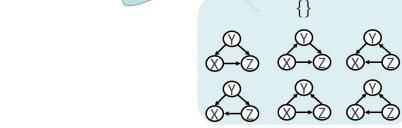
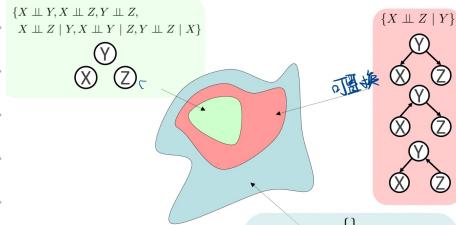


Possibilities

$$\text{Bayes Rule: } P(x|y) = \frac{P(y|x)}{P(y)} P(x)$$



Active Triples



Inactive Triples



common effect extension

IN: evidence instantiation

$$w = 1.0$$

for $i=1, 2, \dots, n$

if X_i is an evidence variable

x_i = observation for X_i

$$\text{Set } w = w * P(x_i | \text{Parents}(X_i))$$

else

$$\text{Sample } x_i \text{ from } P(X_i | \text{Parents}(X_i))$$

return $(x_1, x_2, \dots, x_n), w$

Markov Chain

$$P(X_t) = \sum_{X_{t-1}} P(X_{t-1}=x_{t-1}) P(X_t | X_{t-1}=x_{t-1})$$

$$P(X_{t+1}) = T^T P(X_t) \quad \downarrow \text{filtering}$$

$$P(X_{t+1} | e_{1:t+1}) = \alpha P(e_{t+1} | X_{t+1}) \sum_{x_t} P(x_t | e_{1:t}) P(X_{t+1} | x_t)$$

$$f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t}$$

$$\text{Prediction: } P(x_{t+1} | e_{1:t}) = \sum_x P(x_{t+1} | x_t) P(x_t | e_{1:t})$$

Filtering: $P(X_t | e_{1:t})$

belief state — input to the decision process of a rational agent

Prediction: $P(X_{t+k} | e_{1:t})$ for $k > 0$

evaluation of possible action sequences; like filtering without the evidence

Smoothing: $P(x_k | e_{1:t})$ for $0 \leq k < t$

better estimate of past states, essential for learning

Most likely explanation: $\arg \max_{x_{1:t}} P(x_{1:t} | e_{1:t})$

speech recognition, decoding with a noisy channel

$$A \Rightarrow B \equiv \neg A \vee B$$

State Trellis

$$\text{value of arc: } P(x_t | x_{t-1}) P(e_t | x_t)$$

$$m_{1:t+1} = \text{VITERBI}(m_{1:t}, e_{t+1}) \\ = P(e_{t+1} | X_{t+1}) \max_{x_t} P(X_{t+1} | x_t) m_{1:t}$$

$$A: a_{ji} = P(X_{t+1} = j | X_t = i)$$

$X_t X_{t+1}$	A	B	H	S
A	0.6	0.1	0.1	0.2
B	0.0	0.3	0.2	0.5
H	0.8	0.1	0.1	0.1
S	0.2	0.0	0.1	0.7

$$B: b_{ik} = P(O_t = k | X_t = i)$$

$X_t O_t$	p	e	b	i
A	0.6	0.2	0.1	0.1
B	0.1	0.4	0.1	0.4
H	0.0	0.0	0.7	0.3
S	0.0	0.0	0.1	0.9

$$\pi = P(X_t = i):$$

A	B	H	S
0.5	0.0	0.0	0.5

Step 1: Initialize $\delta_1(i)$

$$O_1 = \underline{\underline{b}}$$

$$\delta_1(i) = \begin{bmatrix} 0.5 \\ 0.0 \\ 0.0 \\ 0.5 \end{bmatrix} \odot \begin{bmatrix} 0.1 \\ 0.7 \\ 0.1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.05 \\ 0.0 \\ 0.0 \\ 0.05 \end{bmatrix}$$

$$\tau_1 = \underline{\underline{0(e)}}$$

Observations:

$$o_{1:t} = \{b, p, l, e\}$$

Find:

