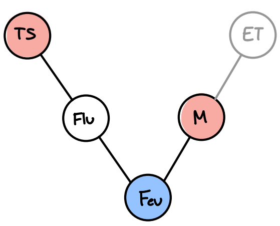
$$P(\text{Fev}) = \sum_{\text{Flu},M,TS,ET} P(\text{Fev}, \text{Flu}, M, TS, ET)$$

$$= \sum_{\text{Flu},M} P(\text{Fev} \mid \text{Flu}, M) \cdot P(\text{Flu}) \cdot P(M)$$

$$\sum_{TS} P(\text{Flu} \mid TS)P(TS) \cdot \sum_{ET} P(M \mid ET)P(ET)$$

For many Parents → Pool Evidence

4.



$$P(\text{Fev} \mid ts, m) = \sum_{\text{Flu}} P(\text{Fev}, \text{Flu} \mid ts, m)$$

$$= \sum_{\text{Flu}} P(\text{Fev} \mid \text{Flu}, ts, m)\, P(\text{Flu} \mid ts, m)$$

$$= \sum_{\text{Flu}} P(\text{Fev} \mid \text{Flu}, ts, m)\, P(\text{Flu} \mid ts)$$

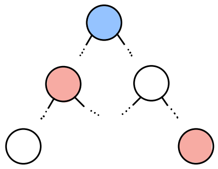
Backward Inference

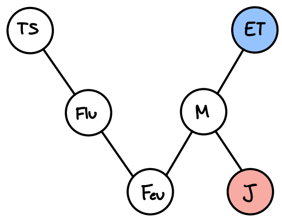
Bayes Rule

Same Idea as fwd. Infer.

$$P(\text{Query} \mid \text{evidence}) = \frac{P(Q)\,P(e \mid Q)}{P(e)}$$

marginal sum over Q



① 

$$P(ET \mid J) = \frac{P(J \mid ET)\,P(ET)}{P(J)}$$

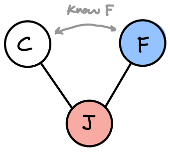
same as fwd. Infer.

$$\sum_M P(J, M \mid ET)$$

$$P(J \mid M, ET)\,P(M \mid ET)$$
$$= P(J \mid M, ET)\,P(M \mid ET)$$
$$= P(J \mid ET)\,P(M \mid ET)$$

marginal sum over query

$$= \sum_{ET} P(ET, J)$$

② 

Recall: C.I. iff. we don't know F

$$P(F \mid J) = \frac{P(J \mid F)\,P(F)}{P(J)}$$

$$= \sum_C P(J, F, C)$$

$$P(J \mid C, F)\,P(C, F)$$
$$= P(J \mid C, F)\,P(C)\,P(F)$$

Variable Elimination

works the same if Z is restricted

Goal: $P(Z \mid \underbrace{Y_1 = v_1, \ldots, Y_z = v_z}_{\text{evidence is optional}})$

Relevant Variables: only include factors w/ these variables

- Z_i's
- Parent of Relevant Var
- Y_i if any descendent is relevent

Init

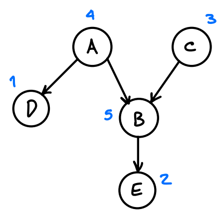
CPTs $\xrightarrow{*}$ Factors (Restrict by evidence)

Order variables & Z, Y

Complexity depends on ordering

Order Tips:

- order outside → in
- Singly connected 1st



Factor Pool

$Z_1, \ldots, Z_K$

Repeat for each Z_i in order

new factor ← $\sum_{Z_i}$(factors w/ Z_i)

Removes them from pool

Add new factor to pool

End

$f(Z) \leftarrow \prod$ remaining factors → $P(Z, \text{evidence})$

*Normalize by $\sum_Z f(Z)$ → $P(Z \mid \text{evidence})$

If needed (bayes rule used)

Making predictions w/out knowing the model.

Given:

- $d = \{(x_1,y_1),\ldots,(x_n,y_n)\}$ dataset

Find:

What model/hypothesis fits best?

