The University Of Sheffield.

Faculty of Engineering

Department of Computer Science

Semi-supervised K-Means clustering for trajectory analysis in behavioural experiments

Avgoustinos Vouros

A thesis presented for the degree of
Doctor of Philosophy

Supervisor: Prof. Eleni Vasilaki

October 02, 2020

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and are the result of my own work and research during the four years of my PhD. Most of the material presented has already been peer reviewed and published, or is about to be, in the form of journal papers. In the studies where I am not the first author, the work presented here is my contribution to the relevant manuscript. None of the contents has been submitted in the whole or in part for consideration for any other degree or qualification in this, or any other university.

Avgoustinos Vouros
March 16, 2021

Acknowledgments

First and foremost I would like to thank my supervisor, Eleni Vasilaki for her invaluable advice, continuous support, and patience during my PhD study. Her immense knowledge and ample experience have encouraged and guided me during the time of my academic research and daily life. Moreover, I would like to thank my PhD committee members and advisors, Achim Brucker, Richard Clayton, Dawn C. Walker and Mauricio Álvarez for their invaluable guidance and support during all the stages of this PhD research. I would also like to express my gratitude to Mike Croucher for introducing me to software engineering practices in research applications and the research software engineering society in Sheffield and beyond. Furthermore, I would like to thank the Numerical Algorithms Group (NAG) for its financial support during my PhD and Stephen Langdell for his constructive criticism and support.

Special thanks also to all my research collaborators. Daniel Wójcik and Carmen Sandi for welcoming me for extended visits to their labs in Warsaw and Switzerland; Kinga Szydłowska and Artur Janusz for their comments on the usability of my research methods and Damien Huzard for his great collaboration. Lastly, a huge thank you to Karishma Chhabria for the countless hours we spent together in the lab trying to make sense of how diabetic zebrafish behave.

I would also like to thank all my labmates and friends that stood beside me throughout my PhD journey. Matt Ellis for proofreading the mathematical formulas, Marialena Dounavi for the very many hours of geekiness we spent together discussing science, Sokratis Kariotis for the endless evenings of board gaming and Alex Chronopoulos for his anti-anxiety medication of video gaming.

Finally I am extremely grateful to my parents Ioannis Vouros and Katerina Valaristou who supported my decision to pursue a PhD and are always there for me, ready to support me in every possible way.

Abstract

Behavioural neuroscience uses a variety of animals to model human diseases, test novel drugs and study how certain factors affect or alter natural behaviour. In its arsenal, it contains a number of different experimental procedures involving navigation and locomotion tasks inside constrained environments and a number of analysis techniques to draw conclusions about various aspects of neuroscience such as the development of learning and memory. With the advancements in technology and the rise of artificial intelligence in many areas of our society, machine learning algorithms and applications are commonly used to draw, with limited user interaction and in a speedy manner, as much intelligence as possible from collections of data. Machine learning has greatly boosted behavioural neuroscience research but in many cases it provides experiment-specific analysis methods requiring domain knowledge in order to be used. This work addresses the first limitation of experiment-specific analysis methods by bringing an integration of common metrics used in different experimental procedures involving path analysis. For the second limitation it proposes a machine learning agnostic framework for data analysis in a common experimental procedure called Morris Water Maze which can also be used to other experiments involving behavioural categorisation tasks. In addition, it proposes a novel machine learning method for detailed analysis of locomotion that can be applied to any navigation task for both automatic categorisation and pattern recognition tasks. Other objectives of this study are to present detailed benchmarks of machine learning techniques that can be used for data analytics in behavioural neuroscience and to expand the usability of the methods it presents by making them easy to use by the research community. For this reason, all the source codes of the presented algorithms and pipelines is publicly available and, when applicable, graphical user interfaces or software tools have been engineered to help executing them.

Contents

List of Figures

List of Tables

Notations

Unless specified otherwise inside the text,

- The vector notation $x_{i:} = [x_{i1}, x_{i2}, \ldots, x_{ip}]$ specifies the i-th element of the matrix containing the data set $\mathcal{X}$ consisting of n observations and p dimensions (or features or data attributes). j is an index on the p dimensions.

