5-2019

Clustering heterogeneous autism spectrum disorder data.

Mariem Boujelbene
University of Louisville

CLUSTERING HETEROGENEOUS AUTISM SPECTRUM DISORDER DATA

By
Mariem Boujelbene
M.SC in Computer Science, 2019

A Thesis
Submitted to the Faculty of the
J.B Speed School of Engineeringof the University of
Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Masters of Science
in Computer Science

Department of Computer Engineering and Computer
Science
University of Louisville
Louisville, Kentucky

May 2019

CLUSTERING HETEROGENEOUS AUTISM SPECTRUM DISORDER DATA

By

Mariem Boujelbene
M.SC in Computer Science, 2019

Thesis approved on

April 26, 2019

by the following Thesis Committee:

Thesis Director
Dr. Olfa Nasraoui

Dr. Hichem Frigui

Dr. Gregory Barnes

DEDICATION

For all the People that supported me.

ACKNOWLEDGMENTS

I would like to thank Dr. Olfa Nasraoui for her guidance and support. I want to thank all the committee members for reviewing my thesis. I want also to thank Dr. Behnoush Abdollahi for her help and contribution during my work. I want to express my gratitude to my friends and family for believing in me

ABSTRACT

CLUSTERING HETEROGENEOUS AUTISM SPECTRUM DISORDER DATA

Mariem Boujelbene

April 26, 2019

Autism spectrum disorder (ASD) is a developmental disorder that affects communication and behavior. Several studies have been conducted in the past years to develop a better understanding of the disease and therefore a better diagnosis and a better treatment by analyzing diverse data sets consisting of behavioral surveys and tests, phenotype description, and brain imagery. However, data analysis is challenged by the diversity, complexity and heterogeneity of patient cases and by the need for integrating diverse data sets to reach a better understanding of ASD.

The aim of our study is to mine homogeneous groups of patients from a heterogeneous set of data consisting of both ADOS and Behavioral datasets and to interpret the discovered clusters within the medical context of the affected brain areas using fMRI data.

We developed an unsupervised machine learning pipeline to mine a heterogenous data set consisting of the Standardized Autism Diagnostic Observation Schedule (ADOS) scores, which are metrics used to measure the autism severity, phenotypical and behavioral data. This ADOS data is used to identify behavioral problems for autistic patients. We also used functional Magnetic Resonance Imaging (fMRI) which is a technique for measuring and mapping brain activity.

Our Big Data pipeline utilizes different clustering algorithms to partition the patients into homogeneous groups: hierarchical clustering, spectral clustering and spectral co-clustering.

In addition to clustering the data, we present a general framework that adds explainability to clustering algorithms in a way that assists the end-user in making sense of the clustering outputs through answering their questions about the results relative to the input data itself or relative to available external evidence.

Our clustering algorithms were able to discover homogeneous groups of patients that share similar behavioral and phenotypical characteristics. Furthermore, we generate an accessible interpretation of clustering results by mapping the discovered clusters onto the brain structure.

Through our clustering and explanation modules, our unsupervised machine learning methodology enables the domain experts to perform a powerful analysis on homogeneous cases, such as discovering hidden associations between the genetic data of patients belonging to the same cluster in order to have a better understanding of Autism Spectrum Disorder (ASD) and to pave the way toward data-driven personalized medicine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

According to the Center for Disease Control (CDC) [1], one in 51 children in the US is diagnosed with Autism Spectrum Disorder (ASD). This disease affects the behavior of children and it is hard to diagnose.

Machine Learning has proven in the last few years [2] to be a very helpful tool to doctors that helped them map genes to specific diseases or detect tumors efficiently. It uses mathematical models in order to detect hidden pattern in a set of heterogeneous data and provides estimations and predictions to achieve a predetermined task. Clustering is a subset of this field that focuses on unlabeled data. It detects clusters of data based on a predefined similarity measure. Clustering can help doctors identify similar groups of patients and hence detect common characteristics that can help identify, study, and understand better ASD.

In this thesis we will use clustering in an attempt to analyze and study ASD. Our motivations for using clustering for autism data are: (1) discovering clusters of similar patients makes it easier to discover significant genes associated with certain behavior phenotypes using genomic analysis; (2) clusters provide a principled methodology to divide patients into (pure) groups in clinical studies for personalized medicine and allow better association with other data modalities (brain structure, genotype, psychological test results, etc), hence accelerating scientific discovery.

Although clustering is a valuable machine learning tool, clustering algorithms can generate a significant amount of output results that need to be interpreted and judged by humans. The interpretation and understanding depends on the format of the clustering results and the expertise of the end users. These end users can have varying levels of expertise in machine learning, specifically clustering; in the application domain (e.g. autism); or both. The outputs of clustering algorithms can also vary in format. For this reason, we propose a general framework to build an explainability module for clustering algorithms that assists the end-user in making sense of the clustering outputs through answering the end user's generic questions about the clustering results.

Through our clustering and explanation modules, our unsupervised machine learning methodology enables the domain experts to perform a powerful analysis on homogeneous cases, such as discovering hidden associations between the genetic data of patients belonging to the same cluster in order to have a better understanding of Autism Spectrum Disorder (ASD) and to pave the way toward data-driven personalized medicine.

1 Objectives

Our research study pursues the following objectives:

- Design an unsupervised machine learning pipeline to perform clustering on different types of ASD related data.

- Design a general framework that adds explainability to clustering in a way that assists the end-user in making sense of the clustering outputs through answering their questions about the results.

2 Organization of this Thesis

The rest of this thesis is organized as follows. Chapter 2 reviews the background and related work on clustering and an overview of clustering ASD data. Chapter 3 presents our methodology, followed by the experimental results in Chapter 4. Finally, we make our summary and conclusions in Chapter 5.

CHAPTER II

LITERATURE REVIEW AND BACKGROUND

In this section, we will first review the clustering techniques used in our work. Second, we will give a brief review of clustering ADOS and concept data for autism.

1 Clustering Techniques

The clustering techniques used in this work fall into three classes: partitioning clustering techniques, graph clustering techniques, and co-clustering.

Partitioning Clustering

The goal of partitioning clustering methods [3] is to divide a dataset of n data points into k partitions. Each data point should belong to exactly one partition and each partition should contain at least one data point. Each partition represents one different cluster. The partitioning techniques depend on the objective function used. The most common partitioning algorithm is k-means [4].

The K-means Algorithm

The K-means algorithm [4] [5] [6] partitions the data into k clusters. The number of clusters k must be known a priori (chosen by the user). The best number of clusters k is the number leading to the best separation between the different clusters. Each data point belongs to exactly one cluster. K-means assigns each data point to its closest mean. The algorithm tries to minimize the intra-distance between clusters. The k-means objective function to minimize is the sum of square error as shown in the following equation:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\| \tag{1}$$

k is the number of clusters predefined by the user. n is the number of data points. c_j is the centroid for cluster j. $x_i^{(j)}$ is data point i belonging to cluster j. K-means algorithm steps are as follows:

1. Divide the data randomly into k clusters. The number of clusters is defined by the user.

2. Calculate the mean of all data points belonging to the same cluster. This mean is called the centroid.

3. Assign the data points to the closest calculated centroids according to the euclidean distance.

4. Recompute the centroids of each cluster

5. Repeat step 4 and 5 until the data points do not change in assigned clusters.

The K-means algorithm will stop when it reaches a local minimum of the objective function J.

Hierarchical Clustering

The objective of hierarchical clustering [7] is to find a hierarchical taxonomy within a database and to use it in order to find clusters. The hierarchy within the data is represented by a dendrogram. There are two types of hierarchical clustering methods: Divisive methods and Agglomerative methods [8] [9] (see figure 1 & 2).

Divisive Method

The divisive method [8] is known as a top-down method since the algorithm assigns all the data to the same cluster as a starting point(see figure 1). The divisive method follows the steps below:

1. Assign all the data points to the same cluster.
2. Divide the initial cluster into two least similar clusters.
3. Select a cluster and split it into two clusters.
4. Repeat step 3 recursively until each data point in the database is assigned to a different cluster.

Agglomerative method

The agglomerative method [9] [10] [11] is known also as the bottom-top approach. In fact, the agglomerative approach starts by considering each data point as a separate cluster on its own, then starts merging the closest clusters together until having all

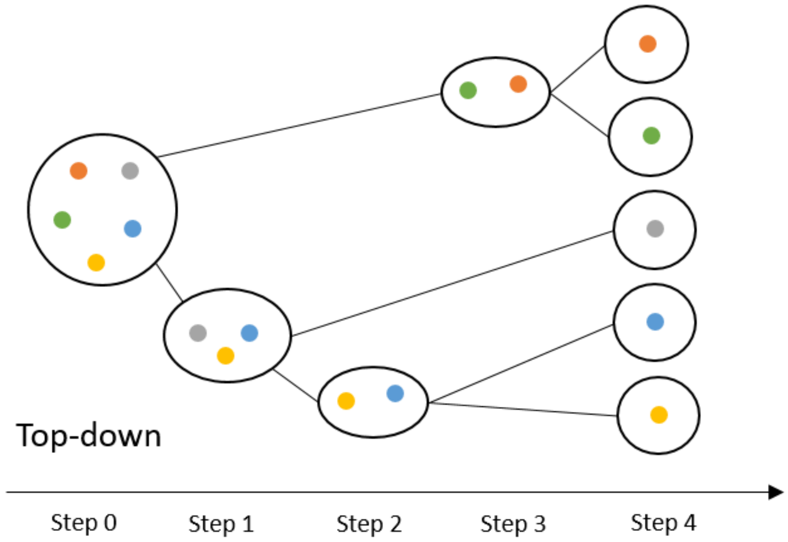


Figure 1. Divisive Approach

the data points into the same cluster (see figure 2). The agglomerative method follows these steps:

1. Assign each data point to a separate cluster.
2. Compute the proximity matrix (distance between each pair of clusters).
3. Merge the closest two clusters.
4. Recompute the proximity matrix between the new cluster and the original clusters.
5. Repeat step 3 and step 4 until all the data points are assigned to the same cluster.

A key step for the agglomerative clustering is to compute the proximity matrix between the different clusters (step 4). The variation of the agglomerative clustering

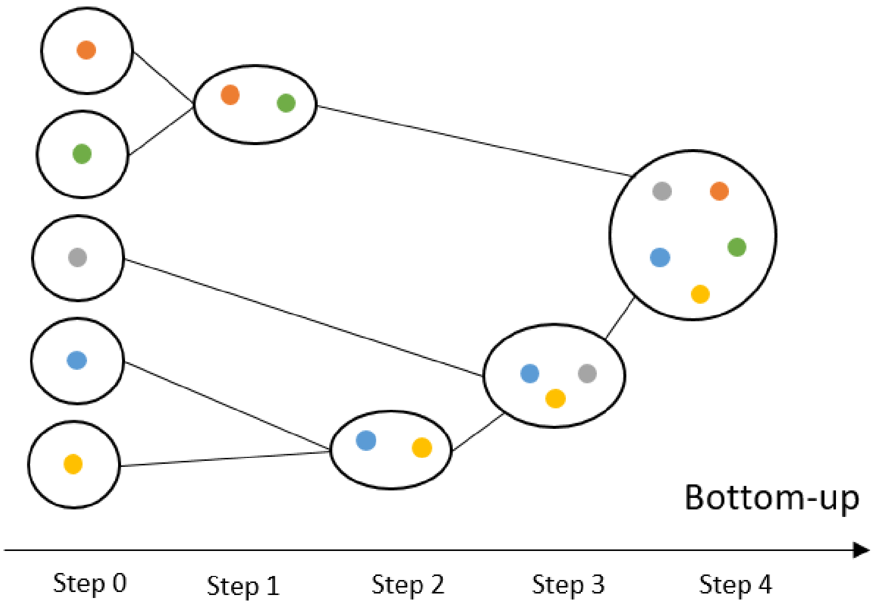


Figure 2. Bottom-up Approach

techniques depends on the approach used to calculate the proximity matrix. The most common techniques to measure the distance matrix or the proximity matrix between two clusters are the Single Linkage, the Complete Linkage and the Average Linkage [9] [10] [11].

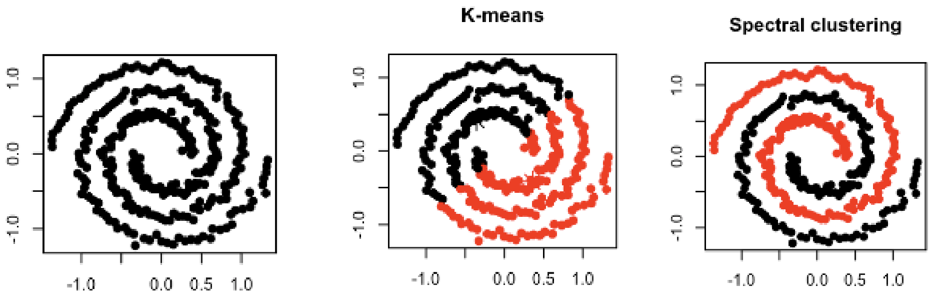
Single Linkage

The distance between two clusters using the single linkage method (called also MIN proximity) is the shortest distance between any two points $x_{i,j}$ and $y_{k,l}$ belonging, respectively, to cluster C_i and cluster C_k.

$$d(C_i, C_k) = min_{x_{i,j} \in C_i, y_{k,l} \in C_k} d(x_{i,j}, y_{k,l}) \tag{2}$$

Complete Linkage

The distance between two clusters using the complete linkage method [12] (called



(a) original data **(b)** partition found by k-means **(c)** partition found by spectral clustering

Figure 3. Spectral Clustering vs K-means (source: `http://scalefreegan.github.io/Teaching/DataIntegration/practicals/p2.html`)

also MAX proximity) is the largest distance between any two data points $x_{i,j}$ and $y_{k,l}$ belonging, respectively, to cluster C_i and cluster C_k.

$$d(C_i, C_k) = max_{x_{i,j} \in C_i, y_{k,l} \in C_k} d(x_{i,j}, y_{k,l}) \tag{3}$$

Average Linkage

The distance between two clusters using the average linkage method (called also group average proximity) is the average distance between each data point in cluster C_i to every data point in cluster C_k.

$$d(C_i, C_k) = \frac{1}{n_{C_i} n_{C_k}} \sum_{j=1}^{n_{C_i}} \sum_{l=1}^{n_{C_k}} d(x_{i,j}, y_{k,l}) \tag{4}$$

Graph clustering

Spectral clustering [13] [14] [15] is a graph based clustering [16] that tries to assign data points that are connected to the same cluster even if the distance between the data points in the same cluster is larger than the data points that belong to the same cluster. Spectral clustering focuses on the connectivity of the data points, contrary to k-means that focuses on their compactness (see figure 3). Spectral clustering is classified as graph partitioning clustering since the data points are considered as graph nodes. The main steps for spectral clustering are shown below:

1. Compute the similarity graph.
2. Map the data in a lower dimensional space using Laplacian graph.
3. cluster the data in the new space.

