Comparative Study on the role of dimensionality in Bigdata

Pradeepan P[#1], Gladston Raj S[#2]
[#1] Varavilathoppu poovar, Trivandrum, Kerala, India
[1]Pradeep0076@gmail.com
[#2]Associate Professor & Head of Computer Science,
Government College Nedumangad, Trivandrum, Kerala, India
[2]gladston@rediffmail.com

Abstract— **The curse of dimensionality refers to all the problems that arise when working with data in the higher dimensions that did not exist in the lower dimensions. As the number of features increase, the number of samples also increases proportionally. The more features we have, the more number of samples we need to have all combinations of feature values well represented in our sample. As the number of features increases, the machine learning model becomes more complex. The more the number of features, the more the chances of over fitting. A machine learning model that is trained on a large number of features, gets increasingly dependent on the data, it was trained on and in turn over fitted, leading to poor performance on real data, beating the purpose avoiding over fitting is a major motivation for performing dimensionality reduction. The paper presents a review and systematic comparison of different dimensionality reduction techniques, also explains the strength and weakness of these techniques.**

Keywords: big data, machine learning, dimensionality reduction

I. INTRODUCTION

We now live in an era of data deluge where large volumes of data are accumulating in all aspects of our lives. Data streams coming from diverse domains contribute to the emerging paradigm of big data. It may be a great opportunity for the data scientist amongst the vast amount and array of data. By discovering associations, analyzing patterns and predicting trends within the data, big data has the potential to change our society and improve the quality of our life. Big data refers to the following three types based on data sources from physical, cyber, and social worlds. While big data brings great opportunities, unpredictable challenges are on the way at the same time. It cannot be stored, analyzed and processed by traditional data management technologies and requires adaptation of some new workflows, platforms and architectures. The field of machine learning which is useful to accomplish tasks of prediction, classification, and association about large amounts of data, is getting more and more attention from researchers in the current time. However, as the big data era is coming, some characteristics of bigdata will bring great challenges to the traditional machine learning methods. Volume is the important aspect of big data. As data volumes and varieties grow, processing and consuming the insights generated becomes challenging. The curse of dimensionality refers to various challenges that arise when analysing data in high-dimensional spaces that do not occur in low-dimensional settings. High dimensionality coupled with large sample sizes creates issues such as heavy computational cost and algorithmic instability.

II. WHY REDUCE DIMENSIONALITY

We need a huge number of data points to get an accurate prediction from a machine learning model in the cases of high dimensional data. The number of data points required for the model to perform well increases exponentially with an increase in dimensionality. It is not feasible in some situations to train the model with a very large data. Main characteristic of dimensionally reduced data listed below

- dataset will be less complex
- dataset will take up small storage space
- dataset will require less computation time
- dataset will have lower chance of model over fitting

III. TECHNIQUES FOR DIMENSIONALITY REDUCTION

There are many techniques that can be used for dimensionality reduction. In this section, we review some main techniques.

A. Feature Selection

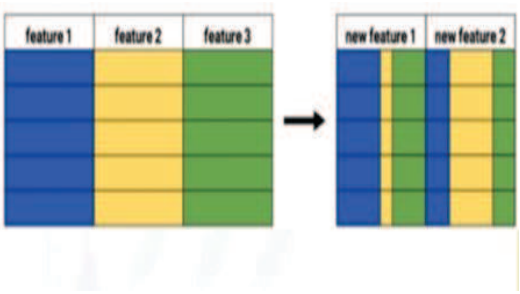
Feature selection is for remove the irrelevant or redundant features from your dataset. The key difference between feature selection and extraction is that feature selection keeps a subset of the original features while feature extraction creates extracted ones.

B. Feature Extraction

Feature extraction is for creating a new, smaller set of features that still extract most of the useful information.

Fig 1: (a) Feature selection methods

Fig 1 :(b) Feature extraction methods



1) High Correlation filter:

High correlation between two variables means they have same trends and maybe carry similar information. This can bring down the performance of machine learning models drastically (linear and logistic regression models, for instance). We can calculate the correlation between independent numerical variables that are numerical in nature. If the correlation coefficient crosses a certain threshold, we can drop one of the variables (dropping a variable is highly subjective and should always be done keeping the domain in mind). We use Pearson correlation coefficient to find out the correlation coefficient.

