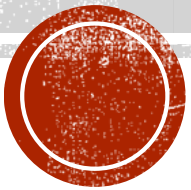
LIMITS OF LEARNING



CS 412 Introduction to Machine Learning

Prof. Zheleva

January 25, 2018

Reading Assignment: CIML: 2, ISL: 2.1-2.2.2

LAST LECTURE: DECISION TREES

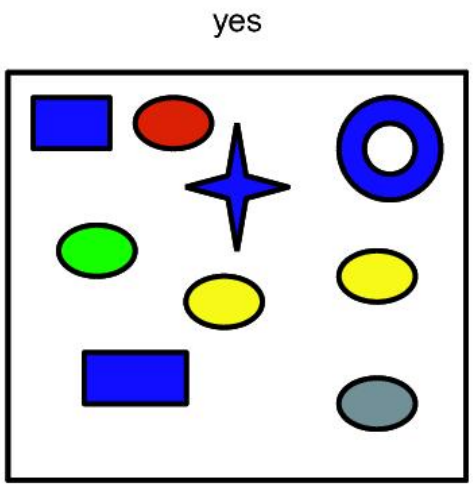
- **What is a decision tree?**
  - **Nodes:** test the value of feature x_j
  - **Branches**: correspond to values
  - **Leafs**: provide the class (prediction)

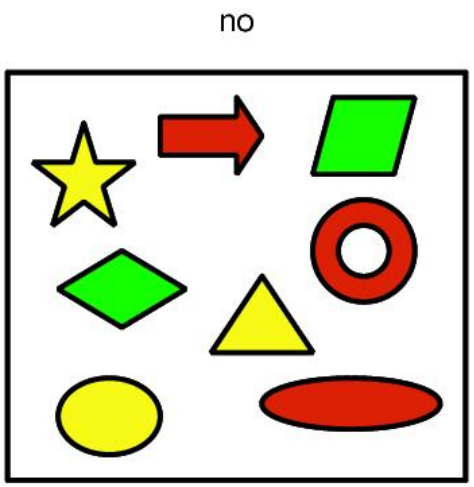
- **How to learn a decision tree from data?**
  - Finding the best tree is NP-hard
  - Scoring functions: Training error, Entropy, Information Gain, Gini Impurity

- **What is the inductive bias?**

Toy dataset and classification problem

Are these objects from outer space?



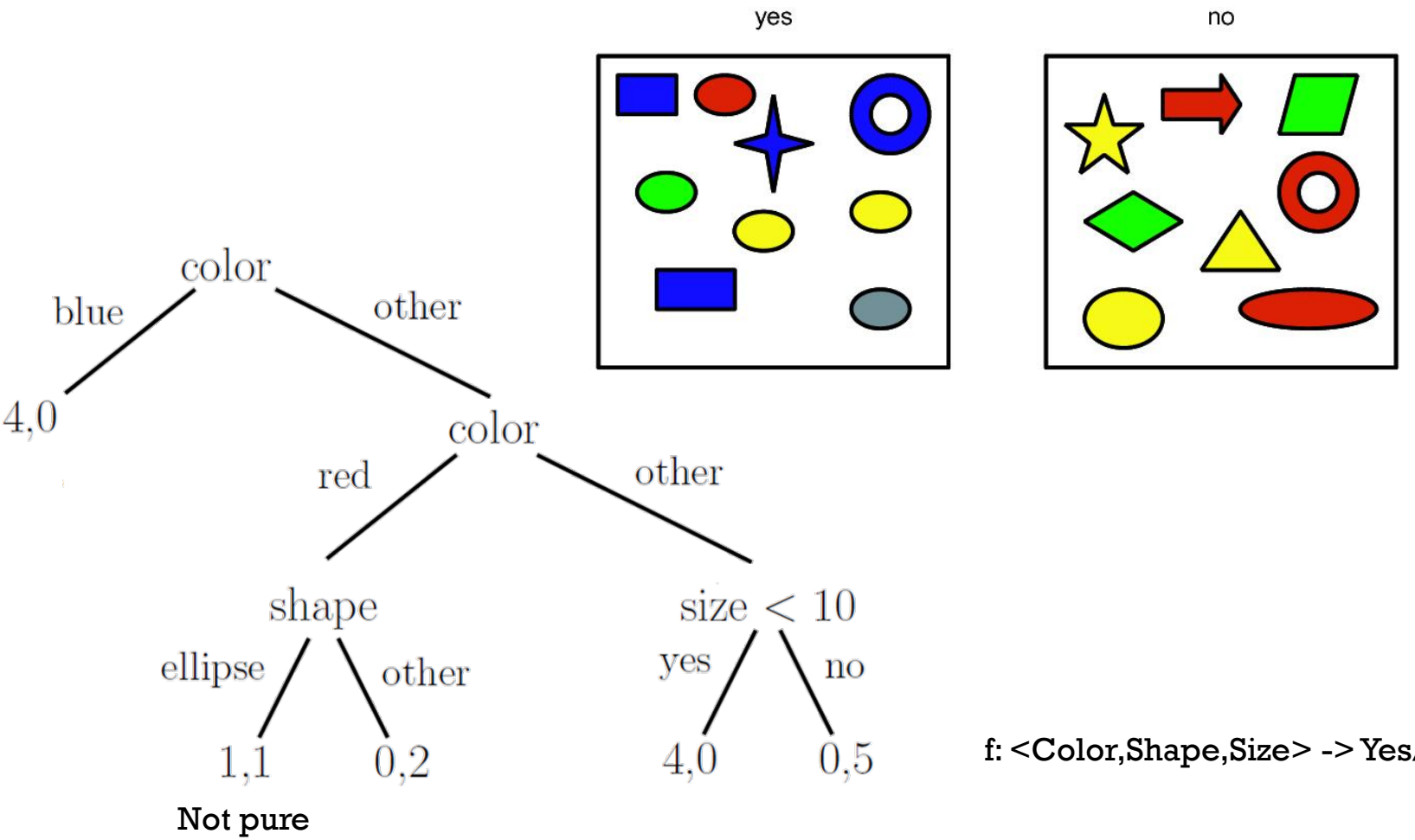


D features (attributes)

N cases

Color	Shape	Size (cm)
Blue	Square	10
Red	Ellipse	2.4
Red	Ellipse	20.7

Label
1
1
0



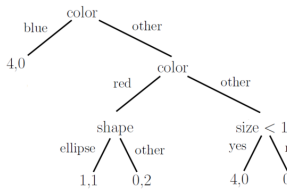
Not pure

f: <Color,Shape,Size> -> Yes/No?

