Machine Learning with Weka: Filters and Clustering

PU Tools, Ressourcen, Infrastruktur

Nils Reiter,
nils.reiter@uni-koeln.de

December 10, 2020
(Winter term 2020/21)

Exercise 5

https://github.com/idh-cologne-tools-ressourcen-infra/exercise-05

Section 2

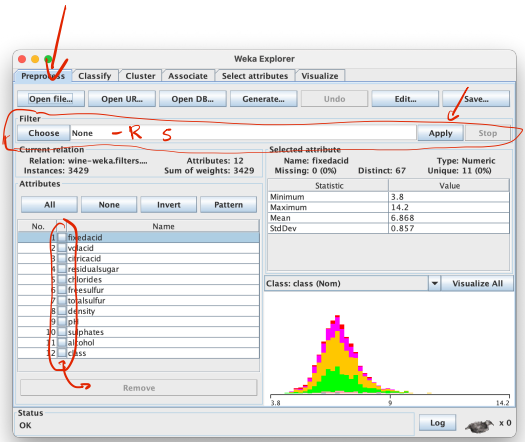
Weka: Filters and Clustering

Motivation

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

Filters

- 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

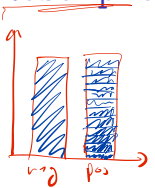
- Merges *values* of nominal attributes
- Implements ›Chi-square automatic interaction detection‹ (CHAID) Kass (1980)
- Idea: Merge values that are not needed for classification

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

String-feature

"The dog barks." → 1 1 0 1 1
"The dog sleeps." → 1 1 1 0 1

`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

Subsection 1

Clustering

Introduction

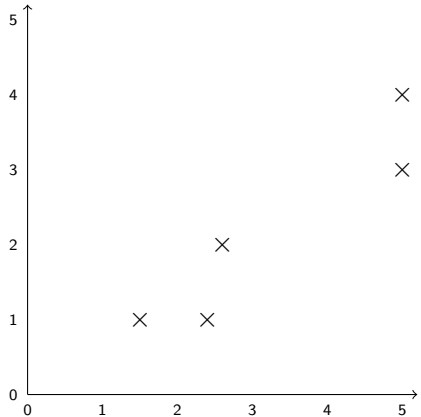
- Group instances based on feature values
- No target class
- Use cases:
  - No annotations
  - Finding patterns in data
- We usually have to specify how many clusters we want
- Assumption: All attributes can be represented numerically

K-Means

- A simple clustering method
- Two alternating steps, run as long as ›something is happening‹
  1. Assignment: Each instance is assigned to the mean of the nearest cluster
  2. Update: For each cluster, recalculate its mean

