CS57300
PURDUE UNIVERSITY

SEPTEMBER 22, 2021

DATA MINING

ANNOUNCEMENTS

- Assignment 2 is out!
  - Implement Naive Bayes Classifier from scratch
  - Due: October 5, 11:59pm
  - If you want to apply extension days, please clearly specify the number of extension days you want to apply in your submitted pdf document

NAIVE BAYES CLASSIFIER: SEARCH

MAXIMUM LIKELIHOOD ESTIMATION

- "Learn" the best parameters by finding the values of θ that maximizes likelihood:

$$\hat{\theta}_{MLE} = \arg\max_{\theta} L(\theta)$$

- Often easier to work with loglikelihood:

$$\begin{aligned} l(\theta|D) &= log\, L(\theta|D) \\ &= log \prod_{i=1}^{n} p(x(i)|\theta) \\ &= \sum_{i=1}^{n} logp(x(i)|\theta) \end{aligned}$$

MLE FOR NBC

- Likelihood: $L(\theta | D) = \prod_{i=1}^{n} \prod_{j=1}^{m} P(x_{ij} | c_i) P(c_i)$

MLE FOR NBC

- Rewrite likelihood: $L(\theta | D) = (\prod_{l=1}^{L} p_l^{N_l})(\prod_{l=1}^{L}\prod_{j=1}^{m}\prod_{k=1}^{K(j)} (q_l^{jk})^{N_l^{jk}})$
    - $N_l = \sum_{i=1}^{n} I(c_i = l)$, i.e., the number of data points in class *l*
    - $N_l^{jk} = \sum_{i=1}^{n} I(c_i = l, x_{ij} = k)$, i.e. the number of data points in class *l*, and its *j*-th attribute is *k*
- Convex maximization
    - $p_l = N_l / n$, i.e., the fraction of data in the training set where its label is *l*
    - $q_l^{jk} = N_l^{jk} / N_l$, i.e. the fraction of data whose *j*-th attribute is *k* among data whose label is *l*

LEARNING CPDS FROM EXAMPLES

		X_1		
		Low	**Med**	**High**
Y	**Yes**	10	13	17
	No	2	13	0

$$P[\,X_1 = \text{Low} \mid Y = \text{Yes}] = \frac{10}{(10+13+17)}$$

$$P[\,Y = \text{No}] = \frac{(2+13)}{(2+13+10+13+17)}$$

NBC LEARNING

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- Estimate prior P(BC) and conditional probability distributions P(A | BC), P(I | BC), P(S | BC), P(CR | BC) independently with maximum likelihood estimation

P(BC)

BC	θ
yes	$9/14$
no	$5/14$

P(A | BC)

BC	A	θ
yes	$<= 30$	$2/9$
	$31..40$	$4/9$
	> 40	$3/9$
no	$<= 30$	$3/5$
	$31..40$	$0/5$
	> 40	$2/5$

P(I | BC)

BC	I	θ
yes	high	$2/9$
	med	$4/9$
	low	$3/9$
no	high	$2/5$
	med	$2/5$
	low	$1/5$

P(S | BC)

BC	S	θ
yes	yes	$6/9$
	no	$3/9$
no	yes	$1/5$
	no	$4/5$

P(CR | BC)

BC	CR	θ
yes	exc	$3/9$
	fair	$6/9$
no	exc	$4/5$
	fair	$1/5$

NBC PREDICTION

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no
31..40	**high**	**no**	**excellent**	**?**

- What is the probability that new person will buy a computer?

$$P(BC = yes|A = 31..40, I = high, S = no, CR = exc)$$
$$\propto P(A = 31..40|BC = yes)P(I = high|BC = yes)$$
$$P(S = no|BC = yes)P(CR = exc|BC = yes)P(BC = yes)$$

P(BC)

BC	θ
yes	$9/14$
no	$5/14$

P(A | BC)

BC	A	θ
	$<= 30$	$2/9$
yes	$31..40$	$4/9$
	> 40	$3/9$
	$<= 30$	$3/5$
no	$31..40$	$0/5$
	> 40	$2/5$

P(I | BC)

BC	I	θ
	high	$2/9$
yes	med	$4/9$
	low	$3/9$
	high	$2/5$
no	med	$2/5$
	low	$1/5$

P(S | BC)

BC	S	θ
yes	yes	$6/9$
	no	$3/9$
no	yes	$1/5$
	no	$4/5$

P(CR | BC)

BC	CR	θ
yes	exc	$3/9$
	fair	$6/9$
no	exc	$4/5$
	fair	$1/5$

IS THERE ANY PROBLEM?