For a new sample: x,

find: $P(y|x,d)$

Approaches:

① **Bayesian Learner**: Distribution over all h's

Posterior / Likelihood / Prior

$$P(h|d) \propto P(d|h)\,P(h)$$

→ $P(y|x,d) = \sum_h P(y,h|x,d)\,P(h|d)$ — Expectation over all h's

② **MAP** Approximation

$$h_{MAP} = \arg\max_h \underbrace{P(h|d)}_{\text{Posterior}}$$

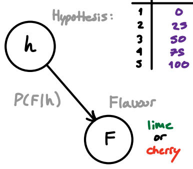
→ $P(y|x,d) = P(y|x, h_{MAP})$

② **MLE** Approximation

$$h_{MLE} = \arg\max_h \underbrace{P(d|h)}_{\text{Likelihood}}$$

→ $P(y|x,d) = P(y|x, h_{MLE})$

Candy Example



Hypothesis:

i	% lime	P(h)
1	0	0.1
2	25	0.2
3	50	0.4
4	75	0.2
5	100	0.1

Given: $d = \{d_1,\ldots,d_n\}$

Find: What's the ratio in the bag (which h_i)

Bayesian: $P(h_i|d) \propto P(d|h_i)P(h_i)$

MAP: $h_{MAP} = \arg\max_{h_i} P(h_i|d)$

MLE: $h_{MLE} = \arg\max_{h_i} P(d|h_i)$

$P(d_{n+1} = \text{Lime}|d)$

$= \sum_i P(d_{n+1} = \text{Lime}, h_i|d)\,P(h_i)$

$= P(d_{n+1} = \text{Lime}|d, h_{MAP})$

$= P(d_{n+1} = \text{Lime}|d, h_{MLE})$

---

① $d_1 = \text{Lime}$

h_i	$P(h_i)$	$P(d\|h_i)$	$P(h_i\|d)$	normal
1	0.1	0	0	0
2	0.2	0.25	0.05	0.1
3	0.4	0.5	0.2	0.4
4	0.2	0.75	0.15	0.3
5	0.1	1.0	0.1	0.2

$\Sigma = 0.5$

$P(d_{n+1} = \text{Lime}|d)$

Bayesian: $(0.1)(0.25) + (0.4)(0.5) + (0.3)(0.75) + (0.2)(1.0) = 0.65$ went up

MAP: $h_{MAP} = h_3 \Rightarrow 0.5$

MLE: $h_{MLE} = h_4 \Rightarrow 1.0$ overfit

---

② $d_2 = \text{Lime}$, $d_3 = \text{Lime}$

h_i	$P(h_i)$	$P(d_{2,3}\|h_i)$	$P(h_i\|d)$	normal
1	0	0^2	0	0
2	0.1	0.25^2	0.0625	0.013
3	0.4	0.5^2	0.1	0.211
4	0.3	0.75^2	0.1688	0.356
5	0.2	1.0^2	0.2	0.421

$P(d_{n+1} = \text{Lime}|d)$

Bayesian: $(.013)(0.25) + (.211)(0.5) + (.356)(0.75) + (.421)(1.0) = 0.79$ went up

MAP: $h_{MAP} = h_5 \Rightarrow 1.0$

MLE: $h_{MLE} = h_5 \Rightarrow 1.0$

③ $d_4 = \text{cherry}$

h_i	$P(h_i)$	$P(d_4\|h_i)$	$P(h_i\|d)$	normal
1	0	1.0	0	0
2	0.013	0.75	0.013	0.05
3	0.211	0.5	0.211	0.51
4	0.356	0.25	0.356	0.44
5	0.421	0	0	0