Most likely hidden state sequence: $X_{1:t+1}^*$

$$O_2 = \underline{\underline{p}}$$

$$\delta_2(i) = \underline{\underline{ }}$$

$$\begin{aligned} A \rightarrow A & (p) & B \rightarrow A & H \rightarrow A & S \rightarrow A \\ \max(0.05 \times 0.6 \times 0.6, & 0.0 \times 0.2 \times 0.6, 0.8 \times 0.8 \times 0.6, 0.05 \times 0.2 \times 0.6) \\ \max(0.05 \times 0.1 \times 0.1, & 0.0 \times 0.7 \times 0.1, 0.0 \times 0.1 \times 0.1, 0.05 \times 0.0 \times 0.1) \\ \max(0.05 \times 0.1 \times 0.9, & 0.0 \times 0.7 \times 0.9, 0.0 \times 0.1 \times 0.9, 0.05 \times 0.0 \times 0.9) \\ \max(0.05 \times 0.2 \times 0.9, & 0.0 \times 0.5 \times 0.9, 0.0 \times 0.1 \times 0.9, 0.05 \times 0.7 \times 0.9) \end{aligned} =$$

$$O_3 = \underline{\underline{l}}$$

$$\delta_3(i) = \underline{\underline{ }}$$

$$\begin{aligned} \max(0.018 \times 0.6 \times 0.1, & 0.0005 \times 0.2 \times 0.1, 0.0 \times 0.8 \times 0.1, 0.0 \times 0.2 \times 0.1) \\ \max(0.018 \times 0.1 \times 0.4, & 0.0005 \times 0.7 \times 0.4, 0.0 \times 0.1 \times 0.4, 0.0 \times 0.0 \times 0.4) \\ \max(0.018 \times 0.1 \times 0.3, & 0.0005 \times 0.2 \times 0.3, 0.0 \times 0.1 \times 0.3, 0.0 \times 0.1 \times 0.3) \\ \max(0.018 \times 0.2 \times 0.9, & 0.0005 \times 0.5 \times 0.9, 0.0 \times 0.1 \times 0.9, 0.0 \times 0.7 \times 0.9) \end{aligned} =$$

States and deltas over time:

$\delta_1(i)$	state
A	/
B	/
H	/
S	/

$\delta_2(i)$	state
A	/
B	/
H	/
S	/

$\delta_3(i)$	state
A	/
B	/
H	/
S	/

$\delta_4(i)$	state
A	/
B	/
H	/
S	/

Backtracking gives two answers:

$$X_{1:t} = \{A, A, A, A\} \text{ and}$$

$$X_{1:t} = \{A, A, S, A\}$$

- Naive Bayes assumes all features are independent effects of the label.
- Naive Bayes for text: $P(Y, W_1, \dots, W_n) = P(Y) \prod_i P(W_i | Y)$, where W_i is the word at position i , $Y \in \{\text{spam, ham}\}$.

Maximum likelihood estimation

- Given the observed set D , find θ to maximize the probability of D
- $\theta = \arg \max_{\theta} P(D|\theta)$

Set the derivative of $P(D|\theta)$ with respect to θ to zero, and solve for θ .

- pretend that we have seen every outcome k extra times.

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

Linear classifiers
feature values(inputs), weights (learned), activation (sum), activation_w(x) = $\sum_i w_i \phi_i(x)$)

Binary perceptron learning process

- start with weights=0
- for each training instance (x, y^*) : classify with current weights, no change if correct else adjust the weight vector by adding or subtracting the feature vector (subtract if $y^*=-1$).

Multiclass perceptron learning process

- start with weights=0
- pick up training examples one by one
- predict with current weights $\hat{y} = \text{argmax}_y(w_y \cdot \phi(x))$, no change if correct. Otherwise, lower score of wrong answer and raise score of right answer $w_{\hat{y}} = w_{\hat{y}} - \phi(x)$, $w_{y^*} = w_{y^*} + \phi(x)$

Probabilistic Perceptron

- softmax function

initialize w (e.g., randomly)

$$\frac{d}{dw_y} \log P_w(y_i|x_i) = x_i(I(y=y_i) - P(y|x_i))$$

repeat for K iterations:

for each example (x_i, y_i) :

compute gradient $\Delta_i = -\nabla_w \log P_w(y_i|x_i)$

compute gradient $\nabla_w \mathcal{L} = \sum_i \Delta_i$

$w \leftarrow w - \alpha \nabla_w \mathcal{L}$

- ❖ α : learning rate -- hyperparameter that needs to be chosen carefully

- ❖ How? Try multiple choices

❖ Crude rule of thumb: update should change w by about 0.1-1%

❖ False \models True ✓

❖ True \models False F

❖ $(A \wedge B) \models (A \Leftrightarrow B)$ ✓

❖ $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B)$ ✓

❖ $(A \vee B) \wedge \neg(A \Rightarrow B)$ is satisfiable ✓

❖ $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ is satisfiable ✓

❖ $(A \Leftrightarrow B) \Leftrightarrow C$ has the same number of models as $(A \Leftrightarrow B)$ for any fixed set of proposition symbols that includes A, B, C ✓