Assignment 3

Clustering: practical issues

The objective of this assignment is to address some typical problems related to the practical usage of clustering algorithms. We will consider again Iris data, but this time there are no experts available to label them. Therefore you want to use an automated approach to assign the flowers to different groups based on their characteristics.

TASK 0: warm up) Read the iris_clusters.csv data into Rapid Miner Studio. This file has been sampled (it contains 300 observations) and does not contain any label information. You can configure the Read CSV operator using the "Import Configuration Wizard", so that you can directly specify the role of the special *id* attribute at this stage.

TASK 1: k-Means clustering) You can start the analysis with the simplifying assumption that you know that there are three species present in the data. Knowing the number of clusters, you can thus try clustering your flowers using k-Means with k=3. Use Euclidean distance as your numerical measure.

K-means outputs both an extended dataset with a new "cluster id" attribute, indicating the cluster to which each flower has been assigned, and a cluster model. First, check the data, sort it by cluster to see how the flowers have been partitioned, and use a Scatter Matrix chart to visualize the clusters, coloring the observations by cluster id. Do the identified clusters correspond to the ones you expected to find, that is, the three Iris species you know from the lectures and from the first lab?

Now check the Cluster Model result tab (Description), and see how many records have been included in each cluster.

TASK 2: preprocessing) To improve the analysis, try to perform some data preprocessing before applying the k-Means operator. More specifically, add:

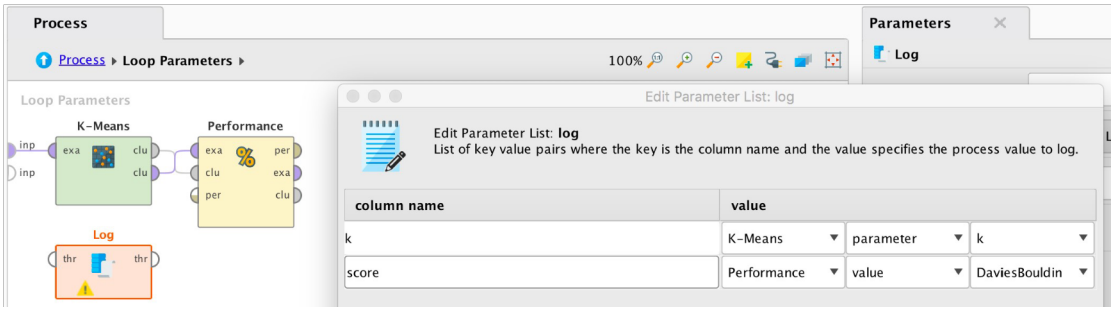
1) A min-max rescaling, scaling all the regular attributes to the interval [0, 1].
2) A Detect Outlier method, to automatically add a column to the dataset with a Boolean attribute indicating if the observation is an outlier. Here you can use the simple operator based on Distances, inspecting the 10 nearest neighbors of each observation and marking 5 of them as outliers.
3) A Filter Examples operator, to filter out the identified outliers.

Is it better to rescale before or after detecting and filtering out the outliers? (If you are unsure, try both and look at the statistics of the data, in particular the min, max and average values in each column.)

Use again a Scatter Matrix chart to visualize the clusters, coloring the observations by cluster id. Do the identified clusters correspond to the ones you expected to find, that is, the three Iris species? Then, check again the Cluster Model result tab. How many records have been included in each cluster (Description)? What are the coordinates of the three centroids representing the three clusters (Centroid table)?

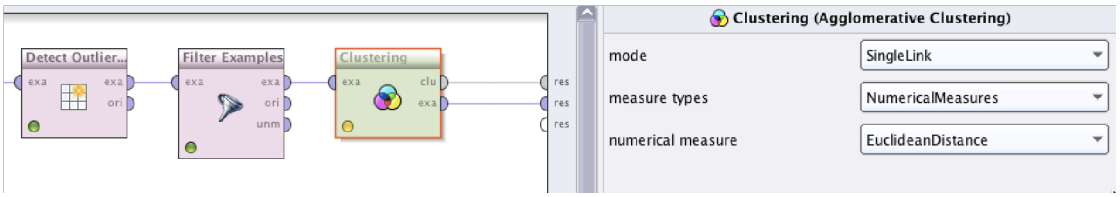
TASK 3: choice of k and internal evaluation) You will now put yourselves in the more realistic scenario where you do not know the number of clusters in advance. In this assignment we want to inspect the performance of the algorithm for each value of K. Therefore, you can use a Loop Parameter operator. Set the Loop Operator to try every value of K from 2 to 10 (do not use k=1, which would be useless).