max for whole d

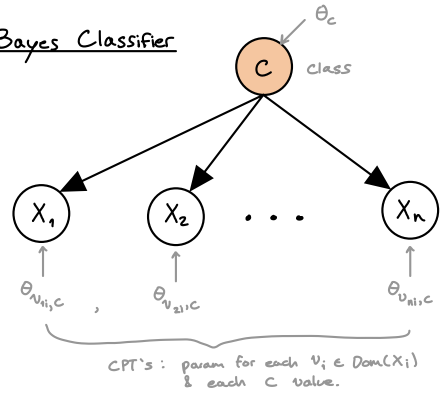
$P(d_{n+1} = \text{Lime}|d)$

Bayesian: $(.05)(0.25) + (.51)(0.5) + (.44)(0.75) = 0.60$

MAP: $h_{MAP} = h_3 \Rightarrow 0.5$

MLE: $h_{MLE} = h_4 \Rightarrow 0.75$

Naive Bayes Classifier

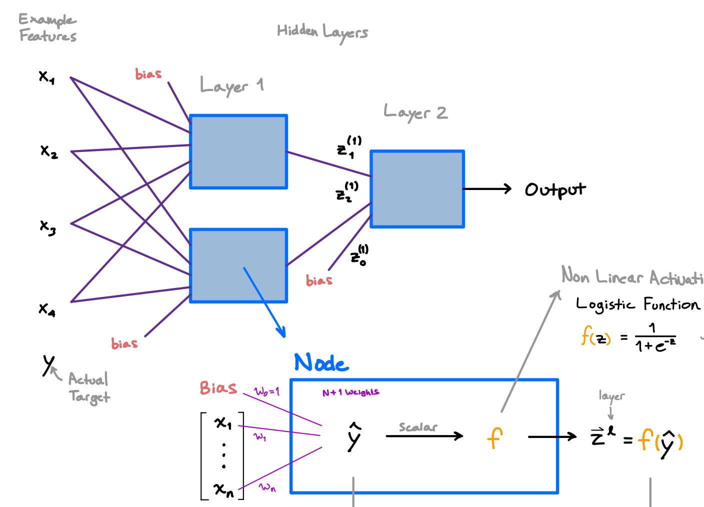


$$P(C|X_1,\dots,X_n) = P(C)\cdot P(X_1,\dots,X_n|C)$$
$$= P(C)\left[\prod_{i=1}^{n} P(X_i|C)\right]$$

iff. Independent X_i's given C

$$\theta^{MLE}_{v_{ji},c} = \frac{\#\text{ data samples w/ } X_j = v_i \ \wedge\ \text{class} = c}{\#\text{ samples w/ class} = c}$$

Neural Network



Non Linear Activation

Logistic Function

$f(z) = \frac{1}{1+e^{-z}} \xrightarrow{\frac{d}{dz}} f(z)(1-f(z))$

Linear Model

$\hat{y} = \langle \vec{w}, \vec{x} \rangle$

Error:

$Error(\vec{x}, \vec{w})$	$(\hat{y} - y)^2$	$(f(\hat{y}(\vec{x})) - y)^2$
$Error(X, \vec{w})$	$\frac{1}{n} \sum_{\vec{x} \in X} (\hat{y} - y)^2$	$\frac{1}{n} \sum_{\vec{x} \in X} (f(\hat{y}(\vec{x})) - y)^2$

Gradient:

$\frac{\partial}{\partial w_i} Error(\vec{x}, \vec{w})$	doesn't matter $2(\hat{y} - y) x_i$	$(\hat{y} - y) f'(\hat{y}) x_i$
$\frac{\partial}{\partial w_i} Error(X, \vec{w})$	$\frac{1}{n} \sum_{\vec{x} \in X} (\hat{y} - y) x_i$	$\frac{1}{n} \sum_{\vec{x} \in X} (\hat{y} - y) f'(\hat{y}) x_i$

Gradient Descent: minimize $Error(X, \vec{w})$ w/ respect to $\vec{w}$ using gradient

Repeat:

Epoch — for $\vec{x} \in X$:

weight update — for $i \leftarrow 0$ to n

$w_i \leftarrow w_i - \underset{\text{"Learn rate"}}{\eta} \boxed{\frac{\partial}{\partial w_i} Error(\vec{x}, \vec{w})}$

Variations:

Stochastic: Choose examples randomly

batch: update weights for batch of e's

For multiple layers

Forward Pass Compute $\vec{z}^{(\ell)} \rightarrow$ output layer

$$Error(z_y^{(out)}, y) = \frac{1}{n} \sum_{X} (\vec{y} - \vec{z}^{(2)})^2$$

Back Propogation compute $\delta^{(\ell)}$ for each ℓ

ℓ = output Layer:

$$\vec{\delta}^{(\ell)} = \frac{\partial Error}{\partial \vec{z}^{(\ell)}} \underset{\text{element wise}}{\times} f'(\vec{a}^{(\ell)})$$

ℓ = other Layers:

$$\vec{\delta}^{(\ell)} = (\vec{\delta}^{(\ell+1)} \underset{\text{matrix dot product}}{\cdot} W^{(\ell+1)^T}) \times f'(\vec{a}^{(\ell)})$$

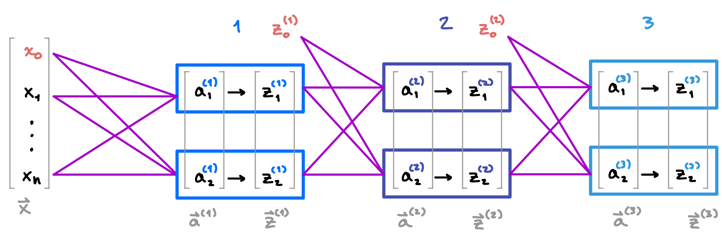
Weight Update:

ℓ = Input Layer:

$$\frac{\partial Error}{\partial W^{(\ell)}} = \vec{x}^T \delta^{(\ell)}$$

ℓ = other Layers:

$$\frac{\partial Error}{\partial W^{(\ell)}} = \vec{z}^{(\ell-1)} \delta^{(\ell)}$$



