JOHANNES KEPLER
UNIVERSITY LINZ

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

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

---

Q&A Zoom Session:	**June 17, 10:15-11:45**
Deadline:	**July 12, 23:59**

The aim of the exercises is to gain practical experience in machine learning. For this we will use the data mining framework `Weka`[1]. 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.

6 Stream Mining: Hoeffding Trees (1.5 P.)

In this exercise you will compare the decision tree classifiers `J48` and `HoeffdingTree` which uses an incremental approach suitable for stream mining[2]. The latter one must first be installed via the package manager. To do this, open it in the start menu via Tools > Package manager, search for `massiveOnlineAnalysis` and install the package. If the classifier `HoeffdingTree` does not appear as an option when you choose the classifier, you might have to restart `Weka` before using it.

Create a new random dataset by choosing Generate in the main tab. Choose the generator `RandomRBF` and run it with the standard settings. Compare the classification accuracies and the time taken to build model of `J48` and `HoeffdingTree` on the generated dataset. Repeat the classification for more generated datasets of bigger size by increasing `-n` (`numExamples`) (for example, 100, 250, 500, 1,000, ..., 100,000). For the three biggest datasets, also compare the size of the trees and the total execution time (in seconds; you can find the start and end time in the log). What conclusions about the classifiers can be drawn from the performances?

[1] `https://waikato.github.io/weka-wiki/downloading_weka/`

[2] `https://www.cs.waikato.ac.nz/~abifet/MOA/API/classmoa_1_1classifiers_1_1trees_1_1_hoeffding_tree.html`

7 Distance-based Methods (1.5 P.)

Weka provides the classifier IBk which has implemented different nearest neighbor approaches discussed in the lecture. To test them, create a new random dataset by choosing Generate in the main tab. Choose the generator RandomRBF and set -c (numClasses) to 10 and -n (numExamples) to 25,000.
Note: If the execution time exceeds five minutes or falls below a second, you can adjust the number of instances accordingly.

1. Compare the classification results and the execution times with LinearNNSearch, KDTree and BallTree as nearestNeighbourSearchAlgorithm.

2. For the fastest search algorithm in the previous task, compare the accuracies for different numbers of nearest neighbors (in Weka: -K (KNN)). For the best three values for k, does a distance weight method further improve the accuracy?

8 Clustering (2 P.)

The dataset cities_greece contains the latitudes and longitudes of 9,882 cities in Greece [3]. In this exercise you will apply the algorithms SimpleKMeans and DBSCAN to cluster the cities. 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 optics_dbScan and install the package. If the clusterer DBSCAN does not appear as an option when you choose the clusterer, you might have to restart Weka before using it.
Load the dataset into Weka, go to the Cluster tab and retain the standard evaluation settings for all experiments (Cluster mode = Use training set).

1. Apply SimpleKMeans with $k = 2$ (in Weka: -N (numClusters)). Have a look at the computed clusters by right clicking in the result list and selecting Visualize cluster assignments (with longitude as x-axis and latitude as y-axis). Repeat the clustering with different values for k, compare the results and find an appropriate value for k. For this k, apply the cluster with three more different seeds. Does the result change? If yes, how much?

2. Now use DBSCAN to cluster the cities. If you use the standard settings of DBSCAN you will just get a single cluster. Adjust the parameters -E (epsilon) and -M (minPoints) to get a more appropriate outcome (Hint: You have to decrease epsilon). Compare the number of clusters and the number of unclustered instances. What is the best configuration in your opinion? Compute for the "best" number of clusters the outcome of SimpleKMeans and compare the results.

[3] http://www.math.uwaterloo.ca/tsp/world/gr9882.tsp