Each step is described below.

Similarity Graph

The goal of this step is to transform a set of data points to an undirected graph G=(V,E) using pairwise similarities $s_{i,j}$ or distances $d_{i,j}$ between data points. The set of vertex V=v_1, v_2,...,v_n corresponds respectively to 1, 2,..., n data points. The most popular construction methods are: the -neighborhood graph, k-nearest neighbor graphs, the fully connected graph.

The ϵ-neighborhood graph

All data points that have a pairwise distance less than ϵ are connected [14]. The ϵ-neighborhood graph is considered as an unweighted graph because all the connected data points have roughly the same scale (at most ϵ).

k-nearest neighbor graphs

In the k-nearest neighbor graphs [17] [18], vertex v_i is connected to vertex v_j if v_i belongs to the k nearest neighbors of v_j. The problem here is that the resulting graph will be a directed graph. There are two methods to make the constructed graph undirected. The first method is to ignore the directions. In fact, if vertex v_i is among the k nearest neighbors of vertex v_j or if vertex v_j is among the k nearest neighbors of vertex v_i, then v_i and v_j are connected with an undirected edge. The obtained graph is called the K-nearest neighbor graph.The second method is to connect the vertices v_i and v_i only if v_i belongs to the K nearest neighbors of v_j and v_j belongs to the k nearest neighbors of v_i. The obtained graph is called the mutual K-nearest neighbor graph. The weights of the graph edges in both cases are the similarity values between the adjacent points.

The fully connected graph

For the fully connected graph, all the data points are connected to each other. The weight of the graph edges is the similarity $s_{i,j}$. Since the goal of constructing the graph is to model local neighborhood, this method is used only if the similarity function is able to encode the local neighborhood such as the Gaussian similarity function.

$$s(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{5}$$

In this case, σ controls the width of the neighborhood similarly to the ϵ in the ϵ-neighborhood graph. To summarize, the output of this step is the similarity graph G which is a positive, undirected, weighted graph.

Dimensionality reduction

After creating the similarity graph G, our goal is to map our data into a lower-dimensional space. To do so, we compute the graph Laplacian matrix [19]. There are several graph Laplacians in the literature. In this section we will present the unnormalized graph Laplacian and the the normalized graph Laplacian.

The unnormalized graph Laplacian

The unnormalized graph Laplacian [20] is defined as:

$$L = D - W \tag{6}$$

Where D is the degree matrix of the graph (diagonal matrix) and W is the weighted adjacency matrix of the graph. The adjacency matrix W=$(w_{i,j})_{i,j=1,..n}$ models the weight between the vertices with $w_{i,j} >= 0$.

$$d_i = \sum_{j=1}^{n} w_{i,j} \tag{7}$$

The most important properties of Laplacian graphs that are useful for spectral clustering are the following:

1. L is symmetric and positive semi-definite.

2. L has n non-negative, real-valued eigenvalues 0= $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$

3. If L has k eigenvalues equal to 0 for k different eigenvectors, then the undirected graph with non-negative weights G has k connected components. For L, the eigenspace of *0* is spanned by the indicator vectors $\mathbb{1}_{A_i}$ of those components.

The eigenvalues (λ) are computed using the following equation:

$$L\boldsymbol{v} = \lambda\boldsymbol{v} \tag{8}$$

Where $\boldsymbol{v}$ is the eigenvector of L that corresponds to the eigenvalue λ.

The purpose behind computing the Laplacian matrix is to find the eigenvalues and eigenvectors for L that will allow working on the data into a lower dimensional space. Hence, after computing the Laplacian matrix L, we compute its first k eigenvectors $(v_i)_{i=1..k}$ that will form a matrix $V \in \mathbb{R}^{n\times k}$ (V is a representation of the data into a lower dimensional space). The last step is to cluster the rows $(y_i)_{i=1..n}$ of matrix V in $\mathbb{R}^k$ using k-means into k clusters $(c_i)_{i=1..k}$. The output of this algorithm is clusters $(C_i)_{i=1..k}$ such as $(C_i) = \{j|y_j \in c_i\}$. The spectral clustering in this case is called unnormalized spectral clustering referring to the use of the unormalized Laplacian matrix.

The normalized graph Laplacian

There are two normalized graph Laplacian matrices in the literature denoted L_{sym} (because the matrix is symmetric) and L_{rw} (because the matrix is closely connected to a random walk). The matrices are defined respectively as follows:

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{9}$$

$$L_{rw} = D^{-1} L = I - D^{-1} L \tag{10}$$

Where D is the degree matrix of the graph (diagonal matrix) and W is the weighted adjacency matrix of the graph. The adjacency matrix W=$(w_{i,j})_{i,j=1,..n}$ models the weight between the vertices with $w_{i,j}$ >= 0. L is the unnormalized graph Laplacian computed as mentioned in the previous section.

$$d_i = \sum_{j=1}^{n} w_{i,j} \tag{11}$$

The purpose behind computing the normalized Laplacian matrix is to find the eigenvalues and eigenvectors for L_{sym} or L_{rw} that will allow the mapping of the data into a lower dimensional space.

The most important properties of L_{sym} and L_{rw} that are useful for spectral clustering are the following:

1. L_{sym} and L_{rw} are positive semi-definite and have n non-negative, real-valued eigenvalues *0*= $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.
2. λ is an eigenvalue of L_{rw} with eigenvector $\boldsymbol{v}$ if and only if λ is an eigenvalue of

L_{sym} with eigenvector $w = D^{-1/2}v$.

3. λ is an eigenvalue of L_{rw} with eigenvector v if and only if λ and v solve the generalized eigenproblem $Lv = \lambda Dv$.

4. *0* is an eigenvalue of L_{rw} with the constant one vector $\mathbb{1}$ as eigenvector. *0* is an eigenvalue of L_{sym} with eigenvector $D^{-1/2}\mathbb{1}$.

5. If L_{rw} and L_{sym} have k eigenvalues equal *0* for k different eigenvectors, then the undirected graph with non-negative weights G has k connected components. For L_{rw}, the eigenspace of *0* is spanned by the indicator vectors $\mathbb{1}_{A_i}$ of those components. For L_{sym}, the eigenspace of *0* is spanned by the vectors $D^{1/2}\mathbb{1}_{A_i}$.

There are two different methods to work the data into a lower dimensional space depending on the method used to compute the normalized graph Laplacian used L_{sym} or L_{rw}.

1. Using L_{sym}: After computing the normalized Laplacian matrix L_{sym}, we compute its first k eigenvectors $(v_i)_{i=1..k}$ that will form a matrix $V \in \mathbb{R}^{n \times k}$ (V is a mapping of the data into a lower dimensional space). After that, we will normalize the matrix V to have row sums equal to norm *1*. Let U be the normalized version of V with $u_{i,j} = v_{i,j}/(\sum_k v_{i,k}^2)^{1/2}$. The last step is to cluster the rows $(y_i)_{i=1..n}$ of matrix U in $\mathbb{R}^k$ using k-means into k clusters $(c_i)_{i=1..k}$. The output of this algorithm is clusters $(C_i)_{i=1..k}$ such as $(C_i) = \{j | y_j \in c_i\}$.

2. Using L_{rw}: First we compute the unnormalized Laplacian matrix L. Then, we compute its first k eigenvectors $(v_i)_{i=1..k}$ of the generalized eigenproblem

$Lv = \lambda Dv$ that will form a matrix $V \in \mathbb{R}^{n \times k}$. Based on the property 3 of the normalized Laplacian matrix, the eigenvectors found are the eigenvalues of the matrix L_{rw}. The last step is to cluster the rows $(y_i)_{i=1..n}$ of matrix V in $\mathbb{R}^k$ using k-means into k clusters $(c_i)_{i=1..k}$. The output of this algorithm is clusters $(C_i)_{i=1..k}$ such as $(C_i) = \{j|y_j \in c_i\}$.

The spectral clustering in both cases is called normalized spectral clustering referring to the use of normalized Laplacian matrices.

Biclustering

Biclustering is a clustering technique that clusters simultaneously both rows and columns of a data matrix [21]. The difference between clustering and biclustering is the following:

1. Biclustering identifies groups of rows with similar/coherent values under a *specific subset* of features.

2. Clustering identifies groups of rows (or features) that show similar values under *all* the features.

In this section, we will review two popular biclustering algorithms that have been widely used for biological data analysis and document clustering: Spectral Co-Clustering and Spectral Biclustering.

Spectral Co-Clustering

The spectral co-clustering algorithm [22] assumes that the data has an exclusive row and column bicluster structure. In other words, each row and each column belong to exactly one bicluster. The algorithm finds biclusters along the diagonal that have increasing values. The Spectral Co-Clustering algorithm treats the input data matrix as a bipartite graph: the rows and columns of the matrix correspond to the two sets of vertices, and each entry corresponds to an edge between a row and a column. It approximates the normalized cut of this graph to find heavy subgraphs. Spectral Co-Clustering follows the following steps:

1. Preprocess the input matrix A as follows:

   $$A_n = R^{-1/2} A \; C^{-1/2} \tag{12}$$

   Where R is is the diagonal matrix with entry i equal to $\sum_j A_{i,j}$, C is the diagonal matrix with entry j equal to $\sum_i A_{i,j}$.

2. Compute the singular vectors of the matrix A_n

   $$A_n = U \Sigma V^T \tag{13}$$

3. Cluster the rows of Z using k-means where Z is:

   $$Z = \begin{bmatrix} R_{-1/2} U' \\ C_{-1/2} V' \end{bmatrix} \tag{14}$$

Where $U' = [u_2, .., u_l + 1]$ is a subset of the singular vectors forming and $V' = [u_2, .., u_l + 1]$ is a subset of the singular vectors. U' provides the row partitions and V' provides the the column partitions. $l = \lceil log_2(k) \rceil$ with k the number of biclusters.

Spectral Biclustering

Spectral biclustering [23] assumes that the data has a checkerboard structure. Spectral biclustering follows the following steps:

1. Normalize the input matrix A using one of the following methods: independent row and column normalization, bistochastization or log normalization.

2. Compute the singular vectors of the matrix A as explained in spectral co-clustering section.

3. Approximate each singular vector found in step 2 by a piecewise-constant vector using one dimensional k-means.

4. Rank the singular vectors found in step 2 based on the results of the approximations found in step 3. The closer the singular vector approximation to a piecewise-constant vector to higher the ranking is. The Euclidean distance is used to rank the vectors.

5. Similarly to the spectral co-clustering, U' and V' are the matrices composed respectively of the best left singular vectors and the best right singular vectors chosen based on the ranking found in step 4. The partition of the rows is

obtained by running k-means on the matrix resulting from working the rows of A to a q-dimensional space using V'. The partition of the columns is obtained in the same way as the rows using the best left singular vectors U'.

2 Overview of clustering ADOS and Behavioral autism data in the literature

Cluster analyses of behavior, cognitive, and sensory issues are rare in the ASD literature and very limited. Most of the studies are about classifying the fMRI data in order to predict whether the subject is ASD or Control [24] [25] [26], which is not the aim of our research. Concerning ADOS data, most studies identified 2-5 major ASD subgroups [27]. In general, the studies do not compare multiple clustering techniques. Instead a few clustering techniques are usually used, such as k-means, hierarchical clustering [28], network analysis [29], and k-dimensional subspace clustering algorithm [30].

CHAPTER III

MACHINE LEARNING PIPELINE

1 Clustering Pipeline

We analyzed the ASD data using clustering techniques adapted from the field of document classification and we also mapped phenotypes to neuroimaging. We developed an unsupervised machine learning pipeline as shown is Figure 4 to mine heterogeneous data sets consisting of ADOS data, concept data, and fMRI prediction data. First, we pre-processed and cleaned the data. After that, we explored the data using several similarity measures to find the ones that capture the differences between the subjects the best. Different clustering techniques were used to find groups of subjects that share similar phenotypes and similar cognitive severity scores. After clustering the data, we evaluated our clusters using both internal and external evaluation metrics. For the external evaluation, we used the fMRI data predictions in order to map our clusters to regions in the brain. In addition to that we used ADOS totals to validate our clusters.

