FAKULTÄT FÜR
INFORMATIK

DEPARTMENT OF KNOWLEDGE MANAGEMENT AND DISCOVERY

REPORT

EXTENSION OF DRESS$^+$

Authors:
ANSHU DAUR (223853)
CHINMAYA HEGDE (224037)
MEGHANA R. DESHMUKH (223727)
SHIVALIKA SUMAN (223942)

Advisor:
DR. TOMMY HIELSCHER

October 30, 2019

Contents

1 Introduction **3**

2 Related Works **4**

3 Methodology **5**
3.1 Dataset 5
3.2 Data Preprocessing 5
3.3 ML-NL constraints 6
3.4 Subspace Quality Score 6
3.5 Clustering 7
3.6 Implementation 7
3.7 Implementation Challenges 11

4 Results and Evaluation **11**

5 Execution Guidelines **17**

6 Conclusion **17**

7 Future Works **18**

8 Acknowledgement **18**

9 Project Timeline **19**

1 Introduction

Collection of data and analysis has been growing exponentially in the medical field and numerous advancements are evolving in the research and development of medical diagnosis. Data Mining techniques are being applied in many medical applications. Medical data mining has humongous potential of exploring medical data sets and are conducive in finding the hidden patterns in clinical diagnosis. The medical data sets are complex and so is its task of analyzing, which needs accurate measures and methodologies. Clustering, an unsupervised learning method is used to classify similar objects in to groups.

In this project, we have explored clustering algorithm which would best suit the cohort study dataset of "Discovery of Relevant Example-constrained Subspaces" (DRESS) [4]. DRESS+ considers the density of clusters and the distance between the cohort study instances to describe a target concept. It makes use of Density-based spatial clustering of applications with noise (DBSCAN).

DRESS+ employs derivation of Must-Link(ML) and Cannot-Link(NL) constraints from the labelled data which is taken care in Hierarchical Density-based spatial clustering of applications with noise (HDBSCAN)[1] algorithm employed in this project for clustering. It extends DBSCAN by converting it into a hierarchical clusters, and then uses a technique to extract a flat clustering based in the stability of clusters. Constraints are derived from pairwise instances which takes care of appropriate data partitioning. Pairwise instances belonging to same class form a set of ML constrains and pairwise instances belonging to different classes for a set of NL constraints, so such instances belonging to ML constraints are closer to each other and those satisfying NL constraints are well separated and far apart.

Our exploration goals for employing a clustering algorithm was to find algorithm that efficiently works with high dimensional data such as our cohort study dataset, a clustering algorithm that worked for arbitrary shaped clusters, reduce the input parameters of a clustering algorithm and to fix the varying epsilon values as we had in DBSCAN algorithm.

Project report is further divided in to sections. Upcoming section will be on similar works, where we discuss few algorithms we encountered that was close enough to our exploration goals. Section 3 gives the complete methodology of our work and in section 4, we discuss our results and also evaluate it with the earlier DBSCAN results and other statistics. We conclude in section 5 and the report winds up in section 6 and 7 of future works and acknowledgement respectively.

2 Related Works

Clustering methods play a vital role in predicting a patient with certain disease. Clustering helps in selecting right subspaces from the data set which are decisive for a certain disease.

DBSCAN in [4] uses two input parameters namely eps value and minpts. Eps is the radius of the cluster. In other words, it is a threshold value for two points in a cluster to become neighbours. Larger values as well as smaller values of eps has its own consequences in finding the appropriate clusters. Minpts are minimum number of neighbours a point should have to become a core point of a cluster. This parameter also impacts the clustering method.

Choosing a right setting is like a trial and error method. These two input parameters makes the clustering process complex since one may not find the right setting all the time. Making an algorithm self sufficient to implement these two parameters is a thing that drives us for this study. Automated eps and minpts by any means is a huge success for any clustering method, specially for the one working with large cohort study data sets. Below are some of the clustering methods which were closest to our exploration goals and which may have the potential to be incorporated with DRESS+ algorithm.

The paper by Xiaowei Xu et al. [9] focuses on reducing the parameter of the algorithm. Unlike DBSCAN, it has no input parameters. The efficiency however, is less when compared to DBSCAN. Another research from Sawant et al. [3] uses K dist plot before the density based DBSCAN. It finds the best eps settings explicitly that can be fed to the DBSCAN algorithm. It also satisfies our exploration goal of finding an algorithm for arbitrary shaped clusters but the reliability of this approach is unknown for high dimensional data.