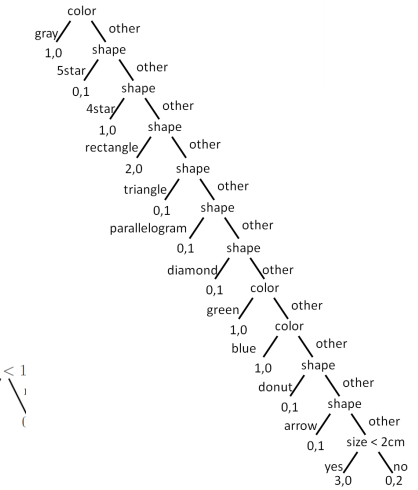
RECAP: SUPERVISED LEARNING SETTING

- Problem setting
  - **X** – set of possible instances
    - Each instance $\boldsymbol{x} \in X$ is a feature vector $\boldsymbol{x} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_D}]$
  - Unknown target function f: **X** $\rightarrow$ Y
    - Classification: Y is discrete valued
  - Set of function hypotheses H={h | h: **X** $\rightarrow$ Y}
    - Each hypothesis h is a decision tree

$\boldsymbol{h_1}$:



$\boldsymbol{h_2}$:



- Input
  - Training examples $\{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots (\boldsymbol{x}^{(N)}, y^{(N)})\}$ of unknown distribution

- Output
  - Hypothesis $h \in H$ that best approximates target function f

TODAY: LIMITS OF LEARNING

- Decision trees
  - What is the inductive bias?
  - Generalization

- Practical concerns:
  - How to estimate error on unseen examples?
  - Dealing with data: from raw data to well-defined examples
  - (Concerns apply to all models, not just decision trees)

NO FREE LUNCH

"All models are wrong, but some are useful."

George Box, Statistician

- Modeling assumptions that work well for one problem, may not work well for another
- No "universal learner" exists
  - We need many different techniques for different domains and task characteristics

BAYES OPTIMAL CLASSIFIER

- What if we had full access to the underlying data distribution D over (**x**,y) pairs?
- Bayes optimal classifier
  - For any input $\hat{x}$, we can compute the label $\hat{y}$

$$f^{(\mathrm{BO})}(\hat{x}) = \arg\max_{\hat{y} \in \mathcal{Y}} \mathcal{D}(\hat{x}, \hat{y})$$

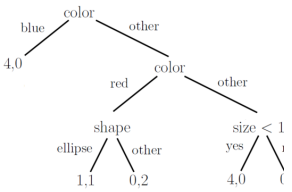
Theorem 1 (Bayes Optimal Classifier). *The Bayes Optimal Classifier $f^{(BO)}$ achieves minimal zero/one error of any deterministic classifier.*

- Bayes error rate: error of Bayes optimal classifier
- Unfortunately, we never have access to D – so what do we do?

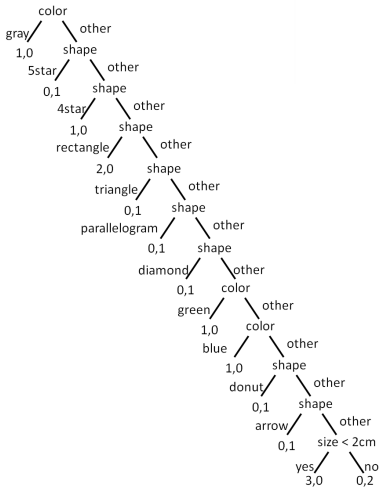
INDUCTIVE BIAS IN DECISION TREES

- Our learning algorithm performs heuristic search through space of decision trees
- It stops at smallest acceptable tree
- Why do we prefer small trees?
  - Occam's razor: prefer the simplest hypothesis that fits the data

h_1:



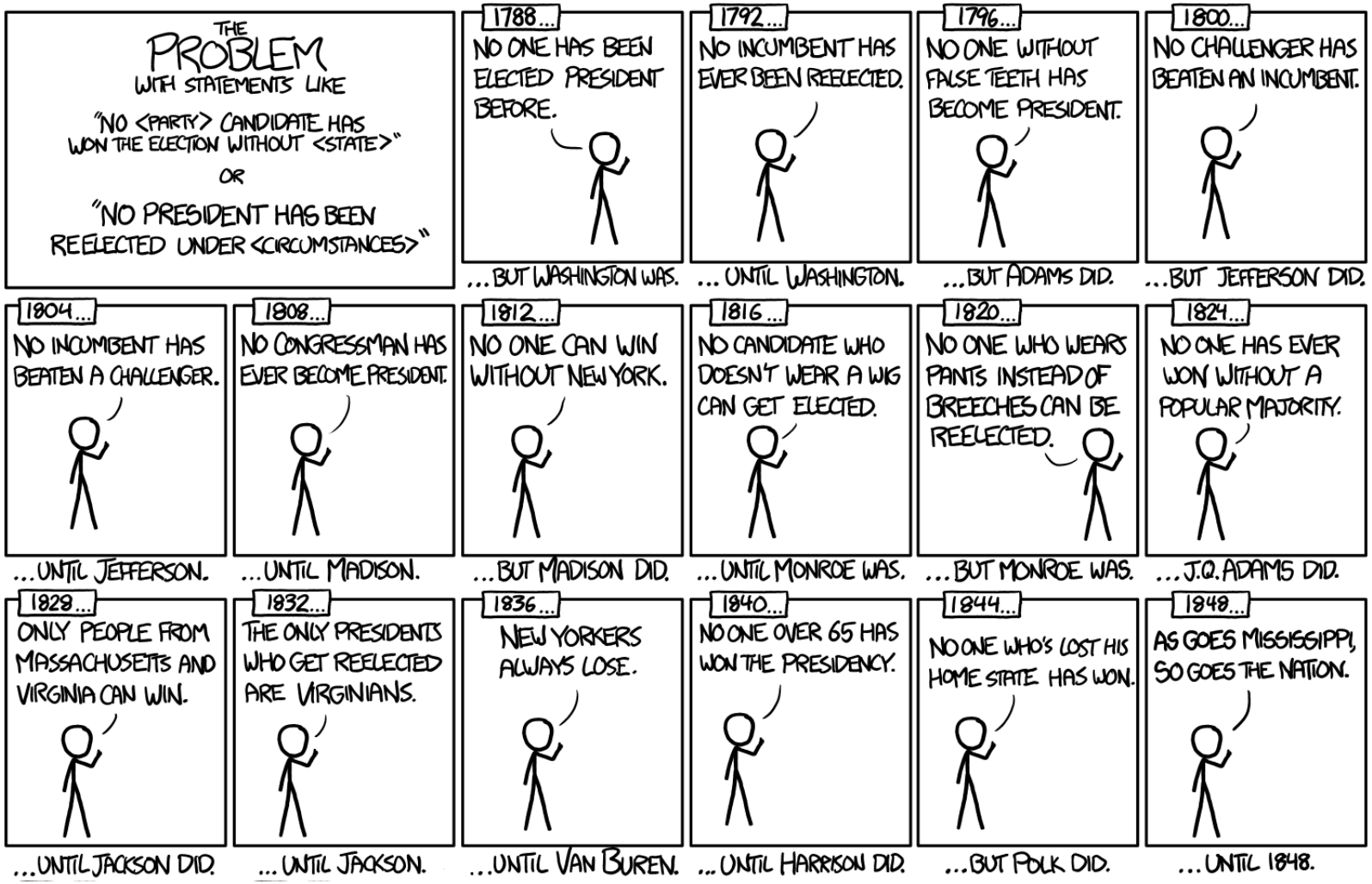
h_2:

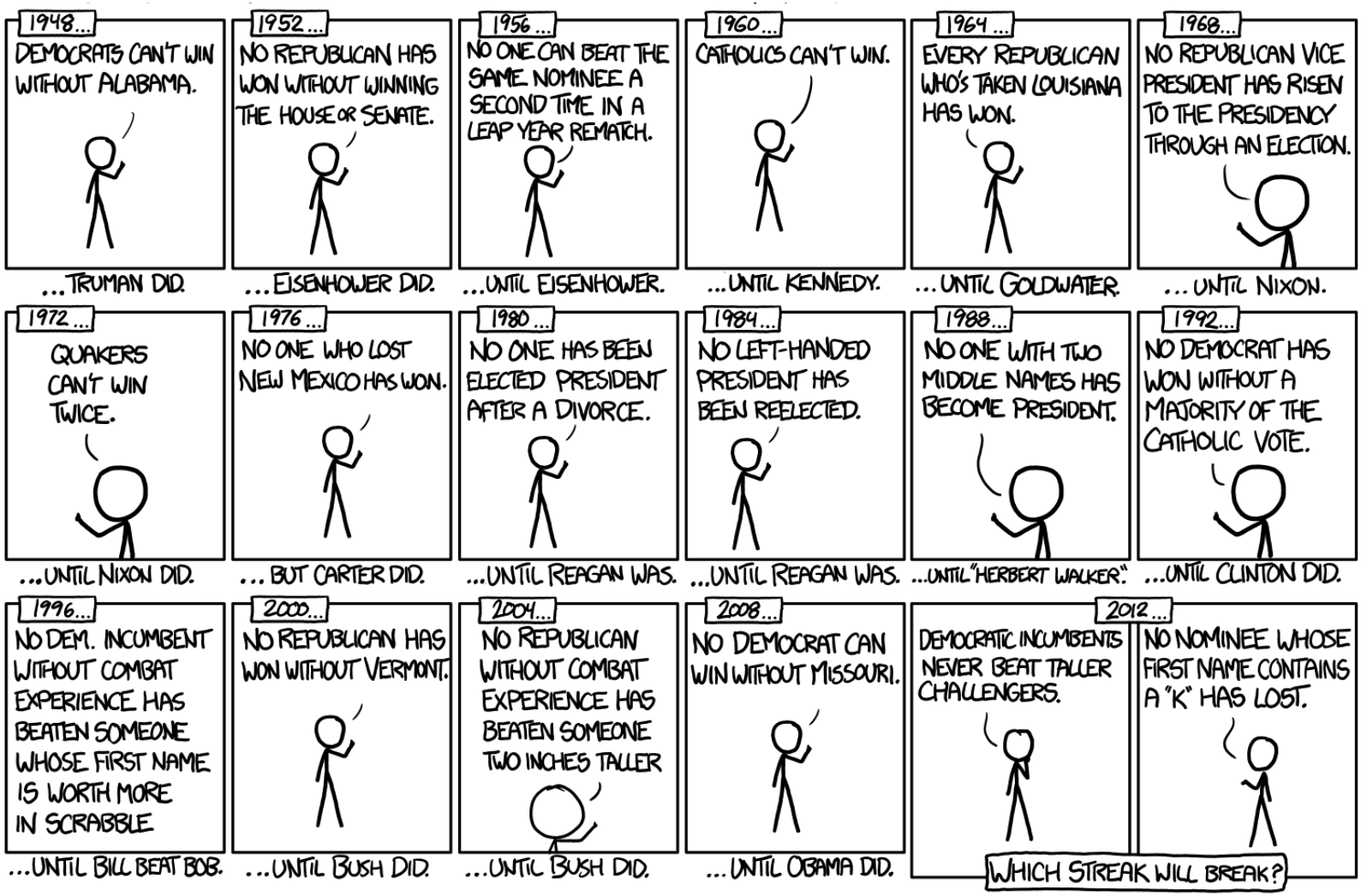


WHY PREFER SHORT HYPOTHESES?

- Pros
  - Fewer short hypotheses than long ones
  - A short hypothesis that fits the data is less likely to be a statistical coincidence

- Cons
  - What's so special about short hypotheses?


THE PROBLEM WITH STATEMENTS LIKE
"NO <PARTY> CANDIDATE HAS WON THE ELECTION WITHOUT <STATE>"
OR
"NO PRESIDENT HAS BEEN REELECTED UNDER <CIRCUMSTANCES>"
1788...
NO ONE HAS BEEN ELECTED PRESIDENT BEFORE.
...BUT WASHINGTON WAS.
1792...
NO INCUMBENT HAS EVER BEEN REELECTED.
...UNTIL WASHINGTON.
1796...
NO ONE WITHOUT FALSE TEETH HAS BECOME PRESIDENT.
...BUT ADAMS DID.
1800...
NO CHALLENGER HAS BEATEN AN INCUMBENT.
...BUT JEFFERSON DID.
1804...
NO INCUMBENT HAS BEATEN A CHALLENGER.
...UNTIL JEFFERSON.
1808...
NO CONGRESSMAN HAS EVER BECOME PRESIDENT.
...UNTIL MADISON.
1812...
NO ONE CAN WIN WITHOUT NEW YORK.
...BUT MADISON DID.
1816...
NO CANDIDATE WHO DOESN'T WEAR A WIG CAN GET ELECTED.
...UNTIL MONROE WAS.
1820...
NO ONE WHO WEARS PANTS INSTEAD OF BREECHES CAN BE REELECTED.
...BUT MONROE WAS.
1824...
NO ONE HAS EVER WON WITHOUT A POPULAR MAJORITY.
...J.Q. ADAMS DID.
1828...
ONLY PEOPLE FROM MASSACHUSETTS AND VIRGINIA CAN WIN.
...UNTIL JACKSON DID.
1832...
THE ONLY PRESIDENTS WHO GET REELECTED ARE VIRGINIANS.
...UNTIL JACKSON.
1836...
NEW YORKERS ALWAYS LOSE.
...UNTIL VAN BUREN.
1840...
NO ONE OVER 65 HAS WON THE PRESIDENCY.
...UNTIL HARRISON DID.
1844...
NO ONE WHO'S LOST HIS HOME STATE HAS WON.
...BUT POLK DID.
1848...
AS GOES MISSISSIPPI, SO GOES THE NATION.
...UNTIL 1848.


