Review-2

Decision Tree Equations

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i$$

S= Collection of Examples
P_i = proportion of S that belongs to class i

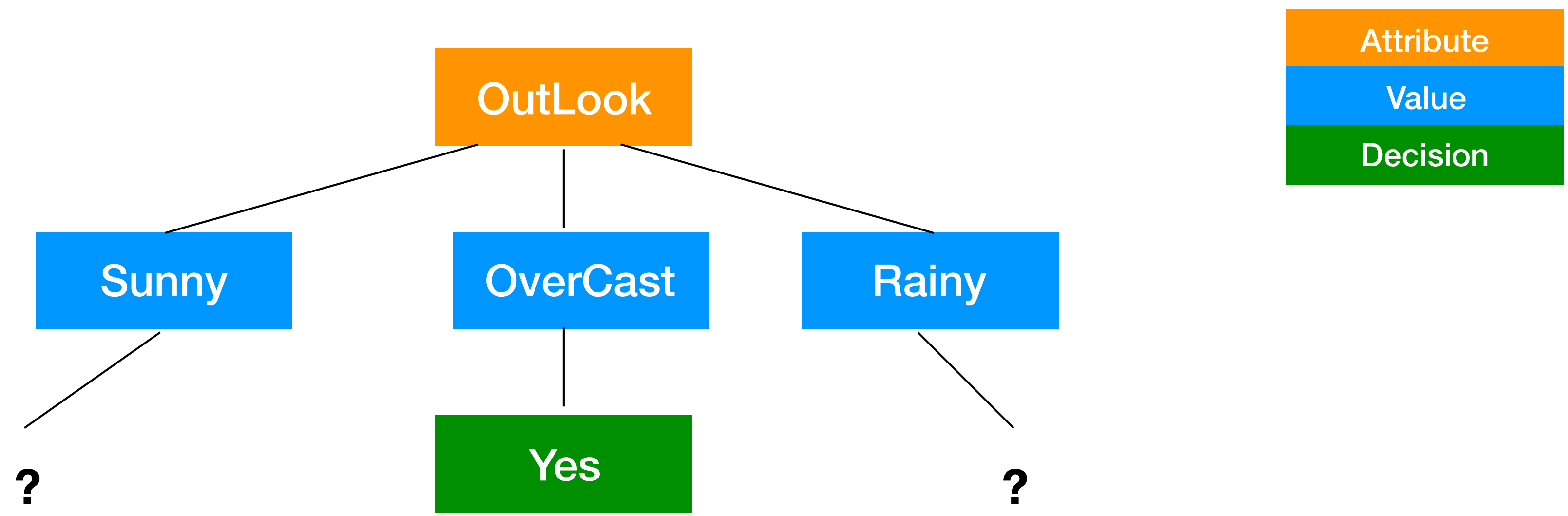
$$Gain(S, A) = Entropy(S) - \sum_{v \in values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

A= Attribute
values(A) = set of all possible values for A
|S| = number of example in S
$|S_v|$ =number of example in S where the value of A is v

