Feature Selection for high dimensional data using Clustering and Correlation

Dinesh Mannari — Jignesh Kirti Nagda — Abhishek Anjani Kumar Singh — Jait Mahawarkar[1]*

Abstract
It has proven challenging to apply evolutionary algorithms to feature selection issues in high-dimensional spaces due to the ”curse of dimensionality” and the high computing cost. Our project extends the HFS-C-P, a three-phase hybrid feature selection algorithm. The approach tackles the problems of processing cost and dimensionality simultaneously by combining correlation-guided clustering and particle swarm optimization (PSO). The HFS-C-P algorithm combines three distinct feature selection techniques, each with its benefits. The search space is condensed in the first and second phases using a filter method and a clustering-based method, respectively. The ideal feature subset is located in the third phase using an evolutionary method with global searchability. The algorithm also incorporates a rapid correlation-guided feature selection approach, a symmetric uncertainty-based feature deletion method, and other features to enhance the performance of each phase.

Keywords
Clustering, Correlation, Hybrid Feature Selection(HFS), Particle Swarm Optimization(PSO)

[1] *Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

Contents

1. Problem and Data Description

1.1 Problem Statement:

Selecting relevant features from high-dimensional data is a key aspect of any Machine Learning model. Usually, it is assumed that high dimensional data leads to an increase in the performance of the model but it is not the case always. This problem will address the issues related to the concept of the Curse of dimensionality and High dimensional data. When data has many attributes it is often the case that some of the features are either redundant or irrelevant. Often high dimensional data requires high computation costs and also in the process of doing so sometimes removes important features which might otherwise be a key factor for deciding the outcome of the model.

1.2 Data Description

1.2.1 Arrhythmia

The Arrhythmia Data Set is a dataset from the UCI Machine Learning Repository that includes data on 279 people who have an arrhythmia, a disease of the heart marked by an irregular heartbeat. The electrocardiogram (ECG) dataset consists of 452 signals, each of which has 279 characteristics. Age, sex, height, weight, blood pressure, and 270 ECG features are among the dataset’s traits. Different techniques, including the Fourier transform, wavelet transform, and statistical features, are used to extract the ECG features. The dataset’s objective is to categorize the ECG signals into 16 distinct arrhythmia classes, each of which corresponds to a particular kind of arrhythmia.

2. Data Preprocessing & Exploratory Data Analysis

2.1 Handling Missing Values

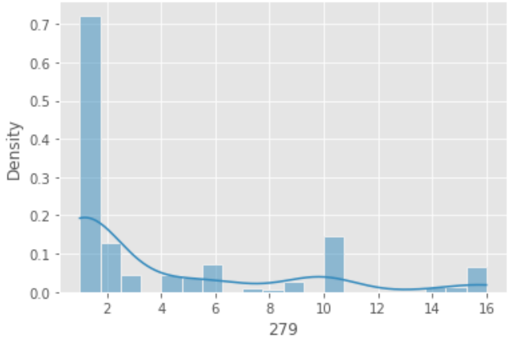
2.1.1 Arrhythmia

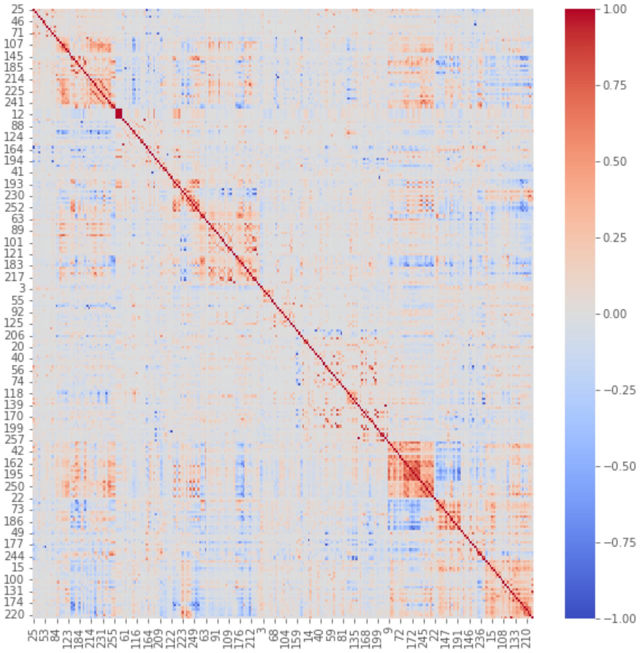
We started our data cleaning process for the Arrhythmia Data Set by checking for null values. To our surprise, there were no missing values in the dataset. However, it's always a good practice to check for missing values because they can affect the accuracy of the analysis. Next, we looked for unwanted characters in the dataset's attributes, which could be due to data collection or entry errors. We found unwanted characters and replaced them with NaN values. This is a common practice in data cleaning to standardize the data and make it easier to analyze. After replacing unwanted characters with NaN values, we imputed missing values with suitable values. In our case, we replaced the NaN values with the mean values of the respective columns. Finally, we normalized the data to prevent any one feature from dominating the others. Normalizing the data is a common technique used to scale the data and make it easier to compare different features. This cleaned data will now be used to apply our proposed method.

