chain rule

$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$

$P(A_1, A_2 \cdots A_k) = P(A_1)P(A_2|A_1)\cdots P(A_k|A_1 \cdots A_{k-1})$

if A and B are mutually independent

$P(A|B) = P(A) \quad P(B|A) = P(B)$

if A and B are conditionally independent given C

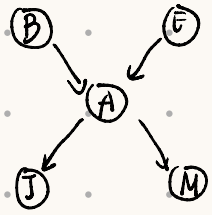
$P(A,B|C) = P(A|C)P(B|C)$

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

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

Inference By Enumeration

1. Collect all the rows consistent with the observed evidence variables.
2. Sum out (marginalize) all the hidden variable.
3. Normalize the table so that it is a probability distribution.



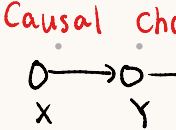
P(B) P(E)
P(A|B,E)
P(J|A)
P(M|A)

$P(X_1, X_2 \cdots X_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$

$P(B, E, A, J, M) = P(B)P(E)P(A|B,E)P(J|A)P(M|A)$

$P(var|Parent(var), Ancestor(var)) = P(var|Parents(var))$

Causal Chains

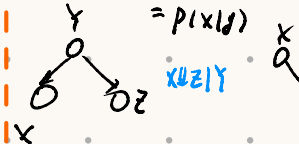


$P(x,y,z) = P(z|y)P(y|x)P(x)$

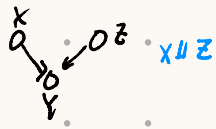
(no observation)

$P(X|Z,y) = \frac{P(X,Z,y)}{P(Z,y)} = \frac{P(Z|y)P(y|X)P(X)}{\sum_x P(X,y,Z)}$

$X \perp Z|Y \quad = \frac{P(Z|y)P(y|X)P(X)}{P(Z|y)\sum_x P(y|x)P(x)} = \frac{P(y|X)P(X)}{\sum_x P(y|x)P(x)} = P(X|y)$



$X \perp Z|Y$



$X \perp Z$

D-seperation Algorithm

1. Shade all observed values $\{Z_1 \cdots Z_k\}$ in the graph
2. Enumerate all undirected paths from X to Y.
3. For each path:
   a. Decompose the path into triples
   b. if all triples are active, this path is active and d-connects X to E.
4. If no path d-seperates X and Y, then $X \perp Y | Z_1 \cdots Z_k$.

Variable Elimination

function ELIMINATION-ASK(X, **e**, bn) **returns** a distribution over X
inputs: X, the query variable
e, observed values for variables **E**
bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

factors ← []
for each *var* **in** ORDER(bn.VARS) **do**
factors ← [MAKE-FACTOR(*var*, **e**)|*factors*]
if *var* is a hidden variable **then** *factors* ← SUM-OUT(*var*, *factors*)
return NORMALIZE(POINTWISE-PRODUCT(*factors*))

by enumeration: $\alpha \sum_s \sum_c P(T)P(s|T)P(c|T)P(t|s,c)$

by variable elimination: $\alpha P(T)\sum_s P(s|T)\sum_c P(c|T)P(t|s,c)$

given P(T) P(C|T)

prior sampling

simply sampling from given cpt.

disadvantage: require large data samples in order to perform unlikely scenerios. (like P=0.01%).

Rejection Sampling

early reject any sample inconsistent with our evidence.

disadvantage: throw away most of our sample

Likelihood Weighting

manually set all variables equal to the evidence in our query. ensures we never generate a bad sample.

iterate through each variable in the Bayes net as we do for normal sampling. sampling a value if the variable is not an evidence variable. or changing the weight for sample if it is an evidence.

function LIKELIHOOD-WEIGHTING(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
inputs: X, the query variable
e, observed values for variables **E**
bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
N, the total number of samples to be generated
local variables: **W**, a vector of weighted counts for each value of X, initially zero

for $j = 1$ to N **do**
x, w ← WEIGHTED-SAMPLE(bn, **e**)
W[x] ← **W**[x] + w where x is the value of X in **x**
return NORMALIZE(**W**)

function WEIGHTED-SAMPLE(bn, **e**) **returns** an event and a weight
w ← 1; **x** ← an event with n elements initialized from **e**
foreach variable X_i **in** $X_1, \ldots, X_n$ **do**
if X_i is an evidence variable with value x_i in **e**
then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
else **x**[i] ← a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
return **x**, w

eg. calc P(T|+c,+e):

① $w_j = 1$. c=true. e=true
② sampling t_j from P(T)
③ for C. $w_j = w_j \cdot P(+c|t_j)$
④ for S. sampling s_j from $P(S|t_j)$
⑤ for e. $w_j = w_j \cdot P(+e|t_j, c_j, s_j)$

