Machine Learning Department
School of Computer Science
Carnegie Mellon University

Overfitting

+

k-Nearest Neighbors

Matt Gormley
Lecture 4
Jan. 31, 2022

Course Staff


Zachary Novack


Sami Kale


Rita Zhang


Brendon Gu


Mukund Subramaniam


Yuxin Guo


Junhui Li


Abuzar Khan


Shubham Phal


Tara Lakdawala


Matt Gormley


Sana Lakdawala

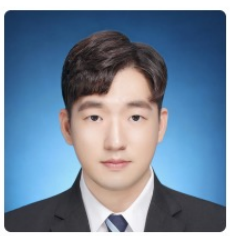

Tori Ciplickas

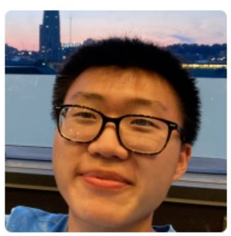

Prasoon Varshney


Alex Xie


Abhishek Vijayakumar


Abbey Pei

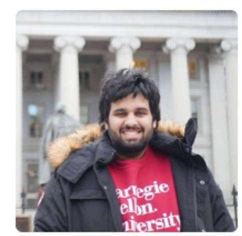

Hayden Kim


Chu Weng


Brynn Edmunds


Joshmin Ray


Shelly Ren


Udai Singh

Course Staff


Zachary Novack


Sami Kale


Rita Zhang


Brendon Gu


Mukund Subramaniam


Yuxin Guo


Junhui Li


Abuzar Khan


Shubham Phal


Tara Lakdawala


Matt Gormley


Sana Lakdawala

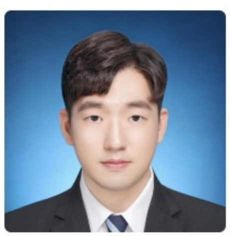

Tori Ciplickas

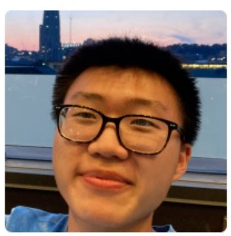

Prasoon Varshney

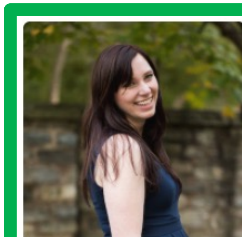

Alex Xie


Abhishek Vijayakumar


Abbey Pei

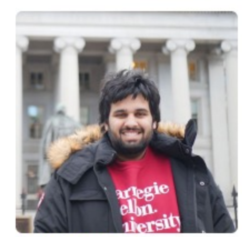

Hayden Kim


Chu Weng


Brynn Edmunds


Joshmin Ray


Shelly Ren


Udai Singh

EAs

Course Staff

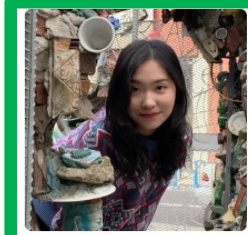

Zachary Novack

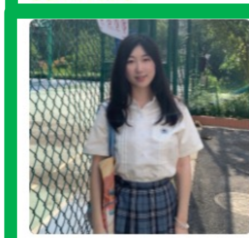

Sami Kale

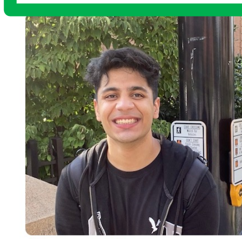

Rita Zhang


Brendon Gu


Mukund Subramaniam


Yuxin Guo

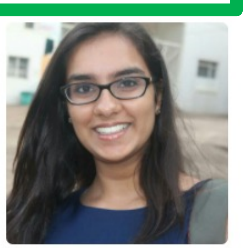

Junhui Li

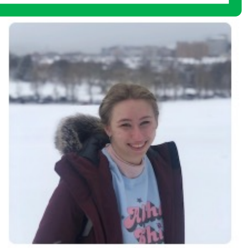

Abuzar Khan


Shubham Phal


Tara Lakdawala


Matt Gormley


Sana Lakdawala

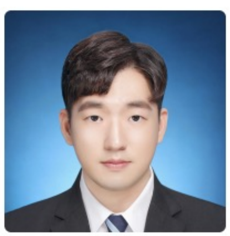

Tori Ciplickas

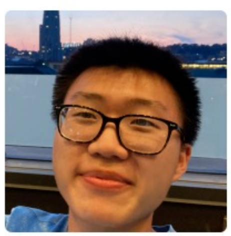

Prasoon Varshney


Alex Xie


Abhishek Vijayakumar


Abbey Pei

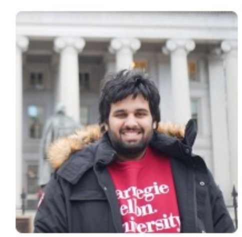

Hayden Kim


Chu Weng


Brynn Edmunds


Joshmin Ray


Shelly Ren


Udai Singh

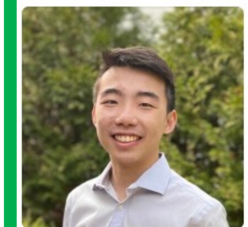
Team A

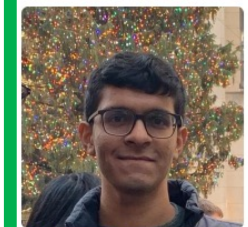
Course Staff


Zachary Novack


Sami Kale

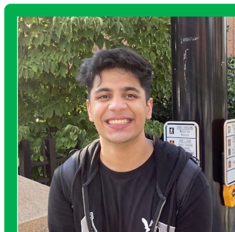

Rita Zhang


Brendon Gu


Mukund Subramaniam


Yuxin Guo


Junhui Li


Abuzar Khan

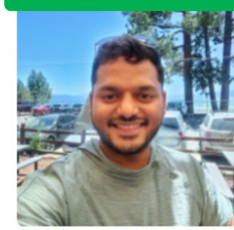

Shubham Phal

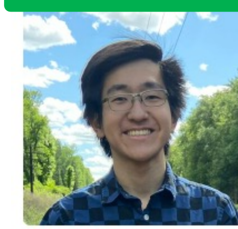

Tara Lakdawala

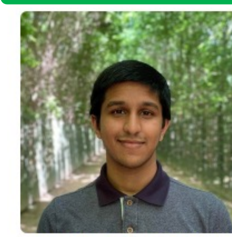

Matt Gormley


Sana Lakdawala

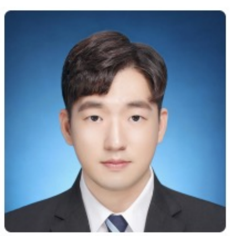

Tori Ciplickas

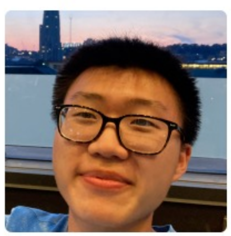

Prasoon Varshney


Alex Xie


Abhishek Vijayakumar


Abbey Pei

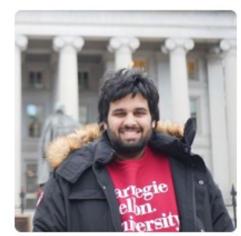

Hayden Kim


Chu Weng


Brynn Edmunds


Joshmin Ray


Shelly Ren


Udai Singh

Team B

Course Staff


Zachary Novack


Sami Kale


Rita Zhang


Brendon Gu


Mukund Subramaniam


Yuxin Guo

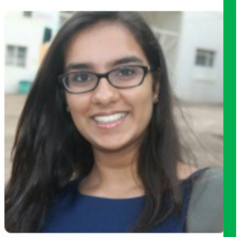

Junhui Li

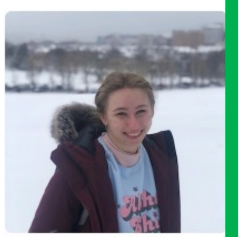

Abuzar Khan

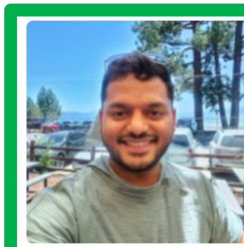

Shubham Phal


Tara Lakdawala


Matt Gormley


Sana Lakdawala

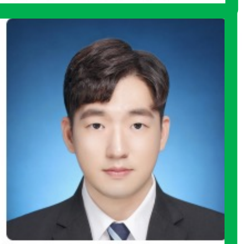

Tori Ciplickas

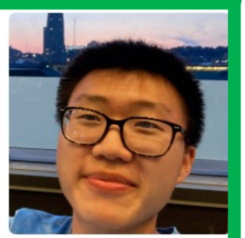

Prasoon Varshney

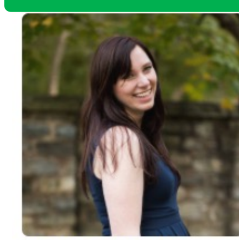

Alex Xie


Abhishek Vijayakumar


Abbey Pei

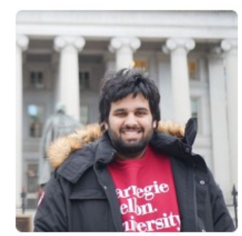

Hayden Kim


Chu Weng


Brynn Edmunds


Joshmin Ray


Shelly Ren


Udai Singh

Team C

Course Staff


Zachary Novack


Sami Kale


Rita Zhang


Brendon Gu


Mukund Subramaniam


Yuxin Guo


Junhui Li


Abuzar Khan


Shubham Phal

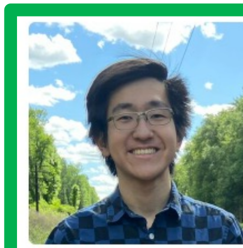

Tara Lakdawala


Matt Gormley

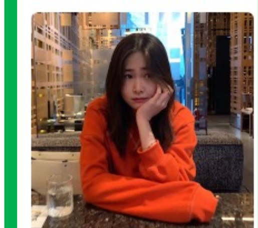

Sana Lakdawala

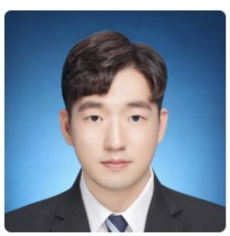

Tori Ciplickas

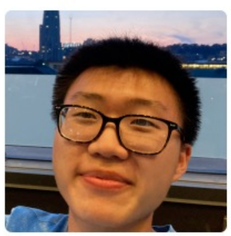

Prasoon Varshney


Alex Xie


Abhishek Vijayakumar

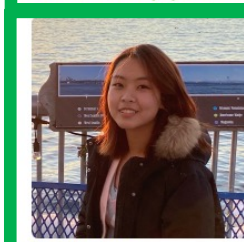

Abbey Pei

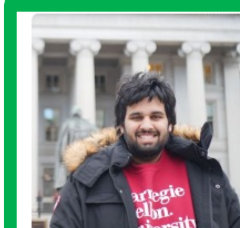

Hayden Kim


Chu Weng


Brynn Edmunds


Joshmin Ray


Shelly Ren