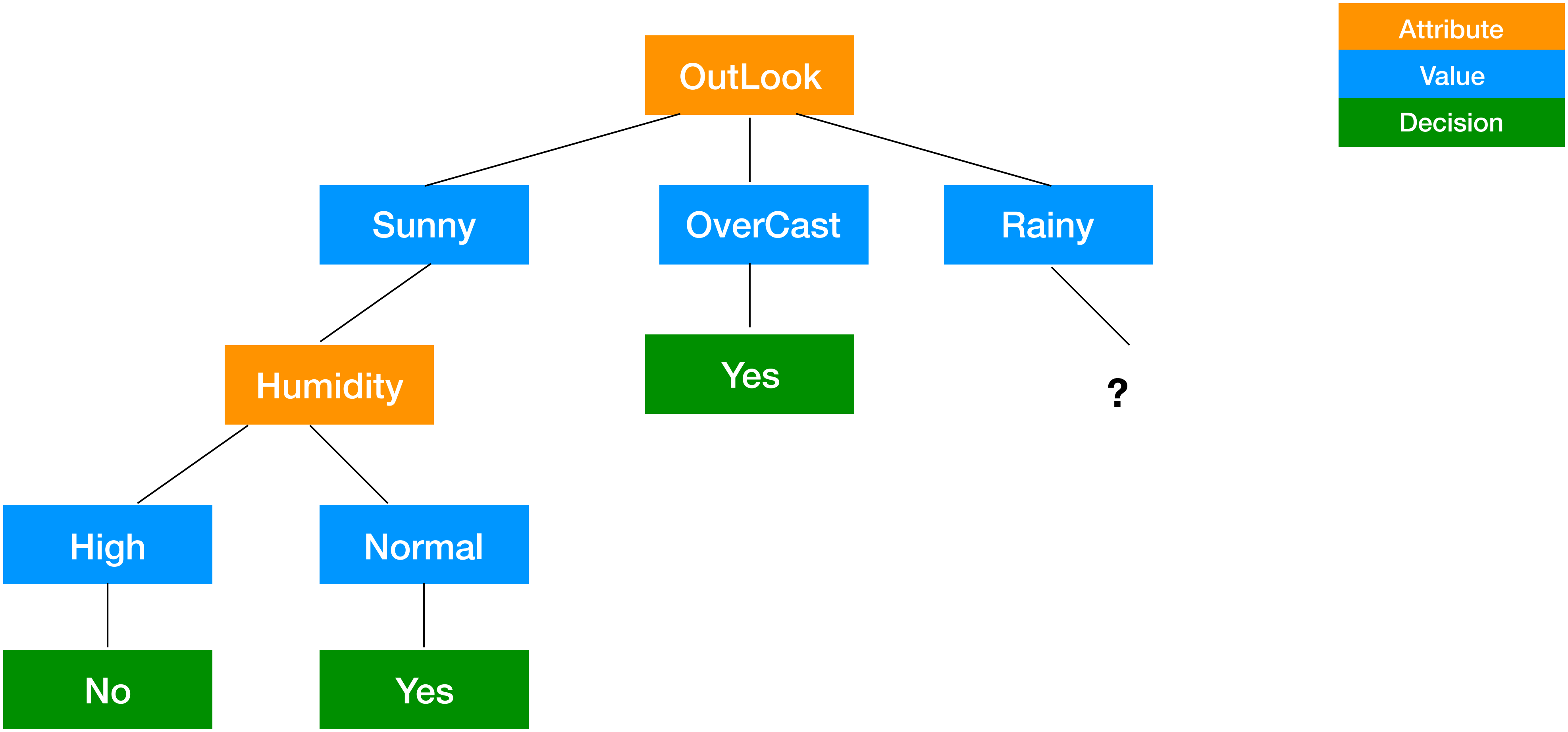
Decision Tree Solution

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree Solution



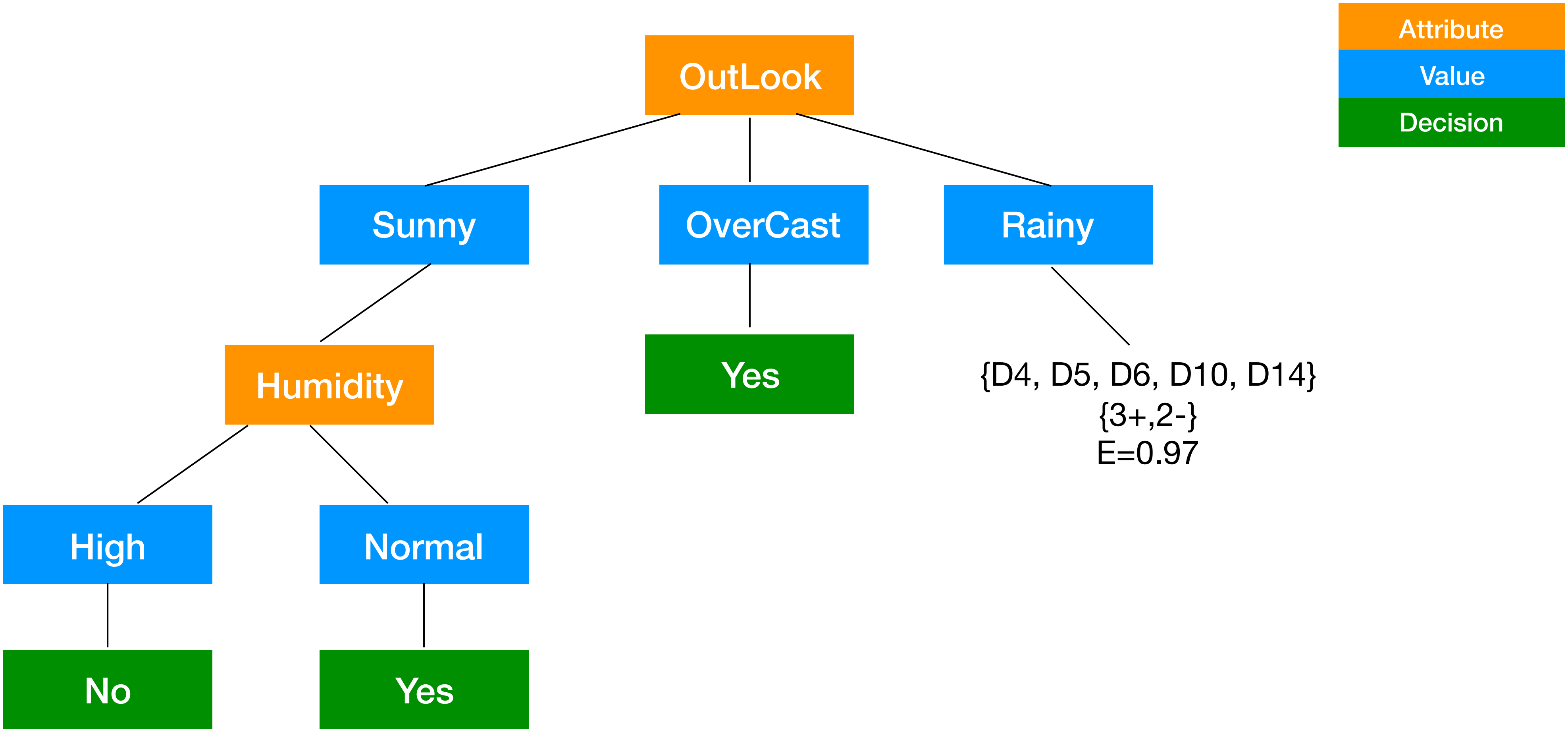
Decision Tree Solution



Decision Tree Solution

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

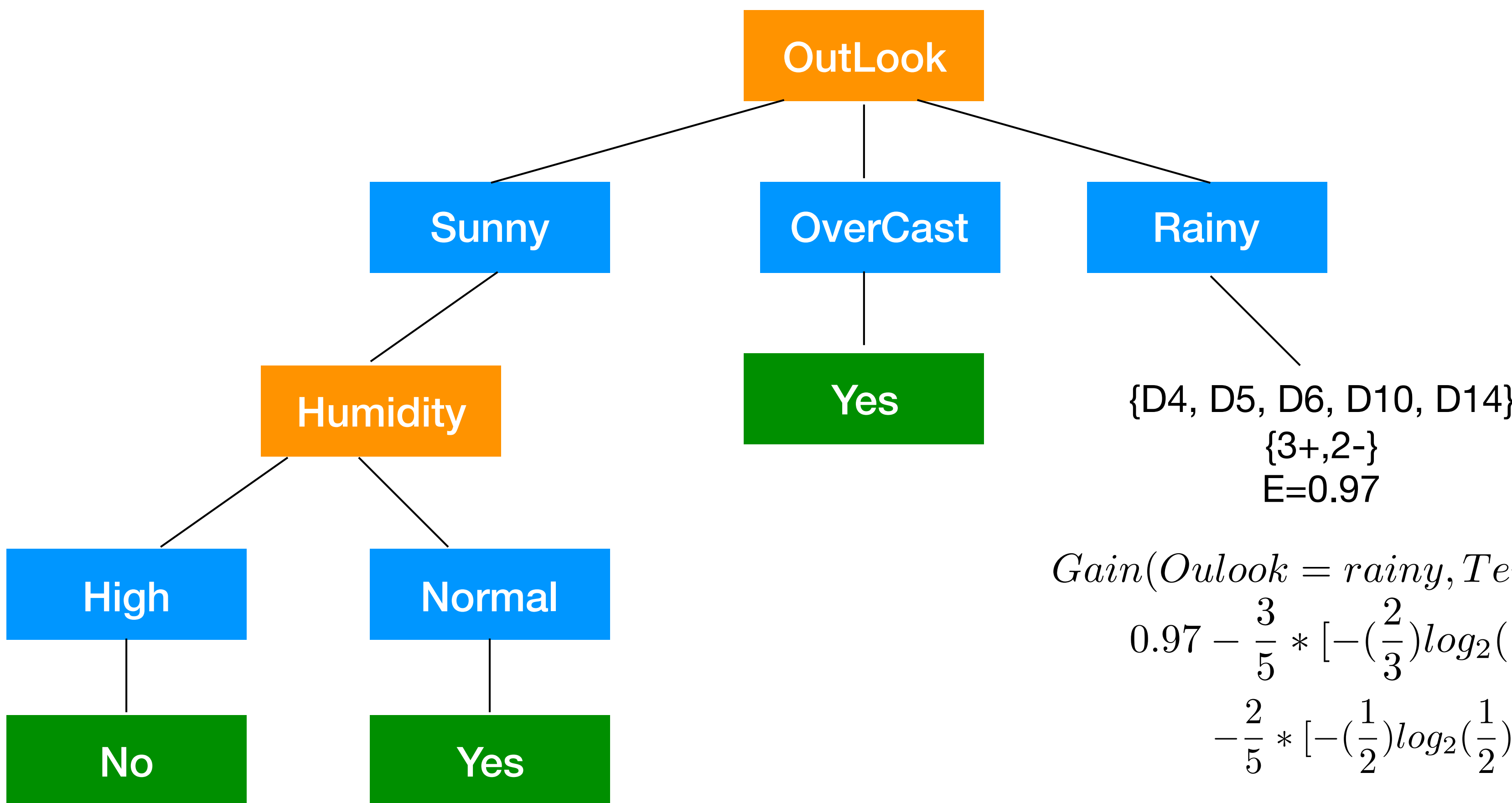
Decision Tree Solution

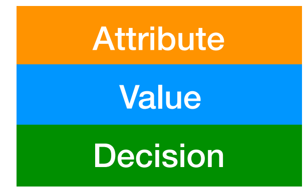


Decision Tree Solution

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree Solution





$$Gain(Oulook = rainy, Temp) =$$
$$0.97 - \frac{3}{5} * [-(\frac{2}{3})log_2(\frac{2}{3}) - (\frac{1}{3})log_2(\frac{1}{3})]$$
$$-\frac{2}{5} * [-(\frac{1}{2})log_2(\frac{1}{2}) - (\frac{1}{2})log_2(\frac{1}{2})]$$
$$= 0.024$$

Probability Review

$$\sum_x P(X = x) = 1$$

Conditional Probability

$$\frac{P(A, B)}{P(B)} = P(A|B)$$

Chain Rule

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

Probability Review

