CS544: Introduction to Weka

January 29, 2013

Zornitsa Kozareva
USC/ISI
Marina del Rey, CA
kozareva@isi.edu
www.isi.edu/~kozareva
WEKA
Waikato Environment for Knowledge Analysis
Weka: Data Mining Software

• Collection of machine learning algorithms
  – open-source package written in Java

• Used for research, education and application

• Main features:
  – data pre-processing tools
  – learning algorithms
  – evaluation methods
  – graphical inference
  – environment for comparing learning algorithms
Weka: Data Mining Software

• Classification algorithms:
  – decision trees, kNN, SVM, Naive-bayes

• Prediction algorithms:
  – regression (linear/SVM), perceptron

• Meta-algorithms:
  – bagging, boosting (AdaBoost)

among others
Getting Started

• Install Weka software (on Linux):
  – Download link:
    • [http://prdownloads.sourceforge.net/weka/weka-3-6-2.zip](http://prdownloads.sourceforge.net/weka/weka-3-6-2.zip)
    • Unzip the software
  – Requirement: Java 1.5 (or higher)
  – Invoke Weka command:
    • `java -cp weka.jar <weka-command>`
Weka GUI Chooser

```
java -Xmx1000M -jar weka.jar
```
Data File Format (.arff)

@relation named_entity

@attribute position numeric
@attribute pos_tag { NN, NP, VB, DT}
@attribute word_length numeric
@attribute in_gazetteer { no, yes}
@attribute class { B-PER, I-PER, B-LOC, I-LOC, O}

@data
3, DT, 3, no, B-ORG
4, NP, 10, yes, I-ORG
15, NP, 6, yes, O
7, NN, 12, ?, B-PER
...

Other attribute types:
• String
• Date

Missing value
The Preprocessing Tab

- **Classification Filter selection**
- **Preprocessing**
- **Statistical attribute selection**

### List of attributes
(last: class variable)

- Attributes:
  - Name
  - years
  - wen
  - work
  - works
  - world
  - worried
  - worst
  - worth
  - wrong
  - wrote
  - yawn
  - year
  - years
  - young
  - ysaart
  - zeepuik
  - zimmer
  - zimmerman
  - zmolek
  - class

### Frequency and categories for the selected attribute

- Statistics about the values of the selected attribute
- Statistics:
  - Minimum
  - Maximum
  - Mean
  - StdDev

### Preprocessing Tab

- Open file...
- Open URL...
- Open DB...
- Undo
- Save...

Slide adapted from Marti Hearst
The attribute whose value is to be predicted from the values of the remaining ones. Default is the last attribute.

Cross-validation: split the data into e.g.
10 folds and 10 times train on 9 folds and test on the remaining one.
Slide adapted from Marti Hearst
all other numbers can be obtained from it
different/easy class
accuracy

### Classifier output

Time taken to build model: 0.07 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances 41 68.3333%
Incorrectly Classified Instances 19 31.6667%
Kappa statistic 0.525
Mean absolute error 0.2062
Root mean squared error 0.4493
Relative absolute error 46.4007%
Root relative squared error 95.3122%
Total Number of Instances 60

=== Detailed Accuracy By Class ===

<table>
<thead>
<tr>
<th>TP Rate</th>
<th>FF Rate</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0.3</td>
<td>0.556</td>
<td>0.75</td>
<td>0.638</td>
<td>misc.forsale</td>
</tr>
<tr>
<td>0.7</td>
<td>0.025</td>
<td><strong>0.933</strong></td>
<td>0.7</td>
<td><strong>0.8</strong></td>
<td>rec.sport.hockey</td>
</tr>
<tr>
<td>0.6</td>
<td>0.15</td>
<td>0.667</td>
<td>0.6</td>
<td>0.632</td>
<td>comp.graphics</td>
</tr>
</tbody>
</table>

=== Confusion Matrix ===

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>--- classified as</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>15</td>
<td>1</td>
<td>4</td>
<td>misc.forsale</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
<td>14</td>
<td>2</td>
<td>rec.sport.hockey</td>
</tr>
<tr>
<td>c</td>
<td>8</td>
<td>0</td>
<td>12</td>
<td>comp.graphics</td>
</tr>
</tbody>
</table>
Correctly Classified Instances

Incorrectly Classified Instances

Kappa statistic

Mean absolute error

Root mean squared error

Relative absolute error

Root relative squared error

Total Number of Instances

=== Detailed Accuracy by Class ===

TP Rate  FP Rate  Precision  Recall  F-Measure  Class
0.995  0.321  0.756  0.995  0.756  newsgroups
0.679  0.005  0.993  0.679  0.993  rec.motorcycles

=== Confusion Matrix ===

a  b  <-- classified as
396  2  |  a = rec.motorcycles
128 271  |  b = rec.sport.hockey
WEKA
Command Line
Weka Specifications

