A Comparative Analysis of SWAY and XPLN with Modified Optimization Techniques

Ashish Sanjay Joshi
Department of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: ajoshi24@ncsu.edu

Aoishi Das
Department of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: adas23@ncsu.edu

Swarnamalya Mohan
Department of Computer Science
North Carolina State University
Raleigh, North Carolina
Email: smohan7@ncsu.edu

***Abstract*—Due to its wide applicability, the problem of multi-objective semi-supervised optimization is attracting increasing attention in machine learning. We examine mathematical models to implement the optimization problem to SWAY and XPLN. In this paper, we propose an effective approach using the T-distributed Stochastic Neighbor Embedding (TSNE) technique to enhance the functional capacity of SWAY algorithm. In addition, Random Forest Classifier ensemble model and is used as an effective implementation to the XPLN feature.When tested on multiple semi-supervised based systems, this implementation performs as well or even better as the prior state-of-the-art, while running 50-80 times faster on average.**

I. Introduction

Optimization is a very important process in engineering. Engineers can create better production only if they make use of optimization tools in reduction of its costs including the consumption time. In this era of Machine Learning, efficient and optimized approaches are very pivotal in the design of accurate and reliable classification and modelling systems. The existing methods demonstrates the use of recursive bi-clustering that splits the candidates into two parts based on their decisions. However, it can be computationally expensive, especially for large datasets. As the process involves repeatedly applying clustering algorithms on subsets of the data, the computational cost can quickly accumulate, leading to increased processing time and resource requirements. As an effective solution to address this shortcoming, we propose to use the t-SNE (t-Distributed Stochastic Neighbor Embedding) as the pre-processing strategy in implementation of SWAY algorithm. T-SNE (t-Distributed Stochastic Neighbor Embedding) is a popular dimensionality reduction technique which helps to reduce the dimensionality and noise in the data. Using t-SNE to project the data into a lower-dimensional space can make it easier to implement the clustering technique and identify patterns in the data.

The technique deployed for the rule generation and classification in xpln uses the range of scores that distinguishes the best from rest. As we tend to use high dimensional datasets with noisy or missing data, there is a need for a versatile algorithm that can be applied for the classification tasks for wide range of datasets. We intend to use Random Forest Classifier to classify the data as best. Owing to its robustness to noise, missing data and outliers, it can be used as part of rule generation and classification. Also the non-parametric nature and reduced variance of the random forest algorithm makes it the most reliable and stable model in XPLN.

To demonstrate the effectiveness of this approach, we explore the following research questions.

RQ1 How effective is the t-SNE for diverse datasets?

Here we assess the effectiveness with all the given 11 datasets with varying structures and project the improvements in the test values attained.

RQ2 How fast is t-SNE?

Here we find that due to lesser dimensions after applying t-SNE, the optimized algorithm runs 50-80 times faster than before.

RQ3 How does the approach compare to existing methods in terms of accuracy?

The use of random forest classifier will undoubtedly yield higher accuracy as compared to other state-of-art models. The results depict the significant difference obtained with the usage in XPLN.

RQ4 How does the approach perform in the presence of noise or missing data?

t-SNE is robust to outliers in the data, which means that it can handle datasets with noisy or outlying data points.

II. Related Work

Extensive literature survey has been performed to evaluate the study of multi-objective semi-supervised explanation problem and the diverse approaches involved. The work demonstrated in [1] has proposed the pattern recognition method that identifies patterns based on their similarity and their association with the outcome of interest. The practical purpose of developing this pattern recognition method is to group patients, who are injured in transport accidents, in the early stages post-injury. A multi-objective optimization model is proposed to group patients. An objective function is the cost function of k-medians clustering to recognize the similar patterns.

Another objective function is the cross-validated root-mean-square error to examine the association with the total cost. The best grouping is obtained by minimizing both objective functions. As a result, the multi-objective optimization model is a semi-supervised clustering which learns health service use patterns in both unsupervised and supervised ways. It also introduces an evolutionary computation approach includes stochastic gradient descent and Pareto optimal solutions to find the optimal solution. In addition, we use the decision tree method to reproduce the optimal groups using an interpretable classification model.The hypothesis for developing a multi-objective optimization model is that it works better than single-objective ones to achieve the aims of the study as it intends to contribute to both objective functions. In addition, reformulating a multi-objective optimization model into a weighted average objective function or a goal programming model leads to a failure for finding non-dominated optimal solutions. A drawback for k-medians clustering (also for k-means clustering) is that it is sensitive to the choice of the initial cluster centres i.e. different initial centres lead to different clusters at the end of algorithm. The implementation in [2] also depicts an other drawback of specifying number of clusters at the beginning. However, in many cases of unsupervised learning, the researchers or practitioners are not aware of the number of clusters.

The experimental study in [3] proposed a probabilistic model for semisupervised clustering based on Hidden Markov Random Fields (HMRFs) that provides a principled framework for incorporating supervision into prototype-based clustering. The model generalizes a previous approach that combines constraints and Euclidean distance learning, and allows the use of a broad range of clustering distortion measures, including Bregman divergences (e.g., Euclidean distance and I-divergence) and directional similarity measures (e.g., cosine similarity).The drawback is that (HRMF) Hidden Markov Random Fields is sensitive to its initial parameters, which can result in suboptimal results if the initialization is not well-tuned.HRMF can be computationally expensive, particularly when dealing with high-dimensional image data. The model's complexity can also make it challenging to train and optimize.