Strengths:

- Applying correlation thresholds is also based on solid intuition: similar features provide redundant information. Some algorithms are not robust to correlated features, so removing the correlated features that can boost performance

Pearson's correlation coefficient equation

$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}}$$

Where:

N	=	number of pairs of scores
Σxy	=	sum of the products of paired scores
Σx	=	sum of x scores
Σy	=	sum of y scores
Σx^2	=	sum of squared x scores
Σy^2	=	sum of squared y scores

Weaknesses:

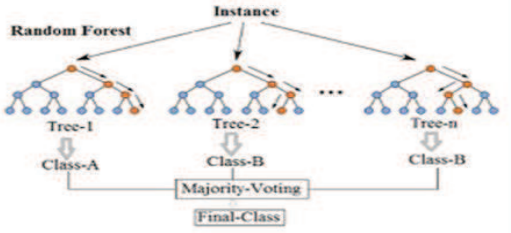
- We must manually set a correlation threshold, which can be tricky to do.
- If you set your threshold too low, there is a risk for dropping useful information. Whenever possible, prefer algorithms with built-in feature selection over correlation thresholds. Even for algorithms without built-in feature selection, Principal Component Analysis (PCA) is often provide a better solution.

2) Random Forests/Ensemble Trees:

Random forest is a supervised ensemble tree based machine learning algorithm that is used for both classifications as well as regression problems. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees mean more robust forest. Similarly, the random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally chooses the best solution by means of voting. It is an ensemble method that is provide better solution than a single decision tree because it reduces the over-fitting by averaging the result.

Decision tree ensembles useful for column selection in addition to being effective classifiers. If we generate a large and carefully constructed set of trees to predict the target classes and then use each column's usage statistics to find the most informative subset of columns. If a column is often selected as the best split, it is very likely to be an informative column that we must keep. For all columns, we calculate a scorecard as the number of times that the column was selected for the split, divided by the number of times in which it was a candidate. The most predictive columns are those which have the highest scores.

Fig 2: simplified random forest



Random forest creation

1. Randomly select "k" features from total "m" features where k << m
2. Among the "k" features, calculate the node "d" using the best split point
3. Split the node into daughter nodes using the best split
4. Repeat the 1 to 3 steps until "l" number of nodes has been reached
5. Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

Strengths:

- Decision trees can learn non-linear relationships, and are fairly robust to outliers. Ensembles perform very well in practice.

Weaknesses:

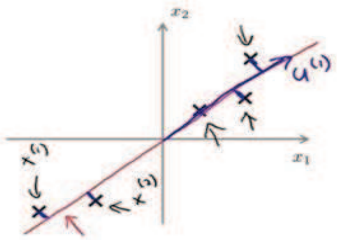
- Unconstrained, individual trees have tendency to over fitting because they can keep branching until they memorize the training data. However, this can be alleviated by using ensembles

3) Principal Component Analysis:

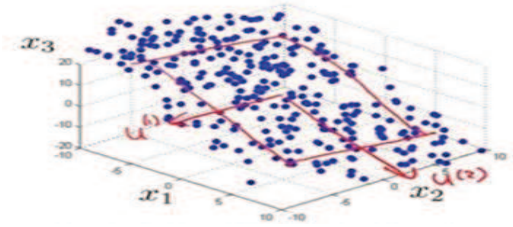
Principal component analysis (PCA) is an unsupervised technique and it is a statistical procedure that transforms the original *n* numeric dimensions of a dataset into a new set of *n* dimensions called principal components. As a result of the transformation, the first principal component has the largest possible variance. After each succeeding principal component has the highest possible variance under the constraint that it is orthogonal to the preceding principal components. Keeping only the first $m < n$ (A number m of linear combinations of the n input features) principal components reduces the data dimensionality while retaining most of the data information, i.e., variation in the data. Notice that the PCA transformation is sensitive to the relative scaling of the original columns, and therefore, the data need to be normalized before applying PCA. Also notice that the new components are not real, system-produced variables anymore. Applying PCA to your dataset loses its interpretability. If interpretability of the results is important for your analysis, PCA is not the transformation that you should apply. The principal component's (PC's) possess some useful properties which are listed below:

1. The PCs are essentially the linear combinations of the original variables, the weights vector in this combination is actually the eigenvector found which in turn satisfies the principle of least squares.
2. The PCs are orthogonal
3. The variation present in the PCs decrease as we move from the 1st PC to the last one, hence the importance.

Fig 3 (a): Reduced data from 2D to 1D



(b): Reduced data from 3D to 2D



Strengths:

- PCA is a versatile technique that works well in practice. It's fast and simple to implement, which means you can easily implement the algorithms with and without PCA to compare performance. In addition, PCA offers several variations and extensions (i.e. kernel PCA, sparse PCA, logistic PCA etc.) to tackle specific roadblocks.

Weaknesses:

- The new principal components are not interpretable, which may be a deal-breaker in some settings. In addition, you must still manually set a threshold for cumulative explained variance.
- Only for numeric columns.

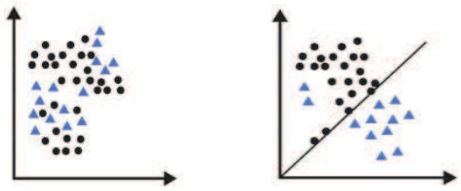
4) Linear Discriminant Analysis (LDA):

Linear Discriminant Analysis seeks to best separate (or discriminate) the samples in the training dataset by their class value. Specifically, the model seeks to find a linear combination of input variables that achieves the maximum separation for samples between classes (class centroids or means) and the minimum separation of samples within each class. A number *m* of linear combinations (discriminant functions) of the *n* input features, with $m < n$, are produced to be uncorrelated and to maximize class separation. These discriminant functions become the new basis for the dataset. All numeric columns in the dataset are projected onto these linear discriminant functions, effectively moving the dataset from the *n*-dimensionality to the *m*-dimensionality. In order to apply the linear discriminant analysis technique for dimensionality reduction, the target column has to be selected first. The maximum number of reduced dimensions (*m)* is the number of classes in the target column minus one, or if smaller, the number of numeric columns in the data. Notice that linear discriminant analysis assumes that the target classes follow a multivariate normal distribution with the same variance but with a different mean for each class.

Strengths:

LDA is supervised, which *can* (but doesn't always) improve the predictive performance of the extracted features. Furthermore, LDA offers variations (i.e. quadratic LDA) to tackle specific roadblocks.

Fig 4 (a) Before LDA (b) After LDA



Weaknesses:

- As with PCA, the new features are not easily interpretable, and you must still manually set or tune the number of components to keep. LDA also requires labeled data, which makes it more situational.
- Only for numeric columns requires normalization and a target column to separate the data.

IV. METHODOLOGY

To test how effective various dimensionality reduction algorithms are at projecting data from high dimensional to low dimension took 'customer satisfaction' dataset from kaggle public data platform and partitioned this dataset into a training set and a testing set. For this review opted 10000 rows and 370 columns. The partition is done as follow: First n rows are selected for test the remaining rows (10000-n) are used for training set and for computing the accuracy and time reduction rate for the above mentioned dimensionality reduction techniques. The various steps used in this work are discussed below:

1) In step-1 feature selection technique filter method used in the dataset for removing constant, quasi-constant and duplicate elements on the customer satisfaction dataset.
2) In step-2, the dataset is experimented using tree based machine learning algorithm Random Forest. The accuracy and computational time of this classifier is calculated.
3) In step-3, highest correlation filter applied on the filtered dataset to extract the most prominent features. The resultant dataset is then experimented on the Random forest algorithm. The ML algorithm using Highest correlation filter is evaluated, the accuracy and computational time.
4) In the step-4, PCA is applied on the filtered dataset. The resultant dimensionally reduced dataset is then experimented using the aforementioned ML algorithm. The results obtained are again evaluated.
5) In step-5 LDA is applied on the filtered dataset. The resultant dimensionally reduced dataset is then experimented using the ML algorithm. The results obtained are again evaluated.
6) In step-6 analyzed and compared the accuracy and computational time of original dataset and after applied the above dimensionality reduction techniques through Random forest classification, then calculated the computational time reduction rate.

