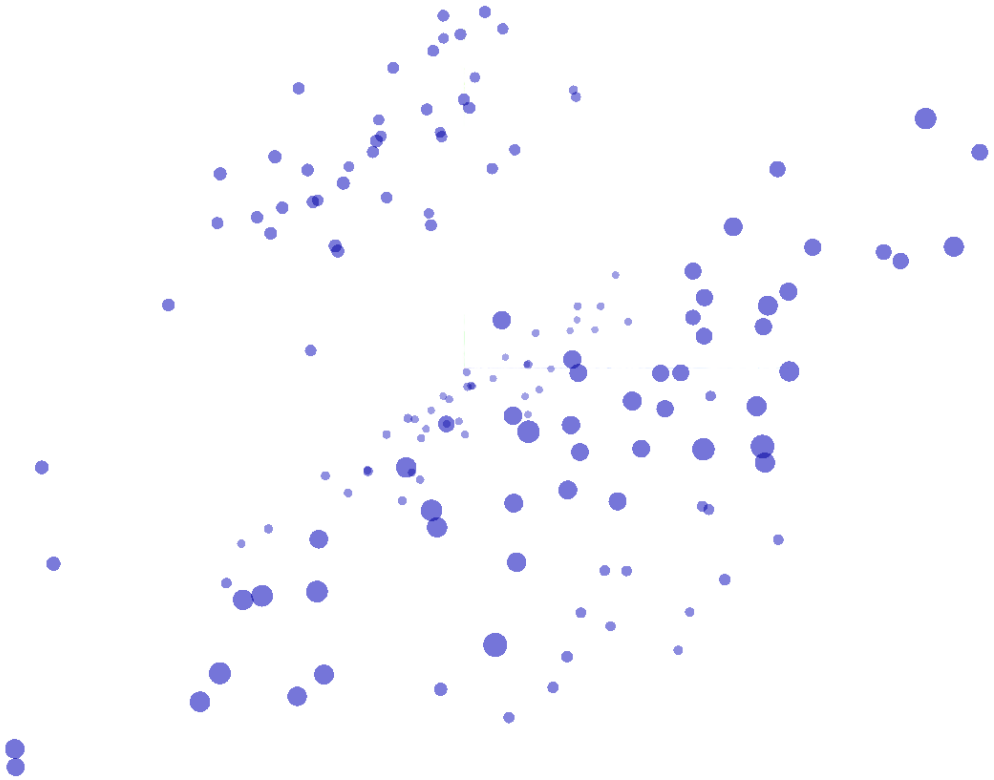
CHALMERS
UNIVERSITY OF TECHNOLOGY



Autonomous Gathering and Clustering of Behavioural Data in Virtual Reality

Master's thesis in Systems, Control and Mechatronics

JULIUS PETTERSSON
TOBIAS BERGSTRÖM

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Master's thesis 2018:33

Autonomous Gathering and Clustering of Behavioural Data in Virtual Reality

JULIUS PETTERSSON
TOBIAS BERGSTRÖM

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Systems & Control
Automation
Chalmers University of Technology
Gothenburg, Sweden 2018

Autonomous Gathering and Clustering of Behavioural Data in Virtual Reality
JULIUS PETTERSSON
TOBIAS BERGSTRÖM

Supervisor: Petter Falkman, Dept. of Electrical Engineering, Chalmers
Co-supervisor: Kerstin W. Falkman, Dept. of Psychology, University of Gothenburg
Examiner: Petter Falkman, Dept. of Electrical Engineering, Chalmers

Master's Thesis 2018:33
Department of Electrical Engineering
Division of Systems & Control
Automation
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A 3-D representation of the collected data.

Typeset in LaTeX
Gothenburg, Sweden 2018

Autonomous Gathering and Clustering of Behavioural Data in Virtual Reality
JULIUS PETTERSSON
TOBIAS BERGSTRÖM
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This thesis aims to investigate whether it is possible to find clusters in eye-tracking data collected during a psychological test in virtual reality. The test used for data collection was a subset of Raven's standard progressive matrices. The data was collected at two universities in Gothenburg and at Ågrenska Foundation. Participants were mainly young adults in higher education or adults visiting Ågrenska Foundation. The dataset was postprocessed to obtain a feature space, that acts as the base for the clustering, and normalized to reduce weighting errors between variables. The algorithms that were used for the clustering were a few unsupervised machine learning algorithms that are well established within the field. The clustering results from several of the applied algorithms indicate that there are indeed some underlying pattern in the data that corresponds to approximately six to eight clusters. The study shows the potential for use of advanced technology in psychological research. This does, however require further evaluation and development.

Keywords: eye-tracking, virtual reality, unsupervised machine learning, clustering, psychological testing, Raven's progressive matrices

Acknowledgements

We would like to thank Petter, our supervisor, for giving us the opportunity to work on this project as well as supplying us with the tools we needed. However, the most important parts for us has been the ability to discuss ideas, problems and the way forward through setbacks with him. Thank you Petter!

The psychological aspects of the project have been a real challenge, thankfully we had the expertise and guidance of Kerstin and Erland at the Department of Psychology, University of Gothenburg. Also, thank you Kerstin for the great help with reading and providing feedback on the report.

We would also like to thank Ågrenska Foundation for their interest in the project as well as for inviting us to test our developed system in collaboration with them.

At last we would like to give a shoutout to the members of "the Alliance Orchestra of Gothenburg" from Chalmers University of Technology who contributed with close to a third of our test participants.

Julius Pettersson & Tobias Bergström, Gothenburg, June 2018

Contents

List of Figures

List of Tables

1
Introduction

This chapter gives an insight into the different scientific fields that the thesis is built upon. It also gives a description of the purpose and aim of the project, as well as the problems and delimitations that have been considered.

1.1 Background

A psychological test is a standardized measurement of various aspects of human behaviour, such as intelligence or personality [3]. A challenge that comes with measuring psychological characteristics of an individual is that these cannot be observed directly; instead, it is necessary to make inferences about them through observable behaviour. This is based on the hypothesis that these behaviours correlate with an underlying psychological trait that is desirable to investigate.

There are different types of psychological tests; those that measure an individual's maximal performance (e.g. intelligence tests) and those that measure how a person usually behaves or reacts in a given situation (e.g. personality tests) [4]. Psychological tests are used in a variety of areas, for example in psychiatry for diagnosis, in recruitment to find the right person for a certain position, and in psychological research.

When performing tests, observations are often done through visual inspection as well as manual analysis of video recordings [5]. This is not efficient since it limits the amount of data that can be collected and the accuracy of the observations. However, the use of modern technology such as sensors and computer programs makes it possible to collect more data, at a higher accuracy and at a higher pace than previously. It is however difficult to analyze these large datasets, sometimes referred to as Big Data [6], using traditional methods.

Machine learning is on the other hand a tool that can be used to process these huge datasets. The area of machine learning has been around since the late 1950's [7] and started out as a way of achieving artificial intelligence. It was, however, reorganized and changed focus during the 1990's from achieving artificial intelligence using symbolic methods towards undertaking the task of solving practical problems using statistics and probability theory [8].

Previous research has shown that machine learning has potential within psychology to predict and increase our understanding of behaviour [9]. Furthermore, a study has shown that machine learning is efficient in facial recognition to determine facial expressions [10]. Consequently this could provide another parameter towards the purposes of analyzing an individual's behaviour since facial expressions

are closely tied to emotion [11]. Another study by [12] shows that both supervised and unsupervised artificial neural networks (ANNs) can be used to analyze how students perform on cognitive diagnostic assessments. It has also been shown in [13] that ANNs can be used to determine if a person has attention deficit hyperactivity disorder (ADHD).

Virtual Reality (VR) is another technology that has proved useful in psychology, for example as a tool to observe the level of distraction amongst children with ADHD [14, 15]. [16] highlights the possibilities and benefits of measuring data using VR such as accuracy, timing and consistency to enhance the analysis. The research in [17] shows that VR can be used to interact with children through facial emotions and expressions. It can also be of great use in the process of treating and rehabilitating arachnophobia [18].

The study of eye gaze movement is something that is already part of psychological research today. Vision is a complex and important sense that, according to [19], has developed in a different way for humans than most animals. The eyes contain multiple levels of information, for the sender as well as the receiver, about the environment, emotional and mental state [19]. Assessing eye movement through ET is already widely used today. One area is the gaming industry where ET can be integrated to enhance the immersion. It is also used for research purposes, in areas such as the theory of mind [20], diagnosing autism [21] and also as an assistive tool for people with movement difficulties [22].

Psychological research can also take place in a VR environment with the addition of automatic eye-tracking (ET) and data gathering [23]. This thesis will strive to further enhance that research.

1.2 Purpose and aim

The purpose of this thesis is to investigate if it is possible to collect and cluster behavioural data using virtual reality with eye-tracking.

Virtual reality could provide a new testing environment that is uniform, i.e an environment with equal conditions for everybody. The required equipment can be bought online and the software can be easily distributed over the Internet. An additional benefit comes from the fact that all the data collected will be stored digitally which enables the use of computer software, which can handle large sets of data, during the analysis process.

There is also a possibility that the automization of the tests could mean that a comprehensive education is no longer required to administrate the tests. However, a psychologist would still be needed to analyze the results. The test administrator would also need clear and detailed instructions to make sure that the test is carried out equally.

1.3 Problem description

The overall problem is to evaluate whether it is possible to cluster behavioural data that has been gathered from a virtual environment with eye-tracking and thereby

find potential underlying patterns. This is presented below through a set of sub-problems.

1.3.1 Development of a test scenario

The current test scenario that is presented in [23] must be extended before this test study to make it possible to collect additional information about the participants, such as age and gender. There will also be a need to implement additional ways of navigating the virtual environment since some of the test persons might not be able to use the current controllers.

1.3.2 Data collection

The success of machine learning is highly dependent on the amount of data that is available to train the algorithm and the thesis work will therefore involve a data collecting process. The test study will aim to collect data from at least 250 test participants and from a few different locations.

Having multiple locations will make it easier to obtain enough data through exposure towards more people. It will also serve as a way of increasing the chances to obtain a dataset with greater variety, as regards to variables such as age and educational background.

