Machine Learning with WEKA
An Introduction

林松江
2005/9/30
WEKA: the bird

Copyright: Martin Kramer (mkramer@wxs.nl)
WEKA is available at

http://www.cs.waikato.ac.nz/ml/weka/

Downloading and installing Weka

Weka 3.4 is the latest stable version of Weka, and the one described in the data mining book. There are different options for downloading and installing it on your system:

- Windows

  Click here to download a self-extracting executable that includes Java VM 1.4 (weka-3-4-5.exe; 22,478,939 bytes)

  Click here to download a self-extracting executable without the Java VM (weka-3-4-5.exe; 8,787,772 bytes)

  These executables will install Weka in your Program Menu. Download the second version if you already have Java 1.4 (or later) on your system.

- Other platforms (Linux, etc.)
The format of Dataset in WEKA(1)

@relation heart-disease-simplified

@attribute age numeric
@attribute sex { female, male}
@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class { present, not_present}

@data
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
...
The format of Dataset in WEKA(2)

@relation heart-disease-simplified

@attribute age numeric
@attribute sex { female, male}
@attribute chest_pain_type { typ_angina, asympt, non_anginal, atyp_angina}
@attribute cholesterol numeric
@attribute exercise_induced_angina { no, yes}
@attribute class { present, not_present}
@data
63,male,typ_angina,233,no,not_present
67,male,asympt,286,yes,present
67,male,asympt,229,yes,present
38,female,non_anginal,?,no,not_present
...
Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
Explorer: pre-processing the data

- Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary
- Data can also be read from a URL or from an SQL database (using JDBC)
- Pre-processing tools in WEKA are called “filters”
- WEKA contains filters for:
  - Discretization, normalization, resampling, attribute selection, transforming and combining attributes, …
Pre-processing tools in WEKA are called “filters”: including discretization, normalization, re-sampling, attribute selection, transforming and combining attributes, …
Visualize class distribution for each attribute
Selected attribute
Name: petallength
Missing: 0 (0%)
Distinct: 43
Type: Numeric
Unique: 10 (7%)

Statistic | Value
--- | ---
Minimum | 1
Maximum | 6.9
Mean | 3.759
StdDev | 1.764
Click here to choose filter algorithm
Click here to set the parameter for filter algorithm.
Set parameter

Discretize -B 10 -M -1.0 -R first-last

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

- attributeIndices: first-last
- bins: 10
- desiredWeightOfInstancesPerInterval: -1.0
- findNumBins: False
- invertSelection: False
- makeBinary: False
- useEqualFrequency: False
An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

- **attributeIndices**: first-last
- **bins**: 10
- **desiredWeightOfInstancesPerInterval**: -1.0
- **findNumBins**: False
- **invertSelection**: False
- **makeBinary**: False
- **useEqualFrequency**: True

Filter options include:
- **Choose**
- **Open file**
- **Open URL**
- **Open DB**
- **Undo**
- **Edit**
- **Save**

Attributes:
- **No.**
- **Name**
- **sepallength**
- **sepalwidth**
- **petallength**
- **petalwidth**
- **class**
apply the filter algorithm
Equal frequency
Building “classifiers”

- Classifiers in WEKA are models for predicting nominal or numeric quantities.
- Implemented learning schemes include:
  - **Decision trees** and lists, instance-based classifiers, **support vector machines**, multi-layer perceptrons, logistic regression, Bayes’ nets, …
  - “Meta”-classifiers include:
    - Bagging, boosting, stacking, error-correcting output codes, locally weighted learning, …
Choose classification algorithm
Click here to set parameter for decision tree
Information about decision tree algorithm
Output setting
Start to build classifier
Weka Explorer

Classifier

Choose J48 -C 0.25 -M 2

Test options

- Use training set
- Supplied test set
- Cross-validation Folds 10
- Percentage split % 66

More options...

Classifier output

Run information:

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2
Relation: iris
Instances: 150
Attributes: 5
- sepal length
- sepal width
- petal length
- petal width
- class
Test mode: 10-fold cross-validation

Classifier model (full training set):

J48 pruned tree

Click right bottom to view more information

Fitted result
View fitted tree
The decision tree model for the 'iris' dataset is displayed in the Weka Explorer. The tree structure is based on the following criteria:

- **petalwidth**
  - <= 0.6
  - > 0.6

For petalwidth <= 0.6:
- **Iris-setosa** (50.0)

For petalwidth > 0.6:
- **petalwidth**
  - <= 1.7
  - > 1.7

For petalwidth <= 1.7:
- **petallength**
  - <= 4.9
  - > 4.9

For petallength <= 4.9:
- **Iris-versicolor** (48.0/1.0)

For petallength > 4.9:
- **petalwidth**
  - <= 1.5
  - > 1.5

For petalwidth <= 1.5:
- **Iris-virginica** (3.0)

For petalwidth > 1.5:
- **Iris-versicolor** (3.0/1.0)
Crosses: correctly classified instances
Squares: incorrectly classified instances
SMO: Support Vector Machine algorithm
Choose RBF kernel
Click right bottom to view more information
clustering data