Gibbs Sampling

first sampling all variables to some random value. then repeatedly pick one variable at one time, clear its value. and resample it.

function GIBBS-ASK(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
local variables: **N**, a vector of counts for each value of X, initially zero
Z, the nonevidence variables in bn
x, the current state of the network, initially copied from **e**

initialize **x** with random values for the variables in **Z**
for $j = 1$ to N **do**
for each Z_i in **Z** **do**
set the value of Z_i in **x** by sampling from $\mathbf{P}(Z_i|mb(Z_i))$
N[x] ← **N**[x] + 1 where x is the value of X in **x**
return NORMALIZE(**N**)

Axioms of Utility

Orderability: $(A \succ B) \vee (B \succ A) \vee (A \sim B)$

Transitivity: $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$

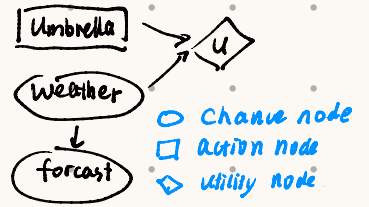
Continuaty: $A \succ B \succ C \Rightarrow \exists p \; [p, A; 1-p, C] \sim B$

Substitutability: $A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-p, C]$.

Monotonicity $A \succ B \Rightarrow (p \geq q) \Leftrightarrow [p, A; 1-p, B] \geq [q, A; 1-q, B]$.

$U([p_1, S_1; \ldots; p_n, S_n]) = \sum_i p_i U(S_i)$

Decision Networks



Maximum Expected Utility for possible action a and evidence e.

$EU(a|e) = \sum_{x_1 \cdots x_n} P(x_1 \cdots x_n | e)\, U(a, x_1 \cdots x_n)$

General Formula

maximum expected utility

$MEU(a|e) = \max_a \sum_s P(s|a)U(s,a)$

if observed new evidence e' before acting

$MEU(e,e') = \max_a \sum_s P(s|e,e')U(s,a)$

if we don't know what the new evidence will be like, regard e' as random variable E'

$MEU(e,E') = \sum_{e'} P(e'|e)MEU(e,e')$

VPI is the increasement of MEU after a new evidence observed

$VPI(E'|e) = MEU(e,E') - MEU(e)$

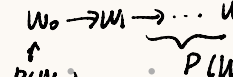
Property of VPI

Nonnegativity: $\forall E', e, \quad VPI(E'|e) \geq 0$

Non additivity: $VPI(E_j, E_k|e) \neq VPI(E_j|e) + VPI(E_k|e)$ in general

Order-Independence: $VPI(E_j, E_k|e)$
$= VPI(E_j|e) + VPI(E_k|e, E_j)$
$= VPI(E_k|e) + VPI(E_j|e, E_k)$

Markov Model



Initial Distribution

Transition Model

Markov Property: Wi only depends on Wi-1, independent to others

$P(W_0, W_1, W_2) = P(W_0)P(W_1|W_0)P(W_2|W_1, W_0)$
$= P(W_0)P(W_1|W_0)P(W_2|W_1)$

$\therefore P(W_0, W_1 \cdots W_n) = P(W_0)\prod_{i=0}^{n-1} P(W_{i+1}|W_i)$

if the transition model is stationary. ($P(W_{i+1}|W_i)$ are identical). Then the markov model can be represented by 2 tables: $P(W_0)$, $P(W_{i+1}|W_i)$

The Mini-Forward Algorithm

$P(W_{i+1}) = \sum_{W_i} P(W_i, W_{i+1})$
$= \sum_{W_i} P(W_{i+1}|W_i)P(W_i)$

stationary Distribution

$$\begin{cases} x+y=1 \\ 0.6x+0.1y=x \\ 0.4x+0.9y=y \end{cases} \Rightarrow \begin{cases} x=0.2 \\ y=0.8 \end{cases}$$

Hidden Markov Models

$W_0 \to W_1 \to W_2 \to W_3 \cdots$
$\downarrow F_1 \quad \downarrow F_2 \quad \downarrow F_3$

$F_1 \perp W_0 \mid W_1$

$W_i \perp \{W_0 \cdots W_{i-2}, F_1 \cdots F_{i-1}\} \mid W_{i-1}$

$F_i \perp \{W_0 \cdots W_{i-1}, F_1 \cdots F_{i-1}\} \mid W_i$

assumption: $P(W_{i+1}|W_i)$, $P(F_i|W_i)$ are stationary.

