Decision trees

Overview

▶ Decision tree learning
▶ Comparison to NN

Example of decision tree

```plaintext
1: if age ≥ 40 then
2:   if genre = western then
3:     return 4.3
4: else if release date > 1998 then
5:     return 2.5
6:   else
7:     ::
8:   end if
9: else if ... then
10: ::
11: end if
```

General structure of decision trees

▶ Decision tree: nested if-then-else rules
▶ Family of possible if-clauses is pre-determined
  ▶ Typically very simple predicates (e.g., “age is at least 40?”)
▶ Axis-aligned / coordinate splits for numerical features
  ▶ For input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, splits are of the form
    \[ I_{\{x_i > \theta\}} = \begin{cases} 
    1 & \text{if } x_i > \theta \\
    0 & \text{if } x_i \leq \theta 
    \end{cases} \]
  ▶ (Other types of splits are possible.)
Example: iris classification

- Three classes of irises \{1, 2, 3\} (red, green, blue)
- Each input \(x = (x_1, x_2)\) represented by two numerical features
  - \(x_1\) = sepal length-to-width ratio
  - \(x_2\) = petal length-to-width ratio
Example: iris classification

- Three classes of irises \( \{1, 2, 3\} \) (red, green, blue)
- Each input \( x = (x_1, x_2) \) represented by two numerical features
  - \( x_1 \) = sepal length-to-width ratio
  - \( x_2 \) = petal length-to-width ratio

Growing a decision tree

- Leaf nodes form a **partitioning** of the input space
- Prediction to use at each leaf node: plurality label

**Greedy algorithm for decision trees:**
- Start with a single leaf node
- Repeat: pick a leaf node and split into two new leaf nodes
- Rule for picking leaf + split: choose leaf and splitting rule to maximally reduce “uncertainty”

Notions of uncertainty

- Fix attention to single leaf
  - Let \( p_k \) be the proportion of examples reaching a leaf with label \( k \)
- Classification error rate: \( 1 - \max_k p_k \)
- Gini index: \( 1 - \sum_k p_k^2 \)
- Entropy: \( \sum_k p_k \ln \frac{1}{p_k} \)
- Each is maximized when labels are in equal proportion
- Each is minimized when only a single label appears

Figure 1: Uncertainty measures
Overall uncertainty

- (Overall) uncertainty:
  \[ \sum_{\text{leaf } \ell} (\# \text{ training examples reaching } \ell) \cdot (\text{uncertainty at } \ell) \]
- In greedy algorithm, we consider reduction in uncertainty from splitting a leaf

Limits of uncertainty notions

- Figure 2: XOR example

Stopping criterion

- Option 1: Stop when tree reaches pre-specified size
  - Tree size is a hyperparameter
- Option 2: Stop when uncertainty is zero
  - Risk of over-fitting (since training error rate is zero)

Pruning a large tree

- An instantiation of the hold-out approach
- Split training data into \( G \) (grow) and \( P \) (prune)
  - Use \( G \) to grow the tree until zero uncertainty
  - Use \( P \) to choose a good pruning of the tree
- Pruning algorithm:
  - Repeat: replace any non-leaf node by leaf node if it improves error rate with respect to (wrt) \( P \)

- Figure 3: Typical error rate curves
- Figure 4: Pruning a tree
Example: spam filtering I

- Spam dataset
- 4601 email messages, about 39% are spam
- Classify message by spam and not-spam
- 57 features
  - 48 are of the form “percentage of email words that is (WORD)”
  - 6 are of the form “percentage of email characters is (CHAR)”
  - 3 other features (e.g., “longest sequence of all-caps”)
- Final tree after pruning has 17 leaves, 9.3% test error rate

Example: spam filtering II

FIGURE 9.5. The pruned tree for the spam example. The split variables are shown in blue on the branches, and the classification is shown in every node. The numbers under the terminal nodes indicate misclassification rates on the test data.

Comparing $k$-NN and decision trees

- $k$-NN
  - Training/fitting: memorize data set
  - Testing/predicting: find neighbors in memorized data set, output plurality label
- Decision tree
  - Training/fitting: greedily partition feature space to reduce “uncertainty”
  - Testing/prediction: traverse tree, output leaf label