Udai Singh

Team D

Q&A

Q: Why don't my entropy calculations match those on the slides?

A: Remember that H(Y) is conventionally reported in "bits" and computed using log base 2.
e.g., $H(Y) = - P(Y=0) \log_2 P(Y=0) - P(Y=1) \log_2 P(Y=1)$

Q: When and how do we decide to stop growing trees? What if the set of values an attribute could take was really large or even infinite?

A: We'll address this question for discrete attributes today. If an attribute is real-valued, there's a clever trick that only considers O(L) splits where L = # of values the attribute takes in the training set. Can you guess what it does?

Q&A

Q: What does decision tree training do if a branch receives no data?

A: Then we hit the base case and create a leaf node. So the real question is what does majority vote do when there is no data? Of course, there is no majority label, so (if forced to) we could just return one randomly.

Q: What do we do at test time when we observe a value for a feature that we didn't see at training time.

A: This really just a variant of the first question. That said, a real DT implementation needs to elegantly handle this case. We could do so by either (a) assuming that all possible values will be seen at train time, so there should be a branch for all attributes even if the partition of the dataset doesn't include them all or (b) recognize the unseen value at test time and return some appropriate label in that case.

Reminders

- Exit Poll: HW1 (required for participation)
- Homework 2: Decision Trees
  - Out: Wed, Jan. 26
  - Due: Fri, Feb. 4 at 11:59pm

EMPIRICAL COMPARISON OF SPLITTING CRITERIA

Experiments: Splitting Criteria

Bluntine & Niblett (1992) compared 4 criteria (random, Gini, mutual information, Marshall) on 12 datasets

Medical Diagnosis Datasets: (4 of 12)

- **hypo**: data set of 3772 examples records expert opinion on possible hypo- thyroid conditions from 29 real and discrete attributes of the patient such as sex, age, taking of relevant drugs, and hormone readings taken from drug samples.
- **breast:** The classes are reoccurrence or non-reoccurrence of breast cancer sometime after an operation. There are nine attributes giving details about the original cancer nodes, position on the breast, and age, with multi-valued discrete and real values.
- **tumor:** examples of the location of a primary tumor
- **lymph**: from the lymphography domain in oncology. The classes are normal, metastases, malignant, and fibrosis, and there are nineteen attributes giving details about the lymphatics and lymph nodes

Table 1. Properties of the data sets

Data Set	Classes	Attr.s	Training Set	Test Set
hypo	4	29	1000	2772
breast	2	9	200	86
tumor	22	18	237	102
lymph	4	18	103	45
LED	10	7	200	1800
mush	2	22	200	7924
votes	2	17	200	235
votesl	2	16	200	235
iris	3	4	100	50
glass	7	9	100	114
xd6	2	10	200	400
pole	2	4	200	1647

Table from Bluntine & Niblett (1992)

Experiments: Splitting Criteria

Table 3. Error for different splitting rules (pruned trees).

Data Set	Splitting Rule: GINI	Info. Gain	Marsh.	Random
hypo	1.01 ± 0.29	0.95 ± 0.22	1.27 ± 0.47	7.44 ± 0.53
breast	28.66 ± 3.87	28.49 ± 4.28	27.15 ± 4.22	29.65 ± 4.97
tumor	60.88 ± 5.44	62.70 ± 3.89	61.62 ± 3.98	67.94 ± 5.68
lymph	24.44 ± 6.92	24.00 ± 6.87	24.33 ± 5.51	32.33 ± 11.25
LED	33.77 ± 3.06	32.89 ± 2.59	33.15 ± 4.02	38.18 ± 4.57
mush	1.44 ± 0.47	1.44 ± 0.47	7.31 ± 2.25	8.77 ± 4.65
votes	4.47 ± 0.95	4.57 ± 0.87	11.77 ± 3.95	12.40 ± 4.56
votes1	12.79 ± 1.48	13.04 ± 1.65	15.13 ± 2.89	15.62 ± 2.73
iris	5.00 ± 3.08	4.90 ± 3.08	5.50 ± 2.59	14.20 ± 6.77
glass	39.56 ± 6.20	50.57 ± 6.73	40.53 ± 6.41	53.20 ± 5.01
xd6	22.14 ± 3.23	22.17 ± 3.36	22.06 ± 3.37	31.86 ± 3.62
pole	15.43 ± 1.51	15.47 ± 0.88	15.01 ± 1.15	26.38 ± 6.92

Key Takeaway: GINI gain and Mutual Information are statistically indistinguishable!

