Decision Tree Classification

AI42001

31 July 2019

Data Representation

- Each data-point represented by D-dimensional feature vector X_i
- Animal classification: [#legs, #tail, colour, size, weight]
- Some of these features more useful for classification
- Sometimes, a single feature is enough to classify

Feature Selection

- Cat vs Snake classification
- “#legs” feature is sufficient!

- Classifier function: #legs = 4: cat; #legs = 0: snake
- Decision function!

- For Cat vs Dog classification, “#legs” is certainly not sufficient
- It is not a “discriminative feature!”

What’s a discriminative feature?

- X1 ={YELLOW, WHITE}, X2 = real number, Y = {CAT,DOG}

	X1=YELLOW	**X1=WHITE**	
#(Y=CAT)	52	48	100
#(Y=DOG)	47	53	100
Total	99	101	200

What’s a discriminative feature?

- X1 ={YELLOW, WHITE}, X2 = real number, Y = {CAT,DOG}

	X2< 5	X2 > 5	
#(Y=CAT)	5	95	100
#(Y=DOG)	1	99	100
Total	6	194	200

What’s a discriminative feature?

- X1 ={YELLOW, WHITE}, X2 = real number, Y = {CAT,DOG}

	X2< 15	**X2 > 15**	
#(Y=CAT)	95	5	100
#(Y=DOG)	10	90	100
Total	105	95	200

What’s a discriminative feature?

- Prob(Y = CAT | X1 = YELLOW) ~ 0.5
- Prob(Y = CAT | X1 = WHITE) ~ 0.5

- Prob(Y = CAT | X2 < 5) ~ 0.9
- Prob(Y = CAT | X2 > 5) ~ 0.5

- Prob(Y = CAT | X2 < 15) ~ 0.9
- Prob(Y = CAT | X2 > 15) ~ 0.1

What’s a discriminative feature?

- Prob(Y = CAT | X1 = YELLOW) ~ 0.5 [Hard to decide]
- Prob(Y = CAT | X1 = WHITE) ~ 0.5 [Hard to decide]

- Prob(Y = CAT | X2 < 5) ~ 0.9 [Easy to decide][Very few examples]
- Prob(Y = CAT | X2 > 5) ~ 0.5 [Hard to decide]

- Prob(Y = CAT | X2 < 15) ~ 0.9 [Easy to decide]
- Prob(Y = CAT | X2 > 15) ~ 0.1 [Easy to decide]

- [X2 <> 15] is a “discriminative feature”, allows easy decisions either way!

Decision Tree Algorithm

- Idea: identify the “most discriminative” feature, use it to classify!
- Problem 1: How to quantify “discriminative-ness”?
- Problem 2: What if no feature is very discriminative?

Decision Tree Algorithm

- Idea: identify the "most discriminative" feature, use it to classify!

- Problem 1: How to quantify "discriminative-ness"?

  - entropy!

- Problem 2: What if no feature is very discriminative?

  - try a sequence of features!

Entropy: measure of discriminativeness

- P(Y=1) = 0.5, p(Y=2) = 0.5 : low discriminative ability
- P(Y=1) = 0.9, p(Y=2) = 0.1 : high discriminative ability

$$\boxed{H = -\sum_i p_i (\log_2 p_i)}$$

- Case 1: H = 1
- Case 2: H = 0.47

Feature selection based on entropy

- Before split: #(Y=cat) = 100, #(Y=dog) = 100. Entropy = 1.

	X1=YELLOW	X1=WHITE	
#(Y=CAT)	52	48	100
#(Y=DOG)	47	53	100
Total	99 (Entropy ~ 1)	101 (Entropy ~ 1)	200

- Information gain =

Original Entropy – (Split1_size*Split1_ entropy + Split2_size*Split2_ entropy)

1 – (99/200*1 + 101/200*1) ~ 0!

Feature selection based on entropy

- Before split: #(Y=cat) = 100, #(Y=dog) = 100. Entropy = 1.

	X2 < 15	X2 > 15	
#(Y=CAT)	95	5	100
#(Y=DOG)	10	90	100
Total	105 (Entropy = 0.45)	95 (Entropy = 0.30)	200