- Given K number of groups (clusters) $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, with $n_1, n_2, \ldots, n_K$ number of elements in each group respectively (n without an index will refer to the number of elements of the whole data set), the vector notation $m_{k:} = [m_{k1}, m_{k2}, \ldots, m_{kp}]$ specifies the k-th group center (centroid). This group center is the mean of the data in the group.

- The notation $\mu_{1:}$ refers to the global center of the data set, which is unique, and $\mu_{1:} = [\mu_{11}, \mu_{12}, \ldots, \mu_{1p}]$.

- Given K class labels ℓ then n_ℓ are the number of elements belonging to each class and $n_\ell^{(k)}$ is the number of elements of class ℓ belonging to cluster k.

- The letters w and a are reserved to specify the weights of each dimension, i.e. $w_1, w_2, \ldots, w_p$ and $a_1, a_2, \ldots, a_p$.

- The stylized letter $\Bbbk$ is used to indicate the $\Bbbk$-fold cross validation.

- The notation,

$$\sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} x_{i:} = \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{j=1}^{p} x_{ij}$$

  specifies a summation of all the p-dimensional data points $x_{i:}$, $i = 1 \ldots n_k$ which belong to the k-th group ($x_{i:} \in c_k$).

Chapter 1

Introduction

Behavioural neuroscience has a rich history of experimental procedures with different animal models in order to identify factors that affect the natural behaviour of organisms. Often such procedures involve navigation into constrained environments, under certain conditions. Understanding the different animal actions throughout an experimental procedure can provide valuable information in various aspects of neuroscience such as how certain areas of the brain operate, how learning and memory is developed, and how the organism is affected by specific stimuli, effects and conditions.

With the advancements in technology, the collection of data in many fields of neuroscience has greatly increased and with that the need for new methods aiming at drawing intelligence from them [Vu et al., 2018]. To this end, the field of machine learning plays an important role in data analysis and applications of machine learning in neuroscience include automatic classification of neuron cell types [Armañanzas and Ascoli, 2015] and neuronal firing patterns (spike-sorting) [Horton et al., 2007], identification of neuron structural boundaries in electron microscopy (EM) images [Jain et al., 2010; Zhu et al., 2014] and recognition of subtypes of depression from functional magnetic resonance imaging (fMRI) [Drysdale et al., 2017]. Applications of machine learning specifically for behavioural neuroscience include measurement, identification, and categorization of different animal behaviours [Han et al., 2018; Hong et al., 2015] during experimental procedures [Illouz, Madar, Clague, Griffioen, Louzoun and Okun, 2016], especially in the procedure of the Morris Water Maze [Garthe et al., 2009; Illouz, Madar, Louzon, Griffioen and Okun, 2016].

For such behavioural tasks involving animal path tracking there is an interest for behaviour as a whole against to the detailed listing of certain path attributes and performance measurements. Examples of such analyses are available mainly for the Morris Water Maze experimental task where, behavioural analysis has proven superior to the performance measurements as behaviours can clearly show different stages of learning and memory and identify differences among animal groups [Dalm et al., 2000; Gehring et al., 2015; Graziano et al., 2003; Wolfer and Lipp, 2000]. In addition, existing procedures for behavioural analysis that are based on machine learning methods are focused on the automatic classification of existing stereotypical behaviours and not on the identification of new ones. Furthermore, many analysis pipelines with one or more machine learning components require a degree of expertise in that field since many methods require manual tuning and adaptation to the specific data at hand. Such expertise is not always available and adaptation may require

additional research.

To this end, the main objective of this dissertation is to bring together behavioural neuroscience and machine learning by proposing and designing auto-tunable solutions and tools for detailed analysis of animal behavioural motifs inside experimental procedures. Aims of this dissertation are as follows:

- To benchmark existing K-Means initialisation methods and their effects on clustering variations to identify the most appropriate and robust methods.
- Design a semi-supervised K-Means clustering methodology capable of performing feature selection and assessment without affecting the classification performance.
- Collect and/or engineer generic features that describe aspects of animal pathing inside constrained environments regardless of the type of the subjects and the experimental procedure that were used.
- Create a generic framework for detailed classification of animal behaviours inside the Morris Water Maze which also provides information about the effect of each feature on the classification task.
- Propose extensions of the researched methods to experimental procedures beyond the Morris Water Maze where information about stereotypical animal behaviours is not available.