Disjunction / Union: $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

Negation: $P(\neg A) = 1 - P(A)$

$$\sum_x P(X = x, Y) = P(Y)$$

Bayes Rule

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

we call P(A) the prior

and P(A|B) the posterior

Applying Bayes Rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\sim A)P(\sim A)}$$

Find the probability of having a flu given that you have just coughed

A = you have the flu, B = you just coughed

Assume:
P(A) = 0.05
P(B|A) = 0.80
P(B| ~A) = 0.2

P(~A) = 0.95
P (A |B) = 0.17

what is P(flu | cough) = P(A|B)?

Bayes' Rule Applied to Documents and Classes

For a document d and a class c

Posterior $\longrightarrow$ $$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \; \boldsymbol{P(d \,/\, c)} \; \boldsymbol{P(C)}$$

Dropping the denominator

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} \, P(x_1, x_2, \ldots, x_n \mid c)P(c)$$

$$d = \{x_1, x_2, x_3, \ldots, x_n\}$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \, P(c_j) \prod_{x \in X} P(x \mid c)$$

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Laplace (add-1) smoothing for Naïve Bayes

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c)}{\sum_{w \in V} \left(count(w, c)\right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c)\right) + |V|}$$

Naïve Bayes as a Log-Linear Model

$$P(\text{spam}|D) \propto P(\text{spam}) \prod_{w \in D} P(w|\text{spam})$$

$$P(\text{spam}|D) \propto P(\text{spam}) \prod_{w \in \text{Vocab}} P(w|\text{spam})^{x_i}$$

$$\log P(\text{spam}|D) \propto \log P(\text{spam}) + \sum_{w \in \text{Vocab}} x_i \cdot \log P(w|\text{spam})$$

Naïve Bayes as a Log-Linear Model

$$\log P(\text{spam}|D) \propto \log P(\text{spam}) + \sum_{w \in \text{Vocab}} x_i \cdot \log P(w|\text{spam})$$

In both Naïve Bayes and Logistic Regression we Compute The Dot Product!

