Chapter 22
Cluster Analysis in R With Big Data Applications

Alicia Taylor Lamere
Bryant University, USA

ABSTRACT

This chapter discusses several popular clustering functions and open source software packages in R and their feasibility of use on larger datasets. These will include the kmeans( ) function, the pvclust package, and the DBSCAN (density-based spatial clustering of applications with noise) package, which implement K-means, hierarchical, and density-based clustering, respectively. Dimension reduction methods such as PCA (principle component analysis) and SVD (singular value decomposition), as well as the choice of distance measure, are explored as methods to improve the performance of hierarchical and model-based clustering methods on larger datasets. These methods are illustrated through an application to a dataset of RNA-sequencing expression data for cancer patients obtained from the Cancer Genome Atlas Kidney Clear Cell Carcinoma (TCGA-KIRC) data collection from The Cancer Imaging Archive (TCIA).

INTRODUCTION

Often in Big Data applications, one of the first steps when working with a new dataset is to perform exploration. Beyond basic examinations of variable distributions, outliers, etc., it can often be very informative to perform clustering analysis to see if any natural separation or grouping exists among the dataset.

In general, data mining techniques can be grouped into two categories: supervised and unsupervised. Supervised methods rely on a response variable of interest that these methods are tasked with predicting—meaning there is a clear objective of the analysis. This is not the case for unsupervised methods. These methods instead are used to make new discoveries about the data, searching for patterns that were not known ahead of time. Clustering, in its many forms, is one of the most widely used unsupervised methods for exploring datasets—due to both its general ease of implementation and of interpretation.

In this chapter, first the foundational mathematics for clustering are discussed, including distance measures, clustering algorithms and dimension reduction methods. Then, three of the most commonly-employed functions in R are explored and illustrated in the specific context of analyzing an RNA-Sequencing

DOI: 10.4018/979-8-3693-3026-5.ch022

expression dataset containing samples from Kidney Clear Cell Carcinoma patients. Their performances are compared and their merits and shortcomings when working with this form of data are discussed.

BACKGROUND

The basic concept behind all clustering methods is to group together similar datapoints based on the variables that describe them. Through this clustering, we can observe characteristics that distinguish data points from cluster to cluster, leading to potential hypotheses about our population. We can also use these clusters to identify subsets of our population, which we can focus on separately in future analysis. This clustering is generally accomplished by measuring the distance between these data points and grouping those that have the smallest distances between them. The goal is to maximize the separation between different clusters while minimizing the separation between data points within each cluster. An important consideration, then, becomes how we choose to measure this distance.

Distance Measures

There are many ways we can choose to measure the distance between two observations. At its core, any distance measure $d()$must have three key features when measuring three points x, y, and z (Tan et al., 2006):

1. Positivity. Distances must have nonnegative values: $d(x,y)\geq 0$, where $d(x,y)=0$ only in the case where $x=y$
2. Symmetry. The order of the points should not impact the distance measure: $d(x,y)=d(y,x)$
3. Triangle Inequality. Distances must satisfy the triangle inequality: $d(x,y)\leq d(x,z)+d(y,z)$

The most commonly used distance measure is the Euclidean distance.

Here, x_k represents the value within dimension k for a given point x, allowing this measure to easily be used for any n-dimensional space. Euclidean distance was generalized by Minkowski by introducing the parameter $h\geq 1$, creating the new formula:

$$d(x,y)=\left(\sum_{k=1}^{n}\left|x_k-y_k\right|^h\right)^{1/h}$$

Note $h=2$ represents Euclidean distance. If we allow $h=1$, we have what is commonly referred to as the Manhattan distance between two points. This is because it can be thought of (within two-dimensional space) as the number of so-called city blocks required to travel between two points. Also contained within this generalized formula is the supremum distance, in which we allow $h\to\infty$. In practice, this results in the maximum difference observed within a particular dimension as the distance measure between a pair of points, and hence this measure is often referred to as L*max* (Han et al., 2011).

For any given pair of data points, the Manhattan distance results in a larger distance measure, the supremum resulting in the smallest, and the Euclidean falling between. The choice of which measure to use should be derived from the application—situations in which differences should be minimized would be better suited for the supremum or Euclidean, for example.