X_1

Y		Low	Med	High
	Yes	10	13	17
	No	2	13	0

ZERO COUNTS ARE A PROBLEM

- If an attribute value does not occur in training data, we assign **zero** probability to that value
- How does that affect the conditional probability P[f(x) | x] ?
- It equals 0!!!
- Why is this a problem?
- Adjust for zero counts by “smoothing” probability estimates

SMOOTHING: LAPLACE CORRECTION

X_1

Y		Low	Med	High
	Yes	10	13	17
	No	2	13	0

Laplace correction

Numerator: ***add 1***

Denominator: ***add k***, *where k=number of possible values of X*

$$P[\,X_1 = \text{High} \mid Y = \text{No}\,] = \frac{0 + 1}{(2 + 13 + 0) + 3}$$

Adds uniform prior

WHAT ABOUT CONTINUOUS VARIABLES

- Discretize continuous variables through binning
  - Split the range of the continuous variable to several bins, assign a categorical value to each bin, and map continuous values fall into that bin to the assigned categorical value

- Model the probability distribution for continuous variables explicitly
  - For example, assume a Gaussian distribution and introduce additional parameters: $P(x_{ij} = x \mid c_i = l) \sim N(\mu_j^l, \sigma_j^l)$

IS ASSUMING INDEPENDENCE A PROBLEM?

- What is the effect on probability estimates?
  - Over-counting evidence, leads to overly confident probability estimate
- What is the effect on classification?
  - Less clear…
  - For a given input x, suppose f(x) = True
  - Naïve Bayes will correctly classify if P[f(x) = True | x] > 0.5
    …thus it may not matter if probabilities are overestimated

NAIVE BAYES CLASSIFIER

- Simplifying (naive) assumption: attributes are conditionally independent given the class
- Strengths:
  - Easy to implement
  - Often performs well even when assumption is violated
  - Can be learned incrementally
- Weaknesses:
  - Class conditional assumption produces skewed probability estimates
  - Dependencies among variables cannot be modeled

NBC LEARNING

- Model space
  - Parametric model with specific form (i.e., based on Bayes rule and assumption of conditional independence)
  - Models vary based on parameter estimates in CPDs
- Search algorithm
  - MLE optimization of parameters (convex optimization results in exact solution)
- Scoring function
  - Likelihood of data given NBC model form

NBC: MAP ESTIMATION

- Consider a simplified scenario: binary classification (i.e., L=2) and each attribute is binary (i.e., $K(j)$=2)
- Priors: $p_1 \sim Beta(a, b), q_l^{j1} \sim Beta(\alpha_l^j, \beta_l^j)$
- MAP estimate:
  - Maximize $P(D \mid \theta)P(\theta)$

NBC: MAP ESTIMATION

$$P(D\,|\,\theta)P(\theta) = (\prod_{i=1}^{n}\prod_{j=1}^{m} P(x_{ij}\,|\,c_i)P(c_i)) \times P(p_1) \times \prod_{l=0}^{1}\prod_{j=1}^{m} P(q_l^{j1})$$

$$= \prod_{l=0}^{1} p_l^{N_l} \prod_{l=0}^{1}\prod_{j=1}^{m}\prod_{k=0}^{1} (q_l^{jk})^{N_l^{jk}} \times P(p_1) \times \prod_{l=0}^{1}\prod_{j=1}^{m} P(q_l^{j1})$$

$$\Downarrow$$

$$p_1 \sim Beta(a + N_1, b + N_0), q_l^{j1} \sim Beta(\alpha_l^{j1} + N_l^{j1}, \beta_l^{j1} + N_l^{j0})$$

