Information Retrieval & Text Mining

Lab sheet #4

Clustering

Overview

Objective:

1. Work with K-Means and Hierarchical clustering algorithm, and
2. Cluster evaluation

Exercises

- There are a 5 of exercises through out the slides
- Put answers to these in a separate word document called ExerciseSolutionLab#4-B000*nnnnn* (student number) and upload to moodle at the end of the lab.

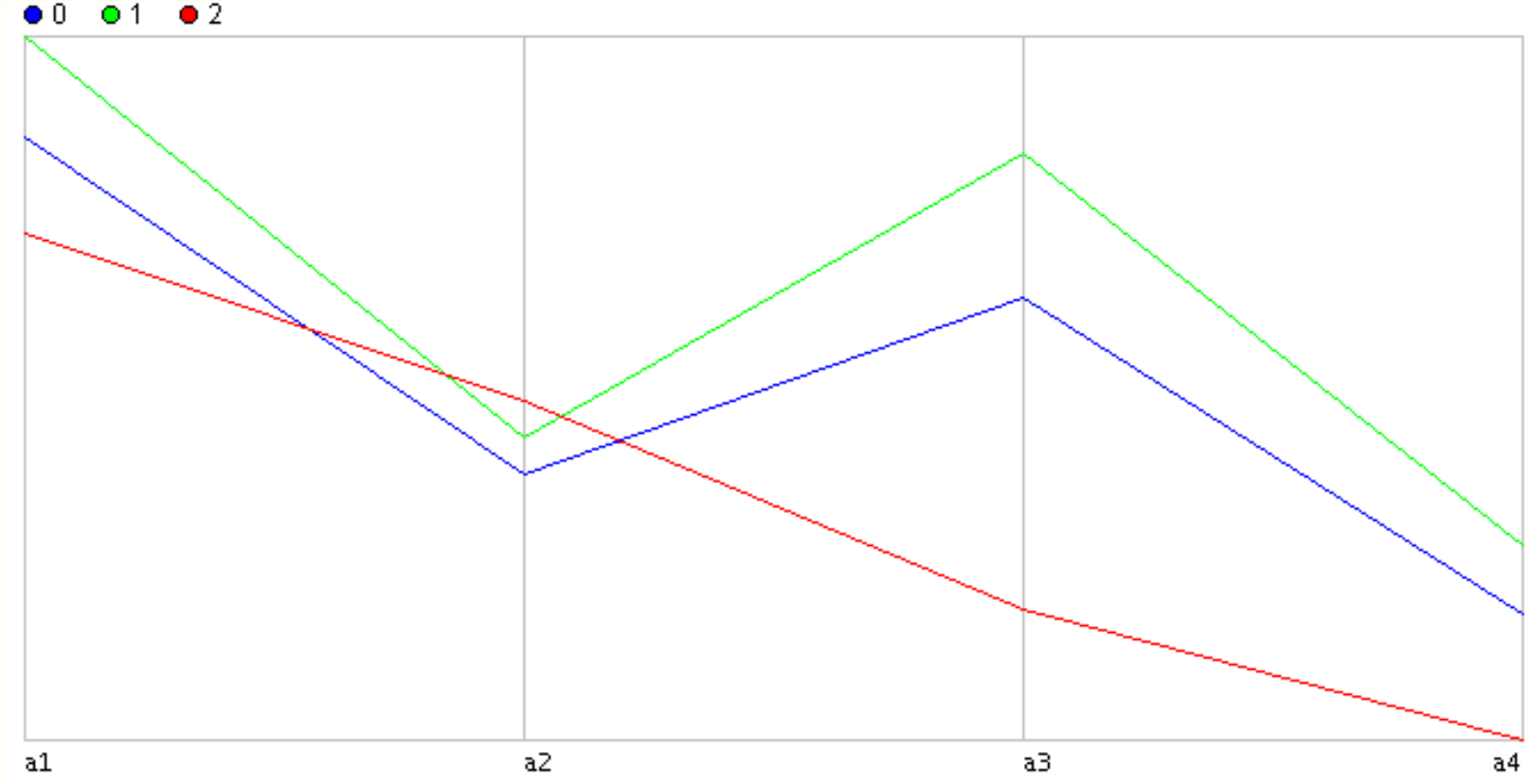
K-Means

- Open sample/processes/07_clustering/01_kmeans.xml which uses the k-means algorithm to cluster the Iris data set. Delete SVD reduction before running it, and output both ports.
- K-means parameters:
  - Add_cluster_attribute - adds a cluster ID to the dataset.
  - Add_as_label – give a clusterID a role of label to allow a classification algorithm attempt to predict the cluster ID. Change this to true (check box).
  - K – number of clusters
  - Max_runs – how many times to run k-means. Each run starts with an initial k, randomly chosen cluster centroids.
  - Determine good start values – don't select initial clusteres randomly.
  - Measurement types – select category of distance measures.
  - Divergence– select distance measure.
  - Max_optimisation steps – within each run, how many optimisation steps (regrouping rows into clusters) are preformed.
- Run the process.

K-Means

- The exampleset (retrieve) tab shows the original table with a cluster attribute added.
- The cluster model output has five views:
  - Text view shows the size of each cluster
  - Folder view and graph view list the object_ids in each cluster
  - Centroid table shows, for each cluster, the value of the cluster centroids for each attribute
  - Centroid plot view graphically illustrates how attribute values vary across each cluster, and is explained on the next slide . . .

Centroid Plot View



- Each line is a different cluster. a1 to a4 are the attributes.
- Looking at a1: the centre of cluster 1 has highest values of a1; the centre of cluster 0 has mid-range values in a1; the centre of cluster 2 has lower values for a1. Cluster 2 has the highest values for a2, but significantly lower values for a3 and a4.

The same plot, colour coded, can be found by selecting exampleset (retrieve), plot view, plotter = deviation, and color column=cluster.

Evaluating K-Means

- In this example, it is easy to evaluate the clusters because the dataset has a class label, so we knew in advance the dataset had three clusters, and we know which rows should be allocated to each cluster.

- To evaluate the process:
  - select the Exampleset tab,
  - select the plot view,
    - create a scatter plot of cluster in the X-axis and label in the y axis.
    - Set the color column to label (the actual class ID).
    - Add some jittering so you can see individual points.

Are the clusters similar to the original classes?

