Weka

Intro

Weka ist ...

- Eine Sammlung von Algorithmen für Machine Learning und Data Mining
- In Java implementiert
- Hat eine GUI und eine API
- Lizensiert unter GNU GPL
- `http://www.cs.waikato.ac.nz/ml/weka/`

Datenformate I

CSV (Comma-Separated Values)

- Ein Beispiel pro Zeile
- Merkmale werden durch Komma getrennt

Example

```
Darth, upper, ""
Vader, upper, Darth
war, lower, Vader
ein, lower, war
Lord, upper, ein
der, lower, Lord
Sith, upper, der
...
```

Datenformate II

ARFF - *Attribute Relation File Format*

Standardformat in Weka

Example

```
@RELATION darth-vader
@ATTRIBUTE token STRING
@ATTRIBUTE case {upper,lower}
@ATTRIBUTE previous STRING
@ATTRIBUTE class {name, other}
@DATA
"Darth", upper, "", name
"Vader", upper, "Darth", name
"war", lower, "Vader", other
"ein", lower, "war", other
...
```

Datenformate III

Syntax von ARFF

- `@RELATION name`
  definiert einen Namen für das Datenset
- `@ATTRIBUTE attribute TYPE`
  definiert ein Attribut namens "attribute" vom Typ TYPE

string	Zeichenketten
numeric, real, integer	Zahlen
{ nom1, nom2 }	Listen nominaler Werte
date	Datumsangaben (yyyy-MM-dd'T'HH:mm:ss)

- `@DATA`
  Hier stehen die einzelnen Elemente (in CSV-Format)

Datenformate IV

Beispiel nominaler Werte

- { red, green, blue }
- { gabi, paula, anna-katharina }
- { one, two, three }
- { true, false }

Konvertierung

Sind alle Zeichenketten in einem Datenset bekannt, können sie automatisch in nominale Werte konvertiert werden.

Datenformate V

Annotation, fehlende Werte, Sonderzeichen

- Klassen werden im Attribut *class* angegeben, normalerweise als letztes Attribut
- Fehlende Werte werden mit einem `?` gekennzeichnet.
- Kommentare beginnen mit '%'
- Sonderzeichen (z.B. '?',',', '%') müssen in Anführungszeichen stehen, wenn sie nicht in ihrer Sonderbedeutung vorkommen.

Weka Benutzeroberfläche

Weka GUI Chooser

- ausführen mit

```
:~$ java -jar /path/to/weka.jar
```

- **Explorer**: Daten importieren, bearbeiten und visualisieren
- **Experimenter**: Experimente mit unterschiedlichen Parametern durchführen
- **KnowledgeFlow**: Komponenten und Datenströme graphisch modellieren
- **Simple CLI**: Command Line Interface

CLI

- Jeder Klassifizierer ist in einer Java-Klasse implementiert
- Aufruf über die Kommandozeile möglich:[5]

```
:~$ java weka.classifiers.trees.J48 <parameter>
```

- Parameter: Manche Parameter werden von jedem Klassifizierer verwendet (zum Beispiel Angabe der Trainings- und Testdaten), manche Parameter sind spezifisch für bestimmte Klassifizierer
- Wird der Klassifizierer ohne Argumente gestartet, zeigt der help screen alle Parameter an

[5] J48: Weka-Implementierung eines Entscheidungsbaums

Weitere Informationen zu Weka

- Witten, Frank & Hall (2011): Data Mining. Morgan Kaufman.
  $\rightarrow$ UB
- Online-Kurse: `https://weka.waikato.ac.nz`

Übung 16