CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

Admin

- Office hours: TODAY 4:30-6pm

- Lab 1 due Tues night (TODAY)
  - Make sure to push your final code on git
  - Math portion on git too
  - Grace period til Wed at noon

- For Friday: Duame 1.3-1.6 and chapter 2

- Lab 2 posted tonight

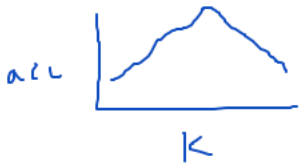
Outline for September 15

- Finish KNN
- Recap featurization
- Overfitting
- Decision Trees
- Entropy

Outline for September 15

- Finish KNN
- Recap featurization
- Overfitting
- Decision Trees
- Entropy

Multiclass KNN



$K = 10$

$N_k(\vec{x})$ = set of neighbors of $\vec{x}$

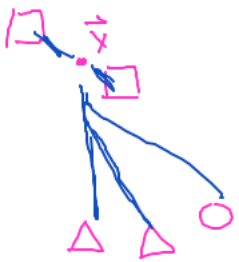
$N_3(\vec{x}) = \{\vec{x}_{121}, \vec{x}_{57}, \vec{x}_{23}\}$

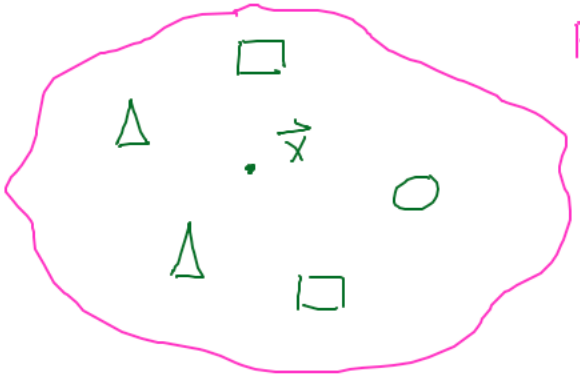
$\text{sign}(z) \begin{cases} +1 & z > 0 \\ -1 & z \leq 0 \end{cases}$

$$p(y = c \mid \vec{x}) = \frac{1}{K} \sum_{\vec{x}_i \in N_K(\vec{x})} \mathbb{I}(y_i = c)$$

specific label

$K = 5$





$p(y = \square \mid \vec{x}) = \frac{2}{5}$

$p(y = \bigcirc \mid \vec{x}) = \frac{1}{5}$

$p(y = \triangle \mid \vec{x}) = \frac{2}{5}$

K-nearest neighbors creates implicit decision boundaries

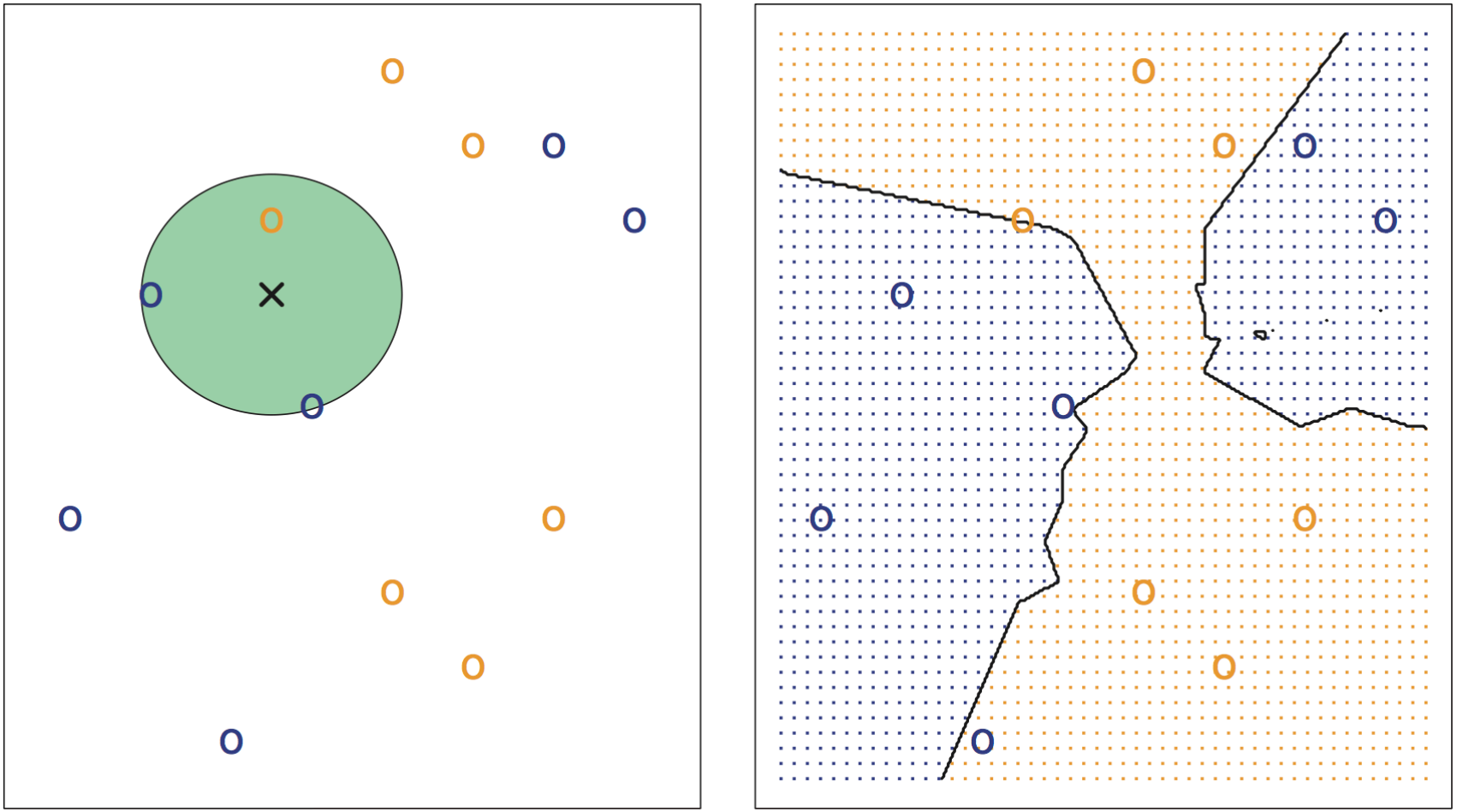


Figure 2.14 from ISL book, KNN with two classes (C=2), and K=3

Outline for September 15

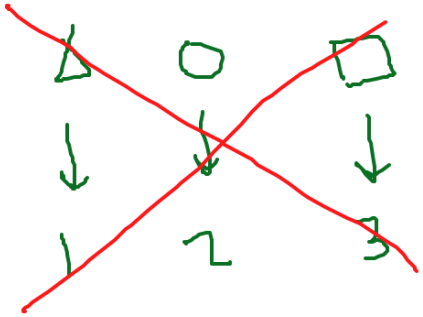
- Finish KNN
- Recap featurization
- Overfitting
- Decision Trees
- Entropy

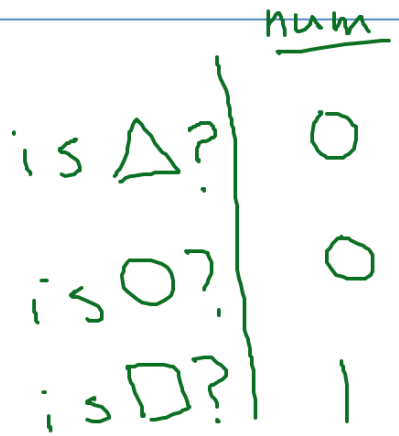
Feature Terminology

- *Features:* feature names
  - i.e. shape
- *Feature values:* what values are possible
  - i.e. {circle, square, triangle}
- *Feature vector:* values for a particular example
  - i.e. $\boldsymbol{x} = [x_1, x_2, x_3, \ldots, x_p]$
- *Decision boundary*: separates regions of the feature space that would be classified as positive or negative (or multiclass)

Featurization: make continuous

- Real-valued features get copied directly. *Duame, Chap 3*
- Binary features become 0 (for false) or 1 (for true).
- Categorical features with V possible values get mapped to V-many binary indicator features.





Haven't discussed:

-normalization

-categorical variables on a spectrum

Outline for September 15

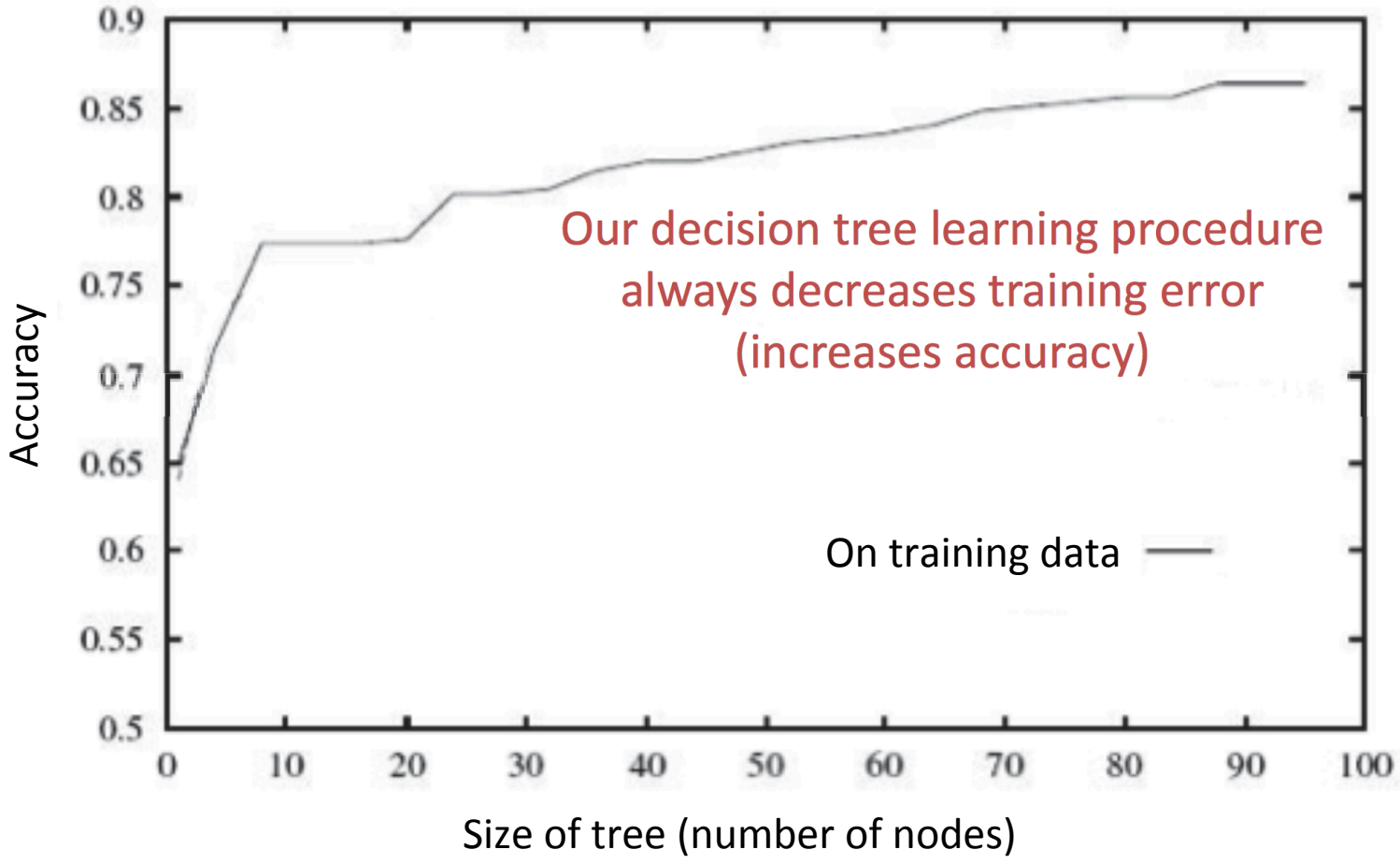
- Finish KNN
- Recap featurization
- Overfitting
- Decision Trees
- Entropy

Overfitting Terminology

- *Underfitting*: "had the opportunity to learn something but didn't" (Duame)

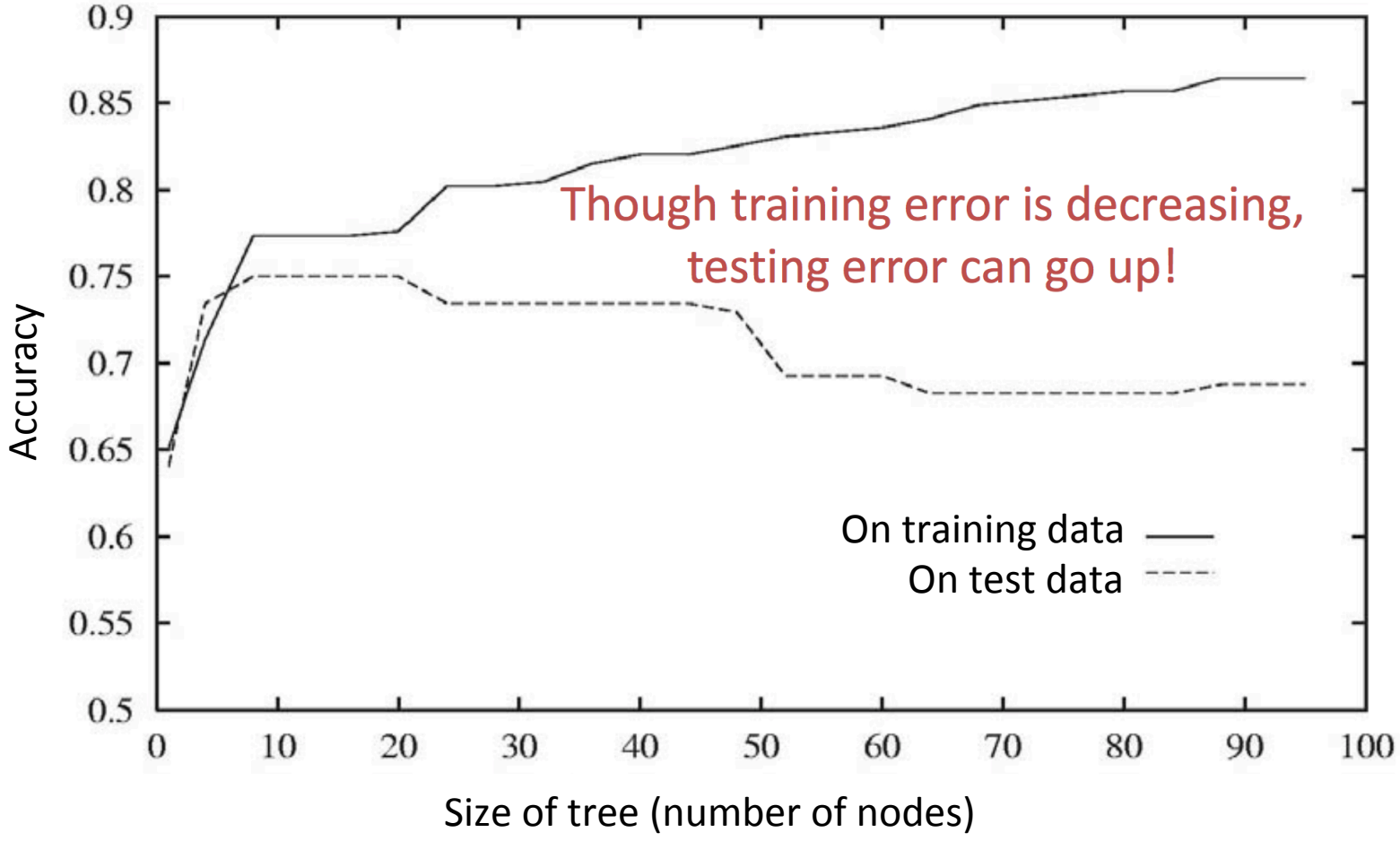
- *Overfitting*: memorized individual training examples (fit to noise) and can't generalize

Overfitting



Slide: modified from Jessica Wu
Based on slide by David Kauchak
(originally by Pedro Domingos)

Overfitting



Slide: modified from Jessica Wu
Based on slide by David Kauchak
(originally by Pedro Domingos)

Overfitting definition

- Consider a hypothesis (tree): *h*
  - Training error: $error_{train}(h)$
  - Error over all possible data: $error_{D}(h)$

Overfitting definition

- Consider a hypothesis (tree): *h*
  - Training error: $error_{train}(h)$
  - Error over all possible data: $error_{D}(h)$

- A hypothesis *h* **overfits** training data if there exists another hypothesis *h'* s.t.
  - $error_{train}(h) < error_{train}(h')$ AND
  - $error_{D}(h) > error_{D}(h')$

Handout 1 (on Piazza)

Comparison of decision boundaries

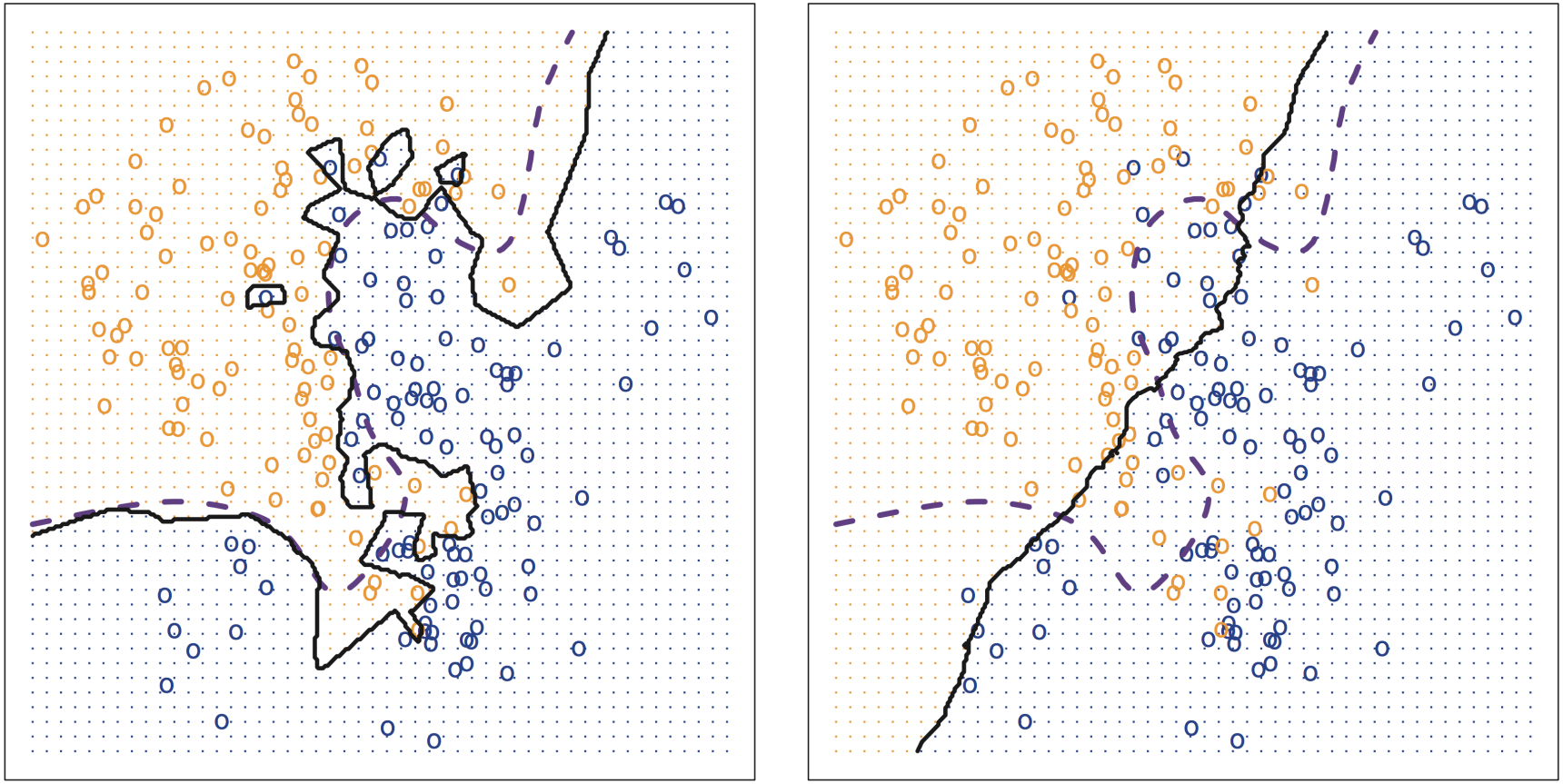


Figure 2.16 from ISL book (dashed line is “ideal” boundary)