Forward Pass

$a_j^{(1)} = \vec{x} \cdot W^{(1)}$ $\quad$ $a_k^{(2)} = z_j^{(1)} \cdot W^{(2)}$ $\quad$ $a_k^{(3)} = z_j^{(2)} \cdot W^{(3)}$

$z_j^{(1)} = f(a_j^{(1)})$ $\quad$ $z_k^{(2)} = f(a_k^{(1)})$ $\quad$ $z_k^{(3)} = f(a_k^{(3)})$

Back Pass

$$\vec{\delta}_k^{(3)} = \left[-\frac{2}{n} \sum_{X} (\vec{y} - \vec{z}_k^{(3)}) \right] \times \left[f'(a_k^{(3)}) \, (1 - f'(a_k^{(3)})) \right] \longrightarrow \frac{\partial Error}{\partial W^{(3)}} = \vec{z}^{(2)} \delta^{(3)}$$

$$\vec{\delta}_k^{(2)} = \left[\underbrace{\overset{2\times1}{\delta^{(3)}} \cdot \overset{2\times3}{W^{(3)}}}_{2\times3} \right] \times \left[f'(a_k^{(2)}) \, (1 - f'(a_k^{(2)})) \right] \longrightarrow \frac{\partial Error}{\partial W^{(2)}} = \vec{z}^{(1)} \delta^{(2)}$$

$$\vec{\delta}_k^{(1)} = \left[\underbrace{\overset{2\times1}{\delta^{(2)}} \cdot \overset{2\times3}{W^{(2)}}}_{2\times3} \right] \times \left[f'(a_k^{(1)}) \, (1 - f'(a_k^{(1)})) \right] \longrightarrow \frac{\partial Error}{\partial W^{(1)}} = \vec{x}^T \delta^{(1)}$$

MLE (freq counts) won't work.

Making predictions when data is missing

EM Algorithm (In general; any model)

Given: data **d**, w/ features X where Z is hidden.

Goal: Find h (hypothesis / model) that best explains data

Guess h_{MLE}: $\boxed{h_0}$ (ie: prior)

Recall: MLE def^n

$$\boxed{h_{MLE} = \underset{h}{\arg\max}\ \underbrace{P(d|h)}_{\text{Likelihood}}}$$

here it's w/ expectation of Z

Repeat

Expectation of missing values

Compute: $P(Z|h_i)$

X_1	$\cdots$	X_n	Z	$P(Z\|h_i,X)$
$\vdots$		$\vdots$	$\vdots$	$\vdots$

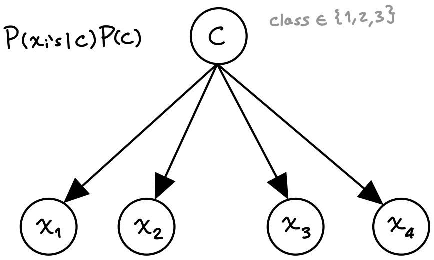
New h_{MLE}

Maximize over h

$$h_{i+1} \leftarrow \underset{h}{\arg\max} \sum_{z} P(z, d | h)\ P(z|h_i, d)$$

EM – Naive Bayes

$$P(c|x_i's) \overset{\text{bayes rule}}{\propto} P(x_i's|c)P(c)$$



class $\in \{1,2,3\}$

Prior (init. guess)

C	P(C)
1	0.4
2	0.1
3	0.5

Data labels 'C' are missing

X_1	X_2	X_3	X_4	C
$\vdots$	$\vdots$	$\vdots$	$\vdots$	?
t	f	t	t	?
$\vdots$	$\vdots$	$\vdots$	$\vdots$	?

E-Step:

For each data point

A

weight

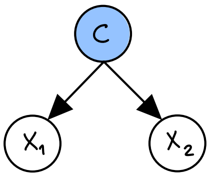
X_1	X_2	X_3	X_4	C	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	?	$\vdots$
t	f	t	t	1	0.4
t	f	t	t	2	0.1
t	f	t	t	3	0.5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	?	$\vdots$

M-Step:

$$P(x_i|C) = \frac{M_i(X_i,C)}{M(C)} \longrightarrow \sum_{X-X_i} A[X_1,\ldots,X_n,C]$$

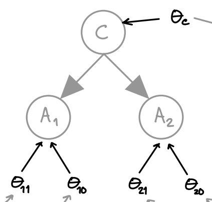
$$P(C) = \frac{M(C)}{n} \longrightarrow \sum_{X_i} M_i(X_i,C)$$

