Weka – Hierarchical Clustering & Attribute Transformation (3)

Lab8 (in-class): **12** DIC 2016 -13:15-15:00 (CHOMSKY)

ACKNOWLEDGEMENTS: INFORMATION, EXAMPLES AND TASKS IN THIS LAB COME FROM SEVERAL WEB SOURCES.

Learning objectives

In this assignment you are going to:

- apply hierarchical clustering, as implemented in weka: *HierarchicalClusterer*
- transform attributes: discretization

Task 1: Warming-Up – HierachicalClusterer

HierarchicalClusterer implements agglomerative (bottom-up) generation of hierarchical clusters. Several different **link types** (see Fig. 1 and Witten et al.,2011:275.276), which are ways of measuring the distance between clusters (*inter-cluster distance*), are available as options.

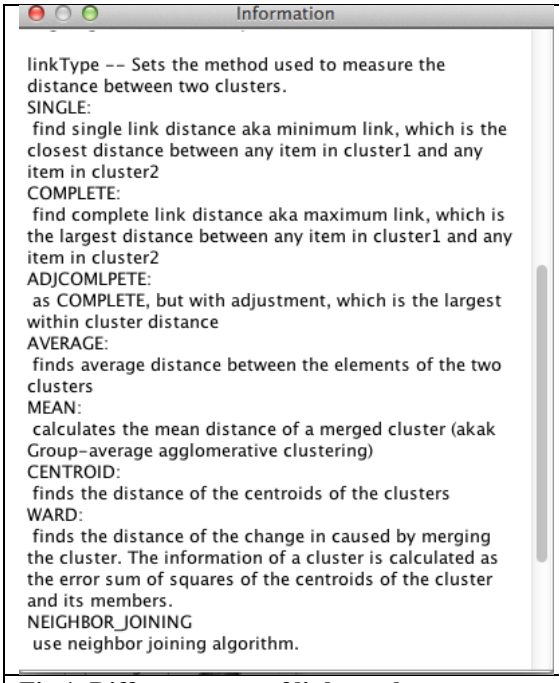


Fig 1. Different types of linkage that measure the inter-cluster distance

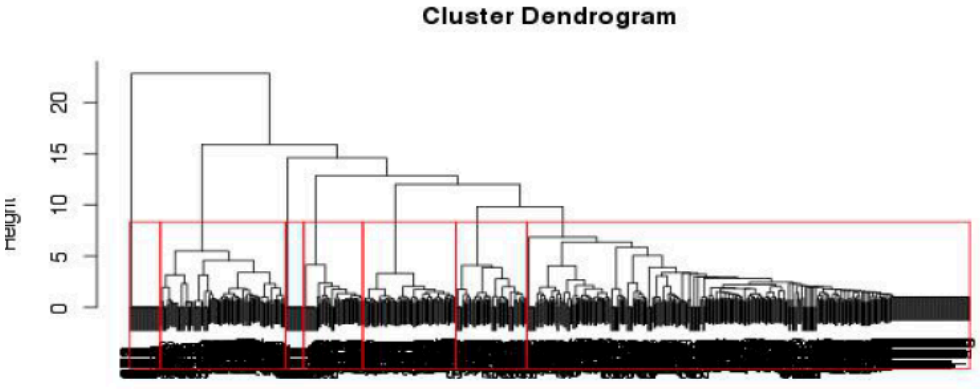
Hierarchical clustering builds a tree for the whole dataset, so large datasets might cause memory space errors.

Download and upload the glass.arff dataset in weka:
<http://stp.lingfil.uu.se/~santinim/ml/2016/Datasets/glass.arff >

First normalize all the numeric values in the dataset into [0,1]. Then choose HierarchicalClusters. Since this datset has 6 classes, we set numClusters to 6. To save time, we set printNewick[1] as False. After you run the clustering algorithm, you can right-click the clustering result and check its hierarchical tree by clicking Visualize Tree.

Since the performance of HierarchicalClustering is not good, we could run k-means algorithm on the same dataset and compare their performance. Do not forget to set the number of clusters to 6.
Save the clustering results as *glass_kmeans_result.arff* file, ie. right-click on the k-means entry, choose Visualize cluster assignment, then save with the name above. Reopen this dataset with Weka. Since clustering results are saved as the last column, they are considered as class labels for the datset. The original class labels are considered as a feature of the dataset. If you trust the clustering you might want to use this newly created dataset for supervised learning.