Comparison of decision boundaries

KNN: K=1

KNN: K=100

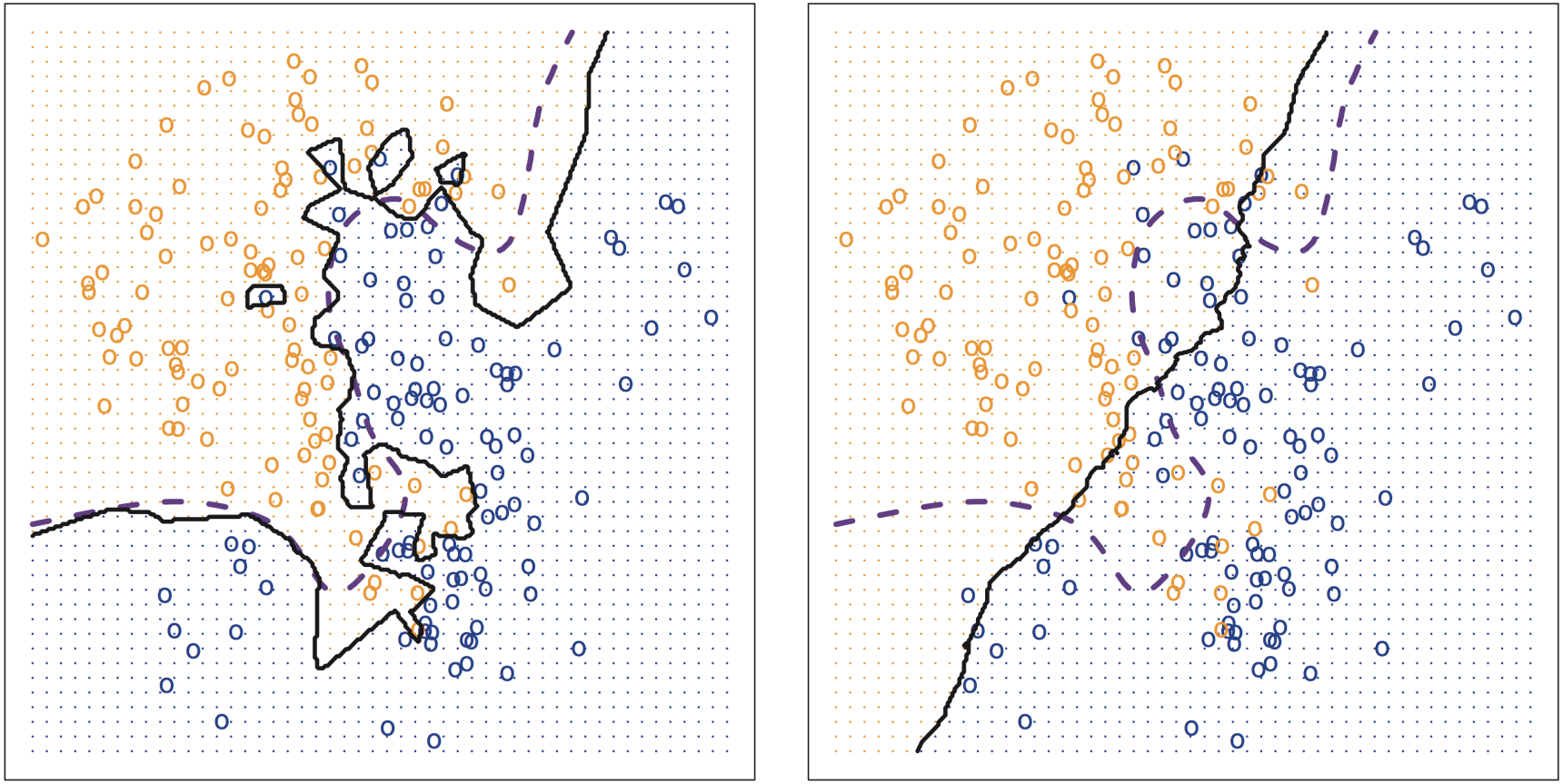
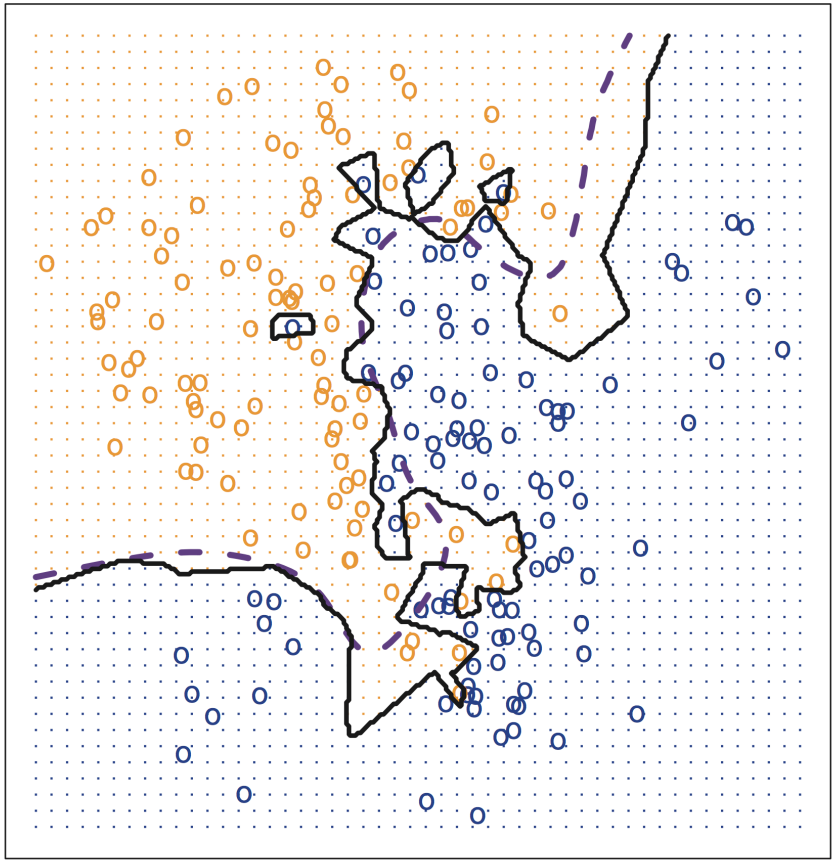


Figure 2.16 from ISL book (dashed line is “ideal” boundary)