The problem of semi-supervised classification is attracting increasing attention in machine learning. Semi-Supervised Support Vector Machines (SVMs) are based on applying the margin maximization principle to both labeled and unlabeled examples. Unlike SVMs, their formulation leads to a non-convex optimization problem. A suite of algorithms have recently been proposed in [4] for solving SVMs.Deterministic annealing (DA) is a global optimization heuristic that has been used to approach hard combinatorial or non-convex problems. In the context of SVMs , it consists of relaxing the discrete label variables to real-valued variables. But it is found that the original DA algorithm alternates between the optimizations and can be understood as block coordinate optimization. The experimental results have proved that it was significantly slower than the other algorithms; its direct gradient-based optimization counterpart, VDA, is usually much faster.

The study in [5] shows that deep generative models and approximate Bayesian inference exploiting recent advances in variational methods can be used to provide significant improvements, making generative approaches highly competitive for semi-supervised learning.It provides new framework for semi-supervised learning with generative models, employing rich parametric density estimators formed by the fusion of probabilistic modelling and deep neural networks.In particular, astochastic variational inference algorithm has been used that allows for joint optimisation of both model and variational parameters, and that is scalable to large datasets. A limitation of the models presented is that they scale linearly in the number of classes in the data sets. Having to re-evaluate the generative likelihood for each class during training is an expensive operation.

III. Methods

A. Algorithms

This section demonstrates our understanding of all the algorithms used for this research study.

1) SPLIT: This algorithm is used for dividing the given candidates into 2 clusters. Two different algorithms are used for splitting continuous decision spaces and binary decision spaces.

a) SPLIT for continuous decision spaces: Here, FastMap heuristic used to split the candidates into two groups. It uses following steps :

- Pick any random candidate from input candidates. Let's call it *rand*
- Find a candidate which is furthest apart from the *rand*. Let's call it *east*.
- Find a candidate which furthest apart from the *east*. Let's call it *west*.
- Imagine a straight line connecting *east* and *west*. We will now project every input candidate on this straight line.
- We will now sort all the input candidates on the basis of their perpendicular distance from the line joining *east* and *west*.
- Among this sorted set of candidates, the first half will be one cluster and the second half will be the other cluster being returned.

b) SPLIT for continous decsion spaces: Instead of projecting candidates on a straight line, here candidates are mapped into a circle. Center for this circle is picked randomly among all the input candidates. All the candidates are mapped into this circle by using their Jaccard distance from the center. We then put all the candidates with similar radius into the same circumference. The points with the smallest and largest radius are chosen as *east* and *west*. Then the candidates with minimum θ in each annulus area from the east will be one cluster and the candidates with maximum θ from the west will be the other cluster being returned.

2) BETTER: Given two clusters of candidates, this algorithm tells us which is the better set of candidates. Zitzler's indicator predicate is used in this function. This will be explained in detail in the Performance Measures section.

3) SWAY: SWAY is an acronym for "The Sampling Way". Given a set of candidates, this algorithm will return the best set of candidates and some sample of the remaining candidates. It uses following steps :

- If the size of input is lesser than a threshold we will return all the items.
- Split the input set of candidates into two clusters using the SPLIT function described above.
- Use the BETTER function described above to judge which cluster is the better one.
- If the first cluster is better then call SWAY recursively on it. Let's call the output of this step cluster1.
- If the second cluster is better then call SWAY recursively on it. Let's call the output of this step cluster2.
- Return Cluster1 and Cluster2.

4) SWAY2: We tried to modify the implementation of SWAY2 by mainly altering the half function. The following changes were introduced in half:

- We applied t-SNE[9] with number o fcomponents = 2. It basically performs dimensionality reduction and brings down the number of dimensions to 2. The value of perplexity was set to 10 after few trials since perplexity should be smaller than number of samples and in certain instances the number of samples were becoming smaller than other higher values of perplexity due to recursive halving.
- Then a custom comparator was written to calculate the distance of every point from origin (based on Euclidean distance).
- Using that custom comparator the rows were sorted according to the distance values.
- The midpoint was found and rows towards left of mid point were put in left and rows right to midpoint were put in right.

Apart from this, the rest functions used in SWAY were not modified.

5) XPLN: This algorithm tries to explain the results generated by SWAY. It tries to understand on what basis SWAY divided the candidates between best and rest clusters. It uses following steps:

- Find ranges that distinguish best from rest.
- Sort the ranges by their values using the VALUE algorithm.
- Try the first ranked range. Then try the combination of first and second ranked range etc.
- Return the best rule as decided by the VALUE algorithm.

6) XPLN2: We tried to modify the implementation of XPLN2. The following changes were introduced:

- For finding out a rule we wanted to train a traditional Machine Learning model so that it learns how to distinguish between best and rest
- This task could simply be understood as a binary classification problem.
- Out of the various models available like Decision Trees, Random Forest, SVM, Logistic Regression : Random Forest seemed like a fair choice since it is an ensemble learning algorithm which is robust towards overfitting unlike Decision Trees.
- We generated x_best and x_rest data from the best and rest rows and target was created by appending 'best' and 'rest' as labels.
- The model was trained on this data.
- All the rows were put in test data and was fed to the model to obtain the predictions.
- It returned the best and rest predictions.
-

7) VALUE: Given a range, this algorithm tell us how well it selects for best using probability and support. It uses following steps: For simplicity let's assume this: There are 50 candidates in best cluster and 100 candidates in the rest cluster. The input range successfully differentiates 25 best candidates from 50 rest candidates.