Density Based Affinity Propagation by Yuan et al. [10] uses a two stage process similar to [3]. Clustering takes place multiple times according to the densities. A Fast Density and Grid Based Clustering Method for Data with Arbitrary Shapes and Noise [8] again uses density regions but are capable of clustering without the distance functions for arbitrary shaped clusters. Affinity Propagation used with Wisconsin Diagnostic Breast Cancer Dataset (WDBC) by [2] promises on computational speed and error rate but it works better with small number of features. High dimensional data usage is unknown.

3 Methodology

The Proposed algorithm is a combination of subspace and constraint based clustering. Subspace clustering is a technique to find subspaces from one or more dimensions.[5] Constraint based clustering[6] uses constraints to be satisfied by objects in the clusters of the outcome. a,b are in must link (ML) constraint if both a and b are in same cluster. If they are not, then they satisfy not link (NL) constraint.[4]. Apart from these two clustering methods we employ semi supervised quality score of the subspaces and 'Heterogeneous Euclidean Overlap Metric(HEOM) distance function. All together, we make sure that ML pairs are more similar and NL pairs are far apart.

3.1 Dataset

SHIP (Study of Health in Pomerania)[7] is a cohort study dataset based on epidemiological study of population. The assessments range from interviews to laboratory analyses, blood pressure measurements, dental, dermatological, cardiometabolic and various ultrasound examinations to more demanding methods such as cardiopulmonary exercise tests, sleep monitoring and whole-body magnetic resonance imaging. The SHIP data set has 4308 instances on 405 features. The data types of the features are int64, float64 and object. Of 4308 instances, 886 are labelled and 3730 are unlabelled. Out of labelled instances Class A has 694, 159 in Class B and 33 in Class C. Categorizing in to male and female, Male has 2116 and Female has 2192 instances. Out of 2116 male attribute 426 are labelled and 1690 are unlabelled. 460 were found to be labelled and 1732 as unlabelled for Female attribute.

3.2 Data Preprocessing

Data types of the features are int64, float64 and object. Initially, the data has been divided according to the data types. The irrelevant attributes like date and serial numbers were removed in this pre-processing section. As all the numerical data were widely ranged, those values were normalised between 0 to 1. This was done for both int64 and float 64. Only the numerical data were normalised in the data pre-processing step. The categorical features were left as it is, because the index of the categorical values had to be passed into HEOM distance function. But, before sending categorical values into heom function, it had to be normalized to be in sync with rest of the data. Missing values in the data set have been replaced with NaN values.

3.3 ML-NL constraints

HDBSCAN DRESS+ uses constraints to guide subspace clustering in semi-supervised way towards better feature space reduction. It uses instance based constraints on pairs of instances that belongs to same or different clusters to update constraint satisfaction within subspace. Must-Link constraints ensure that participants that belong to the same class should be near to each other and hence are part of the same cluster. For e.g. if x and y belong to same Class "A" then ML constraint :

$$con_{=}(x, y)$$

. Not-Link constraints ensure that participants that belong to different class should be far apart from each other and so lie in different clusters. For e.g. if u belong to class A and v belong to class B then NL constraint :

$$con_{\neq}(u, v)$$

MLsat and NLsat contains the list of pairwise instances that satisfy ML and NL constraints respectively.

3.4 Subspace Quality Score

Subspace quality and the corresponding scores were computed as per the paper by Hielscher et al. [4].

