Dimensionality Reduction via Graph-based Feature Selection

Hrishik Sai Bojnal
Department of Computer Science and Engineering
RV Institute of Technology and Management
Bengaluru, India
hrishiksai@outlook.com

Parnika Singh
Department of Computer Science and Engineering
RV Institute of Technology and Management
Bengaluru, India
parnikasingh171203@gmail.com

Dr. Anitha J
Professor, Department of Computer Science and Engineering
RV Institute of Technology and Management
Bengaluru, India
anithaj.rvitm@rvei.edu.in

***Abstract*— Given the rapid influx of data, organizing and managing its features poses a significant challenge. Many features are deemed redundant, impeding efficient storage and training of models. We introduce a method to address this challenge by defining an algorithm that successfully achieves dimensionality reduction. By establishing a threshold, we identify pairs of features with significant linear correlations which are modelled as vertices and edges on a graph. Using graph theory principles, we can pinpoint non-essential features. By strategically removing these points, our method proceeds to streamline data management processes and enhances computational efficiency while outperforming Principal Component Analysis (PCA).**

Keywords—Dimensionality Reduction, Graph Theory, Cut Vertices, R-Squared, Principal Component Analysis

I. Introduction

Over the last few years, the quantity of data generated has increased significantly, outpacing the growth in computing power [1]. In this era of big data, being able to efficiently process vast amounts of information is crucial. However, a persistent issue in this domain is the presence of redundant features within real-world datasets. Redundant features increase computational complexity, storage space, and are known to obscure meaningful patterns inherent in the data. Effective methods for dimensionality reduction are crucial for maintaining data usability in model training and analysis. They are pivotal in improving the speed and efficiency of the machine learning lifecycle. This paper introduces a polynomial-time algorithm for dimensionality reduction that is independent of the dataset type or the choice of training algorithm. Our approach identifies linear correlations by calculating R-squared (R^2) values between features within a specified threshold. We then use graph-theoretic principles to prune as many features as possible while preserving the quality of the dataset.

II. Literature Survey

The current trends in the production of large-scale data, commonly referred to as big data, are determined by various factors including volume, variety, and velocity. The volume of generated data continues to increase exponentially. Often, the collected data includes many unnecessary features that contribute to system complexity and hinder the extraction of meaningful information. Thus, the concept of dimensionality reduction is applied to streamline such data. Recent technological advancements have highlighted shortcomings. In the works by Siyi Feng et. al. [2], it is stated that during the construction of machine learning models, a significant impact on its complexity is seen when there is too large of an input, but when there is insufficient data, then the challenge of dealing with poor generalization of the training model is seen. In a study conducted by Xiaoyan Sun et al. [3], existing algorithms for feature reduction were found to suffer from high computational costs. Typically, these methods involve calculating correlation metrics depending on the number of attributes and selecting a threshold value. Attributes are then dropped based on this threshold to facilitate further analysis. Ryan Nathanael Soenjoto Widodo et al. [4] proposed the Hadoop Data Reduction Framework (HDRF), which showed minimal overhead compared to existing approaches. It achieved up to 48.96% higher throughput and demonstrated superior results in storage reduction. Analysis of data having more features than observations puts a significant strain on the computational costs. In the research by Honglei Zhang et al. [5], their method using Hamming distance surpassed Principal Component Analysis (PCA) and other emerging techniques. In the study by Consolata Gakii et al. [6], a graph-based approach, PCA and recursive feature elimination selects features for classification from RNAseq datasets from two lung cancer datasets. This resulted in the observation that the proposed graph-based feature-selection approach combined with rule mining is the optimal way to find associations between features. Abbas Kazemipour et. al. [7], state that during the construction of machine learning models, a significant impact on its complexity is seen when there is too large of an input, but when there is insufficient data then the challenge of dealing with poor generalization of the training model is seen. These approaches represent current competition in dimensionality reduction, often outperforming existing techniques by varying margins. In our forthcoming sections, we expand the current boundaries of dimensionality reduction techniques.