Example



Data: M=10

A_1	A_2	C
t	t	?
t	f	?
t	f	?
f	f	?
t	t	?
f	t	?
f	f	?
t	t	?
t	t	?
t	t	?



c	P(c)
t	0.6
f	0.4

A_1	c	$P(A_1\mid c)$
t	t	0.9
t	f	0.3

A_1	c	$P(A_1\mid c)$
t	t	0.2
t	f	0.6

Init. Guesses (Priors)

Repeat

E-Step:

Parameters

Calculate $P(C\mid A_1,A_2)$

$$= \frac{P(C,A_1,A_2)}{\sum_c P(C=c,A_1,A_2)}$$

$$\propto P(A_1\mid C)P(A_2\mid C)P(C)$$

A_1	A_2	C	$A_1\mid C$	$A_2\mid C$	C	A_1,A_2,C	$C\mid A_1,A_2$
t	t	t	.9	.2	.6	.108	0.6
		f	.3	.6	.4	.072	0.4
t	f	t	.9	.8	.6	.432	0.9
		f	.3	.4	.4	.048	0.1
f	t	t	.1	.2	.6	.012	0.07
		f	.7	.8	.4	.168	0.93
f	f	t	.1	.8	.6	.048	0.3
		f	.7	.4	.4	.112	0.7

÷ .108+.072

÷ .432+.048

Normalize $\sum_c P(C,A_1,A_2)$

Augment Dataset

A_1	A_2	C	A_1,A_2,C	$C\mid A_1,A_2$
t	t	t	.108	0.6
		f	.072	0.4
t	f	t	.432	0.9
		f	.048	0.1
t	f	t	.432	0.9
		f	.048	0.1
f	f	t	.048	0.3
		f	.112	0.7
t	t	t	.108	0.6
		f	.072	0.4
f	t	t	.012	0.07
		f	.168	0.93
f	f	t	.048	0.3
		f	.112	0.7
t	t	t	.108	0.6
		f	.072	0.4
t	t	t	.108	0.6
		f	.072	0.4
t	t	t	.108	0.6
		f	.072	0.4
Σ			3.36	10

= M

Stop iterating when this converges

M-Step: re-estimate params

$$\theta_c = \frac{\Sigma \text{ weights w/ } c=t}{\Sigma \text{ weights}} = \frac{5.47}{4.53+5.47} = \boxed{0.547}$$

$$\theta_{11} = \frac{\Sigma \text{ weights w/ } c=t,\ A_1=t}{\Sigma \text{ weights w/ } c=t} = \frac{4.8}{5.7} = \boxed{0.88}$$

$$\theta_{10} = \frac{\Sigma \text{ weights w/ } c=f,\ A_1=t}{\Sigma \text{ weights w/ } c=f} = \frac{2.2}{4.53} = \boxed{0.49}$$

$$\theta_{21} = \frac{\Sigma \text{ weights w/ } c=f,\ A_2=t}{\Sigma \text{ weights w/ } c=t} = \frac{3.07}{5.7} = \boxed{0.56}$$

$$\theta_{20} = \frac{\Sigma \text{ weights w/ } c=f,\ A_2=t}{\Sigma \text{ weights w/ } c=f} = \frac{2.93}{4.53} = \boxed{0.65}$$

After Convergence:

$\theta_c = P(C=t) = \boxed{0.547}$ $\theta_{11} = P(A_1=t\mid C=t) = \boxed{0.88}$ $\theta_{10} = P(A_1=t\mid C=f) = \boxed{0.49}$

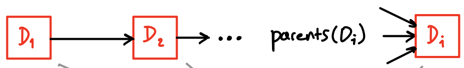
$\theta_{21} = P(A_2=t\mid C=t) = \boxed{0.56}$ $\theta_{20} = P(A_2=t\mid C=f) = \boxed{0.65}$

A_1	A_2	$P(C=t\mid A_1,A_2)$	prediction
t	t	0.493	F
t	f	0.466	F
f	t	0.08	F
f	f	0.05	F

Compute $P(C=t\mid A_1,A_2)$ for each test data, write it in the table to the left, and then threshold at 0.5 to make a prediction for the test data

Sequential Decisions $D_1, \dots D_n$

Each D_i has an information set parents(D_i) whose values will be know when its time to make D_i



Policy Sequence $\delta_1, \delta_2, \dots, \delta_n$

dom[parents(D_i)] dom$[D_i]$

$$\delta_i(o) = d_i$$

expected value of making decisions

$$V(d_1, d_2, \dots) = \sum_{x_1,\dots,x_n} U(d_1,\dots,x_1,\dots)\, P(x_1,\dots \mid d_1,\dots)$$

