Computational Intelligence: Methods and Applications

Lecture 17

WEKA/Yale & knowledge extraction from simplest decision trees

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

A few data mining packages

A large number of data mining packages that include many CI models for data analysis is available.

See long list of DM software, including large commercial packages.

GhostMiner, from Fujitsu (created by our group); please get it.

WEKA started the trend to collect many packages in one system.

YALE, Yet Another Learning Environment – initially a better front-end to WEKA, includes all WEKA models, free source; please get it.

New interesting projects:

Orange, component-based data mining software, includes visualizations, SOM/MDS modules, 2006.

KNIME, based on Eclipse platform, includes Weka and R-scripts, modular data exploration platform, visual data flows.

WEKA Project

Machine learning algorithms in Java:

I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann 1999

Project Web page: www.cs.waikato.ac.nz/ml/weka

One of the most popular packages.

Essentially a collection of Java class libraries implementing various computational intelligence algorithms.

ARFF data format, with data in CSV format (comma separated, exportable from spreadsheets), and additional information about the data, type of each feature, etc.

CLI, or command line interface (only for Unix lowers), ex:

```
java weka.classifiers.j48.J48 -t data/weather.arff
```

calls one of the methods (here J48) from the library.

WEKA Software

"Explorer GUI" for making basic calculations, recently much improved.

"Experimenter" and "Knowledge Flow" environments for performing more complex experiments is provided, this allows for averaging over crossvalidation results or combining different models.

On-line documentation for library classes but little description of methods.

WEKA software contains:

- preprocessing filters, supervised and unsupervised
- many classification models
- rule-based models for knowledge discovery
- association rules (one method)
- regression, or numerical prediction models
- 3 clusterization (unsupervised learning) methods
- scatterogram visualization (2D)
- a collection of sample simple problems (from the UCI repository).

More WEKA Software

Simple “Explorer GUI” for making basic calculations.

Rather rough “Experimenter” environment for performing more complex experiments, such as averaging over crossvalidation results or combining different models is provided.

On-line documentation for library classes.

WEKA software contains:

- preprocessing filters, supervised and unsupervised
- classification models
- rule-based models for knowledge discovery
- association rules (one method)
- regression, or numerical prediction models
- 3 clusterization (unsupervised learning) methods
- scatterogram visualization (2D)
- a collection of sample problems (from UCI repository)

WEKA strong/weak points

Platform independent – Java!

Many projects created around it: listed at the WEKA web page.

Free, contains large collection of filters and algorithms.

May be extended by a serious user.

But ... Java programs are not so stable as Windows programs, there are problems with some Java versions;

rather poor visualization of data and results;

only simple user interface (but has been improved recently),

requires tedious programming to perform experiments (but Knowledge Flow GUI changes this);

algorithms are not described in details in documentation and in the book (only in the class libraries).

YALE

Like WEKA, same models + few more, easier to use.

Free, contains large collection of filters and algorithms.

May be extended by a serious user.

Download Yale, start it and read the tutorial!

Includes 20 visual data exploration methods: scatter, scatter matrix, interactive scatter 3D, parallel, 2D density, radial radviz, gradviz, SOM (U-distance and P-density),

Unfortunately algorithms are not described in details in Yale documentation and you have to study class libraries to understand what exactly GridViz or RadViz does, or read original papers to understand what U, U* and P matrix SOM visualization is.

Check much better descriptions of methods in Orange!

Knowledge representation

Knowledge representation is an important subject in Artificial Intelligence, here only simple forms of knowledge are considered.

Decision rules:

prepositional rules: IF (all conditions are true) THEN facts

M-of-N rules: IF (M conditions of N are true) THEN facts

fuzzy rules: IF (conditions true to some degree) THEN facts are true to some degree

Linguistic variables: favorite-colors, low-noise-level, young-age, etc:

- subsets of nominal or discrete values,
- intervals of numerical values, ex: teenager ={T if age<20}
- constrained subsets of numerical values.