Exercise 1

- Start a new processing called lab4-clustering.
- Copy the process Documents operator, and it's internal processing steps from lab3-generateattributes.
- Set measure type to numeric, and numerical measure to cosine.
- Use k-means to cluster your document vector of 15 documents. Set k = 3.

- 1.1 Did it identify the three categories of documents? (view scatter plot of cluster V label)

- 1.2 For documents assigned to the wrong cluster, hover over their point on the scatter plot to get the row ID. Looking back at the document vector, can you see why these documents were allocated to the wrong cluster?

Agglomerative Hierarchical Clustering

- Open samples/processes/07_Clustering/ 04_AgglomerativeHierarchicalClustering in Rapidminers processes repository which uses Agglomerative Hierarchical Clustering to again cluster the Iris data set.
- Parameters:
  - Mode: how to select merge points. Choices are single link (min distance) or complete link (max distance) See lecture notes slide 38 for an explanation.
  - Measure types: select category of distance measures to use
  - *Distance category*: select distance measure.
- Run the process. The agglomerativeclustering results shows the entire tree, but does not give much information on the make up of the tree
- Return to edit mode. Add the 'flatten Clustering' operator to the end of the process. This flattens the tree to an optimal number of clusters. Number of clusters is set to 3.
- Run the processes again.

Agglomerative Hierarchical Clustering

- The cluster model shows the number of clusters selected.
- On the data table tab, do a scatter plot of cluster versus label to see the distribution of clusters across the labels. Increase the jitter a bit to view the spread of data points. Was is a good separation of the dataset?
- Try other values for measure and mode. Does it make the difference to the clusters?

Exercise 2

Continuing on from your work in exercise 1,replace k-means by agglomerative clustering, and add a flatten cluster, setting k again to 3.
Using the default for mode and measure type.
Set measure type to numeric, and numerical measure to cosine.

- 2.1 Did agglomerative clustering do better or worse the k-means?
- 2.2 Change mode to complete linkage. Does that improve the performance?

SOM

- SOM is not available as a clustering tool in RapidMiner, but is available as a transformation tool, replacing all attributes with just two – the X Y co-ordinates for each row in the SOM grid.

- See the SOM.rmp process on moodle for an example of joining the SOM output with the original dataset, and concatenating the X Y co-ordinates to get a cluster label. A 3D-scatter plot will show how the rows were grouped.
  - Set x-axis to SOM_0, y-axis to SOM_1, and color code the plot with label.