Info. Gain is another name for *mutual information*

Table from Bluntine & Niblett (1992)

Experiments: Splitting Criteria

Table 4. Difference and significance of error for GINI splitting rule versus others.

Data Set	Splitting Rule: Info. Gain	Marsh.	Random
hypo	−0.06 (0.82)	0.26 (0.99)	6.43 (1.00)
breast	−0.17 (0.23)	−1.51 (0.94)	0.99 (0.72)
tumor	1.81 (0.84)	0.74 (0.39)	7.06 (0.99)
lymph	−0.44 (0.83)	0.11 (0.05)	7.89 (0.99)
LED	0.12 (0.17)	[illegible]	[illegible]
mush	0.00 (0.00)	5.86 [illegible]	[illegible]
votes	[illegible]	[illegible]	[illegible]
votes1	[illegible]	[illegible]	[illegible]
iris	[illegible]	[illegible]	[illegible]
glass	[illegible]	[illegible]	[illegible]
xd6	[illegible]	[illegible]	[illegible]
pole	[illegible]	[illegible]	[illegible]

Key Takeaway: GINI gain and Mutual Information are statistically indistinguishable!

Results are of the form A.AA (B.BB) where:

1. A.AA is the **average difference in errors** between the two methods
2. B.BB is the **significance** of the difference according to a two-tailed **paired t-test**

Table from Bluntine & Niblett (1992)

INDUCTIVE BIAS
(FOR DECISION TREES)

Decision Tree Learning Example

Dataset:

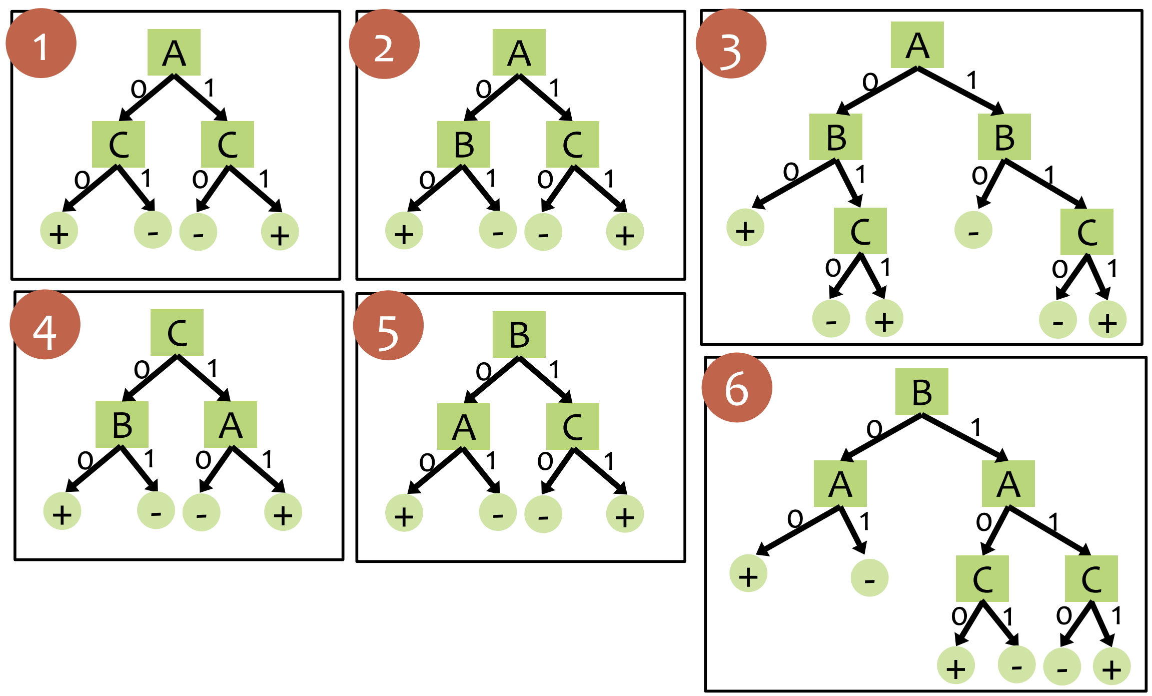
Output Y, Attributes A, B, C

Y	A	B	C
+	0	0	0
+	0	0	1
-	0	1	0
+	0	1	1
-	1	0	0
-	1	0	1
-	1	1	0
+	1	1	1

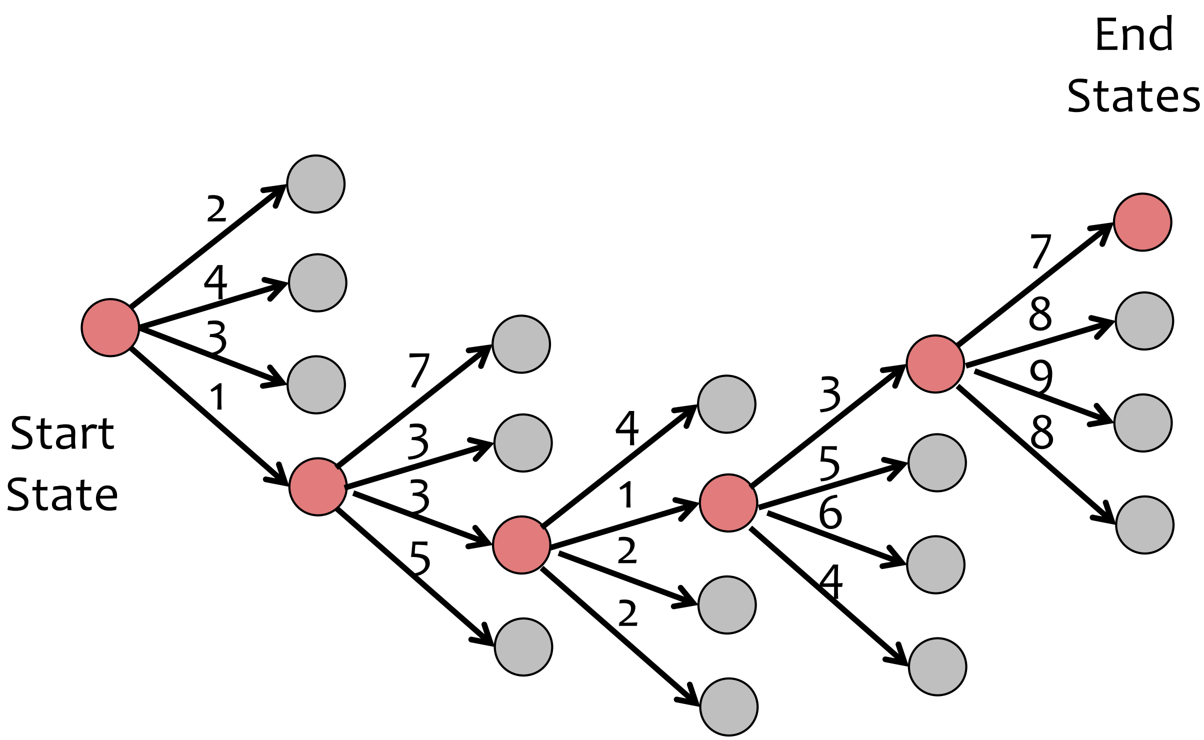
In-Class Exercise

Which of the following trees would be **learned by the the decision tree learning algorithm** using "error rate" as the splitting criterion?

(Assume ties are broken alphabetically.)



