Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Dr. Heiko Niedermayer
Cornelius Diekmann, M.Sc.

Netzwerkanalyse Sommersemester 2014
Assignment 5

Task 1 PCA and Clustering

Use the code clustering.py and pca.py for this assignment. Generate two data sets that contain 3-dimensional data. One data set should be based on circular clusters. For the other data set use polynomials and roots between the dimensions. Add a bit of noise to all data.

a) Apply PCA on the two data sets. Reduce the dimension by one (with lowest eigenvalue). Plot this data. Colorize points of the same cluster in the same color.

b) Now apply k-means or DBSCAN clustering on the data set. Plot it again with the cluster membership determined by the clustering algorithm.

c) What happens if you reduce the dimension with the highest eigenvalue instead of the lowest?

Task 2 Clustering

Use the code clustering.py for this assignment.

a) Generate a data set good for k-means and bad for DBSCAN. Show this by applying the methods and plotting their results.

b) Generate a data set good for DBSCAN and bad for kmeans. Show this by applying the methods and plotting their results.

c) DBSCAN extends the clusters as long as the density is higher than *minpts* per *eps*. Modify it so that clusters extend only to regions as long as the density is similar (with delta=0.25, density in $[1 - delta, 1 + delta] * densityFirstNodeOfRegion$). Test your method on data where 2 clusters overlap, one with high density, one with lower.

Task 3 Decision Tree Learning (Supervised)

Use the code dectree.py and dectreeexample.py for this assignment.

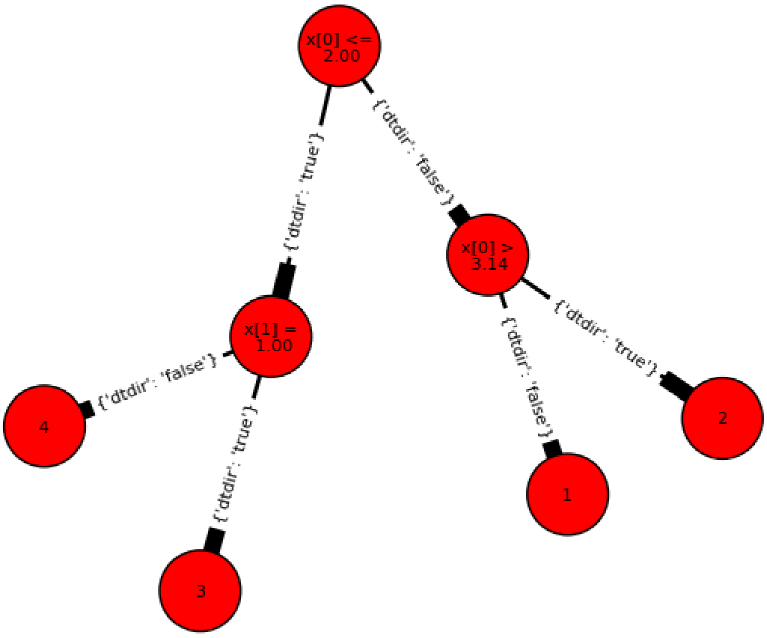

Figure 1: Generate this Decision Tree. Leave nodes contain the output class identifyers, here 1,2,3,4.

a) Modify dectreeexample.py so that the tree G corresponds to the one from Figure 1. Verify with dectree.evalDTreeValue(G, input vector) that the subsequent test values are correct:
$f([3.0, 0]) = 1$
$f([3.2, 7.0]) = 2$
$f([1.4, 1.0]) = 3$
$f([1.2, 7.0]) = 4$.

b) To learn data, we assume that each data vector contains the input fields and as last field the output value (class).

c) Use the dectree.bootstrapSampleFromData method to generate training set and test set. Apply dectree.trainDT to train the tree. Apply dectree.evalCurTree to evaluate the tree on a data set. Vary the parameters like size of the training set and maxRounds and minGain from the learning function. What do you observe on training and test set.

d) Plot the data set and the clusters from the tree to visually evaluate the clustering.