2 Data Description

The data consists of medical records of patients that have been diagnosed for autism. We have three modalities of data: Behavioral data (called also concept data), Autism

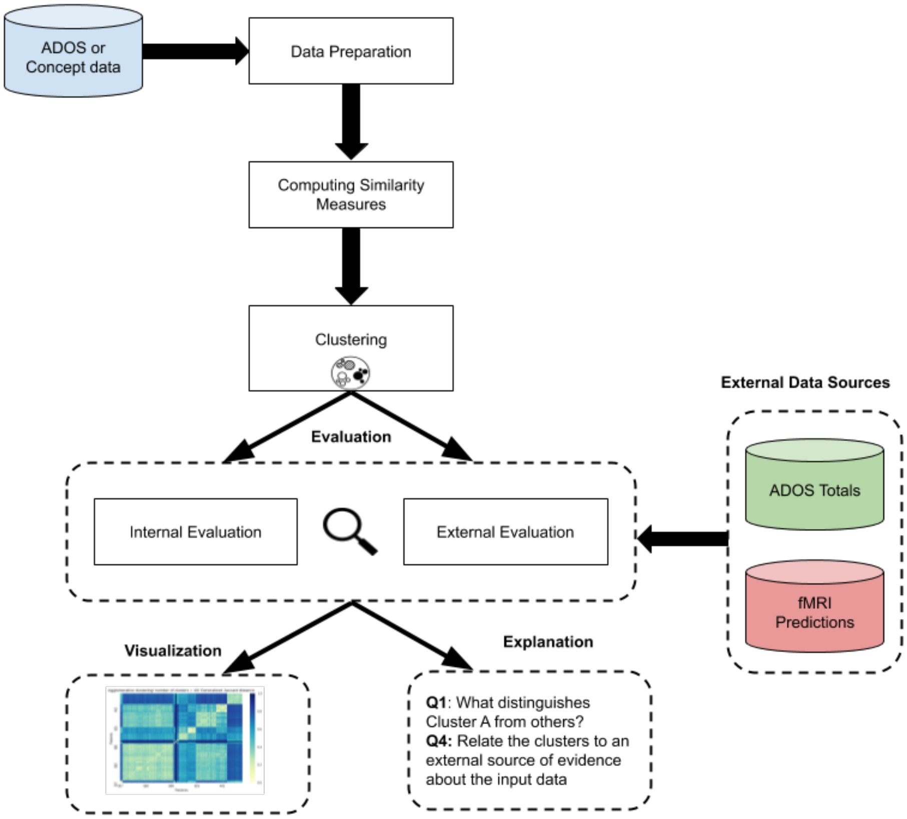


Figure 4. Clustering ASD Data Pipeline

Diagnostic Observation Schedule data (ADOS), and Functional magnetic resonance imaging or functional MRI (fMRI) prediction data.

Autism Diagnostic Observation Schedule data (ADOS data)

The Autism Diagnostic Observation Schedule (ADOS) is a semi-structured assessment of communication, social interaction, and play (or imaginative use of materials) for individuals suspected of having autism or other pervasive developmental disorders. The ADOS is composed of 5 modules depending on the age and the developmental level of the subjects. The modules are: module t, module 1, module 2, module 3 and module 4. The ADOS consists of standardized activities that allow the examiner to observe the occurrence or non-occurrence of behaviors that have been identified as important to the diagnosis of autism and other pervasive developmental disorders [31]. The examiner assigns scores to the test activities and questions in order to describe the severity of the disorder. Hence, the data consists of severity scores assigned by the examiner to the subject based on the subjects answers and reactions. For a better understanding, we present an example of one feature from the ADOS data:

- Example: 'ueye' stands for 'Unusual Eye Contact'. The scores assigned by the examiner could be 0, 1, 2, 3 and 9. The score 0 means that the subject has an appropriate gaze with subtle changes meshed with other communication. The score 1 means that the subject has a definite direct gaze with some modulation; however, it is not consistent and/or it is without subtle changes meshed with other communication. The score 2 means that the subject uses poorly modulated eye contact to initiate, terminate, or regulate social interaction. The score 3 means that the subject uses poorly modulated eye contact to initiate, terminate, or regulate social interaction AND frequently actively avoids eye contact (e.g., by turning away, pushing away, closing eyes). The score 9 means that the corresponding activity cannot be rated for some reason other than that listed above, such as if an examiner makes an error and does not administer a particular ADOS-2 activity.

Behavioral data (concept data)

Behavioral data (also called concept data) presents the phenotype of each subject that has been diagnosed for autism. NDAR [32] distills these phenotypes into a concept vocabulary called the Autism Spectrum Disorder Phenotype Ontology. Autism Spectrum Disorder (ASD) Phenotype Ontology encapsulates the Autism Spectrum Disorder behavioral phenotype, informed by the standard ASD assessment instruments and the currently known characteristics of this disorder. The concepts are distributed across three high-level classes, Personal Traits, Social Competence, and Medical History. Here are three examples of concepts and their hierarchies in the ontology:

- Example 1: Verbal IQ is a phenotype existing under the concept hierarchy *'//Personal Traits//Cognitive Ability//IQ//Verbal IQ'* in the ontology, taking values *'High'*, *'Average'*, or *'Low'*. Hence the resulting concept for the patient will be either *'High Verbal IQ'*, *'Average Verbal IQ'*, or *'Low Verbal IQ'*.

- Example 2: Awareness of Social Cues is a phenotype existing under the concept hierarchy *'//Social Competence//Recognition of Social Norms//Awareness of Social Cues'* in the ontology, taking values *'Good Awareness of Social Cues'*, or *'Poor Awareness of Social Cues'*. Hence the resulting concept for the patient will be either *'Good Awareness of Social Cues'*, or *'Good Awareness of Social Cues'*.

- Example 3: Hearing Loss is a phenotype existing under the concept hierarchy

'//Medical History//Comorbidities//Ear Diseases//Hearing Loss' in the ontology, taking values *'No Hearing Loss/Deafness'*, or *'Known Hearing Loss/Deafness'*. Hence the resulting concept for the patient will be either *'No Hearing Loss/Deafness'*, or *'Known Hearing Loss/Deafness'*.

Functional magnetic resonance imaging or functional MRI Prediction Data (fMRI Prediction data)

The fMRI [33] prediction data consists of predictions [34] that indicate the affected brain areas for subjects that have been diagnosed for ASD. In this thesis we use this data only for explanation purposes.

3 Data Representation

The data will be represented by feature vectors, binary, real valued vectors or adjacency matrix (similarity matrix).

ADOS data

The severity scores in the ADOS data range between 0 and 9. The scores between 0 and 4 relate to the severity of the case. The higher the score is the more severe the case is. The score 9 refers to missing values. The scores that range between 5 and 8 do not have a specific meaning, it depends on the test activity or the test question. Hence, the scores between 5 and 8 are treated case by case based the ADOS data dictionary. Regarding the scores that range between 0 and 4, after discussing with the domain experts, we decided to encode the scores as follows:

- The score 0 means that the subject does not have a disorder for the corresponding activity.
- The score 1 means that the subject has a disorder for the corresponding activity but the disorder is not extremely severe.
- The scores 2 and higher mean that the subject has an extremely severe disorder for the corresponding activity.

The ADOS data was represented by two different ways depending on the clustering technique used:

- First representation: the data is represented by real valued vectors that indicate the severity scores assigned by the examiner to the different subjects.
- Second representation: the data is represented by adjacency matrix that indicates the distance between pairwise subjects. The higher the distance between two patients the more different they are.

Concept data

The concept data was processed using the one hot encoding method. The data was represented by two different ways depending on the clustering technique used:

- First representation: the data is represented as a binary matrix that indicates the behavioral phenotypes for each subject.

- Second representation: the data is represented by an adjacency matrix that indicates the distance between pairwise subjects. The higher the distance between two patients the more different they are.

4 Distance Measures

The success of the majority of clustering techniques depends on the distance measure used. In fact, the distance measures are a tool to quantify the differences between the subjects. The distance measures tested on both data sets (ADOS and concept data) are: Euclidean distance, Manhattan distance, Cosine distance, Jaccard distance and Generalized Jaccard distance. Given two n-vectors $X = (x_i)_{i \in [1..n]}$ and $Y = (y_i)_{i \in [1..n]}$, the distance between X and Y can be defined as:

$$Euclidean\ distance(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_j)^2} \tag{15}$$

$$Manhattan\ distance(X,Y) = \sum_{i=1}^{n} |x_i - y_j| \tag{16}$$

$$Cosine\ distance(X,Y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{17}$$

$$Jaccard\ distance(X,Y) = 1 - \frac{|X \bigcap Y|}{|X \bigcup Y|} \tag{18}$$

$$Generalized\ Jaccard\ distance(X,Y) = 1 - \frac{\sum_{i=1}^{n} min(x_i, y_i)}{\sum_{i=1}^{n} min(x_i, y_i)} \tag{19}$$

5 Clustering Algorithms

In this work, we investigated a variety of clustering algorithms reviewed in Chapter 2, namely:

1. The K-Means algorithm [4] [5] [6]

2. Hierarchical Agglomerative Clustering: we varied the linkage method among Single, Complete and Average. We also tested different distance measures (Euclidean, Cosine, Generalized Jaccard) [9] [10] [11]

3. Spectral Clustering using a variety of kernels (RBF, Linear, Polynomial, Euclidean, Cosine) [14]

4. Spectral Co-clustering [22]

5. Spectral Bi-clustering [23]

6 Evaluation Methodology

Before presenting our clustering results, it is important to present the metrics used to validate our clusters. Clustering evaluation metrics can be categorized into two classes: internal clustering evaluation metrics and external clustering evaluation metrics [35]. The main difference between both classes is that external clustering evaluation metrics use external data.

Internal Clustering Evaluation Metrics

Internal clustering evaluation metrics do not rely on any additional external data. Since the goal of clustering is to assign the data points that are similar to same cluster and the data points that are different to separate clusters, internal evaluation metrics try to measure the compactness and the separation of the clusters. The compactness (or cohesion) is how close are the data points within the same cluster: how similar are the data points belonging to the same cluster. The separation is how distinct the different clusters are from each others. The majority of the evaluation metrics consider both compactness and separation. The evaluation metrics used to evaluate the clustering techniques used are the Silhouette index (S) [36] and the Davies-Bouldin index (DB) [37].

The Silhouette index (S)

The silhouette index (S) [36] measures how similar is a data point is to its own cluster (compactness) compared to other clusters (separation) by calculating the mean intra-cluster distance and the mean nearest-cluster distance for each data point. The equation is the following:

$$S = \frac{1}{NC} \sum_{i} (\frac{1}{n_i} \sum_{x \in C_i} \frac{b(x) - a(x)}{max(a(x), b(x))}) \tag{20}$$

Where NC is the number of clusters, C_i is ith cluster, n_i is the number of data points in C_i, d(x,y) is the distance between two data points x and y, a(x) is the mean intra-cluster distance for cluster C_i, and b(x) is the mean nearest-cluster distance for the data point x. a(x) and b(x) are defined as follows:

$$a(x) = \frac{1}{n_i - 1} \sum_{y \in C_i, y \neq x} d(x, y) \tag{21}$$

$$b(x) = min_{j, j \neq i}[\frac{1}{n_j} \sum_{y \in C_j} d(x, y)] \tag{22}$$

The Silhouette index (S) ranges between -1 and 1, $S \in [-1, 1]$. The clustering is considered good when the Silhouette index is close to 1. When the silhouette index yields negative values that generally indicates that there are data points that have been assigned to the wrong cluster which is considered as bad clustering results. When the silhouette index yields values that are close to 0, that indicates that there are overlapping clusters.

The Davies-Bouldin index (DB)

The Davies-Bouldin index [37] (DB) measures how similar is a data point is to its own cluster (compactness) compared to other clusters (separation). The score is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances.

$$DB = \frac{1}{NC} \sum_i max_{j, j \neq i}([\frac{1}{n_i} \sum_{x \in C_i} d(x, c_i) + \frac{1}{n_j} \sum_{x \in C_j} d(x, c_j)]/d(c_i, c_j)) \tag{23}$$

Where NC is the number of clusters, C_i is ith cluster, c_i is the center of the ith cluster, n_i is the number of data points in C_i, and d(x,y) is the distance between two data points x and y. The Davies-Bouldin index (DB) ranges between 0 and 1, $S \in [0, 1]$. The clustering is considered good when the Davies-Bouldin index is close to 0, since it means that the clusters are the most distinct from each others. One drawback of the Davies-Bouldin index (DB) is that a good value reported by this method does not imply the best information retrieval. In the other hand, the time complexity of the the Davies-Bouldin index computation is less than the time complexity of the Silhouette index.

External Clustering Evaluation Metrics

While internal evaluation methods uses only the internal data and do not need ground truth, external evaluation methods require the presence of labeled data in order to evaluate the quality of the clustering. In fact, the external evaluation methods measures the extent of homogeneous classes in one cluster. In this work, we worked with purity [38] as our external evaluation metric.

Purity

Purity [38] [39] measures how pure are the obtained clusters. In fact, it quantifies the homogeneity in the cluster. If in a given cluster the data is from the same class the the purity is maximized.

$$Purity = \frac{1}{N} \sum_{m \in M} \max_{d \in D} |m \cap d| \tag{24}$$

M is the number of cluster, D is the set of classed and N is the dataset size that we are using.

7 Methodology for Explaining Clustering Results

Clustering algorithms can generate a significant amount of output results that need to be interpreted and judged by humans. The interpretation and understanding depends on the format of the clustering results and the expertise of the end users. These end users can have varying levels of expertise in machine learning, specifically clustering; in the application domain (e.g. autism); or both. The outputs of clustering algorithms can vary in format. Examples of clustering outputs include:

1. **O1**: A set of cluster representatives such as a cluster centroid (a vector where each dimension is the average of one of the features of the data assigned to that cluster) or a cluster medoid (an actual data instance assigned to that cluster).

2. **O2**: The list of data records that are assigned to each cluster.

3. **O3**: Various visualizations of the clusters, such as heatmaps of the feature values of the data assigned to each cluster.

With time, the end user can gain enough expertise in both the application domain and some machine learning to be able to interpret the results. Our aim is to provide a methodology to allow the end user to interrogate the clustering algorithms to explain or justify its results. Specifically the end user may desire to know the answers to the following questions:

1. **Q1**: What distinguishes Cluster A from others
2. **Q2**: What distinguishes Cluster A from Cluster B
3. **Q3**: What distinguishes all the clusters from one another
4. **Q4**: Relate the clusters to an external source of evidence about the input data

Each of the above questions can be answered by reference to the available evidence:

1. **E1**: The data space in which the clustering was performed.
2. **E2**: A different modality or external source of evidence

The right kind of answer to the above questions depends on the nature of the output (**O**), question (**Q**), and evidence (**E**) - see Table 1.