Background: Greedy Search



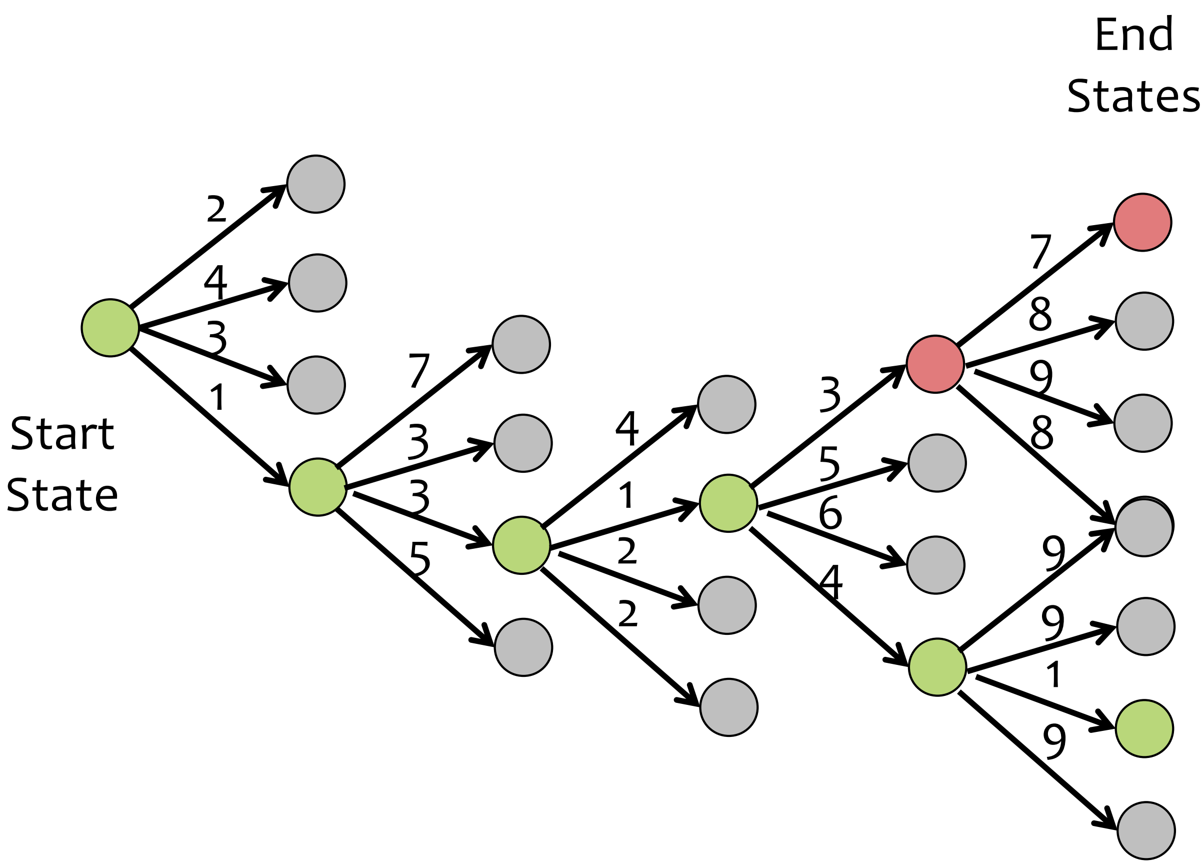
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

Background: Greedy Search



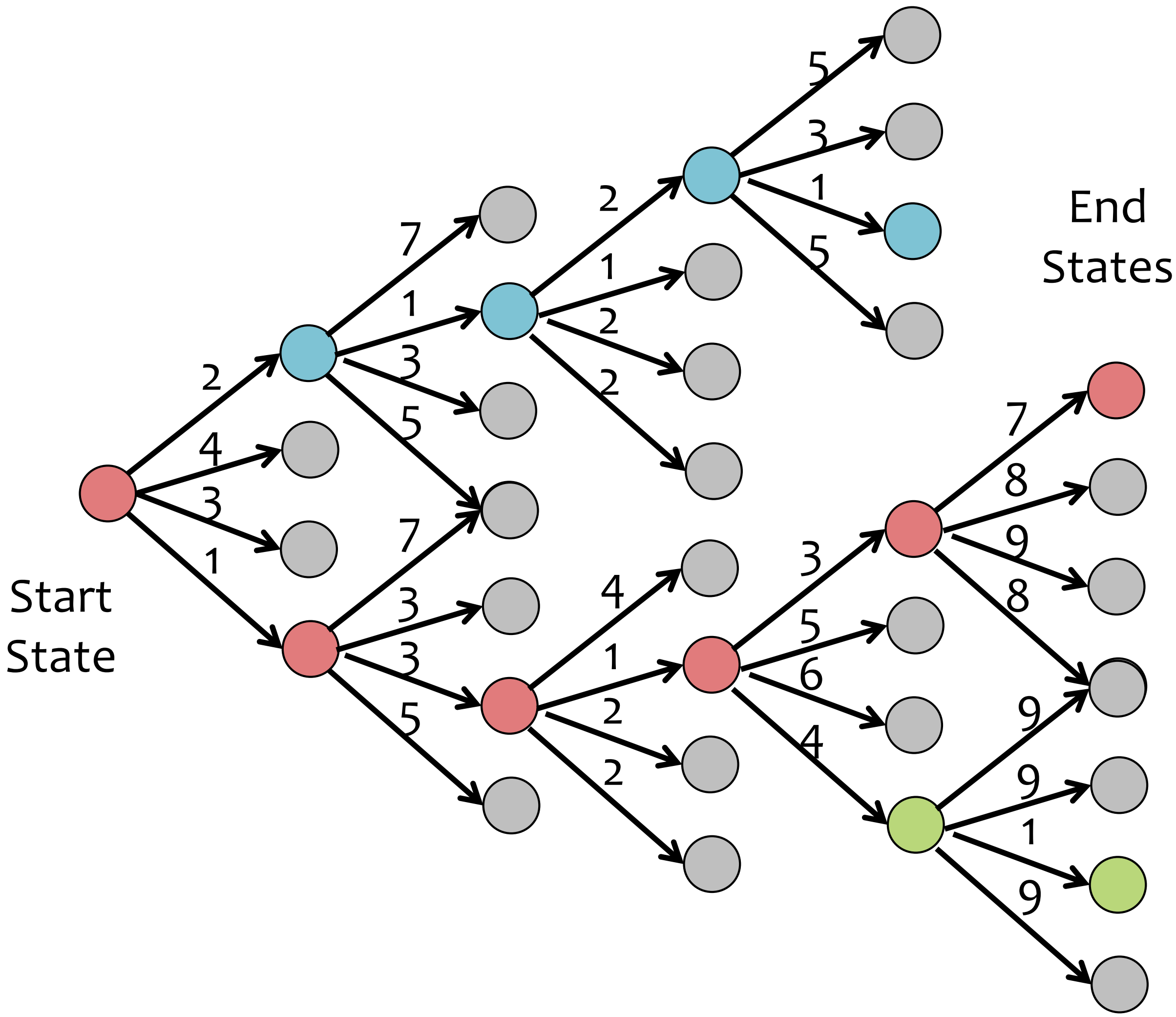
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