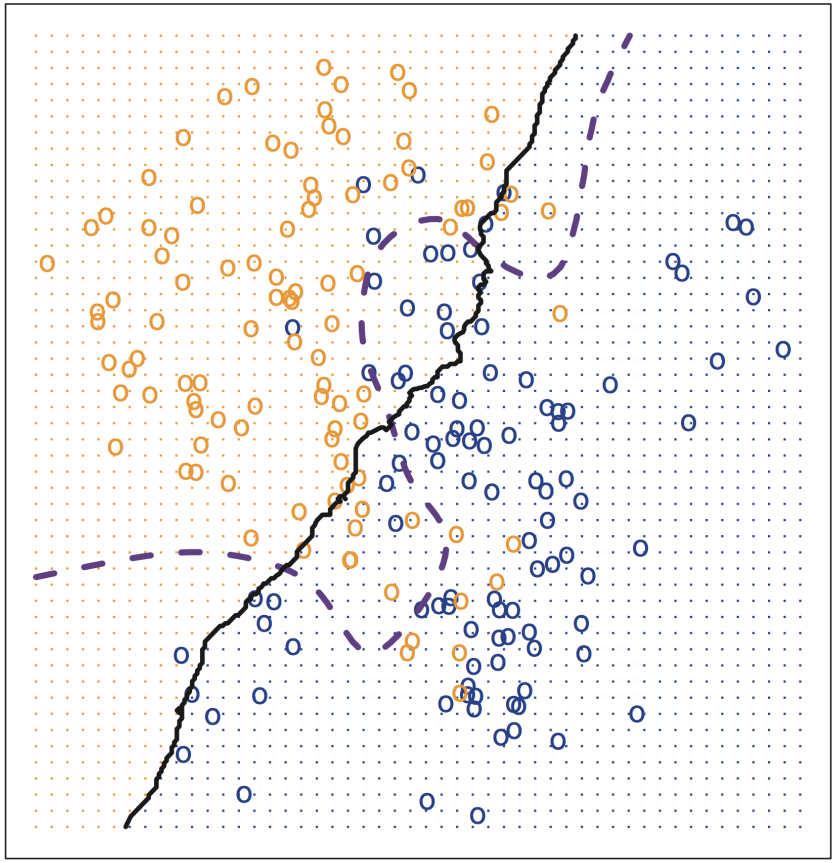
Comparison of decision boundaries

KNN: K=1



Overfitting

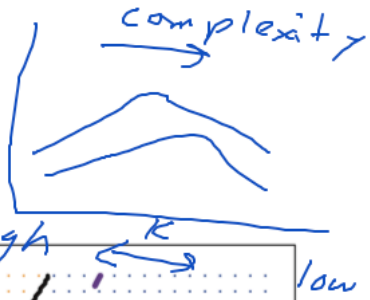
KNN: K=100



Underfitting

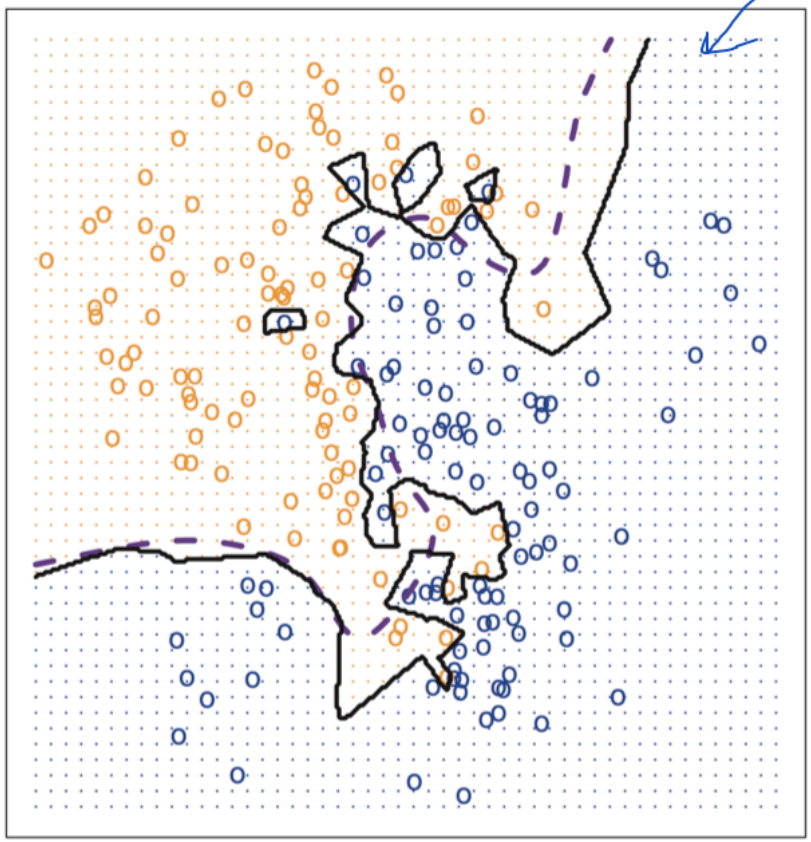
Figure 2.16 from ISL book (dashed line is “ideal” boundary)

Comparison of decision boundaries



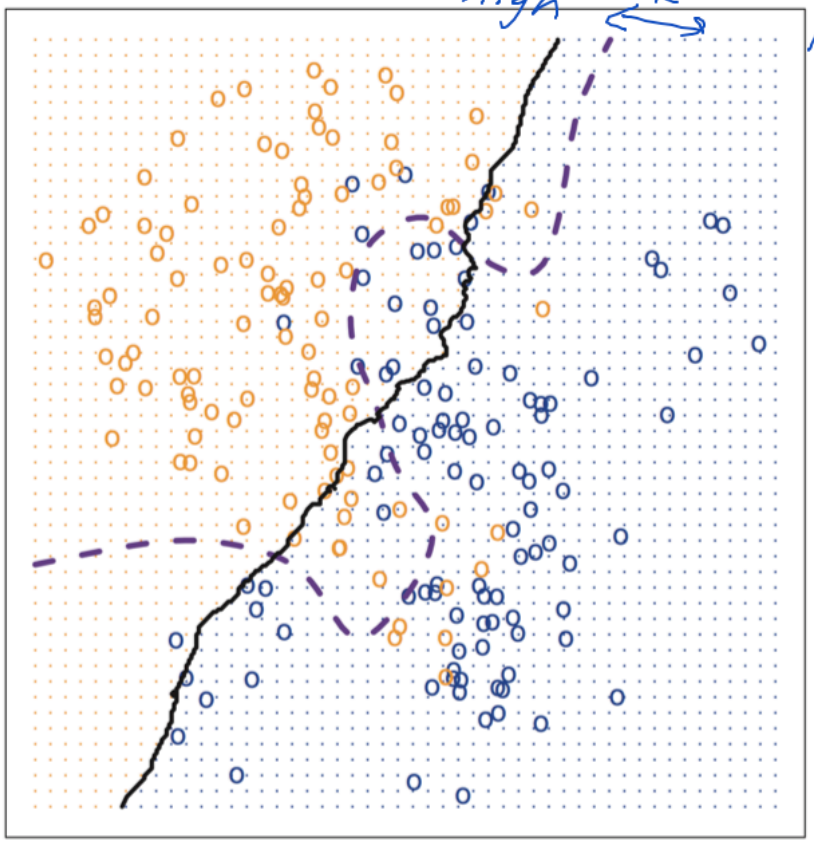
KNN: K=1

more complex



Overfitting

KNN: K=100



Underfitting

Figure 2.16 from ISL book (dashed line is "ideal" boundary)

Handout 1

4. Using your response from the previous question, what would the *feature vector* become for $\boldsymbol{x}_1$?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes

Data from Machine Learning by Tom Mitchell (Table 3.2)

Handout 1

4. Using your response from the previous question, what would the *feature vector* become for $\boldsymbol{x}_1$?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes

Data from Machine Learning by Tom Mitchell (Table 3.2)

Sunny:	{0,1}	
Overcast:	{0,1}	
Rain:	{0,1}	
Temperature:	{0, 1, 2}	(Cool, Mild, Hot)
Humidity:	{0,1}	(Normal, High)
Wind	{0,1}	(Weak, Strong)

Handout 1

4. Using your response from the previous question, what would the *feature vector* become for $\boldsymbol{x}_1$?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes

Data from Machine Learning by Tom Mitchell (Table 3.2)

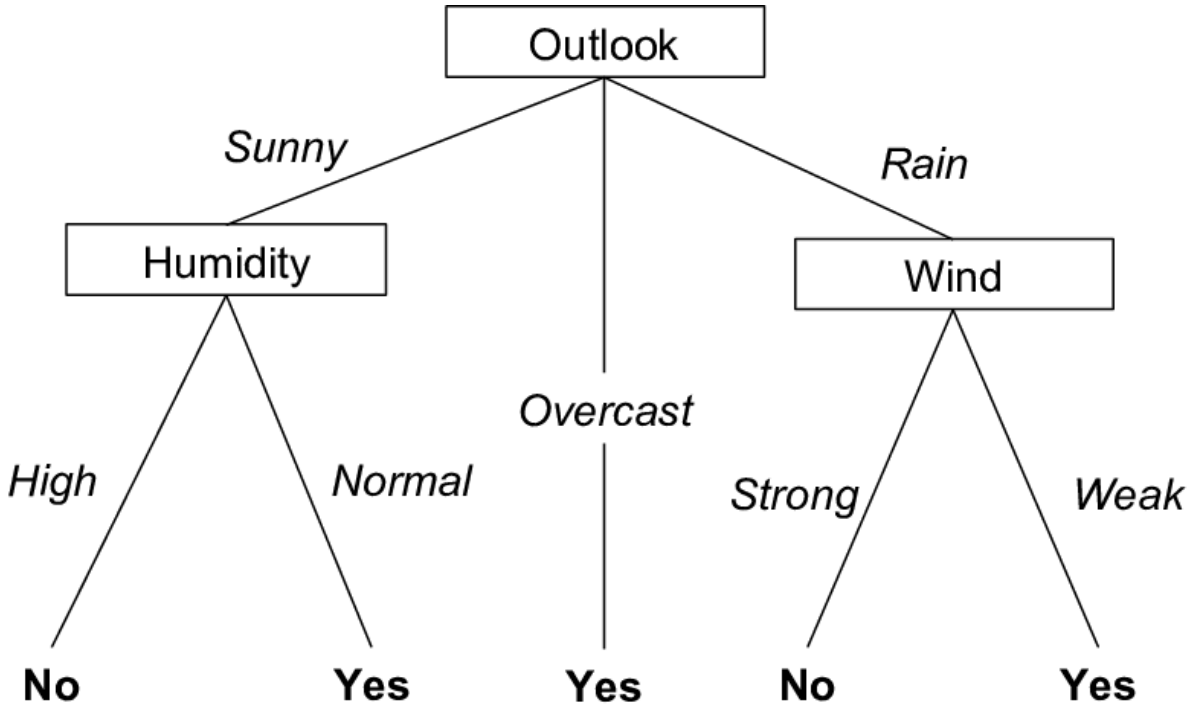
Sunny:	{0,1}	
Overcast:	{0,1}	
Rain:	{0,1}	
Temperature:	{0, 1, 2}	(Cool, Mild, Hot)
Humidity:	{0,1}	(Normal, High)
Wind	{0,1}	(Weak, Strong)

	Sunny	Overcast	Rain	Temp	Humidity	Wind
$\boldsymbol{x}_1$	1	0	0	2	1	0

Outline for September 15

- Finish KNN
- Recap featurization
- Overfitting
- Decision Trees
- Entropy

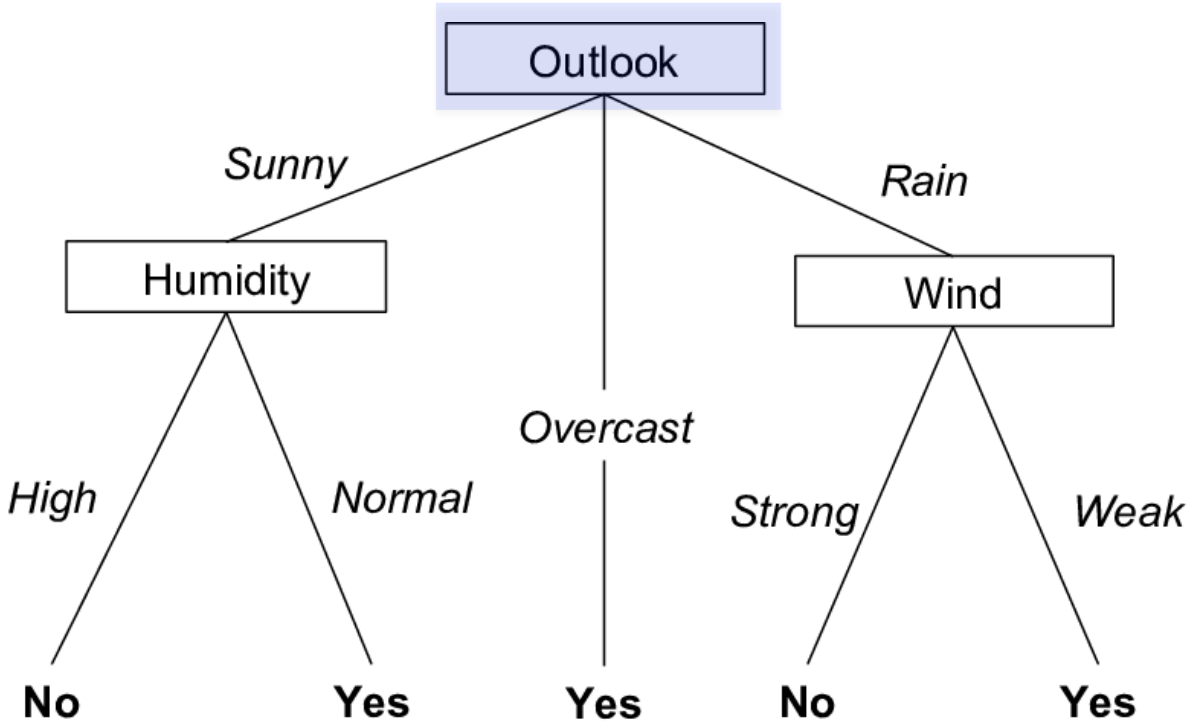
Decision Tree example (tennis data)