For this reason, we propose a general framework to build an explainability module for clustering algorithms that adheres to the methodology that is summarized in Table 1. For instance we can use a transparent supervised model trained on the actual instances in the original feature space in order to answer a question of type Q2 with reference to Evidence E1 using only an output of type O1. In this case, a transparent supervised model (such as a decision tree or rules) will be learned by analysing the most important features that can divide the original data records into the groups that are defined by Cluster A and Cluster B. Such an explanation can help the end user judge the meaning of Cluster A and Cluster B and even to validate the goodness of these two clusters. In this thesis, we rely on decision trees as the white box supervised model. A decision tree is a classification model consisting of nodes

Output	O1 (representatives)		O2 (list of data records per cluster)		O3 (visualiz-ation)	
Evidence Question	E1	E2	E1	E2	E1	E2
Q1	Relationship between features that are different in Cluster A vs. other clusters in original feature space	Relationship between features that are different in Cluster A vs. other clusters in external data space	White box binary supervised model trained on original data (Cluster A against all)	White box binary supervised model trained on external data (Cluster A against all)	TBD	TBD
Q2	Relationship between features that are different in Cluster A vs. Cluster B in original feature space	Relationship between features that are different in Cluster A vs. Cluster B in external data space	White box binary trained on original data supervised model (Cluster A against B)	White box binary trained on external data supervised model (Cluster A against B)	TBD	TBD
Q3	Relationship between features that are different in each cluster vs. each other cluster in original feature space	Relationship between features that are different in each cluster vs. each other cluster in external data space	White box multi-class trained on original data supervised model (Class = Cluster label)	White box multi-class trained on external data supervised model (Class = Cluster label)	TBD	TBD
Q4	TBD	Medoid only: data record in external evidence that corresponds to the medoid instance	TBD	Association (e.g. correlation) between data in Cluster A and corresponding data in external source	TBD	TBD

Table 1. Cluster Explainability Framework. The right kind of answer to the an explanation questions depends on the nature of the output (**O**), question (**Q**), and evidence (**E**)

that test features of the data in such a way that the data gets recursively partitioned into pure groups based on a group purity measure such as entropy. Furthermore, the decision tree model can be easily translated into a rule-based model and hence it can be used to explain the results.

CHAPTER IV

EXPERIMENTAL RESULTS

1 Experimental protocol

In order to find the best partition for our data sets, we followed the following steps:

- Step 1: Compute the adjacency matrix using the distance measure that captures the differences between subjects the best.

- Step 2: Run different clustering algorithms on each data set.

- Step 3: Try different combinations of parameters for each clustering algorithm tested in step 2.

- Step 4: Compute the internal and external validity metrics for each clustering result obtained from step 2 and step 3.

- Step 5: Choose the best clusters based on the results of step 4.

- Step 6: Provide different visualizations to analyze and explain the found clusters.

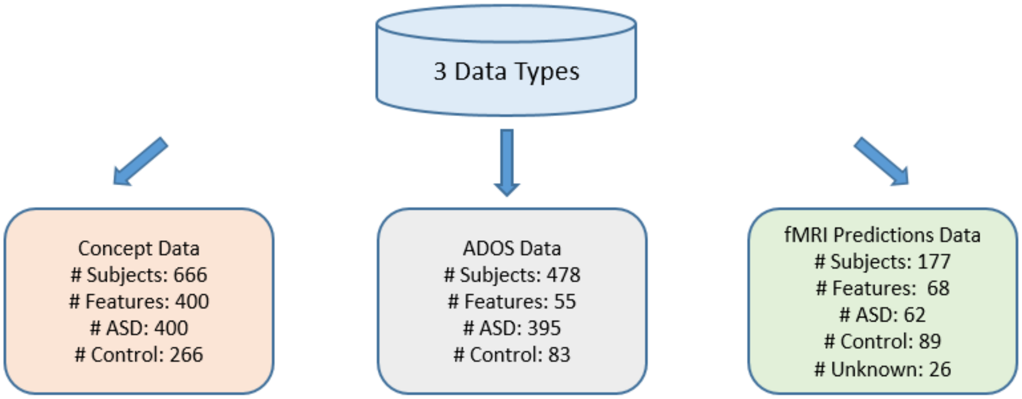


Figure 5. Data Types

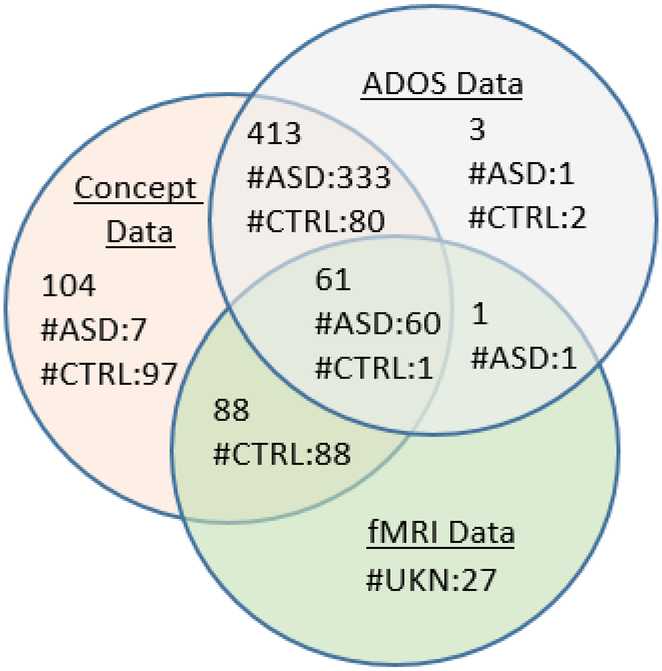


Figure 6. Data set overlap

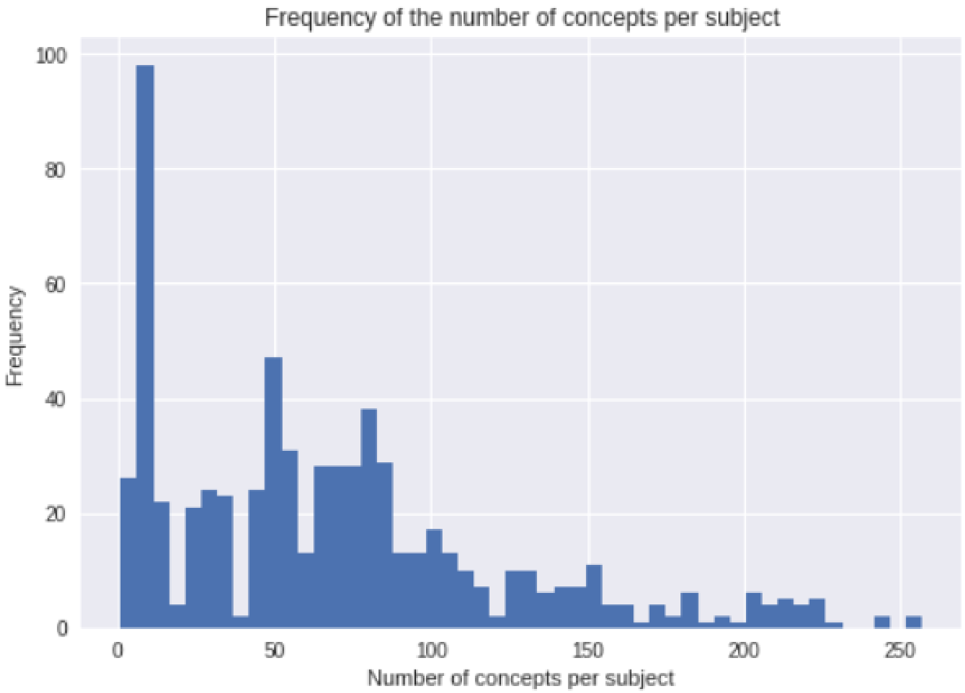


Figure 7. Frequency of the number of concepts per subject

2 Data Exploration

Our experiments are based on three data sets: ADOS data, Concept data and fMRI data. The fMRI data was used only for the validation process in order to map the found clusters to the affected areas of the brain. Figure 5 summarizes the sizes of the data sets used in our research work. The different data sets overlap as described in figure 6. All data sets include both autistic (ASD) and control (non ASD) patients.

- ADOS data comprises 478 subjects: 395 subjects are autistic (ASD) and 83 subjects are non autistic (control).
- Concept data comprises 666 subjects: 400 subjects are autistic (ASD) and 266 subjects are non autistic (control). The data was encoded using one hot

encoding, resulting in binary data set composed from subject keys as rows and concepts as features. The concept data comprises 400 concepts. The resulting data is a sparse data as illustrated in figure 7. Figure 7 presents the distribution of the number of concepts per subject.

- fMRI prediction data: comprises 177 subjects: 62 subjects are autistic (ASD), 89 subjects are non autistic (control), and 26 subjects are unknown..

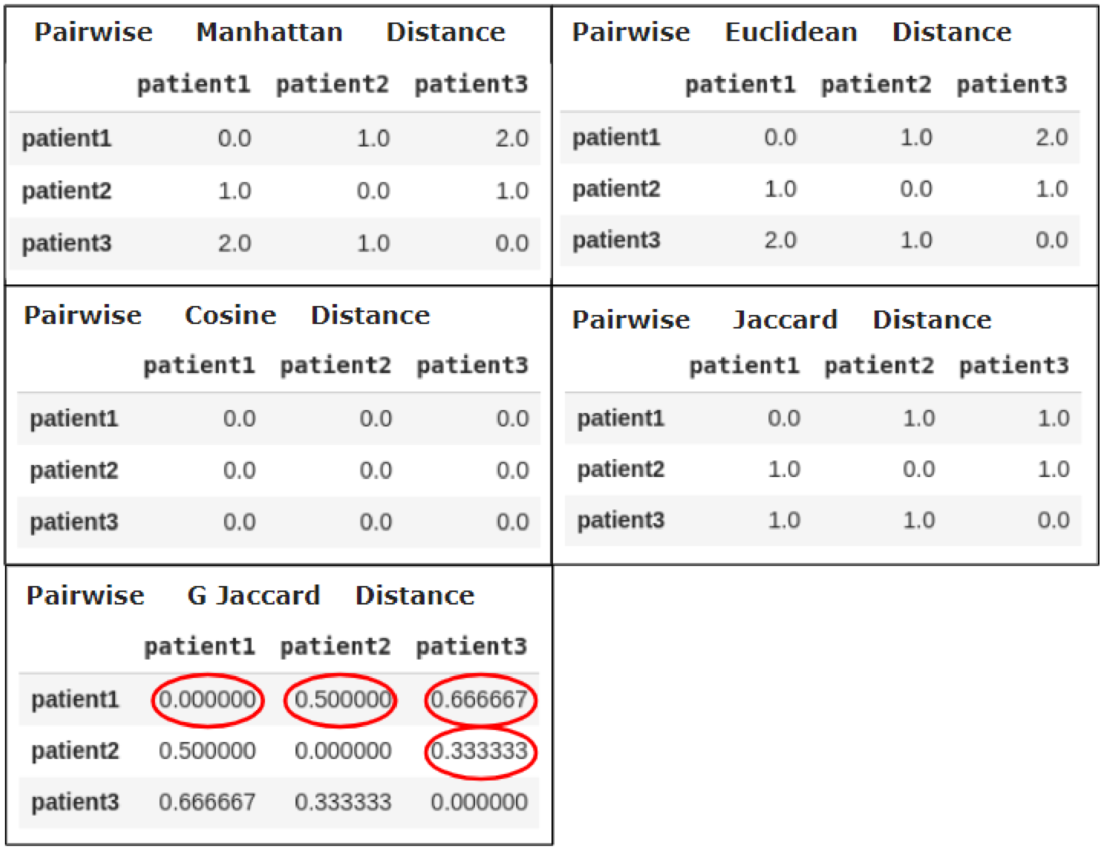
3 Similarity measure selection

Computing distance measures

Measuring the differences between subjects is a key step toward a successful clustering. Our goal is to find the distance measure that can quantify how different the subjects are from each other. Since the performance of the distance measure depends on the data, we evaluate the performance of the distance measures separately for the ADOS data and the concept data.

Selection of distance measure for the ADOS data

For the ADOS data, we tried the following distance measures: Euclidean distance, Cosine distance, Manhattan distance, Jaccard distance, and Generalized Jaccard distance. Given 3 subjects that have severity scores equal to 0, 1 and 2, respectively for a given test activity such that $S_1 = [0, .., 0]$, $S_2 = [1, ..., 0]$, $S_3 = [2, ..., 0]$. According to the domain experts, a good distance measure should encode the following information:



Pairwise Manhattan Distance

	patient1	patient2	patient3
patient1	0.0	1.0	2.0
patient2	1.0	0.0	1.0
patient3	2.0	1.0	0.0

Pairwise Euclidean Distance

	patient1	patient2	patient3
patient1	0.0	1.0	2.0
patient2	1.0	0.0	1.0
patient3	2.0	1.0	0.0

Pairwise Cosine Distance

	patient1	patient2	patient3
patient1	0.0	0.0	0.0
patient2	0.0	0.0	0.0
patient3	0.0	0.0	0.0

Pairwise Jaccard Distance

	patient1	patient2	patient3
patient1	0.0	1.0	1.0
patient2	1.0	0.0	1.0
patient3	1.0	1.0	0.0

Pairwise G Jaccard Distance

	patient1	patient2	patient3
patient1	0.000000	0.500000	0.666667
patient2	0.500000	0.000000	0.333333
patient3	0.666667	0.333333	0.000000

Figure 8. Selection of distance measure for the ADOS data

- The distance between subject 1 and subject 2 should be smaller than the distance between subject 1 and subject 3. In other words, $d(S_1, S_2) \leq d(S_1, S_3)$.

- The distance between subject 2 and subject 3 should be smaller than the distance between subject 1 and subject 2 and the distance between subject 1 and subject 3. In other words, $d(S_2, S_3) < d(S_1, S_2) < d(S_1, S_3)$.

After running our experiments, we found that the Generalized Jaccard distance is the distance measure that best captures the differences between the subjects for the ADOS data as shown in figure 8.