V. RESULTS & DISCUSSION

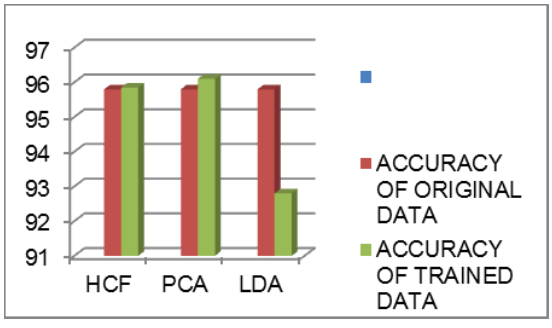
The experimentation is performed on customer satisfaction dataset which using Python 3 in Pentium(R) dual-core processor with 2GB ram.

Table 1: accuracy and process time measured in this methodology

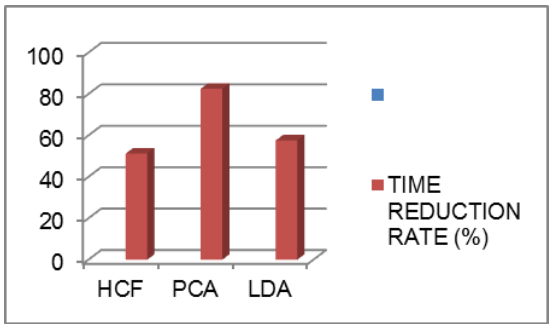
	ORIGINAL DATA	HCF	LDA	PCA
ACCURACY	95.8	95.85	92.8	96.1
PROCESS TIME (Sec)	3.14	1.5	1.33	0.546

The figures describe the accuracy and computational time reduction rate of the above specified techniques when applied them to the database, fig 5(a) shows that the accuracy rate not moving an excessive amount of mentioned dimensionality reduction techniques, principal components analysis provide better accuracy than the original data, high correlation filter provide similar to the original data and linear discriminant analysis provide less accuracy than the original data.

Fig 5(a) : accuracy comparison of original data and dimensionally reduced data.



5 (b): computational time reduction rate of highest correlation filter, principal component analysis and linear discriminant analysis after dimensionality reduction



Within the case of time reduction rate, its make huge impact in the computation time, the fig 5(b) shows that after using dimensionality reduction techniques we can reduce computational time effectively, while not moving the accuracy of the database.

VI. CONCLUSION

To address visualization challenges posed by big and high dimensional data, this algorithms and techniques that compress the amount of data and/or reduce the number of attributes to be analyzed and visualized. By combining these approaches several benefits result. First, the storage space required for computing in-memory visualization is reduced and rendering speeds increase. Next, computational resources and consumption-based cloud data costs are decreased, as there is less data to process. Also, model error and accuracy are improved. Finally, it becomes easier to visualize patterns and trends more clearly, allowing key insights to be generated.

There is no best technique for dimensionality reduction and no mapping of techniques to problems. Each technique has its own merits and demerits; instead the best approach is to use systematic controlled experiments to discover what dimensionality reduction technique when paired with your model of choice, result in the best performance on your dataset.