- Train classifier on training data and output model
  - java -cp weka.jar <classifier-function> -t <train-file> -d <trained-model>
- Run trained classifier model on test data
  - java -cp weka.jar <classifier-function> -T <test-file> -I <trained-model>
- Specifying parameters:
  -t : training file (.arff)
  -T : test file (.arff)
  -d : output filename (trained classifier model)
  -I : input model (for testing)
  -K : number of nearest neighbors for kNN algorithm
  -h : help (check out other parameter options, etc.)
Example: kNN in Weka

• Train a classifier using 2NN algorithm
  • java -cp weka.jar
    
    weka.classifiers.lazy.IBk
    -t data/weather.arff
    -K 2
    -d model.2nn

  Classifier-function in weka
  Training file
  Algorithm parameter
  Output model name

• Run the trained classifier on test data
  • java -cp weka.jar
    
    weka.classifiers.lazy.IBk
    -T data/weather.arff
    -l model.2nn

  Classifier-function in weka
  Test file
  Input model name
Sample Weka Output

--- Error on test data ---

<table>
<thead>
<tr>
<th>Correctly Classified Instances</th>
<th>13</th>
<th>92.8571 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrectly Classified Instances</td>
<td>1</td>
<td>7.1429 %</td>
</tr>
<tr>
<td>Kappa statistic</td>
<td>0.8372</td>
<td></td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.1333</td>
<td></td>
</tr>
<tr>
<td>Root mean squared error</td>
<td>0.2333</td>
<td></td>
</tr>
<tr>
<td>Total Number of Instances</td>
<td>14</td>
<td></td>
</tr>
</tbody>
</table>
More Detailed Output

- Classification labels for each instance (use “-p 1” option)

```java
java -cp weka.jar weka.classifiers.lazy.lbk -T data/weather.arff -l model.2nn -p 1
```

### Predictions on test data

<table>
<thead>
<tr>
<th>inst#</th>
<th>actual</th>
<th>predicted</th>
<th>error</th>
<th>prediction (outlook)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>sunny</td>
</tr>
<tr>
<td>2</td>
<td>2:no</td>
<td>1:yes</td>
<td>0.5</td>
<td>sunny</td>
</tr>
<tr>
<td>3</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>4</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>5</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>6</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>7</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>8</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>sunny</td>
</tr>
<tr>
<td>9</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.5</td>
<td>sunny</td>
</tr>
<tr>
<td>10</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>rainy</td>
</tr>
<tr>
<td>11</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.5</td>
<td>sunny</td>
</tr>
<tr>
<td>12</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>13</td>
<td>1:yes</td>
<td>1:yes</td>
<td>0.967</td>
<td>overcast</td>
</tr>
<tr>
<td>14</td>
<td>2:no</td>
<td>2:no</td>
<td>0.967</td>
<td>rainy</td>
</tr>
</tbody>
</table>
Weka Classification Functions

- kNN: `weka.classifiers.lazy.Ibk`
- Decision trees: `weka.classifiers.trees.J48`
- Naïve Bayes: `weka.classifiers.bayes.NaiveBayes`
- AdaBoost: `weka.classifiers.meta.AdaBoostM1`
Additional Information

• General documentation:
  http://www.cs.waikato.ac.nz/ml/weka/
  http://prdownloads.sourceforge.net/weka/weka.ppt

• Command line doc:
  http://weka.wikispaces.com/Primer
Assignment #1
Named Entity Recognition
• **Given**: a train and development data sets of English sentences tagged with the classes:
  – B-PER, I-PER (people)
  – B-ORG, I-ORG (organization)
  – B-LOC, I-LOC (location)
  – B-MISC, I-MISC (miscellaneous)
  – O (outside, meaning not a named entity class)

• **Your objective is**: to develop a machine learning NE system, which when given a new previously unseen text (i.e. test set) will identify and classify the named entities correctly
Data Description

- The data consists of three columns separated by a single space. Each word has been put on a separate line and there is an empty line after each sentence.

B-TYPE means the word is the beginning of a phrase of type TYPE
O means the word is not part of a phrase

Make sure to preserve the empty lines in the output of the test data
Results for NE Detection

<table>
<thead>
<tr>
<th>Carreras et al.,2002</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIO dev.</td>
<td>92.45</td>
<td>90.88</td>
<td>91.66</td>
</tr>
</tbody>
</table>

CoNLL-2002 Spanish Evaluation Data

<table>
<thead>
<tr>
<th>Data sets</th>
<th>#tokens</th>
<th>#NEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>264,715</td>
<td>18,794</td>
</tr>
<tr>
<td>Development</td>
<td>52,923</td>
<td>4,351</td>
</tr>
<tr>
<td>Test</td>
<td>51,533</td>
<td>3,558</td>
</tr>
</tbody>
</table>