This dissertation is built around the following material:

1. **Vouros, A.**, Gehring, T.V., Szydlowska, K., Janusz, A., Tu, Z., Croucher, M., Lukasiuk, K., Konopka, W., Sandi, C. and Vasilaki, E., 2018. A generalised framework for detailed classification of swimming paths inside the Morris Water Maze. Scientific reports, 8(1), p.15089.
2. Huzard, D., **Vouros, A.**, Monari, S., Astori, S., Vasilaki, E. and Sandi, C., 2019. Constitutive differences in glucocorticoid responsiveness are related to divergent spatial information processing abilities. Stress, pp.1-13.
3. Chhabria, K., **Vouros, A.**, Gray, C., MacDonald, B. R., Jiang, Z., Wilkinson, R. N., Plant, K., Vasilaki, E., Howarth, C., Chico, T. J. A., 2019. Sodium nitroprusside prevents the detrimental effects of glucose on the neurovascular unit and behaviour in zebrafish. Disease Models & Mechanisms 2019 12: dmm039867.
4. **Vouros, A.**, Langdell, S., Croucher, M. and Vasilaki, E., 2019. An empirical comparison between stochastic and deterministic centroid initialisation for K-Means variations. arXiv preprint arXiv:1908.09946. *Revised and resubmitted to Machine Learning.*
5. **Vouros, A.**, & Vasilaki, E. (2020). A semi-supervised sparse K-Means algorithm. arXiv preprint arXiv:2003.06973. *Submitted to Pattern Recognition Letters.*

6. **Vouros, A.**, & Vasilaki, E. (2020). An extended framework for behavioural classification and feature selection in the Morris Water Maze experimental procedure. Poster presentation. 12th FENS Forum of Neuroscience, 11-15 July 2020.

The dissertation structure is as follows:

- **Chapter 2** consists of a short literature review on relevant aspects of this PhD study: *(a)* behavioural neuroscience experiments (the Morris Water Maze and the light/dark preference task), *(b)* K-Means methods, feature engineering, selection and weighting, clustering initialisation techniques; K-Means variations, K-Means sparse clustering, semi-supervised learning and clustering tuning and performance evaluation methods, *(c)* review old and existing data analysis methods for the aforementioned behavioural neuroscience experiments.

- **Chapter 3** is based on [Vouros et al., 2019; Vouros and Vasilaki, 2020] and provides an extensive benchmark on stochastic and deterministic initialisation methods for K-Means clustering and presents a new clustering technique that combines sparse clustering with semi-supervised learning. The benchmark aims to provide experimental evidence that deterministic initialisation methods can, on average, surpass the performance of stochastic methods. Compared with previous benchmark studies synthetic data set models were used in order to generate multiple data sets with different attributes and incorporate hypothesis testing to strengthen the benchmark conclusions. In addition, multiple K-Means algorithms and K-Means inspired methods were considered including K-Medians, Sparse K-Means and semi-supervised K-Means variations. The latter includes a novel modification of the Sparse K-Means algorithm named Pairwise Constrained Sparse K-Means (PCSK-Means).

- **Chapter 4** is focused on feature engineering for experimental procedures involving animal movements. Such features are generic and applicable to any procedure regardless of the animal. A case study is the work of [Chhabria et al., 2019] where I performed behavioural analysis on zebrafish larvae inside the light/dark preference task. This study is based on manual investigation and detection of features for capturing behavioural information.

- **Chapter 5** explores the continuation of the work of [Gehring et al., 2015] for detailed classification of rodents behaviours inside the Morris Water Maze and it is build based on the publications [Vouros et al., 2018] and [Huzard et al., 2019]. In the first study an improved methodology for detailed classification inside the Morris Water Maze using semi-supervised learning is proposed. This improvement is based on a classification boosting technique that nullifies the need of manual tuning of its underline machine learning methods. In addition research was conducted to identify under which path length the different animal behaviours are identifiable. An open=source software called RODA (ROdent Data Analytics) [Vouros et al., 2017] was also implemented to make the framework and the methods proposed in [Vouros et al., 2018] publicly available. RODA was latter used in the study of [Huzard et al., 2019].

