Introduction

Here we classify dimension reduction algorithms from the perspective of linear dimension reduction (LDR) and nonlinear dimension reduction (NLDR). In this write up, LDR will be introduced. NLDR will be introduced in the "nonlinear dimension reduction.pdf" under the same folder.

Linear dimension reduction (LDR)

Many individual LDR algorithms have been already studied in other notes. Some details will be provided only to techniques not appeared in other notes.

PCA LDA ICA FA

- These LDR algorithms are each studied in an individual document in details under the /algorithm folder.
- Linear discriminant analysis (LDA) can sometimes compensate some disadvantages of PCA.

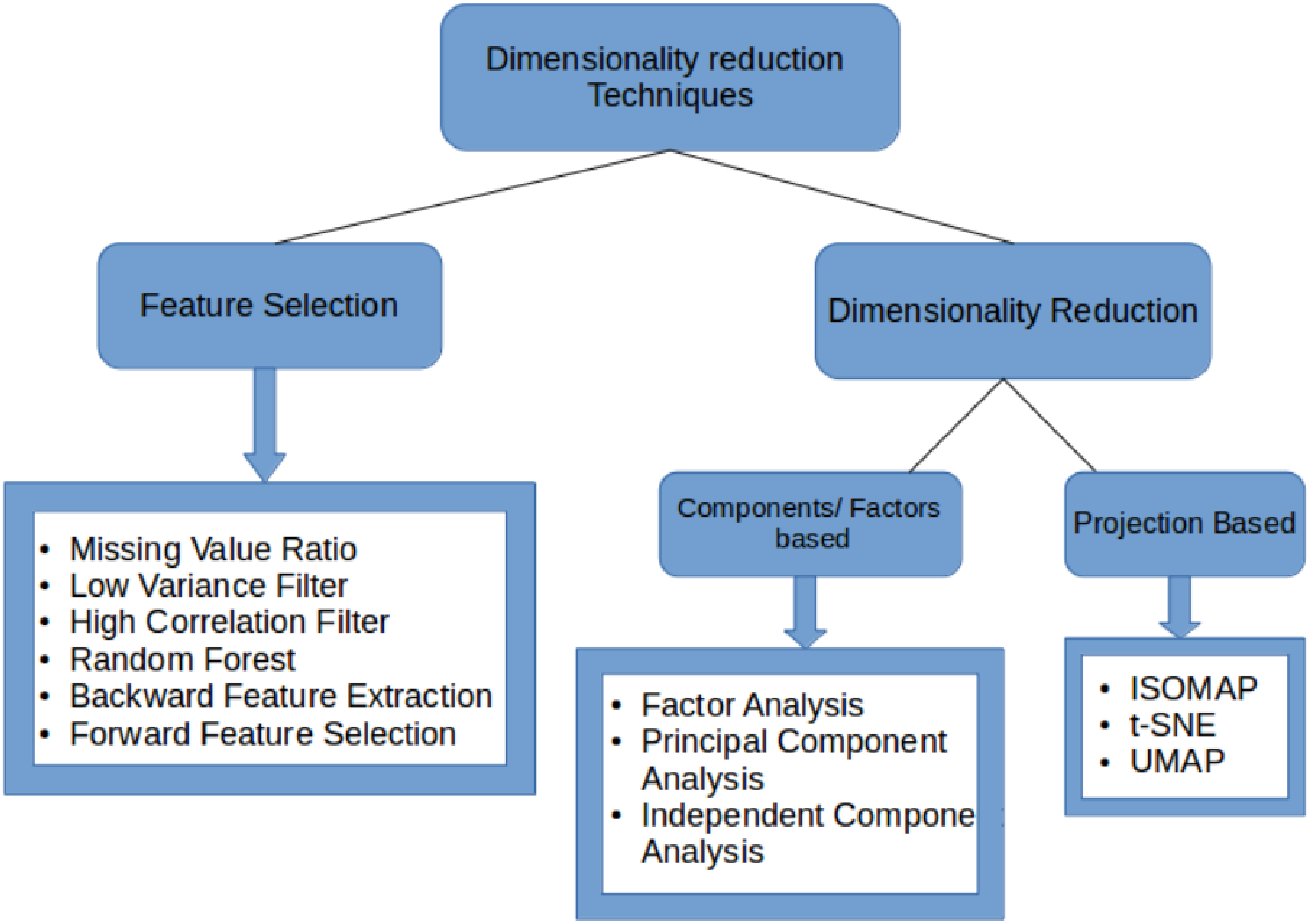
Non-negative matrix factorization (NMF)

- Unlike PCA, UMF can learn parts (see example in "kmeans *hierarchical* tSNE *PCA* NMF.pdf" under /applications folder. NMF has local maxima.
- NMF can learn individual parts. How does compare to ICA? From the paper [https://link.springer.com/article/10.1007/s11634-014-0192-4In (https://link.springer.com/article/10.1007/s11634-014-0192-4In)](https://link.springer.com/article/10.1007/s11634-014-0192-4In), the authors arrived the following results: "we compare the performances of NMF and ICA as blind source separation (BSS) methods using some standard NMF and ICA algorithms, and point out the difficulty in choosing the representative reconstructions originated from the nonuniqueness property of NMF."

Other ways of classifying dimension reduction algorithm

https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+AnalyticsVidhya+%28Analytics+Vidhya%29 (https://www.analyticsvidhya.com/blog/2018/08/dimensionality-reduction-techniques-python/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+AnalyticsVidhya+%28Analytics+Vidhya%29)

Another way of classifying dimension reduction algorithm is shown in the figure below. Three categories exit in this classification, i.e., feature selection, components/Factors based, and projection based.



- Details about feature selection approach for dimension reduction can be found in /dataScience/machineLearning/featureEngineering/feature selection and extraction.pdf

- Components/factor based approaches for dimension reduction, and other linear dimension reduction algorithms are introduced in this write up.
- ISOMAP, t_SNE, UMAP and other nonlinear dimension reduction algorithms are introduced in the "nonlinear dimension reduction.pdf" under the same folder.

A brief Summary

- Missing Value Ratio: If the dataset has too many missing values, we use this approach to reduce the number of variables. We can drop the variables having a large number of missing values in them.
- Low Variance filter: We apply this approach to identify and drop constant variables from the dataset. The target variable is not unduly affected by variables with low variance, and hence these variables can be safely dropped.
- High Correlation filter: A pair of variables having high correlation increases multicollinearity in the dataset. So, we can use this technique to find highly correlated features and drop them accordingly.
- Random Forest: This is one of the most commonly used techniques which tells us the importance of each feature present in the dataset. We can find the importance of each feature and keep the top most features, resulting in dimensionality reduction.
- Both Backward Feature Elimination and Forward Feature Selection techniques take a lot of computational time and are thus generally used on smaller datasets.
- Factor Analysis: This technique is best suited for situations where we have highly correlated set of variables. It divides the variables based on their correlation into different groups, and represents each group with a factor.
- Principal Component Analysis: This is one of the most widely used techniques for dealing with linear data. It divides the data into a set of components which try to explain as much variance as possible.
- Independent Component Analysis: We can use ICA to transform the data into independent components which describe the data using less number of components.