- Information gain =

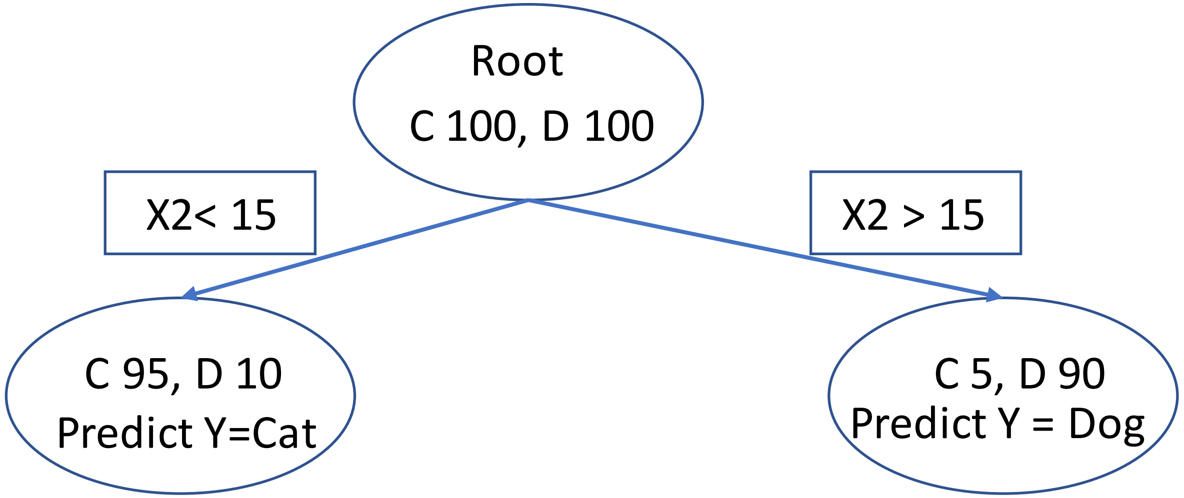
Original Entropy – (Split1_size*Split1_ entropy + Split2_size*Split2_ entropy)

1 – (105/200*0.45 + 95/200*0.3) = 0.62!!

Feature selection based on entropy

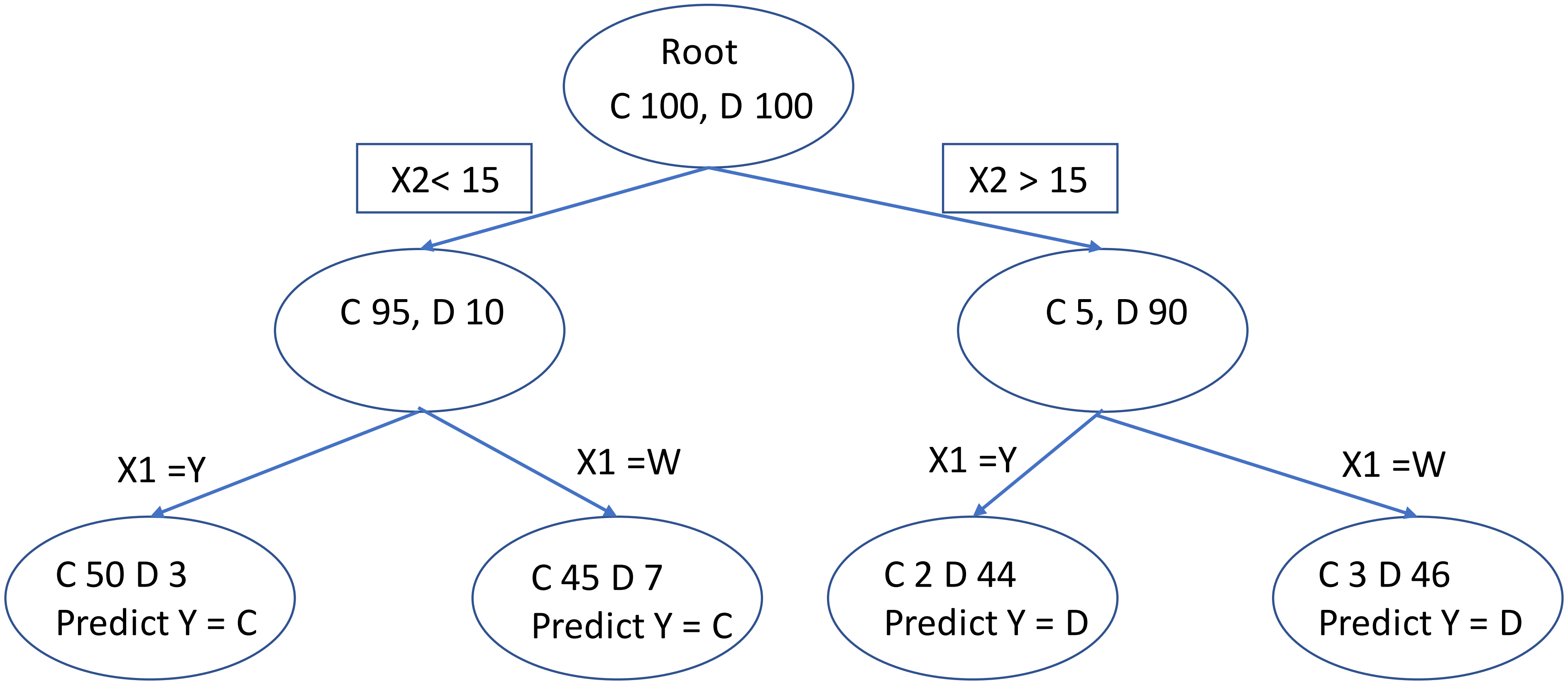
- Each discrete feature splits the dataset
- Continuous features can always be converted to discrete
- “Pure” dataset: - disbalanced class distribution
  - low entropy
  - high information gain
- Choose that feature which provides most information gain!

