Ensemble learning
Lecture 13

David Sontag
New York University

Slides adapted from Navneet Goyal, Tan, Steinbach, Kumar, Vibhav Gogate
Ensemble methods

Machine learning competition with a $1 million prize
Reduce Variance Without Increasing Bias

- **Averaging** reduces variance:

  \[ Var(\overline{X}) = \frac{Var(X)}{N} \]
  
  (when predictions are independent)

Average models to reduce model variance

One problem:
  - only one training set
  - where do multiple models come from?
Bagging: Bootstrap Aggregation

- Leo Breiman (1994)
- Take repeated bootstrap samples from training set $D$
- **Bootstrap sampling**: Given set $D$ containing $N$ training examples, create $D'$ by drawing $N$ examples at random with replacement from $D$.

- Bagging:
  - Create $k$ bootstrap samples $D_1 \ldots D_k$.
  - Train distinct classifier on each $D_i$.
  - Classify new instance by majority vote / average.
General Idea

Step 1: Create Multiple Data Sets

Step 2: Build Multiple Classifiers

Step 3: Combine Classifiers

Original Training data

C^*
Bagging

- Sampling with replacement

<table>
<thead>
<tr>
<th>Data ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bagging (Round 1)</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Bagging (Round 2)</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>7</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Bagging (Round 3)</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

- Build classifier on each bootstrap sample
- Each data point has probability \((1 - 1/n)^n\) of being selected as test data
- Training data = 1- \((1 - 1/n)^n\) of the original data
The 0.632 bootstrap

• This method is also called the 0.632 bootstrap
  – A particular training data has a probability of $1-1/n$ of not being picked
  – Thus its probability of ending up in the test data (not selected) is:

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} = 0.368$$

  – This means the training data will contain approximately 63.2% of the instances
Bagging Example
CART decision boundary

decision tree learning algorithm; very similar to ID3
100 bagged trees

shades of blue/red indicate strength of vote for particular classification
Example of Bagging

Assume that the training data is:

Goal: find a collection of 10 simple thresholding classifiers that collectively can classify correctly.

- Each simple (or weak) classifier is:
  
  \( x \leq K \rightarrow \text{class} = +1 \) or \(-1\) depending on which value yields the lowest error; where \( K \) is determined by entropy minimization.
Random Forests

• Ensemble method specifically designed for decision tree classifiers

• Introduce two sources of randomness: “Bagging” and “Random input vectors”
  – **Bagging method**: each tree is grown using a bootstrap sample of training data
  – **Random vector method**: At each node, best split is chosen from a random sample of $m$ attributes instead of all attributes
Random Forests

Figure 5.40. Random forests.
Methods for Growing the Trees

• Fix a \( m \leq M \). At each node
  – Method 1:
    • Choose \( m \) attributes randomly, compute their information gains, and choose the attribute with the largest gain to split
  – Method 2:
    • (When \( M \) is not very large): select \( L \) of the attributes randomly. Compute a linear combination of the \( L \) attributes using weights generated from \([-1,+1]\) randomly. That is, new \( A = \sum(W_i*A_i), \text{i}=1..L\).
  – Method 3:
    • Compute the information gain of all \( M \) attributes. Select the top \( m \) attributes by information gain. Randomly select one of the \( m \) attributes as the splitting node.
Random Forest Algorithm: method 1 in previous slide

1. For $b = 1$ to $B$:
   
   (a) Draw a bootstrap sample $Z^*$ of size $N$ from the training data.
   
   (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.
      
      i. Select $m$ variables at random from the $p$ variables.
      
      ii. Pick the best variable/split-point among the $m$.
      
      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_{1}^{B}$.

To make a prediction at a new point $x$:

Regression: $\hat{f}_{rf}^{B}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the $b$th random-forest tree. Then $\hat{C}_{rf}^{B}(x) = majority\ vote\ \{\hat{C}_b(x)\}_{1}^{B}$. 
Reduce Bias$^2$ and Decrease Variance?

- Bagging reduces variance by averaging
- Bagging has little effect on bias
- Can we average *and* reduce bias?
- Yes:
  - Boosting