- WEKA contains “clusterers” for finding groups of similar instances in a dataset
- Implemented schemes are:
  - \( k \)-Means, EM, Cobweb, \( X \)-means, FarthestFirst
- Clusters can be visualized and compared to “true” clusters (if given)
- Evaluation based on loglikelihood if clustering scheme produces a probability distribution
clustering data
Choose clustering algorithm
Click here to set parameter

SimpleKMeans -N 2 -S 10

Clusterer

Cluster mode

- Use training set
- Supplied test set
- Percentage split
- Classes to clusters evaluation

(Nom) class

Store clusters for visualization

Ignore attributes

Start

Result list (right-click for options)

Status

OK
Start to perform clustering
Cluster output:

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Count</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>61</td>
<td>41%</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>33%</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
<td>26%</td>
</tr>
</tbody>
</table>

Class attribute: class

Classes to Clusters:

0 1 2  --- assigned to cluster
0 50 0  | Iris-setosa
47 0 3  | Iris-versicolor
14 0 36 | Iris-virginica

Cluster 0  --- Iris-versicolor
Cluster 1  --- Iris-setosa
Cluster 2  --- Iris-virginica

Incorrectly clustered instances : 17.0  11.3333 %
Click right bottom to view more information

View cluster assignments
Finding associations

- WEKA contains an implementation of the Apriori algorithm for learning association rules
  - Works only with discrete data
- Can identify statistical dependencies between groups of attributes:
  - milk, butter $\Rightarrow$ bread, eggs (with confidence 0.9 and support 2000)
- Apriori can compute all rules that have a given minimum support and exceed a given confidence
Finding associations
Attribute selection

- Panel that can be used to investigate which (subsets of) attributes are the most predictive ones

- Attribute selection methods contain two parts:
  - A search method: best-first, forward selection, random, exhaustive, genetic algorithm, ranking
  - An evaluation method: correlation-based, wrapper, information gain, chi-squared, ...

- Very flexible: WEKA allows (almost) arbitrary combinations of these two
Choose attribute evaluator: SVMAttributeEval

Search method: Ranker

Attribute Selection Mode: Ranker
- Ranks attributes by their individual evaluations.
Attribute selection output

=== Attribute Selection on all input data ===

Search Method:
   Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 class):
   SVM feature evaluator

Ranked attributes:
   4 3 petallength
   3 2 sepalwidth
   2 4 petalwidth
   1 1 sepalwidth

Selected attributes: 3,2,4,1 : 4
Data visualization

- Visualization very useful in practice: e.g. helps to determine difficulty of the learning problem
- WEKA can visualize single attributes (1-d) and pairs of attributes (2-d)
  - To do: rotating 3-d visualizations (Xgobi-style)
- Color-coded class values
- “Jitter” option to deal with nominal attributes (and to detect “hidden” data points)
- “Zoom-in” function
Choose range to magnify local result
submit to magnify local result
Weka GUI Chooser

Waikato Environment for Knowledge Analysis

Version 3.4.5

(c) 1999 - 2005
University of Waikato
New Zealand

GUI

Simple CLI
Explorer
Experimenter
KnowledgeFlow
Performing experiments

- Experimenter makes it easy to compare the performance of different learning schemes
- For classification and regression problems
- Results can be written into file or database
- Evaluation options: cross-validation, learning curve, hold-out
- Can also iterate over different parameter settings
- Significance-testing built in!
Step 1: Set output file
Step 2: Add dataset
Step 3: Choose algorithm
Experiment status

Log

01:30:18: Started
01:33:07: Finished
01:33:07: There were 0 errors
View the experiment result
Click here to perform test

Confidence: 0.05 (two tailed)
Date: 2005/9/27 上午 1:37

Dataset | (1) function | (2) function | (3) trees
---------|-------------|-------------|-----------
iris     | (100)       | 96.27       | 96.00     | 94.73     |
weather  | (100)       | 54.00       | 52.00     | 66.50     |
          | (v/ v/*)    | (0/2/0)     | (0/2/0)   |

Skipped:

Key:

(1) functions.SMO -C_1.0 -E_1.0 -G_0.01 -A_250007_-L_0.0010_-P_1.0E-12_-
(2) functions.RBFNetwork -B_2_-S_1_-R_1.0E-8_-M_-1_-W_0_0_1_-3.669814959711217733168393644448E17
Knowledge Flow GUI

- New graphical user interface for WEKA
- Java-Beans-based interface for setting up and running machine learning experiments
- Data sources, classifiers, etc. are beans and can be connected graphically
- Data “flows” through components: e.g., “data source” -> “filter” -> “classifier” -> “evaluator”
- Layouts can be saved and loaded again later
Step 1: choose data source
Step 2: choose data source format
Step 3: set data connection
Visualize data
Step 4: choose evaluation

Step 5: choose filter method

Step 6: choose classifier

Set parameter
Final step: start to run
Thank you