Modeling the Pareto Set and Low-Rank solutions in Multi-Objective Optimization

SIEMENS

Pavan Yachamaneni, Shuangyu Zhao, Tanishq Tanmay, Yena Hong

Purpose

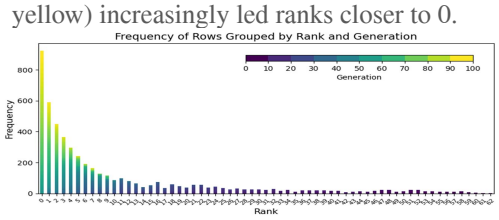
Our project aims to develop and evaluate Machine Learning methods for creating rule sets in *multi-objective optimization* problems, particularly focusing on identifying patterns, relationships that defines Pareto-efficient solutions on standard ZDT-1 datasets. The goal is to establish effective rules for *explaining*, *identifying and classifying Pareto sets.*

Objectives

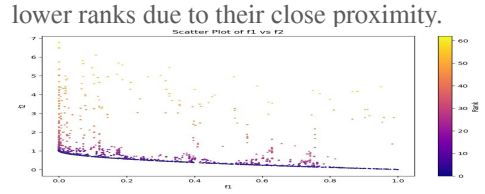
Our project focuses on investigating datasets with 5, 10, and 15 design variables, generated over 100 generations using the pymoo. Each sample is assigned a rank using NSGA-2 based on its corresponding objective values f1 and f2. The primary goal is to generate rules/train models for classifying Pareto sets based on input design parameters.

Exploratory Data Analysis

Data generation progression (from purple to yellow) increasingly led ranks closer to 0.



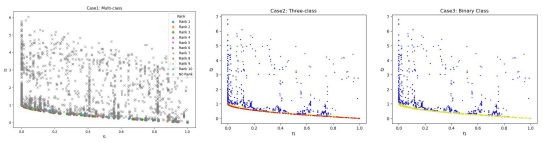
It's challenging to distinguish the Pareto set, indicated by the purple dots, from nearby lower ranks due to their close proximity.



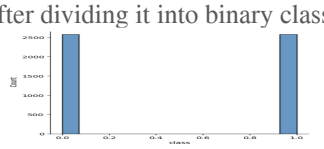
Data Preprocessing

Our data preprocessing, aimed at optimizing the y variable, involved three categories:

- Multi-class with ten y classes (ranks 0-9),
- Three-class with three y classes (lower, middle, high ranks from 0-62), and
- Binary class with two y classes (lower and high ranks from 0-62).



We applied basic supervised machine learning algorithms to predict y in three cases, and also adjusted group boundaries in case 2 and 3 for diverse y classes. The results indicated the lowest accuracy in multi-class, the highest accuracy with binary class, leading us to choose binary class as our final y variable. The data was then balanced, seen in the graph below, after dividing it into binary classes.



Approach

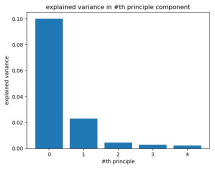
The core underlying aspect of this project lies in partitioning the data to create Pareto, Non-Pareto solution classes with respect to ranks either with or without band gap.

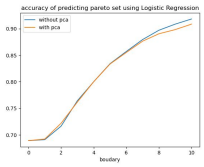
Models implemented

- Decision Tree
- SVM
- Logistic Regression
- Feedforward Neural Network
- Random Forests

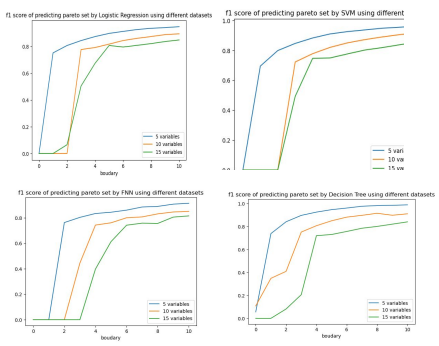
Approach-1: Without Gap

In all datasets, the first two principal components can account for a significant portion of the variance. Utilizing Principal Component Analysis(PCA), we can effectively reduce the dimensionality of datasets with varying numbers of attributes to two, while retaining the majority of the information.





Because, the pareto/near-pareto set is focused, f1 score is chosen for estimation as it approximate both classes. The datasets with lower number of attributes have better performances of models.



The dimensions of these three datasets have already been reduced to 2 using PCA. Consequently, we union these three datasets to train more generalized models.

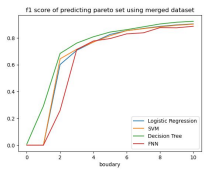
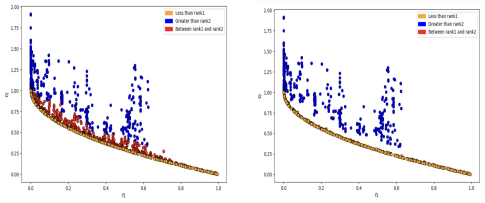


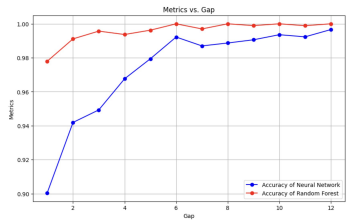
Figure representing F-1 scores for all 3 datasets combined together

Approach-2: With Gap

The data was partitioned into three parts using two rank separator values like in the left graph shown below.



Then, the data points with ranks in between these rank separators were removed, right graph shown above, to create a gap between Pareto and Non- Pareto sets. All the classifiers were implemented on this dataset with gap, but Random Forests and Neural Network showed best results. The gap was then increased from 1 to 12 and a line graph (shown below) was plotted showcasing the accuracies of both classifiers.

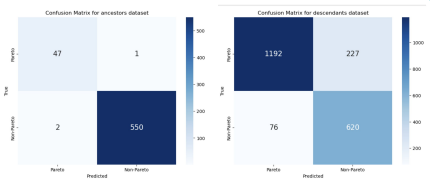


It was observed that as we increase the gap, the classifiers perform better and gives better accuracies.

Final Inference: Ancestors and Descendant

In our efforts to use machine learning, Data engineering has a big impact on the rule generation for ranks which we get from NSGA-2, which is like a smart system that figures out patterns in an evolutionary way.

Therefore, our data is partitioned into 2 categories named ancestors and descendants where algorithm is trained from ancestors and in turn tested across descendants where knowledge is transferred with rules incorporated.



Confusion Matrix depicting high accuracy scores for **ancestors**, low for **descendants** using Random Forest algorithm.

Conclusion

- For Approach-1
  - Decision tree works best.
  - PCA can help to train the generalized model, but the variables are more, the accuracies are lower.
- For Approach-2
  - Creating gap always doesn't ensure to give perfect model but guarantee near perfect classification of pareto solutions.
- In leveraging machine learning, data manipulation can significantly influences rule generation for ranks using an intelligent system uncovering patterns through evolutionary processes.

Key-Words

1. Rank
2. ZDT Data
3. Pareto Set
4. Multi-Objective optimization
5. NSGA-2
6. Gap
7. Boundary
8. Supervised Learning
9. Pymoo
10. Generations
11. Design Parameters
12. Accuracy-Precision-Recall-F1 score

MICHIGAN STATE UNIVERSITY