1948...
DEMOCRATS CAN'T WIN WITHOUT ALABAMA.
...TRUMAN DID.
1952...
NO REPUBLICAN HAS WON WITHOUT WINNING THE HOUSE OR SENATE.
...EISENHOWER DID.
1956...
NO ONE CAN BEAT THE SAME NOMINEE A SECOND TIME IN A LEAP YEAR REMATCH.
...UNTIL EISENHOWER.
1960...
CATHOLICS CAN'T WIN.
...UNTIL KENNEDY.
1964...
EVERY REPUBLICAN WHO'S TAKEN LOUISIANA HAS WON.
...UNTIL GOLDWATER.
1968...
NO REPUBLICAN VICE PRESIDENT HAS RISEN TO THE PRESIDENCY THROUGH AN ELECTION.
...UNTIL NIXON.
1972...
QUAKERS CAN'T WIN TWICE.
...UNTIL NIXON DID.
1976...
NO ONE WHO LOST NEW MEXICO HAS WON.
...BUT CARTER DID.
1980...
NO ONE HAS BEEN ELECTED PRESIDENT AFTER A DIVORCE.
...UNTIL REAGAN WAS.
1984...
NO LEFT-HANDED PRESIDENT HAS BEEN REELECTED.
...UNTIL REAGAN WAS.
1988...
NO ONE WITH TWO MIDDLE NAMES HAS BECOME PRESIDENT.
...UNTIL "HERBERT WALKER."
1992...
NO DEMOCRAT HAS WON WITHOUT A MAJORITY OF THE CATHOLIC VOTE.
...UNTIL CLINTON DID.
1996...
NO DEM. INCUMBENT WITHOUT COMBAT EXPERIENCE HAS BEATEN SOMEONE WHOSE FIRST NAME IS WORTH MORE IN SCRABBLE
...UNTIL BILL BEAT BOB.
2000...
NO REPUBLICAN HAS WON WITHOUT VERMONT.
...UNTIL BUSH DID.
2004...
NO REPUBLICAN WITHOUT COMBAT EXPERIENCE HAS BEATEN SOMEONE TWO INCHES TALLER
...UNTIL BUSH DID.
2008...
NO DEMOCRAT CAN WIN WITHOUT MISSOURI.
...UNTIL OBAMA DID.
2012...
DEMOCRATIC INCUMBENTS NEVER BEAT TALLER CHALLENGERS.
NO NOMINEE WHOSE FIRST NAME CONTAINS A "K" HAS LOST.
WHICH STREAK WILL BREAK?

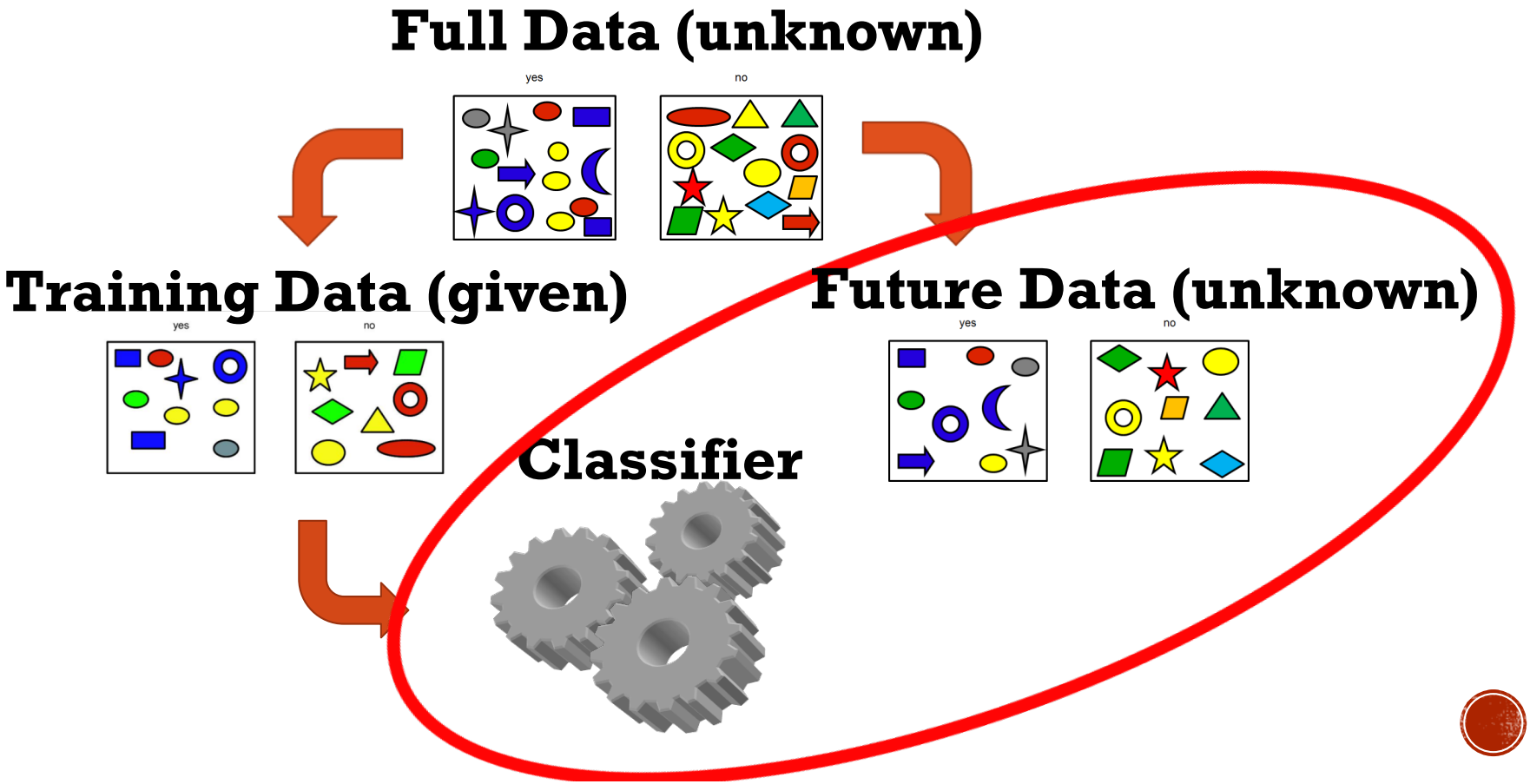
EVALUATING THE LEARNED HYPOTHESIS

- Assume
  - We've learned a tree h using the top-down induction algorithm
  - It fits the data perfectly

- Are we done? Can we guarantee we have found a good hypothesis?

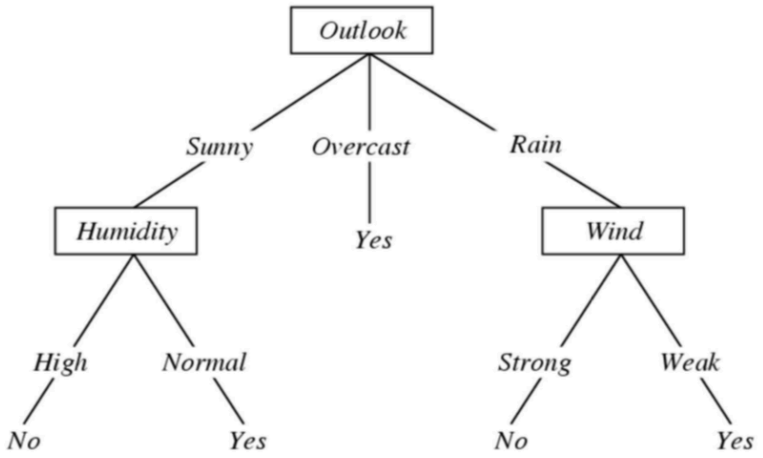
- Ideally, we want a hypothesis that can explain the past but is also simple enough to generalize to the future

SUPERVISED LEARNING



SHOULD WE PLAY TENNIS?