$$qcons(S) = |MLsat(S)| + |NLsat(S)|/|ML| + |NL| \tag{1}$$

$$qdist(S) = davg(S, NL) - davg(S, ML) \tag{2}$$

$$davg(S, NL) = \sum_{x,y} d(S, x, y)/|NL| \tag{3}$$

$$davg(S, ML) = \sum_{x,y} d(S, x, y)/|ML| \tag{4}$$

$$q(S) = qcons(S) * qdist(S) \tag{5}$$

3.5 Clustering

Our proposed algorithm, Hierarchical Density-based spatial clustering of applications with noise (HDBSCAN) replaces the DBSCAN algorithm implemented in DRESS+ by [4]. It transforms the space based on density. It uses single linkage clustering with the assumption of noise in the data. Handling noise is very important for the clustering method since a noise among denser regions can cause problems with the results. HDBSCAN works really well if we have the distance matrix of the points before we cluster them and this also takes care of the noise.
Kth nearest neighbor is a simple approach to make the distance matrix. This will run for all the instances and attributes. And eventually the noise points can be pushed far apart and only the denser regions can be concentrated. This matrix can be fed easily to the clustering algorithm. HDBSCAN now builds a minimum spanning tree to connect all the core points of dense regions and at the end we will have a hierarchy of all connected components of those regions. The DBSCAN would now cut the hierarchy horizontally by the specified intuitive parameter epsilon. But HDBSCAN now condenses the hierarchy and tries to find a minimum cluster size. This is the only parameter that we will have to feed in. Now, the hierarchy is split in to clusters where at each split we check the minimum cluster size. If there are less number of points to form a cluster for a particular point, the point falls out of the cluster. Once algorithm go through the whole hierarchy we will have the clusters which sustained for a long time and some with a very short span. We now extract the clusters by calculating its stability. This can be done by observing and saving the records of when the cluster was formed for a point and when it came to an end. Further, we can calculate the membership strength of a cluster by normalizing the value of Cluster formed and clustering ends.

3.6 Implementation

We modified the original DBSCAN algorithm in [4] with HDBSCAN[1]. Below are our algorithms and code snippet description from the implementation.

Data: Dataset D, original feature set F
Result: Set of subspace clusters C, set of candidate subspaces S, set of subspace quality values Q

Initialize empty set C;

For each f in F Do
Cf = HDBSCAN(Df); cluster;
C = (C Cf); // store initial clusters;
S = (S f); // store subspace candidate;
Q = (Q calcQuality(Df));store subspace quality ;

end For

Algorithm 1: DRESS+ candidate subspace initialization

We further have discussed some of the challenges we faced in execution and things to consider for future works. Python version 3 has been used for the development and we have explored Pandas, NumPy, Matplotlib, Scikit-learn and other important libraries of Machine learning during the implementation.

Python Libraries	
Dataframe	Pandas
Data Pre-processing	Preprocessing from sklearn
Clustering	HDBSCAN and DBSCAN from sklearn
Parallel Processing	Multiprocessor
K Nearest Neighbour and Naive Bayes	sklearn
Evaluation measures	sklearn
HEOM	Distython

Table 1: Python Libraries

Data: Dataset D, set of subspace clusters $\mathcal{C}$, set of candidate subspaces $\mathcal{S}$, set of subspace quality values Q
Result: Set of subspace clusters $\mathcal{C}$
$S_{candidate} \leftarrow pickBest(\mathcal{S}, Q)$; // pick best
$q_{best} \leftarrow q(S_{candidate})$;
// init. best quality value
$\mathcal{S} \leftarrow (\mathcal{S} \setminus \{S_{candidate}\})$;
// clean candidate subspaces
while $|\mathcal{S}| > 0$ **do**
  for *each* $S^* \in \mathcal{S}$ **do**
    $S_{new} \leftarrow (S^* \cup S_{candidate})$; // merge
    if $q_{dist}(S_{candidate}) > q_{dist}(S^*)$ **then**
      $S_{dist} \leftarrow S_{candidate}$;
    else $S_{dist} \leftarrow S^*$;
    $q_{dist}(S_{new}) \leftarrow calcDistQual(D_{S_{new}})$;
    if $q_{dist}(S_{new}) > q_{dist}(S_{dist})$ **then**
      // filter criterion
      $\mathcal{C}_{D_{S_{new}}} \leftarrow \text{HDBSCAN}(D_{S_{new}})$;
      $\mathcal{C} \leftarrow (\mathcal{C} \cup \mathcal{C}_{D_{S_{new}}})$;
      $Q \leftarrow (Q \cup calcQuality(D_{S_{new}}))$;
      if $q(S_{new}) > q_{best}$ **then**
        $q_{best} \leftarrow q(S_{new})$;
        $\mathcal{S} \leftarrow (\mathcal{S} \cup S_{new})$;
        $\mathcal{S} \leftarrow (\mathcal{S} \setminus \{S^*\})$; // clean
      end
    end
  end
  $S_{candidate} \leftarrow pickBest(\mathcal{S}, Q)$;
  $\mathcal{S} \leftarrow (\mathcal{S} \setminus \{S_{candidate}\})$; // clean
end

