Recap

- Last Tuesday
  - Tabular data as input for machine learning systems
  - File formats: CSV/TSV, ARFF
  - Basic statistics about features and classes
    - I.e., how often does each feature value appear?
- Last Thursday: How to evaluate classification systems?
  - Classification: Objects into categories
  - Accuracy: Percentage of correctly classified instances
  - Precision: How many of the objects labeled as X are actually X?
  - Recall: How many of the objects that are X did we find?
  - Different data sets for different purposes: Training/test split

Course Evaluation

Ilias

[SoSe23] Sprachverarbeitung

Veranstaltungsevaluation

`https://www.ilias.uni-koeln.de/ilias/goto_uk_svy_5263327.html`

best worst

1 ⟷ 7

Machine Learning Experiments with Weka

Sprachverarbeitung (VL + Ü)

Nils Reiter

May 16, 2023

INSTITUT FÜR
DIGITAL HUMANITIES
UNIVERSITÄT ZU KÖLN

Exercise

1. Create an ARFF file from titanic.csv. For this, you need to specify the header with the data types, while the actual data set can remain as it is. You need to make a copy of the file into your own directory first.
   You can test your file by asking the class weka.core.converters.ArffLoader to load it:
   ```
   java -cp /tools/weka/weka.jar weka.core.converters.ArffLoader FILE
   ```
2. Train and evaluate a machine learning model with Weka. The simplest command to do that is
   ```
   java -cp /tools/weka/weka.jar weka.classifiers.Evaluation weka.classifiers.bayes.NaiveBayes -t DATAS
   ```
   Please replace DATASET with the filename of your ARFF file. Inspect the output and verify that this is indeed predicting the survival of the passengers. If you leave out the dataset argument, available options are printed on the command line. Alternatively, you can also read about them in the API documentation:
   https://weka.sourceforge.io/doc.stable-3-8/
3. If you're certain Weka does what it's supposed to do, inspect the output of the evaluation and identify parts you don't understand.

Weka

- Open source software
- Written in Java
- Collection of machine learning algorithms
- Common interface

The Book

Ian H. Witten/Eibe Frank (2005). *Data Mining*. 2nd ed. Practical Machine Learning Tools and Techniques. Elsevier

Weka

Design Principles

- Different algorithms and utilities in different packages and classes
  - E.g., `weka.classifiers.bayes.NaiveBayes` contains the implementation of Naive Bayes
- Many packages contain a `main` function
- Main functions show help in the command line
- Many options work the same across different packages

Weka

Design Principles

- Different algorithms and utilities in different packages and classes
  - E.g., `weka.classifiers.bayes.NaiveBayes` contains the implementation of Naive Bayes
- Many packages contain a `main` function
- Main functions show help in the command line
- Many options work the same across different packages
- Graphical user interface (GUI): Next Week ☺

Webpage Javadoc

Java on the command line

- The command `java` is used to launch a Java program
- Two relevant arguments
    - Classpath: Where in the system can we find classes?
        - `-cp SOME_PATH:SOME_JAR`
        - Jar files are containers for Java classes
    - In which class do we look for the `main`-function?
        - Just give the class name as argument
    - Whatever comes after the class name is passed as `String[]` argument into the main function

Naive Bayes

- In Weka: weka.classifiers.bayes.NaiveBayes
- Probabilistic classification algorithm
- Training algorithm: Extract probabilities from corpus
- Prediction model (application): Assign class with highest probability
- More theory: Next week, Tuesday

Weka

Running a Weka Algorithm

- Running a Weka class:`java -cp /tools/weka/weka.jar CLASS_NAME`
- If necessary, more arguments are added after the class name

Weka

Running a Weka Algorithm

- Running a Weka class:`java -cp /tools/weka/weka.jar CLASS_NAME`
- If necessary, more arguments are added after the class name

Experiment = Training and Evaluation

- The main class: `weka.classifiers.Evaluation`
- The ML algorithm: `weka.classifiers.bayes.NaiveBayes`
- Command:

```
java -cp /tools/weka/weka.jar weka.classifiers.Evaluation weka.classifiers.bayes.NaiveBayes
```

  - Train a Naive Bayes model, evaluate it, and print out the results demo

ARFF

(Attribute relation file format)

- Used by machine learning toolkit Weka
- Data as CSV
- Header to define attributes/features
- Name/type for each attribute
    - Nominal values: Possible values

ARFF

Default format used by Weka.

Example

```
% A comment
@RELATION darth-vader

@ATTRIBUTE token STRING
@ATTRIBUTE case { upper, lower }
@ATTRIBUTE length NUMERIC
@ATTRIBUTE class { Noun, Verb, Adjective, Other }

@DATA
"Darth", upper, 5, Noun
"Vader", upper, 5, Noun
"war", lower, 3, Verb
"ein", lower, 3, Other
...
```

Syntax of ARFF

- `@RELATION name`
  defines a name for this data set
- `@ATTRIBUTE attribute TYPE`
  defines an attribute with the name “attribute” and the data type TYPE

string	Character sequences
numeric, real, integer	Numbers
{ nom1, nom2 }	List of nominal values
date	Dates (yyyy-MM-dd’T’HH:mm:ss)

- `@DATA`
  Now the items

Data types

Examples for nominal values

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

- Conversion: If all strings in a data set are known, they can be converted automatically in nominal values
- Not all classifiers can work with all data types!

Section 1

Exercise

Exercise

1. Create an ARFF file from `titanic.csv`. For this, you need to specify the header with the data types, while the actual data set can remain as it is. You need to make a copy of the file into your own directory first.
   You can test your file by asking the class `weka.core.converters.ArffLoader` to load it:

   ```
   java -cp /tools/weka/weka.jar weka.core.converters.ArffLoader FILE
   ```

2. Train and evaluate a machine learning model with Weka. The simplest command to do that is

   ```
   java -cp /tools/weka/weka.jar weka.classifiers.Evaluation weka.classifiers.bayes.NaiveBayes -t DATAS
   ```

   Please replace DATASET with the filename of your ARFF file. Inspect the output and verify that this is indeed predicting the survival of the passengers. If you leave out the dataset argument, available options are printed on the command line. Alternatively, you can also read about them in the API documentation:
   `https://weka.sourceforge.io/doc.stable-3-8/`

3. If you're certain Weka does what it's supposed to do, inspect the output of the evaluation and identify parts you don't understand.