Recap

- Week 8
  - Two classification algorithms
  - Feature-based
- Last Tuesday: Naive Bayes
  - Probability of class, given the feature values: $p(c|f_1, f_2, \ldots, f_n)$
  - Naive: Features are independent
- Last Thursday: Decision Tree
  - Hierarchical data structure as a prediction model
  - Recursive training algorithm
  - Core question: Where to put each feature?
    - The one with the highest information gain (= the highest loss in entropy)

Weka GUI

Sprachverarbeitung (VL + Ü)

Nils Reiter

June 6, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Exercise

This exercise can and should be done on your own laptop.

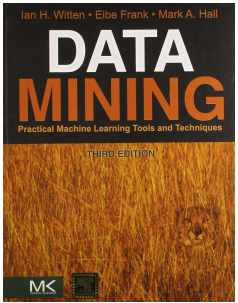
- Download and install Weka from this page: `https://www.cs.waikato.ac.nz/~ml/weka/`. The page provides download packages for Mac, Windows and Linux.
- Download the files `Werther_train.arff` and `feature-table.pdf` from the course web page. The pdf file contains a description of the features that are present in the arff file.
- Load the file in Weka, and train the best model you can, using percentage split evaluation. We will use a proper (unknown) test set later on. What you can do:
  - Exclude features by removing them
  - Apply filters on the features (e.g., change data types, remove rare feature values, ...)
  - Use different ML algorithms

Section 1

Weka

Introduction

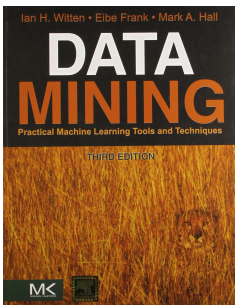
Ian H. Witten/Eibe Frank/Mark A. Hall (2014). *Data Mining*. 3rd ed. Practical Machine Learning Tools and Techniques. Elsevier



Introduction

Ian H. Witten/Eibe Frank/Mark A. Hall (2014). *Data Mining*. 3rd ed. Practical Machine Learning Tools and Techniques. Elsevier

- Open source, Java: `https://www.cs.waikato.ac.nz/ml/weka/`
- Collection of machine learning algorithms
- Playground, GUI, well documented
- Technical limitation: Data sets have to fit in memory



Weka parts (of the ›Explorer‹)

- Preprocess: Remove attributes or instances, rebalance the data set, ...
- Classify: Train and test a classifier
- Cluster: Run a clustering algorithm
- Associate: Investigate associations between features[1]
- Select attributes: Rank attributes according to their importance for a class
- Visualize: Plotting

[1] Association != correlation

Filters

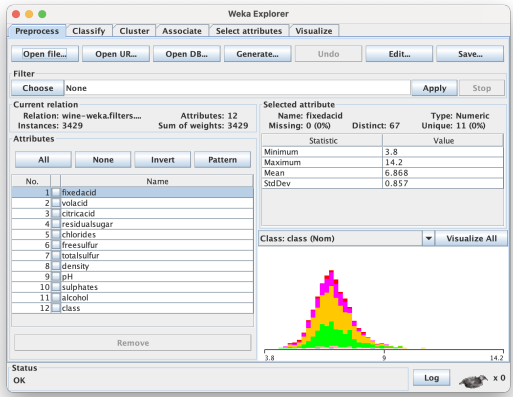
Motivation

- We often don't have the data as we need them to be
- Preprocessing
  - Manipulating CSV ✔
  - Filters in Weka – today

Filters

Motivation

- We often don't have the data as we need them to be
- Preprocessing
  - Manipulating CSV ✔
  - Filters in Weka – today
- Weka Explorer → Preprocess
- Filter → Choose



Filters

Types

- `supervised` – Filters that use a class attribute
- `unsupervised` – Filters that do not use a class attribute

Filters

Types

- `supervised` – Filters that use a class attribute
- `unsupervised` – Filters that do not use a class attribute
- `attribute` – manipulate feature(s)
- `instance` – manipulate instances