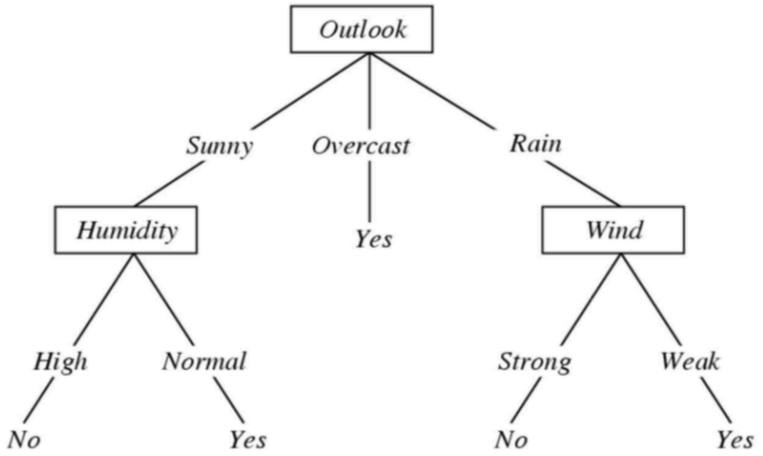
Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal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



LET'S ADD A NOISY EXAMPLE

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
D15	Sunny	Hot	Normal	Strong	No

How does this affect the decision tree?



RECALL: FORMALIZING INDUCTION

- f(**x**) should make good predictions
  - As measured by loss function
  - On future examples (typically) drawn from D

- Given
  - A loss function
  - A sample from some unknown data distribution $\{(\boldsymbol{x}^{(1)}, y^{(1)}),\ldots(\boldsymbol{x}^{(N)}, y^{(N)})\}$

- Compute a function f that has a low expected error over D with respect to the loss

$$\varepsilon \triangleq \mathbb{E}_{(x,y)\sim D}\{l(y, f(x))\} = \sum_{(x,y)} D(x, y) l(y, f(x))$$

TRAINING ERROR IS NOT SUFFICIENT

- We care about **generalization** to new examples

- A tree can classify training data perfectly, yet classify new examples incorrectly

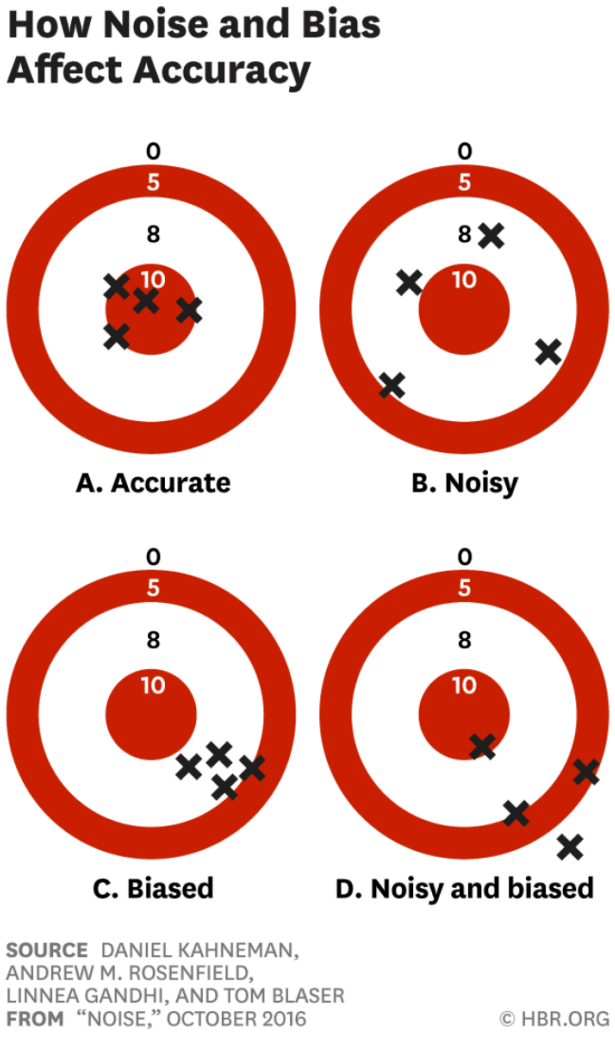
-
  - Because training examples are only a sample of data distribution
    - a feature might correlate with class by coincidence
  - Because training examples could be noisy
    - e.g., accident in labeling

NOISE VS. BIAS IN TRAINING DATA

- Goal: estimate target in dart-throwing
- Training data: four sets of four examples