III. Methodology

This section formally outlines the proposed mechanism to identify redundancy among features. This helps in the understanding of the span of data and choosing an appropriate value to set as a threshold. Values exceeding this threshold are dropped and hence we achieve a reduced dataset.

A. *Feature Redundancy*

We use the *coefficient of determination* (R^2 score) to identify redundancies between two features. The R^2 score, obtained by squaring the Pearson Correlation Coefficient (POC) between two features, quantifies the proportion of variance in one feature that is predictable from another feature. A high R^2 score between two pairs of features suggests a linear redundancy.

We now define the *τ-redundancy* between two features as follows: given a threshold $\tau \in [0,1)$, two features F_1 and F_2 are considered τ-redundant if $R^2(F_1,F_2) \geq \tau$.

Formally, if F_1 and F_2 are τ-redundant for a suitable τ, then for two corresponding entries $f_1 \in F_1$ and $f_2 \in F_2$, we have the approximation:

$$f_2 \approx \beta_0 + \beta_1 \cdot f_1 \tag{1}$$

where β_0 and β_1 are two arbitrary constants.

Thus, for the training of any machine learning model, using only a single feature between the two would suffice. Consider an example of a multilayer perceptron (MLP) training on a dataset D with features $\{F_1, F_2, F_3, ..., F_n\}$. The activation of a neuron located in the initial hidden layer is defined as some activation function ϕ applied to the dot product of the learnable weights of the neuron $\{w_1, w_2, w_3, ..., w_n\}$ added by a bias b with a row of D. Mathematically,

$$activation = \phi(f_1 w_1 + f_2 w_2 + f_3 w_3 + ... + f_n w_n + b) \tag{2}$$

Now, if F_1 and F_2 are τ-redundant for a suitable τ, then, the activation can be approximated as

$$activation = \phi(f_1 w_1 + (\beta_0 + \beta_1 \cdot f_1) \cdot w_2 + f_3 w_3 + ... + f_n w_n + b) \tag{3}$$

Simplifying, we get

$$activation = \phi(f_1 \cdot (w_1 + w_2 \cdot \beta_1) + f_3 w_3 + ... + f_n w_n + (b + w_2 \cdot \beta_0)) \tag{4}$$

Here, $(w_1 + w_2\beta_1)$ and $(b + w_2\beta_0)$ can be learnt as new parameters w_1' and b' respectively. Thus, the feature f_2 can be discarded altogether.

B. *Removing Redundant Features*

To effectively analyze and identify redundant features, we model the dataset as a correlation graph, where features are represented as nodes and correlations between features are represented as edges. In our approach, each feature in the dataset is represented as a node in the graph. Edges between nodes indicate significant correlations between features, determined by an R^2 score above an arbitrary threshold.

Choosing the essential features to preserve can be non-trivial due to the complex correlations among multiple features within a dataset, resulting in a robust network. There are multiple measures of importance for a network. Here, we examine articulation points as an indicator of importance. Articulation points, or cut vertices, are the nodes whose removal would lead to an increase in the number of disconnected components in the graph and are thus important in maintaining the graph's connectivity.

Here, we define the *essential columns* of the dataset as the articulation points of the graph. Conversely, *redundant features* are defined as the remaining non-articulation vertices of the graph. By choosing the articulation points, we effectively prune the peripheral vertices of the graph while preserving the core structure (Fig. 1). By focusing on these points, we ensure that we retain the features central to the structure of the data, while eliminating less critical features.

This method of feature selection helps maintain the integrity of the data's internal relationships. By removing the outer vertices, which represent the more redundant features, we decrease the dimensionality of the dataset without losing significant information. The resulting dataset is more compact and efficient.