1.3.3 Postprocessing of the data

To be able to handle and make sense of the raw data that will be collected it needs to be postprocessed. This means that the raw data will be used to calculate information of interest which should be structured in a dataset ready to use for the clustering algorithms. The dataset has to be analyzed and since it will contain large amounts of information it will be to cumbersome to analyze manually. Thus a more efficient way to analyze the data will be required.

1.3.4 Clustering

Clustering is a way of finding patterns and similarities in data points with a large feature space without the need to label what the data represents. The feature space is a collection of all the features, i.e. pieces of information, describing a single data point. The features that are especially interesting will be decided upon and analyzed in collaboration with researchers from the Department of Psychology, University of Gothenburg.

The clustering should be performed using several machine learning algorithms as overlapping results will increase the validity of any patterns found within the dataset.

1.3.5 Analysis of the clustering

The results of the clustering process will have to be analyzed. This could be done through a comparison between results from different algorithms as well as through

experiments with the aim of finding the best settings for the individual algorithms.

1.4 Delimitations

The work in this thesis will not:

- strive to exactly replicate the test scenario such that the virtual one can be seen as an absolute equal to the real world version,
- try to perform automatic classification,
- focus on the in-depth development of the virtual environment.

2
Theory

The theory related to the four different areas; psychological testing, virtual reality, eye-tracking and data clustering will be presented in this chapter.

2.1 Psychological testing

In [24] the aim of a psychological test is described as a method to measure different abilities that are not easy to observe, such as intelligence, psychopathology or neuropsychology. Psychological tests are often standardized to ensure validity and reliability.

A psychological test is usually designed with a particular population in mind. An individual's result on the test is always presented in relation to this population, on an appropriate scale, for example IQ in cases where intelligence is measured. In a process called standardization, the test is used with a representative sample of the population [4]. From this group's mean values and variance, you then generate a function from raw points to the desired scale.

The reliability and validity of the test, i.e. if the same results are achieved as the measurements are performed multiple times and how well it measures what it intends to measure [4], also has to be calculated. One way to ensure reliability is to standardize the test procedure, for example making sure that the instructions given to the test person are always the same and that the environment in which the test is performed is the same [4], i.e. there are no external interference.

Another key element is to inform the participants about the premise of the testing and what their information will be used for to make them feel comfortable before giving their consent to participate [4]. There are additional factors, described by [4], that might affect the test results and/or the behaviour of the individual being tested such as anxiety, difficulties to concentrate or to communicate properly.

When collecting data for psychological research through the use of psychological testing this is mostly done manually. This means that researchers are often limited in the amount and types of data that can be collected. Observations of behaviour are, for example, made in real-time or through watching video recordings [5] of the test participant.

2.1.1 Raven's Progressive Matrices

The Raven's Progressive Matrices (RPM) [25] tests is used to measure general cognitive ability, i.e. eductive ability which is described by [25] as meaning-making.

They are well known and widely used since they are easy to administer and to interpret in a clear way [26]. The RPM are graphically easy to implement in a virtual environment, and are thus well suited to implement in VR.

The test consists of five different sets where each set contains a number of items. Each set follows a different logic that progressively increases in difficulty [25] with each set becoming more difficult than the previous. Each item has a logical pattern where one piece is missing. The goal is to select the correct alternative amongst a given set of alternatives, which varies from six to eight depending on the item. An example of how these items may look can be seen in Figure 2.1. These tests are the basis of the psychological testing that takes place in [23].

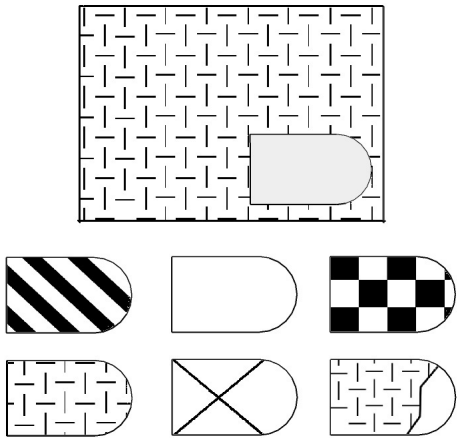


Figure 2.1: A figure showing an example item from Raven's standard progressive matrices.

2.2 Virtual reality

The technology of virtual reality (VR) is designed to deliver visual, audible and haptic stimuli [27] to the user and through that enable the user to become fully immersed within the virtual environment (VE). A way of achieving this is through the use of a head mounted display (HMD) which is a device fitted with a display that presents the VE to the user. The HMD is, in addition to this, equipped with multiple sensors that measures the head motions [27] of the user to make these transferable to the VE.

The first head mounted display (HMD) to be constructed was as early as 1968 by Ivan Sutherland [28] with the purpose to achieve greater immersion in a virtual environment. The VR technology has been vastly improved [29] since then and entered many new scientific fields.

2.3 Eye-tracking

ET is defined by [30] as the technique of measuring what a person is looking at, in what order the objects are gazed upon and for how long the eye gaze stays fixed on

that object. These measurements are interesting from a psychological perspective, it can for example provide information about the underlying neurophysiology of a person [31]. It can also give an insight into the individual's problem solving, reasoning and search strategies [30].

One way of tracking the eyes, as described in [30], is achieved by illuminating them with infrared light, which is used to prevent the user from being dazzled, to get a clear reflection that is captured using a camera. The reflections are then used to calculate a vector of the relationship between the cornea and pupil, which is then used to calculate the gaze direction.

2.4 Data clustering

Clustering is a way of grouping data points with multiple features together using unsupervised machine learning algorithms [32]. The goal of this process is, according to [32], often to retrieve information on underlying patterns or to group data into categories without the use of labels that describe the data. It is also important to consider the following problems, mentioned by [32], when performing cluster analysis; the representation of the data affects the outcome, the number of clusters in the data might vary between different approaches and the algorithms that are used might display clusters even though there are none. The following subsections will present how the initial dataset and its features could be selected and then a few algorithms with the potential to cluster data with large feature space.

2.4.1 Building the dataset and the feature space

The success of clustering unlabeled data, i.e. raw data that lacks a description of how to group it, is highly dependent on the data that is presented to the algorithm [32]. It is therefore, according to [32], important to choose features such that they represent the data from as unique aspects as possible. This is not a trivial task and there is no correct way to do it that always works. [32] however, suggests that domain knowledge should be used to decide upon these features. Since the scale of the numerical values from different features may vary a lot it might be necessary to normalize the data. This could potentially reduce weighting issues during the clustering process.

A common phenomenon that might be present when working with high-dimensional data, i.e. data with a large feature space, is the "curse of dimensionality" [33]. This means that various problems may arise when analyzing the data due to different factors. This could for example be that as the feature space increases, the spatial volume increases so fast that data becomes sparse. E.g. two data points that are close to each other in two dimensions might be largely separated as a third dimension is added. This means that a small increase in the number of features usually requires a large increase in the number of data points to maintain the same level of clustering performance [33].

2.4.2 Unsupervised clustering algorithms

This section will describe the theories for a number of unsupervised clustering algorithms that meet the requirements needed for this study. The algorithms do not process the data itself, they rather try to group it together in different manners with regards to similarities in the features of the data. It is not always possible to visualize these datasets with large feature spaces in two or three dimensions such that it makes sense to a human.

2.4.2.1 t-Distributed Stochastic Neighbour Embedding

t-Distributed Stochastic Neighbour Embedding (t-SNE) is an algorithm that is used to visualize data with large feature space in 2D or 3D [34]. How it works is that it constructs joint probabilities between data points and attempts to minimize the Kullback-Leibler divergence [35], which is a measure of the dissimilarity between two probabilities in the small feature space and the large feature space of the data.

There are two parameters of which t-SNE may be externally affected. The first is the `learning rate`. [34] states that if it is set too high, the result may look like a ball where all data points are equidistant from its neighbouring data points. If it is set too low, all data points may appear compressed into a very dense cluster. The second parameter is the `perplexity` which can be seen as a smooth measure of the effective number of neighbours. It is not very critical since the performance of t-SNE is quite robust for changes in the perplexity, according to [34].

The t-SNE algorithm is not without weaknesses however. The cost function of the algorithm is not convex, which according to [34] means that if the algorithm is initiated differently with the same data, the results will be different. However, [34] states that this is negligible since it will have a minor affect when it is run for a long period of time. Neither is the algorithm guaranteed to converge to a global optimum of the cost function [34]. As many other algorithms that work with data with large feature space, it may still suffer from the curse of dimensionality.

2.4.2.2 Self-Organizing Maps

The Self-Organizing Map (SOM) was developed by Teuvo Kohonen [36] and it is an algorithm that can be used to visualize and cluster data with a large feature space. The algorithm is, according to [36], able to transform advanced nonlinear statistical connections amongst data points with many features into simpler 2D patterns that are more easily displayed. [36] argues that this compression of information is a kind of abstraction that keeps the key elements, topological and metric features, of the primary data set intact.

The main design parameter that one needs to consider when working with SOMs is the size of the map, i.e. the number of nodes in the 2D-grid that forms the map [37]. This should, according to [37], be chosen such that the algorithm is able to find the entire pattern of the input.

2.4.2.3 Density-Based Spatial Clustering of Applications with Noise

The DBSCAN algorithm was proposed back in 1996 [2], and the clustering method is based on looking at the density of the points, creating clusters from the groups of points that are densely packed together. There are two design parameters which the algorithm needs, ϵ which is the radius of each point and `min samples` which is the minimum number of points within the radius of the current point of interest (including itself) to classify it as a core point to form a cluster, described by [2]. The fundamental process of the algorithm is illustrated and described in Figure 2.2.