Belief Distribution

$B(W_i) = P(W_i | f_1 \cdots f_i)$

$B'(W_i) = P(W_i | f_1 \cdots f_{i-1})$

Forward Algorithm

$B'(W_{i+1}) = P(W_{i+1}|f_{1:i}) = \sum_{W_i} P(W_{i+1}, W_i | f_{1:i})$
$= \sum_{W_i} P(W_{i+1}|W_i, f_{1:i})P(W_i|f_{1:i})$
$= \sum_{W_i} P(W_{i+1}|W_i)P(W_i|f_{1:i})$
$= \sum_{W_i} P(W_{i+1}|W_i)B(W_i)$

$B(W_{i+1}) = P(W_{i+1}|f_{1:i+1}) = \frac{P(W_{i+1}, f_{i+1}|f_{1:i})}{P(f_{i+1}|f_{1:i})}$
$\propto P(W_{i+1}, f_{i+1}|f_{1:i})$
$\propto P(f_{i+1}|W_{i+1}, f_{1:i})P(W_{i+1}|f_{1:i})$
$\propto P(f_{i+1}|W_{i+1})B'(W_{i+1})$
$\propto P(f_{i+1}|W_{i+1})\sum_{W_i} P(W_{i+1}|W_i)B(W_i)$

Time Elapse Update: $B(W_i) \to B'(W_{i+1})$

Observation Update: f_{i+1}, $B'(W_{i+1}) \to B(W_{i+1})$

Particle Filter

gives an approximation of $P(X_N | e_{1:N})$

particle number $n \ll d$ (possible states)

but still enough to generate valid approximation

① particles ② passage of time

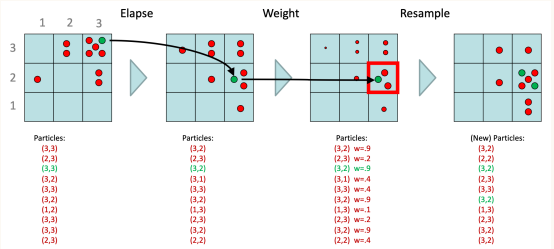
$x' = \text{sample}(P(X'|x))$

X'	$P(X'\|X=(3,3))$
$(4,2)$	0.8
$(3,3)$	0.1
$(2,3)$	0.1

③ observe

$w(x) = P(e|x)$

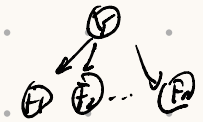
$B(X) \propto P(e|X)B'(X)$



1. 初始粒子
粒子数 $n = 10$，初始状态分布：
$[15, 12, 12, 10, 18, 14, 12, 11, 11, 10]$
2. 时间推移更新
转移模型：
$P(14 \mid 15) = 0.1, P(15 \mid 15) = 0.8, P(16 \mid 15) = 0.1$
随机数采样后更新粒子：
$[15, 13, 13, 11, 17, 15, 13, 12, 12, 10]$
3. 观测更新
假设观测 $F = 13$，传感器模型：
$P(F \mid 13) = 0.8$，其他状态 $P(F \mid T) = 0.02$
计算权重，归一化后重新采样：
粒子更新为 $[13, 13, 13, 13, 13, 13, 13, 13, 15, 13]$
4. 输出信念分布

T	10	11	12	13	14
$B(T)$	0.0	0.0	0.0	0.9	0.1

Naive Bayes



$P(Y=\text{spam} | F_1=f_1 \cdots F_n=f_n)$
$P(Y=\text{ham} | F_1=f_1 \cdots F_n=f_n)$

$P(Y=\text{spam}|F) \propto P(Y=\text{spam})\prod_{i=1}^{n} P(F_i|Y=\text{spam})$

$P(F_1, F_2 \cdots F_n|Y) = \prod_{i=1}^{n} P(F_i|Y)$

$\text{prediction}(f_1 \cdots f_n) = \underset{y}{\text{argmax}}\, P(Y=y|F)$
$= \underset{y}{\text{argmax}}\, P(Y=y, F_1=f_1 \cdots F_n=f_n)$
$= \underset{y}{\text{argmax}}\, P(Y=y)\prod_{i=1}^{n} P(F_i=f_i|Y=y)$

$P(Y, F_1=f_1 \cdots F_n=f_n) = \begin{bmatrix} P(Y=y_1, F_1=f_1 \cdots F_n=f_n) \\ \vdots \\ P(Y=y_k, F_1=f_1 \cdots F_n=f_n) \end{bmatrix}$

$= \begin{bmatrix} P(Y=y_1)\prod_i P(F_i=f_i|Y=y_1) \\ \vdots \\ P(Y=y_k)\prod_i P(F_i=f_i|Y=y_k) \end{bmatrix}$

$\text{prediction}(F) = \underset{y_i}{\text{argmax}}\, P(Y=y_i)\prod_j P(F_j=f_j|Y=y_i)$

Parameter Estimation

$L(\theta) = P_\theta(x_1 \cdots x_N) = \prod_{i=1}^{N} P_\theta(x_i)$ let $\frac{\partial}{\partial\theta}L(\theta)=0$