- Let's calculate probability that the given range will identify a best candidate. It will be 25/50 = 0.5. Let's call this probability *b*.
- Let's calculate probability that the given range will identify a rest candidate. It will be 50/100 = 0.5. Let's call this probability *r*.
- Return $b\hat{2}$/(b+r) as the value of this range.

B. Data

The following datasets have been used in the project. We have plotted heatmaps for each dataset to visualize which features have high correlation with the target features.

1) auto2.csv:

- Number of datapoints : 93 , Number of columns : 23
- Target columns : CityMPG+ , HighwayMPG+ , Weight- , Class-
- High correlation with target : Engine_size, Horsepower, Fuel_tank_capacity, Passenger_capacity, Length, Wheelbase, Width, U-turn_space

2) auto93.csv:

- Number of datapoints : 499 , Number of columns : 19
- Target columns : Lbs-, Acc+, Mpg+
- High correlation with target : Clnds, Volume

3) china.csv:

- Number of datapoints : 499 , Number of columns : 19
- Target columns : N_effort -
- High correlation with target : EffortX, Added, AFP

4) coc1000.csv:

- Number of datapoints : 1000 , Number of columns : 25
- Target columns : LOC+ , AEXP- , PLEX- , RISK- , EFFORT-
- High correlation with target : ACAP, PCAP, SCED

5) coc10000.csv:

- Number of datapoints : 10000 , Number of columns : 25
- Target columns : Loc+ , Risk- , Effort-
- High correlation with target : Acap, Pcap , Sced

6) healthCloseIsses12mths0001-hard.csv:

- Number of datapoints : 10000 , Number of columns : 8
- Target columns : MRE- , ACC+ , PRED40+
- High correlation with target : None of the features show any good amount of correlation with the target features

7) healthCloseIsses12mths0001-easy.csv:

- Number of datapoints : 10000 , Number of columns : 8
- Target columns : MRE- , ACC+ , PRED40+
- High correlation with target : N_estimators

8) nasa93dem.csv:

- Number of datapoints : 93 , Number of columns : 29
- Target columns : Kloc+ , Effort- , Defects- , Months-
- High correlation with target : idX, centerX, YearX

9) pom.csv:

- Number of datapoints : 10000 , Number of columns : 13
- Target columns : Cost- , Completion+ , Idle-
- High correlation with target : Criticality, Criticality Modifier, Dynamism, Size

10) SSM.csv:

- Number of datapoints : 239360 , Number of columns : 15
- Target columns: numberiterations-, timetosolution-
- High correlation with target : jACOBI

11) SSN.csv:

- Number of datapoints : 53662 , Number of columns : 19
- Target columns : PSNR- , Energy-
- High correlation with target : no_cabac , Seek

C. Performance measures

For ranking the data, the clustering algorithm can be tweaked to convert it into an optimization algorithm. One of the problems with such optimization is deciding how to trade off between competing concerns since we are trying to optimize a multi objective function. The **standard boolean domination (bdom)** predicate says one thing dominates another if:

- RULE 1: it never worse on any goals
- RULE 2: it is better for at least one goal

However, when the number of goals exceed 3, boolean domination fails to distinguish because according to RULE1: "never worse on any goal" condition fails because as the number of goals increase, the more ways you can be a tiny bit worse on at least one goal. Hence nothing seems to be better than anything else. In such cases Zitzler Continuous domination is used to compute which one is better. The BETTER function implements the Zitzler Continuous domination principle in the code. The main idea behind Zitzler is it judges the domination status of pair of individuals by running a "what-if" query which checks the situation when we jump from one individual to another, and back again.
For the forward jump,

$$s_1 = -\sum_i e^{w_i(a_i-b_i)/n} \qquad (1)$$

For the backward jump,

$$s_2 = -\sum_i e^{w_i(b_i-a_i)/n} \qquad (2)$$

where a_i and b_i are the values on the same feature from two rows, n is the number of goals and w_i is the weight like -1,1 if we are minimization or maximizing the goal . According to **Zitzler**[6] , one example is preferred to another if we lost the least jumping to it; i.e.

$$s_1 < s_2$$

.

D. Summarization methods

1) Influence of Methods on Budget: The methods that are emphasized have been evaluated using the correlation matrix to compare the performance of different methods with the total number of independent variables used in each method. The plot of the number of independent variables vs accuracy is depicted. The graph includes a series of data points or lines representing the performance of different methods, with the size or color of each point or line indicating the relative complexity or number of independent variables used.A clear trend should emerge showing that better performing methods tend to have a lower total number of independent variables, while less effective methods tend to require a higher number of independent variables.