- **Chapter 6** describes a real-world application of the PCSK-Means algorithm proposed in this dissertation. The aim of this study is to provide an automatic

way of feature selection and assessment for the Morris Water Maze behavioural task that can also be adopted to other experiments. In this study PCSK-Means is compared against MPCK-Means which is the algorithm used in [Gehring et al., 2015; Huzard et al., 2019; Vouros et al., 2018] and is able to handle both classification and identification of important path features. Results from this chapter were presented in a poster session at the Federation of European Neuroscience Societies (FENS) 2020.

- **Chapter 7** contains the conclusions of this PhD study and discussion on alternative methods for feature engineering and behavioural classification using neural networks and deep learning summarising their advantages and disadvantages.

Chapter 2

Literature review

This chapter aims to introduce the reader to the relevant topics of this PhD study. These topics are split into three distinctive sections: (a) behavioural experiments, (b) machine learning concepts and (c) data analytics methods for the aforementioned behavioural experiments. In more detail this Chapter,

(a) Describes some generic concepts about navigation tasks in behavioural neuroscience and introduce the experimental procedures of the Morris Water Maze and the light/dark preference task.

(b) Briefly describe the concepts of supervised and unsupervised machine learning and introduce the K-Means clustering. Description of K-Means formulation, variations and algorithms will be provided including semi-supervised modifications. Finally, this section will include common criteria used for clustering tuning and performance evaluation.

(c) Bridges sections (a) and (b) by describing common data analytics methods for the experimental procedures of the Morris Water Maze and the light/dark preference task and proceed on describing how machine learning has influenced them.

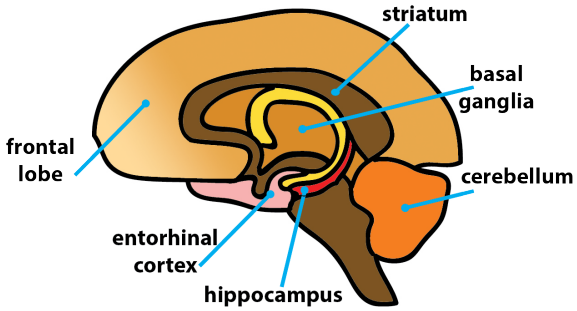
2.1 Behavioural experiments involving navigation

2.1.1 Navigation tasks in animal models

Navigation, the ability to learn and remember various locations, is an essential survivability process for nearly every animal specie. As a whole, the navigation task is achieved by two processes: the allocentric and the egocentric [Braun et al., 2012]. The allocentric process involves mainly the hippocampus and the entorhinal parts of the brain (refer to Figure 2.1) and is referring to the ability of encoding spatial information of a specific object in relation to other objects in space [Vorhees and Williams, 2014]. The egocentric process involves mainly the dorsal striatum (refer to Figure 2.1) and is referring to the ability of encoding spatial information of the objects in relation to the observer [Vorhees and Williams, 2014].

Navigation tasks is one of the main methods for studying spatial learning and memory and most of the research on this field has been focused on various such tasks which are performed by rodents inside mazes [Paul et al., 2009]. By designing

Figure 2.1: Parts of the brain involved in navigation. Allocentric process involves mainly the hippocampus and the entorhinal cortex; egocentric process involves mainly the dorsal striatum (image adapted from https://bit.ly/2Vu3AW2).



specific experimental procedures it is possible to study specific structures of the brain, investigate how certain conditions affects them and how to design specialized treatments to undo the damage caused by negative effects (e.g. ageing [Wei et al., 2005], diseases [Ingram et al., 1994], various accidents). The preferred experimental animals for these procedures are usually rodents and specifically rats and mice because of their anatomical, physiological and genetic similarities to humans [Bryda, 2013].