- Noise = variance

Example from https://hbr.org/2016/10/noise

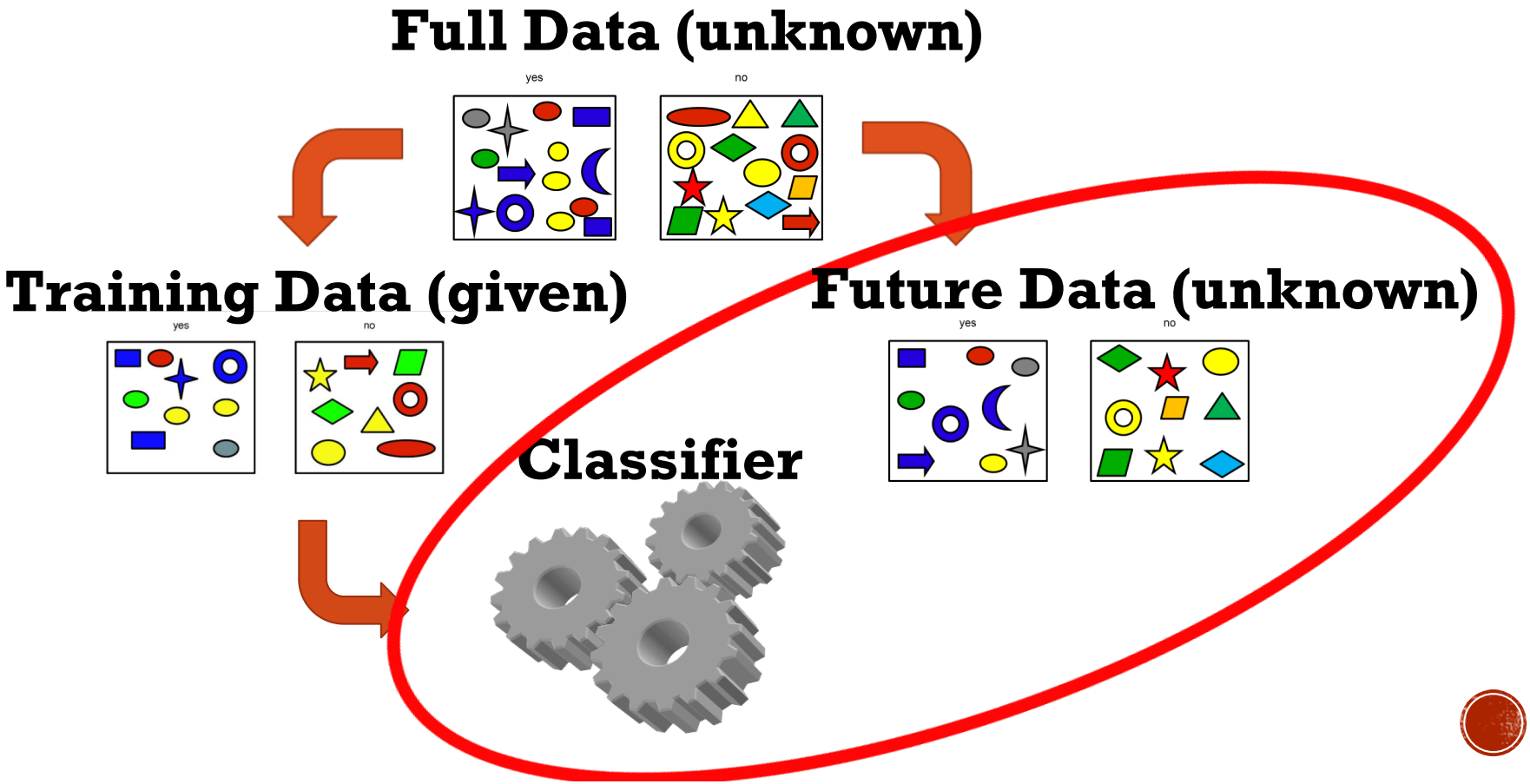


NEXT: ERROR ON UNSEEN EXAMPLES

- Decision trees
  - What is the inductive bias?
  - Generalization

- **Practical concerns**
  - How to estimate error on unseen examples?
  - Dealing with data: from raw data to well-defined examples

SUPERVISED LEARNING

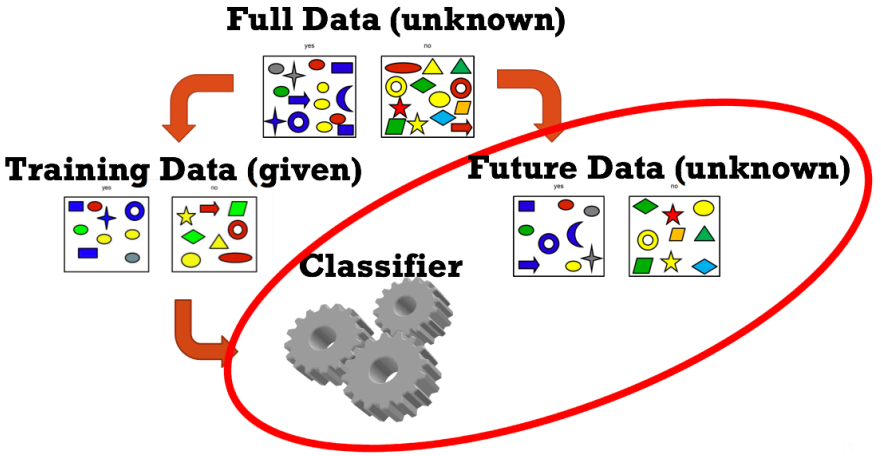


OVERFITTING

- Consider a hypothesis h and its:
  - Error rate over training data $error_{train}(h)$
  - True error rate over full data $error_{true}(h)$

- We say that h overfits the training data if
  - $error_{true}(h) > error_{train}(h)$

- Amount of overfitting
  - $error_{true}(h) - error_{train}(h)$

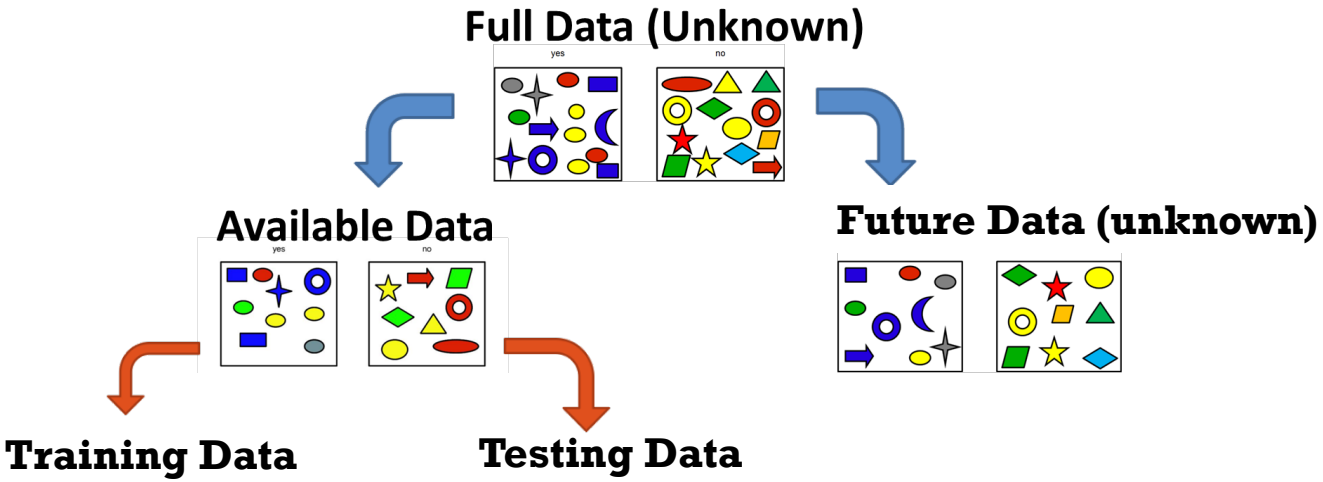


HOW DO WE EVALUATE PERFORMANCE?

- Problem
  - we don't know $error_{true}(h)$!

- How do we evaluate then?
  - we set aside a testing set from the available data
    - some examples that will be used for evaluation
  - we don't look at them during training! (pretend we have not seen them)
  - after learning a model, we calculate $error_{test}(h)$

