Decision Trees & Limits of Learning

CMSC 422

MARINE CARPUAT
marine@cs.umd.edu

Credit: some examples & figures by Tom Mitchell

Today's Topics

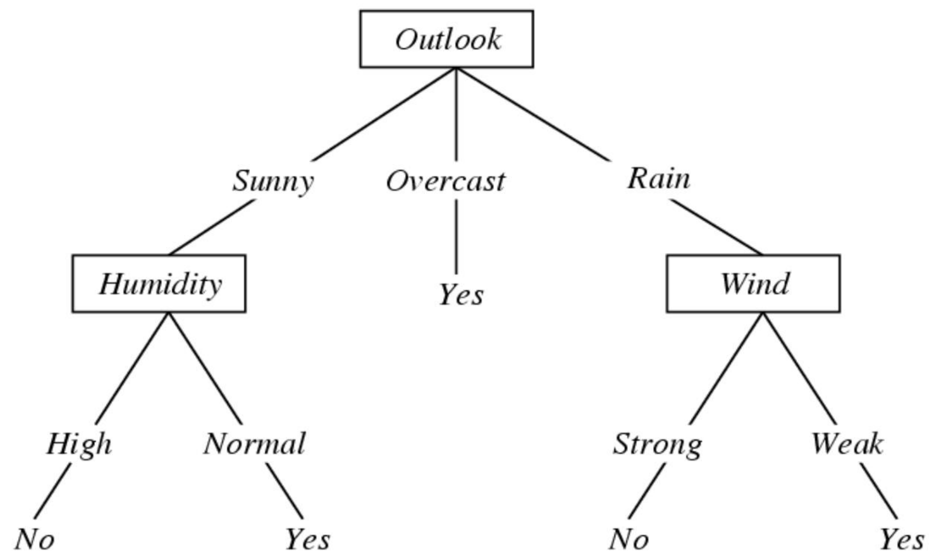
- Decision trees
  - What is the inductive bias?
  - Generalization issues: overfitting/underfitting
- Practical concerns: dealing with data
  - Train/dev/test sets
  - From raw data to well-defined examples

DECISION TREES

Recap: An example training set

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

Recap: A decision tree to decide whether to play tennis



Recap: Function Approximation with Decision Trees

Problem setting

- Set of possible instances X
  - Each instance $x \in X$ is a feature vector $x = [x_1, \dots, x_D]$
- Unknown target function $f: X \rightarrow Y$
  - Y is discrete valued
- Set of function hypotheses $H = \{h \mid h: X \rightarrow Y\}$
  - Each hypothesis h is a decision tree

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \dots (x^{(N)}, y^{(N)})\}$ of unknown target function f

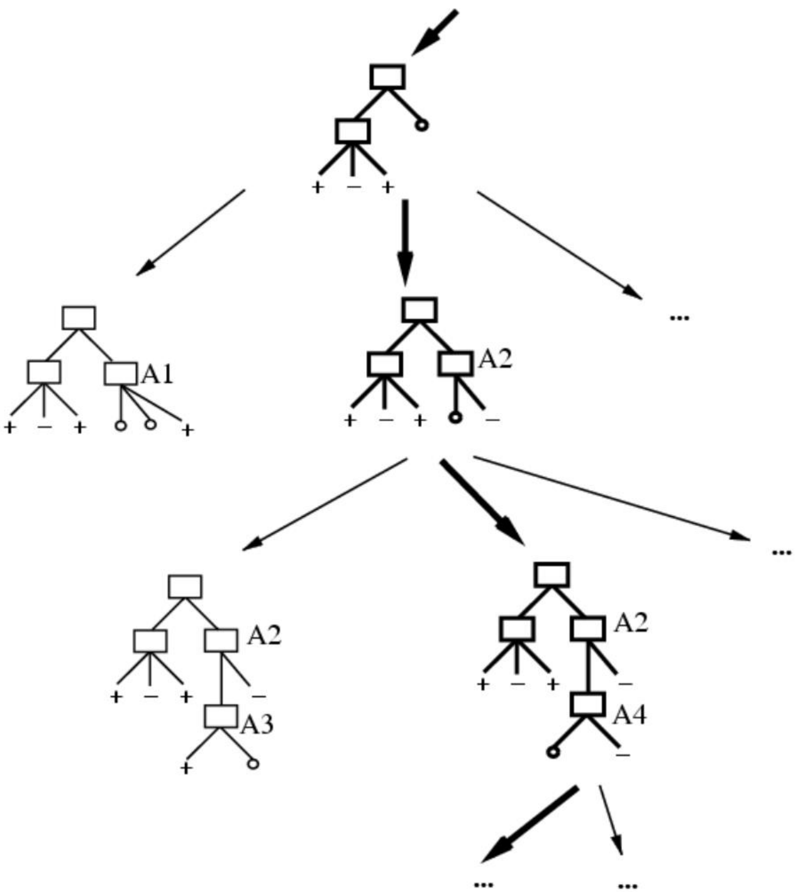
Output

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

Decision Trees

- What is a decision tree?
- How to learn a decision tree from data?
- What is the inductive bias?
- Generalization?
  - Overfitting/underfitting
  - Selecting train/dev/test data

Inductive bias in decision tree learning



- 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

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?