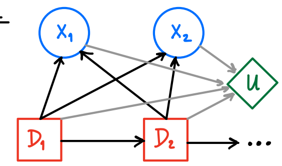
expected Utility

$$\mathcal{E}(U \mid \delta) = \sum_{\omega \models \delta} P(\omega)\, U(\omega)$$

all worlds possible with this policy

maximizes

Decision Network



Agent ⟶ action ?

depends on

ability, beliefs — Std. Bayes Net; preferences (& tradeoffs)

Decision Network — specified by: decision Variables, Utility

Algorithm: Find δ^* (optimal policy)

Init:

Factors : each CPT, Utility Function ⟶ Pool

all of them

remaining D_i's ⟵ $D_1, D_2 \dots D_n$

Repeat:

$D_x \cdots D_y \cdots D_z$ D_i

Pick D_i w/ no children (i.e. in reverse order)

Eliminate by maximizing.

ALL random var's that don't have remaining as child (skip if none)

V.E Algorithm (1 by 1 in any order)

Removes from pool (does not reset) ⟶ Pool

note: only max over factors that contain D_i

f_{k+1} — add

$$f_{k+2} = \max_{D_i} f_{k+1} \quad \mathcal{E}(U \mid D_i)$$

$$\delta_i^* = \arg\max_{D_i} f_{k+1} \quad (\text{optimal for } D_i) \longrightarrow \delta_1, \delta_2, \dots, \delta_i \text{ policy}$$

$\emptyset$

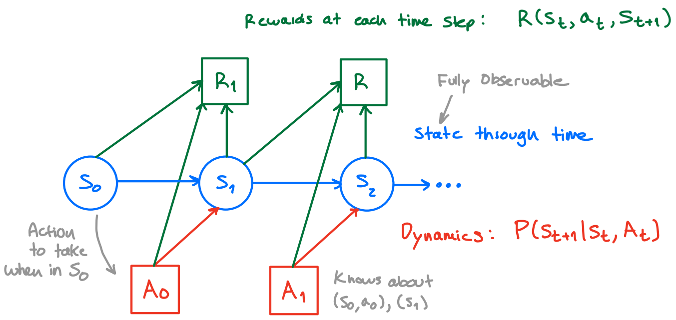
w/ expected value

End:

① Eliminate remaining variables ⟶ $\dfrac{\mathcal{E}(U \mid \delta^*)}{P(\text{Evidence})}$

② Divide by P(Evidence) (if any)

MDP Decision network w/ actions, rewards over time



2 Assumptions:

Markov Assumption | Stationary Dynamics

$$P(S_{t+1} | \underbrace{S_0, A_0, \ldots,}_{\text{past}} \underbrace{S_t, A_t}_{\text{curr State}}) = P(S_{t+1} | S_t, A_t) = P(S_{k+1} | S_k, A_k) \quad \forall k$$

(State, Action, time)

Discounted Rewards:

Utility of decisions — rewards worth less in future

$$V = \sum_{t=1}^{n} \gamma^{t-1} r_t$$

Policy: $\pi : S \longrightarrow A$

$$\pi^*(s) = \arg\max_a Q^*(s,a) \longrightarrow$$

expected value of following π

$\mathbb{E}$[doing a in s, then following π]

For all States — Prob. of ending up in that State × Value of being in that State (Immediate, Discounted)

$$Q^{\pi}(s,a) = \sum_{s'} [P(s'|a,s) \cdot (r(s,a,s') + \gamma V^{\pi}(s'))]$$

ie: taking a from s

$\mathbb{E}$[following π from s]

