JOHANNES KEPLER
UNIVERSITY LINZ

Exercise Part 1 of 3 - KV Computational Data Analytics (351.008)

Summer term 2020 - Prof. Dr. Johannes Fürnkranz, Florian Beck

---

Q&A Zoom Session:	**April 22, 10:15-11:45**
Deadline:	**May 3, 23:59**

The aim of the exercises is to gain practical experience in machine learning. For this we will use the machine learning frameworks `Weka`[1] and `KNIME`[2]. The exercises are split in three parts which will be uploaded when the corresponding lecture notes have also been uploaded. In total, up to 15 points are awarded for the successful completion of the exercises (usually two points per exercise), which are included in the overall grade with 15%. The tasks can be completed alone or in a small group (maximum three students, name and matriculation number must be visible in the submission). The submission should be in the form of a self-explanatory presentation (e.g. PDF, OpenOffice or PowerPoint) with a focus on interpretation or analysis. Results are only to be recorded to the extent that they are necessary to verify the statements. Further files (e.g. in TXT/ARFF/CVS) are not considered. The submissions are made in Moodle (one per group) and must be made by 23:59 on the deadline day at the latest.

1 Rule Learning: Application and Interpretation (2 P.)

In this introduction task you should explore and compare the results of different rule classifiers on different datasets in `Weka`. A few datasets are already included in the data folder of the installation path (e.g. `C:/Program Files/Weka-3-8-4/data`), further datasets can be downloaded on a separate homepage[3]. A short description of the dataset and its attributes is usually included in the beginning of the `arff`-file.
In `Weka` you will find `JRip`, a reimplementation of the most popular rule learner `Ripper`, and `ConjunctiveRule`, a learner which learns only one single rule. The latter must first be installed via the package manager. To do this, open it in the start menu via Tools > Package manager, search for `ensembleLibrary` and install the package. If the classifier `ConjunctiveRule` does not appear as an option when you choose the classifier, you might have to restart `Weka` before using it.
In this exercise you will compare `JRip`, `JRip` without pruning (by setting `usePruning=False`) and `ConjunctiveRule` on different datasets. Therefore, choose ten datasets and try to have as

[1] `https://waikato.github.io/weka-wiki/downloading_weka/`
[2] `https://www.knime.com/downloads/download-knime`
[3] `https://waikato.github.io/weka-wiki/datasets/`

much variance in the number of instances, the number of attributes and the attribute data types as possible.

1. Compare the number of rules, conditions and predicted classes of the resulting rule sets with respect to
   - the datasets
   - the rule classifiers
2. Is there a default rule for all algorithms? If so:
   - Which class is usually chosen as default rule?
   - How do you interpret the quality of the default rule?
3. On the basis of the previous subtasks, can you make a statement which of the datasets is the easiest or best to learn?
4. Perform a Friedman-Nemenyi test on the results and check whether there is a significant difference between the performance of the classifiers.

2 Noise and Pruning (2 P.)

Choose the dataset with the highest accuracy in the previous task and at least 50 instances. Disturb the class information in this dataset by adding different levels of noise (for example, 5%, 10%, 25%, 50%, 75%, 100%) with the filter `weka.filters.unsupervised.attribute.AddNoise` during pre-processing. Observe the accuracy and size of the learned trees on the original and the noisy datasets for the tree classifier `J48`

- with default parameters.
- without pruning (`unpruned=True` / `-U`) and minimum one instance per leaf (`minNumObj=1` / `-M 1`).

Experiment a little with the parameters `-C` (`confidenceFactor`) and `-M` for pruned trees and try to find the combination that gives the highest accuracy on the data disturbed with 10% noise.

Note: A x% noise level is created by replacing the example label at x% of all examples with a randomly selected label from one of the other classes. For two-class problems, you will notice that the performance at 100% noise is identical to the performance at 0% noise (Why?). In this case, adapt the bounds in an appropriate way (here 50% noise corresponds to random data).

3 Evaluation Methods (2 P.)

In this task different evaluation methods using `Weka` are to be applied and their results discussed. Apply the rule classifier `JRip` to five datasets (e.g. those from task 1) by first dividing each dataset into two equal stratified parts. Both sets are used for training and later on also serve as a test set for the other set. A stratified split can be achieved with the filter `weka.filters.supervised.instance.StratifiedRemoveFolds`. Set the parameter `-N` (`numFolds`) to 2 and `-F` (`fold`) to 1, apply the filter and save the first part of the dataset. Analogously save the second part using `-F` 2.
Note: Since you can not save in the data folder of Weka, you have to select a different directory.

1. Now train `JRip` on each of these training sets and evaluate the accuracy (percentage of correctly classified examples) of the resulting classifiers (without changing customized options like random seed) using:

   - 1x5 cross-validation
   - 1x10 cross-validation
   - 1x20 cross-validation
   - leave-one-out
   - the training set itself

   How do you assess the quality of the accuracy estimates obtained?

2. Repeat the previous task with the difference that you should now use a 10x10 cross-validation for evaluation. To do this, apply a 1x10 cross-validation ten times with ten different random seeds and average the achieved accuracies. Repeat the previous task another time using a 5x2 cross-validation. Compare the accuracy estimates obtained in this way with the estimates from the previous task. In your opinion, does a smart selection of random seeds lead to a better estimation?

3. Determine the accuracy on the other set by loading it via Set > Open file as the supplied test set. Then swap the roles of training and test set and compare the results. Assuming that these test sets are real use cases, how do you assess the estimates of the evaluation methods from the previous two tasks?

4. Select a sufficiently large dataset of a binary classification problem and compare the ROC curve and AUC for `J48` and `NaiveBayes`. The ROC curve can be created by right clicking in the result list and selecting Visualize threshold curve.