  - Does a 2 x 2 grid (grid size =2) work better?

Exercise 3

Continuing on from your work in exercise 2,use a SOM to organise the 15 documents in a 2 x 2 grid (grid size = 2).
Plot the result with a scatter plot, setting x-axis to SOM_0, y-axis to SOM_1, and colour code the plot with a label. Add a little jitter.

- 3.1 How did the performance of SOM compare with agglomerative clustering with complete linkage?

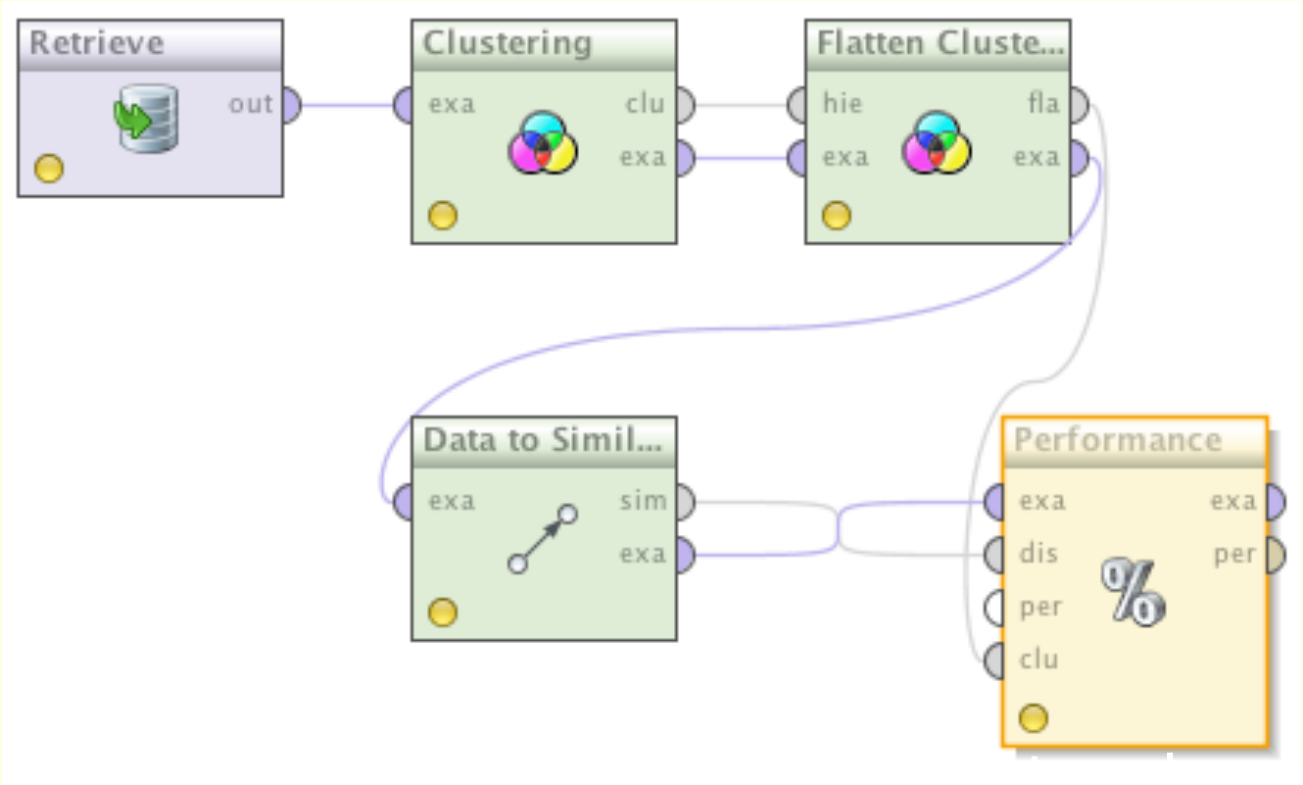
Cluster performance operators

Located under evaluation/cluster, there are a number of operators for evaluating clusters.

1. Cluster distance performance provides two values: a Davies Bouldin index (see lecture notes for details) and an average distance to cluster centre, calculated across all clusters. The operator can only be used if the clustering algorithm calculates a cluster centre such as k-means.
2. Item distribution performance gives a measure from 0 to 1 of how evenly items are distributed across clusters. A value close to 0 means uniform distribution, a value close to 1 means unequal distribution.