WEKA filters

Many filters that can be applied to attributes (features) or to instances (vectors, samples), some specific to signal/time series data.

- Divided into supervised/unsupervised, attribute or instance.
- Create new attribute from existing ones using algebraic operations;
- remove instances with attribute values in some range, for example missing values; delete attributes of specific type (ex. binary)
- change nominal values to binary combinations, ex. $X_i \in \{a,b,c,d\} \Rightarrow (X_{i1},X_{i2}) \in (\{0,1\},\{0,1\})$
- rank the usefulness of attributes (several schemes);
- evaluate usefulness of subsets of features (several schemes);
- perform PCA; normalize features in many ways;
- discretize attributes, define simple bins or look for more natural discretization, for example bins created by the Minimum Description Length (MDL) principle (called “use Kononenko”).
- many others ...

WEKA classification algorithms

Divided into:

- Bayes – versions of probabilistic Bayesian methods
- Functions – parameterized functions, linear and non-linear
- Lazy – no parameter learning, all work done when classifying
- Meta – committees, voting, boosting, stacking ... metamodels.
- Misc – untypical models, fuzzy lattice, hyperpipes, voting features
- Tree-building models, recursive partitioning
- Rule learning models

These algorithm enable:

- knowledge discovery, or data mining (trees, rules);
- predictive modeling in classification or regression tasks.

See WEKA 3-3 detailed presentation:
http://prdownloads.sourceforge.net/weka/weka.ppt

WEKA decision rules

Algorithm for knowledge discovery, or data mining, 10 rule and 10 tree-based, providing knowledge in form of logical rules.

- Zero-R, predicting majority class (or mean values)
- One-R, simplest one-level (one attribute) decision tree.
- Decision stump, one-level tree
- C4.5, called here J.48, since this is Java implementation of the version 8 of C4.5 decision tree algorithm.
- M5’ model tree learner.
- Naive Bayes tree classifier.
- PART rule learner (covering algorithm).

Prototype – based algorithms:

- Instance –based learner (IB1, IBk, ID3) nearest neighbor method
- Decision table

WEKA regression algorithms

Regression (function) and classification algorithms include:

- Naive Bayes (2 versions)
- Linear Regression, or LDA
- Additive regression
- Logistic regression
- LWR, Locally Weighted Regression
- MLP (multi-layer perceptron) neural network,
- VPN, voted perceptron network
- SMO, or Support Vector Machine algorithm
- K*, similarity based system with algorithmic complexity minimization.

WEKA other algorithms

Statistical algorithms for model improvement (meta-algorithms):

- bagging,
- boosting,
- adaboost
- logit boost,
- stacking

Clusterization:

- K-means,
- Expectation Maximization,
- Cobweb

Association: find relations between attributes.

Visualization of 2D scatterograms

WEKA example

Contact lenses: do I need hard, soft or none?

Very small data set, 24 instances: contact-lens.arff

What is in the database?

1. age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic
2. spectacle prescription: (1) myope, (2) hypermetrope
3. astigmatic: (1) no, (2) yes
4. tear production rate: (1) reduced, (2) normal

Class Distribution:

1. hard contact lenses: 4
2. soft contact lenses: 5
3. no contact lenses: 15

ZeroR

Zero method:

- for a small number of classes (categorical class variables) predict the majority class;
- for numerical outputs (regression problems) predict the average.

Useful to establish the base rate, zero variance, large bias:
if any method obtains results that are worse than ZeroR serious overfitting of data occurs.

For contact-lenses: confusion matrix

```
=== Confusion Matrix ===

  a  b  c   <= classified as
  0  0  5 |  a = soft
  0  0  4 |  b = hard
  0  0 15 |  c = none
```

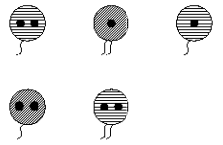
15 classified correctly, 62.5% on the whole data.

What happens in 10xCV?

