CPSC 425: Computer Vision

Lecture 20: Image Classification (part 4)
Menu for Today (March 24, 2020)

Topics:
- Decision Tree
- Boosting

Readings:
- **Today’s** Lecture: Forsyth & Ponce (2nd ed.) 16.1.3, 16.1.4, 16.1.9
- **Next** Lecture: Forsyth & Ponce (2nd ed.) 17.1–17.2

Reminders:
- **Assignment 5** is due Tuesday, March 31st
- Midterm *review*
- Final structure / grading announced
A decision tree is a simple non-linear parametric classifier.

Consists of a tree in which each internal node is associated with a feature test.

A data point starts at the root and recursively proceeds to the child node determined by the feature test, until it reaches a leaf node.

The leaf node stores a class label or a probability distribution over class labels.
Learning a decision tree from a training set involves selecting an efficient sequence of feature tests

**Example:** Waiting for a restaurant table

<table>
<thead>
<tr>
<th>Example</th>
<th>Attr.</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alt Bar Fri Hun Pat Price Rain Res Type Est WillWait</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X₁</td>
<td>T F F T Some $$$ F T French 0–10 T</td>
<td></td>
</tr>
<tr>
<td>X₂</td>
<td>T F F T Full $ F F Thai 30–60 F</td>
<td></td>
</tr>
<tr>
<td>X₃</td>
<td>F T F F Some $ F F Burger 0–10 T</td>
<td></td>
</tr>
<tr>
<td>X₄</td>
<td>T F T T Full $ F F Thai 10–30 T</td>
<td></td>
</tr>
<tr>
<td>X₅</td>
<td>T F T F Full $$$ F T French &gt;60 F</td>
<td></td>
</tr>
<tr>
<td>X₆</td>
<td>F T F T Some $$ T T Italian 0–10 T</td>
<td></td>
</tr>
<tr>
<td>X₇</td>
<td>F T F F None $ T F Burger 0–10 F</td>
<td></td>
</tr>
<tr>
<td>X₈</td>
<td>F F F T Some $$ T T Thai 0–10 T</td>
<td></td>
</tr>
<tr>
<td>X₉</td>
<td>F T T F Full $ T F Burger &gt;60 F</td>
<td></td>
</tr>
<tr>
<td>X₁₀</td>
<td>T T T T Full $$$ F T Italian 10–30 F</td>
<td></td>
</tr>
<tr>
<td>X₁₁</td>
<td>F F F F None $ F F Thai 0–10 F</td>
<td></td>
</tr>
<tr>
<td>X₁₂</td>
<td>T T T T Full $ F F Burger 30–60 T</td>
<td></td>
</tr>
</tbody>
</table>
The entropy of a set $S$ of data samples is defined as

$$H(S) = - \sum_{c \in C} p(c) \log(p(c))$$

where $C$ is the set of classes represented in $S$, and $p(c)$ is the empirical distribution of class $c$ in $S$

Entropy is highest when data samples are spread equally across all classes, and zero when all data samples are from the same class.
In general we try to select the feature test that maximizes the information gain:

\[ I = H(S) - \sum_{i \in \{\text{children}\}} \frac{|S^i|}{|S|} H(S^i) \]

In the previous example, the information gains of the two candidate tests are:

\[ I_{\text{Patrons}} = 0.541 \quad I_{\text{Type}} = 0 \]

So we choose the ‘Patrons’ test.
Lecture 19: Re-cap — Decision Tree

Following this construction procedure we obtain the final decision tree:

![Decision Tree Diagram]

Figure credit: Russell and Norvig (3rd ed.)
**Decision Tree**

A *random forest* is an ensemble of decision trees.

Randomness is incorporated via training set sampling and/or generation of the candidate binary tests.

The prediction of the random forest is obtained by averaging over all decision trees.

Forsyth & Ponce (2nd ed.) Figure 14.19. Original credit: J. Shotton et al., 2011
Example 1: Kinect

Kinect allows users of Microsoft’s Xbox 360 console to interact with games using natural body motions instead of a traditional handheld controller. The pose (joint positions) of the user is predicted using a random forest trained on depth features.

Figure credit: J. Shotton et al., 2011
Example 1: Kinect

Kinect allows users of Microsoft’s Xbox 360 console to interact with games using natural body motions instead of a traditional handheld controller. The pose (joint positions) of the user is predicted using a random forest trained on depth features.

Figure credit: J. Shotton et al., 2011
Example 1: Kinect

\[ f_\theta(I, x) = d_I \left( x + \frac{u}{d_I(x)} \right) - d_I \left( x + \frac{v}{d_I(x)} \right) \]

**Figure credit:** J. Shotton et al., 2011
Example 1: Kinect

Figure credit: J. Shotton et al., 2011
Combining **Classifiers**

One common strategy to obtain a better classifier is to combine multiple classifiers.

A simple approach is to train an ensemble of independent classifiers, and average their predictions.

**Boosting** is another approach.

– Train an ensemble of classifiers sequentially.
– Bias subsequent classifiers to correctly predict training examples that previous classifiers got wrong.
– The final boosted classifier is a weighted combination of the individual classifiers.
Combining Classifiers: **Boosting**
Combining Classifiers: **Boosting**

*Figure credit: Paul Viola*
Combining Classifiers: **Boosting**

Figure credit: Paul Viola
Combining Classifiers: **Boosting**

**Figure credit:** Paul Viola
Combining Classifiers: **Boosting**

Figure credit: Paul Viola
Combining Classifiers: **Boosting**

Final classifier is a combination of weak classifiers

*Figure credit: Paul Viola*
Summary

A **decision tree** passes a data point through a sequence of feature tests. A random forest is an ensemble of decision trees.

Factors that make image classification hard
— intra-class variation, viewpoint, illumination, clutter, occlusion...

A codebook of **visual words** contains representative local patch descriptors — can be constructed by clustering local descriptors (e.g. SIFT) in training images

The **bag of words** model accumulates a histogram of occurrences of each visual word

The **spatial pyramid** partitions the image and counts visual words within each grid box; this is repeated at multiple levels