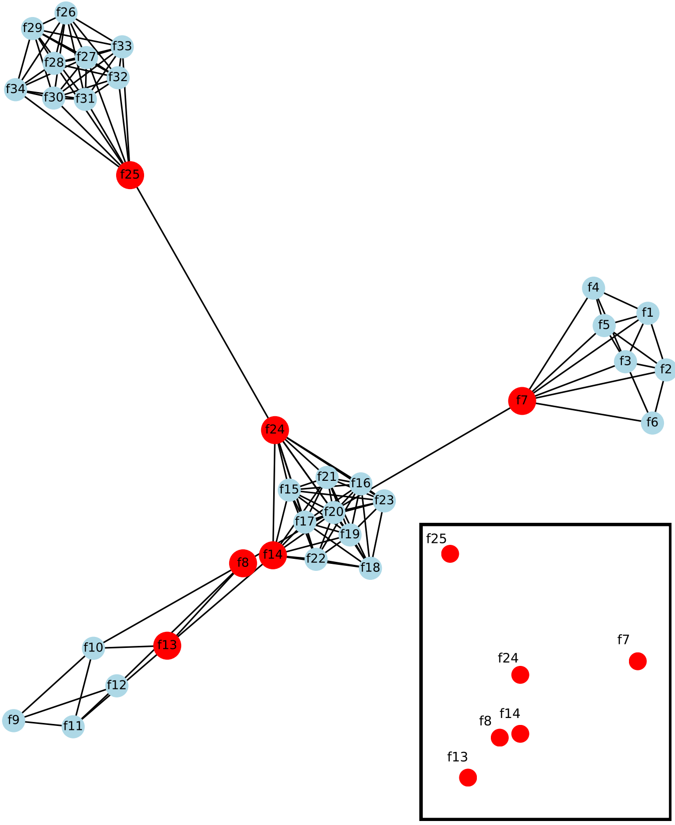

Fig. 1 Articulation points (shown in red) summarize the general structure of the graph.

C. Algorithm

To formalize this approach, we introduce Algorithm 1, which performs dimensionality reduction by identifying and retaining key features according to their importance in the graph structure. The algorithm begins by constructing a graph where each feature is represented as a node. Edges are formed between nodes with significant correlations, as indicated by an R2 score above a predefined threshold. Following this, the algorithm utilizes Tarjan's method to identify articulation points—nodes whose removal would lead to an increase in the graph's disconnected components.

Algorithm 1 Dimensionality Reduction via Graph-based Feature Selection

```
function REDUCEDIMENSIONALITY ($D$: Dataset, $\tau$: Threshold): Reduced Dataset
G ← (V, E)
pairs ← GETPAIRS(D)
for ($F_i$, $F_j$) in pairs do
        $R^2_{ij}$ ← COMPUTER$^2$($F_i$, $F_j$)
        if $R^2_{ij} > \tau$ then
                add($F_i$, V) if $F_i \notin V$
                add(($F_i$, $F_j$), E)
        end if
end for
articulationPoints ← TARJAN(G)
D' ← ∅
for $F_i$ in D do
        if $F_i \in$ articulationPoints or $F_i \notin V$ then
                add($F_i$, D')
        end if
end for
return D'
end function
```

D. Time Complexity Analysis

This subsection examines the computational complexity of our method. To facilitate this analysis, we introduce the following notation:

- Let N be the number of features in the dataset.
- Let n be the number of data points in each feature.

The algorithm can be separated into distinct levels, each with its own specific time complexity.

a) Calculation of R^2 Values: We perform linear regression over all possible pairs of features. For N features, we have $N\,(N - 1)\,/\,2$ total pairs. The calculation of R^2 values involves the calculation of the Pearson correlation coefficient, an O(n) operation[1], and squaring it for all pairs of features. Therefore, we can calculate the overall time complexity of this stage as

$$O\left(\frac{N(N-1)}{2} \times n\right) = O(N^2 n) \qquad (5)$$

b) Constructing the Graph: After the pairwise calculations, we construct a graph by adding edges between pairs of features whose R^2 values exceed τ. For a complete graph, there is an edge between pair of features, totalling to $N\,(N - 1)\,/\,2$ edges, resulting in an $O(N^2)$ time complexity.

c) Obtaining the Articulation Points: The most efficient algorithm for the detection of articulation points was proposed by Tarjan [8], which has a complexity of O(V+E), where V and E are the cardinalities of the sets of vertices and edges respectively. The number of edges E is again $O(N^2)$ in the worst case, resulting in an overall complexity of $O(N^2)$.
The time complexity over all levels is clearly dominated by the R-Squared value calculations and remains polynomial, given by:

$$O(N^2 n + N^2 + N^2) = O(N^2 n) \qquad (6)$$

IV. Experimental Results

To assess the efficacy of our proposed feature selection method, we use a real-world dataset. This section outlines the criteria for dataset selection, describes the specific dataset chosen, details the experimental workflow, and compares our feature selection approach with PCA.