- Each internal node: test one feature
- Each branch from node: selects one value of the feature
- Each leaf node: predict *y*

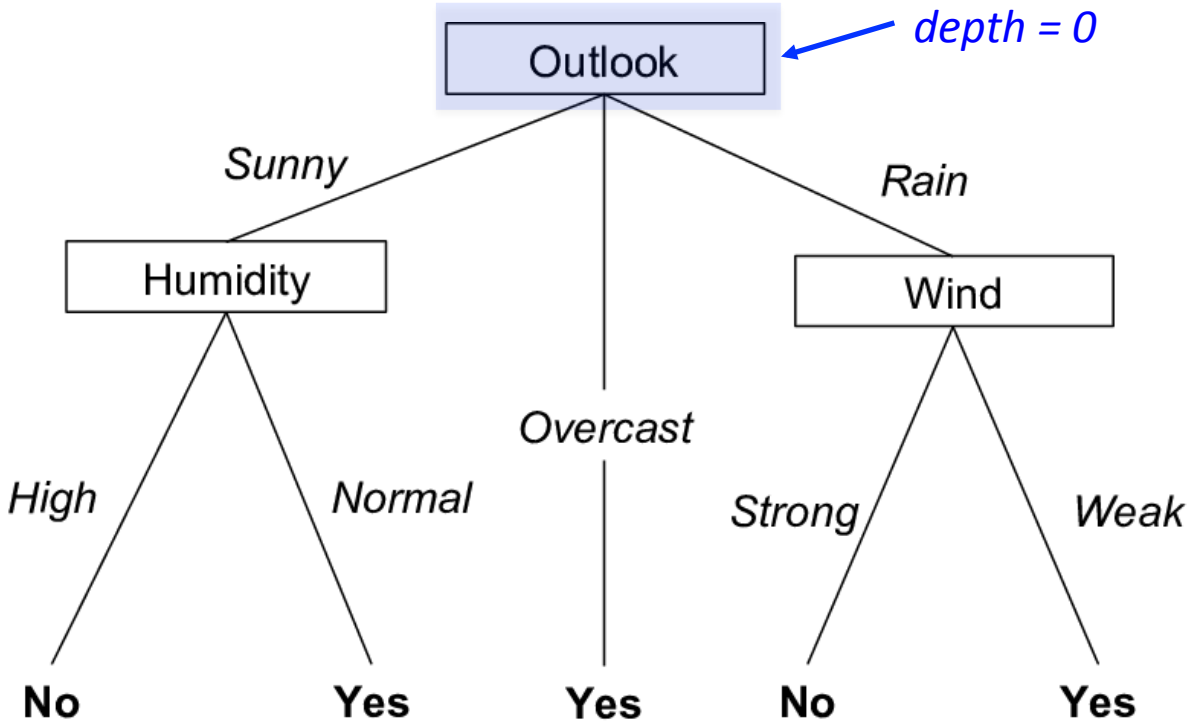
Based on slides by Jessica Wu and Eric Eaton
[originally by Tom Mitchell]

Decision Tree example (tennis data)



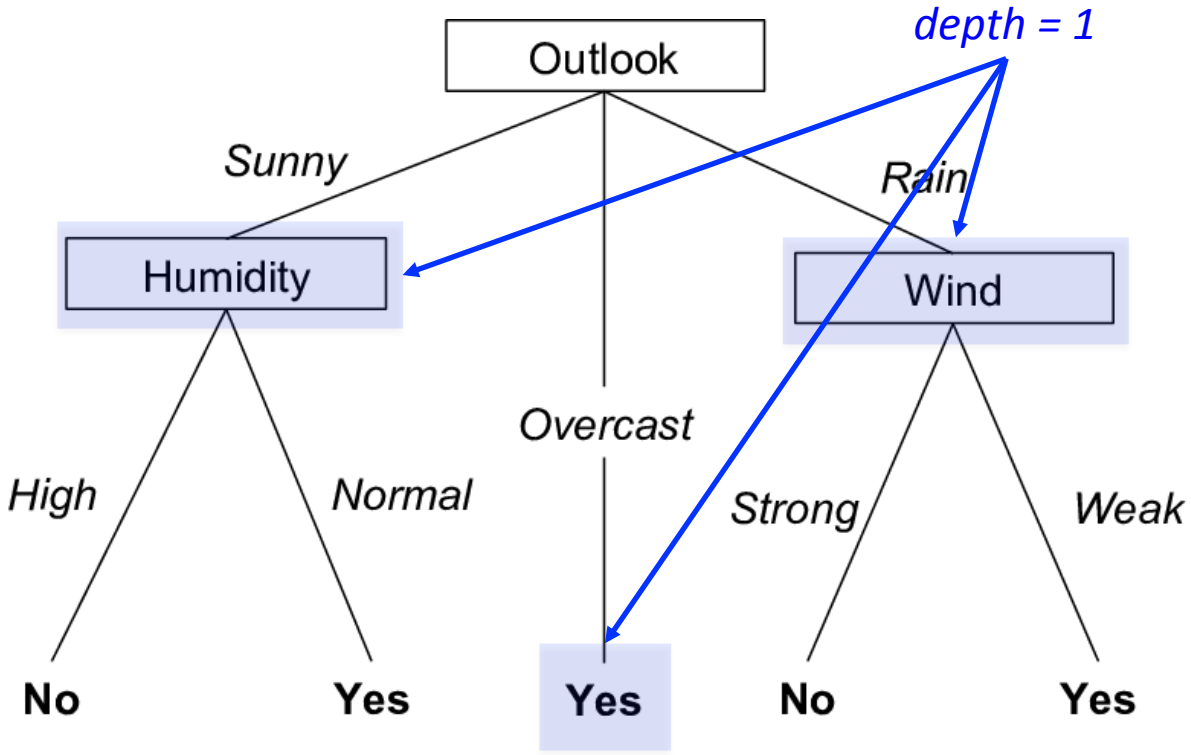
Key term: *depth*

Decision Tree example (tennis data)



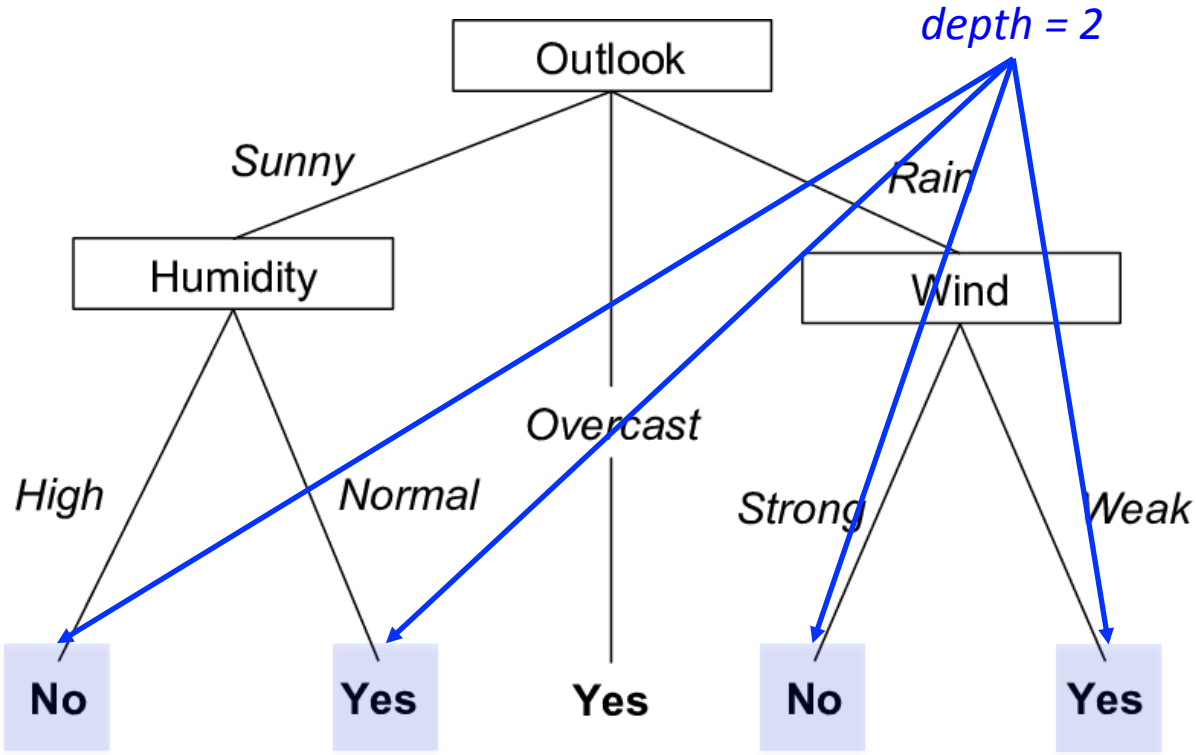
Key term: *depth*

Decision Tree example (tennis data)