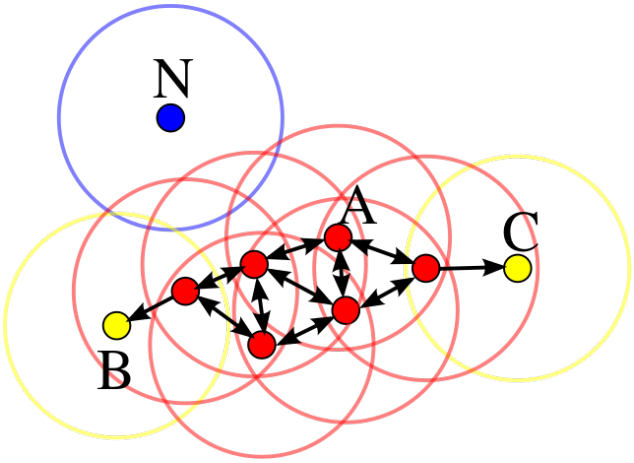


Figure 2.2: Illustration of the DBSCAN clustering from [1]. The parameter $minsamples = 4$, and the red points are core points since each has at least 4 points within radius ϵ, including itself. These points will form a cluster since they all are reachable to each another. Points B and C are not core points since they do not have at least 4 points within their respective ϵ radius, however they are still reachable from point A and thus belong to the cluster. Point N is classified as noise since it is neither a core point nor is reachable from any other point [2].

2.4.2.4 Mean-shift

The Mean-shift algorithm was proposed by K. Fukunaga and L. Hostetler back in 1975 [38]. The idea of the algorithm is to estimate the kernel distribution for a set of data, meaning that each data point gets a kernel (weight) and then all the kernels are added together, creating a density function. Then with the density function the algorithm assigns the data points to the nearest density center, generating clusters as described in [38]. A visualization of an example from performing the Mean-shift algorithm is shown in Figure 2.3.

2.4.2.5 Affinity propagation

Unlike previous algorithms, this algorithm sends messages between pairs of samples to find suitable exemplars (number of samples) which will represent the other samples, as proposed by B. J. Frey and D. Dueck in [39]. The messages sent between the pairs is used to check whether one of the two samples is a suitable exemplar of

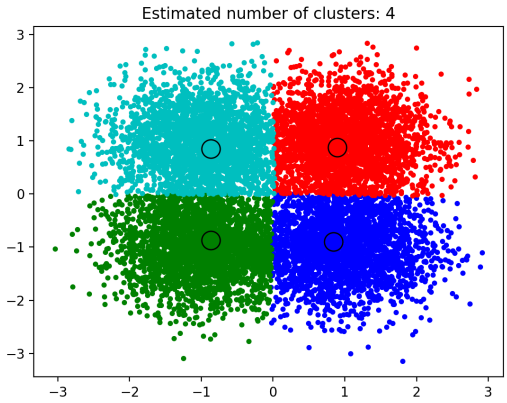


Figure 2.3: An illustration of the final clustering from the Mean-shift algorithm. In this example the samples are Gaussian distributed with four different centres $c_1 = (1, -1)$, $c_2 = (-1, 1)$, $c_3 = (1, 1)$ and $c_4 = (-1, -1)$. The total number of samples generated are $n = 10,000$.

the other, and how well-suited the chosen exemplar is to the sample not chosen, with the support from other samples that has chosen the same exemplar. This process works iteratively and updates until it converges and the final exemplars have been chosen as described in [39], at which point a final clustering is presented. A visualization of an example from performing the Affinity Propagation algorithm is shown in Figure 2.4.

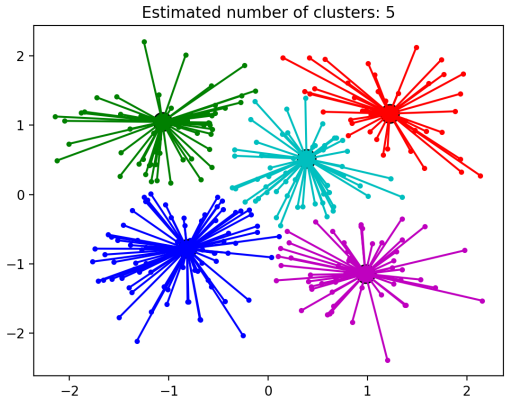


Figure 2.4: An illustration of the final clustering from the Affinity Propagation algorithm. In this example the samples are Gaussian distributed with five different centres $c_1 = (0, 0)$, $c_2 = (-1, 1)$, $c_3 = (1, 1)$, $c_4 = (-1, -1)$ and $c_5 = (1, -1)$. The total number of samples generated are $n = 300$.

2.5 Cluster validation

Analyzing and validating the resulting clusters can be done through various methods depending on how much that is known about the data. For example [40]:

- Internal validation: Based on the internal information known about the data.
- External validation: Based on the previously known information about the data.

Since the objective of this thesis is to investigate if there are any underlying patterns in behavioural data without any previous knowledge, i.e. the data being unlabeled, the approach of internal validation is the appropriate one of these two methods.

2.5.1 Silhouette

The silhouette [41] validation method is a measure of the total compactness of the data and how separated the clusters are. For each data point i, $a(i)$ is defined as the average distance of i within the cluster to the other data points, i.e. a measure for how well assigned i is to its cluster. $b(i)$ is the lowest average distance that data point i is to the remainder of the data points, i.e. the data points of the other clusters. The method is defined as

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}. \tag{2.1}$$

It gives a value within the range $-1 \leq s(i) \leq 1$, where a score of -1 is considered a poor clustering, 1 is considered a great clustering and values close to 0 are considered to have clusters that are not well separated.

2.5.2 Calinski-Harabasz

The Calinski-Harabasz [42] validates the clustering based on the average compactness within the clusters and the separation of the clusters. $s(i)$ is defined as the validated score, where i is the cluster index, B_i is the separation between the clusters, W_i is the compactness of the cluster and N is the number of data points.

$$s(i) = \frac{trace(B_i)}{trace(W_i)} \cdot \frac{N - i}{i - 1} \tag{2.2}$$

The score from this validation is considered to be better the higher the value is.

2.5.3 Davies-Bouldin

The Davies-Bouldin [43] is a validation method that is defined as the ratio $R(S_i, S_j, M_{i,j})$ between the cluster compactness and separation. Here $M_{i,j}$ is the separation between clusters i and j, and S_i is the compactness of the data points within cluster i. The method is defined by the constraints as

$$R(S_i, S_j, M_{i,j}) \geq 0, \tag{2.3}$$

$$R(S_i, S_j, M_{i,j}) = R(S_j, S_i, M_{j,i}), \tag{2.4}$$

$$R(S_i, S_j, M_{i,j}) = 0 \text{ iff } S_i = S_j = 0, \tag{2.5}$$

$$\text{if } S_j = S_k \text{ and } M_{i,j} < M_{i,k} \text{ then } R(S_i, S_j, M_{i,j}) > R(S_i, S_k, M_{i,k}), \tag{2.6}$$

$$\text{if } M_{i,j} = M_{i,k} \text{ and } S_j > S_k \text{ then } R(S_i, S_j, M_{i,j}) > R(S_i, S_k, M_{i,k}). \tag{2.7}$$

The total score of the clustering is then calculated as

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}}, \tag{2.8}$$

and will give a value that is always positive, where a low value means a better clustering.

3
Method

The methods and tools that were used to obtain the results will be presented in this chapter. This includes the hardware that the system was built around, the VR environment that the thesis was built upon and the data collection that was carried out to gather the necessary data. The selection of features, postprocessing of the data and implementation of the algorithms will also be covered.

3.1 Selection of psychological test

The reasons that the Raven's progressive matrices was chosen as the test to use in this thesis are:

- it is non-verbal which reduces the risk for interpretation errors when collecting the answers,
- it is built upon simple geometries and textures which makes it easy to convert to VR,
- the test is well established, well known and accessible.

The Raven's standard progressive matrices usually contains 60 items but in this thesis only ten will be used. These have been somewhat randomly selected but with the intention to have at least one from each level of difficulty (A-E). The results from this thesis can therefore not be used for psychological research purposes, but it serves well as a basis for method development and to evaluate the hypothesis that it is possible to identify behavioural patterns and to be able to cluster these.

3.2 The VR- and ET-hardware

The hardware that was used in this thesis is the "*Tobii Pro VR Integration*". It consists of a head mounted display (HMD) and two hand-held controllers. This VR-kit has an extra built-in feature, i.e. the Tobii ET that was utilized during the thesis. A software development kit is included to aid the development of programs for the ET. A full specification for the "*Tobii Pro VR Integration*" can be found at [44].

3.3 The VR environment used for data collection

The VR environment used in [23] was built using a platform called "*Unity*" [45] which supports 3D, VR and ET. "*Unity*" worked well as a tool for this purpose and

will therefor be used in this thesis for further development of the VR-environment used. The software applications in [23] were written in the programming languages C# and Python whereas the plots were made in MATLAB, and because we want to continue developing the software applications from [23] these will stay the same in the this thesis.

The data parameters that were collected in [23] and that were used in this thesis are:

- ET data such as the direction of the eye gaze and what objects that are gazed upon,
- the position and rotation of the HMD as well as both of the controllers,
- the completion time and the test person's test score.

3.4 The test study used for data collection

The data points in the dataset were collected at two demonstration sessions at the Department of Psychology, University of Gothenburg and at one technical fair at Chalmers University of Technology. There has also been some testing of people that was recruited through direct messaging and face-to-face interaction.

A week-long test study at the Ågrenska Foundation was also performed. This was carried out by an employee who was taught to use the entire system without the presence of the authors of this thesis.

The participants at University of Gothenburg were a mixture of students, teachers and visiting high-school students whereas there were mainly students participating at Chalmers University of Technology. The participants from Ågrenska Foundation were mainly personnel and volunteer workers.

3.4.1 Procedure for data collection

The test procedure at the two universities involved setting up the equipment and the informational poster in an open area where people in general pass through. Information about the thesis work and the test was then given to groups of people that stopped by. These areas were often quite crowded with a lot of background noise.

The equipment and the poster have also been set up in the authors office during most part of the project. The office is a smaller and quieter space compared to the open areas at the universities. This is where people that have been directly contacted have come to do the test.

The environment at Ågrenska Foundation was very similar to the one in the office. The tests were carried out in a smaller, separate room with little disturbances.

