Hierarchical K-Prototypes Based Semi-Supervised Explanation System

Surya Prakash Susarla
Computer Science
NCSU
Unity: spsusarl

Sushanth Reddy Chilla
Electrical Engineering
NCSU
Unity: schilla

Kaushik Pillalamarri
Electrical Engineering
NCSU
Unity:spillal2

***Abstract*—Semi-supervised explanation systems are useful in many domains, including finance, healthcare, and education. They help users understand the reasoning behind the predictions made by machine learning models. In this paper, we attempt to leverage research on clustering of mixed data in large datasets in combination with custom metrics for selecting rules to maximize the rule learning capability of the system. Our system utilizes the clustering technique, K-Prototypes, to group data instances based on both numerical and categorical data, and then creates a hierarchical structure of these groups. We also introduce a semi-supervised approach, where user feedback is used to improve the quality of the explanations provided by the system. We demonstrate the effectiveness of our approach on several datasets, and show that our system can provide meaningful explanations for both numerical and categorical data.**

I. Introduction

A multi-objective semi-supervised explanation system is a type of machine learning system that is designed to provide explanations for the outputs or decisions it makes. The general process for this involves two steps, the first is the dataset pruning and the second is rule generation from the pruned data. The main idea for pruning data is to find the best possible bin which will be used for rule generation, and the general implementation involves clustering the data and extracting the best possible subset of the data. Some of the classical implementations for Clustering involves K-Means [1], Spectral Clustering. Another implementation we explored takes some random samples from the data and then recursively separates them into two clusters until an endpoint is reached. The second step involves taking the best bin and generating a rule from it.

We encountered some problems in the classical methods, in case of K-Means, it was not able handle non-numeric data, and spectral clustering was computationally very expensive on huge data like [14] as it needed to compute various metrics like similarity matrix, and then compute eigenvalues and eigenvectors of this matrix. In contrast to this, the studied method compares the distances between the rows based on the face values and does not consider the broader statistics of the column to create a relative ordering between the values. Some of the methods also have a hard time trying to optimize data which has both numeric and symbolic type instead of having just one of either. To explore the issues we apply test driven development methodology to the provided datasets and try to answer some the research questions.

RQ1 : Is there a way to deal with datasets which have more than one type of data present?
We believe using a mix of algorithms which can deal with the data types individually but can be combined using a relevant scoring function instead of just using a single technique to handle multi-type data seems to give better performance overall.

RQ2 : How much impact does a scoring function have on rule generation?
We observed that adding a penalty term which measures how much entropy is present in a certain cluster gives improved performance.

RQ3 : Does the correlation between dependent variables explain the final generated rule?
We are able to explain some inconsistencies with regards to the final rule generated where the optimization is not in the direction we desire.

Upon observing the given datasets, we can observe that all the dependent columns are equally weighted. This causes the current algorithms to prioritise optimisation of all columns equally. By observing the correlation matrices it can be seen that some of the columns which are to be optimized in opposite directions have very strong positive correlation which implies that by maximising or minimising one of the columns the other column is affected adversely. This phenomenon is currently unaccounted for and creates a trade-off that is not directly governed by the algorithm. This problem could have been alleviated if the columns to be optimized had an associated priority which would help the algorithm to optimize higher priority variables under these circumstances.

The rest of this paper is structured as follows. Section 2 describes related work and Section 3 explains our experimental methods, and the datasets used, comparision study of the algorithms implemented and descriptions of

methods used to validate the observations. Section 4 shows the results observed and presents the summary of the studies performed and the observations made.