2) Statistical Analysis: Statistical tests are used to determine the likelihood that an observed difference or relationship between two or more groups or variables is due to chance or is statistically significant. They are an important tool in scientific research and are used to make conclusions about populations based on sample data.

a) Cohen's d effect size test: : Cohen's d is a measure of the standardized difference between two means of data. It indicates the magnitude of the effect by comparing the difference between the means to the pooled standard deviation of the two groups. Since the prime objective of this project is to determine the significant levels of optimization, Cohen's d test would be an apt strategy for meaningful comparisons to be made across the original and the modified methods.This information can be important in helping to interpret the results of a study and to evaluate the practical significance of any findings.

b) Bootstrap Test: : Bootstrap is a non-parameteric significant testing method that can be used to assess the variability of a statistic. Since Cohen's d test is not suitable for non-normal distributions, we use the boostrap testing that is more powerful and flexible specifically for unknown or non-normal population distribution. In bootstrap testing, a large number of bootstrap samples (or resamples) are drawn from the original dataset, with replacement. Each bootstrap sample is of the same size as the original dataset. For each bootstrap sample, the statistical difference/delta is calculated and the significant differences are counted. This process is repeated many times (typically several hundred or thousand times), resulting in a count that indicates if the distributions are different.

The results section below deploys the usage of Cohen's d test and Bootstrap test analysis to evaluate the notable differences in the optimization methods being used.

IV. Results

We ran SWAY1, SWAY2, XPLN1, XPLN2 on all the provided datasets for 20 iterations. We compared performance of SWAY2 and XPLN2 with SWAY1 and XPLN1 by considering them baseline. We chose cliff's delta as effect size test and bootsrap as significance test. We chose non parametric tests because parametric tests make certain assumptions about the distribution of the data. On analyzing the datasets, we concluded that it's difficult to make assumptions about their distributions since it's multi objective and they can turn to be unrealistic. We have included the result table and statistical tests table for each dataset below.

1) *auto2.csv:*

	CityMPG+	HighwayMPG+	Weight-	Class-	n_evals avg
all	21	28	3040	17.700	0
sway1	29.300	33.750	2185.750	9.950	5
sway2	28.200	32.300	2261	9.830	5
xpln1	28.550	32.850	2357.500	10.800	5
xpln2	28.550	33.550	2283.250	10.070	5
top	33	39.800	2052	8.880	93

	CityMPG+	HighwayMPG+	Weight-	Class-
all to all	=	=	=	=
all to sway1	≠	≠	≠	≠
sway1 to sway2	≠	≠	≠	≠
sway1 to xpln1	≠	≠	≠	≠
sway2 to xpln2	=	=	=	=
sway1 to top	≠	≠	≠	≠

2) *auto93.csv:*

	Lbs-	Acc+	Mpg+	n_evals avg
all	2800	15.500	20	0
sway1	2089.500	16.535	33	6
sway2	2184.750	16.780	29	6
xpln1	2172.700	16.265	29.500	6
xpln2	2188.500	16.635	29.500	6
top	1985	18.780	40	398

	Lbs-	Acc+	Mpg+
all to all	=	=	=
all to sway1	≠	≠	≠
sway1 to sway2	≠	≠	≠
sway1 to xpln1	≠	=	≠
sway2 to xpln2	≠	=	=
sway1 to top	≠	≠	≠

3) *china.csv:*

	N_effort-	n_evals avg
all	2098	0
sway1	806.850	6
sway2	1038.150	6
xpln1	800.100	6
xpln2	1040.550	6
top	148	499

	N_effort-
all to all	=
all to sway1	≠
sway1 to sway2	≠
sway1 to xpln1	≠
sway2 to xpln2	=
sway1 to top	≠

4) *coc1000.csv:*

	LOC+	AEXP-	PLEX-	RISK-	EFFORT-	n_evals avg
all	1061.950	3	3	5.150	19287.300	0
sway1	1046.800	2.850	2.900	4.100	19264	6.150
sway2	1297.950	2.750	2.850	5.500	43879.150	6.150
xpln1	1055.900	3	3	4.900	19798.700	6.150
xpln2	1292.450	2.750	2.850	5.650	43869.300	6.150
top	1534.700	2	1	3.150	29972	1000

	LOC+	AEXP-	PLEX-	RISK-	EFFORT-
all to all	=	=	=	=	=
all to sway1	≠	=	≠	≠	≠
sway1 to sway2	≠	≠	≠	≠	≠
sway1 to xpln1	≠	≠	≠	≠	≠
sway2 to xpln2	=	=	=	=	=
sway1 to top	≠	≠	≠	≠	≠

5) *coc10000.csv:*

	Loc+	Risk-	Effort-	n_evals avg
all	1006.150	5.050	20153.500	0
sway1	984.300	4.100	16570.750	8
sway2	984.300	4.100	16570.750	8
xpln1	1009.950	5	19854.400	8
xpln2	1010.950	5	19789.450	8
top	1959	0	17752	10000

	Loc+	Risk-	Effort-
all to all	=	=	=
all to sway1	≠	≠	≠
sway1 to sway2	=	=	=
sway1 to xpln1	≠	≠	≠
sway2 to xpln2	=	≠	≠
sway1 to top	≠	≠	≠

6) *healthCloseIsses12mths0001-hard.csv:*

	MRE-	ACC+	PRED40+	n_evals avg
all	74.978	7.130	25	0
sway1	73.808	7.580	25	8
sway2	73.165	7.690	25	8
xpln1	73.784	7.548	25	8
xpln2	73.634	7.572	25	8
top	64.910	11.330	25	10000