Algorithm 2: DRESS+ subspace processing and cluster generation

Code Snippets Description	
readConstraints()	Reads MLC and NLC constraints text file which contains pairwise instances satisfying ml and nl constraints. Returns dictionary with index id of instances as key and satisfying ML and NL constrained instances in an array. For example : 1:[10,2,8] 1 being the index of instance and 10,2 and 8 are the instances which form ml constraint pair with 1(as they belong to the same class). It also returns the count of ML and NL constraints
getQdist()	Function accepts dataframe and returns normalised values in a dataframe for both numerical and categorical attributes
normalise()	Function returns quality value for subspace cluster which is dependent only on pairwise distance similarity between ml and nl constraint pairs satisfying the current subspace
getQuality()	Calculates the quality of current subspace using ML and NL constraint. Function call expects: current subspace for which quality needs to be calculated, heom metric object instance, mlc, nlc constraints and constraints count . It returns the product of quality value(qcons) for satisfied ml and nl constraints which ignores the actual distance with quality value(qdist) which returns the average distance between ml and nl pairs. quality = qcons*qdist
generateConstraints()	Function returns text file containing pairwise ml and nl constraints in the current working directory

dressPlus()	Wrapper function which formats the output given by the CoreFunction in the required format.
getSubSpaceClusters()	Function returns the best quality aggregates from different threads for different subspaces
coreFunction()	Function returns subspace which calls HDBSCAN if the quality calculated for new subspace is higher than the candidate subspaces. If the quality is lower than the subspace is not added to candidate-subspace

Table 2: Function Description

3.7 Implementation Challenges

The HEOM library from Distython used in the algorithm, requires categorical features list along with the dataset to be passed. In our algorithm, we are using the distance function on subspaces which have the possibility of having no categorical data. So in the case where we pass a subspace and it has only numerical data then the categorical list should go as empty. This was not accepted by the HEOM library. In DBSCAN implementation, epsilon value should be calculated based on the 8(log—D—) NN distance plot and then finding the knee of the curve, where knee point would be the epsilon. But for the feature set, just having 0 and 1 value, knee point was coming 0, hence epsilon was 0 for that particular data set.

4 Results and Evaluation

For the 886 labelled instances of the data set, we ran Naive Bayes classifier on the subspaces retrieved via DBSCAN and HDBSCAN. Considering the class imbalance, K-fold cross validation has been used with k=10, and the average Accuracy, Precision and Recall were calculated. F measure: a measure for test's accuracy considering both Precision and Recall. It is a harmonic mean of both Precision and Recall. Precision is true positives divided by actual results or we could say out of all the examples the classifier labeled as positive, what fraction

were correct? and Recall is true positives divided by predicted results or out of all the positive examples there were, what fraction did the classifier pick up?

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

We implemented DBSCAN and HDBSCAN for labelled data and Accuracy, Precision, Recall, F measure and Kappa were recorded. Further, HDBSCAN and DBSCAN algorithms were run on the dataset which was divided with respect to gender:male and female. The SHIP dataset has certain features which are mostly related to only female. Because of this, after the division of the dataset based on gender, certain features having no relevance and valid data were removed from the male dataset. The removed features from the male dataset are: 'parity_w_s0', 'birth_w_s0', 'fruehgeb_w_s0', 'menopaus_w_s0', 'menopaus_yn_w_s0','hormonrepl_w_s0', 'pillever_w_s0', 'pillnow_w_s0', 'parity_w_s1', 'menopaus_w_s1', 'menopaus_yn_w_s1', 'hormonrepl_w_s1', parity_w_s2', 'fruehgeb_w_s2', 'hormonrepl_w_s2', 'menopaus_w_s2', 'menopaus_yn_w_s2', 'pillever_w_s2', 'pillnow_w_s2', 'premat_w_s0', 'stillbirth_w_s0', 'uterus_w_s0', 'hyst_w_s0', 'adnexe_w_s0', 'op_preg_w_s0', 'breast_w_s0', 'vag_chir_w_s0', 'steri_w_s0', 'menopause_w_s0', 'menostat_w_s0', 'use_mht_w_s0' .This process of removal, resulted in male dataset having lesser dimensions compared to that of the female dataset. The results on the dataset for the above measures for both DBSCAN and HDBSCAN, reflected the reduced dimensions of the male dataset. The results shown in this report for the male and female dataset are partial because of high runtime complexity of HDBSACN and lack of computational resources. Another set of dataset was processed for Female and Male aged 20 to 50, 51 to 81 data using DBSCAN and HDBSCAN and the results have been recorded. All the results are tabulated in the below table.