A. Dataset Selection Criteria

We conduct the evaluation on the ClimSim atmospheric physics dataset, generated by the Energy Exascale Earth System Model (E3SM)-Multiscale Modelling Framework (MMF) model. The dataset consists of a number of attributes with multiple corresponding levels per attribute. The data, therefore shows high levels of granularity and versatility. Due to this structure, we show that our model adapts to various types of data at versatile scales. This specific dataset also aids in the direct real-world application in regards to the weather prediction systems. Combining such values gives us a positive outlook on the criteria. Therefore, the dataset makes it an ideal reference for our evaluation.

B. Dataset Overview

The original dataset contains 556 columns corresponding to 25 input variables and 368 columns corresponding to 14 target variables. Some input columns represent vertical levels whereas others are scalar. The dataset encapsulates the effects of small-scale processes on large scale climate patterns but this comes at great computational cost that lowers its usage potential.

[1] The Pearson correlation coefficient is calculated using

$$r = \frac{\sum (x_i - \bar{x})\,(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

The input variables sampled at 60 levels include: air temperature, specific humidity, cloud liquid mixing ratio, cloud ice mixing ratio, zonal wind speed, meridional wind speed, ozone volume mixing ratio, and nitrous oxide volume mixing ratio. Single dimension input variables include: surface pressure, solar insolation, surface latent heat flux, surface sensible heat flux, zonal surface stress, meridional surface stress, cosine of solar zenith angle, albedo for diffuse longwave radiation, albedo for direct longwave radiation, albedo for diffuse shortwave radiation, albedo for direct shortwave radiation, upward longwave flux, sea-ice areal fraction, land areal fraction, ocean areal fraction, snow depth over land.

C. *Workflow*

We employed a random forest regressor to predict the heating tendency at the first sampled level. The prediction is based on the dataset's 556 input variables. For this evaluation, we utilized the first 1,000 rows of the dataset, which were partitioned into training and testing sets with an 80-20 split. The random forest regressor algorithm was applied to these sets to train the model and make predictions.

D. *Performance Evaluation and Comparison with PCA*

Without any dimensionality reduction, the model achieved an R^2 value of 0.979, with an MSE of 1.713e-11. Applying our feature selection method, resulted in a 48.56 % reduction in features (277 redundant features). The model trained on the reduced dataset achieved a value of 0.939 with an MSE of 5.095e-11. Interestingly, a PCA reduction over a range of components only yielded a minimum MSE of 1.29e-10 (Fig 2.) and a maximum R^2 score of 0.85 (Fig. 3.).

We can infer from these results that the proposed method outperforms PCA and maintains a high level of predictive accuracy, with only a marginal increase in error compared to pre-reduction.

V. Conclusion

In this study, we have identified a novel method for dimensionality reduction (feature selection) by modelling the dataset as a graph and identifying essential features using articulation points. Our method was validated on ClimSim, where it demonstrated significant dimensionality reduction with only a marginal loss in predictive accuracy. This graph-based approach effectively preserves the core structure of the data, ensuring that critical information is retained while redundant features are eliminated. Future improvements could include creating faster alternatives for calculating R^2 scores for all possible feature pair, which would enhance the scalability of our method for larger datasets. Additionally, exploring heuristics beyond articulation points for identifying essential features may yield further improvements in feature selection accuracy. Investigating algorithms with a lower complexity than PCA could also be beneficial for reducing dimensionality more efficiently.