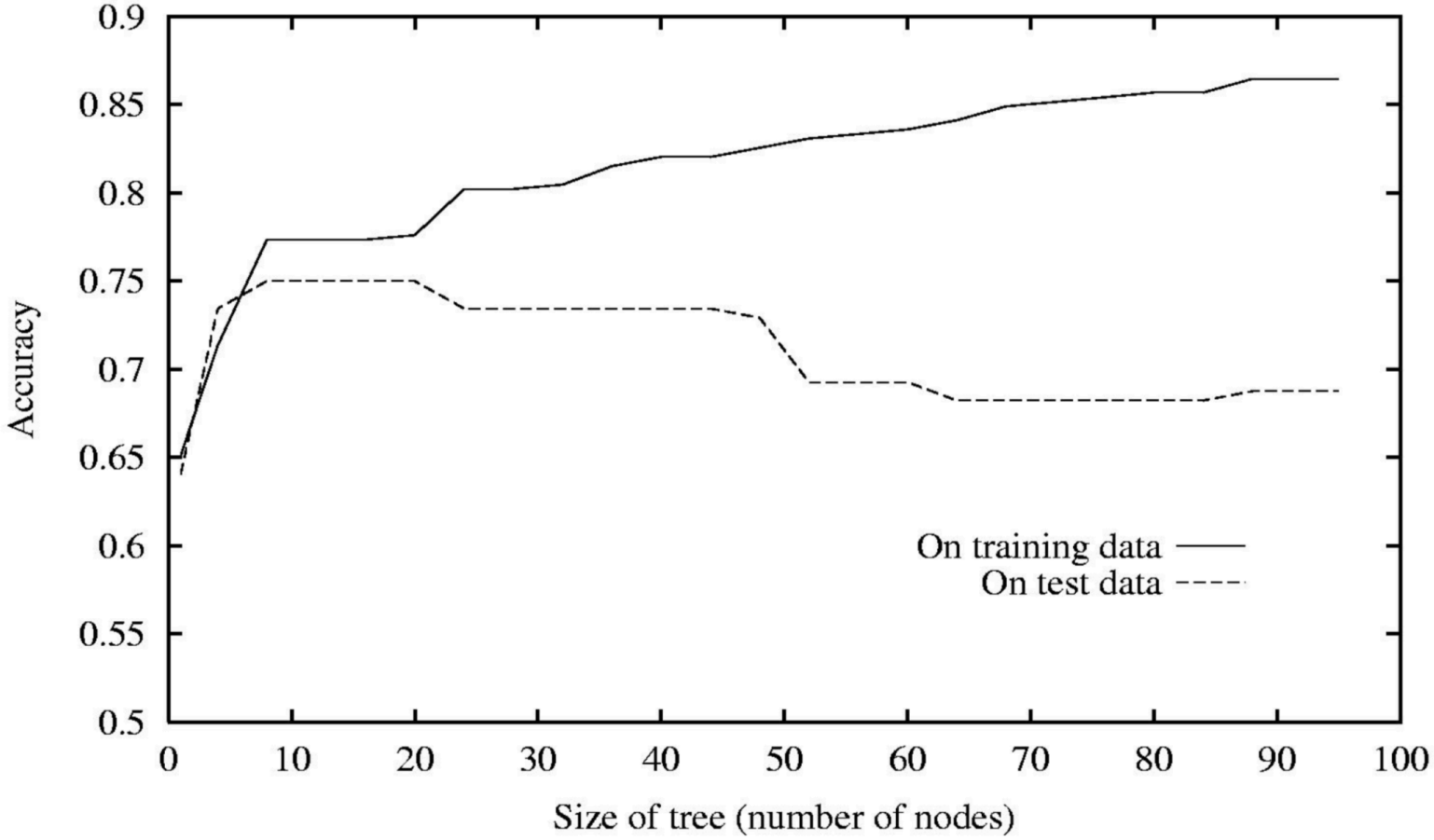
SUPERVISED LEARNING EVALUATION



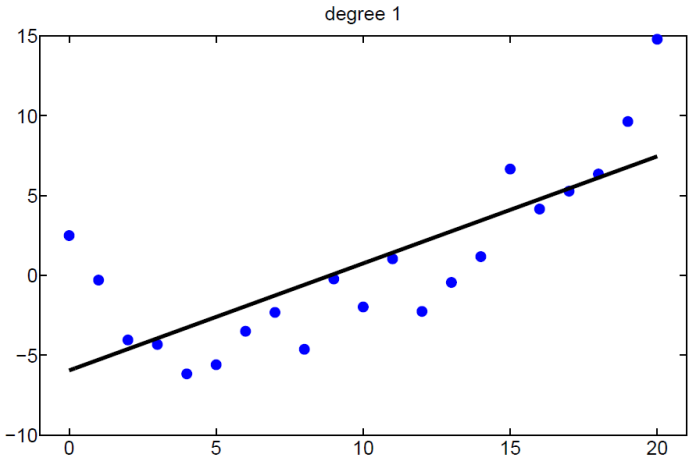
We use testing data for evaluation

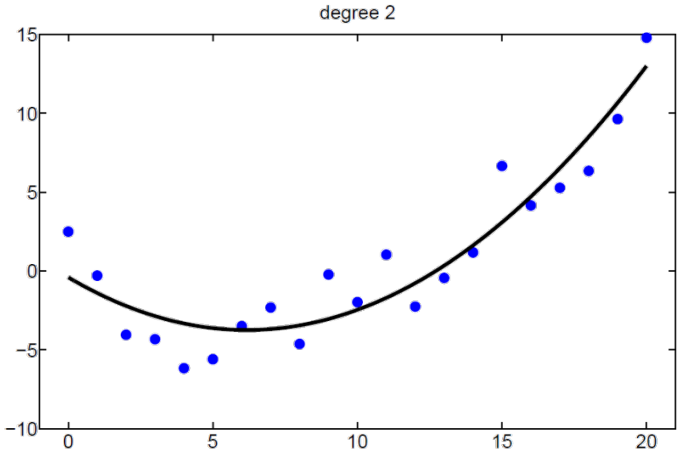
We don't look at testing data during training!

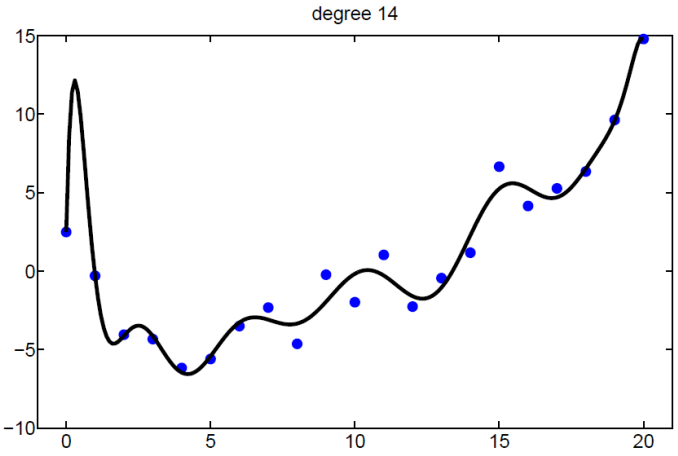
MEASURING OVERFITTING EFFECT

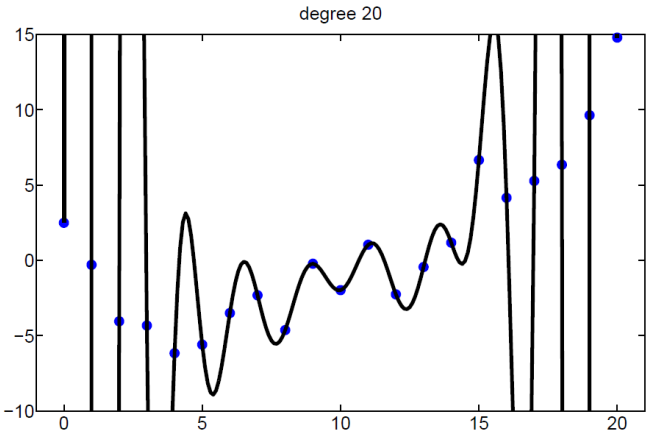


OVERFITTING IN REGRESSION









OVERFITTING

- We say that h overfits the training data if
  - $error_{true}(h) > error_{train}(h)$