Performance Measures

3. Cluster Density Performance gives the average distance between objects in the same cluster. It must be used in conjunction with an operator which calculates distances between objects such as Data to Similarity, as shown below.



Performance Measures

4. Map Clustering on Labels can be used if the dataset has a class label. It will attempt to map clusters into classes, choosing the best fitting pairs.

This is also referred to as semantic evaluation.

5. Where clusters are well defined, the output of hierarchical clustering - agglomerative paired with flatten clustering - can in itself give useful insight into the number of clusters in the dataset.

Objective cluster evaluation: evaluating K-Means

- Return to the first process in the labsheet: using k-mean to cluster the iris dataset (sample/processes/07_clustering/01_kmeans.xml). Remove the SVD reduction operator, and save the process to your own repository if its not already there, calling it lab4-clusterEvaluation.
- Under Evaluation / Performance Measurement / Clustering, add the Cluster Distance Performance operator.
  - This will give the average distance between objects in each cluster
  - The Davies-Bouldin index is calculated from both within cluster distances and between cluster distances.

Objective cluster evaluation: evaluating K-Means

- Open ../sample/prcesses/07_clustering/ 09_KMeansWithPlot.
- This process clusters the iris dataset, and then evaluates the clusters generated using a cluster distance measure.
  - Parameteriterisation loops through each combination of parameter settings specified in the **edit list** parameter. In this example it is looking at different values of **k**.
  - Double click to view the inner operators: kmeans, followed by a cluster_distance_performance operator (renamed to evaluation) followed by a process log to log the output of each loop. For each value of k it outputs two measures: average distance within each cluster, and DaviesBouldin which is a ratio of within and between cluster measures.
- Output the result from ParameterIterator and run the process. . . .

Objective cluster evaluation: evaluating K-Means

- When the process is complete, go to the results perspective, and look at the ProcessLog.
- The table view shows the value of Davies Bouldin (DB) and average cluster distance (w) for each value of k.

Exercise 4

4.1 What value of K gives the smallest DB value?
4.2 Looking at a plot of the iris dataset, and ignoring the class label, does two clusters make sense?

Replace the retrieve operator with your process documents operator to create the dataset of 15 document vectors.
4.3 According to a DB evaluation, how many clusters are in the dataset?

Defining a cluster

- One of the most time consuming tasks in cluster analysis is understanding the make up of each cluster, and so evaluating the clusters identified by the cluster algorithm, particularly of unlabelled dataset
- One technique to do this is as follows:
  - Run the clustering algorithm, and add the clusterID to the original dataset
  - Make this clusterID the class label
  - Run a classification algorithm to predict the class label using an algorithm which gives information on how the decision was made, such as a decision tree. This is done in the following process:
- Open ../sample/processes/07_clustering/07_clusterclassification.xml which adds a decision tree to define the make up of each cluster.
- Run the process to review the decision tree produced.

Working with a dataset you are not familiar with . . .

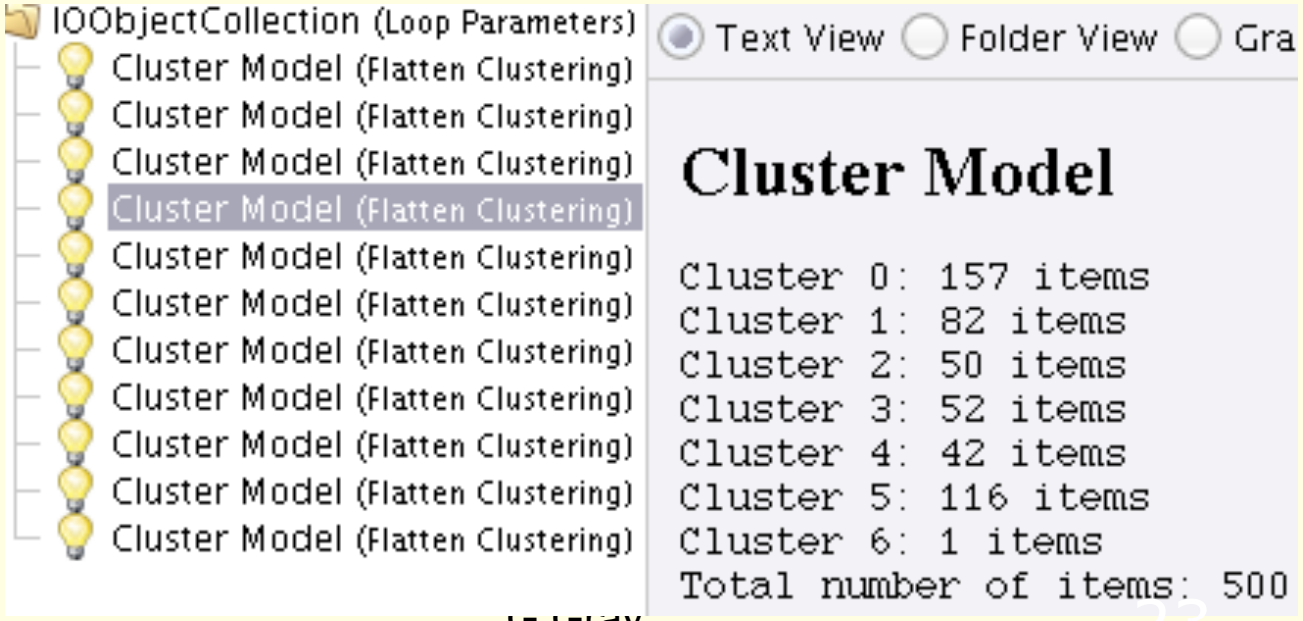
The following two experiments give examples of clustering processing and model evaluation