3.4.2 Instructions given to participants

The test participant is first of all given a brief explanation of what the thesis work is about and the purpose of the data collection. After that there are a series of steps that the test instructor walks through to aid the participant through the test. These are described below:

1. The participant is told to put on the HMD with their eyes centered in the middle of the lenses and adjust the fit with the screw on the back of the headset.
2. The instructor hands over the hand controller.
3. The participant is asked to select the most suitable language, either Swedish or English, using the laser pointer and the touchpad on the controller.
4. The participant is told to stand still and just use eye gaze to complete the calibration step.
5. The instructor informs that the selection of objects in the VE is now done using eye gaze, but the final choice is still acknowledged using the touchpad on the controller.
6. The participant is instructed to complete the information form that is displayed in the VE and that the actual test will begin after that.
7. The instructor is standing in close proximity of the participant in case he or she has additional questions.

3.5 The obtained dataset

The dataset consists of 166 unique data points which have been collected during the course of six months. The gathering of data has taken place, for the most part, at the two universities mentioned in Section 3.4, which resulted in a dataset with a majority of younger adults that are studying at higher level education. The age distribution for the dataset is displayed in Figure 3.1 and ranges from the youngest being 17 and the oldest 70 years old. The gender division amongst the participants is 36.1% female, 63.9% male and 0% other.

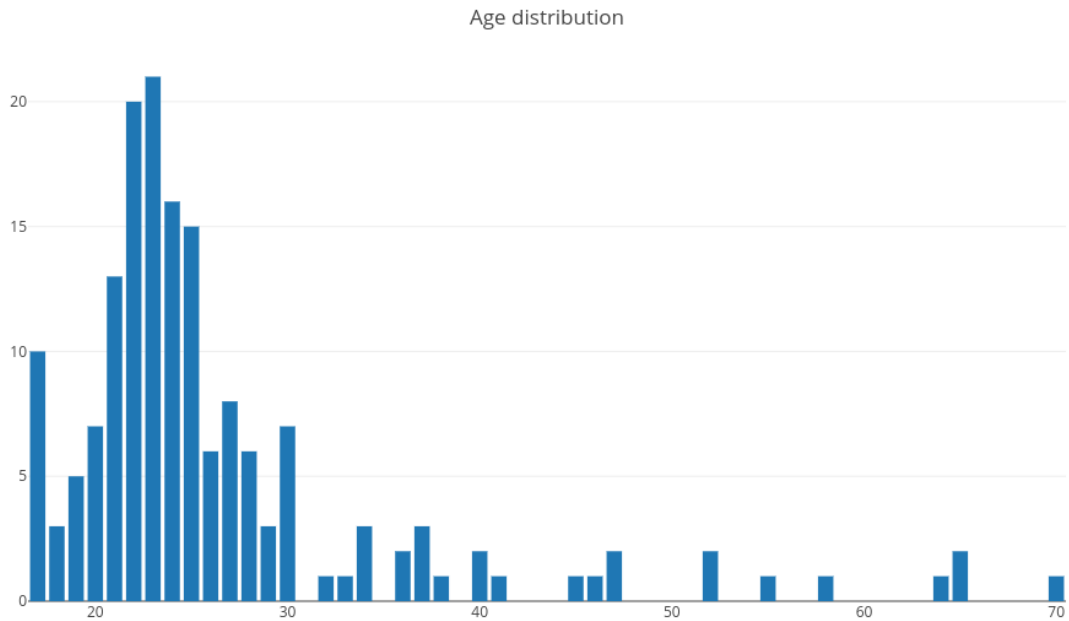


Figure 3.1: Plot that shows the age distribution of the dataset.

3.6 Selection of features for data clustering

The feature space that was used in the data clustering, obtained from the raw data, was decided upon in dialogue with researchers from the Department of Psychology at the University of Gothenburg. It is also somewhat restricted by the information that can be extracted from the model.

After a brief analysis of the dataset it could be seen that many participants spent a lot of time and looked around quite a bit during the first item of the test. This is not concurrent with the difficulty of Item 1 but can most likely be described as a consequence of the participants unfamiliarity with VR and ET. It was therefore decided that Item 1 should be seen as a training item and it has for that reason been excluded from the data clustering.

3.7 Selection of algorithms

The clustering algorithms were selected based on a set of criteria defined below:

- the algorithm must operate unsupervised and without knowing the number of clusters that the dataset should have since there is no prior information available about the division of the dataset,
- the algorithm should be well established and have multiple cases of implementation in different studies,
- the algorithm should perform well on a broad spectra of datasets to improve the chances of working well with the data in the thesis.

Many algorithms that fit the first criterion were found during the research. Several of these algorithms described "state-of-the-art" performance on their specific datasets but were found nowhere else in the literature or lacked practical implementation. They were therefore discarded since they did not fulfill the second criterion and/or the third one.

The algorithms have in addition, from previously mentioned criteria, been selected based on how easy they seemed to implement in Python, i.e. if they were part of any of the more well known machine learning packages.

3.8 Implementation of the algorithms

Python was chosen as the programming language because it comes with several packages which offer a wide variety of machine learning algorithms.

The package *scikit-learn* contains implementations of several unsupervised clustering algorithms. *scikit-learn* is built to work cross-platform and is utilizing the central processing unit (CPU) of the device.

Another package is called *tensorflow*, which is a powerful tool that allows for efficient computations on data with a large feature space by utilizing the graphics processing unit (GPU). It also contains a user interface (UI) called *tensorboard* that gives the user the ability to analyze the algorithms built with *tensorflow*.

The data clustering will be done using the algorithms described in Section 2.4.2.

The programming language Python was also used for the work done in the project described in [23] to post-process the gathered data. The continued use of Python made the integration of the previous work into the software that was developed easier.

3.8.1 Algorithms from the tensorflow Python package

The t-SNE algorithm is available in the *tensorflow* package such that it may easily be opened in the *tensorboard* UI. The parameters of the algorithm can be changed, the algorithm can be run and the results can be plotted in real-time, either in two or three dimensions, all within the UI. This allowed for easy testing of multiple parameter combinations without interruptions. The UI also gives the user the ability to save the current state of the clustering as well as the final results to a file on the computer.

3.8.2 Algorithms from the scikit-learn Python package

The algorithms DBSCAN, Mean-shift and Affinity Propagation are all part of the python package *scikit-learn*. These were implemented and adapted to fit the data that was collected during the thesis work. The implementation features the ability to use all of the different algorithms with the data as well as the ability to compare the clustering results between them. It is also possible to tune the most important parameters for each specific algorithm. One such parameter is the metric for the DBSCAN which calculates the distances between the instances of features in the dataset. All of the viable metrics implemented in *scikit-learn* were tested to find the optimal one for the data used in this thesis. These metrics are; `cityblock`, `cosine`, `euclidean`, `l1`, `l2`, `braycurtis`, `canberra`, `chebyshev`, `correlation`, `jaccard`, `matching` and `sqeuclidean`.

3.8.3 Algorithms from MATLAB

The SOM algorithm was tested using the MATLAB software and the built-in function called `selforgmap`. This function includes the ability to adjust parameters such as; the size of the SOM, the number of iterations to run, the topology function and the distance function. It is also possible to plot the results of the clustering using specific SOM plotting functions.

3.9 Evaluation of the clustering

The implemented algorithms were tested on the data in several ways to determine which parameters that gave the most interesting results. This was done through a series of experiments, for each algorithm, that combined the parameters in different ways. The results from each experiment was compared both to the other results from the same algorithm but also to the ones from the other algorithms.

The internal validation methods that were described in Section 2.5 were used to validate the cluster results. The silhouette and Calinski-Harabasz methods are

already implemented and available through *scikit-learn*, while the Davies-Bouldin method was manually implemented using Python and the distance calculations that are available in the *scipy* package.

3.9.1 Calculating the overlap between clusters

The set of mutual members, $C_{i,j}$, between two clusters has been calculated as:

$$C_{i,j} = C_i \cap C_j,$$

where C_i is the set of members in the first cluster and C_j is the set of members in the second one. The overlap between clusters of similar size has then been calculated as:

$$\frac{2n_{C_{i,j}}}{n_{C_i} + n_{C_j}},$$

where $n_{C_{i,j}}$ is the size of $C_{i,j}$, n_{C_i} is the size of C_i and n_{C_j} is the size of C_j.

4
Results

This chapter will present the results from the work carried out in the thesis. That covers the improvements made to the VR environment that is used for the test, the dataset, the developed system as well as the results from the data clustering algorithms.

4.1 Improved test model

The test model that is described in Section 3.3 has been further improved. The entire test scenario is now following a linear structure to make it easier to understand and to complete without additional instructions. It has also been extended with an information form that collects data related to the test person such as; gender and age. No data making it possible to identify a single participant was, however, collected. In addition to this there is also a separation between each test item. This was done by removing the previous item and having a +-sign appear for the duration of 1.5 seconds before moving on to the next item. This serves the purpose of resetting the users attention to the middle of the screen. The entire process can be seen in the flow chart in Figure 4.1

4.2 Selected features for data clustering

The feature space has been divided into two different categories; global features and item specific features. The global features consist of the individual's information, and these are ID, age and gender. The ID is the anonymous name-tag that has been generated for each participant. The ID is however not used in the clustering algorithms for anything else than identifying the members of the obtained clusters to determine if there are similarities between clusters from the different runs.

The item specific features consists of the information that has been gathered from each participant for each item during the test. Most of the item specific features are the same for all items and are therefore reoccurring ten times. These features can be seen in Table 4.1 in the colon called "All Items". Item 7 to 10 contains two additional features and another version of the feature called "Provided answer" which is due to the fact that these items contain eight alternatives instead of six.