Background: Greedy Search



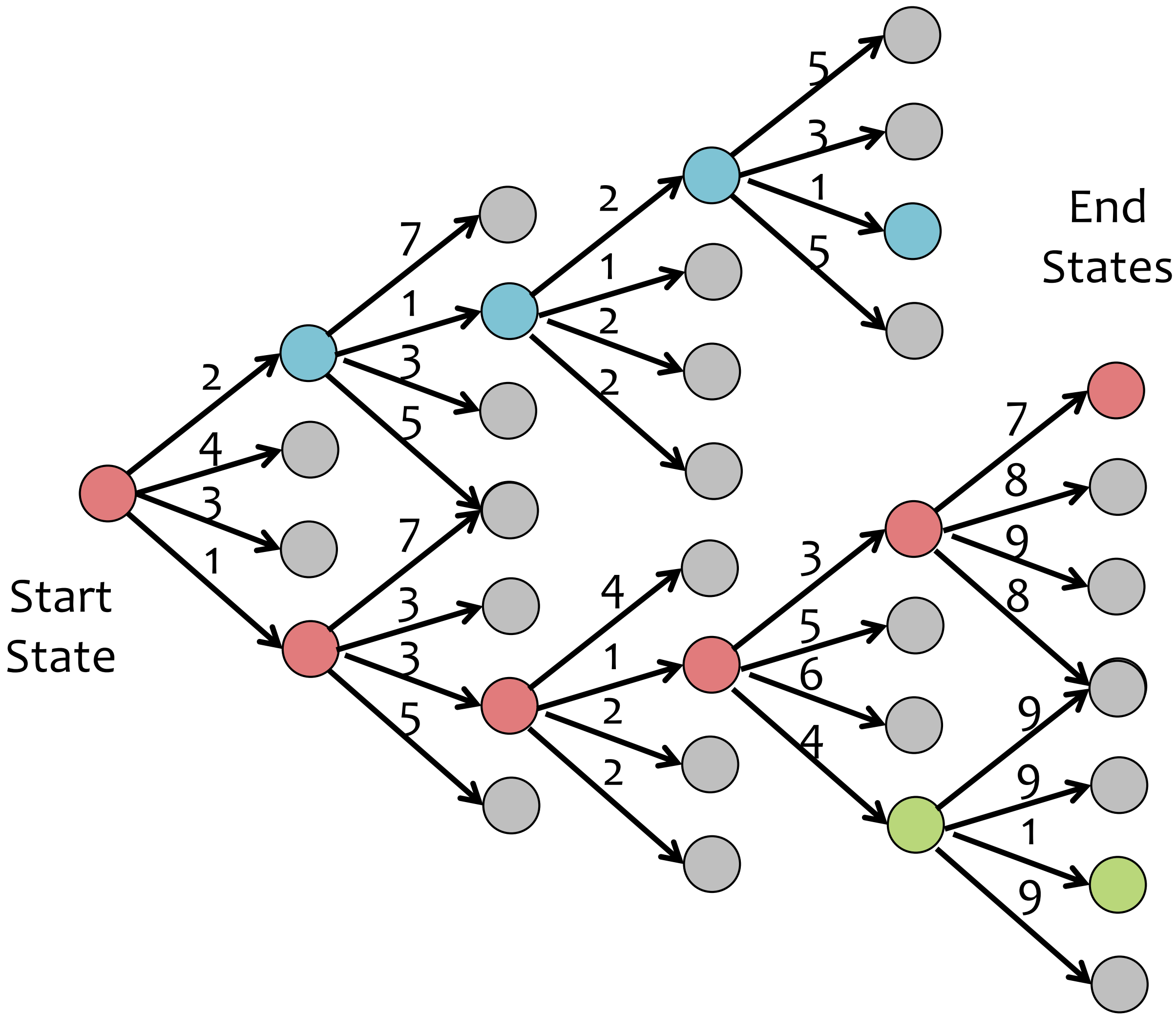
Goal:

- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Greedy Search:

- At each node, selects the edge with lowest (immediate) weight
- **Heuristic** method of search (i.e. does *not* necessarily find the best path)
- Computation time: **linear** in max path length

Background: Global Search



Goal:

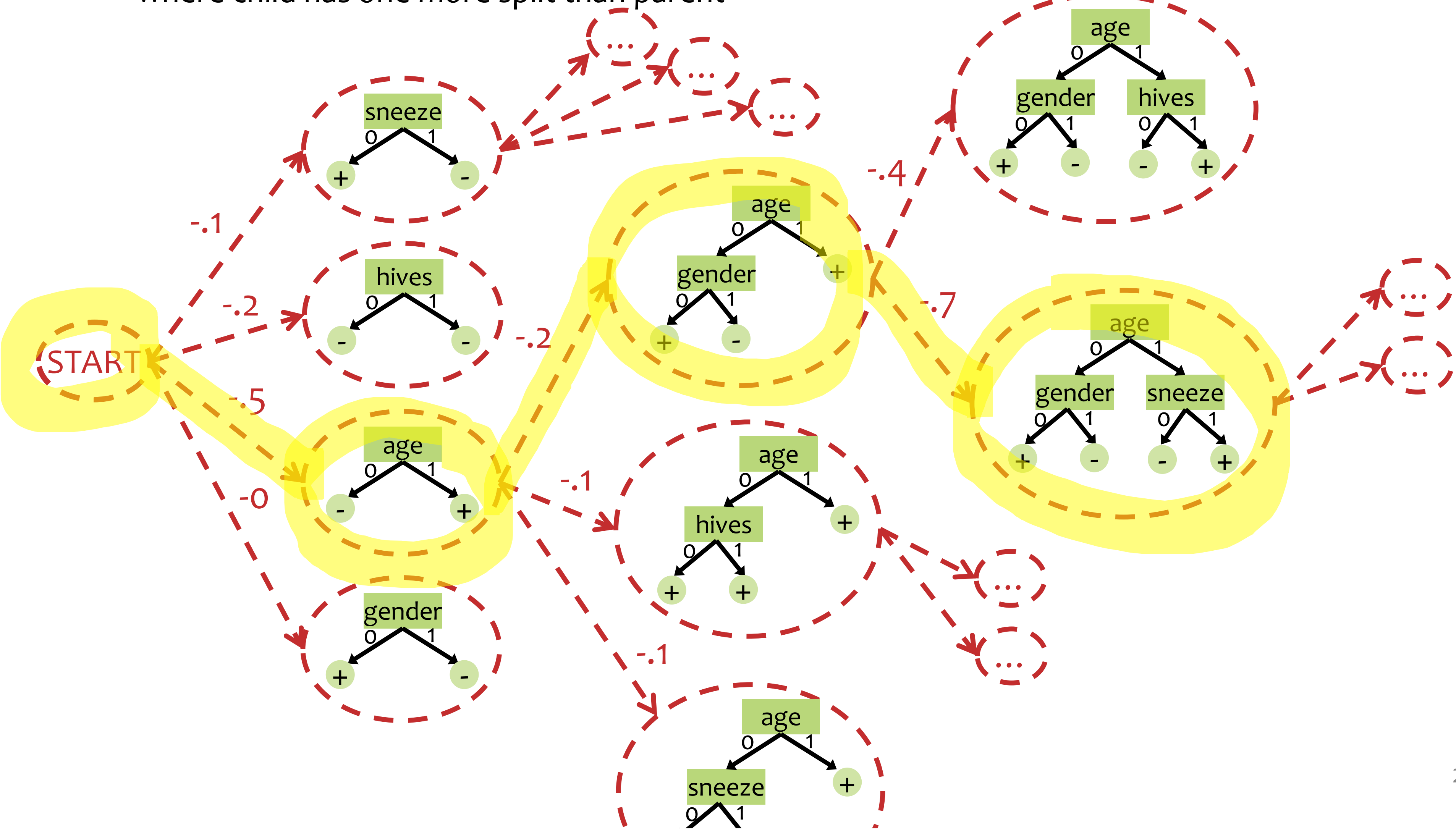
- Search space consists of nodes and weighted edges
- Goal is to find the lowest (total) weight path from root to a leaf

Global Search:

- Compute the weight of the path to **every** leaf
- **Exact** method of search (i.e. gauranteed to find the best path)
- Computation time: **exponential** in max path length

Decision Tree Learning as Search

1. **search space:** all possible decision trees
2. **node:** single decision tree
3. **edge:** connects one full tree to another, where child has one more split than parent
4. **edge weight:** (negative) splitting criterion
5. **DT learning:** greedy search, maximizing our splitting criterion at each step



Big Question:

How is it that your ML algorithm can generalize to unseen examples?