Experiment 1: looking at the output from agglomerative clustering WITHOUT using performance operators
Dataset: Globular clusters.csv (generated dataset with a class label)

- Load globularClusters.csv into your repository, and retrieve it into a new process. Run the process to explore the dataset.
- Connect this to a 'loop parameter' operator, to test the output for various numbers of clusters.
- Nested within loop parameter include:
  - agglomerative clustering (hierarchical),
  - followed by flatten clustering.
    - Recap: flatten cluster will output the hierarchy level that corresponds to the number of clusters needed, which is set as a parameter. Leave this at the default value for now.
- Return back to loop parameter, and select the parameter: edit parameter settings.
  - Our objective here is to try various values for 'number of clusters' in the flatten clustering operator.
- Under 'operator' select 'Flatten Clustering'
- Under 'Parameters' select 'number_of_clusters'
- Under Grid/Range set min to 1; set max to 10; set steps to 10 (increments by 1 each time).

Experiment 1 continued . . .

- Make sure you are outputting the 2nd parameter from loop parameter – the cluster model
- Run the process and look at the output (IOObjectCollection (loop parameters))
- The first cluster model(flatten cluster) shows the results for one cluster; the second shows the results for two clusters etc.



Experiment 1 continued . . .

- As you work through each output, you will see initially all cluster sizes are large, until you get to cluster 6 (7 clusters), when a cluster of size one is created.

- This denotes the point at which there is no longer an inter cluster gap which is larger than a intra cluster gap. This may be caused by a once off outlier, or may denote the start of splitting clusters.
- As you continue to work through the models, all additional clusters added are size one, indicating the data set has 6 well defined clusters, or has a number of individual outliers.

Experiment 1 continued . . .

- The default value for mode in agglomerative clustering in single linkage which finds non globular structures and is sensitive to noise.
- Change the mode to average linkage, and run the process again.
- Does this change the number of clusters found?

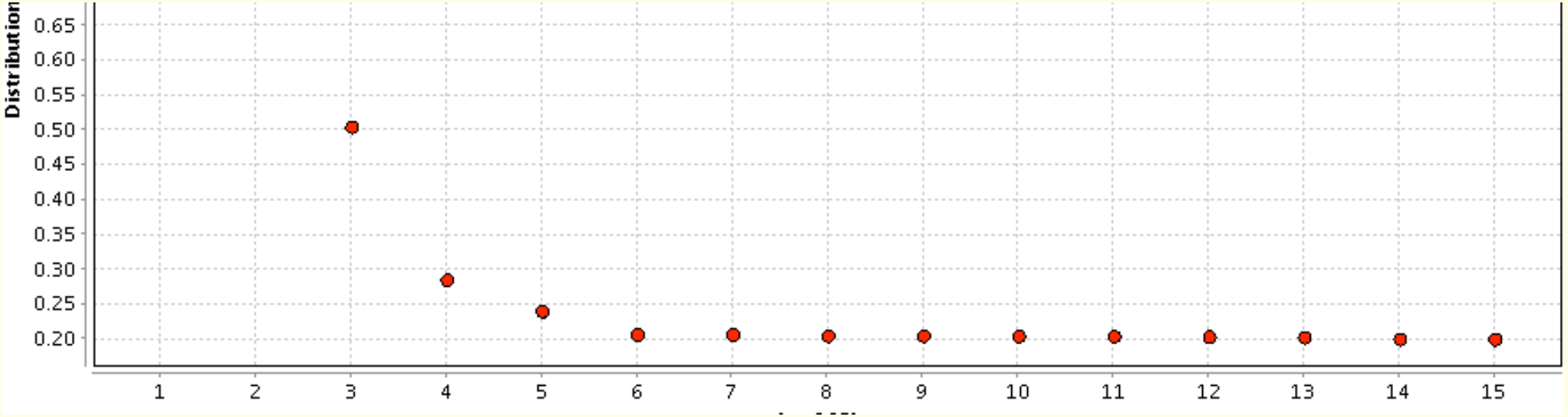
Selecting split points using average linkage suggest about 8 clusters.