Apart from rodents, other animal models are also being used including octopuses [Boal et al., 2000], zebrafish [Avdesh et al., 2012; Roberts et al., 2013] and bees [Hammer and Menzel, 1995; Menzel and Erber, 1978]. Some experimental procedures have also been adapted in virtual reality environments to be applicable on human subjects [Gillner and Mallot, 1998; Waller et al., 2001].

Throughout the bibliography there is a variety of different mazes and experimental procedures because the brain is a complex system which consists of many networks. Thus, it is impossible to completely isolate a specific task or a specific condition only to a particular brain region [Vorhees and Williams, 2014]. Moreover, nearly every experiment is affected by a variety of factors such as sex, age, specie and nutrition of the participant animal subjects [D'Hooge and De Deyn, 2001]. For these reasons this dissertation will consider only two distinctive experimental procedures: the Morris Water Maze (MWM) [Morris, 1981; Morris et al., 1986] with rodents and the light/dark preference task with zebrafish [Maximino et al., 2012].

2.1.2 The Morris Water Maze

The Morris Water Maze (MWM) is one of the most commonly used tasks in behavioural neuroscience. It was designed by Richard Morris and was first described back in 1981 on a study regarding the spatial localization of rats [Morris, 1981, 2008]. The popularity of this task was massive and by the end of eighties a large number of published work using the MWM had been reported [Brandeis et al., 1989]. In addition, the review work of D'Hooge and Deyn mentions more than 2000 scientific reports regarding the Morris Water maze task in the decade 1990-2001 [D'Hooge and De Deyn, 2001]. Finally, this task remains popular even to-date since in the work of Daugherty et. al. (published on 2015) a virtual MWM had been used on human subjects to study the effects of ageing in navigation [Daugherty et al., 2015]. The same concept has also been used in further studies with human participants such as [Piber et al., 2016] and [Korthauer et al., 2017].

Throughout the years since the MWM was first reported a number of variations, paradigms and training protocols were implementing for this experiment, specifically targeting different aspects of learning and memory and the different brain regions implementing them [Vorhees and Williams, 2014]. Furthermore MWM has extensively been used to test, validate and evaluate neurocognitive treatments or conditions [D'Hooge and De Deyn, 2001]. Next, the general procedure and the most common protocols of MWM will be described.

2.1.2.1 Basic procedure

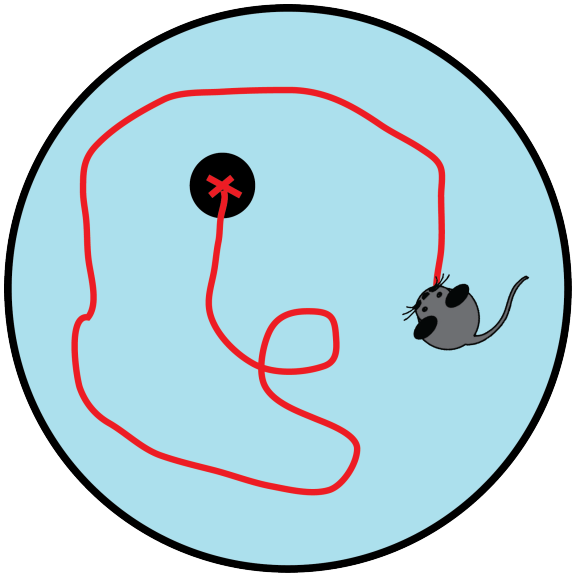
In a typical MWM experiment (illustrated in Figure 2.2) the rodent is placed inside a circular pool filled with water and is tasked to find a hidden platform which is placed in one of the four quadrants of the pool [Morris, 2008]. Since it is unable to see the platform it has to relay on external visual cues in order to navigate inside the pool and finally find the platform [Morris, 1984]. After many trials it is expected that the animal will have memorize the location of the platform thus it will be able to find it more quickly. Over the years this procedure has been changed in numerous ways thus this dissertation will limit itself only to the more commonly used ones.