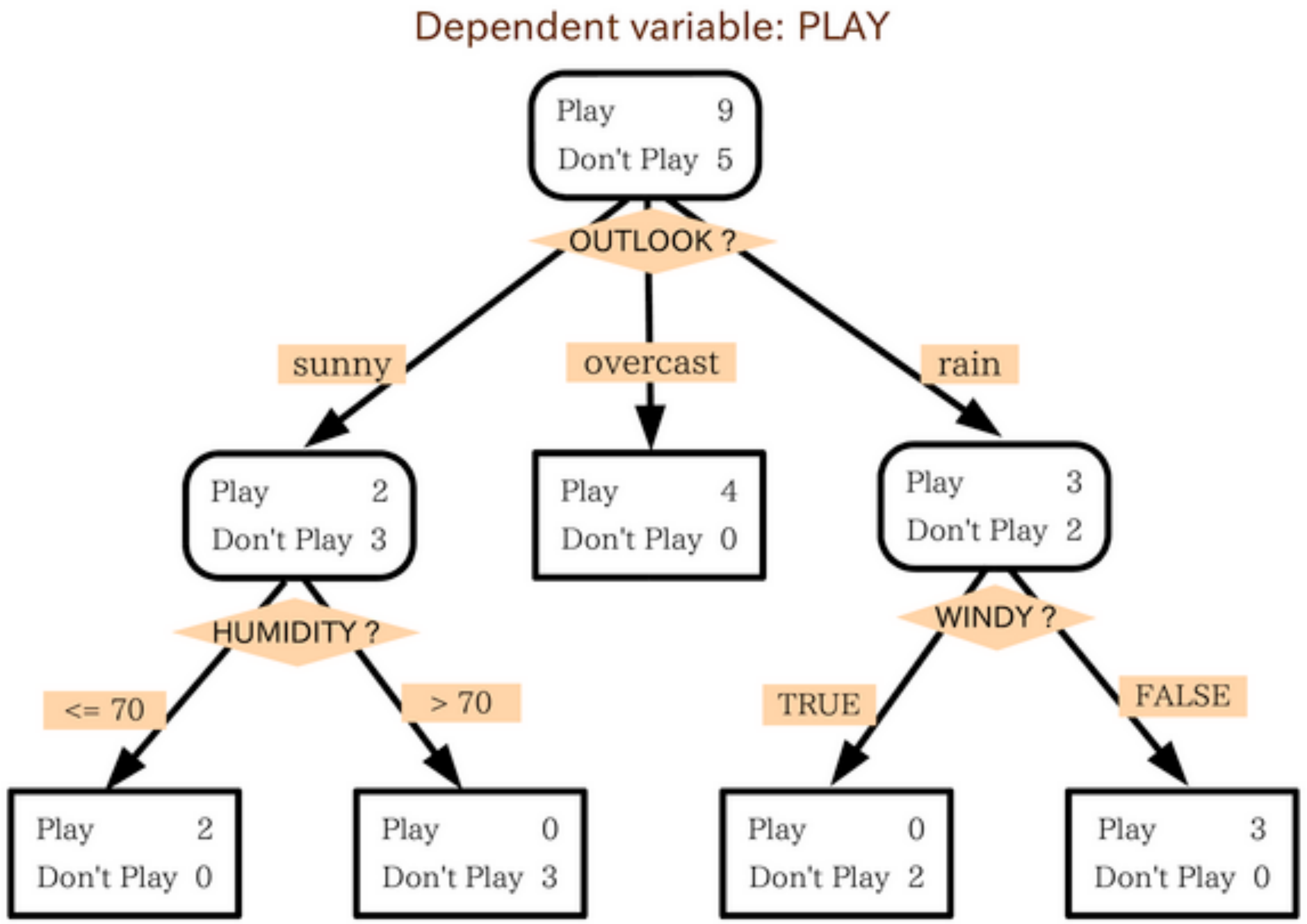
$$\Downarrow$$

$$[p_1]_{MAP} = \frac{a + N_1 - 1}{a + b + n - 2}, [q_l^{l1}]_{MAP} = \frac{\alpha_l^{j1} + N_l^{j1} - 1}{\alpha_l^{j1} + \beta_l^{j1} + N_l - 2}$$