Algorithm, Dataset and constraints	Evaluation Measures	Subspaces
HDBSCAN on Labelled Data 886 instances MLC:253559 NLC:5247	Accuracy :94% Precision :96% Recall :94% f1-measure: 95% kappa :85%	'stea_s2', 'atc_c03c_s0', 'gluc_s_s2', 'mrt_liverfat_s2', 'goiter_s1', 'spindengen_s0', 'mrt_lower', 'mrt_mean', 'mrt_upper', 'tg_s_s1', 'sex_s0', 'tg_s_s0', 'stea_alt75_s0'
HDBSCAN on Female Data Aged between 20-50 1173 instances MLC: 37366 NLC: 310	Accuracy :23% Precision :49% Recall :23% f1-measure :13% kappa :0.10%	'mac_s0'
HDBSCAN on Female Data Aged between 51-81 1019 instances MLC: 37366 NLC: 140	Accuracy 78% Precision 62% Recall 78% f1-measure 69% kappa -2%	'il6_s0','blt_beg_s2'

DBSCAN on Female Data 2192 instances MLC:76121 NLC:885	Accuracy :92% Precision :94% Recall :92% f1-measure :92% kappa :79%	gluc_s_s2', 'mrt_mean', 'mrt_liverfat_s2', 'mrt_lower', 'fs_risk_s0', 'alcg7d_s0', 'smoking_s0', 'smoking_s2', 'il6_s0', 'tg_s_s0', 'tetai_s0', 'doc12mths_s0', 'op_preg_w_s0', 'pillever_w_s2', 'heartr_s0', 'breast_w_s0', 'vag_chir_w_s0', 'alkligt_s1', 'lipo_a_s0', 'packyrs_s0', 'som_bmi_s1', 'ncigd_s0', 'pcs_sf12_s0', 'apoa1_s0', 'steri_w_s0', 'menopause_w_s0', 'menostat_w_s0', 'use_mht_w_s0', 'gluc_s_s1', 'genintdoc12m_s0'
DBSCAN on male Dataset 2116 instances MLC: 52381 NLC: 1800	Accuracy :22% Precision :78% Recall :22% f1-measure :30% kappa: 4%	'alkligt_s1', 'tsh_s0', 'sd_volg_s0', 'atc_c08ca05_s2', 'ggt_s_s1', 'alcg30d_s0', 'atc_c07a_s2', 'atc_c09aa02_s2', 'mcs_sf12_s0', 'apob_s0', 'hyperlipid_s2', 'gout_s2', 'atc_c08ca01_s2', 'testo_m_s1', 'atc_c08da01_s2', 'stea_alt75_s2', 'pcs_sf12_s0', 'ffs_pattern_s0', 'atc_c09aa05_s2', 'atc_h02a_s2', 'stea_s2', 'w_sample_s0', 'chol_s_s1', 'atc_c08ca08_s2', 'chol_hdl_s0', 'atc_c08ca05_s1', 'hba1c_s0',
HDBSCAN on Female Data 2192 instances MLC:76121 NLC:885	Accuracy :94% Precision :95% Recall :94% f1-measure 94% kappa :83%	'mrt_liverfat_s2', 'atc_g04c_s0', 'mrt_lower', 'mrt_mean', 'mrt_upper', 'chol_s_s2', 'som_groe_s2', 'pillnow_w_s0', 'tg_s_s1', 'heartr_s0', 'igfbp3_s0','gluc_s_s2', 'physact_s0','fs_risk_s0', 'school_s0', 'vag_chir_w_s0','il6_s0','stenos_s0'

