Lecture 23: Cluster Analysis

Winter 2015

Ref: Matlab Statistics Toolbox documentation

23.1 Introduction

We often want to group data into clusters, which may represent a true underlying structure (e. g. grouping the animals in a zoo into species or ice crystals into habits) or may just be a pragmatic grouping of continuously varying data into representative categories (e. g. weather states).

Often we choose our clusters ahead of time, called *supervised classification*, e. g. compositing, in which we use some predetermined ranges of an index parameter to define a cluster.

Cluster analysis is a form of *unsupervised classification*, when we don't have a predetermined set of known categories into which to group the data. There are diverse clustering algorithms with different intended purposes, including *hierarchical clustering* (a tree-like structure of nested clusters) and *partitional clustering* of the data by a single set of clusters. Some clustering schemes allow objects to partially belong to multiple clusters.

23.2 K-means clustering

One partitional clustering scheme is **k-means clustering** (Matlab: **kmeans**). The user specifies a number K of clusters, and the scheme seeks a set of K cluster centroids, which minimizes the sum of the squared distances between all m-dimensional data objects and their nearest centroids.

1. Select a set of K points as initial centroids. Using K random data samples is the Matlab default, but it is more robust to use the 'cluster' option for 'start', which randomly selects 10% of the data, clusters it, and uses those centroids as the initial guesses for the full clustering.

2. Assign each m-dimensional data sample to its closest centroid.

3. Recompute the centroid of each cluster.

Steps 2 and 3 are iterated to approximate convergence.

The script **cluster_cities** illustrates K-means clustering, first on a 1D dataset made from PC1 of the **cities** dataset, then on the 9-dimensional full dataset.