Evaluating the learned hypothesis $\boldsymbol{h}$

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

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

Recall: Formalizing Induction

- Given
  - a loss function l
  - a sample from some unknown data distribution D

- Our task is to compute a function f that has low expected error over D with respect to l.

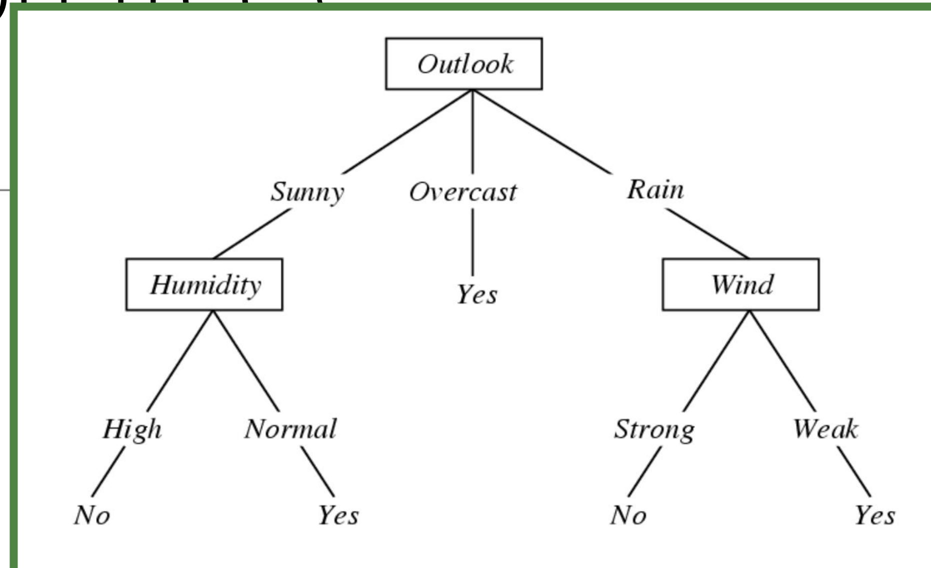
$$\mathbb{E}_{(x,y)\sim D}\{l(y, f(x))\} = \sum_{(x,y)} D(x, y) l(y, f(x))$$

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

Let's add a noisy training example. How does this affect the learned decision tree?



Day	Outlook	Temperature	Humidity	Wind	
D1	Sunny	Hot	High	Weak	
D2	Sunny	Hot	High	Strong	
D3	Overcast	Hot	High	Weak	
D4	Rain	Mild	High	Weak	
D5	Rain	Cool	Normal	Weak	
D6	Rain	Cool	Normal	Strong	
D7	Overcast	Cool	Normal	Strong	
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

