Clustering

Prof. Eric A. Suess

May 6, 2020

Introduction

Today we will discuss Clustering.

Clustering

Clustering algorithms are for Unsupervised Learning .

There are **no labels** in the dataset.

Clustering produces labels for similar groups in the data.

Applications of Clustering

- segmenting customers
- identifying patterns that fall outside of known clusters
- simplify larger datasets
- useful for data visualization

Clustering

Unlabeled examples are given a cluster label and inferred entirely from the relationships within the data.

Clustering is **unsupervised classification**

Clustering produces ‘’new data”

Clustering

A problem with clustering is that the class labels produced *do not have meaning*.

Clustering will tell you which groups of examples are closely related but it is up to you to apply meaning to the labels.

Semi-Supervised Learning

If we begin with unlabeled data, we can use **clustering** to create **class labels**.

From there, we could apply a **supervised learner** such as **decision trees** to find the most important predictors of these classes.

k-means algorithm

The **k-means algorithm** is perhaps the most commonly used clustering method.

See the CRAN Task View: Cluster Analysis & Finite Mixture Models for a list of all the packages R has related to Clustering and beyond.

k-means algorithm

k-means is not **kNN**

The only similarity is that you need to specify a **k**.

The goal is to **minimize** the differences within each cluster and to **maximize** the differences between clusters.

k-means algorithm

The algorithm:

- starts with k random selected **centers/centroids**.
- assigns examples to an initial set of k clusters.
- it updates the assignments by adjusting the cluster boundaries according to the examples that fall into the cluster.
- the process of updating and assigning occurs several times until making changes no longer improves the cluster fit.

When using **k-means** it is a good idea to run the algorithm more than once to check the robustness of your findings.

Using distance

As with kNN, k-means treats feature values as coordinates in a multidimensional feature space.

Euclidean distance is used

$dist(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$

Using this **distance function**, we find the distance between each example and each cluster center.

The example is then assigned to the nearest cluster center.

Using distance

Because we are again using a distance measure, we need

- **numeric features**
- to **normalize** the features

Choosing the appropriate number of clusters

We need to balance the number of clusters **k**, try not to over-fit the data.

Rule-of-thumb is to set **k** equal to $\sqrt{n/2}$.

Or use the **elbow method**

- homogeneity within clusters is expected to increase as additional clusters are added.
- heterogeneity will decrease with more clusters.

Pick **k** at the elbow.

Other Clustering methods

There are many algorithms that can be used to cluster data:

- k-means **kmeans**
- Model based clustering **Mclust**
- Hierarchical clustering **hclust**
- Density based clustering **dbscan**

Big Data and Parallel Processing

See the **bigmemory** and **biganalytics** packages in R for **k-means** on *very big data* using *parallel processing*.

Example:

The author gives an example of clustering teens using *social media data*.