Starting from the pool and the platform, there are no exact standards nor systematic research on their exact sizes. On his first experiments with rats Morris used pools with diameters 132 cm and 214 cm, but with no clear description of these decisions. Other researchers used smaller pools in order to experiment with mice with success. In their work, Vorhees and Willians [Vorhees and Williams, 2014] conclude that in case the pool is too small then the task is solved very quickly thus less information about learning is obtained and, furthermore, there is the risk that the animal will complete the task without using any cues of its environment. Moreover there seems to be a difference between the rats and the mice; rats have the ability to solve the task even in large environments while mice fail to do so. Regarding to the platform, Vorhees and Willians [Vorhees and Williams, 2006] refer to a typical squared 10 cm^2 or 11 cm^2 platform or to a circular one with diameter 10 to 12 cm.

Another important parameter in the MWM is the water temperature. Nearly every review on the MWM (e.g. [Brandeis et al., 1989; D'Hooge and De Deyn, 2001; Sahgal, 1993; Vorhees and Williams, 2014]) has a reference to the disadvantage of the sudden immersion of the subject into the water as this can cause extensive stress to the animal resulting into the failure of the experiment. Moreover there is the risk of hypothermia which causes disruptions on learning and memory [Lindner and Gribkoff, 1991; Panakhova et al., 1984; Rauch et al., 1989]. Fundamental factor for the temperature selection is to trigger the escape motivation of the animal in order to finish the task [Tonkiss et al., 1994], but there are also more factors that needs to be considered such as the housing conditions [Tonkiss et al., 1994] and the age of the animals (older animals have higher risk of hypothermia [Lindner and Gribkoff, 1991]). A method which is used to ensure that temperature is not affecting the experiment is to place each animal inside a warming cage after each trial to warm it up [Weitzner et al., 2015]. Finally it should be mentioned that the above concepts are applicable on both rats and mice. For example on the reference studies [Bromley-Brits et al., 2011; Stackman et al., 2012], mice have been used instead of rats and the procedures used were almost equivalent.

An additional point regarding the MWM experiment is the visual cues. The animal needs to be able to navigate based on the cues of the room surrounding

Figure 2.2: The basic Morris Water Maze task. A round pool is filled of opaque water and the rodent needs to find a hidden platform (black circle) within the specified time period of the trial. The rodent trajectory (red line) is recorded throughout the trial. The animal navigates based on the surrounding environment with no direct cues to the platform location



the pool [Brandeis et al., 1989]. More cues (e.g. images, geometric objects, etc. [Buccafusco, 2000]) could be placed intentionally or hidden (in some trials a curtain is placed around the pool [Weitzner et al., 2015]). Moreover the room surrounding the pool is important to remain totally unchanged as visual cues might be obscured otherwise [Vorhees and Williams, 2006]. On the other hand the location of the platform needs to be absolutely hidden and this can be achieved by making the water opaque (e.g. by using milk) [Morris, 2008] or, in case of clear water, the platform needs to be made from plexiglass [Buccafusco, 2000].

Finally it should be mentioned that the starting location of the animals inside the maze is random but researchers must make sure to place each animal into all the four quadrants of the pool [Buccafusco, 2000]. Each trial begins the moment that the animal is placed in the pool until it reaches the platform and during the trial the whole animal swimming path (trajectory) is recorded [Buccafusco, 2000].

2.1.2.2 Test protocols

- **Pre-training and Cued trials:** These trials target to make the animal familiar with the MWM set-up in order to eliminate nonspatial behaviours [Vorhees and Williams, 2014]. Examples of nonspatial behaviours can be the sudden stress of putting the animal into water, the inability for the animal to recognize the platform as the escape point or extended use of a strategy called Thigmotaxis where the animal is moving only around the wall of the pool or tries to climb it [Vorhees and Williams, 2014; Weitzner et al., 2015]. Usually in these trials the platform is totally visible or has a very distinctive cue like a flag [Weitzner et al., 2015]. Finally these kind of trials are important to identify any cognitive or sensory impairment (e.g. the animal may have impaired vision or motor skills) [Buccafusco, 2000]. It has been reported that cue trials can be performed before or after the hidden trials (described next) but they are beneficial mainly before [Vorhees and Williams, 2014].