DECISION TREES

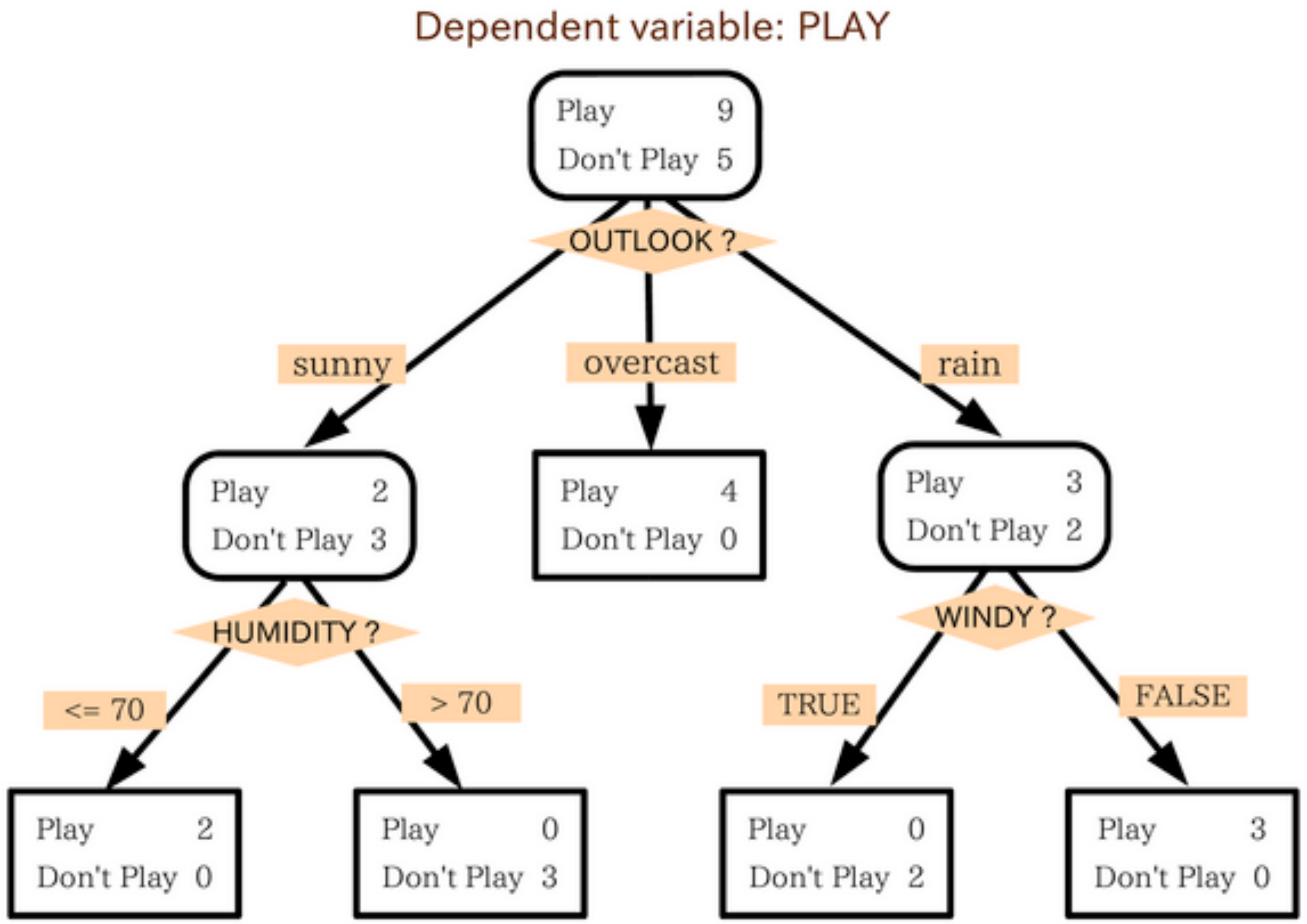
TREE MODELS: KNOWLEDGE REPRESENTATION

- A decision tree has 2 kinds of nodes
  - Each internal node is a question on features. It branches out according to the answers
  - Each leaf node has a class label, determined by the majority vote of training examples reaching that leaf
- Advantages
  - Easy inference
  - Can handle mixed variables
  - Easy for humans to understand