II. Related Work

Initial clustering methods involved partitioning the data to the nearest mean, and this process was thoroughly surveyed by Jin & Hang [1], where they provided a comprehensive review of the K-means algorithm. In the same paper they discussed the algorithm's sensitivity to outliers, the impact of distance metric and the appropriate number of clusters to be chosen. Dr. Menzies's method was able to reduce the impact of outliers by first sorting the whole data, and running its own version of clustering on only 90% of the actual by removing the first and last 5% by considering them as outliers. While the appropriate distance metric is more of a trial and error thing, the way the distance is assigned is an issue for symbolic data in K-means, as the mean of a symbolic data has no inherent value or meaning. To tackle this issue *Miguel and Wang* proposed a variant of the K-means[2]. They propose to use a Hamming Distance to calculate the distance between two data points instead of Sum of Squared Errors used in K-means, and instead of clustering to the nearest mean they propose to cluster to the nearest mode. The paper[3] by Zhexue Huang proposes extensions to K-Means called K-Prototypes that can handle datasets with mixed numerical and categorical attributes. The algorithm alternates between optimizing the numerical and categorical components of the distance measure, and the cluster center updates are modified accordingly. A sampling-based method for clustering large datasets is also proposed in the paper, which involves randomly selecting a subset of the data points and clustering them using either k-means or k-modes [2] depending on the data point. The resulting clusters are then used to further cluster the remaining data points using a modified version of the algorithm. Zengyou He, Xiaofei Xu, Shengchun Deng[4] proposed a method which involves creating multiple partitions of the dataset using different clustering algorithms and parameters, each of which is designed to handle either the numerical or categorical attributes.

All of these above discussed methods play a key role in the second part of the problem i.e., rule generation. The consideration given to outliers determines how effective the generated rules would perform when run against actual data. In addition to this, in Dr. Menzies's method, the comparison performed over the generated bins is based on the relative density of highly optimized rows when compared to poorly performing rows[16]. This helps the rule learner generate rules which determine the attributes required for selection of highly optimized rows but does not fully cover the cases where we can learn from the poorly performing rows to actively avoid them[17]. We perform such augmentation using the entropy function to prioritize bins that have a polar distribution of either the best or the worst rows in order to maximize the information gain in both directions.

Given that the problem is semi-supervised[18] and we only perform limited dependent variable look-ups, the impact of considering both the outliers as well as the information gain in both optimal and sub-optimal data points for the rule learner is required to ensure that the chosen algorithm maximizes the available information in the independent variables for better clustering and maximizing gain from the chosen clusters.

III. Methods

A. Algorithms

In the given version of algorithm we first perform the clustering using the method 'sway' whose purpose is to extract a predefined number of optimal data points and a subset of the remaining sub-optimal data points. This is done by first ordering the data using a distance metric which assigns scores to each data point via their impact on the dependent variables. After ordering the data we choose one random point and then pick another random point which is located at a distance of $95\%(tunableviahyper - parameters)$ to be considered as a representative of the opposite end of the data. Once this is done we assign the remaining data points to either one of the clusters using the cosine distance as the similarity measure. Finally we perform a comparison using the cluster anchor points and repeat the process recursively on the best cluster until we reach the threshold of the minimum rows desired.

After the above process is done we use the best rows determined from the above process to create bins in each of the independent variable columns which have a score assigned based on the relative domination of the optimal rows. This information is then used to order the flattened list of all the bins which constitute the rules learned by the algorithm.

In the proposed mechanism, we first modify the clustering method by replacing the above mentioned method with the K-Prototypes algorithm which uses the K-Means and K-Modes algorithms to improve the clustering of the data. The process follows a similar manner of splitting the data into clusters at every level and repeatedly applying the algorithm to generate the best cluster. We finally select the best cluster after reaching the threshold and a sample among all the remaining data points. This helps us evaluate the proposed improvements of using mean for numeric data and mode for symbolic data as a part of the K-Prototypes algorithm which can lead to tighter clusters.

Once the above process is done, we use the generated rows to perform rule generation with the augmented scoring function for the bins which uses the newly added entropy factor to reward the domination of either the best performing or the worst performing rows in the given bin. This is expected to help us with the problem mentioned earlier regarding the learning potential from sub-optimal data points.