	MRE-	ACC+	PRED40+
all to all	=	=	=
all to sway1	≠	≠	≠
sway1 to sway2	≠	≠	≠
sway1 to xpln1	≠	≠	≠
sway2 to xpln2	≠	≠	≠
sway1 to top	≠	≠	≠

7) healthCloseIsses12mths0011-easy.csv:

	MRE-	ACC+	PRED40+	n_evals avg
all	119.318	-12.226	0	0
sway1	5.027	-0.026	79.163	8
sway2	5.083	-0.080	79.163	8
xpln1	0	0	83.330	8
xpln2	15.084	-0.106	66.664	8
top	0	0	83.330	10000

	MRE-	ACC+	PRED40+
all to all	=	=	=
all to sway1	≠	≠	≠
sway1 to sway2	≠	≠	≠
sway1 to xpln1	≠	≠	≠
sway2 to xpln2	≠	≠	≠
sway1 to top	≠	≠	≠

8) nasa93dem.csv:

	Kloc+	Effort-	Defects-	Months-	n_evals avg
all	47.500	252	2007	21.400	0
sway1	14.635	72.700	565.400	13.300	5
sway2	28.625	140	1087.150	126.265	5
xpln1	12.930	67.800	474.350	12.840	5
xpln2	37.435	314.380	1460.550	19.025	5
top	4.700	15.120	146.800	8.580	93

	Kloc+	Effort-	Defects-	Months-
all to all	=	=	=	=
all to sway1	≠	≠	≠	≠
sway1 to sway2	≠	≠	≠	≠
sway1 to xpln1	≠	=	=	
sway2 to xpln2	≠	≠	≠	≠
sway1 to top	≠	≠	≠	≠

9) pom.csv:

	Cost-	Completion+	Idle-	n_evals avg
all	326.922	0.899	0.232	0
sway1	242.491	0.904	0.208	8
sway2	242.491	0.904	0.208	8
xpln1	236.505	0.894	0.240	8
xpln2	235.480	0.899	0.243	8
top	138.750	1	0	10000

	Cost-	Completion+	Idle-
all to all	=	=	=
all to sway1	≠	=	≠
sway1 to sway2	=	=	=
sway1 to xpln1	≠	=	≠
sway2 to xpln2	≠	≠	≠
sway1 to top	≠	≠	≠

10) SSM.csv:

	NUMBERITERATIONS-	TIMETOSOLUTION-	n_evals avg
all	6.950	134.377	0
sway1	4.900	121.666	10
sway2	4.900	121.666	10
xpln1	5.450	121.482	10
xpln2	5.400	121.263	10
top	4	59.982	239360

	NUMBERITERATIONS-	TIMETOSOLUTION-
all to all	=	=
all to sway1	≠	≠
sway1 to sway2	=	=
sway1 to xpln1	≠	≠
sway2 to xpln2	≠	≠
sway1 to top	≠	≠

11) SSN.csv:

	PSNR-	Energy-	n_evals avg
all	45.6285	1238.06	0
sway1	42.634	1267.21	9
sway2	42.634	1267.21	9
xpln1	45.3805	1432.33	9
xpln2	45.302	1460.05	9
top	26.7075	466.114	53662

	PSNR-	Energy-
all to all	=	=
all to sway1	≠	≠
sway1 to sway2	=	=
sway1 to xpln1	≠	≠
sway2 to xpln2	≠	≠
sway1 to top	≠	≠

V. Discussion

It is important to consider potential sources of error or bias that could affect the validity of the results or conclusions. The purpose of a threats to validity section in a quantitative study is to acknowledge any factors that might have undue influence on the research or skew the data being collected. With relevance to this project, there exists some limitations in the usage of t-SNE technique as it is non-deterministic in nature, yielding different results each time it is run on the same data with the same parameters. This can make it difficult to reproduce results and compare different runs of the algorithm.Also, t-SNE has several hyperparameters that can affect the quality of the results, such as the perplexity and the learning rate. These hyperparameters can be difficult to tune, and the choice of hyperparameters can impact the quality of the results. As an alternative, dimensionality reduction and visualization of high-dimensional data can be performed using high-end and more sophisticated techniques like Autoencoder and Isomap.

VI. Conclusion

In this paper we have shown that even after reducing the dimensionality by using t-SNE, SWAY2 wasn't able to beat SWAY's performance. On the other hand, XPLN2 was able to beat XPLN1's performance. We used Random Forest Classifier in XPLN2 to understand how SWAY2 is dividing the candidates into best and rest.

VII. Future Work

Modifying SWAY could involve changing various parts of the algorithms like modifying the half function, better function, using a different approach for calculating the distance between two rows. In the implementations listed above we tried some of these changes. Some of the proposed changes that could be implemented are:

A. Changing the optimization algorithm

Finding a single optimum approach for multi-objective optimization when there are more than four dependent features can be difficult because the method chosen will rely on the particular situation and goals that need to be optimized. However, some approaches work better than others for solving higher-dimensional issues.

For instance, evolutionary algorithms with the capacity to maintain a diverse collection of solutions and completely traverse the search space, such as NSGA-II[7] and SPEA[8], are well known to be efficient for higher-dimensional problems. These techniques can manage a large number of dependent features because they employ a population-based approach in which many solutions are assessed and refined in each iteration.