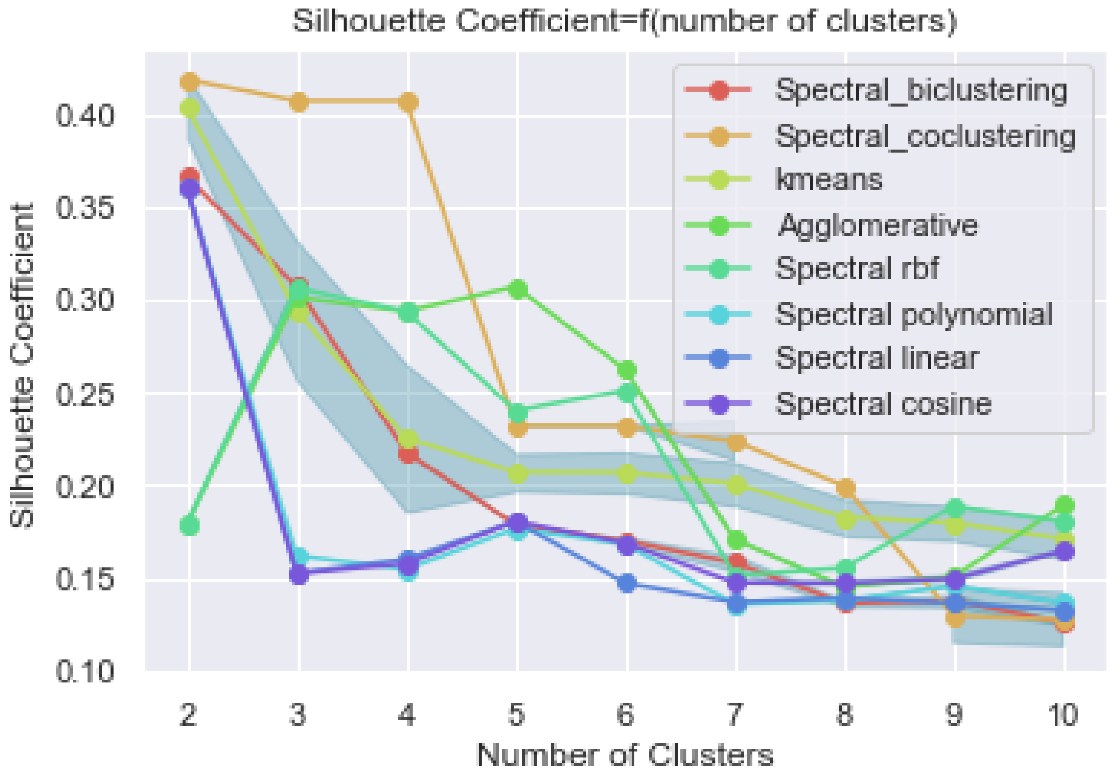


Figure 9. Variation of Silhouette Coefficient with the number of clusters for ADOS data

4 Clustering Evaluation

The clustering of both ADOS and Concept data was evaluated by several clustering techniques including K-means algorithm, Agglomerative Hierarchical clustering, spectral clustering, and biclustering.

ADOS Data

First, we evaluated several clustering algorithms while varying the number of clusters. We computed the silhouette coefficient and the Davies Bouldin index internal evaluation metrics. Figure 9 shows the silhouette score for the different clustering algorithms used. For the silhouette score, the higher the score the better the parti-

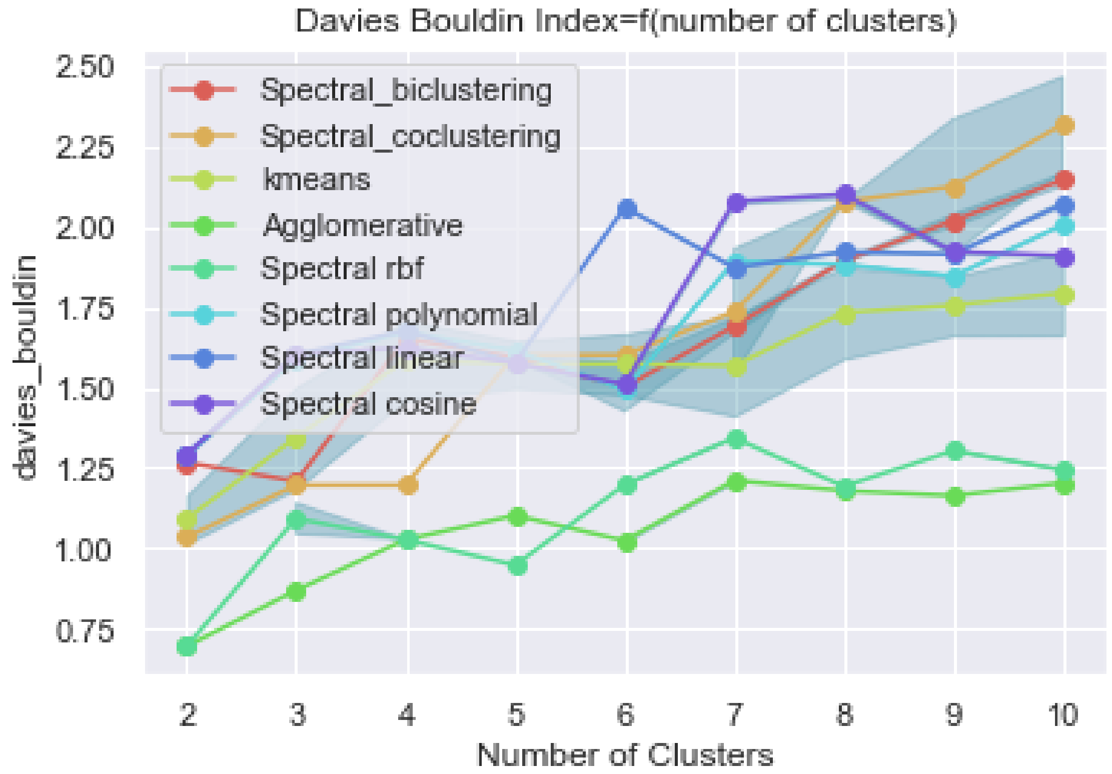


Figure 10. Variation of Davies Bouldin Index with the number of clusters for ADOS data

tion. Figure 10 shows the Davies Bouldin index for the different clustering algorithm results. For the Davies Bouldin index, the lower the score the better the partition. Based on our experiments Agglomerative hierarchical clustering and Spectral biclustering provide the best results for the ADOS data.

Agglomerative Clustering Analysis for ADOS Data

After selecting the best clustering algorithms, we will show several data visualizations to analyze our clustering results. Figure 11 shows the clustering quality. The diagonal of the heatmap shows similar pattern detected by the algorithm. The Figure shows

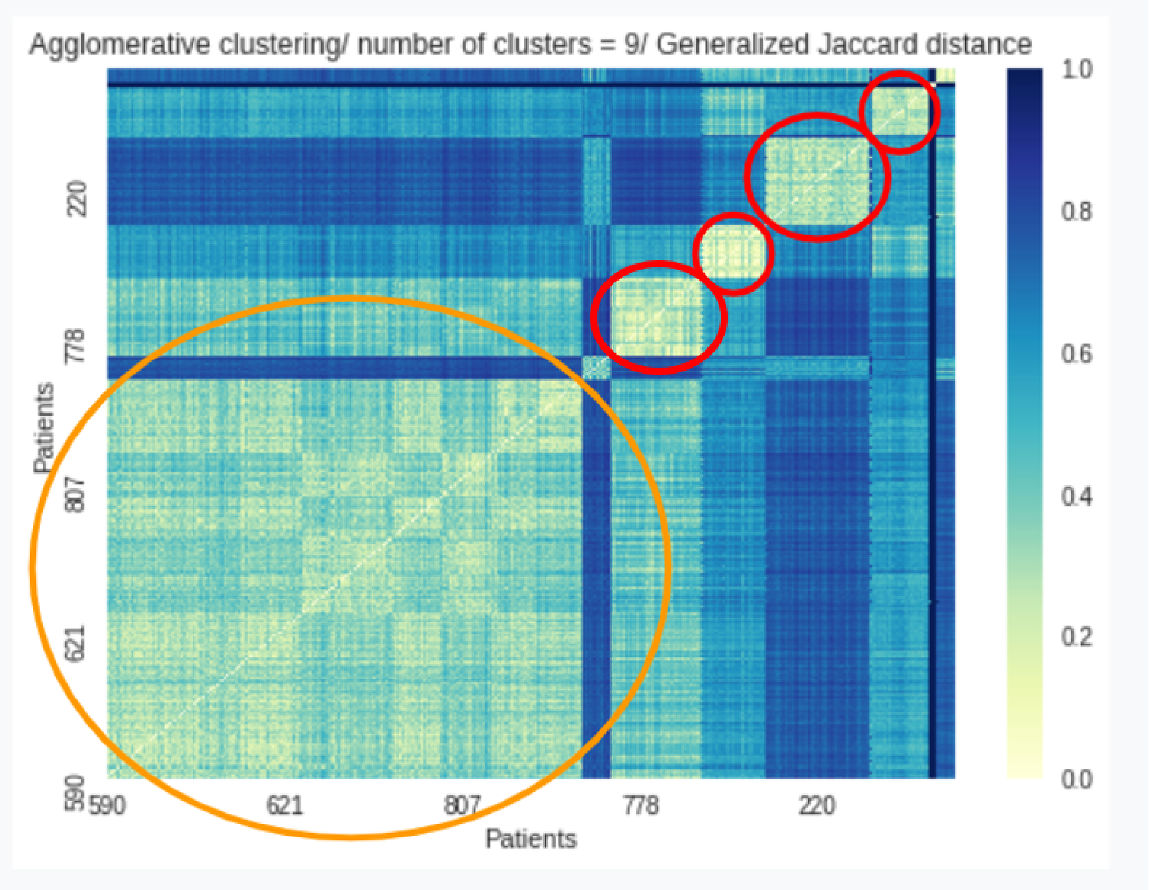


Figure 11. Agglomerative Clustering for ADOS data: Cluster the rows (patients) Input data: Generalized Jaccard distance, k = 8 optimal (validated) clusters

four good clusters (see red circles) with high internal cohesion and good inter-cluster separation and 1 large cluster (yellow circle) with many nested smaller clusters.

We have the cluster labels of ASD patients and control patients. The purity metric will inform us about the clustering performance. Agglomerative clustering returned a score of **0.91**. This shows that the algorithm gave good clusters.

Biclustering Clustering Analysis for ADOS Data

Figure 12 shows the raw data before and after the biclustering. Spectral Biclustering discovers clusters in both rows (patients) and columns (ADOS survey questions). The

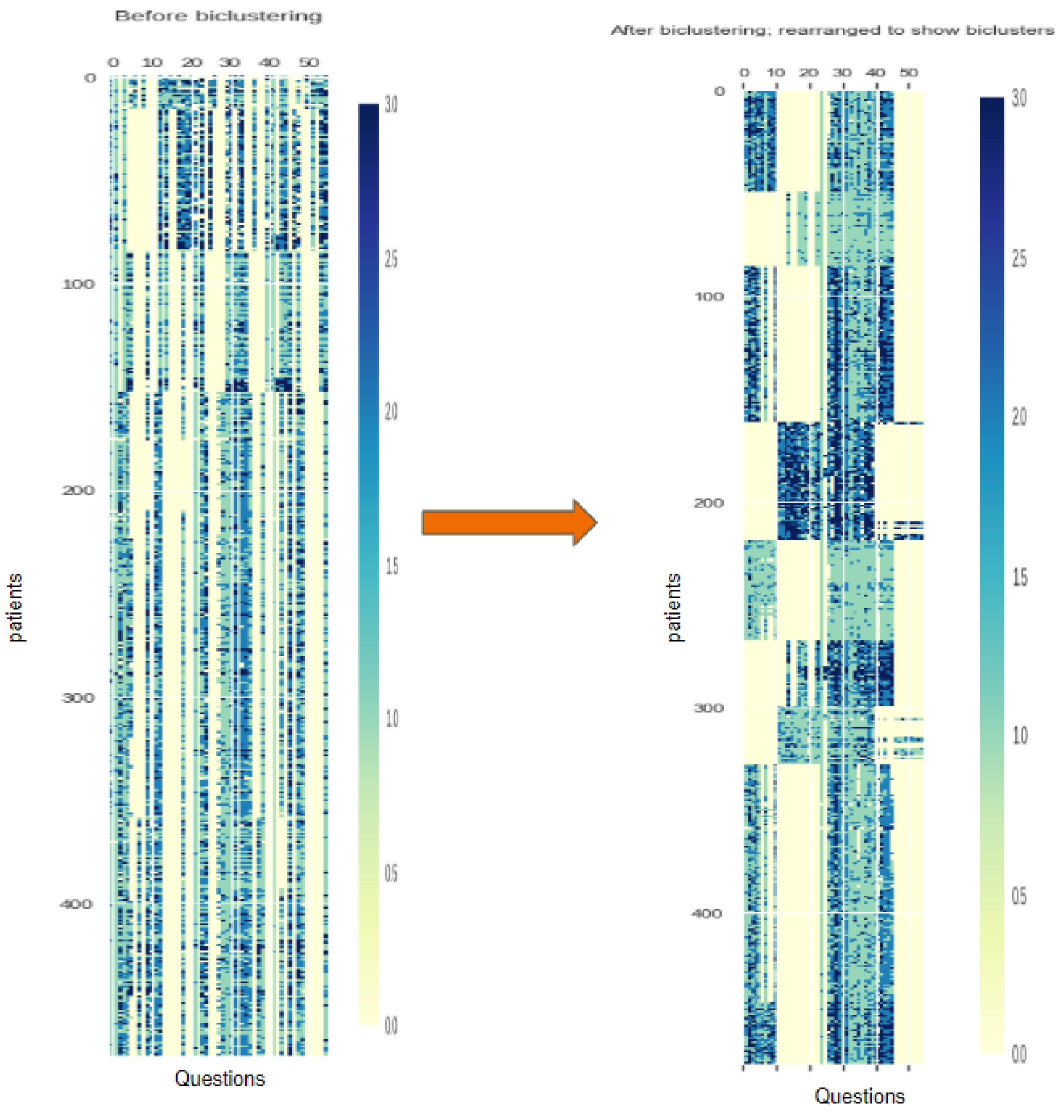


Figure 12. Spectral BiClustering on ADOS data: Cluster the rows (patients) and the columns (autism test questions) simultaneously Input data: raw data; : k = (9,5) validated clusters