B. Data

There are 11 datasets given in total, and in this section we will try to give a brief overview of each dataset based on their domain. For all the datasets, if the target column name ends with a '+' it implies we are trying to maximize the column and it it ends with a '-' it implies we are trying to

width=0.7[htbp]

	CityMPG+	HighwayMPG+	Weight-	Class-
CityMPG+	1.000000	0.945271	-0.835735	-0.621572
HighwayMPG+	0.945271	1.000000	-0.776756	-0.625925
Weight-	-0.835735	-0.776756	1.000000	0.737489
Class-	-0.621572	-0.625925	0.737489	1.000000

TABLE I
CORRELATION MATRIX OF THE DEPENDENT VARIABLES OF AUTOSET2.CSV

	Kloc+	Effort-	Defects-	Months-
Kloc+	1.000000	0.594941	0.963195	0.896749
Effort-	0.594941	1.000000	0.578790	0.741530
Defects-	0.963195	0.578790	1.000000	0.871345
Months-	0.896749	0.741530	0.871345	1.000000

TABLE II
CORRELATION MATRIX OF THE DEPENDENT VARIABLES OF NASA93DEM.CSV

minimize the column. If the column ends with 'X' it means we drop the column.

1) Car Design: The datasets in this domain contain information related to the fuel consumption of various cars. There are two datasets in this domain. One is *autoset2.csv*[5] which has 23 columns in total, out of which four of them are our target variables, which are 'CityMPG+', 'HighwayMPG+', 'Weight-', 'Class-'. The correlation between the four target variables can be seen in (table 1) is the perfect example for a dataset for which we can say that the algorithm will perform well on the data as the attributes that need to be maximized have positive correlation with each other while having negative correlation with the ones that need to be minimized.
The other one is *autoset93.csv*[6] which has 8 columns in total out of which three of them are our target variables, which are 'Lbs-', 'Acc+', 'Mpg+'. Even this has a similar correlation matrix as autoset2.csv implying potential for good performance.

2) Software Project Estimation: There are three datasets in this domain,
In *china.csv*[7] there is only one target column 'Neffort-'. The *coc1000.csv*[8] has 21 columns out of which there are 5 target columns 'LOC+', 'AEXP-', 'PLEX-', 'RISK-', 'EFFORT-, and while analysing the dataset we observed that 'AEXP-', 'PLEX-', 'RISK-' are not correlated to any other columns, but 'LOC+' and 'EFFORT-' are positively correlated, this might effect the performance of the algorithm and we might get a sub-optimal solution as both of the columns can not be optimized at the same time in opposite directions.
The *coc10000.csv*[9] has 25 columns out of which there are 3 target columns, 'Loc+', 'Risk-', 'Effort-', and the same story repeats with 'Loc+', 'Effort-' as observed in case of the previous dataset.
3) Software effort+detects estimation: The dataset in this domain is *nasa93dem.csv*[10] which has 26 columns out of which there are 4 target columns 'Kloc+', 'Effort-', 'Defects-', 'Months-, and its correlation matrix can be seen in (table 2) is a case where the algorithm will result in a sub-optimal rule generation, as the target variable which has to be maximized is highly positively correlated with the target variables which need to be minimized.

4) Issue Close Time: : One dataset in this domain is *healthCloseIsses12mths0001-hard.csv*[11] which has 8 columns out of which 3 are target variables, 'MRE-', 'ACC+', 'PRED40+', and the correlations between the target variables are in the desired directions. The other dataset is *healthCloseIsses12mths0001-easy.csv*[12] which is similar to the above dataset in terms of the column names and the correlation nature observed.

5) Agile Project Management: The dataset *pom.csv*[13] has three target columns 'Cost-', 'Completion+', 'Idle-', which can be translated to lower the idle rate and improve the completion rate while also decreasing the cost, but looking at the correlation matrix In this scenario we again observe from the data that one of the columns can be optimized where as the other two have a positive correlation although they move in opposite directions.