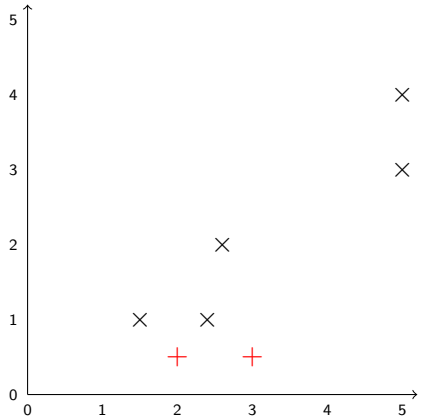
K-Means

Example



K-Means

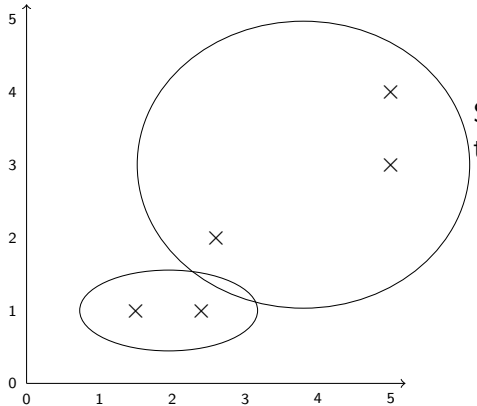
Example



Step 1: Start with random means

K-Means

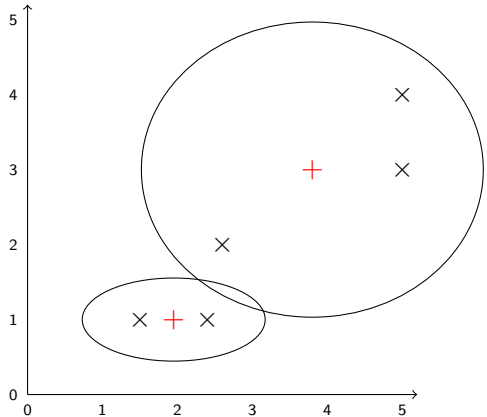
Example



Step 2: Assign each point to nearest mean

K-Means

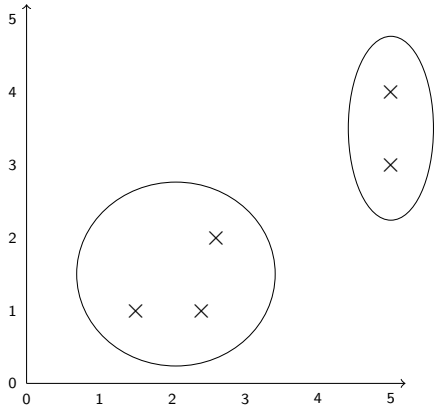
Example



Step 3: Update means

K-Means

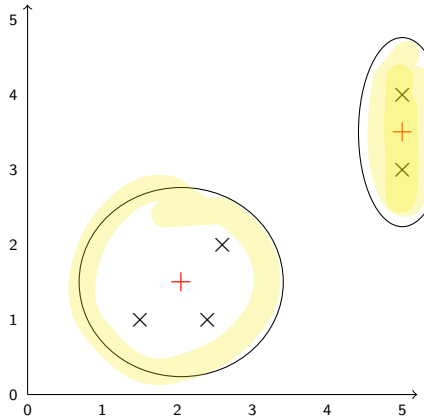
Example



Step 4: Assign each point to nearest mean

K-Means

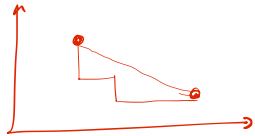
Example



Step 5: Update means

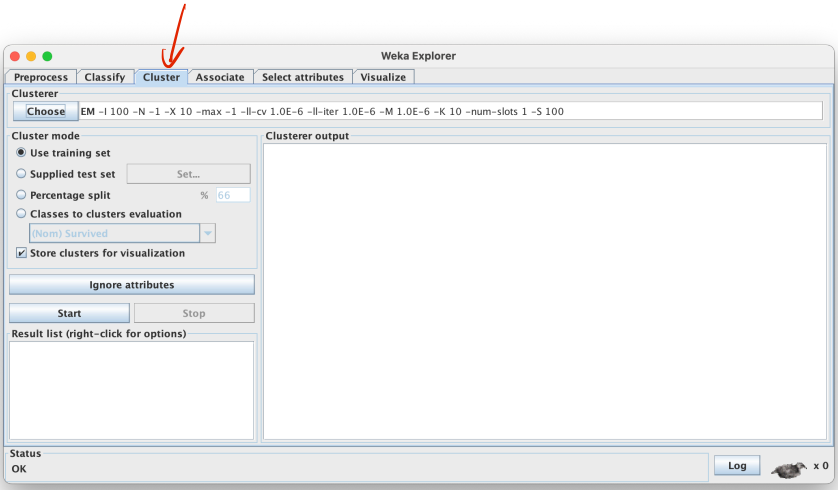
K-Means

Parameters



- How many means = how many clusters
- How to calculate distance
  - Euclidean, Manhatten, ...
- How to calculate means
  - Weighting for some attributes, ...
- How to start
  - Random points? Random instances?

Clustering in Weka



Exercise 6

https://github.com/idh-cologne-tools-ressourcen-infra/exercise-06

References I

 Kass, Gordon V. (1980). »An Exploratory Technique for Investigating Large Quantities of Categorical Data«. In: *Applied Statistics* 29.2, pp. 119–124.