$$V^{\pi}(s) = Q^{\pi}(s, \pi(s))$$

Value Iteration: Find π^*

Init

$$\boxed{V^0 = R(s) \quad t = 1}$$

Iterate until $\forall s, \| V^t(s) - V^{t-1}(s) \| \le$ threshold

$t \leftarrow t-1$

$$Q^t(s,a) = \sum_{s'} [P(s'|a,s) \cdot (r(s,a,s') + \gamma V^{t-1}(s'))]$$

$$\pi^t(s) = \arg\max_a Q^t(s,a)$$

policy w/ t stages to go — actions that maximize this

$$V^t(s) = \max_a Q^t(s,a)$$

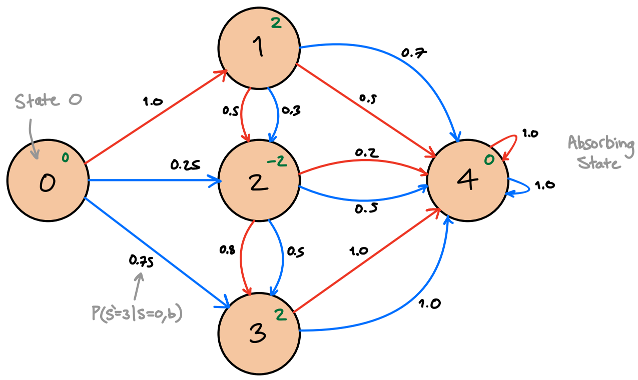
maximize rewards

Result: $\pi^*(s) : a[\cdots]^{s}$

State Space

actions: a, b

$R(S) = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$
$R(S) = [\,0 \quad 2 \quad -2 \quad 2 \quad 0\,]$



dynamics:

$P(S'|S, A=a)$

S \ S'	0	1	2	3	4
0	/	1	/	/	/
1	/	/	.5	/	/
2	/	/	/	.8	.2
3	/	/	/	/	1
4	/	/	/	/	1

$P(S'|S, A=b)$

S \ S'	0	1	2	3	4
0	/	/	.25	.75	/
1	/	/	.3	/	.7
2	/	/	/	.5	.5
3	/	/	/	/	1
4	/	/	/	/	1

Let $\gamma = 0.9$

t	$V^{t-1}(s')$	$Q^t(s,a)$: $R(s) + \gamma \sum_{s'} [P(s' \mid a,s) V^{t-1}(s')]$	$\max_a Q^t(s,a) \rightarrow V^{t+1}(s)$
1	$R(s) = [0, 2, -2, 2, 0]^T$	a: 1.8, 1.1, -0.56, 2.0, 0; b: 1.22, 1.85, -1.1, 2.0, 0	arg max$_a$: 1.8, 1.85, -0.56, 2.0, 0; $\pi^{t+1}(s)$: a, b, a, a, a
2	1.8, 1.85, -0.56, 2.0, 0	a: 1.31, 1.75, -0.56, 2.0, 0; b: 1.22, 1.85, -1.1, 2.0, 0	1.31, 1.85, -0.56, 2.0, 0; $\pi^{t+1}(s)$: a, b, a, a, a
⋮	⋮	⋮	⋮
			$V^*(s)$: 1.66, 1.85, -0.56, 2.0, 0; $\pi^*(s)$: a, b, a, a, a

State Variables : 48 states (×)

tired (T) : 3 no, a bit, yes

pass test (P) : 2 no, yes

Knows (K) : 4 nothing, a bit, a lot, all

goodtime (G) : 2 no, yes

Actions:

Study (st)

sleep (sl)

take test (tt)

party (pa)

Rewards:

+20 : P ← true

+2 : G ← true

Dynamics $P(s'|s,a) = [48 \times 48]$ ← ! only 1 action for a given state

Rewards $R(s) = [48 \times 1]$

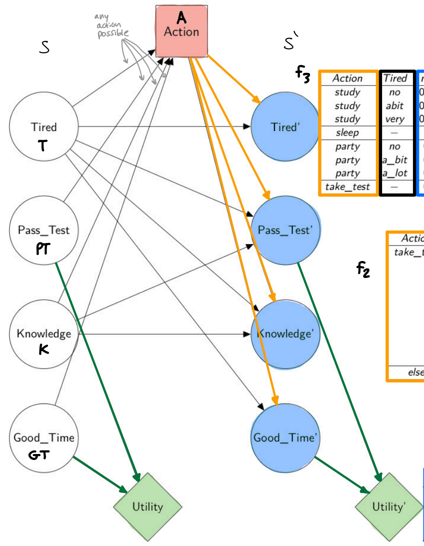
DDN:

f_1

Action	Tired	Knowledge	Knowledge′: nothing	a_bit	a_lot	everything
study	no	nothing	0.5	0.5	0	0
	no	a_bit	0	0.5	0.5	0
	no	a_lot	0.0	0.0	0.5	0.5
	no	everything	0		0	1.0
	_bit	nothing	0.5	0.5	0	0
	_bit	a_bit	0	0.5	0.5	0.0
	_bit	a_lot	0	0	0.5	0.5
	_bit	everything	0	0	0	1.0
	very	nothing	1.0	0	0	0
	very	a_bit	0	1.0	0.0	0.0
	very	a_lot	0	0	1.0	0.0
	very	everything	0	0	0	1.0
party	–	nothing	1.0	0.0	0	0
	no	a_bit	0.5	0.5	0.0	0.0
	no	a_lot	0.0	0.5	0.5	0.0
	no	everything	0	0	0.5	0.5
sleep	–	nothing	1.0	0.0	0	0
	–	a_bit	0.0	1.0	0.0	0.0
	–	a_lot	0.0	0.0	1.0	0.0
	–	everything	0	0.0	0.0	1.0
take_test	–	–	1.0	0.0	0	0