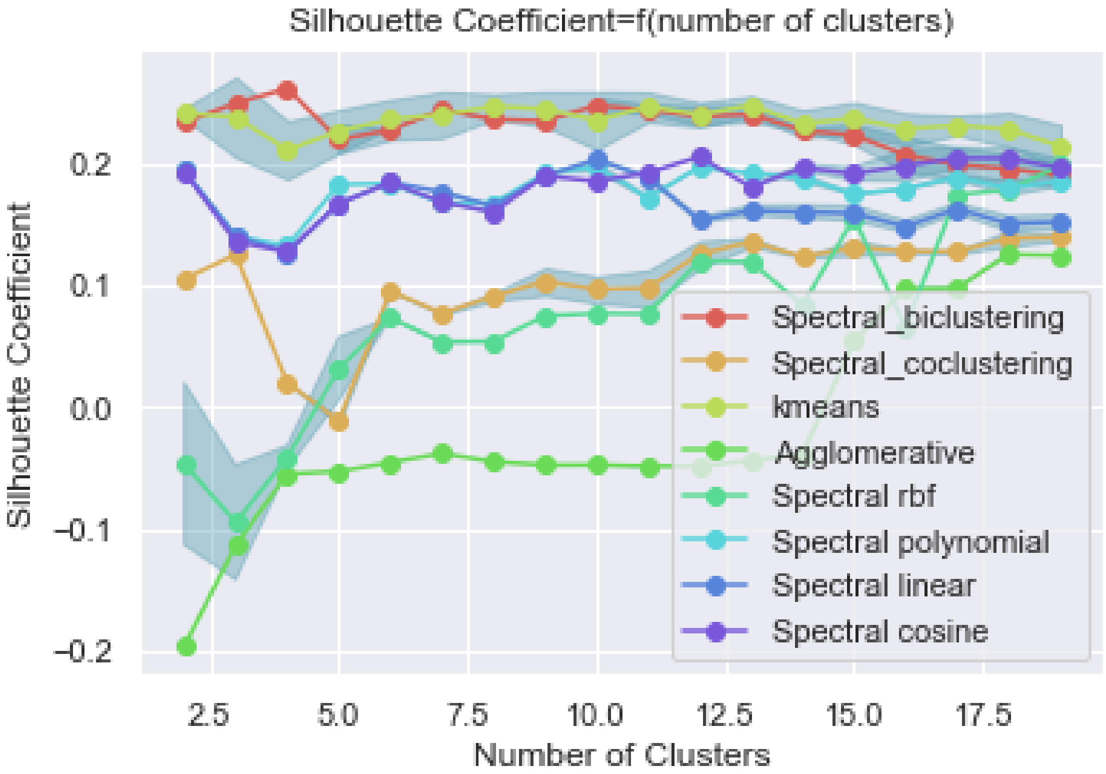


Figure 13. Variation of Silhouette Coefficient with the number of clusters for Concept Data

figure shows that the clusters are more meaningful and pure.

Finally to externally validate the purity of the algorithms, we also calculated the purity of the Spectral Biclustering clusters. The purity was 0.90 which confirms the quality of the clustering.

Concept Data

Now we are going to perform the same experimental protocol we implemented in the previous section for the Concept Data. First, we evaluate several clustering algorithms in order to identify the best algorithm based on the internal validity metric.

According to Figure 13 Spectral Biclustering and K-means are outperforming

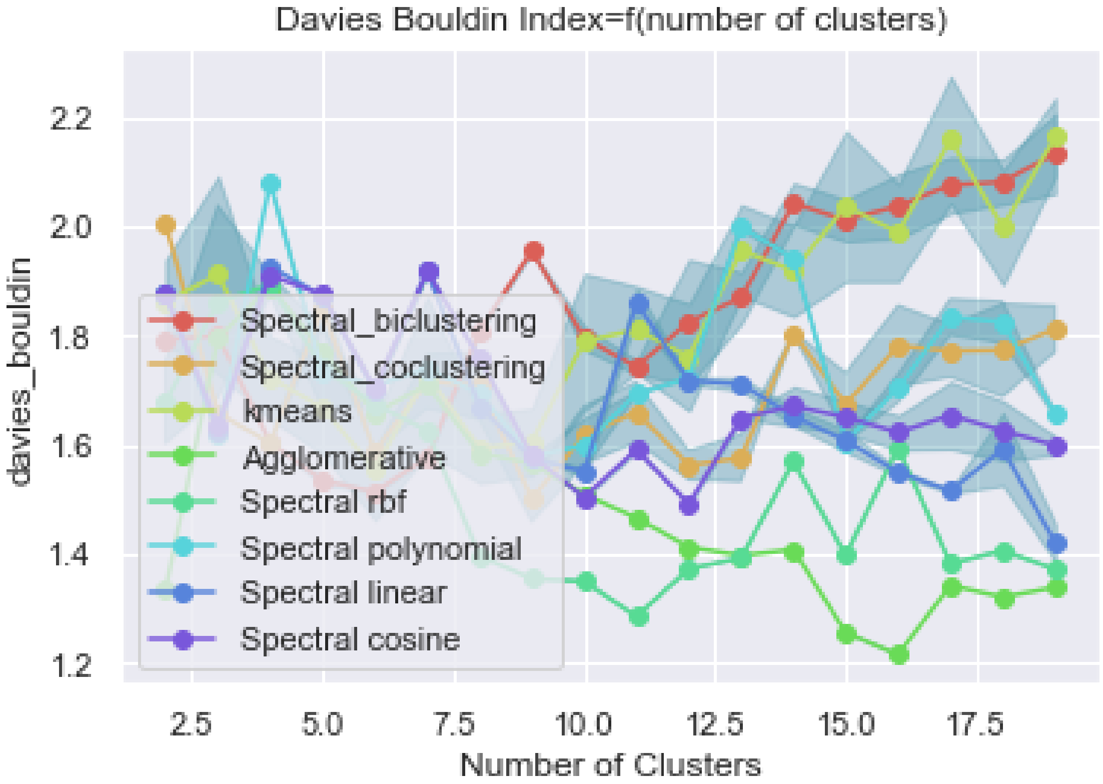


Figure 14. Variation of Davies Bouldin Index with the number of clusters for Concept Data

other algorithms. However K-means has a higher a higher variance because of the random initialization. Furthermore, the Davies Bouldin evaluation in Figure 14 shows a superior performance for Agglomerative clustering. However a low DB score does not necessarily mean a good clustering performance. For this reason, we will choose to work with Spectral Biclustering.

Spectral Biclustering for Concept Data

Figure 15 shows the raw data before and after applying the biclustering algorithm. We can clearly see the apparent blocks of the similar subjects.

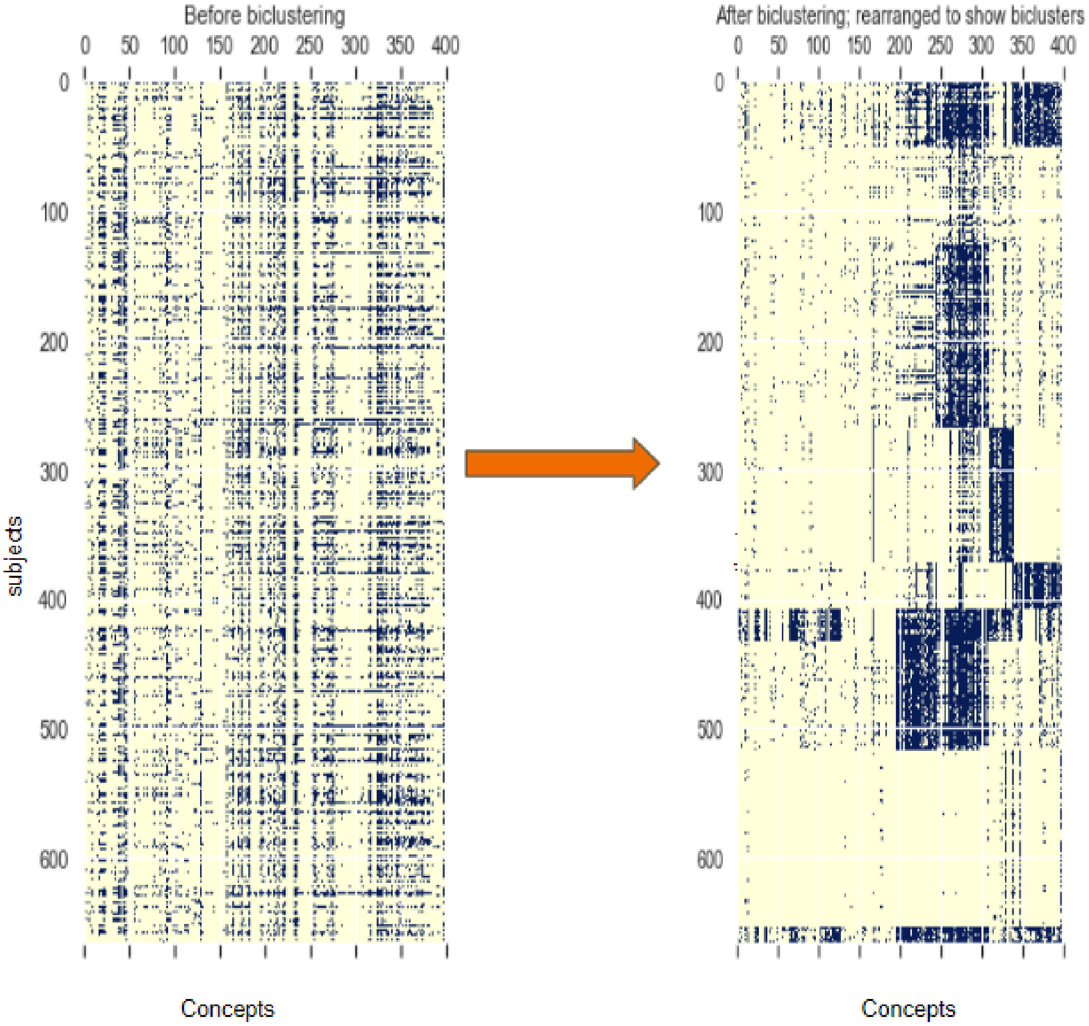


Figure 15. Spectral BiClusteringon Concept data: Cluster the rows (patients) and the columns (autism test questions) simultaneously Input data: raw data; : k = (9,5) validated clusters

5 Clustering Explainability

In this section we present the results of our expainability module. The outputs of clustering algorithms can vary in format. Our clustering outputs include:

1. **O2**: The list of data records that are assigned to each cluster.

2. **O3**: Various visualizations of the clusters, such as heatmaps of the feature values of the data assigned to each cluster.

The end user may desire to know the answers to the following questions:

1. **Q1**: What distinguishes Cluster A from others

2. **Q2**: What distinguishes Cluster A from Cluster B

3. **Q3**: What distinguishes all the clusters from one another

4. **Q4**: Relate the clusters to an external source of evidence about the input data

Each of the above questions can be answered by reference to the available evidence:

1. **E1**: The data space in which the clustering was performed which can be in our case either the ADOS data or the Concept data.

2. **E2**: A different modality or external source of evidence. In our case, this is either the ADOS Total scores, The ADOS modules or the fMRI data.

For convenience, we duplicate our Clustering Expainability framework table in Table 2.

Output	O1 (representatives)		O2 (list of data records per cluster)		O3 (visualization)	
Evidence Question	E1	E2	E1	E2	E1	E2
Q1	Relationship between features that are different in Cluster A vs. other clusters in original feature space	Relationship between features that are different in Cluster A vs. other clusters in external data space	White box binary supervised model trained on original data (Cluster A against all)	White box binary supervised model trained on external data (Cluster A against all)	TBD	TBD
Q2	Relationship between features that are different in Cluster A vs. Cluster B in original feature space	Relationship between features that are different in Cluster A vs. Cluster B in external data space	White box binary trained on original data supervised model (Cluster A against B)	White box binary trained on external data supervised model (Cluster A against B)	TBD	TBD
Q3	Relationship between features that are different in each cluster vs. each other cluster in original feature space	Relationship between features that are different in each cluster vs. each other cluster in external data space	White box multi-class trained on original data supervised model (Class = Cluster label)	White box multi-class trained on external data supervised model (Class = Cluster label)	TBD	TBD
Q4	TBD	Medoid only: data record in external evidence that corresponds to the medoid instance	TBD	Association (e.g. correlation) between data in Cluster A and corresponding data in external source	TBD	TBD

Table 2. Cluster Explainability Framework. The right kind of answer to the an explanation questions depends on the nature of the output (**O**), question (**Q**), and evidence (**E**)

Below we show our results for each of a select set of possible questions for each clustering result presented in the previous section.

Q2: What distinguishes Cluster A from cluster B?

We trained a transparent supervised model on the actual instances in the original feature space in order to answer a question of type Q2 with reference to Evidence E1 using only an output of type O1. In this case, we learned a decision tree by analyzing the most important features that can divide the original data records into the groups

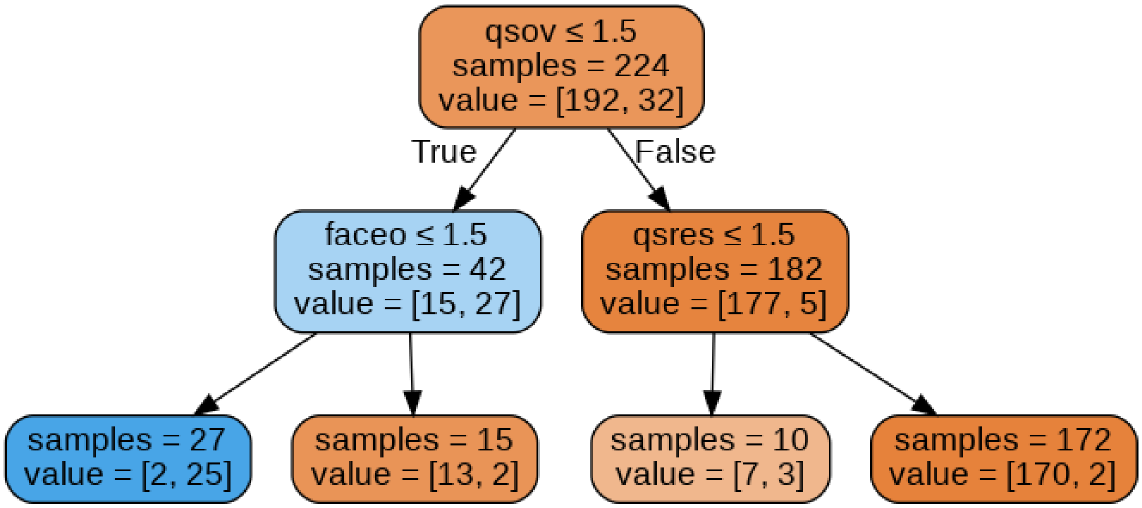


Figure 16. Example of Decision tree used in an explanation of type Q2-E1 for an output O2 resulting from clustering ADOS data using Agglomerative results, comparing cluster 0 to cluster 2 (accuracy=92.7%)

that are defined by Cluster A and Cluster B. The decision tree model can be easily translated into a rule-based model and hence it can be used to explain the results in human understandable format. Such an explanation can help the end user judge the meaning of Cluster A and Cluster B and even to validate the goodness of these two clusters. Although our long term goal is to convey these explanations in natural language, at the time being, we will show results based on rules and tree-based model visualization.