2.2 Exploratory Data Analysis

2.2.1 Arrhythmia

After cleaning the Arrhythmia Data Set, the next step was to explore the data and gain some insights. We started by plotting the distribution of the arrhythmia classes in the dataset to see how they were distributed. This helped us to get an idea of how common each type of arrhythmia was in the dataset. Plotting the distribution of classes is also essential to identify any class imbalance issues, which can impact the accuracy of the analysis. Next, we calculated the correlation matrix to find the correlation between different features in the dataset. The correlation matrix provides information on how strongly different features are related to each other. By analyzing the correlation matrix, we were able to identify which features were highly correlated with each other and which ones were not. Knowing which features are highly correlated with each other is important because it can help us to reduce the dimensionality of the dataset by eliminating redundant features





3. Algorithm and Methodology

3.1 Phase I - Removing Irrelevant Features

The input for the algorithm is the original set of features, denoted by F, and the set of class labels, denoted by C. The output of the algorithm is the set of strong relevance features, denoted by F'.

The algorithm proceeds as follows:

1. For each feature fi in the original set F, calculate its C-relevance value, denoted by SU(fi,C).

2. For each feature f_i in the original set F, calculate its C-relevance value, denoted by $SU(f_i, C)$. Determine a threshold value, denoted by ρ_0, using the formula:

$$\rho_0 = \min(0.1 \times SU_{\max}, \frac{SU_D}{\log D} - th)$$

where $SU_{\max}$ is the maximum C-relevance value among all features in F, D is the number of features in F, and th is a constant threshold value.

3. For each feature f_i in F, if its C-relevance value is greater than or equal to ρ_0, then save it into the set F'.

4. Output the set F'.

3.2 Phase II - Feature Clustering method

Input: The set of strong relevance features F'. from Phase will be used as input to this Phase.
Output: The feature cluster results : $cluster^k$ Phase 2 proceeds as follows:

1. Sort all the features in F by their C-relevance values, denoted the sorting results as U_0;

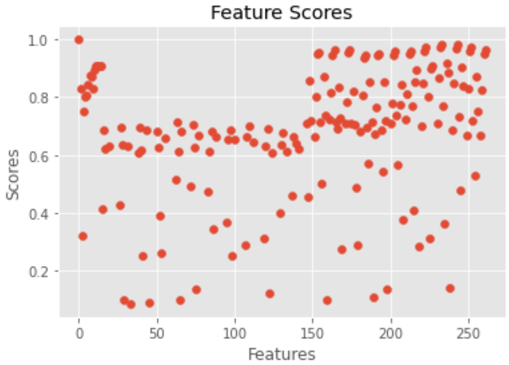
2. Let $k = 1$;
3. Set the temporary set $U_1 = U_0$;
4. Let the first feature in the U_1 to be f_{1-th}, and its C-relevance value to be $SU(f_{1-th}, C)$;
5. Initialize the k-th feature cluster to be $Cluster_k = f_{1-th}$;
6. Remove weak co-relation features :
7. for $i = 2 : |U_1|$ do from the second feature to the last one in the U_1;
8. if $dt(1-th, i) > \rho_1$ then Remove f_i from U_1, i.e., $U_1 = U_1/\{f_i\}$;
9. end
10. Now, We find all the co-relating features and append to the $cluster^k$
11. Set $U_0 = \frac{U_0}{\text{Cluster}_k}$.
12. If $|U_0| > 1$, let $k = k+1$, return to Step 3; otherwise, set the last cluster to be U_0, and output the clustering result.