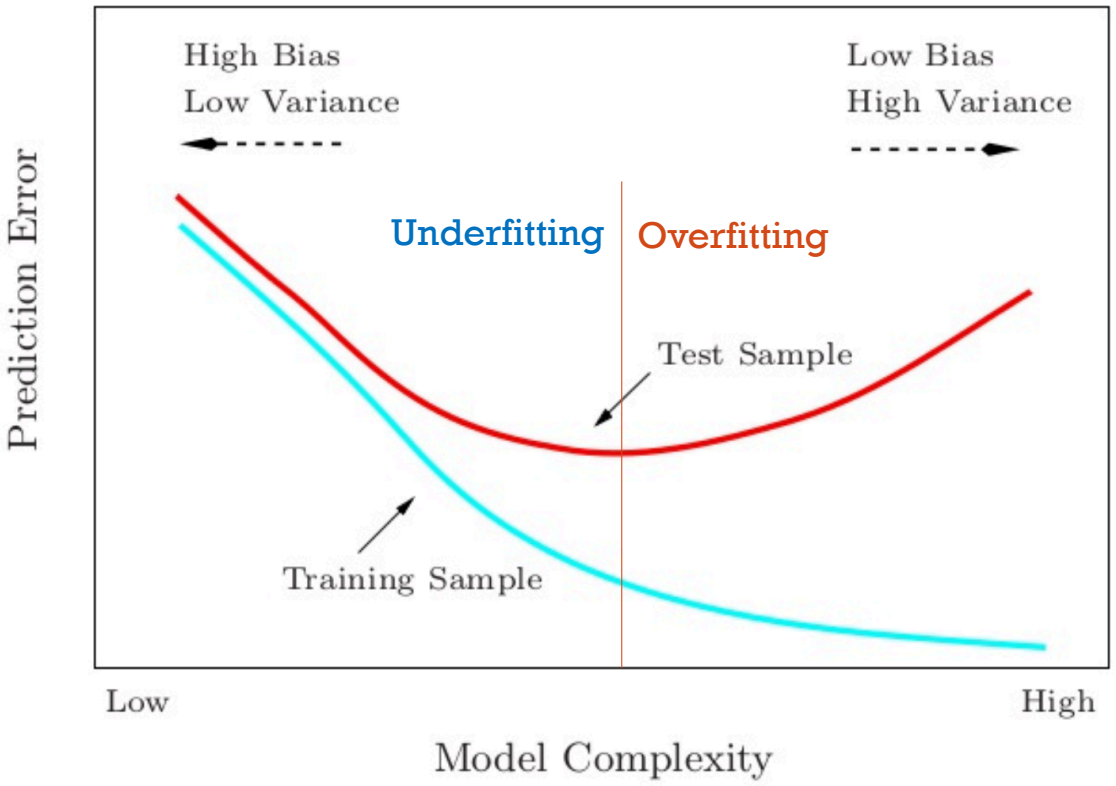
- Another way of putting it: A hypothesis h is said to overfit the training data, if there is another hypothesis h', such that
  - h has a smaller error than h' on the training data
  - but h has larger error on the test data than h'.

UNDERFITTING VS. OVERFITTING

- Underfitting
  - Learning algorithm had the opportunity to learn more from training data, but didn't

- Overfitting
  - Learning algorithm paid too much attention to idiosyncracies of the training data; the resulting tree doesn't generalize

UNDERFITTING VS. OVERFITTING



PRACTICAL IMPACT

- What we want:
  - A decision tree that neither underfits nor overfits
  - Because it is expected to do best in the future

- How can we encourage that behavior?
  - Set a maximum tree depth D

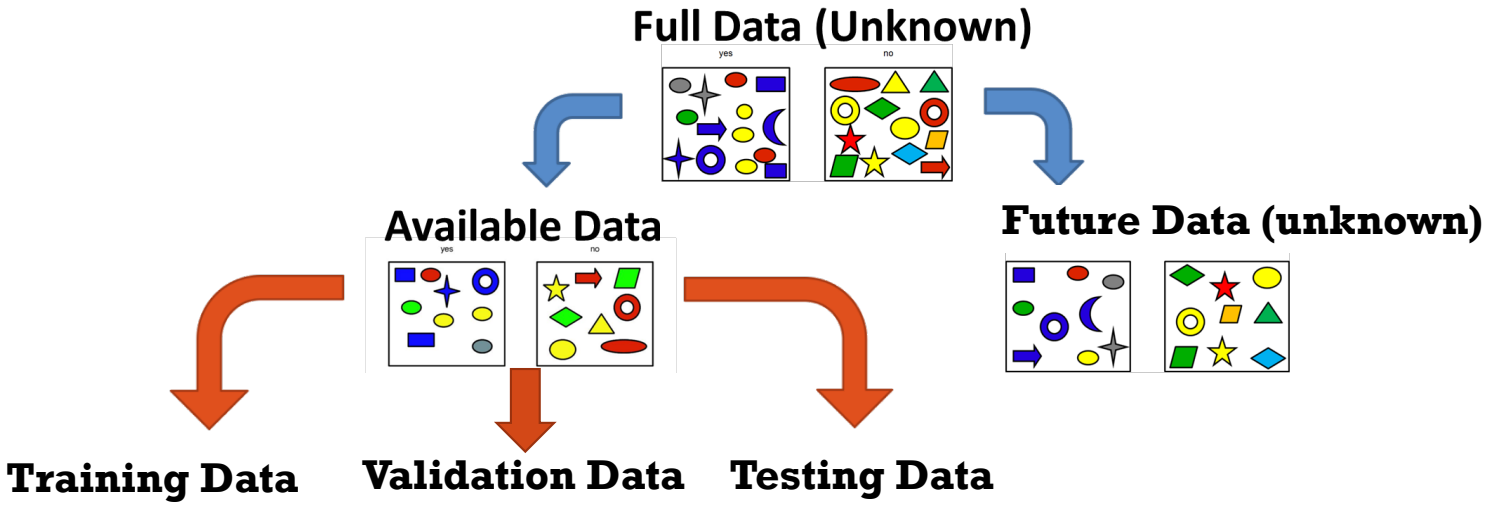
- How do we learn this maximum tree depth D?
  - Hint: We can't use testing data during training!

HOW TO LEARN FROM DATA

- In practice, we always split examples into 3 distinct sets
- **Training set**
  - Used to learn the **parameters** of the ML model
  - e.g., what are the nodes and branches of the decision tree
- **Validation set**
  - aka tuning set, aka development set, aka held-out data
  - Used to learn **hyperparameters**
    - Parameter that controls other parameters of the model
      - e.g., max depth of decision tree
- **Test set**

  – Used to evaluate how well we're doing on new unseen examples

SUPERVISED LEARNING



We use testing data for evaluation

We don't look at testing data during training!

SUMMARY

- Decision trees
  - What they are, and how to learn one
- Fundamental machine learning concepts
  - What inductive bias is and what is its role in learning
  - What underfitting and overfitting means, and how it's related to the bias-variance tradeoff
  - Why you should never touch your test data

ANNOUNCEMENTS

- Quizzes have been graded
  - If you signed up for Gradescope, you would be able to see your result there
- If you are still trying to register and you have taken the quiz, fill out the form on Piazza
  - I will send individual emails to students the class can accommodate
- Any questions about the homework?

ACKNOWLEDGEMENTS

- These slides use materials by Brian Ziebart, Marine Carpuat, Tom Mitchell, and Jake Hofman