Decision Stump



- Training accuracy: 95/100 for cats, 90/100 for dogs

Decision Tree



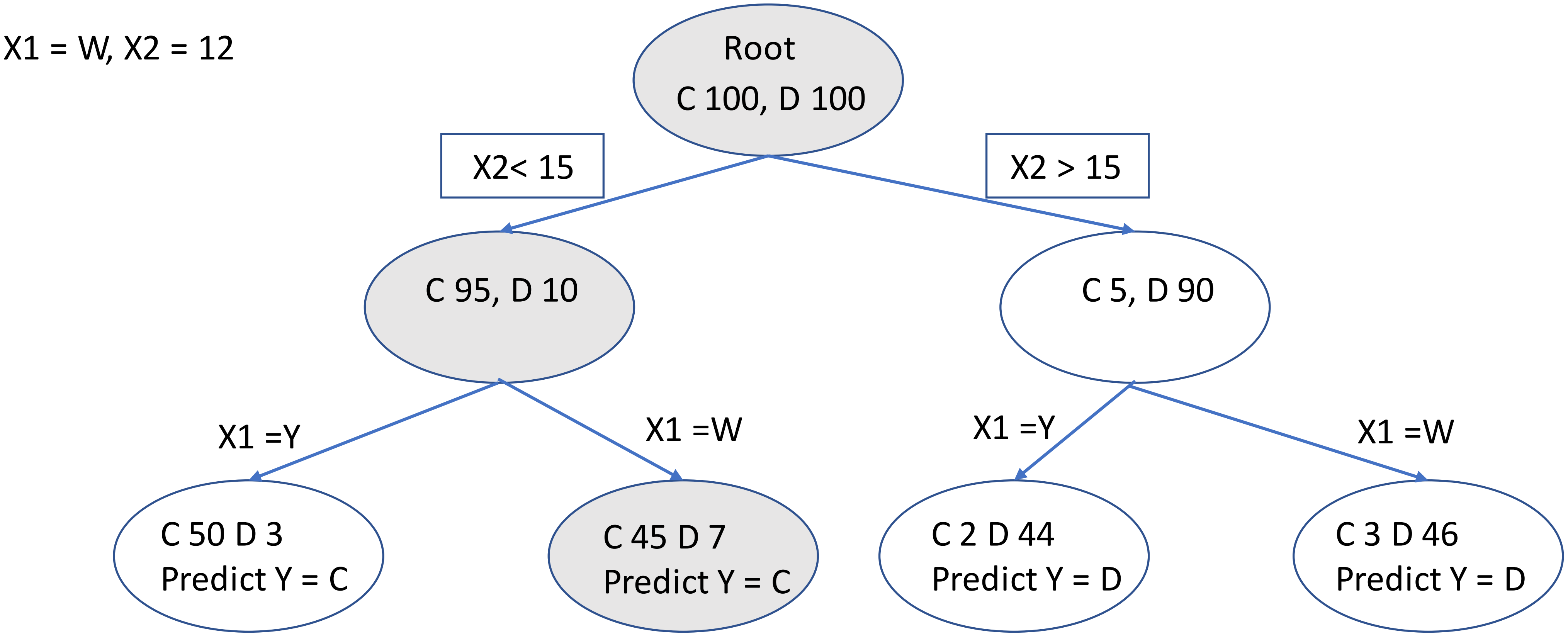
- Does this split provide "information gain"???
- If yes, split. If no, stop at previous step

Decision Tree algorithm

- 1. Identify the feature that results in maximum information gain
- 2. Split the dataset accordingly
- 3. Identify if any feature can result in further information gain on the split sets
- 4. If yes, split further. If no, stop.
- 5. Goto 3
- 6. At each leaf, the prediction is the mode label

- Test:
- Follow the sequence of decisions based on the features of test example
- Make prediction according to leaf

Decision Tree for Testing



- Prediction: Y= C

Advantages and Disadvantages

Advantage:

- Easy to interpret
- Easy to classify at test time
- Provides a ranking of features (according to usefulness)

Disadvantages:

- No optimal solution known, IG is just heuristic, can create many small branches
- Can cause overfitting if tree grows deep (need to stop growing)

Regression Trees

- Decision trees can also be used for regression
- Measure of homogeneity at each node: variance of labels (instead of entropy)
- Split criteria: reduction in total variance (instead of information gain)
- Final prediction: Mean label in the leaf node (instead of mode)