COS 324: Lecture 13

Beyond linear classifiers: decision trees

Elad Hazan    Yoram Singer

This lecture contains material from the T. Michel text “Machine Learning”, and slides adapted from David Sontag, Luke Zettlemoyer, Carlos Guestrin, and Andrew Moore
Admin

• New exercise – theory – due in two weeks
  (formal announcement next week, but out now for your convenience)
Agenda

Thus far:
• Rigorous definition of (PAC) learnability
• Efficient algorithms for learning based on convex optimization
• linear classifiers (perceptron, SGD, multiclass,...)

Today:
• Decision trees
• Build up for other non-linear machines (& neural networks)
Classification

Goal: Find best mapping from domain (features) to output (labels)

• Given a document (email), classify spam or ham. Features = words, labels = {spam, ham}

• Given a picture, classify if it contains a chair or not. Features = bits in a bitmap image, labels = {chair, no chair}

GOAL: automatic machine that learns from examples

Terminology for learning from examples:
• Set aside a ”training set” of examples, train a classification machine
• Test on a “test set”, to see how well machine performs on unseen examples
Classifying fuel efficiency

- 40 data points
- Goal: predict MPG
- Need to find:
  \[ f : X \rightarrow Y \]
- Discrete data (for now)

<table>
<thead>
<tr>
<th>mpg</th>
<th>cylinders</th>
<th>displacement</th>
<th>horsepower</th>
<th>weight</th>
<th>acceleration</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>75to78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>75to78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>asia</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>70to74</td>
<td>asia</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>high</td>
<td>75to78</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>medium</td>
<td>low</td>
<td>low</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>medium</td>
<td>79to83</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>70to74</td>
<td>america</td>
<td></td>
</tr>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>75to78</td>
<td>europe</td>
<td></td>
</tr>
<tr>
<td>bad</td>
<td>5</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>75to78</td>
<td>europe</td>
<td></td>
</tr>
</tbody>
</table>
Decision trees for classification

• Why use decision trees?
• What is their expressive power?
• Can they be constructed automatically?
• How accurate can they classify?
• How well do decision trees generalize? (sample complexity)
• Computational complexity of finding the best tree
Decision trees for classification

Some real examples (from Russell & Norvig, Mitchell)

• BP’s GasOIL system for separating gas and oil on offshore platforms - decision trees replaced a hand-designed rules system with 2500 rules. C4.5-based system outperformed human experts and saved BP millions. (1986)
• learning to fly a Cessna on a flight simulator by watching human experts fly the simulator (1992)
• can also learn to play tennis, analyze C-section risk, etc.
Decision trees for classification

- interpretable/intuitive, popular in medical applications because they mimic the way a doctor thinks
- model discrete outcomes nicely
- C4.5 and CART - from “top 10 data mining methods” - very popular
- very expressive
decision trees $f : X \rightarrow Y$

- Each internal node tests an attribute $x_i$
- One branch for each possible attribute value $x_i=v$
- Each leaf assigns a class $y$
- To classify input $x$: traverse the tree from root to leaf, output the labeled $y$

Human interpretable!
Expressive power of DT

Consider Boolean functions
\[ F = \{0,1\}^n \rightarrow \{0,1\} \]

- How many functions can DT express?

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>F(X1,X2,X3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Sample complexity of DT

- Sample complexity of all decision trees?
- Smaller trees? (bound their size)

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>F(X1,X2,X3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What is the Simplest Tree?

<table>
<thead>
<tr>
<th>mpg</th>
<th>cylinders</th>
<th>displacement</th>
<th>horsepower</th>
<th>weight</th>
<th>acceleration</th>
<th>modelyear</th>
<th>maker</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>4</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>78 to 78</td>
<td>asia</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>78 to 78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>78 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>78 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>78 to 74</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>78 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>78 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>75 to 78</td>
<td>europe</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>6</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>4</td>
<td>low</td>
<td>medium</td>
<td>medium</td>
<td>medium</td>
<td>75 to 78</td>
<td>america</td>
</tr>
<tr>
<td>bad</td>
<td>8</td>
<td>high</td>
<td>high</td>
<td>high</td>
<td>low</td>
<td>75 to 78</td>
<td>america</td>
</tr>
</tbody>
</table>

Is this a good tree?

predict mpg=bad

[22+, 18-] Means: correct on 22 examples incorrect on 18 examples
Are all decision trees equal?