6) Computational Physics: One dataset in this domain is *SSM.csv*[14] which has 15 columns out of which 2 are target variables 'NUMBERITERATIONS-', 'TIMETOSOLUTION-', both need to be minimized and both are highly positively correlated which makes it easy for the algorithm to optimize. The other dataset is *SSN.csv*[15] which has 19 columns out of which 2 are target variables 'PSNR-', 'Energy-'. While both of them need to be minimized their correlation matrix can be shown below,

	PSNR-	Energy-
PSNR-	1.000000	0.045113
Energy-	0.045113	1.000000

TABLE III
CORRELATION MATRIX OF SSN.CSV TARGET VARIABLES

So the optimal rule generated depends purely on randomness as the optimal rule for one might not be the same for other since both of them are almost uncorrelated.

C. Performance Measures

1) auto93.csv Dataset: Improved sway produced better results than original sway for all the three attributes $'ACC+'$,$'Lbs-'$ and $'MPG+'$. The improved xpln resulted in better results compared to the 'ALL' results, this can be seen in the T-test significance values of Sway and Xpln.The confidence of some of the attributes which were seen to be close to 0.05 can also be used to describe how our model performed on this dataset.

2) auto2.csv Dataset: In this dataset we had to minimize 'Class-', 'Weight-' and maximize 'CityMpG', 'HighwayMPG'. Even though the data set had a positive correlation between same polar objectives and a negative correlation between opposite polar objectives, the model didn't work so well.

3) coc10000.csv Dataset: As explained in the above section the model achieved a sub-optimal measure because of the positive correlation between maximizing and minimizing objectives.Sway improved and Xpln improved achieved not so great T-test results. The confidence Test achieved good results for 'Risk-' and 'Effort-' for xpln improved. 'Effort' achieve desired results for confidence level on sway improved.

4) closelsses12mths0011-easy.csv: In this dataset we are supposed to minimize 'MRE-' and maximiz 'ACC+','PRED40+'. The T-test confidence values for sway imporved are produced very cloose to zero. Xpln Improved performed almost similar to original xpln, the T-test for significance of Xpln resulted in cloose to zero values for all three attributes. The ztop values are also found to be similar with the xpln and xpln improved values.

5) coc1000.csv Dataset: In this dataset we are supposed to minimize 'AEXP-','EFFORT-','PLEX-','RISK-' and maximize 'LOC+'. The model doesnt seem to work so well on this dataset. Here The performance is similar to 'ALL'. T-test for significance resulted in cloose to zero values. The results of xpln improved were slightly better than original xpln method.

6) pom.csv Dataset: In this dataset we are supposed to minimize 'Cost-', 'Idle-' and maximize 'Completion+'. T-test for confidence on sway resulted in ¡0.05 value for Cost on the other hand it achieved ¿0.1 value on the other two datasets.T-test for significance on sway resulted in suboptimal values.

7) China.csv Dataset: In this dataset we are supposed to optimize one objective i.e, 'N effort-'. T-test significance value for Sway Improved is very good for this dataset, the confidence value is also less than 0.05 for sway. T-test significance value for xpln improved for this data sewt is also negative, where as the confidence seems to a little higher at 0.1. Overall the model performed well on this dataset.

8) SSM.csv Dataset: In this dataset we are supposed to optimize 'NUMBERITERATIONS' and 'TIMETOSOLUTION-'. Both Sway Improved" and "Xpln Improved " failed to perform better than the original methods.Both sway and Xpln achieved ¿5 values on significance test which is not desirable.

9) nasa93dem.csv Dataset: In this dataset we are supposed to minimize 'Effort-', 'Defects-', 'Months-' and maximize 'Kloc+'. Sway improved was able to successfully minimize all the objectives that are supposed to be minimized compared to original sway method, but it failed to maximize 'Kloc+'.Same was the case with Xpln improved, It optimized 'Effort-', 'Defects-', 'Months-' very well but failed in optimizing 'Kloc+'. The confidence for both sway and Xpln was ¿0.2 fall each of the objectives. This behaviour can also be explained using the positive correlation between all of the objectives that are required to be optimized.

10) healthCloseIsses12mths0001-hard.csv Dataset: In this dataset we are supposed to minimize 'MRE-' and maximize 'ACC+', 'PRED40+'. The model tried to optimize 'PRED40+' and because of negative correlation with MRE and ACC, it minimized the both. The significance test for PRED40 and MRE seemed good whereas it wasnt that great on ACC. The confidence test was suboptimal for each of the objectives on both sway and xpln.