REFERENCES

[1] N. Sharma and K. Saroha, "Study of dimension reduction methodologies in data mining," International Conference on Computing, Communication & Automation, Noida, india:IEEE 2015, pp. 133-137

[2] J. Menezes and N. Poojary, "Dimensionality reduction and classification of hyperspetral images using DWT and DCCF," 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), Muscat: IEEE 2016, pp. 1-6,

[3] S.Velliangiri. Alagumuthu krishnan Iwin Thankumar joseph journal of 'A Review of Dimensionality Reduction Techniques for Efficient Computation' Procedia Computer Science Volume 165, 2019

[4] A. Sellami and M. Farah, "Comparative study of dimensionality reduction methods for remote sensing images interpretation," 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse: IEEE 2018, pp. 1-6,

[5] Astha Pareek, Manish Gupta: Review of Data Mining Techniques in Cloud Computing Database, International Journal of Advanced Computer Research, Volume-2 Number-2 Issue-4 June-2012

[6] Jeffrey Voas and Jia Zhang, "Cloud Computing: New Wine or Just a New Bottle?" Database Systems Journal vol. III, no. 3/2012 71 IEEE Internet Computing Magazine, 2000.

[7] M. S. Chen, J. Han, and P. S. Yu. "Data mining: an overview from database perspective." IEEE Trans. On Knowledge and Data Engineering, 5(1):866—883, Dec.1996

[8] Muhammad Husnain Zafar, and Muhammad Ilyas: "A Clustering Based Study of Classification Algorithms", International Journal of Database Theory and Application, Vol.8, No.1 (2015), pp.11-22

[9] Yogita Rani, Dr. Harish Rohil: "A Study of Hierarchical Clustering Algorithm", International Journal of Information and Computation Technology, Volume 3, (2013).

[10] Boniface, M.; et al. (2010), —"Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds", 5th International Conference on Internet and Web Applications and Services (ICIW), Barcelona, Spain: IEEE, pp.155-160

[11] B. Cheng, L. Zhuo and J. Zhang, "Comparative Study on Dimensionality Reduction in Large-Scale Image Retrieval," 2013 IEEE International Symposium on Multimedia, Anaheim, CA: IEEE 2013, pp. 445-450.

[12] F. S. Tsai and Kap Luk Chan, "Dimensionality reduction techniques for data exploration," 2007 6th International Conference on Information, Communications & Signal Processing, Singapore, IEEE 2007, pp. 1-5,

[13] A. A. A. El Hamid, M. S. El Tokhy, K. S. Gerges, I. I. Mahmoud, B. A. Abozalam and G. A. M. Atlam, "Comparison between dimensionality reduction techniques for pileup detection in digital gamma ray spectroscopy," 2018 35th National Radio Science Conference (NRSC), Cairo, IEEE, 2018, pp. 465-474

[14] S. K. Joshi and S. Machchhar, "An evolution and evaluation of dimensionality reduction techniques — A comparative study," 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, IEEE, 2014, pp. 1-5

[15] M. Denguir and S. M. Sattler, "A dimensionality-reduction method for test data," 2017 International Mixed Signals Testing Workshop (IMSTW), Thessaloniki, IEEE,2017, pp. 1-6

[16] Z. Zheng, "Analysis and Comparison of Dimensional Reduction Based on Capture Data," 2010 Asia-Pacific Conference on Wearable Computing Systems, Shenzhen, 2010, pp. 163-164

[17] S. Keller, A. C. Braun, S. Hinz and M. Weinmann, "Investigation of the impact of dimensionality reduction and feature selection on the classification of hyperspectral EnMAP data," 2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Los Angeles, CA, IEEE, 2016, pp. 1-5.

[18] C. Deisy, B. Subbulakshmi, S. Baskar and N. Ramaraj, "Efficient Dimensionality Reduction Approaches for Feature Selection," International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, Tamil Nadu, IEEE, 2007, pp. 121-127.

[19] A. Shyr, R. Urtasun and M. I. Jordan, "Sufficient dimension reduction for visual sequence classification," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, 2010, pp. 3610-3617

[20] L. Wolf and S. Bileschi, "Combining variable selection with dimensionality reduction," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 801-806 vol. 2

[21] M. Dash, H. Liu and J. Yao, "Dimensionality reduction of unsupervised data," Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence, Newport Beach, CA, USA, 1997, pp. 532-539.

[22] R. Ramachandran, G. Ravichandran and A. Raveendran, "Evaluation of Dimensionality Reduction Techniques for Big data," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 226-231.