• Many trees can represent the same concept
• But, not all trees will have the same size!
  – e.g., ((A and B) or (not A and C))

• Which tree do we prefer?
Sample complexity of DT

- How many trees over d Boolean variables with k nodes?
  \[ \leq d^k \times (2k + 1)! \]

A binary tree over k variables has k+1 leaves. To bound how many such trees: add k+1 "leaves", write down the tree in DFS order.

Choose k variables from d, with repetition
Sample complexity of DT

- How many trees over \(d\) Boolean variables with \(k\) nodes?
  \[d^k \times (2k + 1)\!

Thus, by fundamental theorem of statistical learning, sample complexity is:

\[
O\left(\frac{|H| \log \frac{1}{\delta}}{\varepsilon}\right) = O\left(\frac{k \log(d) + \log \frac{1}{\delta}}{\varepsilon}\right)
\]
Computational complexity of DT

• Good news: sample complexity is descent. Fundamental theorem says we can learn with ERM rule!

• Bad news: Learning the simplest (smallest) decision tree is an NP-complete problem [Hyafil & Rivest ’76]

• Solution 1: Resort to a greedy heuristic:
  – Start from empty decision tree
  – Split on next best attribute (feature)
  – Recurs

• Next week: more rigorous theoretical solution – Boosting!
A Decision Stump

mpg values: bad good

root
22 18
pchance = 0.001

cylinders = 3
  0 0
  Predict bad

cylinders = 4
  4 17
  Predict good

cylinders = 5
  1 0
  Predict bad

cylinders = 6
  8 0
  Predict bad

cylinders = 8
  9 1
  Predict bad
Key idea: Greedily learn trees using recursion

Take the Original Dataset...

And partition it according to the value of the attribute we split on

Records in which cylinders = 4
Records in which cylinders = 5
Records in which cylinders = 6
Records in which cylinders = 8

mpg values: bad good

root
22 18
pchance = 0.001

cylinders = 3
cylinders = 4
cylinders = 5
cylinders = 6
cylinders = 8

0 0
4 17
1 0
8 0
9 1

Predict bad Predict good Predict bad Predict bad Predict bad
Recursive Step

Records in which cylinders = 4
Records in which cylinders = 5
Records in which cylinders = 6
Records in which cylinders = 8

Build tree from These records..
Build tree from These records..
Build tree from These records..
Build tree from These records..

mpl values: bad good

root
22 18
pvalue = 0.001

cylinders = 3
0 0
Predict bad

cylinders = 4
4 17
Predict good

cylinders = 5
1 0
Predict bad

cylinders = 6
8 0
Predict bad

cylinders = 8
9 1
Predict bad
Second level of tree

Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

(Similar recursion in the other cases)
A full tree

mpg values: bad good

root

- cylinders = 3
  - maker = america
    - horsepower = low
      - acceleration = low
        - Predict bad
      - acceleration = medium
        - Predict good
      - acceleration = high
        - Predict bad
    - horsepower = medium
      - Predict bad
    - horsepower = high
      - Predict bad
  - maker = asis
    - Predict good
    - pchance = 0.317
  - maker = europe
    - horsepower = low
      - acceleration = low
        - Predict bad
      - acceleration = medium
        - Predict bad
      - acceleration = high
        - Predict bad
    - horsepower = medium
      - pchance = 0.717
    - horsepower = high
      - pchance = 0.717

- cylinders = 4
  - pchance = 0.135
  - Predict bad

- cylinders = 5
  - pchance = 0.085
  - Predict bad
  - Predict good

- cylinders = 6
  - Predict bad

- cylinders = 8
  - Predict bad

pchance = 0.001
Splitting: choosing a good attribute

Would we prefer to split on $X_1$ or $X_2$?

Idea: use counts at leaves to define probability distributions, so we can measure uncertainty!
Measuring uncertainty

- Good split if we are more certain about classification after split
  - Deterministic good (all true or all false)
  - Uniform distribution bad
  - What about distributions in between?

\[
\begin{array}{cccc}
P(Y=A) &=& 1/2 & P(Y=B) = 1/4 \\
& & & P(Y=C) = 1/8 \\
& & & P(Y=D) = 1/8 \\
\end{array}
\]

\[
\begin{array}{cccc}
P(Y=A) &=& 1/4 & P(Y=B) = 1/4 \\
& & & P(Y=C) = 1/4 \\
& & & P(Y=D) = 1/4 \\
\end{array}
\]
Entropy

Entropy $H(Y)$ of a random variable $Y$

$$H(Y) = - \sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i)$$

More uncertainty, more entropy!