Experiment 2: looking at the output from agglomerative clustering WITH performance operators
Dataset: Globular clusters.csv (generated dataset with a class label)

- Add TWO performance operators to the process, Item Distribution and Cluster Density.
- Connect Item Distribution Performance to flatten clustering, passing in the 'fla' cluster model parameter as input.
- Next connect a log operator to the performance operator to log the distribution calculation for each loop of the process (i.e. each value for 'number_of_clusters)
- Select the log operator, and select it's edit list parameter.
- Click 'add entry', give this first column the name 'number_of_clusters', select the 'flatten clustering' operator in the first drop down box, select parameter in the second, and finally select 'number_of_clusters' in the 3rd drop down box.
- Click 'add entry' again. Call the column distribution, and select 'performance', 'value', 'item_distribution' in the three drop down boxes.

Experiment 2: looking at the output from agglomerative clustering WITH performance operators

- Run the process. Select the log tab, and plot number_of_clusters against distribution.
  - You will see item_distribution converges at about 6 clusters.



Experiment 2: looking at the output from agglomerative clustering WITH performance operators

Next, we are going to add a second performance operator which will illustrate the MultiplyIO operator.

- Bring cluster density performance onto the process window.
  - The second input parameter is a similarity measure, which means it needs the similarity measure between all points in the data set.
  - Add the data to similarity operator to the process window (found under modeling/similarity computation).
- Connect 'exa' from flatten clustering to 'exa' in data to similaity.
- Connect 'sim' in data to similarity to 'dis' in cluster density performance, and 'exa' in data to similarity to 'exa' in cluster density performance.

Experiment 2: looking at the output from agglomerative clustering WITH performance operators

- Finally this operator also needs the cluster model from flatten clustering. However this is currently connected to item distribution operator, so we will need two copies of the model output
- Attempt to connect 'fla' in flatten clustering to 'clu' in cluster density performance. Rather than selecting 'Disconnect ports and connect', click the drop down box to select 'Insert IO multipliers as needed'.
- The multiply operator which has been added makes multiple copies of the cluster model, passing one to each performance operator.
- The final step is to log the output of the cluster density performance operator.
  - Connect density performance to the log operator
  - Edit the parameter list, and add a new entry called 'density' with drop down box values of performance (2), value, cluster density.

Experiment 2: looking at the output from agglomerative clustering WITH performance operators

Dataset: Globular clusters.csv (generated dataset with a class label)

- Run the process, and again plot the outputs in the log tab, comparing cluster density with item distribution.

Exercise 5:

5.1 Do they tell you the same information?

5.2 Will they always give the same results?

5.3 Does changing MODE effect the output?

Summarise what we have learnt about this dataset . . . .

- At this point, we have evaluated the number of possible cluster in globular cluster.csv as being in the order of 5 to 8, with algorithms favoring globular clusters suggesting about 6 clusters, and algorithms for less regular structures suggesting about 8. Is that a reasonable assumption?