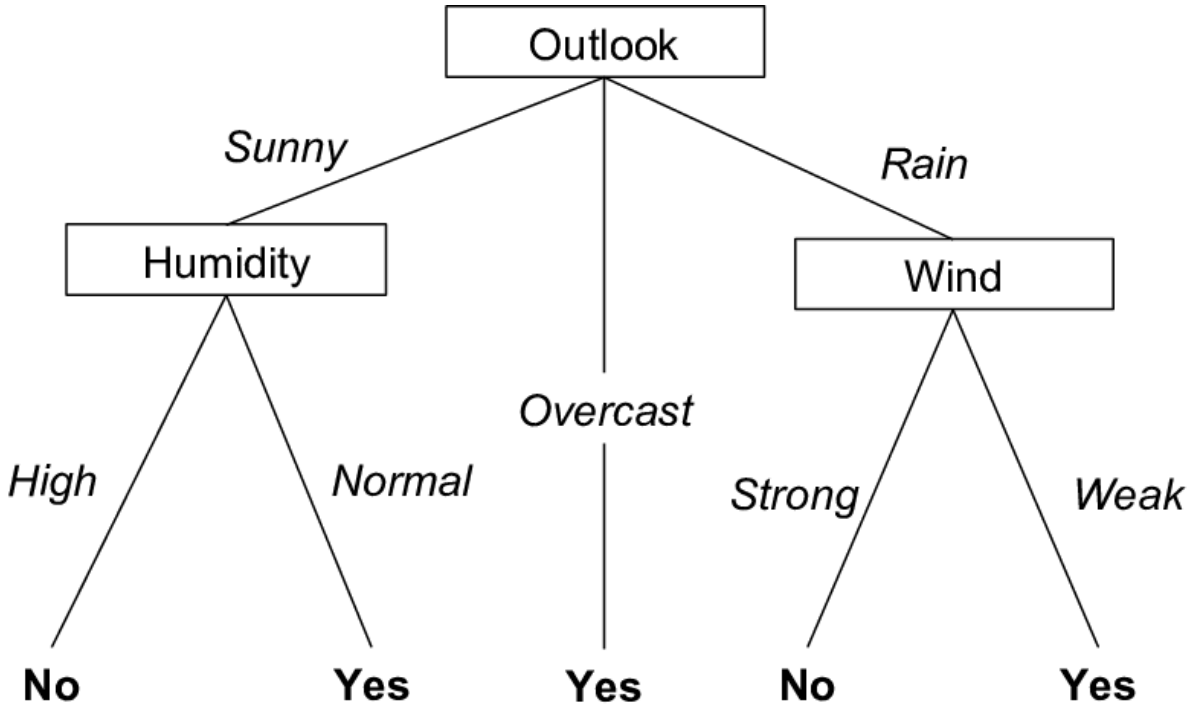
Key term: *depth*

Decision Tree example (tennis data)



Key term: *depth*

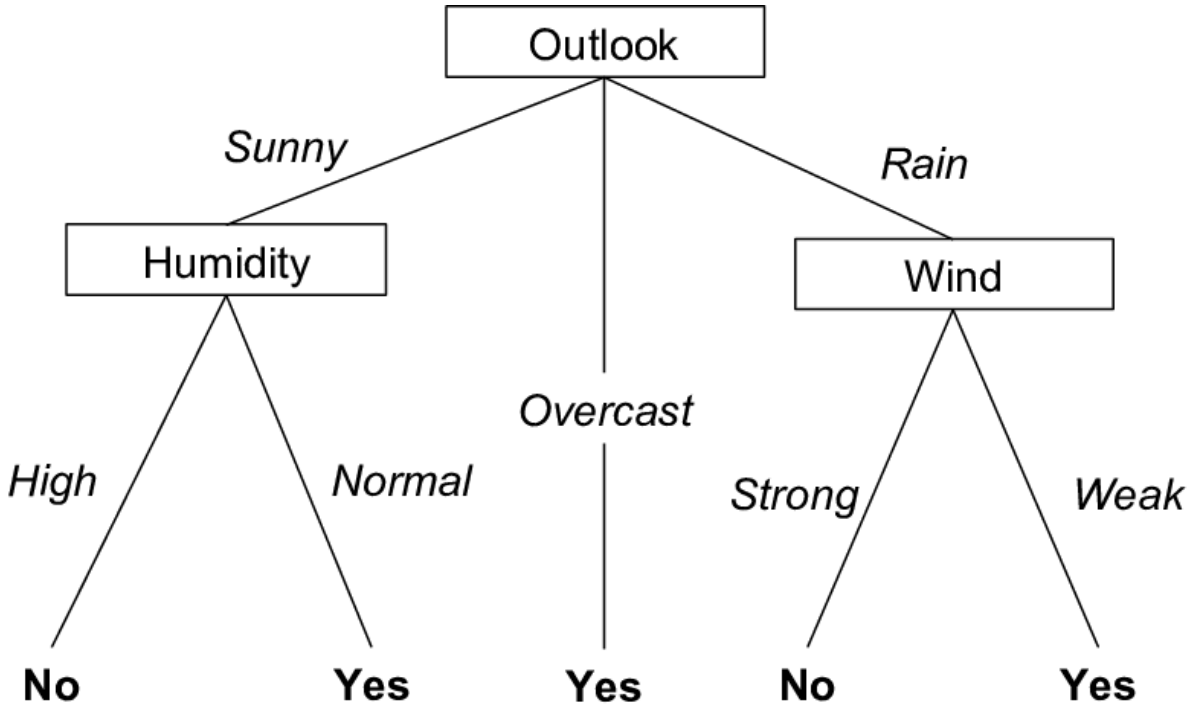
Decision Tree example (tennis data)



(test example) ***x*** =

Outlook	Temp	Humidity	Wind
Rain	Hot	High	Strong

Decision Tree example (tennis data)