TREE LEARNING

- Model space: All possible decision trees
  - Each layer can include different attributes
  - Each attribute can split on different values
  - Can have different number of layers
- Scoring function: Misclassification rate
- Search process: Heuristic search
  - Greedy, recursive divide and conquer

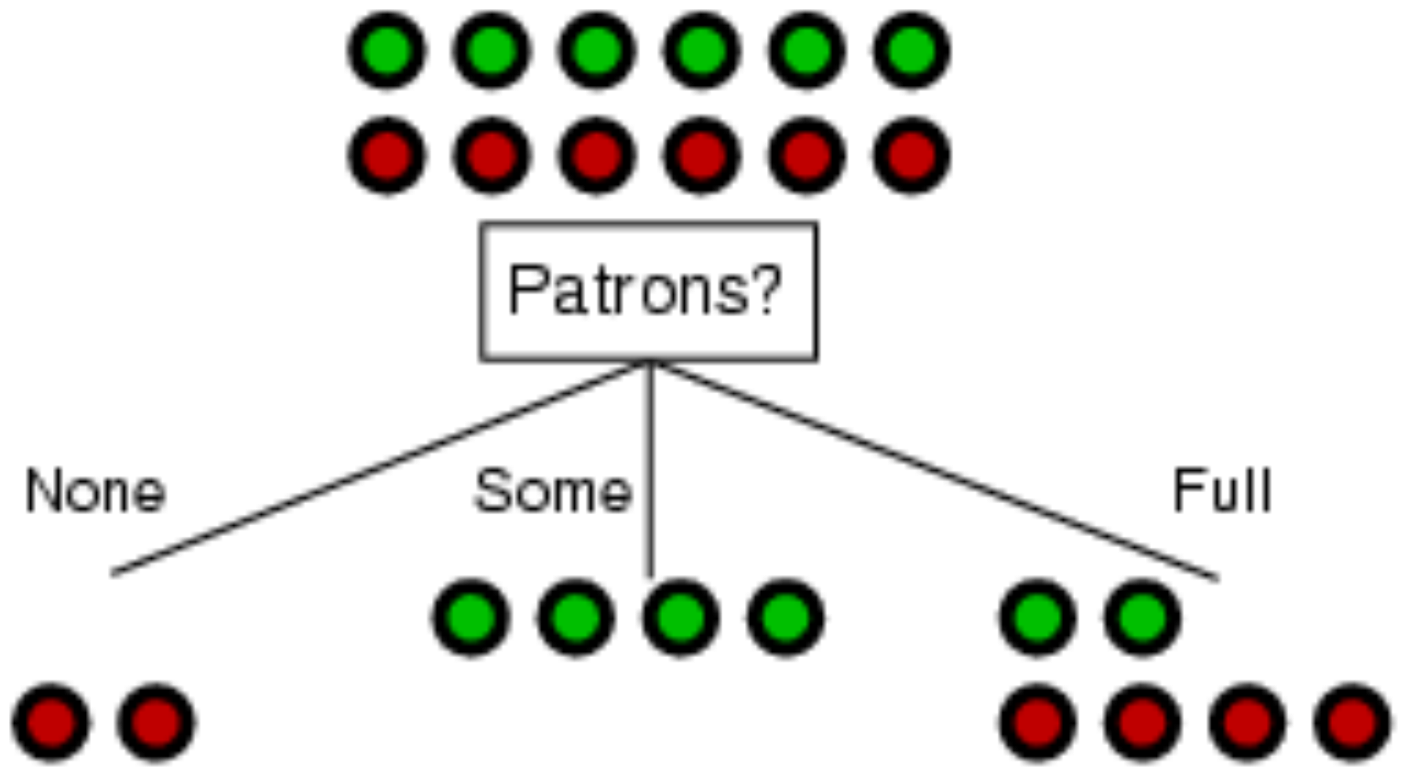


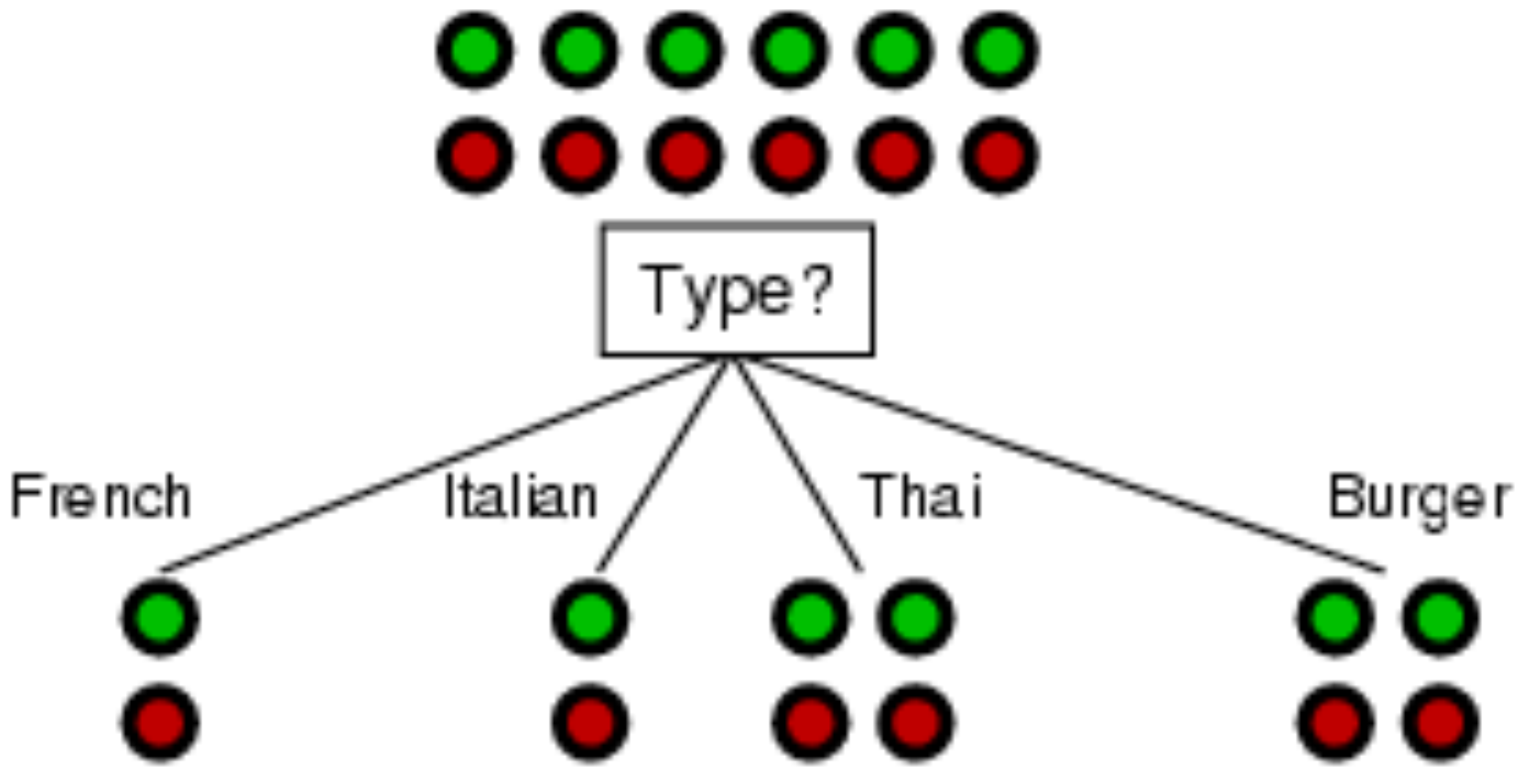
TREE LEARNING

- Top-down recursive divide and conquer algorithm
  - Start with all training examples at root
  - Select **best** attribute/feature: Take a greedy view to decide how "good" an attribute is
  - Partition examples by selected attribute
  - Recurse and repeat
- Other issues:
  - When to stop growing
  - Pruning irrelevant parts of the tree

CHOOSING AN ATTRIBUTE/FEATURE

- Be greedy: choose an attribute that can immediately minimize the misclassification rate (i.e., as if no further subtree will grow)
- A good feature splits the examples into subsets that distinguish among the class labels as much as possible... ideally into pure sets of "all positive" or "all negative"