DT: Remarks

ID3 = Decision Tree Learning with Mutual Information as the splitting criterion

Question: Which tree does ID3 find?

DT: Remarks

ID3 = Decision Tree Learning with Mutual Information as the splitting criterion

Question: Which tree does ID3 find?

Definition:

We say that the **inductive bias** of a machine learning algorithm is the principal by which it generalizes to unseen examples

Inductive Bias of ID3:

Smallest tree that matches the data with high mutual information attributes near the top

Occam's Razor: (restated for ML)

Prefer the simplest hypothesis that explains the data

Decision Tree Learning Example

Dataset:

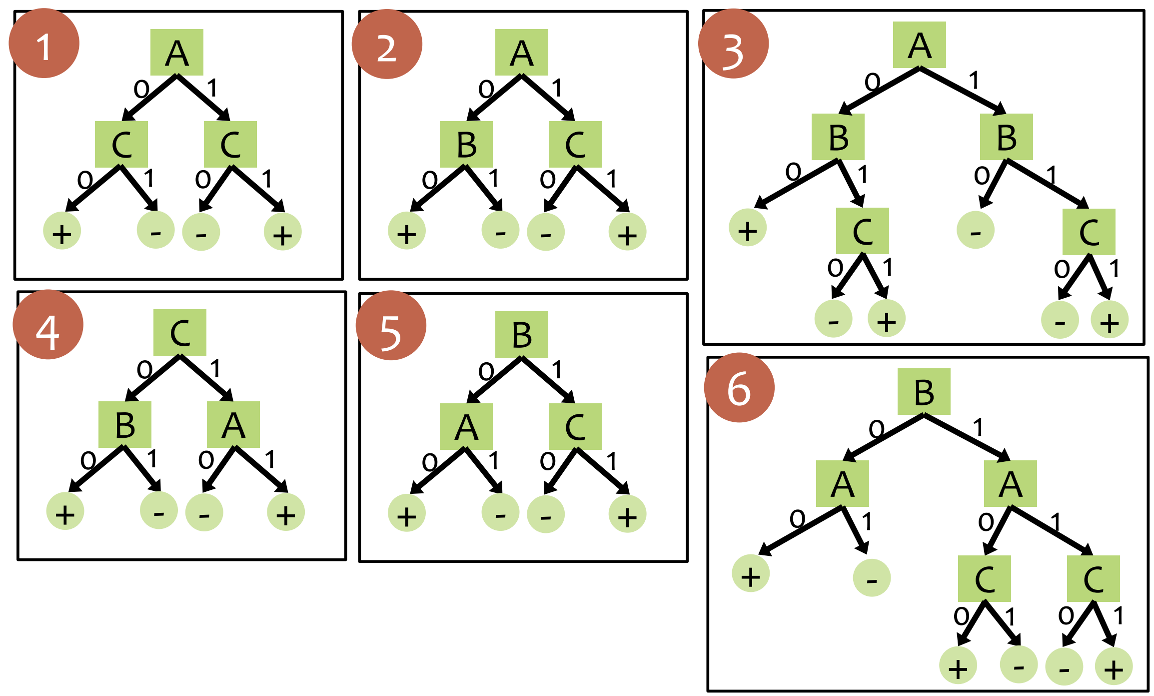
Output Y, Attributes A, B, C

Y	A	B	C
+	0	0	0
+	0	0	1
-	0	1	0
+	0	1	1
-	1	0	0
-	1	0	1
-	1	1	0
+	1	1	1

In-Class Exercise

Suppose you had an algorithm that found **the tree with lowest training error that was as small as possible (i.e. exhaustive global search)**, which tree would it return?

(Assume ties are broken by choosing the smallest.)



OVERFITTING
(FOR DECISION TREES)

Decision Tree Generalization

Question:

Which of the following would generalize best to unseen examples?

A. Small tree with low training accuracy
B. Large tree with low training accuracy
C. Small tree with high training accuracy
D. Large tree with high training accuracy

Answer:

Overfitting and Underfitting

Underfitting

- The model…
  - is too simple
  - is unable captures the trends in the data
  - exhibits too much bias
- *Example*: majority-vote classifier (i.e. depth-zero decision tree)
- *Example*: a toddler (that has **not** attended medical school) attempting to carry out medical diagnosis

Overfitting

- The model…
  - is too complex
  - is fitting the noise in the data or fitting "outliers"
  - does not have enough bias
- *Example*: our "memorizer" algorithm responding to an irrelevant attribute
- *Example*: medical student who simply memorizes patient case studies, but does not understand how to apply knowledge to new patients