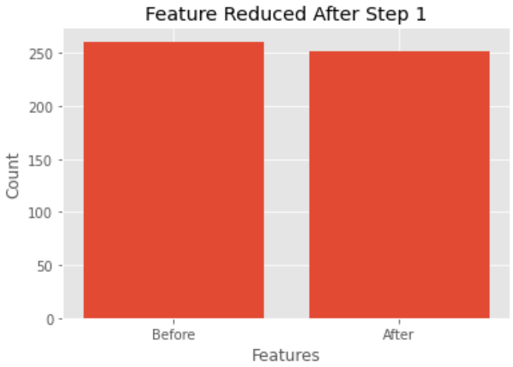
3.3 Phase III - IBPSO Algorithm

1. Set related parameters, including the size of swarm N, the maximum evaluation time FE_MAX and so on;
2. Initialize the swarm according to the method as below:
    - The M feature clusters, Cluster$j$, $j = 1, 2, \ldots, M$
    - The N particles, X_i, $i = 1, 2, \ldots, N$
    - Calculate the selected probability for each cluster, pcj, $j = 1, 2, \ldots, M$
    - For$i = 1 : N$
    - For$j = 1 : M$ uIfrand(0, 1) $< p_{cj}$
    - Randomly select a feature from the Cluster$j$, denoted by the a-th feature;
Set $Xi, j = a$; Else Set $X_{i,j} = 0$;
    - Output the N particles, X_i, $i = 1, 2, \ldots, N$;
3. Calculate the fitness of each particle in the swarm, and update its P_{best} and G_{best};
4. Update all the particles according to following equation: $pc_j = \frac{Cv_{\max,j}}{Cv_j}, \quad j = 1, 2, \ldots, M.$
5. Perform the adaptive disturbance based on the following:
$$x_{i,j} = \begin{cases} \frac{p_{\mathrm{bi},j}+g_{\mathrm{bi},j}}{2} + G(0,1) \times \sqrt{\frac{p_{\mathrm{bi},j}-g_{\mathrm{bi},j}}{2}}, & \text{if rand} > 0.5\ p \\ \text{otherwise} \end{cases}$$
6. If satisfy the stop condition, stop the algorithm and output the optimal solution; otherwise, go to Step 2.

4. Experiments and Results

4.1 Phase 1- filtering the features based on Symmetric uncertainty

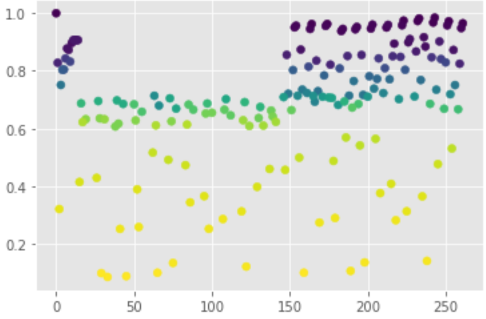




The first graph depicts the c-relevance scores of various features in a dataset, indicating their correlation or relevance to the target variable. These scores can aid in correlation-based clustering, which is an integral part of data analysis or machine learning projects.

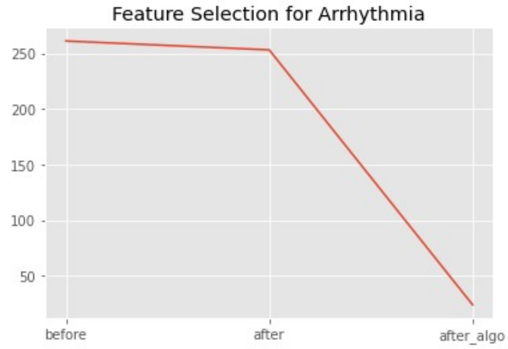
The second graph displays the impact of phase 1 of the algorithm on the features, both before and after its execution. The algorithm employs a formula to determine a suitable threshold based on the dataset being analyzed, rather than assuming a fixed threshold. This approach guarantees an appropriate threshold for the particular data and can lead to improved results.

4.2 Phase 2- Clustering the features



In Phase 2, the graph illustrates how clusters of features are formed after removing weakly correlated features. The representative feature is selected based on its high score, and other features are clustered around it. If such a feature is not found, another cluster is formed. The graph also demonstrates that these clusters are clearly visible, indicating the effectiveness of the clustering process in identifying groups of strongly correlated features. By eliminating weakly correlated features and clustering the remaining ones, data analysts and machine learning practitioners can streamline the dataset and enhance model performance by focusing on the most relevant features.