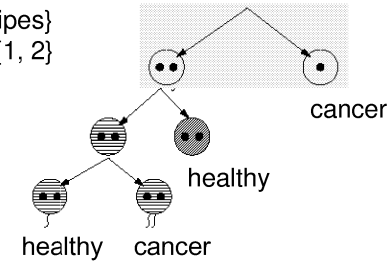
DT - idea

Class: {cancer,

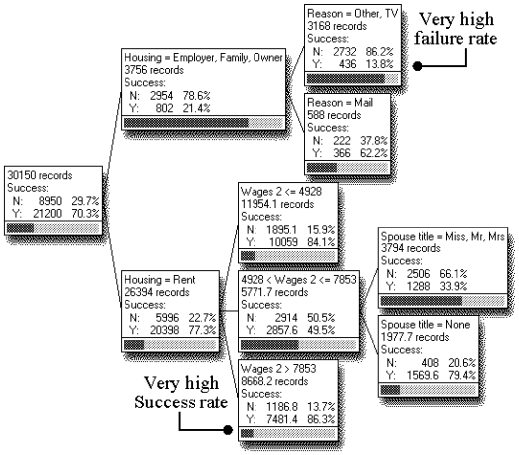
healthy}



Features: cell body: {gray, stripes}
nuclei: {1, 2}; tails: {1, 2}



More ambitious tree



1R

1R: simplest useful tree (Holte 1993), sometimes results are good.
One level tree, nominal attributes.

1R algorithm:

for every attribute X

for every attribute value X_i:

count the class frequencies $N(X_i, \omega_j)$

find the most frequent class $c = \arg\max_j N(A_i, \omega_j)$

create a rule (majority classifier): IF X_i THEN ω_c

Calculate accuracy of this rule.

Select rules of highest accuracy.

Missing value ? is treated as any other nominal value.

1R example

Example take from WEKA book: weather condition and decision to play an in-door games (tennis); 14 examples are given
Task: find the decision rule (weather.nominal.arff).

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temperature	Hot → No*	2/4	5/14
	Mild → Yes	2/6	
	Cool → Yes	1/4	
Humidity	High → No	3/7	4/14
	Normal → Yes	1/7	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	

Attribute: Outlook

Outlook = Sunny has 3 examples with No and 2 with Yes;

Outlook = Overcast has 4 examples with Yes

Rainy has 3 examples with Yes and 2 with No;

Optimal rules: using only Outlook, or only Humidity.

Dataset is too small to evaluate accuracy, but rules are reasonable.

1R continuous

How should the continuous values be treated?

Divide the range of continuous attribute into intervals $I_i(X)$ (discretize the attribute);

treat intervals as nominal values, i.e. write $X=I_i$ if $X \in I_i(X)$.

For each attribute X

sort all cases according to the increasing X values;
find the intervals $I_i(X)$ with constant majority class $N(I_i(X), \omega_c)$.

This should decrease the number of errors in 1R algorithm.

Problem: noisy data; a single example may be quite untypical, it should not be take as a rule!

Solution: bucket

Example with continuous v.

Example: discretization of temperature, instead of hot, mild, cool.

```
64   65   68   69   70   71   72   72   75   75   80   81   83   85
Yes | No | Yes Yes Yes | No No Yes | Yes Yes | No | Yes  Yes | No
```

To avoid noise intervals containing not less than 4 elements are used.

```
64   65   68   69   70   71   72   72   75   75   80   81   83   85
Yes | No | Yes Yes Yes | No No Yes | Yes Yes | No | Yes  Yes | No
```

WEKA implementation:

Bucket size = min number of elements in interval.

Slightly more accurate solution is found.

weather.arff data

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temperature	≤ 77.5 → Yes	3/10	5/14
	> 77.5 → No*	2/4	
Humidity	≤ 82.5 → Yes	1/7	3/14
	> 82.5 and ≤ 95.5 → No	2/6	
	> 95.5 → Yes	0/1	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	