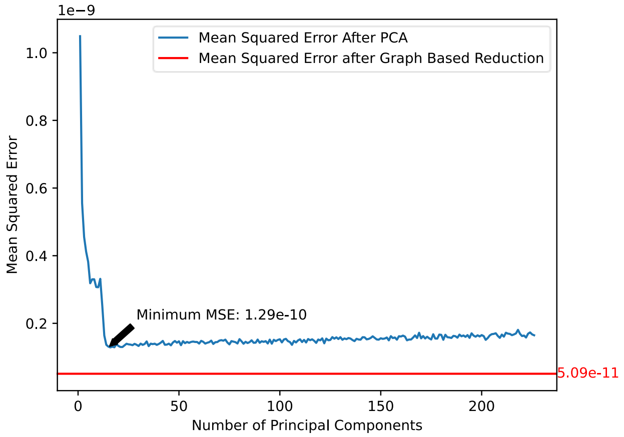

Fig. 2. A comparison of the mean squared errors of the dataset after PCA reduction (shown in blue), against our method.

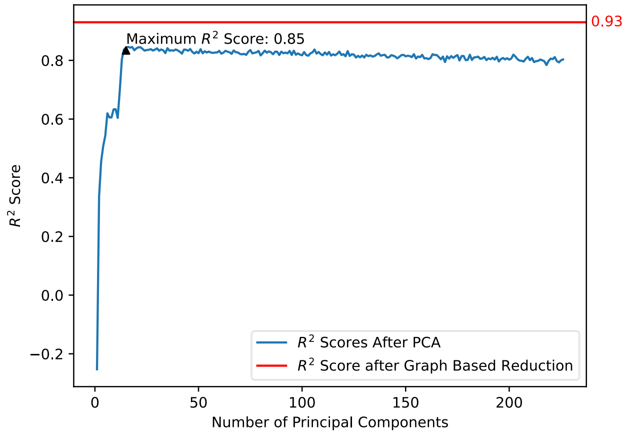

Fig. 3. A comparison of R^2 scores of the dataset after PCA reduction (shown in blue), against our method.

References

[1] Zhu, S., et al., "Intelligent Computing: The Latest Advances, Challenges, and Future," *Intelligent Computing,* vol. 2, p. 0006, 2023.

[2] S. Feng and H. Wang, "Comparison of PCA and LDA Dimensionality Reduction Algorithms based on Wine Dataset," in *33rd Chinese Control and Decision Conference (CCDC)*, 2021.

[3] X. Sun, L. Liu, C. Geng and S. Yang, "Fast Data Reduction With Granulation-Based Instances Importance Labeling," *IEEE Access},* vol. 7, pp. 33587-33597, 2019.

[4] R. N. S. Widodo, H. Abe and K. Kato, "Hadoop Data Reduction Framework: Applying Data Reduction at the DFS Layer," *IEEE Access,* vol. 9, pp. 152704-152717, 2021.

[5] H. Zhang and M. Gabbouj, "Feature Dimensionality Reduction with Graph Embedding and Generalized Hamming Distance," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018.

[6] C. Gakii, P. Mireji and R. Rimiru, "Graph Based Feature Selection for Reduction of Dimensionality in Next-Generation RNA Sequencing Datasets," *Algorithms,* vol. 15, 2022.

[7] A. Kazemipour and S. Druckmann, "Nonlinear Dimensionality Reduction Via Polynomial Principal Component Analysis," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018.

[8] J. Hopcroft and R. Tarjan, "Algorithm 447: efficient algorithms for graph manipulation," *Commun. ACM,* vol. 16, p. 372–378, 1973.

[9] S. Yu, W. Hannah, L. Peng, J. Lin, M. A. Bhouri, R. Gupta, B. Lütjens, J. C. Will, G. Behrens, J. Busecke, N. Loose, C. Stern, T. Beucler, B. Harrop, B. Hillman and A. Jenney, "ClimSim: A large multi-scale dataset for hybrid physics-ML climate emulation," in *Advances in Neural Information Processing Systems*, 2023.