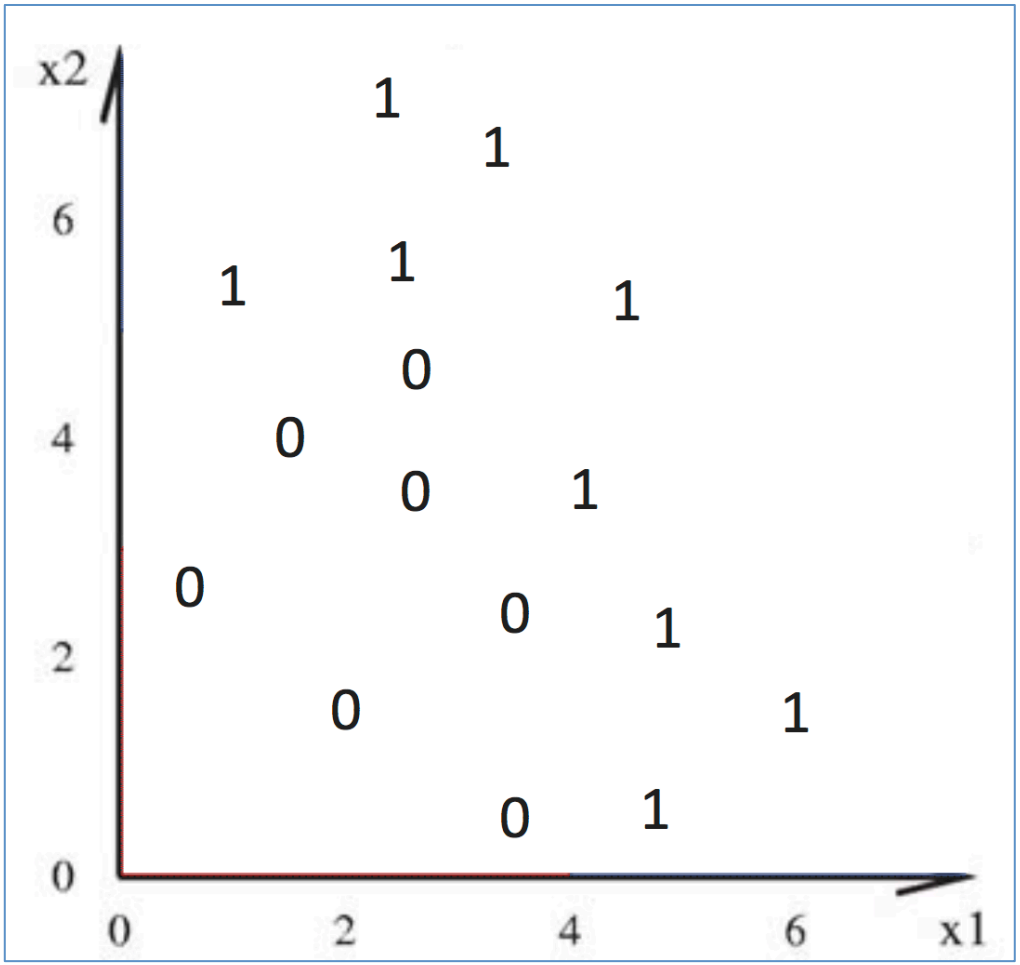


(test example) ***x*** =

Outlook	Temp	Humidity	Wind
Rain	Hot	High	Strong

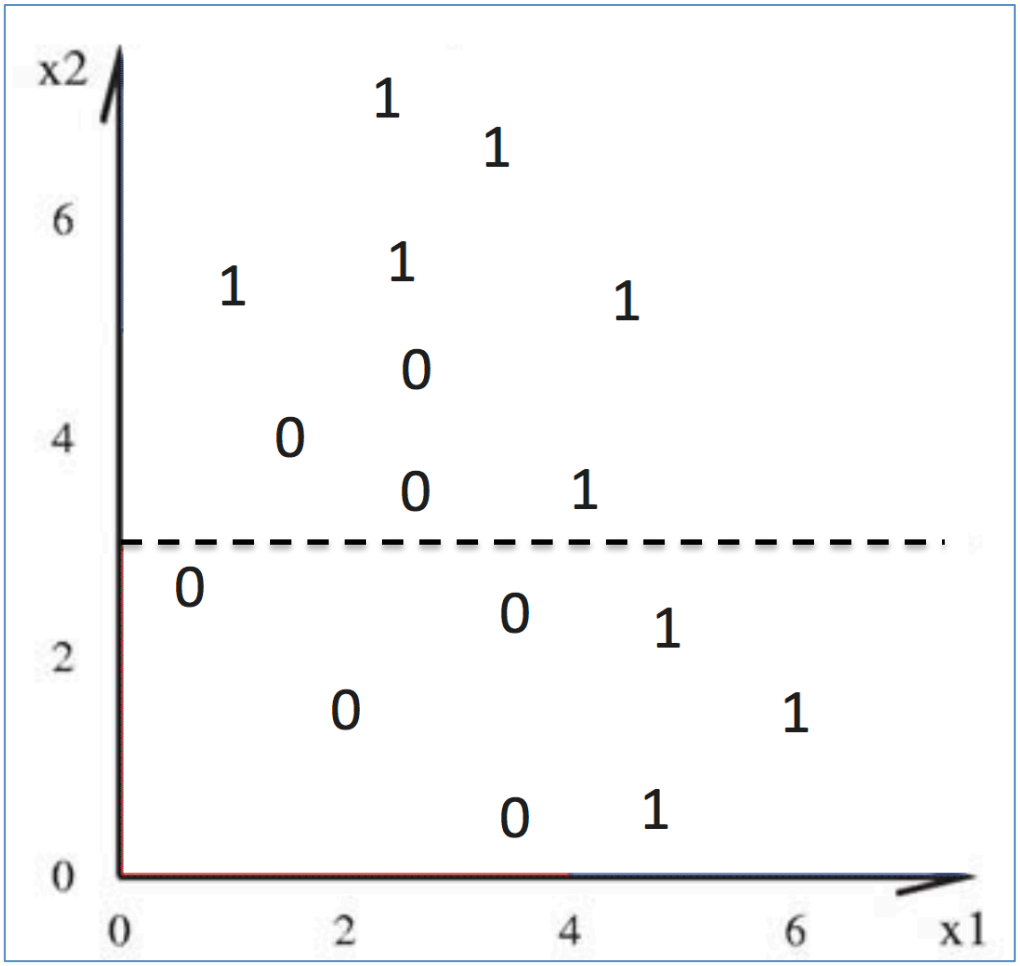
y_{pred} = No

Can also consider continuous features



Example by: Eric Eaton

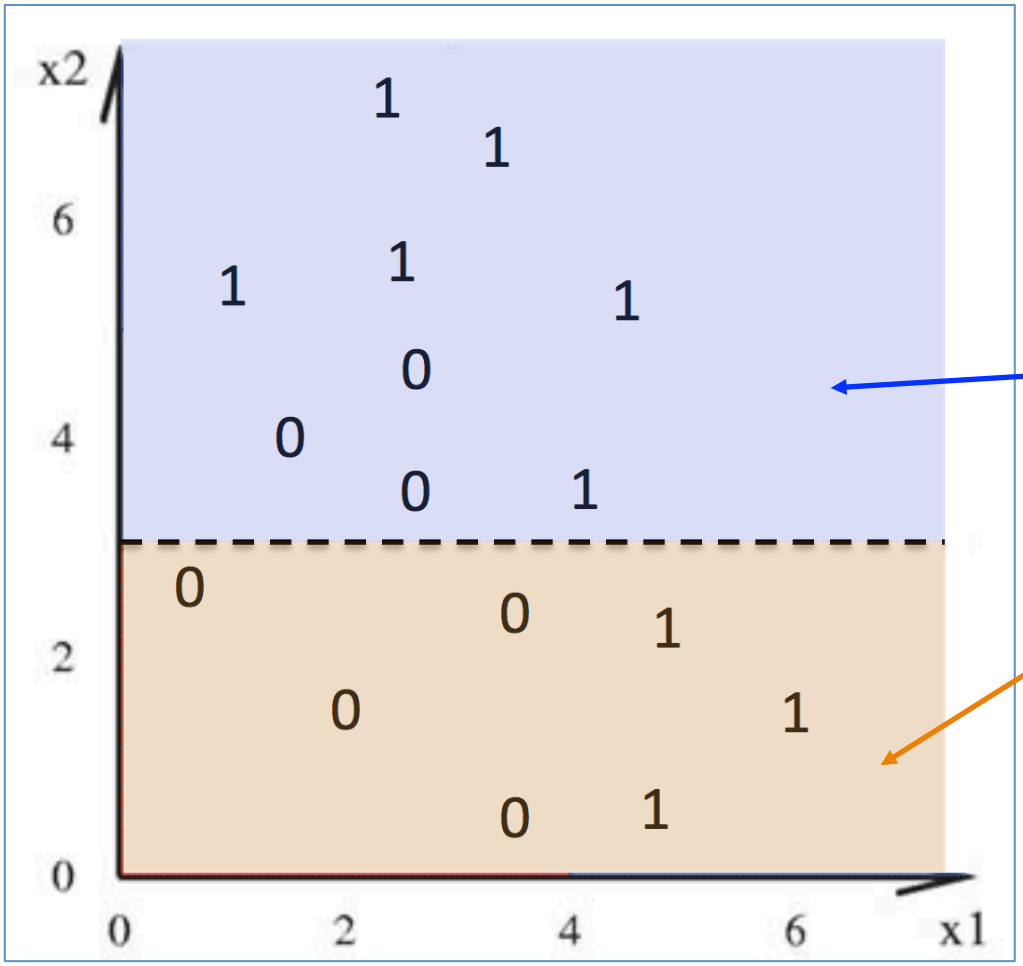
Can also consider continuous features

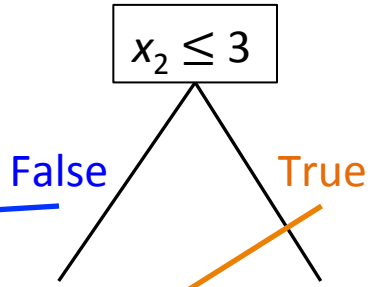


$x_2 \leq 3$

Example by: Eric Eaton

Can also consider continuous features





Correction!

Example by: Eric Eaton

Decision Tree pros/cons

- Very interpretable! Easy to say *why* we made a classification (can point to which features)
- Compact representation and fast predictions

Decision Tree pros/cons

- Very interpretable! Easy to say *why* we made a classification (can point to which features)
- Compact representation and fast predictions

- Can be brittle (not looking at each example holistically)
- Featurization and implementation difficulties

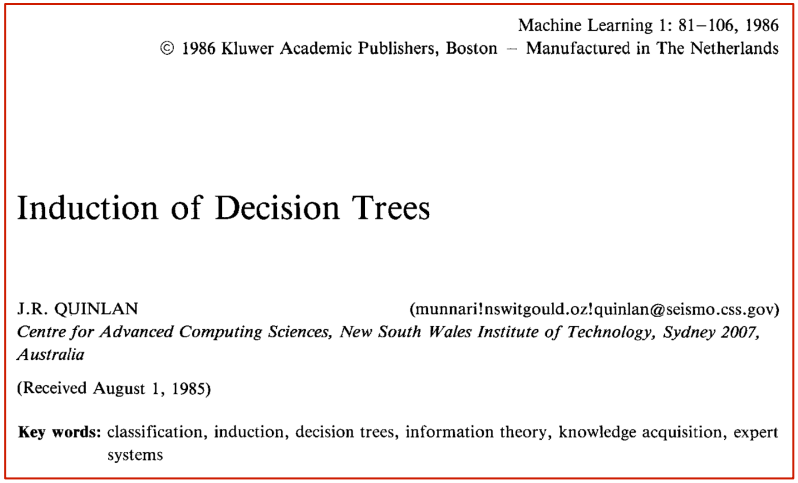
ID3 Decision Tree algorithm (1986)

- Select feature that "best" informs label prediction (i.e. *y*)

- **Divide**: partition data into branches based on their value at this feature

- **Conquer**: recurse on each partition

Posted as optional reading



Machine Learning 1: 81–106, 1986

Induction of Decision Trees

J.R. QUINLAN (munnari!nswitgould.oz!quinlan@seismo.css.gov)
Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney 2007, Australia

(Received August 1, 1985)

Key words: classification, induction, decision trees, information theory, knowledge acquisition, expert systems

Top-Down decision tree algorithm

Dataset (X,y)

Features

```
MakeSubtree(D, F)
  if stopping criteria met
      make a leaf node N
      determine class label/probabilities for N
  else
      make an internal node N
      S = FindBestFeature(D, F)
      for each outcome k of S
          Dk = subset of instances that have outcome k
          N.child[k] = MakeSubtree(Dk,F-S)
  return subtree rooted at N
```

For us: use majority label (break ties arbitrarily)

Why don't we want to use this feature again?

Slide: modified from Ameet Soni

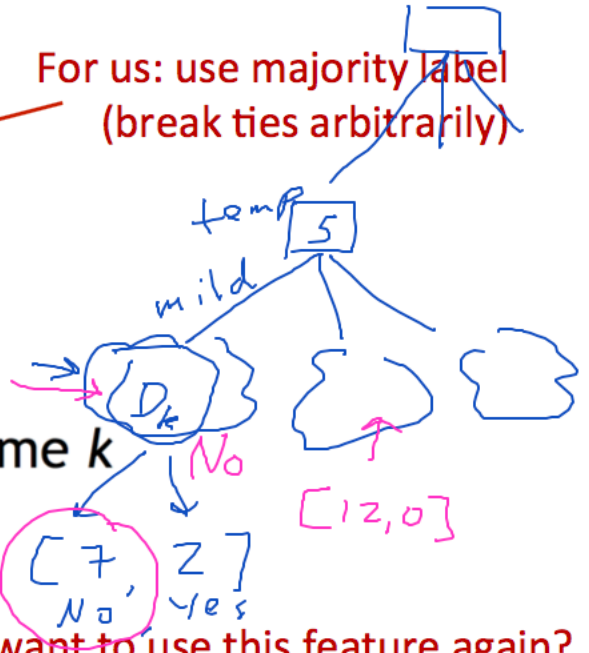
Top-Down decision tree algorithm

Dataset (X,y)

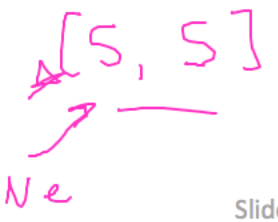
Features

```
MakeSubtree(D, F)
  if stopping criteria met
     make a leaf node N
     determine class label/probabilities for N
  else
     make an internal node N
     S = FindBestFeature(D, F)
     for each outcome k of S
          Dk = subset of instances that have outcome k
          N.child[k] = MakeSubtree(Dk,F-S)
  return subtree rooted at N
```

For us: use majority label (break ties arbitrarily)



Why don't we want to use this feature again?