NB vs. LR: Parameter Learning

- Naïve Bayes:
  - Learn conditional probabilities **independently** by counting
- Logistic Regression:
  - Learn weights **jointly**

LR: Learning Weights

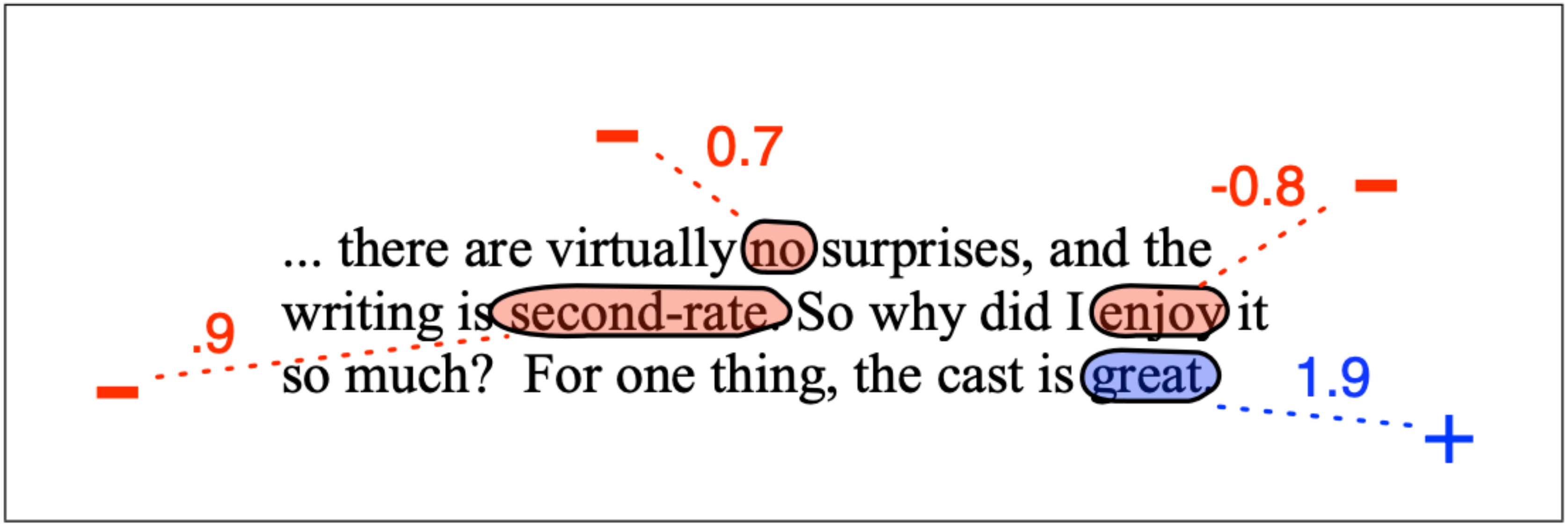
$$
\begin{array}{rcc}
 & \begin{array}{ccccc} \phantom{x_{11}} & \phantom{x_{12}} & \text{Feature} & & \phantom{x_{1n}} \end{array} & \text{Labels} \\
\text{Document} \rightarrow & \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \cdots & x_{dn} \end{bmatrix} & \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}
\end{array}
$$

$$w_{\text{MLE}} = \operatorname{argmax}_w \log P(y_1, \ldots, y_d | x_1, \ldots, x_d; w)$$

MultiClass Logistic Regression

$$\boxed{P(y|x) = \frac{e^{w \cdot f(d,y)}}{\sum_{y' \in Y} e^{w \cdot f(d,y')}}}$$

Features for movie review



$$P(+|x) = \frac{e^{1.9}}{e^{1.9}+e^{.9}e^{.7}e^{-.8}} = .82$$

$$P(-|x) = \frac{e^{.9}e^{.7}e^{.8}}{e^{1.9}+e^{.9}e^{.7}e^{-.8}} = .18$$

MultiClass Perceptron Algorithm

$Initialize\ weight\ vector\ w = 0$

$Loop\ for\ K\ iterations$

$Loop\ For\ all\ training\ examples\ x_i$

$$y_{pred} = argmax_y(w_y * x_i)$$

$$if\ y_{pred} \mathrel{!}= \ y_i$$

$$w_{y_{gold}} += x_i$$

$$w_{y_{pred}} -= x_i$$

Adversarial Search

- **Minimax search**
  - A state-space search tree
  - Players alternate
  - Choose move to position with highest minimax value = best achievable utility against best play

- **Expectimax**
  - What if we don't know what the result of an action will be? E.g.,
    - In solitaire, next card is unknown
    - In minesweeper, mine locations
    - In pacman, the ghosts act randomly
  - Can do **expectimax search**
    - Chance nodes, like min nodes, except the outcome is uncertain
    - Calculate expected utilities
    - Max nodes as in minimax search
    - Chance nodes take average (expectation) of value of children

Cheat Sheet

- 1 page letter sized
- Can be handwritten or printed
- Can write on both sides
- Can write in any font

Devices

- **No Laptop**
- **No Mobile**
- **Only Calculator is allowed**