Final Report

Tram Chau

2023-10-13

Contents

1. Introduction & Motivation **2**

2. Development Process **2**

2.1 Reimplementation algorithms 2

2.2 Developement objects' functions 2

2.3 Customizing the performance 3

3. Testing **3**

4. Self-Reflection **3**

5. References **3**

1. Introduction & Motivation

It is challenging to visualize very high dimensional data. Thus, the data transformation techniques have long been developed to to transform these high dimensional data to a lower dimensional one. This helps to visualize data easier.

This package will incorporate 3 existing transformation algorithms in CRAN R in separate packages. They are Principal Component Analysis (PCA) , Kernel PCA and Non-Negative Matrix Factorization (NMF) [1] [2] [3]. On top of that, six new functions are developed supporting for viewing, plotting, predicting new data and inversing data. The main object of the package is called Transformer, through this object, it is easy for user to approach 3 different transformation algorithms interchanably.

Users are able to visualize the results easily before diving into the algorithms' mechanism. Moreover, users can easily interchange between techniques and compare the results to explore the different angle to the data and possibly spot on the most suitable technique for a specific data.

2. Development Process

The development process is divided to 3 main stages: reimplementing the algorithms, developing the object's methods, and customizing the performance for big dataset.

2.1 Reimplementation algorithms

Reimplement the existing algorithms to build the creator functions for the 'dtrans' package. As the main purpose is to integrating the existing different algorithms into one single place with extra supporting function, all three algorithms are cloned from the existing CRAN packages. The replication is done by 3 reasons:

- Avoid the dependency on the multiple packages which user would have to install when they install 'dtrans' packages
- Standardize the input and output of three algorithms to make it convenient to use them interchangably
- To understand the algorithms of creating, it is foundation to develop the predict and inverse function.
- To control the customizing process which integrate with Rcpp to speed up the algorithms for large dataset

Starting with the foundation prcomp function from stat package for PCA algorithms, the other two algorithms are standadized to align with the output of PCA. It is quite troublesome as they are different technically. Thus the output of the object include the attribute 'other' to cater for the differential among three algorithms.

2.2 Developement objects' functions

After the 'transformer' object is created, there are two main groups of object function are developed to support manipulation the object and data.

- Presentation functions includes print, summary, plot and plot3d
- Manipulation functions include predict and inverse.

These are augmentation functions for the 'Transformer' object, which support user to explore the data transformation. The inverse function is built based on the idea of image processing, where each component tend to present specific high-level feature of the dataset. This behaviour is different in each algorithms. However, this application is not demonstrated as the time limitation.

2.3 Customizing the performance

Three algorithms' performance suffer from large dataset in different way. PCA algorithm are suffer more from dataset with high number of columns than from dataset with high number of rows.

NMF

KPCA

3. Testing

4. Self-Reflection

5. References

1. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philosophy magazine
2. P. Paatero UT (1994) Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values, environmetrics
3. Schölkopf, Bernhard MK-R Smola, Alex (1998) Nonlinear component analysis as a kernel eigenvalue problem, neural computation