Internally, the Loop Parameter operator must contain k-Means and a measurement of its performance. As you have no labels to check how well k-Means is performing, you need to use an *internal* (i.e., only based on your data) measure computing the cohesion and separation of your clusters. Here, you can use the Davies-Bouldin index, provided by the Cluster Distance Performance operator. This index has a lower value when the clusters are better separated. Also add a separate Log operator to store the value of the Davies-Bouldin index for each K



Execute the process. No Result tab will open (because Log does not send anything to the output), but if you open yourself the Result perspective you will find the data logged by the Log operator, that you can inspect using a Plot view. What is the value of K leading to the most compact and separated clusters?

TASK 4: Hierarchical clustering) Now you should try to use a different approach, in particular agglomerative hierarchical clustering. You can start using the SingleLink method (which is one of the options in the agglomerative clustering operator), which implements the approach you have seen during the lectures applying the MIN cluster similarity function. (We remind you that CompleteLink uses the MAX function, and AverageLink the average over all the cluster points.)



Inspect the result, checking the Description, the Dendrogram, looking at its Graph (tree) and Folder representations. Apart from the root of the dendrogram, trivially containing all the observations, how many records have been included in each of the two top (largest) clusters?

While very rich, you have probably noticed that a dendrogram can be difficult to evaluate because it does not specify any partition into a specific number of clusters. Therefore, you can now use a Flatten Clustering operator to cut the dendrogram into three clusters (we again assume that you know that there are three species represented in the data). Execute this process using SingleLink, CompleteLink and AverageLink and check which ones (if any) are more or less correctly identifying the three species, through a visual inspection of the results using a Scatter Matrix.

TASK 5: DB-Scan) As a last algorithm, you should try using DB-SCAN.

1) How many clusters does DB-SCAN find with default settings – epsilon 1, min points 5?
2) Leaving minPoints unchanged (5), can you manually find a value for epsilon leading to two clusters (plus noise)?

Now we can try to (semi-)automatically find a good value for the parameter epsilon. In particular, as seen during the lectures, you will use the method to estimate epsilon based on the distance to the N^{th} nearest neighbors. To do this, you can use the Data to Similarity operator, that computes the distance between all pairs of records. The result can be used to compute, for each record, the distance to its N^{th} nearest neighbor, and to inspect a plot with these distances sorted.

In the Result perspective, you can open the k-distances view, set k to an appropriate value and check the corresponding plot. You should notice some points of discontinuity in the curve, that is, points where the curve has a less regular shape, like a sudden increase/decrease of the distance: think of the effect of setting an epsilon threshold corresponding to those points, and check whether your intuition is correct by running DB-Scan with that epsilon.

[OPTIONAL] TASK 6: Document clustering) This last task is intended to be performed independently, so we only provide limited instructions. The knowledge acquired during the lectures and from the assignments should be sufficient for you to set up this process from the beginning to the end.

You will use the data *bbc.zip*, containing five folders (business, tech, sport, …), each containing several news articles about the same topic of the folder. The objective of the task is to see what clusters you would obtain by applying k-means to a vectorial representation of these documents. You can use k=5 (and can also try other values of k, or try to find the best value for k, if you have time – this is not requested), but of course you should not let the clustering algorithm know about the topics.

To perform this task, we recommend to install the Text processing **extension**, that can be installed from inside RapidMiner Studio. (While you install this extension, you may also be interested in looking at other available extensions, for your own interest.) As part of the extension, you will find an operator called Process Documents from Files, that automates the creation of a document collection from the input text files. This is a

sub-process, inside which you can add all the text processing operations you consider relevant (tokenization, stemming, …, as seen during the lectures).

CONCLUSION) With this and the previous assignments you have used a number of methods to preprocess the data and several data mining algorithms. We have tried to give you a "well guided" plan to emphasize the features of these approaches.
However, remember that when you apply these methods to new data you will need to take each of the steps that you have taken during these assignments and think if it is appropriate for the problem at hand. Here are a few additional questions, so that you can reflect about what you have done. You can think about them yourself, and also discuss them with your course and group mates.

1) In which of the tasks performed during this assignment would dimensionality reduction techniques be useful?
2) What would be the consequence of not removing outliers before using DBSCAN?
3) Although we know that there are three clusters in the data, the best clustering seem to contain only two groups of records. What does this mean? Is it our assumption that there are three clusters wrong? Or are the clustering algorithms not working well? Or, do you have any other explanation for this sub-optimal behavior (if you think it is sub-optimal)?