SMAI

Assignment 5
Dimensionality Reduction and Clustering
Report

Saurabh Jain
201301128

Answers to the questions

1. k-medians clustering

In statistics and data mining, k-medians clustering is a cluster analysis algorithm. It is a variation of k-means clustering where instead of calculating the mean for each cluster to determine its centroid, one instead calculates the median. This has the effect of minimizing error over all clusters with respect to the 1-norm distance metric, as opposed to the square of the 2-norm distance metric (which k-means does.)

This relates directly to the k-median problem which is the problem of finding k centers such that the clusters formed by them are the most compact. Formally, given a set of data points x, the k centers ci are to be chosen so as to minimize the sum of the distances from each x to the nearest ci.

k-medoids

The k-medoids algorithm is a clustering algorithm related to the k-means algorithm and the medoidshift algorithm. Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k-means algorithm, k-medoids chooses datapoints as centers (medoids or exemplars) and works with an arbitrary metrics of distances between datapoints instead of This method was proposed in 1987 for the work with norm and other distances.

k-medoid is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters known a priori. A useful tool for determining k is the silhouette.

It is more robust to noise and outliers as compared to k-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances.

A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. i.e. it is a most centrally located point in the cluster.

2. You tend to use the covariance matrix when the variable scales are similar and the correlation matrix when variables are on different scales.

Using the correlation matrix standardizes the data. In general they give different results. Especially when the scales are different.

3. Perhaps the most widely used algorithm for manifold learning is kernel PCA. It is a combination of Principal component analysis and the kernel trick. PCA begins by computing the covariance matrix of the $m \times n$ matrix $\mathbf{X}$

$$C = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}}.$$

It then projects the data onto the first k eigen vectors of that matrix. By comparison, KPCA begins by computing the covariance matrix of the data after being transformed into a higher-dimensional space,

$$C = \frac{1}{m} \sum_{i=1}^{m} \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^{\mathsf{T}}.$$

It then projects the transformed data onto the first k eigen vectors of that matrix, just like PCA. It uses the kernel trick to factor away much of the computation, such that the entire process can be performed without actually computing $\Phi(\mathbf{x})$. Of course Φ must be chosen such that it has a known corresponding kernel. Unfortunately, it is not trivial to find a good kernel for a given problem, so KPCA does not yield good results with some problems when using standard kernels.

KPCA has an internal model, so it can be used to map points onto its embedding that were not available at training time.