Overfitting

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

- We say h overfits the training data if

$$error_{train}(h) < error_{true}(h)$$

- Amount of overfitting =

$$error_{true}(h) - error_{train}(h)$$

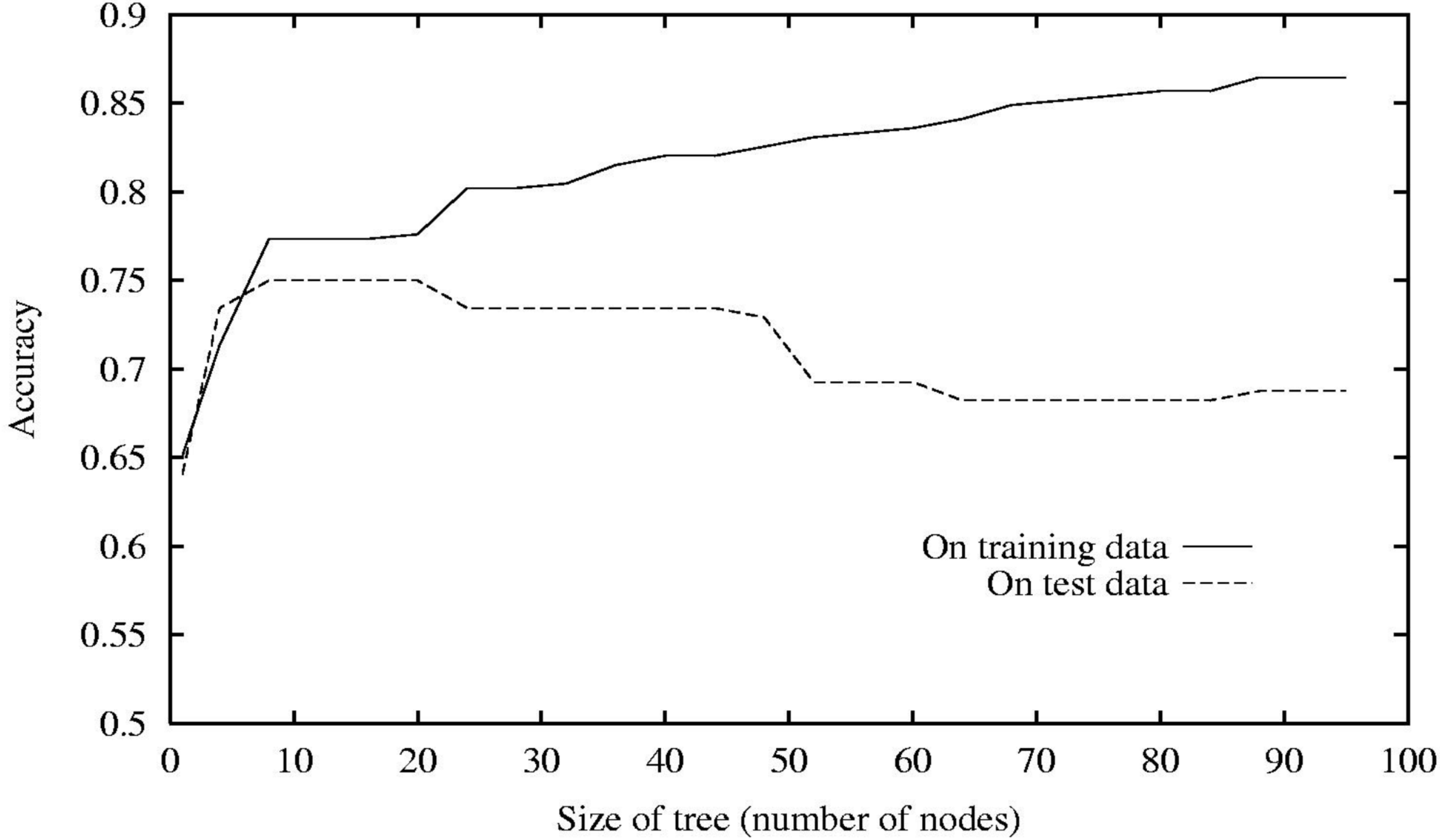
Evaluating on test data

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

- Solution:
  - we set aside a test set
    - some examples that will be used for evaluation
  - we don't look at them during training!
  - after learning a decision tree, we calculate

$$error_{test}(h)$$

Measuring effect of overfitting in decision trees



Overfitting

- 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/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

Practical impact on decision tree learning

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

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

Decision Trees

- What is a decision tree?
- How to learn a decision tree from data?
- What is the inductive bias?
  - Occam's razor: preference for short trees
- Generalization?
  - Overfitting/underfitting

Your thoughts?

What are the pros and cons of decision trees?

DEALING WITH DATA

What real data looks like...

Class y

Example

1 robocop is an intelligent science fiction thriller and social satire , one with class and style . the film , set in old detroit in the year 1991 , stars peter weller as murphy , a lieutenant on the city's police force . 1991's de … a police departmen … rity concepts … are threateni … e , a savage group of … city . […]

0 do the … cy ? they have resu … ed it into a live acti … ffects , embarrassing writing and kid-friendly slapstick . wasn't mr . magoo enough , people ? obviously not . inspector gadget is not what i would call ideal family entertainment . […]

How would you define input vectors x to represent each example? What features would you use?

Train/dev/test sets

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
- **Development set**
  - aka tuning set, aka validation 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

Cardinal rule of machine learning:

Never *ever* touch your test data!

Summary: what you should know

Decision Trees

- What is a decision tree, and how to induce it from data

Fundamental Machine Learning Concepts

- Difference between memorization and generalization
- What inductive bias is, and what is its role in learning
- What underfitting and overfitting means
- How to take a task and cast it as a learning problem
- **Why you should never ever touch your test data!!**