Evaluation Measures

\[
\text{Precision} = \frac{\# \text{correct identified NEs}}{\# \text{identified NEs}}
\]

\[
\text{Recall} = \frac{\# \text{correct identified NEs}}{\# \text{gold standard data}}
\]
Results for NE Detection

<table>
<thead>
<tr>
<th>Carreras et al.,2002</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIO dev.</td>
<td>92.45</td>
<td>90.88</td>
<td>91.66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CoNLL-2002 Spanish Evaluation Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data sets</td>
</tr>
<tr>
<td>Train</td>
</tr>
<tr>
<td>Development</td>
</tr>
<tr>
<td>Test</td>
</tr>
</tbody>
</table>
## Results for NE Classification*

<table>
<thead>
<tr>
<th>Spanish Dev.</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>79.04</td>
<td>80.00</td>
<td>79.52</td>
</tr>
<tr>
<td>MISC</td>
<td>55.48</td>
<td>54.61</td>
<td>55.04</td>
</tr>
<tr>
<td>ORG</td>
<td>79.57</td>
<td>76.06</td>
<td>77.77</td>
</tr>
<tr>
<td>PER</td>
<td>87.19</td>
<td>86.91</td>
<td>87.05</td>
</tr>
<tr>
<td>overall</td>
<td>79.15</td>
<td>77.80</td>
<td><strong>78.47</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Spanish Test.</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>85.76</td>
<td>79.43</td>
<td>82.47</td>
</tr>
<tr>
<td>MISC</td>
<td>60.19</td>
<td>57.35</td>
<td>58.73</td>
</tr>
<tr>
<td>ORG</td>
<td>81.21</td>
<td>82.43</td>
<td>81.81</td>
</tr>
<tr>
<td>PER</td>
<td>84.71</td>
<td>93.47</td>
<td>88.87</td>
</tr>
<tr>
<td>overall</td>
<td>81.38</td>
<td>81.40</td>
<td><strong>81.39</strong></td>
</tr>
</tbody>
</table>

*System of Carreras et al., 2002*
# Timeline

<table>
<thead>
<tr>
<th></th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train&amp;Development Data</td>
<td>January 29\textsuperscript{th} 2013</td>
</tr>
<tr>
<td>Test Data</td>
<td>January 7\textsuperscript{th} 2013</td>
</tr>
<tr>
<td>Result Submission Deadline</td>
<td>January 8\textsuperscript{th} 2013 (11:59 pm GMT)</td>
</tr>
<tr>
<td></td>
<td><strong>Later submissions will not be accepted</strong></td>
</tr>
<tr>
<td>Technical Report Deadline</td>
<td>January 8\textsuperscript{th} 2013</td>
</tr>
</tbody>
</table>
Submit

• The source code for the feature generation (make sure it will run under Linux)

• The official train and test feature files used in the final run, together with the final output of your system for the test data

• Additionally generated resources (if any)

• Write ~4 paged brief description of your approach explaining:
  – used NLP tools
  – designed features
  – employed machine learning algorithm&mation
Evaluation is based on

• ranking of your system against the rest
• designed features
  – novel, previously unknown features will be favored
  – system’s pre or post processing
• generated resources
  – size, methods and sources for gazetteer extraction
  – trigger lists
• quality of the paper description
  – structure
  – use of literature
  – error analysis
Generate Your Own Resources

• Extract gazetteers from Wikipedia
  – People (singers, teachers, mathematicians etc.)
  – Locations (cities, countries)
  – Organizations (universities, IT companies etc.)

• Extract trigger words from WordNet
  – look for hyponyms of person, location, organization

• Extract and rank the patterns in which the NEs occurred in the train and development data. Show what percentages of these were found in the final test data.

• Extract lists of verbs found next to the NEs. Do you find any similarity/regularity of the verbs associated with each one of the NE categories?
What must I do ...

• Use the train and development data to design and tune your NE system

• Decide on the features you would like to incorporate in your NE system

• Choose a machine learning classifier from Weka
  • Intro by Marti Hearst [http://courses.ischool.berkeley.edu/i256/f06/lectures/lecture16.ppt](http://courses.ischool.berkeley.edu/i256/f06/lectures/lecture16.ppt)

This is a big assignment start early!
What you must not do ...

• Use existing named entity system(s) or library either as a feature generator, output generator etc.

• If you do, then you will have to run your system for two additional languages Spanish, Italian 😊
Available Resources

• WordNet [http://wordnet.princeton.edu/]

• Part-of-speech taggers
  – TreeTagger
    [http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html]
  – Stanford PoS Tagger
    [http://nlp.stanford.edu/software/tagger.shtml]

• NP chunker

• Parser
  – Stanford Parser
    [http://nlp.stanford.edu/software/lex-parser.shtml]

• Other
  [http://nlp.stanford.edu/links/statnlp.html]
Good Luck!