Remember: Deciding the number of clusters when you do not have any knowledge or intuition of the patterns in your data is quite difficult. One rule of thumb: look at the dendrogram, where height change looks big, cut off the tree.



Run hierarchicalClusterer on the original dataset, this time choose *Use training set* as Cluster mode. Visualize the dendrogram.

Q1: Where would you cut this dentrogram? how many clusters would you create for this dataset, if you did not new in advance the number of classes?

[1] The Newick Standard for representing trees in computer-readable form makes use of the correspondence between trees and nested parentheses.

Task 2: Hierachical Clustering vs. k-Means

For testing accuracy and efficiency of simple k-means and hierarchical clustering algorithms, uses the Iris dataset, the Diabetes dataset and the junk dataset. Run the two unsupervised algorithms, and compare their accuracy on these 3 datasets. Fill up this table:

	k-means running time (sec)	hierarchical clustering running time (sec)	k-means accuracy (%)	hierarchical clustering accuracy (%)
iris				
diabetes				
junk				

Q2: how do they perform?

Task 3: Discretization & J48

Discretization is the process of tranforming numeric data into nominal data, by putting the numeric values into distinct groups, which lenght is fixed. Common approaches:
• Unsupervised (Equal-width binning or Equal-frequency binning
• Supervised (classes are taken into account). Generally speaking, unsupervised transformations are "class blind,", while supervised ones take the class value of the instances into account.

Weka's main unsupervised method for discretizing numeric attributes is *weka.filters.unsupervised.attribute.Discretize*. The main supervised technique for discretizing numeric attributes is *weka.filters. supervised.attribute.Discretize*.

Here we examine the effect of discretization when building a J48 decision tree for the data in ionosphere.arff
<http://stp.lingfil.uu.se/~santinim/ml/2016/Datasets/ionosphere.arff >

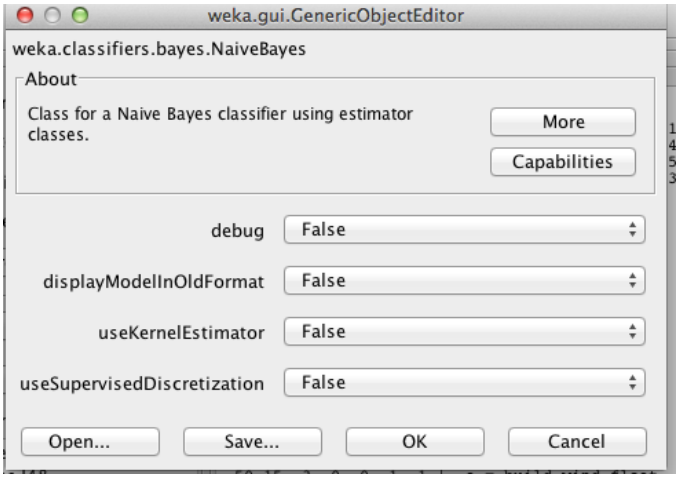
This dataset contains information about radar signals returned from the ionosphere. "Good" samples are those showing evidence of some type of structure in the ionosphere, while for "bad" ones the signals pass directly through the ionosphere. Begin with unsupervised discretization.

Q3: For J48 , compare cross-validated accuracy and the size of the trees generated for
(1) the raw data,
(2) data discretized by the unsupervised discretization method in default mode, and
(3) data discretized by the same method with binary attributes (select appropriate option in the filter).

Q4: Repeat with the junk dataset: what happens?

Task 4: Discretization: NaiveBayes

By default, Weka's NaiveBayes classifier assumes that the attributes are normally distributed given the class. You should override this by setting **useSupervisedDiscretization** to true using the Generic Object Editor window.



This will cause NaïveBayes to discretize the numeric attributes in the data with a supervised discretization technique. In most practical applications of NaïveBayes , supervised discretization works better than the default method. It also produces a more comprehensible visualization.

In a previous lab we saw that naivebayes with default value was statically significant worse that the other classifier (see Lab6, NaïveBayes, default = 79.28% accuracy).

Q5: What happens if we apply supervised discretization here?

----the end------