CPSC 425: Computer Vision

Instructor: Jim Little
little@cs.ubc.ca

Department of Computer Science
University of British Columbia

Lecture Notes 2016/2017 Term 2
Menu March 23, 2017

Topics:
  Classification (cont.)
  Image Classification

Reading:
  Today: Forsyth & Ponce (2nd ed.) 16.1.3–16.1.4
  Next: Forsyth & Ponce (2nd ed.) 17.1–17.2

Reminders:
  Assignment 6 due Tuesday, March 28
  www: http://www.cs.ubc.ca/~little/cpsc425/
Today’s “Fun” Example: Eulerian Video Magnification

Point your browser to Erik Olsen’s February, 27, 2013, NY Times article

Scientists Uncover Invisible Motion in Video

and view the embedded video

You can also go to Miki Rubinstein’s MIT CSAIL project page where you can download, among other things, the SIGGRAPH 2012 paper, additional videos, and Matlab code that reproduces all the results in the paper
Lecture 19: Re-cap

- Classifiers take as input a set of features and output (predict) a class label.

- Classifiers need to take into account “loss” associated with each kind of classification error.

- A receiver operating characteristic (ROC) curve plots the trade-off between false negatives and false positives.

- Non-parametric classifiers, like k-nearest neighbour, are data driven. New data points are classified by comparing to the training examples directly.

- Parametric classifiers, like support vector machines, are model driven. New data points are classified by evaluating a model learned from the training examples.
Decision Trees

- 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
## Decision Trees

- 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>Attributes</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alt</td>
<td>Bar</td>
<td>Fri</td>
</tr>
<tr>
<td>X₁</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>X₂</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>X₃</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>X₄</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>X₅</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>X₆</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>X₇</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>X₈</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>X₉</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>X₁₀</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>X₁₁</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>X₁₂</td>
<td>I</td>
<td>I</td>
<td>I</td>
</tr>
</tbody>
</table>
Decision Trees

Which test is more helpful?

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

- The **entropy** of a set $S$ of data samples is defined as

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

(1)

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:

$$I = H(S) - \sum_{i \in \{\text{children}\}} \frac{|S^i|}{|S|} H(S^i)$$  \hspace{1cm} (2)

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

- $I_{\text{Patrons}} = 0.541$
- $I_{\text{Type}} = 0$

So we choose the ‘Patrons’ test.
Following this construction procedure we obtain the final decision tree:
Random Forests

- 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
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

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

Figure credit: Paul Viola
Combining Classifiers: Boosting

Final classifier is a combination of weak classifiers

Figure credit: Paul Viola
We next discuss image classification, where we pass a whole image into a classifier and obtain a class label as output.
What Makes Image Classification Hard?

Intra-class variation, viewpoint, illumination, clutter, and occlusion (among others!)

Figure source: Jianxiong Xiao. Original credit: ?
Image Classification

In addition to images containing single objects, the same techniques can be applied to classify natural scenes (e.g. beach, forest, harbour, library).

Why might classifying scenes be useful?
In addition to images containing single objects, the same techniques can be applied to classify natural scenes (e.g. beach, forest, harbour, library).

Why might classifying scenes be useful?

Visual perception is influenced by expectation. Our expectations are often conditioned on the context.
What Is This Object?

Figure source: Jianxiong Xiao
What Is This Object?

Figure source: Jianxiong Xiao
What Is This Object?

Figure source: Jianxiong Xiao
What Is This Object?

Figure source: Jianxiong Xiao
What Is This Object?

Figure source: Jianxiong Xiao
What Is This Object?

Look-Alikes by Joan Steiner

Figure source: Jianxiong Xiao
Visual Words

- Many algorithms for image classification accumulate evidence on the basis of visual words.

- To classify a text document (e.g. as an article on sports, entertainment, business, politics) we might find patterns in the occurrences of certain words.
Visual Words

- In images, the equivalent of a word is a local image patch. The local image patch is described using a descriptor such as SIFT.

- We construct a **vocabulary** or **codebook** of local descriptors, containing representative local descriptors.

- Question: How might we construct such a codebook? Given a large sample of SIFT descriptors, say 1 million, how can we choose a small number of ‘representative’ SIFT codewords, say 1000?
Visual Words

Fig. 16.6 in Forsyth & Ponce (2nd ed). Original credit: Sivic and Zisserman, 2008.
Bag of Words Representation

- Given a codebook containing $K$ codewords, the bag of words representation of an image simply counts the number of times each codeword occurs, and summarizes the result in a histogram.

Figure credit: Rob Fergus
Bag of Words Representation

Algorithm:

Initialize an empty $K$-bin histogram, where $K$ is the number of codewords
Extract local descriptors (e.g. SIFT) from the image
For each local descriptor $x$
    Map (Quantize) $x$ to its closest codeword $\rightarrow c(x)$
    Increment the histogram bin for $c(x)$
Return histogram

We can then classify the histogram using a trained classifier, e.g. a support vector machine.
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.