4.3 Phase 3- Particle swarm optimization algorithm



As shown in the graph during Phase 3, after implementing various steps outlined in the methodology, we utilized Particle Swarm Optimization to obtain the best features using the concepts of Pbest and Gbest of the particles. We iterated through Pbest and Gbest by updating the position of the particles. Additionally, to avoid converging on local optima, we utilized the technique of Difference Adaptive Differences, which reassigns the particle if it is converging optimally.

5. Deployment and Maintenance

The Feature selection model algorithm has been packaged for deployment and maintenance, with all the necessary functions and scripts included. The package contains functions for swarm initialization, particle position updating, fitness evaluation, and global best position determination. Additionally, the package provides the ability to set bounds on particle positions to ensure they remain within a specified range.

Using the package is straightforward, with the user simply importing the required functions and providing the necessary parameters. The package has been designed to be flexible and customizable, enabling users to modify algorithm parameters and settings to suit their specific optimization problems.

The package has been extensively tested on various optimization problems, proving to be effective and efficient. It also includes comprehensive documentation and examples to facilitate its use and maintenance. The package can be found used by following command:
pip install Hfs_Cp allowing for easy installation and use in regular classification problems based on features selected by our algorithm.

6. Summary and Conclusions

The article presents a new hybrid feature selection algorithm, HFS-C-P, to address the complex problem of high-dimensional feature selection. The proposed algorithm incorporates the benefits of three types of feature selection algorithms in a three-phase hybrid framework. The first and second phases are designed to minimize the search space of the particle swarm optimization-based method used in the third phase, which significantly improves the search speed of the swarm. The second phase also reduces the computational cost of clustering features and eliminates the need to specify the clustering number.

In addition, the integer binary particle swarm optimization algorithm used in the third phase benefits from relevance-guided swarm initialization and difference-based adaptive disturbance, which enhances its effectiveness in solving feature selection problems. Experimental results demonstrate that HFS-C-P outperforms several state-of-the-art feature selection algorithms on various datasets.

Overall, the proposed algorithm provides a promising solution to the challenging problem of high-dimensional feature selection, offering faster and more effective feature selection results.

While our algorithm is effective in extracting relevant features, it can face challenges when dealing with an extremely large feature space, resulting in significant computation time. This can be a limiting factor when processing large datasets, where the number of features is very high. To address this challenge, researchers can explore various techniques for optimizing the algorithm's computational efficiency, such as data pre-processing or parallel computing. Further research in this area may lead to the development of more efficient feature selection algorithms that can handle large feature spaces without significant computation time.

Acknowledgments

We wish to express our sincere gratitude to Professor Hasan Kurban and the Head Teaching Assistant Md Rysul Kabir for their invaluable contributions to this project. Their guidance, support, and expertise have been instrumental in the successful completion of this endeavor.

Professor Kurban's vast knowledge and insightful comments were invaluable in shaping our understanding of the subject matter and improving the quality of our work. The Head Teaching Assistant's prompt responses and helpful suggestions were instrumental in resolving any queries or issues that arose during the project.

We are truly grateful for the opportunity to work with such esteemed professionals and appreciate their commitment to our project. We would like to thank them both for their invaluable contributions, without which this project would not have been possible.

References

Xian-Fang Song, Yong Zhang, Dun-Wei Gong and Xiao-Zhi Gao, ”A Fast Hybrid Feature Selection Based on Correlation-Guided Clustering and Particle Swarm Optimization for High-Dimensional Data” IEEE Trans. Cybern., vol. 45, no. 5, pp. 9573 - 9586, Sept 2022

K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, “A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition,” IEEE Trans. Cybern., vol. 47, no. 6, pp. 1496–1509, Jun. 2017.

M. Liang and X. Hu, “Feature selection in supervised saliency prediction,” IEEE Trans. Cybern., vol. 45, no. 5, pp. 914–926, May 2015.

B. Hu et al., “Feature selection for optimized high-dimensional biomedical data using an improved shuffled frog leaping algorithm,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 15, no. 6, pp. 1765–1773, Nov./Dec. 2018.

L. Yu and H. Liu, “Efficient feature selection via analysis of relevance and redundancy,” J. Mach. Learn. Res., vol. 5, pp. 1205–1224, Oct. 2004.

E. Hancer, B. Xue, and M. Zhang, “A survey on feature selection approaches for clustering,” Artif. Intell. Rev., vol. 53, no. 2, pp. 4519–4545, 2020.