ADOS Data Explanation Results

Concept Data Explanation Results

Q3: What distinguishes all the clusters from one another?

We followed the same strategy as the first question by training a decision tree model. We presented the results for both the ADOS data and the Concept data.

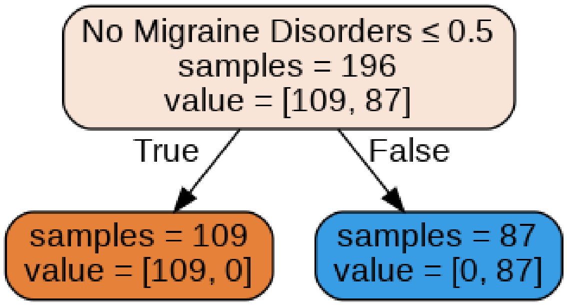


Figure 17. Example of Decision tree used in an explanation of type Q2-E1 for an output O2 resulting from clustering Concept data using Spectral bi-clustering results, comparing cluster 2 to cluster 7 (accuracy=98%). The existence of Migraine Disorder is distinguishing the clusters

ADOS Data Explanation Results

To explain the results of the biclustering algorithm, we trained a Decision tree model. Figure 18 shows the decision tree obtained with an accuracy of 86%. The path of the tree can be translated to rule-based explanations that can explain why a given instance is included in a given cluster. For instance, Figure 18 shows that the ”empth” feature is the most important one that divides the clusters followed by ”ssmle” and ”spabn”.

Concept Data Explanation Results

To explain the results of the biclustering algorithm, we trained a Decision tree model. The resulting tree in Figure 19 will help us find out the most important features that were used to differentiate each cluster. The classification result we obtained has a 90 % accuracy. Figure 19 shows that the features *imaginative/creative*, *Good Social Interest* and *No OCD* are the most important features used in order to create the clusters.

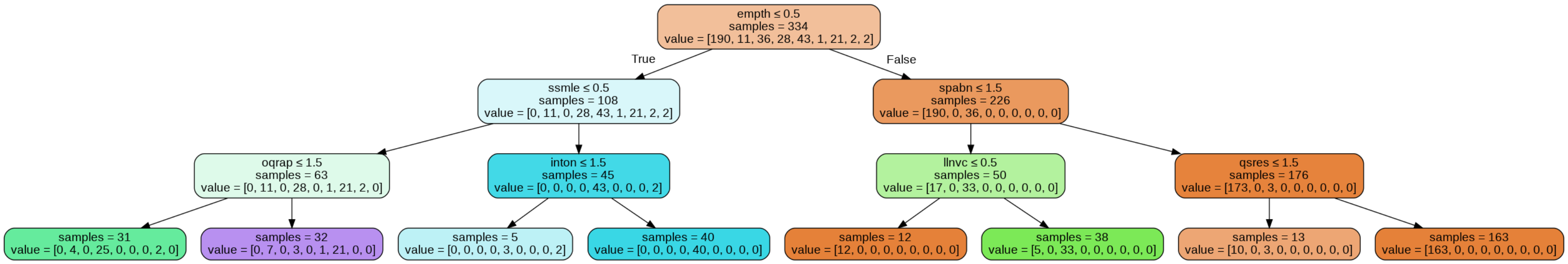

empth ≤ 0.5
samples = 334
value = [190, 11, 36, 28, 43, 1, 21, 2, 2]
True
False
ssmle ≤ 0.5
samples = 108
value = [0, 11, 0, 28, 43, 1, 21, 2, 2]
spabn ≤ 1.5
samples = 226
value = [190, 0, 36, 0, 0, 0, 0, 0, 0]
oqrap ≤ 1.5
samples = 63
value = [0, 11, 0, 28, 0, 1, 21, 2, 0]
inton ≤ 1.5
samples = 45
value = [0, 0, 0, 0, 43, 0, 0, 0, 2]
llnvc ≤ 0.5
samples = 50
value = [17, 0, 33, 0, 0, 0, 0, 0, 0]
qsres ≤ 1.5
samples = 176
value = [173, 0, 3, 0, 0, 0, 0, 0, 0]
samples = 31
value = [0, 4, 0, 25, 0, 0, 0, 2, 0]
samples = 32
value = [0, 7, 0, 3, 0, 1, 21, 0, 0]
samples = 5
value = [0, 0, 0, 0, 3, 0, 0, 0, 2]
samples = 40
value = [0, 0, 0, 0, 40, 0, 0, 0, 0]
samples = 12
value = [12, 0, 0, 0, 0, 0, 0, 0, 0]
samples = 38
value = [5, 0, 33, 0, 0, 0, 0, 0, 0]
samples = 13
value = [10, 0, 3, 0, 0, 0, 0, 0, 0]
samples = 163
value = [163, 0, 0, 0, 0, 0, 0, 0, 0]

Figure 18. Example of Decision tree used in an explanation for ADOS data

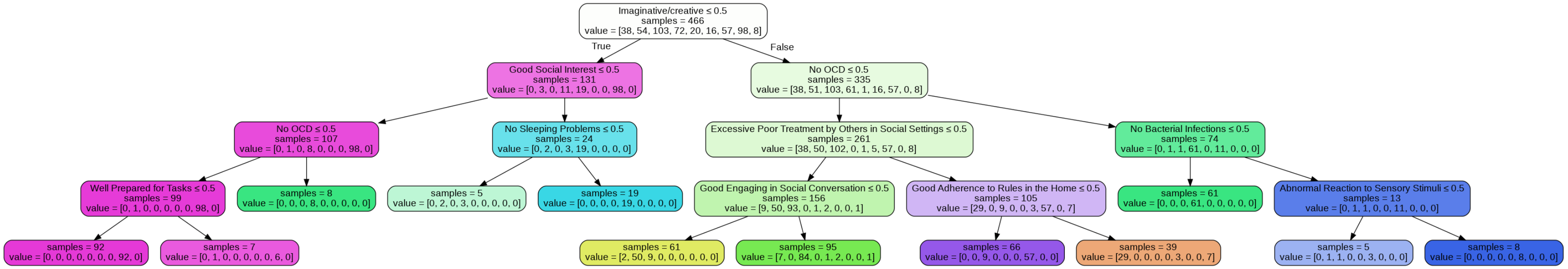

Imaginative/creative ≤ 0.5
samples = 466
value = [38, 54, 103, 72, 20, 16, 57, 98, 8]
True
False
Good Social Interest ≤ 0.5
samples = 131
value = [0, 3, 0, 11, 19, 0, 0, 98, 0]
No OCD ≤ 0.5
samples = 335
value = [38, 51, 103, 61, 1, 16, 57, 0, 8]
No OCD ≤ 0.5
samples = 107
value = [0, 1, 0, 8, 0, 0, 0, 98, 0]
No Sleeping Problems ≤ 0.5
samples = 24
value = [0, 2, 0, 3, 19, 0, 0, 0, 0]
Excessive Poor Treatment by Others in Social Settings ≤ 0.5
samples = 261
value = [38, 50, 102, 0, 1, 5, 57, 0, 8]
No Bacterial Infections ≤ 0.5
samples = 74
value = [0, 1, 1, 61, 0, 11, 0, 0, 0]
Well Prepared for Tasks ≤ 0.5
samples = 99
value = [0, 1, 0, 0, 0, 0, 0, 98, 0]
samples = 8
value = [0, 0, 0, 8, 0, 0, 0, 0, 0]
samples = 5
value = [0, 2, 0, 3, 0, 0, 0, 0, 0]
samples = 19
value = [0, 0, 0, 0, 19, 0, 0, 0, 0]
Good Engaging in Social Conversation ≤ 0.5
samples = 156
value = [9, 50, 93, 0, 1, 2, 0, 0, 1]
Good Adherence to Rules in the Home ≤ 0.5
samples = 105
value = [29, 0, 9, 0, 0, 3, 57, 0, 7]
samples = 61
value = [0, 0, 0, 61, 0, 0, 0, 0, 0]
Abnormal Reaction to Sensory Stimuli ≤ 0.5
samples = 13
value = [0, 1, 1, 0, 0, 11, 0, 0, 0]
samples = 92
value = [0, 0, 0, 0, 0, 0, 0, 92, 0]
samples = 7
value = [0, 1, 0, 0, 0, 0, 0, 6, 0]
samples = 61
value = [2, 50, 9, 0, 0, 0, 0, 0, 0]
samples = 95
value = [7, 0, 84, 0, 1, 2, 0, 0, 1]
samples = 66
value = [0, 0, 9, 0, 0, 0, 57, 0, 0]
samples = 39
value = [29, 0, 0, 0, 0, 3, 0, 0, 7]
samples = 5
value = [0, 1, 1, 0, 0, 3, 0, 0, 0]
samples = 8
value = [0, 0, 0, 0, 0, 8, 0, 0, 0]

Figure 19. Example of Decision tree used in an explanation for Concept Data

Cluster Label	**Number of subject**	**Communication + Social Interaction Total**	**Restricted and Repetitive Behavior Total**
0	268	**0 0.253**	
1	16	-0.337	-0.286
2	52	-0.438	-0.259
3	36	-0.332	-0.195
4	59	0.335	0.438
5	1		
6	32	0.3	0.152
7	4		
8	10	-0.211	-0.155

Table 3. The correlation between the clusters and the ADOS totals for the Agglomerative Clustering. Only significant correlations are shown ($p - value < 0.05$)

Q4: Relate the clusters to an external source of evidence about the input data

We computed the Pearson correlation between the data in Cluster A and the corresponding data in external sources which are the ADOS totals, the ADOS modules and the fMRI data. In this section, we are answering Q4 under evidence type E2 for output type O2 -see Table 2. The explanation results in this section are organized by the type of external data used as part of Evidence Type 2. For each external data used, we will present our explanation for the best clustering results found for the ADOS data and the Concept data.

Explanation results using ADOS Totals

ADOS Data explanation results

We calculated the Pearson correlation between the clusters and the ADOS totals: 'abtotal' stands for *Communication + Social Interaction Total*, 'adtotal' stands for

Cluster Label	**Number of subject**	**Communication + Social Interaction Total**	**Restricted and Repetitive Behavior Total**
0	74	**0.3**	0.185
1	53	0.352	0.421
2	34	-0.341	-0.205
3	16	-0.337	-0.286
4	50	-0.4	-0.257
5	77	0.224	
6	16	-0.17	
7	124	-0.171	
8	34	0.292	0.153

Table 4. The correlation between the clusters and the ADOS totals for the Spectral BiClustering of the ADOS data. Only significant correlations are shown ($p-value < 0.05$)

Restricted and Repetitive Behavior Total. Our goal is to find the correlation between the subjects within the same cluster and the type of the disorders. Table 3 shows that subjects in cluster 0 have communication and social interaction disorder. We show only the significant correlations that have p-value less than 0.05 ($p - value < 0.05$).

Table 4 shows that subjects belonging to cluster five, six, and seven have Communication and Social interaction disorders.

Concept Data explanation results

Table 5 shows that the subjects belonging to cluster 6 have Communication and Social interaction disorders. The subjects belonging to cluster 2 and cluster 5 have restricted and repetitive behavior disorders. Subjects belonging to cluster 7 have both communication and social interaction disorders, and restricted and repetitive

Cluster Label	Number of subject	Communication + Social Interaction Total	Restricted and Repetitive Behavior Total
0	52		
1	75		
2	142		-0.104
3	104		
4	36		
5	25		-0.097
6	83	0.23	
7	136	-0.159	-0.134
8	13		

Table 5. The correlation between the clusters and the ADOS totals for the Spectral BiClustering of the Concept data, Only significant correlations are shown ($p-value < 0.05$)

Cluster Label	Number of Subjects	Module t	Module 1	Module 2	Module 3	Module 4
0	268	0	0	0	183	85
1	16	16	0	0	0	0
2	52	0	0	0	22	30
3	36	0	0	36	0	0
4	59	0	59	0	0	0
5	1	0	0	0	1	0
6	32	0	0	32	0	0
7	4	0	0	2	1	1
8	10	0	10	0	0	0

Table 6. The distribution of clusters across the different ADOS modules using Agglomerative Clustering Algorithm

behavior disorders.

Explanation results using ADOS Modules

ADOS Data explanation results

Table 6 shows the distribution of each module in each cluster generated by the Agglomerative clustering algorithm. Table 6 shows that subjects belonging to module

Cluster Label	Number of Subjects	Module t	Module 1	Module 2	Module 3	Module 4
0	74	0	0	0	54	20
1	53	0	53	0	0	0
2	34	0	0	34	0	0
3	16	16	0	0	0	0
4	50	0	0	2	25	23
5	77	0	0	0	24	53
6	16	0	16	0	0	0
7	124	0	0	0	104	20
8	34	0	0	34	0	0

Table 7. The distribution of clusters across the different modules using Spectral Biclustering algorithm

t are assigned to the same cluster. The subjects belonging to module 1 are assigned to two different clusters. The subjects belonging the module 2 are mainly assigned to two different clusters. Cluster 0 and cluster 2 include subjects that belong to module 3 and subjects that belong to module 4.