Overfitting

- Given a hypothesis *h*, its…

  …error rate over all training data: $error(h, D_{train})$

  …error rate over all test data: $error(h, D_{test})$

  …true error over all data: $error_{true}(h)$

- We say h overfits the training data if…

$$error_{true}(h) > error(h, D_{train})$$

- Amount of overfitting =

$$error_{true}(h) – error(h, D_{train})$$

In practice, $error_{true}(h)$ is **unknown**

Slide adapted from Tom Mitchell

Overfitting in Decision Tree Learning

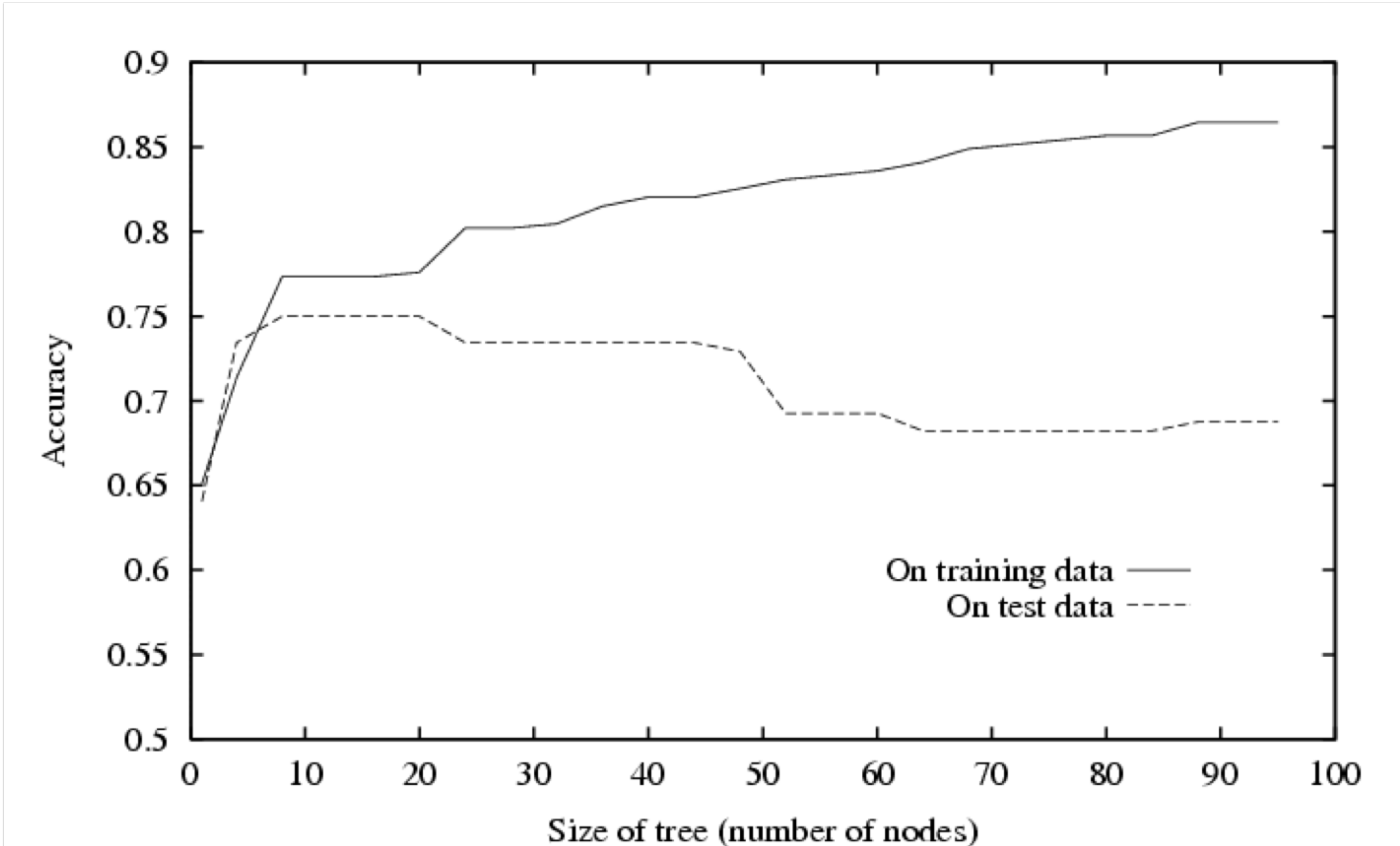


Figure from Tom Mitchell

How to Avoid Overfitting?

For Decision Trees…

1. Do not grow tree beyond some **maximum depth**
2. Do not split if splitting criterion (e.g. mutual information) is **below some threshold**
3. Stop growing when the split is **not statistically significant**
4. Grow the entire tree, then **prune**

Reduced-Error Pruning

Split data into *training* and *validation* set

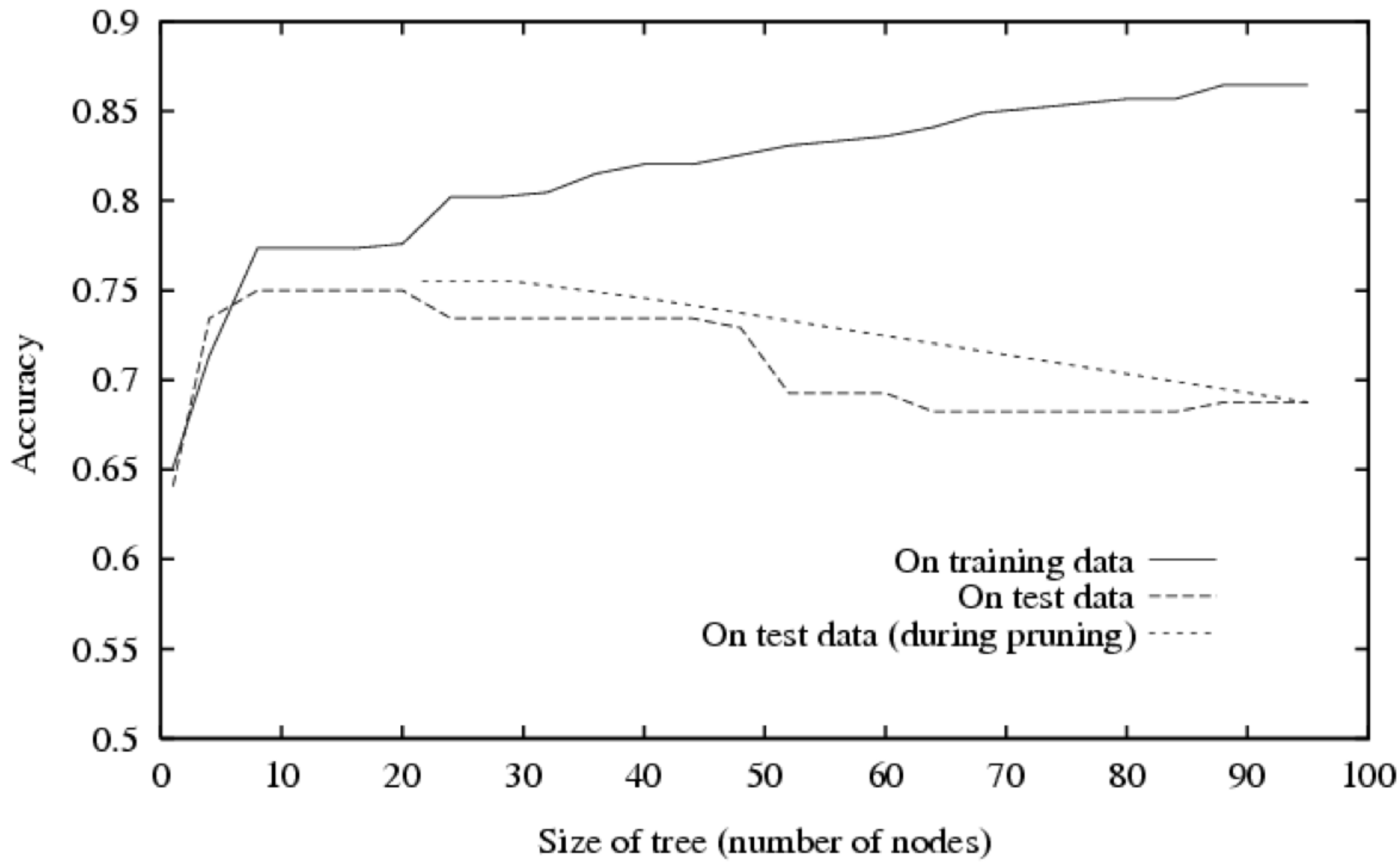
Create tree that classifies *training* set correctly

Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

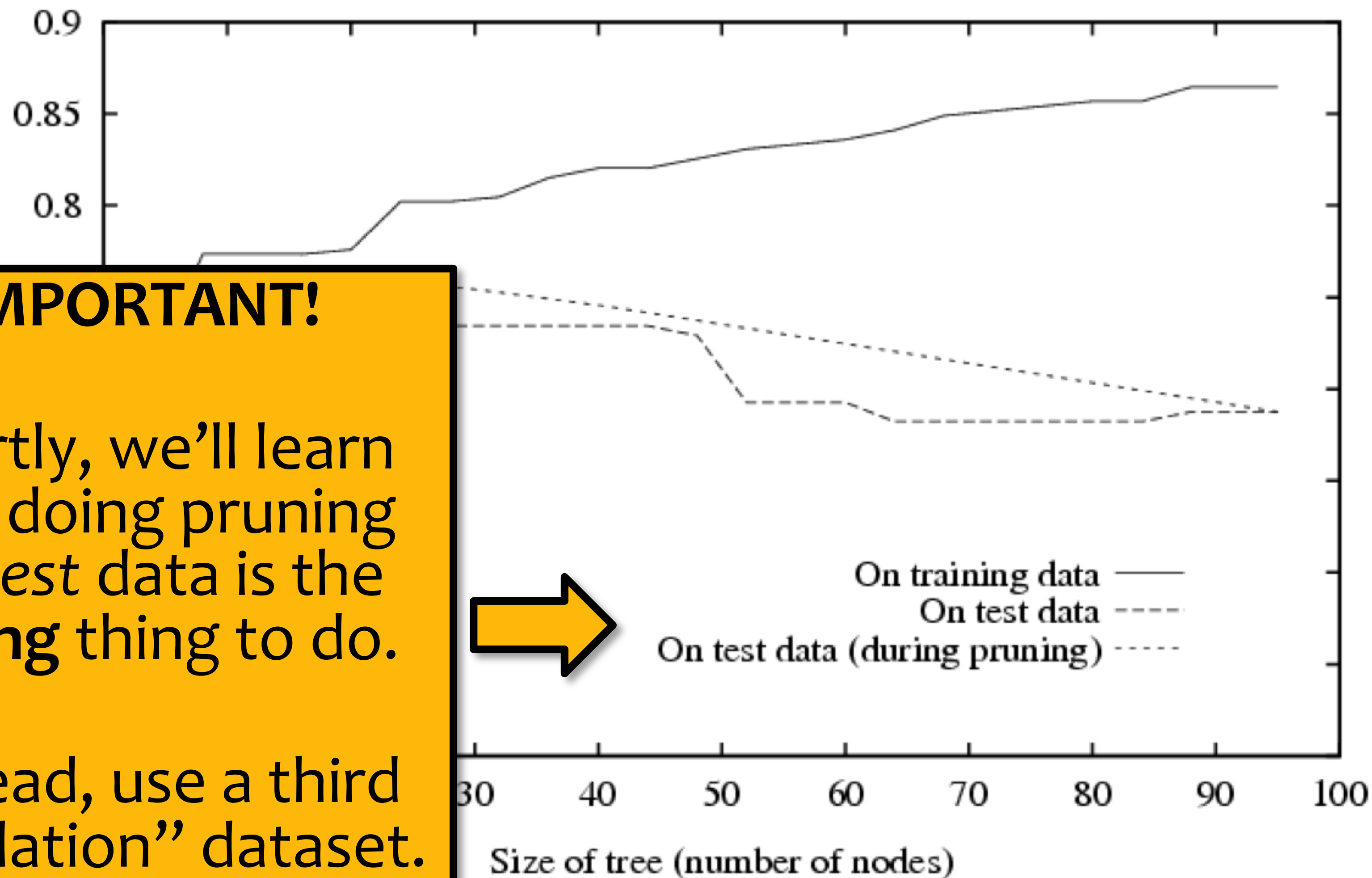
- produces smallest version of most accurate subtree
- What if data is limited?