D. Summarization methods

Non-parametric effect size measures are used to estimate the magnitude of the effect of an independent variable on a dependent variable without relying on the assumption of normality or equal variances.
Non-parametric significance tests, on the other hand, are used to determine if there is a significant difference between two data groups without relying on the assumption of normality. Examples of non-parametric significance tests include the Mann-Whitney U test, the Wilcoxon signed-rank test, and the Kruskal-Wallis test.
T-tests, on the other hand, are parametric tests used to determine if there is a significant difference between two groups of data, assuming that the data is normally distributed and that the variances of the two groups are equal.
Non-parametric effect size measures and non-parametric significance tests are non-parametric statistical tools that do not rely on normality or equal variances assumptions. They are helpful when the data does not meet the assumptions of a parametric test, such as a t-test. On the other hand, T-tests are parametric tests that assume normality and equal variances and are powerful tools for detecting differences between groups when these assumptions are met.

IV. Results

A. Effect of sampling budget

In the proposed methods the sampling budget is not directly controlled and is impacted tertiarily through the effect of minimum cluster size. As the cluster size is reduced the algorithm attempts to dive deeper into the data to identify better data which requires higher dependent variable lookups i.e., greater budget is required. This can be controlled in the code via the minimum cluster size hyper-parameter (K-MIN). The samples are selected by exponentiation of the total number of rows by the chosen minimum cluster size. Upon experimentation we observed that smaller K MIN values which lead to greater y value lookups performed better when compared to larger K MIN values. We can observe the effect of this through the average performance of sway over the dataset[11]. Below are the means results over 20 runs with different sampling budgets for the dataset:

	MRE-	**ACC+**	**PRED40+**
SWAY$^{(0.7)}$	76.12	6.8	0.0
SWAY$^{(0.3)}$	73.52	7.6	25.0

TABLE IV
BUDGET AND PRUDENCE STUDY

As observed above a greater sampling budget helped generate better results with improved performance observed over all columns. Similar results were observed with greater variance for smaller increments in the sampling budget with occasional observations of one of the objects being optimized with trade-offs in other objectives.

B. Prudence Study

As observed in the above section, the performance of the algorithm when run on larger sampling budget is better when compared to the runs performed on smaller budgets. This has been measured through the changes in the minimum cluster size which increases the sampling sizes. It has to be noted that in some cases the results were inconclusive due to the random nature of selection and sampling among the clusters for rule generation. In some cases, due to the nature of selection one of the objective was optimized much better at the expense of other objectives leading to an overall reduction in performance.

C. Observations for research questions

For the research questions used as baseline for this study we found the following observations: **RQ1 : Is there a way to deal with datasets which have more than one type of data present**

RA1 : We have observed from our attempts that algorithms such as K-Prototypes used in our scenario have the capacity to perform better even with a mixture of numeric and symbolic data which was the one of our main points of exploration. The changes here performed very well in some of the datasets to produce very effective best columns whereas it was also observed in some cases that the objectives selected were extremely biased based on the initial selection and the correlation among the columns.

RQ2 : How much impact does a scoring function have on rule generation?

RA2 : Again, we could observe that in some cases the rules generated had good performance when compared to the default rules. In general through the impact of the scoring function was almost similar to the default function and the additional entropy measure caused the rules to be suboptimally selected in some cases.

RQ3 : Does the correlation between dependent variables explain the final generated rule.

RA3 : Yes, in most cases our observations were coherent with the correlation patterns observed in the data. This also emphasized one of the limiations of the problem to have weights assigned to the objective columns in cases where there are multiple conflicting optimization possibilities. Currently, this leads to random selection wherein the results can oscillate between optimizing one vs the other.