Table 3: Results and Evaluation

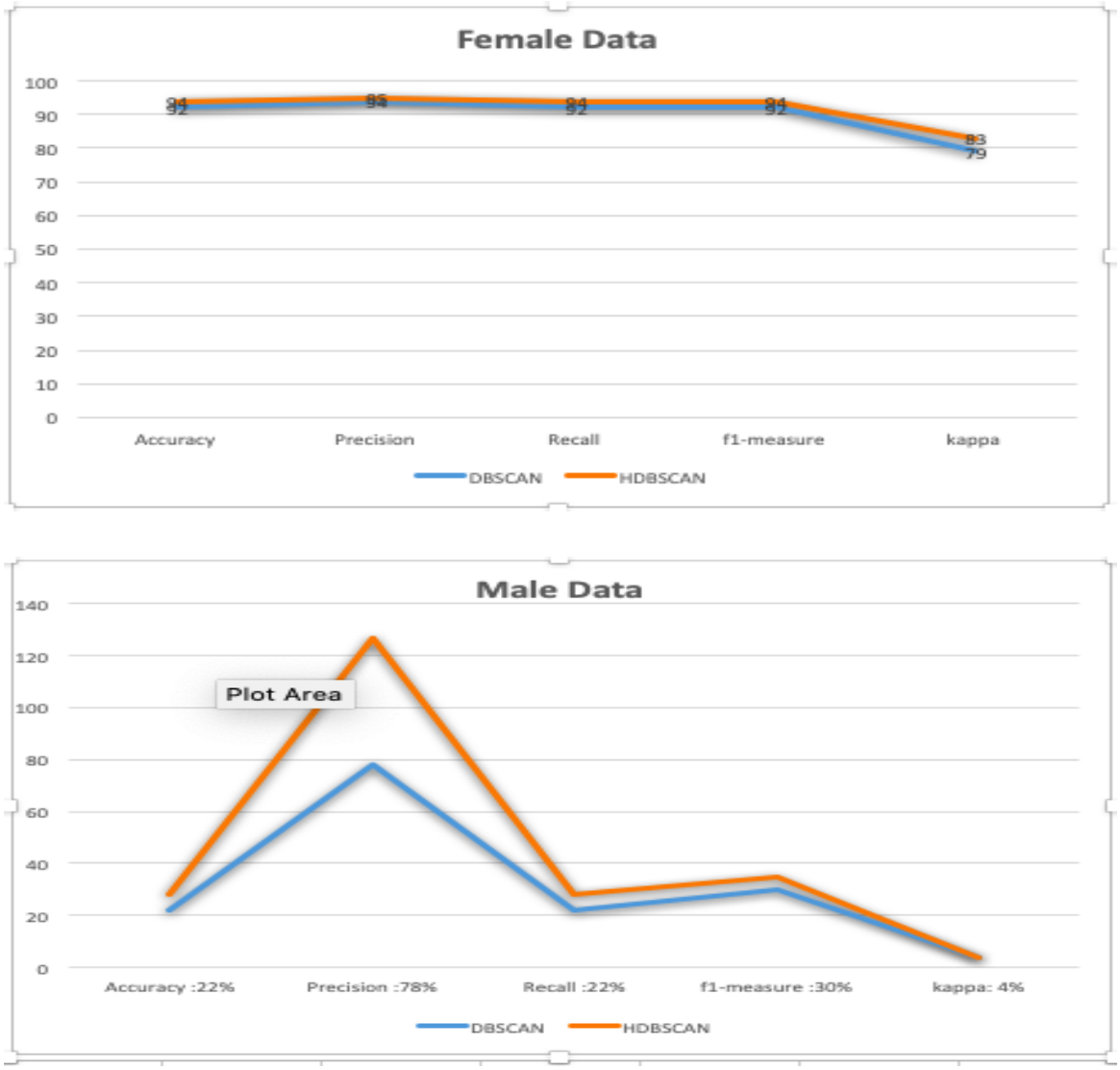

Figure 1: HDBSCAN vs DBSCAN on Male and Female Data

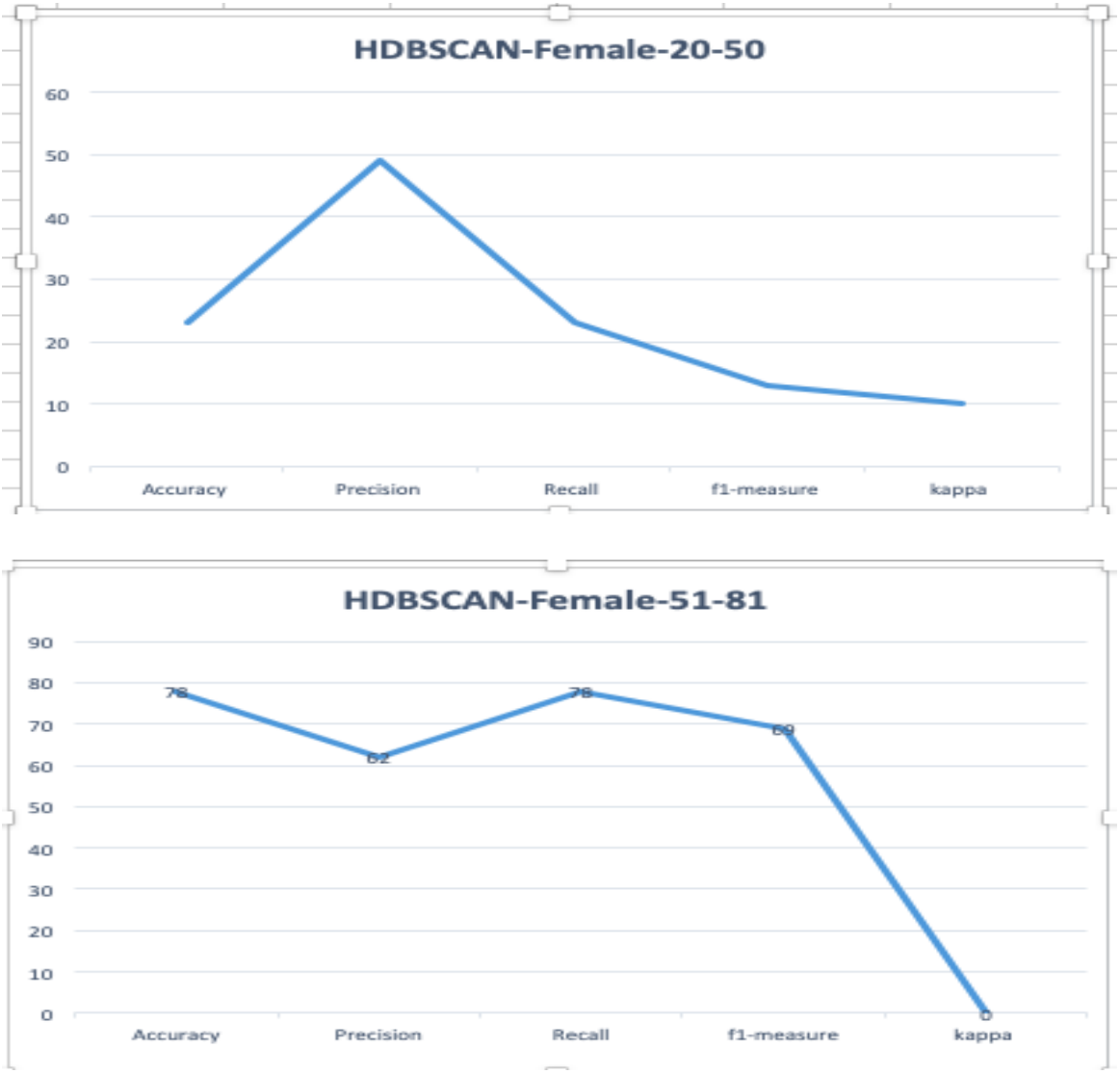

Figure 2: HDBSCAN on Sliced Female Data

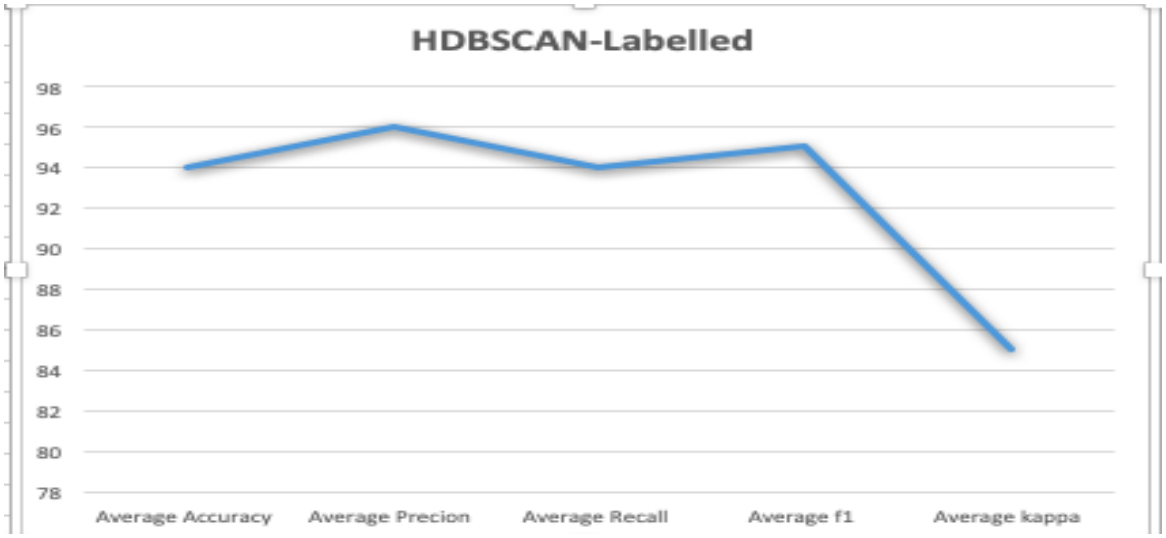

Figure 3: HDBSCAN on Labelled Data

5 Execution Guidelines

The entry point for execution is dress_final.py file. It is required to keep the file and dataset in the same folder. Once this file is executed, it will first generate mlc constraints file(mlc.txt) and nlc constraints file(nlc.txt) in the same folder which should not be deleted during the execution of the program.