Effect of Reduced-Error Pruning



Slide from Tom Mitchell

Effect of Reduced-Error Pruning



IMPORTANT!

Shortly, we'll learn that doing pruning on *test* data is the **wrong** thing to do.

Instead, use a third "validation" dataset.

Slide from Tom Mitchell

Decision Trees (DTs) in the Wild

- DTs are one of the most popular classification methods for practical applications
  - Reason #1: The learned representation is **easy to explain** a non-ML person
  - Reason #2: They are **efficient** in both computation and memory
- DTs can be applied to a wide variety of problems including **classification**, **regression**, **density estimation,** etc.
- **Applications of DTs** include…
  - medicine, molecular biology, text classification, manufacturing, astronomy, agriculture, and many others
- **Decision Forests** learn many DTs from random subsets of features; the result is a very powerful example of an **ensemble method** (discussed later in the course)

DT Learning Objectives

You should be able to…

1. Implement Decision Tree training and prediction
2. Use effective splitting criteria for Decision Trees and be able to define entropy, conditional entropy, and mutual information / information gain
3. Explain the difference between memorization and generalization [CIML]
4. Describe the inductive bias of a decision tree
5. Formalize a learning problem by identifying the input space, output space, hypothesis space, and target function
6. Explain the difference between true error and training error
7. Judge whether a decision tree is "underfitting" or "overfitting"
8. Implement a pruning or early stopping method to combat overfitting in Decision Tree learning

REAL VALUED ATTRIBUTES


sepal
petal

Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher Iris Dataset

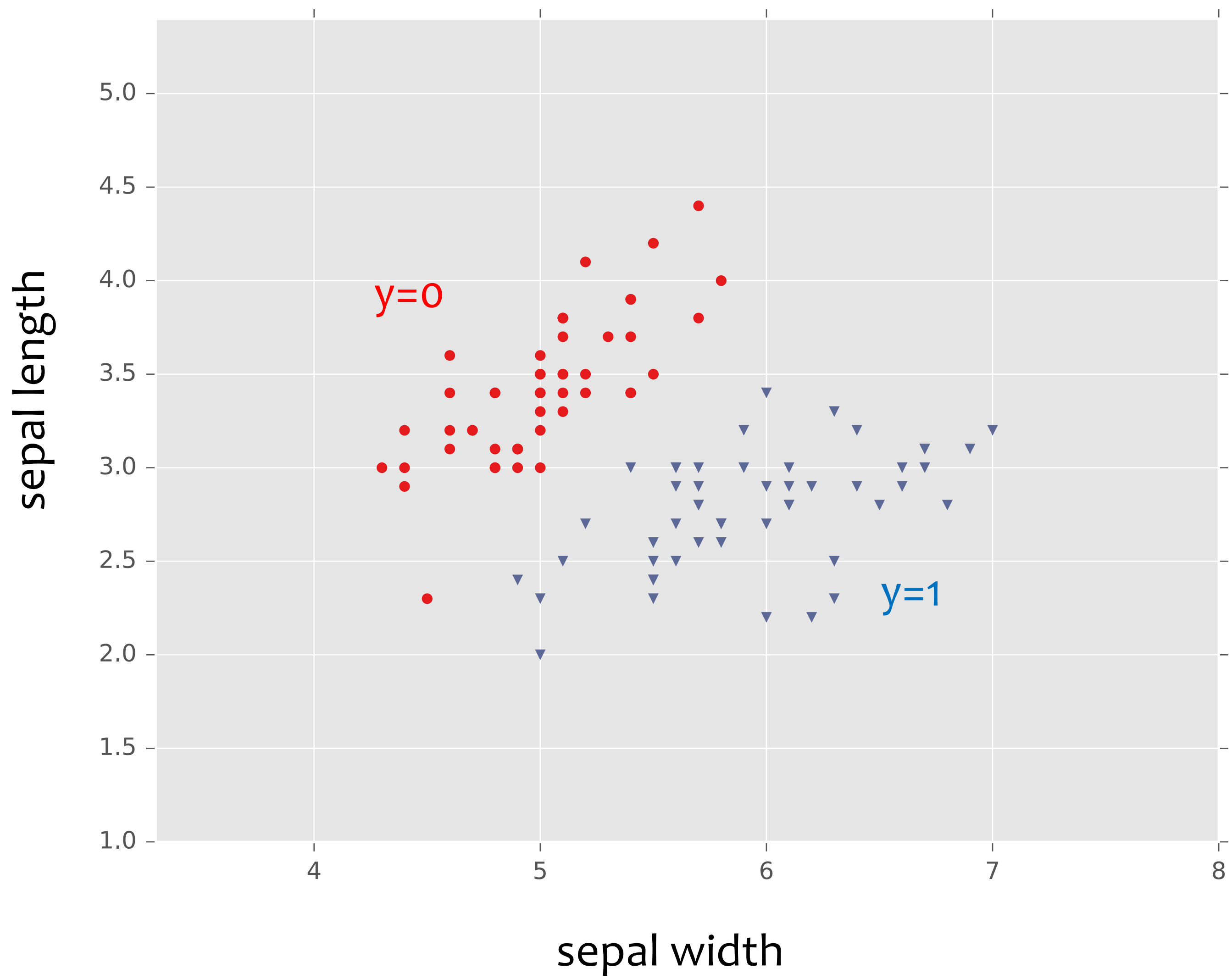
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width
0	4.3	3.0
0	4.9	3.6
0	5.3	3.7
1	4.9	2.4
1	5.7	2.8
1	6.3	3.3
1	6.7	3.0

Deleted two of the four features, so that input space is 2D

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher Iris Dataset



K-NEAREST NEIGHBORS

Classification & KNN

Whiteboard:

- Binary classification
- 2D examples
- Decision rules / hypotheses
- Nearest Neighbor and k-Nearest Neighbors classifiers
- KNN for binary classification