Maximum Likelihood for Naive Bayes

$L(\theta) = \prod_{j=1}^{N_h} P(F_i = f_i^{(j)} | Y=\text{ham}) = \prod_{j=1}^{N_h} \theta^{f_i^{(j)}}(1-\theta)^{1-f_i^{(j)}}$

$\log L(\theta) = \log\left(\prod_{j=1}^{N_h} \theta^{f_i^{(j)}}(1-\theta)^{1-f_i^{(j)}}\right)$
$= \sum_{j=1}^{N_h} \log\left(\theta^{f_i^{(j)}} + (1-\theta)^{1-f_i^{(j)}}\right)$
$= \log\theta \sum_{j=1}^{N_h} f_i^{(j)} + \log(1-\theta)\sum_{j=1}^{N_h}(1-f_i^{(j)})$

$\frac{\partial}{\partial\theta}\left(\log\theta \sum_{j=1}^{N_h} f_i^{(j)} + \log(1-\theta)\sum_{j=1}^{N_h}(1-f_i^{(j)})\right) = 0$

$\Rightarrow \frac{1}{\theta}\sum_{j=1}^{N_h} f_i^{(j)} - \frac{1}{1-\theta}\sum_{j=1}^{N_h}(1-f_i^{(j)}) = 0$

$\theta = \frac{1}{N_h}\sum_{j=1}^{N_h} f_i^{(j)}$

Laplace Smoothing

$P_{MLE}(x) = \frac{\text{count}(x)}{N}$

$P_{LAP,k}(x) = \frac{\text{count}(x)+k}{N+k|X|}$

$P_{LAP,k}(x|y) = \frac{\text{count}(x,y)+k}{\text{count}(y)+k|X|}$

$P_{LAP,0}(x) = P_{MLE}(x)$ $P_{LAP,\infty}(x) = \frac{1}{|X|}$

Perceptron

$\text{activation}_w(x) = h_w(x) = \sum_i w_i f(x)_i = w^T f(x) = w \cdot f(x)$

$\text{classify}(x) = \begin{cases} + & \text{if } h_w(x) > 0 \quad \cos\theta > 0 \quad \theta < \frac{\pi}{2} \\ - & \text{if } h_w(x) < 0 \quad \cos\theta < 0 \quad \theta > \frac{\pi}{2} \end{cases}$

$h_w(x) = w \cdot f(x) = \|w\| \|f\| \cdot \cos\theta$

Binary Perceptron

① Initialize $w=0$

② $y = \text{classify}(x) = \begin{cases} +1 & \text{if } h_w(x) = w^T f(x) > 0 \\ -1 & \text{if } h_w(x) = w^T f(x) < 0 \end{cases}$

③ if $y = y^*$ do nothing
else. $w \leftarrow w + y^* f(x)$

Linear Regression

$h_w(x) = w_0 + w_1 x^1 + \cdots + w_n x^n$

$\text{Loss}(h_w) = \frac{1}{2}\sum_{j=1}^{N} L_2(y^j, h_w(x^j))$
$= \frac{1}{2}\|y - Xw\|_2^2$

$y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix}$ $X = \begin{bmatrix} 1 & x_1^1 \cdots & x_1^n \\ 1 & x_2^1 \cdots & x_2^n \\ \vdots & & \vdots \\ 1 & x_N^1 \cdots & x_N^n \end{bmatrix}$ $w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}$

$\nabla_w \frac{1}{2}\|y - Xw\|_2^2 = \nabla_w \frac{1}{2}(y-Xw)^T(y-Xw)$
$= -X^T y + X^T X w = 0$

$\therefore \hat{w} = (X^T X)^{-1} X^T y$

$h_{\hat{w}}(x) = \hat{w}^T x$

Gradient Ascent

① Randomly initialize w

② while w is not converged

$w \leftarrow w + \alpha \nabla_w f(w)$

Gradient Descent

① Randomly initialize w

② while w is not converged:

$w \leftarrow w - \alpha \nabla_w f(w)$

Least Squares Gradient Descent

① Randomly initialize w

② While w is not converged:

$w \leftarrow w - \alpha(-X^T y + X^T X w)$

Logistic Regression

$h_w(x) = \frac{1}{1+e^{-w^T x}}$

$P(y=+1 | f(x); w) = \frac{1}{1+e^{-w^T f(x)}}$

$P(y=-1 | f(x); w) = 1 - \frac{1}{1+e^{-w^T f(x)}}$

$\text{Loss}(w) = \frac{1}{2}(y - h_w(x))^2$

$\frac{\partial}{\partial w_i}\frac{1}{2}(y-h_w(x))^2 = -(y-h_w(x))h_w(x)(1-h_w(x))x_i$

$g(z) = \frac{1}{1+e^{-z}}$ $g'(z) = g(z)(1-g(z))$