f_0

Action	Tired	Good_Time′: no	yes
party	no	0.0	1.0
party	a_bit/very	1.0	0.0
other	–	1.0	0.0



f_3

Action	Tired	Tired′: no	a_bit	very
study	no	0.8	0.2	0.0
study	abit	0.0	0.8	0.2
study	very	0.0	0.0	1.0
sleep	–	1	0	0
party	no	0	1	0
party	a_bit	0	0	1
party	a_lot	0	0	1
take_test	–	0	0	1

f_2

Action	Tired	Knowledge	PassTest′: no	yes
take_test	no	nothing	1	0
	no	a_bit	0.9	0.1
	no	a_lot	0.7	0.3
	no	everything	0.1	0.9
	a_bit	nothing	1	0
	a_bit	a_bit	0.9	0.1
	a_bit	a_lot	0.7	0.3
	a_bit	everything	0.1	0.9
	very	–	1	0
else	–	–	1	0

Good_Time	Pass_Test	U(Good_Time, Pass_Test)
yes	yes	22
yes	no	2
no	yes	20
no	no	0

Finding π^*, Variable Elimination:

Factors:

$f_0(GT', A, T) = P(GT'|A,T)$

$f_1(K', A, K, T) = P(K'|A,K,T)$

$f_2(PT', A, K, T) = P(PT'|A,K,T)$

$f_3(T', A, T) = P(T'|A,T)$

$f_4(PT', GT') = Util(PT', GT')$

Order GT', K', PT', T' :

$f_5(A,T,PT') = \sum_{GT'} f_0(GT',A,T)\, f_4(PT',GT')$

$f_6(A,K,T) = \sum_{K'} P(K'|A,K,T) = 1.0$ → since not ancestors of Util at last time step.

$f_7(A,K,T) = \sum_{PT'} f_2(PT',A,K,T)\, f_5(A,T,PT')$

$f_8(A,K,T) = \sum_{T'} f_3(T',A,T) = 1.0$

Value Elimination: ...

reinforcement learning (learning while acting)

TD Formula

Sequence of values: $\underbrace{v_1, v_2, \ldots, v_{k-1}, v_k}$

Avg of first k (running estimate)

$$Avg_k = \frac{v_1 + v_2 + \cdots + v_{k-1} + v_k}{k}$$

$$= \underbrace{\frac{k-1}{k} Avg_{k-1}}_{\text{mostly the previous value}} + \underbrace{\frac{1}{k} v_k}_{\text{some of the new estimate}}$$

$$= (1-\alpha) Avg_{k-1} + \alpha v_k \quad \text{or} \quad \boxed{Avg_{k-1} + \alpha (v_k - Avg_{k-1})}$$

In practice we will consider α as fixed constant (fine for large k)

Asynchronous Value Iteration

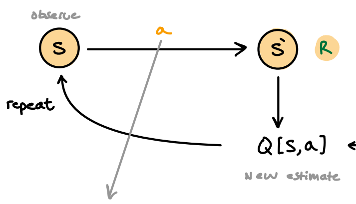
Q-Learning

① set $Q[s,a]$ arbitrarily

② Repeat Sequence of experiences $\underbrace{s_0, a_0, r_0, s_1}_{\text{curr estimate}} \cdots \underline{s, a, r, s'}$

new info

provides 1 new peice of evidence for what $P(s'|s,a)$ is...



$$\boxed{\alpha \left[Q[s,a] + r + \gamma \max_{a'} Q[s', a'] \right]}$$

Learn Rate

how to select?

Exploration Policy

How to manage choosing between

① Exploit: choose based on $Q[s,a]$

② Explore: some other action

Ex: ε-greedy

with Probabity ε: random action

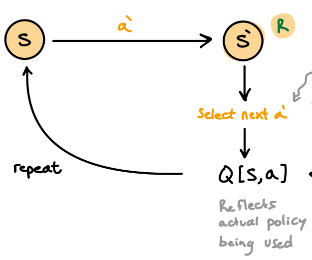
Off Policy

updates Q based on following π^*.

ie. Q is an estimate of Q^*

Other option is...

On Policy (SARSA)



e.g. w/ ε-greedy Policy

i.e. next action based on current Q

$$\boxed{\alpha \left[Q[s,a] + r + \gamma Q[s', a'] \right]}$$