B. Using alternate methods for splitting the dataset into half

1) BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies): It is a hierarchical clustering algorithm that uses a tree structure to represent the clustering hierarchy, and it is designed to be fast and efficient for large datasets.

2) Gaussian Mixture Models (GMM): GMM is a probabilistic clustering algorithm that models the data points as a mixture of Gaussian distributions. It allows for more flexible clustering than K-Means as it can handle different cluster shapes and can assign probabilities of membership to each point.

C. Modifying the distance function

1) Mahalanobis distance: Mahalanobis distance is a metric that takes into account the covariance structure of the data. It measures the distance between two points while accounting for the correlation between the variables. One advantage of using Mahalanobis distance is that it is robust to the scaling and correlation of the variables.

2) Chebyshev distance: Chebyshev distance is the maximum absolute difference between the coordinates of two points. It is particularly useful when dealing with data that has a large number of irrelevant features or when the features have very different scales.

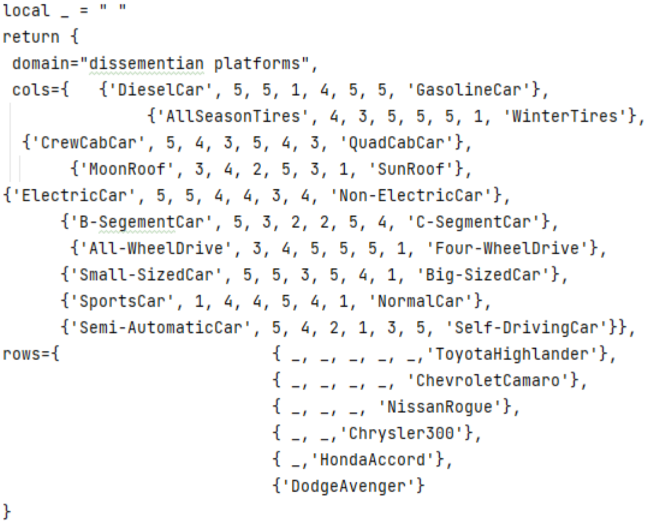
VIII. Requirements Study

To complete bonus activity 1, 5 people were asked to complete repertory grids regarding their evaluation of car brands and attributes. Their perspectives were evaluated against the data provided to identify any gaps in our modeling techniques. The following summarizes the report yielded by the repgrids. The survey was conducted with 6 cars on 10 different attributes. Clustering process resulted in the following inferencees. HondaAccord and NissanRogue were clustered together, but it could fall into different classification categotries. But the consistent clustering of Chrysler300 and DodgeAvenger is apparent as they are Diesel cars falling under common category. ToyotaHighlander being the only Big Sized Car, resulted in it always being clustered by itself, so this is as expected.But the behaviour of ChevroletCamaro and NissanRogue not being clustered together, with ChevroletCamaro consistently being clustered by itself. Since both of these cars are similar in the sense that they belong to the category of Winter Tires, but NissanRogue was clustered with HondaAccord. The cause for this would appear to be the scoring of attributes outside of the genre. Attributes such as whether the car is semi-automatic or self-driving resulted in NissanRogue and HondaAccord being clustered together instead of NissanRogue and ChevroletCamaro. The other major reason we may have seen these results is because we may simply have just chosen people that have different perspectives. The results would have yielded differently with a different set of people

As for the attribute synonyms, we got more varying results between people. The only consistent result we had was that the attribute sets AllSeasonTires:WinterTires and All-WheelDrive:Four-WheelDrive resulted in them being synonyms. Besides the other attributes in synonym with the other is B-SegementCar:C-SegmentCar and CewCabCar:QuadCabCar which is expected. In conclusion, these were the results obtained from the trials conducted.

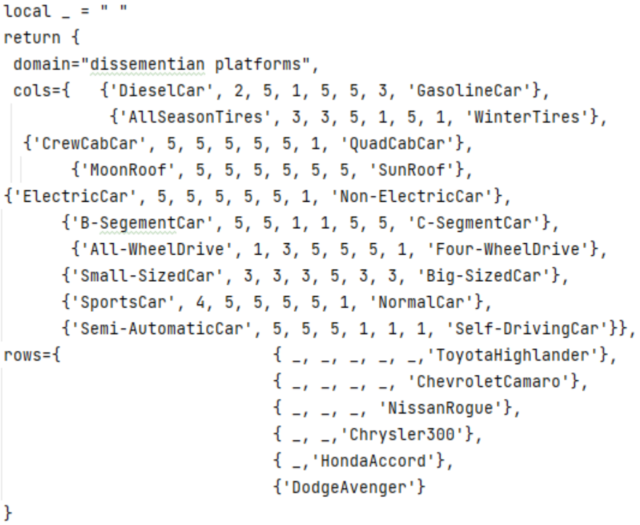
The following depicts the 5 repgrids surveyed and inputted to test:

Case 1



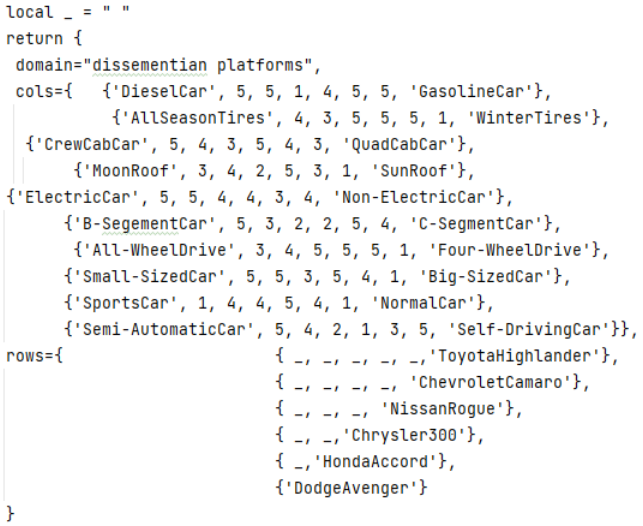
```
local _ = " "
return {
 domain="dissementian platforms",
 cols={   {'DieselCar', 5, 5, 1, 4, 5, 5, 'GasolineCar'},
               {'AllSeasonTires', 4, 3, 5, 5, 5, 1, 'WinterTires'},
  {'CrewCabCar', 5, 4, 3, 5, 4, 3, 'QuadCabCar'},
       {'MoonRoof', 3, 4, 2, 5, 3, 1, 'SunRoof'},
{'ElectricCar', 5, 5, 4, 4, 3, 4, 'Non-ElectricCar'},
      {'B-SegementCar', 5, 3, 2, 2, 5, 4, 'C-SegmentCar'},
       {'All-WheelDrive', 3, 4, 5, 5, 5, 1, 'Four-WheelDrive'},
      {'Small-SizedCar', 5, 5, 3, 5, 4, 1, 'Big-SizedCar'},
      {'SportsCar', 1, 4, 4, 5, 4, 1, 'NormalCar'},
      {'Semi-AutomaticCar', 5, 4, 2, 1, 3, 5, 'Self-DrivingCar'}},
rows={                      { _, _, _, _, _,'ToyotaHighlander'},
                            { _, _, _, _, 'ChevroletCamaro'},
                            { _, _, _, 'NissanRogue'},
                            { _, _,'Chrysler300'},
                            { _,'HondaAccord'},
                            {'DodgeAvenger'}
}
```

Case 2



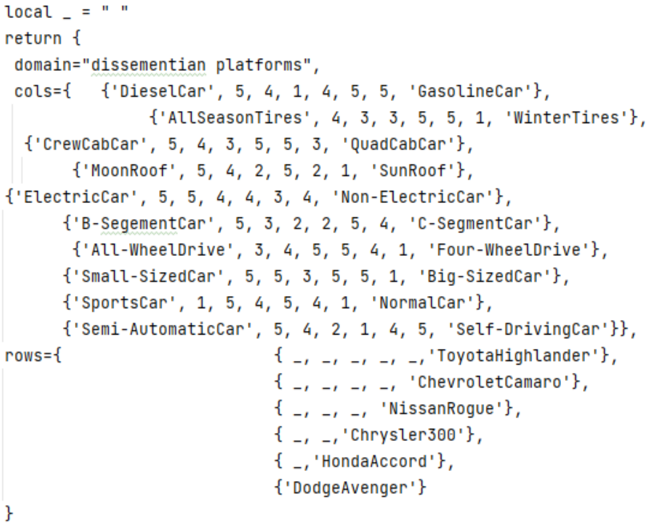
```
local _ = " "
return {
 domain="dissementian platforms",
 cols={   {'DieselCar', 2, 5, 1, 5, 5, 3, 'GasolineCar'},
               {'AllSeasonTires', 3, 3, 5, 1, 5, 1, 'WinterTires'},
  {'CrewCabCar', 5, 5, 5, 5, 5, 1, 'QuadCabCar'},
       {'MoonRoof', 5, 5, 5, 5, 5, 5, 'SunRoof'},
{'ElectricCar', 5, 5, 5, 5, 5, 1, 'Non-ElectricCar'},
      {'B-SegementCar', 5, 5, 1, 1, 5, 5, 'C-SegmentCar'},
       {'All-WheelDrive', 1, 3, 5, 5, 5, 1, 'Four-WheelDrive'},
      {'Small-SizedCar', 3, 3, 3, 5, 3, 3, 'Big-SizedCar'},
      {'SportsCar', 4, 5, 5, 5, 5, 1, 'NormalCar'},
      {'Semi-AutomaticCar', 5, 5, 5, 1, 1, 1, 'Self-DrivingCar'}},
rows={                      { _, _, _, _, _,'ToyotaHighlander'},
                            { _, _, _, _, 'ChevroletCamaro'},
                            { _, _, _, 'NissanRogue'},
                            { _, _,'Chrysler300'},
                            { _,'HondaAccord'},
                            {'DodgeAvenger'}
}
```