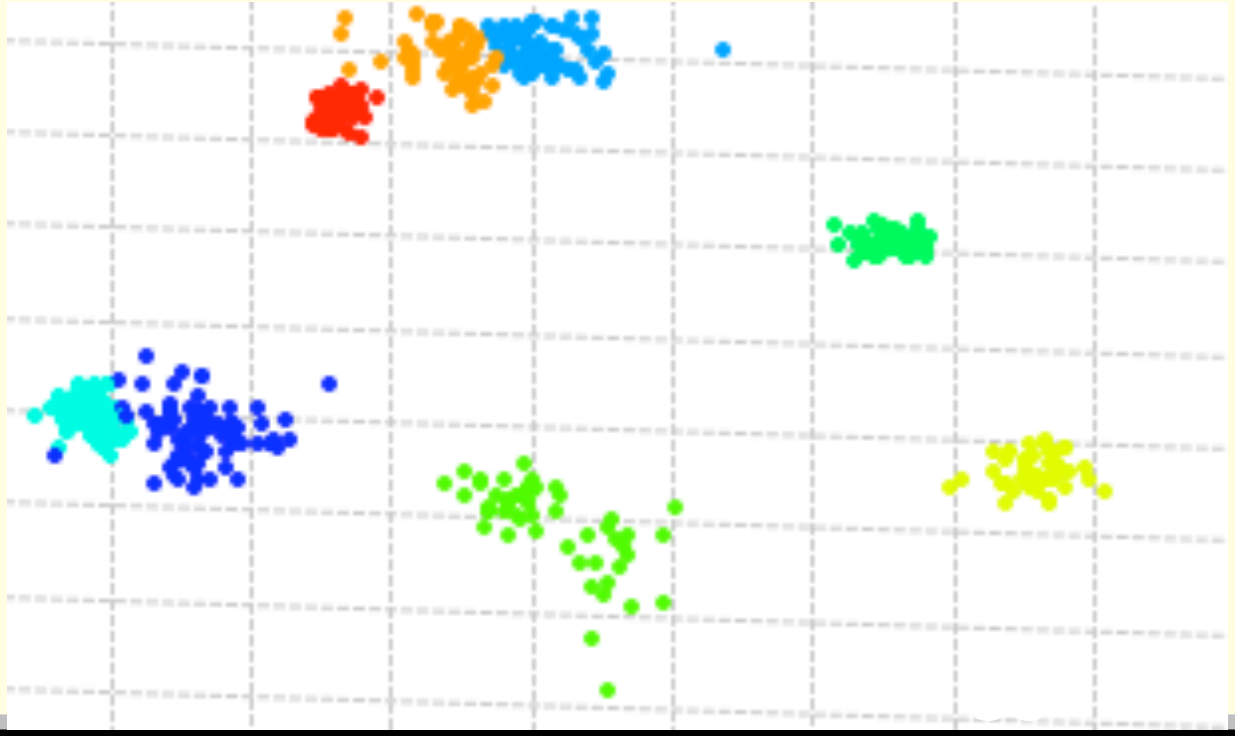
- The next step is to estimate the make up of each cluster, to see if the rows are grouped in a way that makes sense. There are a number of options here:
  - Use a centroid plot to look at the attribute values in each cluster
  - Predict the cluster label using a classification algorithm.
  - Map clusters to class labels (can only be used if the original dataset has a class label)

Map clustering on Labels

- This operator will recommend a mapping from clusters to classes/label values for datasets with a class label (such as globular cluster.csv)
- The operator requires the same number of clusters as classes. Globular cluster.csv has 8 classes.
- Start a new process retrieving globular cluster.csv .
- Add a kmeans operator, setting k to 8, and selecting 'add as label'.
- Add a Map Clustering on Labels operator
- Run the process and look at the resulting example set.
  - The data view shows the actual label (label), the cluster K-means allocated the row to (cluster) and the label the row was mapped to (prediction label), which, if mapped correctly, should be the same as label.
- Plot label versus prediction label and add jitter

Map clustering on Labels

- When I ran this process, rows from cluster 3 were split across two clusters by k-means, while rows in cluster 0 and cluster 5 were merged into one cluster.
- A scatter 3-D color plot will illustrate this more clearly. Set X, Y, and Z to atr1, atr2 and atr3. Alternate between label and cluster to view how k-means split the actual cluster.



Predict cluster label

- Start a new process, retrieving globular_clusters.csv again.
- Connect a k-means operator, setting k to a value between 6 & 8. Tick the parameter box 'add as label'.
- Next connect a 'set role' operator, setting cluster to the label (ignore actual label for now).
- Finally connect a decision tree operator, which will attempt to predict the cluster ID, indicating the make up of each cluster.
- Run the process to view the resulting tree.