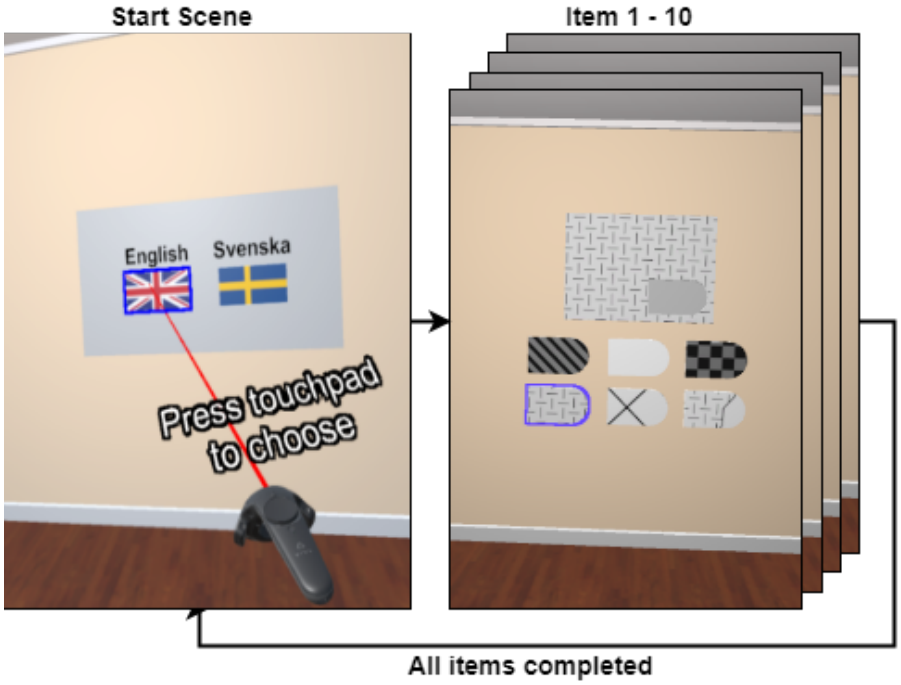


Figure 4.1: The updated VR environment model, starting with a main menu from which you have the option to choose language between English and Swedish before proceeding to the calibration and information form steps before the test sequence starts.

4.2.1 Calculating and normalizing the features

All the objects within the virtual environment are uniquely tagged as different areas of interest(AOI) such as; the pattern to solve as "Board" and all the individual alternatives as "Alt. 1-6/1-8". The interior of the room, i.e. walls, floor and ceiling, are tagged as "Not AOI". The features "Sample inside - <tag>" are calculated as the number of occurrences of a certain tag divided by the total number of samples that were collected during the duration of the item. "Number of changes in AOI" is a value that represents the number of times the participant changed focus between the different AOI described above.

The features that has string or boolean entries have to be converted into numerical values, which is required by the clustering algorithms. The numeric version of the dataset is then normalized feature-wise to reduce the effect of weighting issues in the algorithms related to large differences in scale between values from different features. The normalization is performed using the L_2-norm and the `normalize()` method from the preprocessing library of *scikit-learn*.

Table 4.1: The feature space that was decided upon and used during the testing of the algorithms. These features are what represents each participant.

Feature		
Global	**All Items**	**Item 7-10**
ID	Time	Samples inside - Alt. 7
Age	Answer correct (Yes/No)	Samples inside - Alt. 8
Gender	Provided answer (1-6)	Provided answer (1-8)
	Number of changes in AOI	
	Samples inside - Board	
	Samples inside - Not AOI	
	Samples inside - Alt. 1	
	Samples inside - Alt. 2	
	Samples inside - Alt. 3	
	Samples inside - Alt. 4	
	Samples inside - Alt. 5	
	Samples inside - Alt. 6	

4.3 The developed system

A software has been developed throughout the thesis with the aim to process the raw data, perform the clustering and to simplify the analysis process. The different components and their functions will be explained in this section. These components can be categorized as the handling of the data, the implemented algorithms and the graphical user interface (GUI) which connects these two.

4.3.1 Handling of the data

To be able to analyze the data that has been collected it will have to be presented in a intelligible way, and therefore tools to post-process the raw data were developed. In the project [23] some of the post-processing had already been developed, presenting the time and provided answer for each item during the test for each test participant. Slight modifications were made to these python scripts and more calculations were added to handle all the features that are shown in Table 4.1.

4.3.1.1 Generating the dataset

The dataset is compiled into one CSV-file using a developed script that automates the process of looking through each subdirectory for files containing raw data. The data that have been collected are stored in separate folders according to the date that the test was carried out to simplify the process of generating and updating the dataset. The script goes through the dates of these folders and adds the data that has been collected since the last time the dataset was updated, which is indicated by its filename, and then updates the filename to reflect the current date.

4.3.1.2 Handling the dataset

A data handler was developed to manage the dataset during run time and it provides methods which return specific information from the dataset, filtered by either specific test participants or specific features. The handler is also providing all the statistical information about the dataset.

4.3.1.3 Handling and comparison of clusters

The clusters that are found by the algorithms are stored in separate objects of a `Cluster`-class that was developed. This class contains methods that are used to compare different clusters to each other. The cluster objects are in turn stored in a `ClusterResult` which contains the functionality to compare and evaluate different runs of one or several of the algorithms.

4.3.2 Graphical user interface

The GUI was developed with the aim to aid the user in the analysis process. It therefore contains functions that provides the ability to:

- view statistics that summarizes the dataset as both text and graphs,
- view the dataset and subsets of the dataset,
- sort, filter and compare the data points with regards to the features in the feature space,
- test all the implemented clustering algorithms and adjust their respective parameters.

The GUI has been created in Python using the package called *dash* which allows for creation of graphical content using HTML components. The GUI is displayed in the users browser. In Figure 4.2 the summary page is displayed, in which some general statistics from the dataset are presented and illustrated, and it is the starting page when opening the GUI. More pages exist where all the algorithms can be run directly within the GUI as well as displaying the dataset.

A complete illustration of the flow in the library developed for the GUI can be seen in Figure 4.3. The green field describes the files and functions used to handle the GUI as well as the user input, while the red field describes the data handling. The blue field is the implementation of the algorithms described in Section 2.4.2.

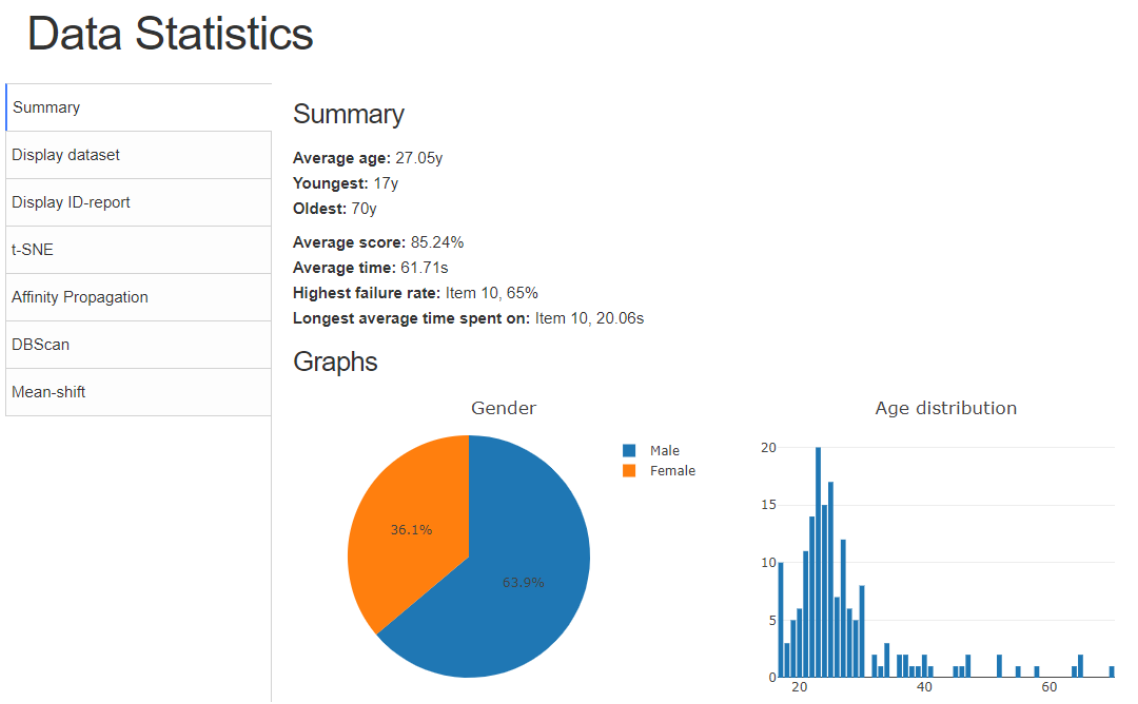


Figure 4.2: An overview of the GUI, representing the summary page.

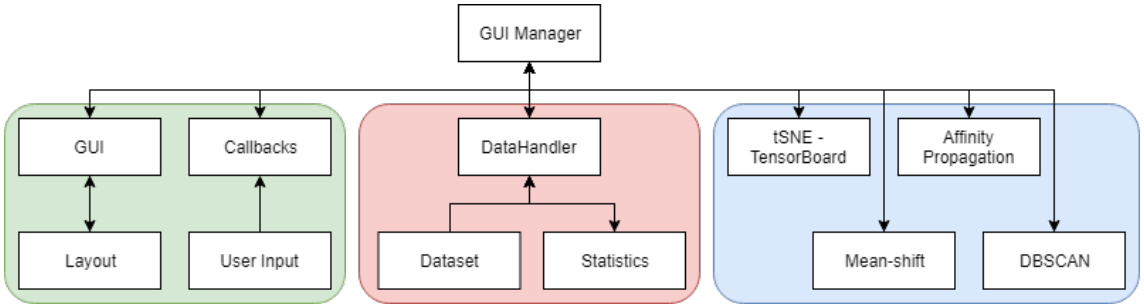


Figure 4.3: An illustration of the library flow of the GUI.

4.4 Clustering results

This section will present the results obtained from the clustering algorithms described in Section 2.4.2. The algorithms will be tested a number of times each with varying parameter settings.

The test that gave the best result from each of the algorithms that performed well, with regards to number of clusters and the cluster sizes, i.e. the number of data points within each cluster, are presented in Table 4.2. These were used to compare the results from all the algorithms and the comparisons are presented in Tables 4.3 through 4.8. The clusters from each of the chosen results have been sorted in descending order based on the cluster size. The overlap, described in Section 4.3.1.3, has been calculated after that and represented as a percentage of the mutual members in both clusters. If there are no mutual members between the

clusters or one algorithm found more clusters than the other (no comparison can be made), the comparisons will be presented as `N/A` (Not Available).