Slide: modified from Ameet Soni

Handout 2 (on Piazza)

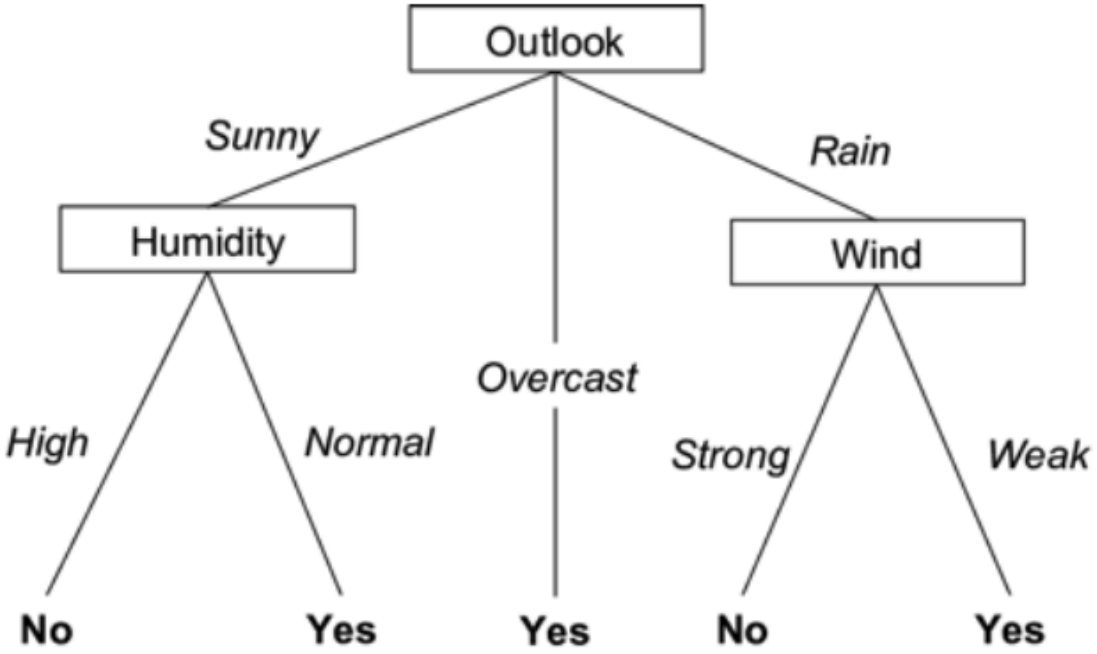
Handout 2

1. First, what is n (number of data points)? What is p (number of features)? Given the training data and decision tree shown below, what is the classification error on this data?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis (y)
$\boldsymbol{x}_1$	Sunny	Hot	High	Weak	No
$\boldsymbol{x}_2$	Sunny	Hot	High	Strong	No
$\boldsymbol{x}_3$	Overcast	Hot	High	Weak	Yes
$\boldsymbol{x}_4$	Rain	Mild	High	Weak	Yes
$\boldsymbol{x}_5$	Rain	Cool	Normal	Weak	Yes
$\boldsymbol{x}_6$	Rain	Cool	Normal	Strong	No
$\boldsymbol{x}_7$	Overcast	Cool	Normal	Strong	Yes
$\boldsymbol{x}_8$	Sunny	Mild	High	Weak	No
$\boldsymbol{x}_9$	Sunny	Cool	Normal	Weak	Yes
$\boldsymbol{x}_{10}$	Rain	Mild	Normal	Weak	Yes
$\boldsymbol{x}_{11}$	Sunny	Mild	Normal	Strong	Yes
$\boldsymbol{x}_{12}$	Overcast	Mild	High	Strong	Yes
$\boldsymbol{x}_{13}$	Overcast	Hot	Normal	Weak	Yes
$\boldsymbol{x}_{14}$	Rain	Mild	High	Strong	No

Handout 2

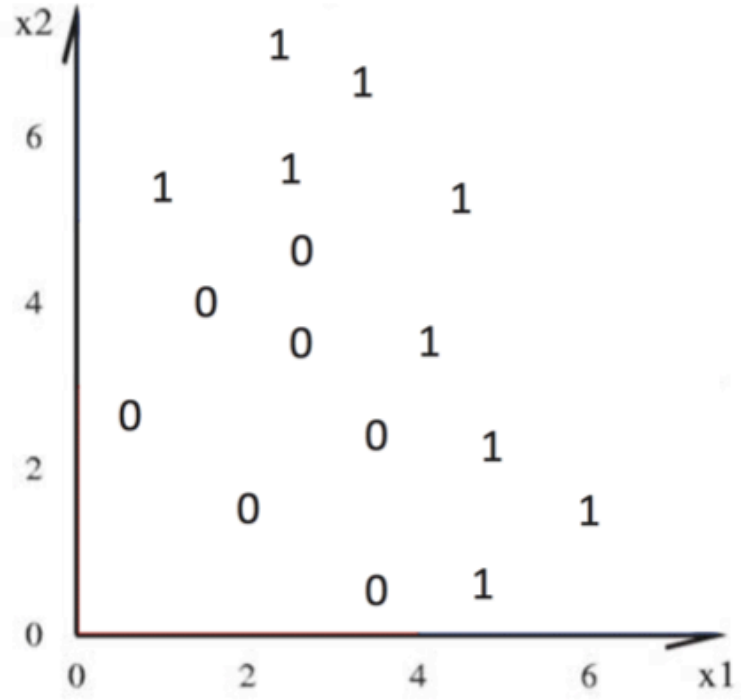
2. On the tree below, the children of each node divide the training data into partitions. Label each node (both internal nodes and leaves) with the counts of "No" and "Yes" labels based on the partition. For example, the counts for the node labeled *Outlook* would be [5, 9].



3. What if we had restricted the tree's *depth* to be 1? What would the tree look like and what would be the classification error?

Handout 2

4. For the dataset below, the label $y \in \{0, 1\}$. What is n? What is p? Devise a decision tree for this data that perfectly classifies the given examples. Internal node labels should be of the form "$x_j \leq a$", where a is some constant.



5. Repeat Question (2) for this decision tree (i.e. label each node with the "0" and "1" counts.)

Design choice: stopping criteria

1. All the data points in our partition have the same label

2. No more features remain to split on

3. No features are informative about the label

4. Reached (user specified) max depth in the tree

Avoiding overfitting for us

- Stop when leaf label reaches a certain fraction (i.e. 95% “yes”, 5% “no”)

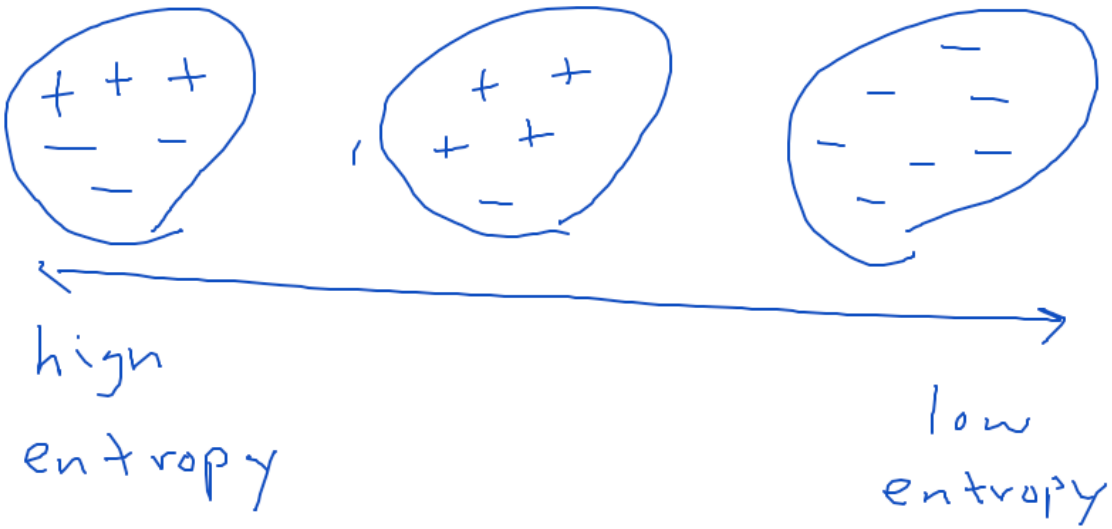
For our Lab 2 implementation

- Set a maximum depth for the tree

- Set a minimum number of examples in leaf (i.e. if we have a 2-1 split, stop)

Outline for September 15

- Finish KNN
- Recap featurization
- Overfitting
- Decision Trees
- Entropy



How to select “best” features?

X			Y
Color	**Shape**	**Size**	**Likes toy?**
red	square	big	+
blue	square	big	+
red	circle	small	-
blue	square	small	-
red	circle	big	+

Example by: Ameet Soni

How to select “best” features?

X			Y
Color	**Shape**	**Size**	**Likes toy?**
red	square	big	+
blue	square	big	+
red	circle	**big**	-
blue	square	**big**	-
red	circle	big	+

Example by: Ameet Soni

Entropy

Idea: avg # bits needed to transmit info

Year	prob (p)	Idea	Cumulative prob	Binary		
Senior	0.5					
Junior	0.25					
Sophomore	0.125					
First year	0.125					

Next time!