Table 7 shows the distribution of each module in each cluster generated by the Spectral biclustering algorithm. The distribution of the modules across the clusters is similar to the results found using the agglomerative clustering. In fact, we have clusters that include subjects belonging only to module t (cluster 3), module 1 (cluster 1 and 6), or module 2 (cluster 2 and 8). Clusters 1, 5 and 7 include subjects belonging to module 3 and module 4 in the same cluster. Cluster 4 include subjects belonging to module 2, 3, and 4. It is the only cluster that includes more than two modules at the same time.

Cluster Label	**Brain Region**	**Pearson Correlation**
0	lh_isthmuscingulate_area	0.3
	lh_pericalcarine_area	0.3
	rh_caudalanteriorcingulate_area	0.3
2	lh_isthmuscingulate_area	-0.3
	lh_pericalcarine_area	-0.3
	rh_caudalanteriorcingulate_area	-0.3

Table 8. Mapping the clusters resulting from Hierarchical Agglomerative Clustering to the brain regions using fMRI data. Only significant correlations are shown ($p - value < 0.05$)

Explanation results using fMRI data

ADOS Data explanation results

In order to map our clusters to the brain regions, we used the brain fMRI data. We considered the overlap of the ADOS data and the fMRI data and calculated the correlation between the clusters and the brain region predictions. Significant correlations ($p - value < 0.05$) are shown in Table 8.

Table 9 shows the mapping to the brain regions for the obtained clusters.

Concept Data explanation results

Table 10 shows the mapping to the brain regions for the obtained clusters resulting from the spectral biclustering algorithm.

Cluster Label	**Brain Region**	**Pearson Correlation**
0	lh_bankssts_area	-0.267
	rh_entorhinal_area	-0.311
4	lh_inferiorparietal_area	-0.488
	lh_isthmuscingulate_area	-0.568
	lh_medialorbitofrontal_area	-0.433
	lh_parsorbitalis_area	-0.488
	rh_caudalanteriorcingulate_area	-0.568
	rh_paracentral_area	-0.391
	rh_postcentral_area	-0.359
	rh_rostralmiddlefrontal_area	-0.488
5	lh_parahippocampal_area	-0.488
	rh_fusiform_area	-0.324
7	rh_entorhinal_area	0.265
	rh_postcentral_area	0.258
	rh_rostralmiddlefrontal_area	0.265

Table 9. Mapping the clusters resulting from the Spectral Biclustering to the brain regions using fMRI data for ADOS. Only significant correlations are shown ($p-value < 0.05$)

Cluster Label	**Brain Region**	**Pearson Correlation**
0	lh_insula_area	-0.204
	rh_entorhinal_area	-0.196
	rh_parsorbitalis_area	-0.216
	rh_precentral_area	-0.169
	rh_rostralmiddlefrontal_area	-0.196
	rh_transversetemporal_area	-0.23
1	lh_rostralanteriorcingulate_area	-0.196
2	rh_paracentral_area	-0.188
	rh_insula_area	-0.183
4	lh_inferiortemporal_area	-0.204
	rh_lateralorbitofrontal_area	-0.273
	rh_middletemporal_area	-0.204
	rh_superiortemporal_area	-0.204
6	lh_caudalanteriorcingulate_area	-0.235
	lh_entorhinal_area	-0.218
	lh_parahippocampal_area	-0.177
	lh_paracentral_area	-0.214
	lh_postcentral_area	-0.177
	lh_supramarginal_area	-0.479
	lh_temporalpole_area	-0.177
	rh_bankssts_area	-0.177
	rh_fusiform_area	-0.177
	rh_supramarginal_area	-0.273
7	rh_precentral_area	0.171
	rh_precuneus_area	-0.194

Table 10. Mapping the clusters resulting from the Spectral Biclustering to the brain regions using fMRI data for Concept data. Only significant correlations are shown $(p - value < 0.05)$

CHAPTER V

CONCLUSIONS

In this thesis, we investigated different clustering algorithms to find homogeneous groups of patients that share similar ASD symptoms.

In addition, we presented a general framework to build an explainability module for clustering algorithms that assists the end-user in making sense of the clustering outputs through answering the end user's generic questions about the clustering results.

Through our clustering and explanation modules, our unsupervised machine learning methodology enables the domain experts to perform a powerful analysis on homogeneous cases, such as discovering hidden associations between the genetic data of patients belonging to the same cluster in order to have a better understanding of Autism Spectrum Disorder (ASD) and to pave the way toward data-driven personalized medicine.

Our results showed that Agglomerative clustering and bi-clustering had better performance in terms of their results for the ADOS data.

Our methods and findings may help doctors perform an early diagnosis of the disease and help them assign more specific treatments. Machine Learning is a powerful tool that enables the detection of hidden pattern and similarities beyond human understanding. For this reason, it can be a very helpful tool to facilitate discovery and disease analysis.

Our work has several limitations, such as the narrow choice of distance metrics despite the complexity and rich variety of data modalities. In ongoing and future work, we are developing semantic similarity measures to capture the meaning of the hierarchical ASD phenotype Concept data, and discover meaningful clusters. In addition, we plan to investigate the performance of additional clustering methods, such as deep clustering techniques.

REFERENCES

[1] J. Baio, L. Wiggins, D. L. Christensen, M. J. Maenner, J. Daniels, Z. Warren, M. Kurzius-Spencer, W. Zahorodny, C. Robinson, Rosenberg, T. White, M. S. Durkin, P. Imm, L. Nikolaou, M. Yeargin-Allsopp, L.-C. Lee, R. Harrington, M. Lopez, R. T. Fitzgerald, A. Hewitt, S. Pettygrove, J. N. Constantino, A. Vehorn, J. Shenouda, J. Hall-Lande, K. Van, Naarden, Braun, and N. F. Dowling, "Prevalence of autism spectrum disorder among children aged 8 years — autism and developmental disabilities monitoring network, 11 sites, united states, 2014," MMWR. Surveillance Summaries **67**, 1–23 (2018).

[2] B. A. McKinney, D. M. Reif, M. D. Ritchie, and J. H. Moore, "Machine learning for detecting gene-gene interactions: a review," Appl. Bioinformatics **5**, 77–88 (2006).

[3] L. Rokach and O. Maimon, *Clustering Methods* (Springer US, Boston, MA, 2005), pp. 321–352.

[4] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in "Proceedings of the Eighteenth International Conference on Machine Learning," (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001), ICML '01, pp. 577–584.

[5] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," Applied Statistics **28**, 100 (1979).

[6] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," Pattern Recognition **36**, 451 – 461 (2003). Biometrics.

[7] S. C. Johnson, "Hierarchical clustering schemes," Psychometrika **32**, 241–254 (1967).

[8] C. D. and, "Cluster merging and splitting in hierarchical clustering algorithms," in "2002 IEEE International Conference on Data Mining, 2002. Proceedings.", (2002), pp. 139–146.

[9] W. H. E. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," Journal of Classification **1**, 7–24 (1984).

[10] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," Data Mining and Knowledge Discovery **10**, 141–168 (2005).

[11] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**, 86–97 (2011).

[12] H. K. Seifoddini, “Single linkage versus average linkage clustering in machine cells formation applications,” Computers Industrial Engineering **16**, 419 – 426 (1989).

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in “Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic,” (MIT Press, Cambridge, MA, USA, 2001), NIPS’01, pp. 849–856.

[14] U. von Luxburg, “A tutorial on spectral clustering,” Statistics and Computing **17**, 395–416 (2007).

[15] Yu and Shi, “Multiclass spectral clustering,” in “Proceedings Ninth IEEE International Conference on Computer Vision,” (IEEE, 2003).

[16] S. White and P. Smyth, “A spectral clustering approach to finding communities in graphs,” in “Proceedings of the 2005 SIAM International Conference on Data Mining,” (Society for Industrial and Applied Mathematics, 2005).

[17] W. Dong, C. Moses, and K. Li, “Efficient k-nearest neighbor graph construction for generic similarity measures,” in “Proceedings of the 20th International Conference on World Wide Web,” (ACM, New York, NY, USA, 2011), WWW ’11, pp. 577–586.

[18] P. Franti, O. Virmajoki, and V. Hautamaki, “Fast agglomerative clustering using a k-nearest neighbor graph,” IEEE Transactions on Pattern Analysis and Machine Intelligence **28**, 1875–1881 (2006).

[19] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in “Advances in Neural Information Processing Systems 14,” , T. G. Dietterich, S. Becker, and Z. Ghahramani, eds. (MIT Press, 2002), pp. 585–591.

[20] T. Bühler and M. Hein, “Spectral clustering based on the graph p-laplacian,” in “Proceedings of the 26th Annual International Conference on Machine Learning,” (ACM, New York, NY, USA, 2009), ICML ’09, pp. 81–88.

[21] S. Madeira and A. Oliveira, “Biclustering algorithms for biological data analysis: a survey,” IEEE/ACM Transactions on Computational Biology and Bioinformatics **1**, 24–45 (2004).

[22] I. S. Dhillon, “Co-clustering documents and words using bipartite spectral graph partitioning,” (2001).

[23] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, “Spectral biclustering of microarray cancer data: Co-clustering genes and conditions,” Genome Research **13**, 703–716 (2003).

[24] X. an Bi, Y. Wang, Q. Shu, Q. Sun, and Q. Xu, "Classification of autism spectrum disorder using random support vector machine cluster," Frontiers in Genetics **9** (2018).

[25] M. N. Coutanche, S. L. Thompson-Schill, and R. T. Schultz, "Multi-voxel pattern analysis of fMRI data predicts clinical symptom severity," NeuroImage **57**, 113–123 (2011).

[26] A. D. Martino, C.-G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto, B. Deen, S. Delmonte, I. Dinstein, B. Ertl-Wagner, D. A. Fair, L. Gallagher, D. P. Kennedy, C. L. Keown, C. Keysers, J. E. Lainhart, C. Lord, B. Luna, V. Menon, N. J. Minshew, C. S. Monk, S. Mueller, R.-A. Müller, M. B. Nebel, J. T. Nigg, K. OHearn, K. A. Pelphrey, S. J. Peltier, J. D. Rudie, S. Sunaert, M. Thioux, J. M. Tyszka, L. Q. Uddin, J. S. Verhoeven, N. Wenderoth, J. L. Wiggins, S. H. Mostofsky, and M. P. Milham, "The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism," Molecular Psychiatry **19**, 659–667 (2013).

[27] M. V. Lombardo, , M.-C. Lai, B. Auyeung, R. J. Holt, C. Allison, P. Smith, B. Chakrabarti, A. N. V. Ruigrok, J. Suckling, E. T. Bullmore, A. J. Bailey, S. Baron-Cohen, P. F. Bolton, E. T. Bullmore, S. Carrington, M. Catani, B. Chakrabarti, M. C. Craig, E. M. Daly, S. C. L. Deoni, C. Ecker, F. Happé, J. Henty, P. Jezzard, P. Johnston, D. K. Jones, M.-C. Lai, M. V. Lombardo, A. Madden, D. Mullins, C. M. Murphy, D. G. M. Murphy, G. Pasco, A. N. V. Ruigrok, S. A. Sadek, D. Spain, R. Stewart, J. Suckling, S. J. Wheelwright, S. C. Williams, C. E. Wilson, C. Ecker, M. C. Craig, D. G. M. Murphy, F. Happé, and S. Baron-Cohen, "Unsupervised data-driven stratification of mentalizing heterogeneity in autism," Scientific Reports **6** (2016).

[28] T. Obafemi-Ajayi, D. Lam, T. N. Takahashi, S. Kanne, and D. Wunsch, "Sorting the phenotypic heterogeneity of autism spectrum disorders: A hierarchical clustering model," in "2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)," (IEEE, 2015).

[29] G. M. Anderson, F. Montazeri, and A. de Bildt, "Network approach to autistic traits: Group and subgroup analyses of ADOS item scores," Journal of Autism and Developmental Disorders **45**, 3115–3132 (2015).

[30] K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, T. N. Takahashi, S. Kanne, and D. Wunsch, "Ensemble statistical and subspace clustering model for analysis of autism spectrum disorder phenotypes," in "2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)," (IEEE, 2016).

[31] "About the ados exam," .

[32] "Nimh data archive -," .

[33] G. H. Glover, “Overview of functional magnetic resonance imaging,” Neurosurg. Clin. N. Am. **22**, 133–139 (2011).

[34] O. Dekhil, H. Hajjdiab, A. Shalaby, M. T. Ali, B. Ayinde, A. Switala, A. Elshamekh, M. Ghazal, R. Keynton, G. Barnes, and A. El-Baz, “Using resting state functional MRI to build a personalized autism diagnosis system,” PLoS ONE **13**, e0206351 (2018).

[35] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, “Understanding of internal clustering validation measures,” in “2010 IEEE International Conference on Data Mining,” (IEEE, 2010).

[36] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” Journal of Computational and Applied Mathematics **20**, 53–65 (1987).

[37] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-1**, 224–227 (1979).

[38] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, New York, NY, USA, 2008).

[39] “Evaluation of clustering,” .

CURRICULUM VITAE

NAME: Mariem Boujelbene

ADDRESS: Computer Engineering & Computer Science Department
Speed School of Engineering
University of Louisville
Louisville, KY 40292

EDUCATION:

M.Sc., Computer Science & Engineering
January 2018 - May 2019
University of Louisville, *Louisville, Kentucky*

B.Eng., Applied Mathematics
September 2014 - June 2017
Ecole Polytechnique de Tunisie, *Tunis, Tunisia*

Employment:

Research Assistant
April 2017 - Present
University of Louisville, *Louisville, Kentucky*

Data Scientist Intern
June 2016 - August 2016
Focus Digital, *Paris, France*

Software Engineer Intern
June 2015 - August 2015
Tunisie Telecom, *Tunis, Tunisia*

Achievements and Awards

- Dean Citation for academic excellence: April 2019
- CECS Arthur M. Riehl Award: April 2019
- Best Poster Award in the Regional Graduate Research Conference: February 2019