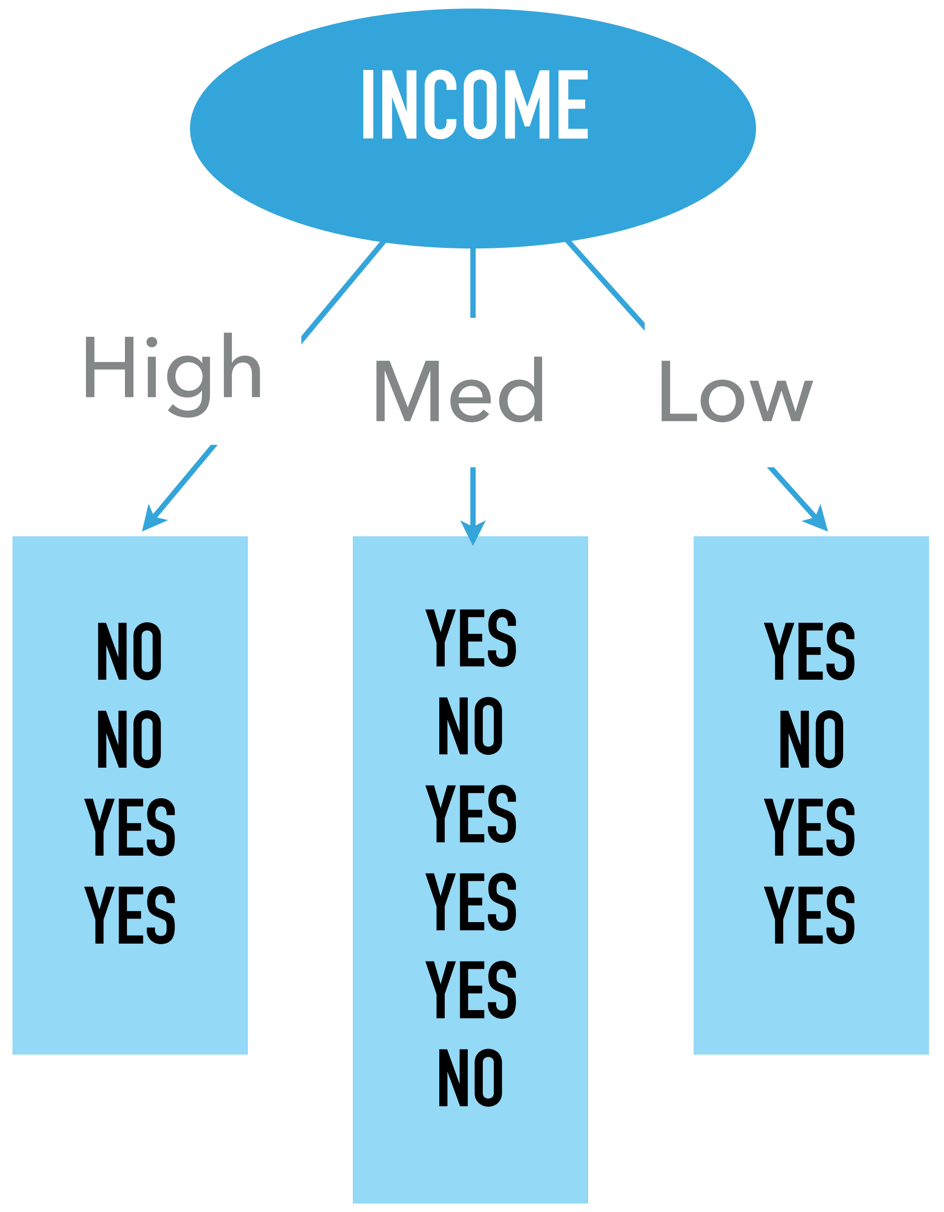




ASSOCIATION BETWEEN ATTRIBUTE AND CLASS LABEL

Data

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no



Contingency table

Class label value

Attribute value	Buy	No buy
High	2	2
Med	4	2
Low	3	1

A good attribute leads to **highly certain** prediction for training examples sharing the same value on that attribute!