CS544: Classification Algorithms

February 2, 2012

Zornitsa Kozareva
USC/ISI
Marina del Rey, CA
kozareva@isi.edu
www.isi.edu/~kozareva

Today

• Named Entity Recognition

• Multi-class classification
  – Decision trees
  – $k$ Nearest Neighbor

• Binary classification
  – Perceptron
Named Entity Recognition

Adam Smith works for IBM, London since February 2010.

- Identify mentions in text and classify them into a predefined set of categories of interest:
  - Person: Adam Smith
  - Organizations: IBM
  - Locations: London
  - Date: February 2010
Types of Machine Learning

- **Supervised Learning**
  - labeled training examples with correct responses (targets) are provided
  - based on the training set, the algorithm *generalizes* to respond correctly to all possible inputs

- **(Some) Methods:**
  - Hidden Markov Models, k-Nearest Neighbors, Decision Trees, AdaBoost, SVM

- **NLP Tasks:**
  - Named Entity recognition, POS tagging, Parsing

---

Types of Machine Learning

- **Unsupervised Learning**
  - correct responses (targets) are not provided
  - the algorithm identifies similarities between the inputs based on something in common

- **Method:**
  - Clustering

- **NLP Tasks:**
  - Named Entity Disambiguation, Text Categorization
Types of Machine Learning

• Semi-Supervised Learning
  – small percentage of labeled examples with correct responses are provided, the rest are unlabeled
  – label the unlabeled examples using the labeled ones, add the newly labeled data to the training data set

• Method:
  – Co-training, self-training, active learning

• NLP Tasks:
  – Named Entity Recognition, POS-tagging, Parsing

Multi-Class Classification (Example)

• Named Entity Recognition
  – person, organization, location, miscellaneous name

• Text Categorization by Topic
  – economy, sport, entertainment

• Weather Forecast
  – sunny, foggy, snowy, rainy

• Author Identification
Muti-Class Classification

• **Given**: some data items that belong to one of $N$ possible classes

• **Task**: train a classifier to predict the class for a new data item

• Geometrically: hard
(Some) Multi-class Classification Algorithms

- Linear
  - Decision trees
  - Naïve Bayes

- Non Linear
  - K-nearest neighbors
  - Neural Networks
Things Students Enjoy Doing

✓ going to pub
✓ watching TV
✓ going to a party
✓ Studying

Build an algorithm that will let you decide what to do each evening without having to think about it every night?

• If you have an assignment due next day, you need to study
• If you feel lazy, the you don’t like going to the pub
• If there is no party, you cannot go to it

Decision Trees

• The classifier has a tree structure, where each node is either:
  – a leaf indicating the value of the target attribute (class) of examples
  – a decision specifying some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test

• An instance $x_v$ is classified by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance
Decision Tree on How to Spend the Evening

Constructing Decision Trees

- Build a tree in a greedy manner starting at the root
- Choose the most informative feature at each step by computing the entropy
  \[ H(p) = -\sum_i p_i \log_2 p_i \]
- Estimate how much the entropy of the whole training set would decrease if a particular feature is chosen for the next classification step

\[ \text{Gain}(S,F) = \text{Entropy}(S) - \sum_{f \in \text{values}(F)} \frac{|S_f|}{|S|} \text{Entropy}(S_f) \]
### Walkthrough Example

**Set of Examples (S) | Feature (f1) | Feature (f2) | Feature (f3) | Outcome**
---|---|---|---|---
| s1 | 0 | 1 | 0 | True |
| s2 | 0 | 1 | 0 | False |
| s3 | 0 | 0 | 1 | False |
| s4 | 1 | 0 | 0 | False |

\[
\text{Entropy}(S) = -p_{true} \log_2 p_{true} - p_{false} \log_2 p_{false}
\]

\[
= -\frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4}
\]

\[
= 0.5 + 0.311 = 0.811
\]

---

### Walkthrough Example

**Set of Examples (S) | Feature (f1) | Feature (f2) | Feature (f3) | Outcome**
---|---|---|---|---
| s1 | 0 | 1 | 0 | True |
| s2 | 0 | 1 | 0 | False |
| s3 | 0 | 0 | 1 | False |
| s4 | 1 | 0 | 0 | False |

\[
\frac{|S_f|}{|S|} \text{Entropy}(S_f) = \frac{1}{4} \times \left( -\frac{0}{1} \log_2 \frac{0}{1} - \frac{1}{1} \log_2 \frac{1}{1} \right) = 0
\]

\[
\frac{|S_f|}{|S|} \text{Entropy}(S_f) = \frac{2}{4} \times \left( -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} \right) = \frac{1}{2}
\]

\[
\frac{|S_f|}{|S|} \text{Entropy}(S_f) = \frac{1}{4} \times \left( -\frac{0}{1} \log_2 \frac{0}{1} - \frac{1}{1} \log_2 \frac{1}{1} \right) = 0
\]

\[
\text{Gain}(S,F) = \text{Entropy}(S) - \sum_{f \in \text{value}(F)} \frac{|S_f|}{|S|} \text{Entropy}(S_f)
\]