V. Discussion

A. Threats to validity

As per our observations, the oscillation of the algorithms in case of conflicting optimization objectives poses a threat to the overall process since this leads to random selection of rows between trials. Such selection causes the overall results to contain selections where the target objective might be extremely optimal in some cases at the cost of other objectives which leads to an overall negative score in the final comparison. Upon close observation the individual entries reveal that the selection was directed extremely narrowly leading to this phenomenon being exhibited. This effect also carries over to the subsequent aspects of rule generation where the generated rules maybe found performing sub-optimally.

B. Insights

One of the keen insights we had through the study was the impact of effective selection criteria and the effect of sampling on overall significance of the results. In some of the cases as mentioned earlier we observed that the oscillation of such selection had a significant impact on the final rule generation and also on the summarization methods which sometimes had low confidence to the high variance in the data in such scenarios.

C. Future Work

One of the possible areas where improvements can be made would be the inclusion of a measure to observe the number of optimization objectives being affected by each of the choices and to weigh this factor in when making selection between conflicting objectives with no natural weights as is the case here. This would certainly help improve scenarios where the quality of rules generated was negatively impacted by the correlation issues mentioned earlier.

VI. Conclusion

The method proposed seems to work very well on some of the datasets, while performing a bit poorly than the standard implemntation on the other datasets. The use of k-prototypes seems to handle the mix in datatypes very well. Using the correlation matrix we are able to give an acceptable explanation as to why a generated rule is optimal or not in case of most of the datasets. An important conclusion in terms of rule generation is that if the target variables were assigned some sort of priorities it would be helpful in more optimal rule generation.

References

[1] Xin Jin Jiawei Han , *K-Means Clustering https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8425*

[2] Miguel and Weiran Wang, *The K-Modes Algorithm for Clustering https://arxiv.org/pdf/1304.6478.pdf*

[3] Zhexue Huang , *Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values https://link.springer.com/article/10.1023/A:1009769707641*

[4] Zengyou He, Xiaofei Xu, Shengchun Deng, *Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach, https://arxiv.org/abs/cs/0509011*

[5] auto2.csv, *https://archive.ics.uci.edu/ml/datasets/auto+mpg*
[6] auto93.csv, *https://archive.ics.uci.edu/ml/datasets/auto+mpg*
[7] china.csv, *https://arxiv.org/pdf/1609.05563.pdfpage=5*
[8] coc1000.csv, *https://arxiv.org/pdf/1609.05563.pdfpage=5*
[9] coc10000.csv, *https://arxiv.org/pdf/1609.05563.pdfpage=5*
[10] nasa93dem.csv, *https://arxiv.org/pdf/1609.05563.pdfpage=5*
[11] healthCloseIsses12mths0001-hard.csv, *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.htmlsklearn.ensemble.ExtraTreesClassifier*
[12] healthCloseIsses12mths0011-easy.csv, *https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.htmlsklearn.ensemble.ExtraTreesClassifier*
[13] pom.csv, *see section 4.1.2 in https://arxiv.org/pdf/1608.07617.pdf*
[14] SSM.csv, *See "trimesh" in https://arxiv.org/pdf/1801.02175.pdfpage=2*
[15] SSN.csv, *See "trimesh" in https://arxiv.org/pdf/1801.02175.pdfpage=2*
[16] Data Modelling and Specific Rule Generation via Data Mining Techniques, *http://ecet.ecs.uni-ruse.bg/cst06/Docs/cp/SIII/IIIA.17.pdf*
[17] A Comparative Analysis of Association Rules Mining Algorithms, *https://citeseerx.ist.psu.edu/document?repid=rep1type=pdfdoi=37eda2498924195c2e8a5c39f91187812fa0a0b9*
[18] A Comparative Analysis of Association Rules Mining Algorithms, *https://citeseerx.ist.psu.edu/document?repid=rep1type=pdfdoi=37eda2498924195c2e8a5c39f91187812fa0a0b9*
[19] Semi-supervised learning, *Hady, M.F.A., Schwenker, F. (2013). Semi-supervised Learning. In: Bianchini, M., Maggini, M., Jain, L. (eds) Handbook on Neural Information Processing. Intelligent Systems Reference Library, vol 49. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-36657-4*$_7$