Filters

Types

- `supervised` – Filters that use a class attribute
- `unsupervised` – Filters that do not use a class attribute
- `attribute` – manipulate feature(s)
- `instance` – manipulate instances
- Next slides: One filter from each group
  - Full overview (javadoc): `https://javadoc.io/static/nz.ac.waikato.cms.weka/weka-stable/3.8.4/weka/filters/package-summary.html`

weka.filters.supervised.attribute.MergeNominalValues

- Merges *values* of nominal attributes
- Implements ›Chi-square automatic interaction detection‹ (CHAID) Kass (1980)
- Idea: Merge values that are not needed for classification

weka.filters.supervised.attribute.MergeNominalValues

- Merges *values* of nominal attributes
- Implements ›Chi-square automatic interaction detection‹ (CHAID) Kass (1980)
- Idea: Merge values that are not needed for classification

Parameters

- `-D` Turns on output of debugging information.
- `-L` <double> The significance level (default: 0.05).
- `-R` <range> Sets list of attributes to act on (or its inverse). Default: `first-last`
- `-V` Invert matching sense (i.e. act on all attributes not specified in list)
- `-O` Use short identifiers for merged subsets.

weka.filters.supervised.instance.Resample

- Produce random subsample of a dataset
- With replacement or without replacement
- Only for nominal class attributes
- Can be used to even the data set

Parameters

- `-S <num>` Specify the random number seed. Default: 1
- `-Z <num>` The size of the output dataset (perc. of input). Default: 100
- `-B <num>` Bias factor towards uniform class distribution. 0 = distribution in input data – 1 = uniform distribution. Default: 0
- `-no-replacement` Disables replacement of instances (default: with replacement)
- `-V` Inverts the selection - only available with `-no-replacement`.

`weka.filters.unsupervised.attribute.StringToWordVector`

- Takes `string` attributes and turns them into occurrence vector representation
- Each vector dimension becomes an individual attribute

`weka.filters.unsupervised.attribute.StringToWordVector`

- Takes `string` attributes and turns them into occurrence vector representation
- Each vector dimension becomes an individual attribute

Example

$$\begin{array}{l} \text{The dog barks.} \\ \text{The dog sleeps.} \end{array} \rightarrow \begin{array}{ccccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{array}$$

weka.filters.unsupervised.attribute.StringToWordVector

- Takes `string` attributes and turns them into occurrence vector representation
- Each vector dimension becomes an individual attribute

Example

$$\begin{array}{l} \text{The dog barks.} \\ \text{The dog sleeps.} \end{array} \rightarrow \begin{array}{ccccc} 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{array}$$

Parameters

`https://javadoc.io/static/nz.ac.waikato.cms.weka/weka-stable/3.8.4/weka/filters/unsupervised/attribute/StringToWordVector.html`

weka.filters.unsupervised.instance.RemovePercentage

- Removes a given percentage of a dataset

Parameters

- `-P <percentage>` Specifies percentage of instances to select. Default: 50
- `-V` Specifies if inverse of selection is to be output

Section 2

Exercise

The Task

- Text material: Goethes' *Die Leiden des jungen Werther*
- Goal: Identify references to entities
  - Entities: Characters, locations, ...
  - References to entities: Proper names (»John Snow«), descriptions (»the wall«).
    - No pronouns!

The Task

- Text material: Goethes' *Die Leiden des jungen Werther*
- Goal: Identify references to entities
  - Entities: Characters, locations, ...
  - References to entities: Proper names (»John Snow«), descriptions (»the wall«).
    - No pronouns!
- Annotations are token-wise and distinguishes entity types
  - Tokens that are not part of an entity are marked with »O«
  - Example:

John	Snow	goes	to	the	wall	.
B-EntityPER	I-EntityPER	O	O	B-EntityLOC	I-EntityLOC	O

Exercise

This exercise can and should be done on your own laptop.

- Download and install Weka from this page: `https://www.cs.waikato.ac.nz/~ml/weka/`. The page provides download packages for Mac, Windows and Linux.
- Download the files `Werther_train.arff` and `feature-table.pdf` from the course web page. The pdf file contains a description of the features that are present in the arff file.
- Load the file in Weka, and train the best model you can, using percentage split evaluation. We will use a proper (unknown) test set later on. What you can do:
  - Exclude features by removing them
  - Apply filters on the features (e.g., change data types, remove rare feature values, ...)
  - Use different ML algorithms