Table 4.2: Presentation of one good clustering result from each of the algorithms that performed well. The last three columns displays the cluster validations as S (silhouette), CH (Calinski-Harabasz) and DB (Davies-Bouldin).

			Clusters									**Validation**		
		Nr of cluster	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	S	CH	DB
Algorithm	Affinity Propagation	9	1	28	9	26	33	12	38	15	4	0.035	5.467	0.462
	DBSCAN	8	33	41	29	10	6	4	4	39	N/A	0.163	5.073	0.371
	t-SNE	7	67	32	25	28	6	4	4	N/A	N/A	0.01	5.022	0.512
	SOM	6	42	15	52	12	41	4	N/A	N/A	N/A	0.062	8.414	0.488

Table 4.3: Comparison of the Affinity Propagation and DBSCAN clusterings displaying the cluster sizes as well as the overlap.

	Algorithm			
	Affinity Propagation	DBSCAN	Sizes	Overlap [%]
Cluster	C_6	C_2	38 vs 29	77.6
	C_1	C_1	28 vs 41	69.6
	C_4	C_0	33 vs 33	63.6
	C_2	C_4	9 vs 6	53.3
	C_8	C_4	4 vs 6	40.0
	C_3	C_7	26 vs 39	18.5
	C_0	N/A	N/A	N/A
	C_5	N/A	N/A	N/A
	C_7	N/A	N/A	N/A

Table 4.4: Comparison of the SOM and DBSCAN clusterings displaying the cluster sizes as well as the overlap.

	Algorithm			
	SOM	DBSCAN	Sizes	Overlap [%]
Cluster	C_2	C_1	52 vs 41	88.2
	C_4	C_0	41 vs 33	86.5
	C_0	C_2	42 vs 29	81.7
	C_1	C_3	15 vs 10	80.0
	C_5	C_4	4 vs 6	20.0
	C_3	N/A	N/A	N/A

Table 4.5: Comparison of the t-SNE and DBSCAN clusterings displaying the cluster sizes as well as the overlap.

	Algorithm			
	t-SNE	DBSCAN	Sizes	Overlap [%]
Cluster	C_1	C_0	32 vs 33	64.6
	C_2	C_1	25 vs 41	51.5
	C_6	C_4	4 vs 6	40.0
	C_0	C_7	67 vs 39	37.7
	C_3	C_7	28 vs 39	32.8
	C_5	C_6	4 vs 4	25.0
	C_4	N/A	N/A	N/A

Table 4.6: Comparison of the t-SNE and SOM clusterings displaying the cluster sizes as well as the overlap.

	Algorithm			
	t-SNE	SOM	Sizes	Overlap [%]
Cluster	C_0	C_0	67 vs 42	75.2
	C_1	C_4	32 vs 41	65.8
	C_3	C_2	28 vs 52	45.0
	C_2	C_1	25 vs 15	35.0
	C_4	N/A	N/A	N/A
	C_5	N/A	N/A	N/A
	C_6	N/A	N/A	N/A

Table 4.7: Comparison of the Affinity Propagation and SOM clusterings displaying the cluster sizes as well as the overlap.

	Algorithm			
	Affinity Propagation	SOM	Sizes	Overlap [%]
Cluster	C_6	C_0	38 vs 42	95.0
	C_3	C_1	26 vs 15	73.2
	C_1	C_2	28 vs 52	70.0
	C_2	C_3	9 vs 12	66.7
	C_4	C_4	33 vs 41	64.9
	C_7	C_3	15 vs 12	7.4
	C_0	N/A	N/A	N/A
	C_5	N/A	N/A	N/A
	C_8	N/A	N/A	N/A

Table 4.8: Comparison of the Affinity Propagation and t-SNE clusterings displaying the cluster sizes as well as the overlap.

	Algorithm			
	Affinity Propagation	t-SNE	Sizes	Overlap [%]
Cluster	C_6	C_0	38 vs 67	70.5
	C_4	C_1	33 vs 32	49.2
	C_3	C_2	26 vs 25	43.1
	C_1	C_2	28 vs 25	41.5
	C_8	C_6	4 vs 4	25.0
	C_7	C_3	15 vs 28	14.0
	C_0	N/A	N/A	N/A
	C_2	N/A	N/A	N/A
	C_5	N/A	N/A	N/A

4.4.1 Results from t-SNE

The t-SNE has been tested with different combinations of the parameters `perplexity` and `learning rate`. The results from these tests can be seen in Table 4.9. Working with the t-SNE proved more challenging than previously assumed, mostly due to the fact that the algorithm had to run for quite some time before a convergence might be assumed. Then, if the clustering results were poor, the parameters had to be tweaked slightly and run again.

Another factor that might have affected the results could be that the cost function is not convex, which would give different results if the dataset is initiated differently. However, it would probably have had a minor affect if at all during these tests, as described in 2.4.2.1, since they were run for a longer period of time where no differences were found when run with the same parameters.

Table 4.9: Table presenting the top results from using the t-SNE algorithm with different parameter values. Each test shows the total number of clusters, the parameter values and the number of samples in each cluster. The last three columns displays the cluster validations as S (silhouette), CH (Calinski-Harabasz) and DB (Davies-Bouldin).

			Parameters		**Clusters**							**Validation**		
		Nr of cluster	Perplexity	Learning rate	C_0	C_1	C_2	C_3	C_4	C_5	C_6	S	CH	DB
Test	#1	7	15	10	101	19	28	9	4	3	2	−0.005	4.097	0.462
	#2	7	14	0.1	67	32	25	28	6	4	4	0.01	5.022	0.512
	#3	6	13	10	92	30	26	11	3	4	N/A	0.019	4.844	0.542
	#4	6	14	10	111	25	19	7	2	2	N/A	0.002	4.111	0.547

4.4.2 Results from Affinity Propagation

The Affinity Propagation has been tested with different values for the `damping` parameter, as shown in Table 4.10. While the algorithm was able to produce several clustering results, only two were were considered good. Slight adjustments to the damping parameter made the algorithm perform considerably worse and it started

dropping data points from the clusters, leading to a lot of data points being labeled as noise. These results are not displayed in the table since the algorithm just produced 10-20 clusters of size one and reduced the other ones a bit.

Table 4.10: Table presenting the top results from using the Affinity Propagation algorithm with different parameter values. Each test shows the total number of clusters, the parameter values and the number of samples in each cluster. The last three columns displays the cluster validations as S (silhouette), CH (Calinski-Harabasz) and DB (Davies-Bouldin).

		Parameters		**Clusters**									**Validation**		
		Nr of cluster	Damping	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	S	CH	DB
Test	#1	9	0.99	1	28	9	26	33	12	38	15	4	0.035	5.467	0.462
	#2	9	0.94	1	28	9	26	33	12	38	15	4	0.035	5.467	0.462

4.4.3 Results from DBSCAN

The DBSCAN was tested and the results are shown in Table 4.11. Working with DBSCAN was quite different from the others since it has three parameters all of which may have great influence on the result. The implementation of the algorithm was designed to try all combinations through the use of nested loops that covered a range of values for the parameters of `epsilon` and `min sample` while also going through a list of viable metrics mentioned in Section 3.8.2. The last cluster from every run represents the data that has been labeled as noise by the algorithm.

Table 4.11: Table presenting the top results from using the DBSCAN algorithm with different parameter values. Each test shows the total number of clusters, the parameter values and the number of samples in each cluster. The last three columns displays the cluster validations as S (silhouette), CH (Calinski-Harabasz) and DB (Davies-Bouldin).

		Parameters				**Clusters**									**Validation**		
		Nr of cluster	Metric	Epsilon	Min sample	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	S	CH	DB
Test	#1	9	Chebyshev	0.86	4	34	41	31	11	6	4	5	4	30	0.159	4.982	0.32
	#2	8	Chebyshev	0.85	4	33	41	29	10	6	4	4	39	N/A	0.163	5.073	0.371
	#3	3	Cosine	0.14	2	155	2	9	N/A	N/A	N/A	N/A	N/A	N/A	0.294	3.977	1.519
	#4	3	Jaccard	0.14	2	133	2	31	N/A	N/A	N/A	N/A	N/A	N/A	0.036	2.841	0.658
	#5	3	Bray-Curtis	0.25	2	158	2	6	N/A	N/A	N/A	N/A	N/A	N/A	0.23	4.495	0.735

4.4.4 Results from Mean-shift

The results from the testing of Mean-shift are shown in Table 4.12. Mean-shift was not able to cluster the collected data efficiently as it was unable to produce more than one large cluster with many smaller clusters by the size of one data point. The `bandwidth` parameter for the Mean-shift was estimated using the method `estimate_bandwidth(data, quantile=quantile)` from *sklearn.cluster* where the input that was varied through the tests was the quantile parameter. This parameter, however, only changed the size of the large cluster with the remainder of data points receiving their own clusters. Since the algorithm was unable to produce comparable results to the other algorithms, regardless of settings, no more than two runs are shown in Table 4.12.

Table 4.12: Table presenting the top results from using the Mean-shift algorithm with different parameter values. Each test shows the total number of clusters, the parameter values and the number of samples in each cluster. The last three columns displays the cluster validations as S (silhouette), CH (Calinski-Harabasz) and DB (Davies-Bouldin).

		Parameters		**Clusters**										**Validation**		
		Nr of cluster	Quantile	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	S	CH	DB
Test	#1	10	0.5	157	1	1	1	1	1	1	1	1	1	0.184	2.892	0.064
	#2	5	0.8	162	1	1	1	1	N/A	N/A	N/A	N/A	N/A	0.24	3.196	0.12

4.4.5 Results from SOM

The SOM has been tested with different values for the parameters `map size`, `metric` and `iterations`. The experiments were carried out using nested loops to easily test all combinations. The results from the three best experiments are shown in Table 4.13. The last cluster from every run represents the data that has been labeled as noise by the SOM.

Table 4.13: Table presenting the top results from using the SOM algorithm with different parameter values. Each test shows the total number of clusters, the parameter values and the number of samples in each cluster. The last three columns displays the cluster validations as S (silhouette), CH (Calinski-Harabasz) and DB (Davies-Bouldin).