Execute dress_final.py using python dress_final.py in Windows or python3 dress_final.py in Linux. Evaluation file executes Naive Bayes on the subspaces returned by algorithm.

6 Conclusion

HDBSCAN eliminates the epsilon parameter and is robust to varying epsilon values. It is also suitable for high dimensional data such as cohort study data. The comparison of HDBSCAN and DBSCAN shows the improvement in evaluation measures shown in female data set results in our project, but the run time complexity still remains debatable.

As mentioned, speed of the algorithm and time complexity were the major challenges faced. Both DBSCAN and HDBSCAN have the worst case time complexity of $O(n^2)$. HEOM has the time complexity of $O(n^2)$. Which also suggests that with the increase in number of instances and features, the time taken to execute increases quadratically. Though the time complexity of the algorithm cannot be changed, we noted the execution times of DBSCAN, HDB-

SCAN and HEOM to replace the most time consuming algorithm with a faster one. Following observation was made for the subspace having 18 features and 4300 instances: HDBSCAN - approx. 2028 seconds, DBSCAN: approx. 1902 seconds and HEOM: approx. 63 seconds. From the results obtained, replacing DBSCAN with HDBSCAN has increased the execution speed of the algorithm and is hence slower. It is also suspected that number of MLC and NLC constraints enforced also increased the time to cluster the subspaces as all of the constraints generated on the dataset were enforced during execution.

7 Future Works

Experiments can be done with various statistics of the data set considering both HDBSCAN and DBSCAN. Since it takes more than 10 days for each run, a proper experimental result goals can be set and results can be evaluated. Clustering can be visualized deeper and key insights can be derived. There is a scope to improve the computation time by optimizing the methods used and by converting the code to run in Graphical Processing Unit(GPU) environments. We have to delve in deep to understand the computation speed of methods used separately, like HEOM distance function and data frames vs NumPy. Implementing a better distance function in terms of quality and computation speed can be considered as further opportunities in the project.

8 Acknowledgement

We would like to express our deep gratitude and sincere thanks to M.Sc. Tommy Hielscher for his valuable guidance and support throughout this project. We also want to thank Prof. Dr. rer. nat. habil. Myra Spiliopoulou for giving us an opportunity to work on this project.

9 Project Timeline

References

[1] Ricardo J G B Campello and Computer Sciences. "Hierarchical Density Estimates for Data Clustering , Visualization , and Outlier Detection". In: 10.1 (2015), pp. 1–51.

[2] N Emami and A Pakzad. "A New Knowledge-based System for Diagnosis of Breast Cancer by a combination of Affinity Propagation Clustering and Firefly Algorithm". In: 7.1 (2019), pp. 59–68. DOI: 10.22044/JADM.2018.6489.1763.

[3] Manisha Naik Gaonkar and Kedar Sawant. "AutoEpsDBSCAN : DBSCAN with Eps Automatic for Large Dataset". In: 2 (2013), pp. 11–16.

[4] Tommy Hielscher et al. "A Framework for Expert-Driven Subpopulation Discovery and Evaluation Using Subspace Clustering for Epidemiological Data". In: (2019).

[5] Lance Parsons. “Subspace Clustering for High Dimensional Data : A Review”. In: 6.1 (), pp. 90–105.

[6] Carlos Ruiz, Myra Spiliopoulou, and Ernestina Menasalvas. “C-DBSCAN: Density-Based Clustering with Constraints”. In: *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*. Ed. by Aijun An et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 216–223. ISBN: 978-3-540-72530-5.

[7] Henry Vlzke et al. “Cohort Profile: The Study of Health in Pomerania”. In: *International Journal of Epidemiology* 40.2 (Feb. 2010), pp. 294–307. ISSN: 0300-5771. DOI: 10.1093/ije/dyp394. eprint: http://oup.prod.sis.lan/ije/article-pdf/40/2/294/2169177/dyp394.pdf. URL: https://doi.org/10.1093/ije/dyp394.

[8] Bo Wu et al. “Transactions on Industrial Informatics A Fast Density and Grid Based Clustering Method for Data with Arbitrary Shapes and Noise”. In: 3203.c (2016). DOI: 10.1109/TII.2016.2628747.

[9] Xiaowei Xu et al. “A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases D-80538 München 3 . A Notion of Clusters Based on the Distance Distribution”. In: ().

[10] Hanning Yuan et al. “DAPPFC : Density-Based Af fi nity Propagation for Parameter Free Clustering”. In: (2016), pp. 495–506. DOI: 10.1007/978-3-319-49586-6.