KD–Tree Based Clustering for Gene Expression Data

Damodar Reddy Edla
National Institute of Technology, Goa, India

Prasanta K. Jana
Indian School of Mines, India

Seshaiah Machavarapu
Tata Consultancy Services, India

INTRODUCTION

K-means is one of the widely researched clustering algorithms. But it is sensitive to the selection of initial cluster centers and estimation of the number of clusters. In this chapter, we propose a novel approach to find the efficient initial cluster centers using *kd*-tree and compute the number of clusters using joint distance function. We have carried out excessive experiments on various synthetic as well as gene expression data. Dunn validity index is used to examine the quality of the clusters in case of multi-dimensional gene expression data. The experimental results are compared with the existing techniques using the Dunn validity index and number of iterations.

BACKGROUND

Clustering (Jain, 1988) is a well-known data mining technique to group the given data into homogeneous subsets called clusters. It is a commonly used data segmentation tool in various fields such as medicine (Villmann & Albani, 2001), economics (Garibaldi et al., 2006), computational biology (Madeira & Oliveira, 2004) and geology (Parks, 1966). There are two main categories of clustering, namely, hierarchical and partitional. In hierarchical clustering, the clusters are formed recursively by using agglomerative mode or divisive (top down) mode (Jain, 1988). Unlike hierarchical, in partitional clustering algorithms all the clusters are formed concurrently as a partition of the data and do not impose a hierarchical structure (Jain, 1988). Significant amount of research has been drawn and number of algorithms (Sibson, 1973), (Defays, 1977), (Al-Daoud & Roberts, 1996), (Lu, 2008) have been developed using these models. However, partitional methods have extensively been adopted over hierarchical because of their fastness and simplicity.

MAIN FOCUS

K-means (MacQueen, 1967) is a well known partitional clustering algorithm. The clusters of *K*-means are represented by iteratively-changing centroids chosen randomly. *K*-means find the squared distances between these centers and the given objects to assign the objects to their closer centroids. However, *K*-means has the demerit of the random selection of initial cluster centers. It also has the problem of estimating the number of clusters. Many researches proposed various methods to overcome these problems, a good review of which can be seen from Jain (2010) and Al-Daoud and Roberts (1996).

Motivated with them, we propose an algorithm to enhance the *K*-means clustering. This algorithm has two phases. In the first phase, we use the approach of *kd*-tree to find the efficient

DOI: 10.4018/978-1-4666-5202-6.ch122

initial seeds for *K*-means clustering. Then, in the next phase we use the joint distance function (Butenko, Chaovalitwongse, & Pardalos, 2009) to find the right number of clusters. We have carried out excessive experiments on various synthetic as well as gene expression (GE) data. The results are compared with classical *K*-means (MacQueen, 1967), improved *K*-means (Geraci et al., 2007), CCIA (Khan & Ahmad, 2004) and fuzzy *C*-means (Bezdek, Ehrlich, & Full, 1984) algorithms. Dunn validity index (Halkidi, Batistakis, & Vazirgiannis, 2001) is used to examine the quality of the clusters of multi-dimensional data. Finally, the proposed method is compared with the existing methods using number of iterations.

Related Work

Higgs et al. (1997) and Snarey et al. (1997) developed a method using MaxMin algorithm to choose a subset of the original database as initial cluster centers. Tou and Gonzales (1974) recommended a method based on the distance between the successive seeds and a threshold value. But this method entirely depends on the order of the points in the database. Linde, Buzo, and Gray (1980) proposed a method based on Binary Splitting (BS) which splits the cluster centre using a small random vector. This method is computationally expensive. Kaufman and Rousseeuw (1990) developed a method which is based on the reduction in the Distortion. Here the seeds that increase the reduction in the distortion are chosen for the next step. Huang and Harris (1993) projected a method called Direct Search Binary Splitting (DSBS). Here the splitting is done efficiently through the Principle Component Analysis (PCA) based on the vector of Linde, Buzo, and Gray (1980). Thiesson et al. (1997) introduced a method that depends on the mean value of the whole given data set which creates a set of *K*-points around the mean of the data. An enhanced *K*-means algorithm has been proposed by Redmond and Heneghan (2007) using the *kd*-tree approach. In this paper, the density information of the leaf buckets in the *kd*-tree is used to locate the *K*-cluster centers by finding the *K* leaf bucket centroids far away from each other and have large densities. This is similar to the KKZ method in which the densities are not considered along with the distances. This method is unable to deal with the outliers.

Preliminaries

1. **K-means Clustering:** *K*-means algorithm (MacQueen, 1967) finds the clusters by partitioning the data to minimize the squared error between the centroids of the clusters and the given points. Let C_k denote the k^{th} cluster of the data: $\{x_1, x_2, \ldots, x_n\}$. Then if μ_j is the mean of the cluster C_j, the squared error between μ_k and the point x_i within C_j is as follows (Jain, 2010).

$$S(C_j) = \sum_{x_i \in C_j} || x_i - \mu_j || \quad (1)$$

The aim of *K*-means is to reduce the sum of squared error for all the '*K*' clusters. i.e., to minimize *S*(*C*).

$$S(C) = \sum_{j=1}^{K} \sum_{x_i \in C_j} || x_i - \mu_j || \quad (2)$$

The algorithm (Bandyopadhyay & Maulik, 2002) is as follows.

Step 1: Select *K* initial cluster centers $c_1, c_2, \ldots, c_K$ randomly from the given *n* points $\{x_1, x_2, \ldots, x_n\}$, $K \leq n$.

Step 2: Assign each point x_i, $i = 1, 2, \ldots, n$ to the cluster C_j corresponding to the cluster center c_j, for $j = 1, 2, \ldots, K$ iff

$$\left\| x_i - c_j \right\| \leq \left\| x_i - c_p \right\|,$$

$p = 1, 2, \ldots, K$ and $j \neq p$.