		Parameters				**Clusters**									**Validation**		
		Nr of cluster	Map size	Metric	Iterations	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	S	CH	DB
Test	#1	9	7 x 7	Euclidean	500	9	24	15	33	22	2	8	12	41	0.001	4.789	0.508
	#2	8	5 x 5	Manhattan	500	8	5	19	17	47	12	40	18	N/A	0.002	6.182	0.503
	#3	8	7 x 7	Manhattan	500	9	31	40	29	9	3	3	42	N/A	−0.036	4.957	0.657
	#4	7	5 x 5	Euclidean	500	12	35	36	11	39	14	19	N/A	N/A	0.044	7.021	0.639
	#5	6	3 x 3	Euclidean	500	42	15	52	12	41	4	N/A	N/A	N/A	0.062	8.414	0.488
	#6	6	3 x 3	Manhattan	500	42	15	51	40	14	4	N/A	N/A	N/A	0.044	8.661	0.775

5
Discussion

Analysis of the results of the clustering, discussion of the difficulties of clustering as well as the behaviour of the different algorithms will be presented in this chapter. There will also be a discussion related to future work, including a couple of suggestions of what the next step could be.

5.1 Analysis of the dataset

The dataset was analyzed in collaboration with the researchers at the Department of Psychology at University of Gothenburg to determine if the collected data could be considered valid. One important observation that was made was that the feature "Samples inside - Board" is steadily increasing, for a majority of the participants, as the test progresses. This corresponds to what is to be expected since the test is becoming increasingly more difficult as it progresses, i.e. the patterns displayed on the board gets more difficult to solve. A subset of participants that illustrates this can be seen in Figure 5.1.

ID	Samples inside Board (%) - Item 2	Samples inside Board (%) - Item 6	▼Samples inside Board (%) - Item 10
CD61AED406222C62	33	31	95
C0684B472FC65269	62	72	94
E062C0B3AD3B0E50	37	62	94
C9F70EA338E80D4D	71	55	92
B7B16D2DBBC61F48	43	47	91
FF0076C7C8B59318	42	43	89
08975698266B6B47	33	45	89
C0542EE18D77DAD7	21	24	88
64AB7952D87CD1A8	7	45	88
2AB968A1E0AE3F4C	6	37	88
E11F2BE478B69DC9	35	32	88

Figure 5.1: A figure that displays the features; "Samples inside Board(%) - Item 2", "Samples inside Board(%) - Item 6" and "Samples inside Board(%) - Item 10" on a subset of the data.

Another interesting observation that was made was that the time spent on an item seems to decrease slightly as the difficulty stays relatively the same on the following item. Then it takes a jump up when the difficulty increases before slowly declining

again. This could indicate that the participant gets somewhat used to solving tasks that are similar to each other and then has to put some effort into solving the ones that differ more from previous patterns.

5.2 External disturbances

Different external environments, described in Section 3.4.1, were used to gather the data. These had different levels of background noise which might have affected the performance of the participants. The large open areas and queues of people that wanted to do the test might for example have resulted in that the test person had difficulties to concentrate on the task at hand or to feel the urge to rush through the test. The more secluded areas might, on the other hand, have helped people to stay focused and perform at maximum level.

The audible disturbances could be reduced through the use of noise cancelling headphones paired with pre-recorded instructions. This would further improve the standardization of the test since everyone would receive exactly the same instructions, apart from any translation differences. However, [4] is emphasizing the importance of making sure that the test participant is feeling comfortable rather than the test supervisor focusing on delivering the instructions verbatim. With that aspect in mind it could mean that the headphones could make the test participant feel less comfortable. This could in turn affect the person's performance negatively in a similar fashion to the effects of a noisy environment which means that the problem is moved from one area to another.

5.3 Analysis of the clustering

When comparing the most similar clustering results from the different algorithms it can be seen that there are approximately four clusters that are appearing in the data. These have a high number of mutual members and clusters sizes in the same ranges.

It is difficult to determine what the underlying factors are to these clusters since there is not that much background information available about the participants. One way of gaining some more insight could be to manually analyze the clustering results using the GUI and try to identify which features that weigh the most for the different clusters.

5.3.1 The difficulties of clustering

Data clustering proved to be difficult for many different reasons. First of all there were problems related to the dataset since the outcome of the clustering is highly dependent on the quality of the data. This is also the case when it comes to other types of machine learning. The data might need to be converted to numeric representations that are supported by the algorithm and/or normalized to make sure that the different features are not falsely weighted due to different scaling ratios.

Another important part of the dataset is the choice of feature space which also greatly affect the clustering results. The features need to represent the data in a way which highlights as many of the differences between the data points as possible. This can be done through a visual inspection of the data set, with the help of experts in the field, to make an interpretation of the results. However, it is also often necessary to keep the number of features as low as possible since too large a feature space introduces issues in some clustering algorithms, e.g. "curse of dimensionality" as described in Section 2.4.1.

Then there is the difficulty of choosing the algorithm that is most suitable to solve the task at hand. This requires a good domain knowledge regarding the task and also an understanding of the behaviour of the algorithms as well as how they differ. The algorithms might also require various amounts of prior knowledge about the expected clustering results to work properly. This means that if the goal is to obtain this information through clustering then such algorithms are not viable options. The number of data points in the set is also affecting the choice of algorithm.

The algorithm that is decided upon in the previous step might exist in several different versions and implementations which makes it difficult to adapt it to the current task. Especially if the algorithm should be integrated into another piece of software.

The final problem with data clustering is the analysis of the results. This challenge is quite easy as long as there is known information about the desired clusters in beforehand, however, that is rarely the case since the goal with clustering often is to obtain said information.

5.3.2 The behaviour of the algorithms

Some of the algorithms showed behaviours that were unexpected. For example, the Mean-shift algorithm would only converge with one large cluster and varying amount of small ones by the size of one, while the DBSCAN converges at either 7-8 clusters or 2 clusters with an additional cluster containing the data points labeled as noise/outliers. This was highly depending on the parameters that were used. Since both these algorithms are based on finding clusters based on density it might explain their mutual problem of classifying a larger amount of data points as noise/outliers. However, the DBSCAN finds more unique clusters to divide the majority of the data points into, which the Mean-shift does not seem to be able to do.

The t-SNE algorithm clustered well with few data points labeled as noise/outliers, however it showed a tendency to often label a large portion, i.e. more than half the data points, as one cluster. While this was true for the Mean-shift and some runs of the DBSCAN as well, the t-SNE also found multiple smaller clusters instead of marking the rest as noise/outliers. This behaviour was more similar to the other runs of the DBSCAN and the SOM with regards to the number of clusters and the cluster sizes.

The SOM algorithm managed to find 5-8 clusters with an additional cluster containing the data points that were considered as noise/outliers. This last cluster seems to increase in size as the size of the SOM map increases. That could be as a result of the map size getting close to the same size as the dataset meaning that

close to each node in the map finds a match with a data point.

Quite a large damping was required to make the Affinity Propagation able to find a reasonably low amount of clusters of sizes similar to the other algorithms. A slight decrease in damping increased the number of clusters by approximately 15 and most of these being the size of just one data point.

These different behaviours of the algorithms implies that more knowledge and understanding of the dataset would be very helpful to properly tune the parameters without extensive testing. A deeper understanding of the dataset would also make it easier to determine which algorithms that are most likely to generate interesting results.

5.3.3 The validation of the clustering

Using the internal validation methods, described in Section 2.5, for every clustering result it was possible to get numerical ratings that was comparable across all algorithms. The silhouette method gave results ranging from -0.036 to 0.294, which shows that there is barely any separation between the clusters. This could mean that the data is dense, i.e. a single cloud with a few outliers, and that no single feature is dominating and solely representing a cluster. This is not entirely unexpected since the feature space of the data is large while, at the same time, the number of data points is quite low.

It is also worth noting that the algorithms that did score a higher value, roughly 0.2 or above, were the algorithms that labeled the majority of the data points as one true cluster. This higher score could be the consequence of the data being so dense that labeling it all as one cluster is considered good by this measure.

The values obtained during Calinski-Harabasz validation are on the other hand the highest (higher is better) when about 6-8 clusters of various sizes are found. All the values lie in between 2.841 and 8.661 with the majority around $4 - 5.5$. The Davies-Bouldin validation gave values ranging from 0.064 to 1.519 where lower is better. The best results by this measure are however experiencing a behaviour that is similar to the silhouette method.

All three validation methods and their results indicate that the number of clusters around 6-8 were the best clusterings that were found during the tests. This could mean that the underlying pattern of the dataset could correspond to this number of clusters, but there could also exist other more accurate patterns that have not been found yet.

Also, by looking at the algorithm comparisons of mutual members, it shows that some algorithms are able to find reasonably similar clusters, when looking at the number and sizes of the clusters. All the algorithms do not find exactly the same patterns, however, this comparison across algorithms supports the hypothesis that there are some underlying patterns to be further investigated.

5.4 Reasoning around the use of a GUI

The idea of a GUI arose during the process of analyzing the dataset. It proved very difficult to draw any useful conclusions, regarding the validity of the dataset,

from different printouts to the command line in Python. The GUI was therefore initially just a tool to display the dataset in a useful way but evolved into a core component that binds all parts of the developed software together.

In the discussions with the researchers at the Department of Psychology, University of Gothenburg it proved very effective to have the GUI since it gave a clear way of displaying the dataset. They highlighted the usefulness of filtering the dataset to compare specific features for one or multiple participants, as well as the benefit of displaying statistical information in different graphs, such as pie and bar charts.

The GUI could potentially be used by researchers in the field of psychology in the future to analyze datasets and perform cluster analysis. This would be very useful since it enables the use of advanced technology in psychological research without the need of competence in both psychology and computer science.

5.5 Future work

The work in this thesis could be further extended in several directions, for example through the addition of more virtual environments that are designed for other purposes. It would also be interesting to try the system with other datasets where more prior knowledge is available to verify and/or aid the analysis of the clusters.