---

2/2/12
Walkthrough Example

<table>
<thead>
<tr>
<th>Set of Examples (S)</th>
<th>Feature (f1)</th>
<th>Feature (f2)</th>
<th>Feature (f3)</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>True</td>
</tr>
<tr>
<td>s2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>False</td>
</tr>
<tr>
<td>s3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>False</td>
</tr>
<tr>
<td>s4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>False</td>
</tr>
</tbody>
</table>

Gain($S, F$) = $0.811 - (0 + 0.5 + 0) = 0.311$

Another Classification Example

- List everything that you have done for the past few days to get a decent dataset

<table>
<thead>
<tr>
<th>Deadline?</th>
<th>Is there a party?</th>
<th>Lazy?</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urgent</td>
<td>Yes</td>
<td>Yes</td>
<td>Party</td>
</tr>
<tr>
<td>Urgent</td>
<td>No</td>
<td>Yes</td>
<td>Study</td>
</tr>
<tr>
<td>Near</td>
<td>Yes</td>
<td>Yes</td>
<td>Party</td>
</tr>
<tr>
<td>None</td>
<td>Yes</td>
<td>No</td>
<td>Party</td>
</tr>
<tr>
<td>None</td>
<td>No</td>
<td>Yes</td>
<td>Pub</td>
</tr>
<tr>
<td>None</td>
<td>Yes</td>
<td>No</td>
<td>Party</td>
</tr>
<tr>
<td>Near</td>
<td>No</td>
<td>No</td>
<td>Study</td>
</tr>
<tr>
<td>Near</td>
<td>No</td>
<td>Yes</td>
<td>TV</td>
</tr>
<tr>
<td>Near</td>
<td>Yes</td>
<td>Yes</td>
<td>Party</td>
</tr>
<tr>
<td>Urgent</td>
<td>No</td>
<td>No</td>
<td>Study</td>
</tr>
</tbody>
</table>
Decision Trees

**Pros**
+ generate understandable rules
+ provide a clear indication of which features are most important for classification

**Cons**
- error prone in multi-class classification and small number of training examples
- computationally expensive to train (need to compare all possible splits; and also because of pruning)

$O(N \log N)$ tree construction
$O(\log N)$ to return particular leaf

Non Linear (ex: $k$ Nearest Neighbor)
**k Nearest Neighbor**

- Classification rule:
  - to classify a new object, find the object in the training set that is most similar
  - then assign the class of this neighbor to the new object

- **k Nearest Neighbor:**
  - consult k nearest neighbors
  - decision based on majority category of the neighbor
3-Nearest Neighbor

choose the category of the majority of the neighbors

but wait, this one is closer

4-Nearest Neighbor?

It is good for the value of k to be odd to avoid ties
**k Nearest Neighbor Algorithm**

- Learning is just storing the representations of the training examples.

- Testing instance $x_0$:
  - compute similarity between $x_0$ and all training examples
  - take vote among $x_0$, $k$ nearest neighbours
  - assign $x_0$ with the category of the most similar example in $T$

---

**Similarity Computation**

- Nearest neighbor method uses similarity (or distance) metric.

- Given two objects $x$ and $y$ both with $n$ values

\[
x = (x_1, x_2, \ldots, x_n)
\]

\[
y = (y_1, y_2, \ldots, y_n)
\]

calculate the Euclidean distance as

\[
d(x, y) = \sqrt{\sum_{i=1}^{n} [x_i - y_i]^2}
\]
An Example

<table>
<thead>
<tr>
<th></th>
<th>isPersonName</th>
<th>isCapitalized</th>
<th>isLiving</th>
<th>teachesCSS44</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zornitsa Kozareva</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>USC</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>no</td>
</tr>
<tr>
<td>eduard hovy</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>and</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>no</td>
</tr>
</tbody>
</table>

\[
d(ZornitsaKozareva, USC) = \sqrt{(1^2 + 0 + 1^2)} = 1.41
\]

\[
d(ZornitsaKozareva, eduardhovy) = \sqrt{(0 + 1^2 + 0)} = 1
\]

\[
d(ZornitsaKozareva, and) = \sqrt{(1+1+1)} = 1.73
\]

k Nearest Neighbours

**Pros**
+ robust
+ simple
+ training is very fast (storing examples)

**Cons**
- depends on similarity measure & k-NNs
- easily fooled by irrelevant attributes
- computationally expensive
Next Couple of Lectures

- Perceptron
- Putting Machine Learning into practice - NER
- Types of Features and Feature Generation
- Semi-Supervised Algorithms
- Introduction to Weka
- Homework #2