Outline

• Learning decision trees
• Extensions: random forests
round ears
sharp claws
stripes
long tail
dog
cat
dog
cat
Decision Tree Learning

- Greedily choose best decision rule
- Recursively train decision tree for each resulting subset

```plaintext
function fitTree(D, depth)
    if D is all one class or depth >= maxDepth
        node.prediction = most common class in D
        return node
    rule = BestDecisionRule(D)
    dataLeft = {(x, y) from D where rule(D) is true}
    dataRight = {(x, y) from D where rule(D) is false}
    node.left = fitTree(D_left, depth+1)
    node.right = fitTree(D_right, depth+1)
```
function fitTree(D, depth)
    if D is all one class or depth >= maxDepth
        node.prediction = most common class in D
        return node
    rule = BestDecisionRule(D)
dataLeft = {(x, y) from D where rule(D) is true}
dataRight = {(x, y) from D where rule(D) is false}
node.left = fitTree(D_left, depth+1)
node.right = fitTree(D_right, depth+1)
Choosing Decision Rules

• Define a cost function $\text{cost}(D)$
  • Misclassification rate
  • Entropy or information gain
  • Gini index
Misclassification Rate

\[ \hat{\pi}_c := \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \mathbb{I}(y_i = c) \]  

class proportion  
(estimated probability)

\[ \hat{y} := \arg\max_c \hat{\pi}_c \]  
best prediction

\[ \text{cost}(\mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \mathbb{I}(y_i \neq \hat{y}) = 1 - \hat{\pi}_{\hat{y}} \]  
error rate

\[ \text{cost}(\mathcal{D}) - \left( \frac{|\mathcal{D}_L|}{|\mathcal{D}|} \text{cost}(\mathcal{D}_L) + \frac{|\mathcal{D}_R|}{|\mathcal{D}|} \text{cost}(\mathcal{D}_R) \right) \]  

cost reduction
Entropy and Information Gain

\[ \hat{\pi}_c := \frac{1}{|D|} \sum_{i \in D} \mathbb{I}(y_i = c) \]

\[ H(\hat{\pi}) := -\sum_{c=1}^{C} \hat{\pi}_c \log \hat{\pi}_c \]

\[ \text{infoGain}(j) = H(Y) - H(Y|X_j) \]

\[ = -\sum_y \Pr(Y = y) \log \Pr(Y = y) + \sum_{x_j} \Pr(X_j = x_j) \sum_y \Pr(Y = y|X_j = x_j) \log \Pr(Y = y|X_j = x_j). \]

\[ \text{cost}(D) = \left( \frac{|D_L|}{|D|} \text{cost}(D_L) + \frac{|D_R|}{|D|} \text{cost}(D_R) \right) \]
Information Gain

\[ \text{infoGain}(j) = H(Y) - H(Y|X_j) \]

\[ = - \sum_y \Pr(Y = y) \log \Pr(Y = y) + \]

\[ \sum_{X_j=x_j} \Pr(Y = y|X_j = x_j) \sum_y \Pr(Y = y|X_j = x_j) \log \Pr(Y = y|X_j = x_j). \]

\[ X_j = Y \quad \quad \quad X_j \perp Y \]
Gini Index

\[
\sum_{c=1}^{C} \hat{\pi}_c (1 - \hat{\pi}_c) = \sum_{c} \hat{\pi}_c - \sum_{c} \hat{\pi}_c^2 = 1 - \sum_{c} \hat{\pi}_c^2
\]

like misclassification rate, but accounts for uncertainty
Comparing the Metrics

% Fig 9.3 from Hastie book

\[ p = 0 : 0.01 : 1; \]
\[ \text{gini} = 2 \times p \times (1 - p); \]
\[ \text{entropy} = -p \times \log(p) - (1 - p) \times \log(1 - p); \]
\[ \text{err} = 1 - \max(p, 1 - p); \]

% scale to pass through (0.5, 0.5)
\[ \text{entropy} = \frac{\text{entropy}}{\max(\text{entropy})} \times 0.5; \]

figure;
plot(p, err, 'g-', p, gini, 'b:', p, ...
   entropy, 'r--', 'linewidth', 3);
legend('Error rate', 'Gini', 'Entropy')
Overfitting

• A decision tree can achieve 100% training accuracy when each example is unique

• Limit depth of tree

• Strategy: train very deep tree
  • Adaptively prune
Pruning with Validation Set

Validation accuracy: 0.4
Pruning with Validation Set

Validation accuracy: 0.4
new validation accuracy: 0.41
Random Forests

- Use **bootstrap aggregation** to train many decision trees
  - Randomly subsample $n$ examples
  - Train decision tree on subsample
  - Use average or majority vote among learned trees as prediction
  - Also randomly subsample features
  - Reduces variance without changing bias
Summary

• Training decision trees
• Cost functions
  • Misclassification
  • Entropy and information gain
  • Gini index (expected error)
• Pruning
• Random forests (bagging)