Case 3



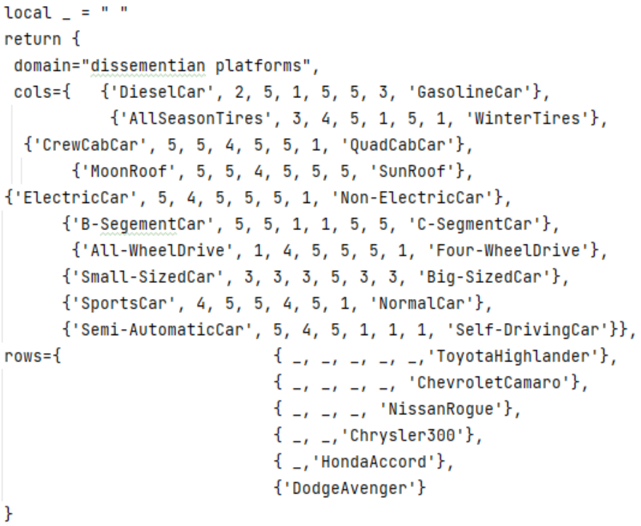
```
local _ = " "
return {
 domain="dissementian platforms",
 cols={   {'DieselCar', 5, 5, 1, 4, 5, 5, 'GasolineCar'},
           {'AllSeasonTires', 4, 3, 5, 5, 5, 1, 'WinterTires'},
  {'CrewCabCar', 5, 4, 3, 5, 4, 3, 'QuadCabCar'},
       {'MoonRoof', 3, 4, 2, 5, 3, 1, 'SunRoof'},
{'ElectricCar', 5, 5, 4, 4, 3, 4, 'Non-ElectricCar'},
      {'B-SegementCar', 5, 3, 2, 2, 5, 4, 'C-SegmentCar'},
       {'All-WheelDrive', 3, 4, 5, 5, 5, 1, 'Four-WheelDrive'},
      {'Small-SizedCar', 5, 5, 3, 5, 4, 1, 'Big-SizedCar'},
      {'SportsCar', 1, 4, 4, 5, 4, 1, 'NormalCar'},
      {'Semi-AutomaticCar', 5, 4, 2, 1, 3, 5, 'Self-DrivingCar'}},
rows={                        { _, _, _, _, _,'ToyotaHighlander'},
                              { _, _, _, _, 'ChevroletCamaro'},
                              { _, _, _, 'NissanRogue'},
                              { _, _,'Chrysler300'},
                              { _,'HondaAccord'},
                              {'DodgeAvenger'}
}
```

Case 4



```
local _ = " "
return {
 domain="dissementian platforms",
 cols={   {'DieselCar', 5, 4, 1, 4, 5, 5, 'GasolineCar'},
               {'AllSeasonTires', 4, 3, 3, 5, 5, 1, 'WinterTires'},
  {'CrewCabCar', 5, 4, 3, 5, 5, 3, 'QuadCabCar'},
       {'MoonRoof', 5, 4, 2, 5, 2, 1, 'SunRoof'},
{'ElectricCar', 5, 5, 4, 4, 3, 4, 'Non-ElectricCar'},
      {'B-SegementCar', 5, 3, 2, 2, 5, 4, 'C-SegmentCar'},
       {'All-WheelDrive', 3, 4, 5, 5, 4, 1, 'Four-WheelDrive'},
      {'Small-SizedCar', 5, 5, 3, 5, 5, 1, 'Big-SizedCar'},
      {'SportsCar', 1, 5, 4, 5, 4, 1, 'NormalCar'},
      {'Semi-AutomaticCar', 5, 4, 2, 1, 4, 5, 'Self-DrivingCar'}},
rows={                        { _, _, _, _, _,'ToyotaHighlander'},
                              { _, _, _, _, 'ChevroletCamaro'},
                              { _, _, _, 'NissanRogue'},
                              { _, _,'Chrysler300'},
                              { _,'HondaAccord'},
                              {'DodgeAvenger'}
}
```

Case 5



```
local _ = " "
return {
 domain="dissementian platforms",
 cols={   {'DieselCar', 2, 5, 1, 5, 5, 3, 'GasolineCar'},
           {'AllSeasonTires', 3, 4, 5, 1, 5, 1, 'WinterTires'},
  {'CrewCabCar', 5, 5, 4, 5, 5, 1, 'QuadCabCar'},
       {'MoonRoof', 5, 5, 4, 5, 5, 5, 'SunRoof'},
{'ElectricCar', 5, 4, 5, 5, 5, 1, 'Non-ElectricCar'},
      {'B-SegementCar', 5, 5, 1, 1, 5, 5, 'C-SegmentCar'},
       {'All-WheelDrive', 1, 4, 5, 5, 5, 1, 'Four-WheelDrive'},
      {'Small-SizedCar', 3, 3, 3, 5, 3, 3, 'Big-SizedCar'},
      {'SportsCar', 4, 5, 5, 4, 5, 1, 'NormalCar'},
      {'Semi-AutomaticCar', 5, 4, 5, 1, 1, 1, 'Self-DrivingCar'}},
rows={                        { _, _, _, _, _,'ToyotaHighlander'},
                              { _, _, _, _, 'ChevroletCamaro'},
                              { _, _, _, 'NissanRogue'},
                              { _, _,'Chrysler300'},
                              { _,'HondaAccord'},
                              {'DodgeAvenger'}
}
```

References

[1] Multi-objective semi-supervised clustering to identify health service patterns for injured patients

[2] A fast and recursive algorithm for clustering large datasets with -medians

[3] A Probabilistic Framework for Semi-Supervised Clustering

[4] Optimization Techniques for Semi-Supervised Support Vector Machines

[5] Semi-supervised Learning with Deep Generative Models

[6] Indicator-Based Selection in Multiobjective Search

[7] A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II

[8] An Evolutionary Algorithm for Multiob jective Optimization The Strength Pareto Approach

[9] t-SNE