Information Theory interpretation: $H(Y)$ is the expected number of bits needed to encode a randomly drawn value of $Y$ (under most efficient code)
High, Low Entropy

• “High Entropy”
  – Y is from a uniform like distribution
  – Flat histogram
  – Values sampled from it are less predictable

• “Low Entropy”
  – Y is from a varied (peaks and valleys) distribution
  – Histogram has many lows and highs
  – Values sampled from it are more predictable

(Slide from Vibhav Gogate)
Entropy Example

\[ H(Y) = - \sum_{i=1}^{k} P(Y = y_i) \log_2 P(Y = y_i) \]

\[ P(Y=t) = \frac{5}{6} \]

\[ P(Y=f) = \frac{1}{6} \]

\[ H(Y) = - \frac{5}{6} \log_2 \frac{5}{6} - \frac{1}{6} \log_2 \frac{1}{6} \]

\[ = 0.65 \]
Conditional Entropy

Conditional Entropy $H(Y | X)$ of a random variable $Y$ conditioned on a random variable $X$

$$H(Y | X) = - \sum_{j=1}^{v} P(X = x_j) \sum_{i=1}^{k} P(Y = y_i | X = x_j) \log_2 P(Y = y_i | X = x_j)$$

Example:

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$Y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

$P(X_1=\text{t}) = 4/6$

$P(X_1=\text{f}) = 2/6$

$H(Y|X_1) = - 4/6 (1 \log_2 1 + 0 \log_2 0) - 2/6 (1/2 \log_2 1/2 + 1/2 \log_2 1/2) = 2/6$
**Information gain**

- Decrease in entropy (uncertainty) after splitting

\[ IG(X) = H(Y) - H(Y \mid X) \]

In our running example:

\[
IG(X_1) = H(Y) - H(Y \mid X_1) \\
= 0.65 - 0.33
\]

\[ IG(X_1) > 0 \rightarrow \text{we prefer the split!} \]
Learning decision trees

• Start from empty decision tree
• Split on next best attribute (feature)
  – Use, for example, information gain to select attribute:

\[
\arg \max_i IG(X_i) = \arg \max_i H(Y) - H(Y \mid X_i)
\]

• Recurs
Suppose we want to predict MPG

Look at all the information gains...
When to stop?

First split looks good! But, when do we stop?
Don’t split a node if all matching records have the same output value.

Base Case
One
Don’t split a node if data points are identical on remaining attributes.

Base Case

Two
Base Cases: An idea

- **Base Case One**: If all records in current data subset have the same output then *don’t recurs*
- **Base Case Two**: If all records have exactly the same set of input attributes then *don’t recurs*

**Proposed Base Case 3:**
If all attributes have small information gain then *don’t recurs*

*•This is not a good idea*
The problem with proposed case 3

\[ y = a \text{ XOR } b \]

The information gains:

<table>
<thead>
<tr>
<th>Input</th>
<th>Value</th>
<th>Distribution</th>
<th>Info Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If we omit proposed case 3:

\[ y = a \text{ XOR } b \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Instead, perform **pruning** after building a tree.

The resulting decision tree:
Non-Boolean Features

• Real-valued features?
Real-> threshold

- Number of thresholds <= # of different values in dataset
- Can choose threshold based on information gain
Summary: Building Decision Trees

BuildTree(\textit{DataSet}, \textit{Output})

- If all output values are the same in \textit{DataSet}, return a leaf node that says “predict this unique output”
- If all input values are the same, return a leaf node that says “predict the majority output”
- Else find attribute $X$ with highest Info Gain
- Suppose $X$ has $n_X$ distinct values (i.e. $X$ has arity $n_X$).
  - Create a non-leaf node with $n_X$ children.
  - The $i$’th child should be built by calling
    \[
    \text{BuildTree}(DS_i, \textit{Output})
    \]
    Where $DS_i$ contains the records in $\textit{DataSet}$ where $X = i$th value of $X$. 
Machine Space Search

- ID3 / C4.5 / CART search for a succinct tree that perfectly fits the data.
- They are not going to find it in general (NP-hard)
The test set error is much worse than the training set error…

...why?
Decision trees will overfit

![Graph showing accuracy vs. size of tree (number of nodes)]
Overfitting

• Precise characterization – statistical learning theory

• Special technics to prevent overfitting in DT learning
  • Pruning the tree, e.g. “reduced error” pruning:
    Do until further pruning is harmful:
    1. Evaluate the impact on validation (test) set of the data of pruning each possible node (and it’s subtree)
    2. Greedily remove one that most improves validation (test) error

• Next lecture: a theoretically sound way to make use of tree heuristics: BOOSTING!