5.5.1 Virtual environments

One idea is that it would be interesting to add more virtual environments to give the test participant other tasks that serve different purposes in the psychological testing. These environments could for example be designed to capture a persons interactions in an imitated day-to-day environment such as a classroom or workplace. It would also be of interest to create a system where several tests could be performed by a test person and where the same data types are collected from multiple virtual environments. This would make sure that it is easy to collect a lot of diverse data from the same person and that the data is comparable across different test scenarios.

Another interesting area would be to develop tests that are specifically designed to measure certain aspects of an individual's abilities. These abilities could for example be working memory, attention and processing speed.

5.5.2 Extensive targeted data collection

The clustering algorithms and the obtained results from applying these could be further evaluated and improved through more extensive and targeted data collection. For example, a larger dataset would be beneficial since a much higher number of data points than features generally means that the algorithms perform better and that the curse of dimensionality is reduced.

It is, as discussed earlier, important to have a good knowledge about what is to be expected of the dataset to be able to properly analyze it. This is also making the process of tuning the parameters for the algorithms much easier. Collecting more background data in general about the test participants could therefore aid the analysis of the clusters. It would also be interesting to test, for example, people

with known difficulties in a certain area to investigate whether this is visible in the collected data and if that is the case, to what extent.

5.5.3 Additional data parameters

The information that is gathered while performing the test has the potential to be extended. Additional information that could be interesting to obtain while performing the test includes; the electrodermal activity, the heart frequency and the number of eye blinks. The addition of more data parameters like these might improve the clustering results, since they grant additional information about the participants physical state during the test procedure.

However, the time it takes to test one person will also increase as a result since there will be more equipment to apply in the form of different sensors. This will make the process of gathering large datasets more time consuming but also more complex due to the fact that more knowledge of how to use the equipment is required.

5.5.4 Improving the GUI

The developed GUI proved useful for multiple reasons, as mentioned previously, however it could still be improved through the addition of more functionality. The ability to switch between datasets could be useful since it opens up the possibility to easily analyze multiple datasets that has been collected, e.g. from different virtual testing environments.

Another functionality that could be useful within the GUI is the cluster comparison. This opens up the possibility to more easily compare the algorithms and their individual clustering results as well as to visualize the overlap between clusters. The results could also be saved to allow the user to choose what to compare at a later time.

6
Conclusion

This thesis has investigated whether it is possible to find clusters in behavioural data that have been collected using a VR representation of a psychological test. The clustering results from several of the applied algorithms indicates that there are indeed some underlying pattern in the data that corresponds to approximately six to eight clusters. The study shows the potential for use of advanced technology in psychological research. This does, however require further evaluation and development of the algorithms and the clustering results. The latter could potentially be improved through more extensive and targeted data collection.

The technological advancements within VR as well as machine learning shows promise when used for psychological research, and will most likely in the near future have a function within this field. The possibility to immerse the participants within a VE, could further ensure that tests are used with equal conditions regardless of location in reality. Another advantage gained by applying VR could be that the accuracy of the raw data collected while testing increases significantly.

The process of analyzing the gathered data became more efficient as well as easier to interpret with the development and use of a GUI. Further development could be done to improve the GUI through the addition of more functionality, such as the possibility to switch datasets and to ability to perform the cluster comparison directly within the GUI.

Bibliography

[1] Chire, Wikipedia, the free encyclopedia, "Dbscan illustration," 2011.

[2] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, vol. 96, pp. 226–231, 1996.

[3] R. M. Bakwin, A. Weider, and H. Bakwin, "Mental testing in children," *The Journal of pediatrics*, vol. 33, no. 3, pp. 384–394, 1948.

[4] A.-C. Smedler and E. Tideman, *Att testa barn och ungdomar : om testmetoder i psykologiska utredningar*. Stockholm: Natur & kultur, 1. utg. ed., 2009.

[5] J. H. Elder, "Videotaped behavioral observations: enhancing validity and reliability," *Applied Nursing Research*, vol. 12, no. 4, pp. 206–209, 1999.

[6] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution," *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.

[7] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

[8] P. Langley, "The changing science of machine learning," *Machine Learning*, vol. 82, no. 3, pp. 275–279, 2011.

[9] T. Yarkoni and J. Westfall, "Choosing prediction over explanation in psychology: Lessons from machine learning," *Perspectives on Psychological Science*, vol. 12, no. 6, pp. 1100–1122, 2017.

[10] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 568–573, IEEE, 2005.

[11] J. A. Russell and J. M. Fernández-Dols, *The psychology of facial expression*. Cambridge university press, 1997.

[12] Y. Cui, M. Gierl, and Q. Guo, "Statistical classification for cognitive diagnostic assessment: An artificial neural network approach," *Educational Psychology*, vol. 36, no. 6, pp. 1065–1082, 2016.

[13] G. Deshpande, P. Wang, D. Rangaprakash, and B. Wilamowski, "Fully connected cascade artificial neural network architecture for attention deficit hyperactivity disorder classification from functional magnetic resonance imaging data," *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2668–2679, 2015.

[14] R. Adams, P. Finn, E. Moes, K. Flannery, and A. Rizzo, "Distractibility in attention/deficit/hyperactivity disorder (adhd): The virtual reality classroom," *Child Neuropsychology*, vol. 15, no. 2, pp. 120–135, 2009.

[15] Y. Pollak, P. L. Weiss, A. A. Rizzo, M. Weizer, L. Shriki, R. S. Shalev, and V. Gross-Tsur, "The utility of a continuous performance test embedded in virtual reality in measuring adhd-related deficits," *Journal of Developmental & Behavioral Pediatrics*, vol. 30, no. 1, pp. 2–6, 2009.

[16] A. A. Rizzo, M. Schultheis, K. A. Kerns, and C. Mateer, "Analysis of assets for virtual reality applications in neuropsychology," *Neuropsychological Rehabilitation*, vol. 14, no. 1-2, pp. 207–239, 2004.

[17] M. Dyck, M. Winbeck, S. Leiberg, Y. Chen, R. C. Gur, and K. Mathiak, "Recognition profile of emotions in natural and virtual faces," *PLoS One*, vol. 3, no. 11, p. e3628, 2008.

[18] P. Lindner, A. Miloff, W. Hamilton, L. Reuterskiöld, G. Andersson, M. B. Powers, and P. Carlbring, "Creating state of the art, next-generation virtual reality exposure therapies for anxiety disorders using consumer hardware platforms: design considerations and future directions," *Cognitive Behaviour Therapy*, pp. 1–17, 2017.

[19] N. J. Emery, "The eyes have it: the neuroethology, function and evolution of social gaze," *Neuroscience & Biobehavioral Reviews*, vol. 24, no. 6, pp. 581–604, 2000.

[20] A. I. Goldman *et al.*, "Theory of mind," 2012.

[21] N. I. Vargas-Cuentas, D. Hidalgo, A. Roman-Gonzalez, M. Power, R. H. Gilman, and M. Zimic, "Diagnosis of autism using an eye tracking system," in *Global Humanitarian Technology Conference (GHTC), 2016*, pp. 624–627, IEEE, 2016.

[22] A. Armanini and N. Conci, "Eye tracking as an accessible assistive tool," in *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pp. 1–4, IEEE, 2010.

[23] J. Pettersson, A. Albo, J. Eriksson, P. Larsson, K. Falkman, and P. Falkman, "Cognitive ability evaluation using virtual reality and eye tracking," in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) (CIVEMSA 2018)*, (Ottawa, Canada), June 2018.

[24] Psykologförbundet, "Hantering och förvaring av psykologiska test inom hälso- och sjukvården." [Online], 2013. `https://www.psykologforbundet.se/globalassets/omforbundet/hantering-och-forvaring-av-psykologiska-test.pdf`.

[25] J. Raven *et al.*, "Raven progressive matrices," in *Handbook of nonverbal assessment*, pp. 223–237, Springer, 2003.

[26] J. Raven, "The raven's progressive matrices: change and stability over culture and time," *Cognitive psychology*, vol. 41, no. 1, pp. 1–48, 2000.

[27] M. Dahl, A. Albo, J. Eriksson, J. Pettersson, and P. Falkman, "Virtual reality commissioning in production systems preparation," in *22nd IEEE International Conference on Emerging Technologies And Factory Automation, September 12-15, 2017, Limassol, Cyprus*, 2017.

[28] I. E. Sutherland, "A head-mounted three dimensional display," in *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pp. 757–764, ACM, 1968.

[29] S. Choi, K. Jung, and S. D. Noh, "Virtual reality applications in manufacturing industries: Past research, present findings, and future directions," *Concurrent Engineering*, vol. 23, no. 1, pp. 40–63, 2015.

[30] A. Poole and L. J. Ball, "Eye tracking in hci and usability research," *Encyclopedia of human computer interaction*, vol. 1, pp. 211–219, 2006.

[31] T. D. Gould, T. M. Bastain, M. E. Israel, D. W. Hommer, and F. X. Castellanos, "Altered performance on an ocular fixation task in attention-deficit/hyperactivity disorder," *Biological psychiatry*, vol. 50, no. 8, pp. 633–635, 2001.

[32] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[33] E. Keogh and A. Mueen, "Curse of dimensionality," in *Encyclopedia of Machine Learning and Data Mining*, pp. 314–315, Springer, 2017.

[34] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[35] D. Polani, "Kullback-leibler divergence," in *Encyclopedia of Systems Biology*, pp. 1087–1088, Springer, 2013.

[36] T. Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 1–6, 1998.

[37] F. Farzad and A. H. El-Shafie, "Performance enhancement of rainfall pattern–water level prediction model utilizing self-organizing-map clustering method," *Water Resources Management*, vol. 31, no. 3, pp. 945–959, 2017.

[38] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40, 1975.

[39] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[40] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *International Journal of computers and communications*, vol. 5, no. 1, pp. 27–34, 2011.

[41] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, "Adaptive affinity propagation clustering," *arXiv preprint arXiv:0805.1096*, 2008.

[42] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[43] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[44] Tobii AB, "Tobii pro vr integration." [Online], 2017. `https://www.tobiipro.com/siteassets/tobii-pro/product-descriptions/tobii-pro-vr-integration-product-description.pdf/?v=1.6`.

[45] Unity Technologies, "Unity3d: The world's leading content-creation engine," May 2018.