- **Hidden platform trials:** The main acquisition phase of the MWM experiment where the platform is completely hidden and the animal needs to be able to find it in order to escape the maze. There is a huge variation in the number of trials per day but the most common number is four trials per day [Vorhees and Williams, 2014]. The animal is usually having a timeout of 90 seconds in order to find the platform and if it fails to do so then it is manually placed at the location of the platform [Buccafusco, 2000].

- **On-demand trials:** This kind of procedure was introduced by Buresova et al. [Burešová et al., 1985] some years after the introduction of the MWM. In this modification of the original MWM the hidden platform was collapsible and it was raised only after the animal remained at the platform's location for a specific time [Brandeis et al., 1989]. Buresova et al. used this procedure to increase the accuracy of the MWM on experiments regarding the cognitive maps of the animals and the implications that arise after specific interventions [Brandeis et al., 1989]. This kind of technique (also known as the Atlantis platform) is also useful as sometimes the animals tend to jump from the platform as soon as they have reached it [Morris, 2008].

- **Reversal trials:** As the name suggests, during these trials the platform is moved to the opposite quadrant of the pool from which it was first located [Vorhees and Williams, 2014]. The reversal trials may be used to assess if the animal has fully learn the platform position, thus it moves straight to it [Vorhees and Williams, 2006], detect damage in the hippocampus area of the brain [Morris et al., 1986] or investigate how quickly the animal adapts to the situation and starts searching for the new platform location [Morris, 2008]. Specifically for the last case, there are differences between the behaviour or rats and mice; rats quickly learn to search for the new platform location whereas mice remain focused on the old platform location for longer amount of time [Vorhees and Williams, 2006].

- **Transfer or Probe trials:** In this kind of trials the platform is completely removed from the pool offering a way to assess the spatial memory of the animal [Morris, 2008]. This is achieved as the animal will typically swim straight to the quadrant of the platform thus various measurements can be collected such as quadrant preference and the number of crossings over the location in which the platform was previously located [Buccafusco, 2000; Vorhees and Williams, 2014].

- **Discrimination trials:** Same as in the cue trials the location of the platform during the discrimination trials is visible. The difference here is that two platforms are used inside the pool; one rigid, which can support the animal and one floating, which will immerse in the water due to the animal weight [Brandeis et al., 1989]. These kind of trials can be used to assess both spatial and nonspatial learning as the platforms can be visually identical or each one can have a distinctive appearance (different colour or shape). In the first case the animal is required to learn the location of the correct one using the distal cues of the room while in the other it needs to learn the different appearance of the correct one [Morris, 2008; Vorhees and Williams, 2006].

- **Working Memory trials:** In the working memory procedure the location of the hidden platform is changing every day and generally only 2 types of trials are performed per day; the sample trial and the successive (test) trial(s) [Vorhees and Williams, 2006]. These trials are useful to assess how well the animal obtains information about the location of the platform during the sample trial and use it to find the platform faster during the successive trial(s) [Brandeis et al., 1989; Morris, 2008]. This specific task can be altered by increasing or decreasing the time between the sample and the successive trial(s), which is known as inter-trial interval (ITI) or memory delay [Buccafusco, 2000; Morris, 2008].

2.1.2.3 Applications of the Morris Water Maze

The MWM has been successfully used in many studies focused on specific areas of the brain which are closely connected to learning and memory, the assessment of animal models for neurocognitive disorders, the research on neurocognitive therapies and the neuropharmacology of spatial learning [D'Hooge and De Deyn, 2001; Morris, 1984]. All of these studies approach learning and memory from a different prospective but they share a lot of similarities; for example the Alzheimer's disease has been extensively examined using animal models imitating different stages of the disease and at the same time many different treatments has been tested on these models to alleviate its severe neurocognitive dysfunctions.

2.1.2.4 Conclusions about the Morris Water Maze

After the discussion on some of the procedures of the MWM as well as a limited review on its vast number of applications, the following advantages can be stated about this experimental procedure:

1. Simplicity of the problem. It is easy for the animals to learn the MWM procedure and since rodents are natural swimmers little training is required [Vorhees and Williams, 2014].

2. Extremely flexible to adaptation [Buccafusco, 2000]. A variety of different test procedures and protocols has been implemented over the years for the MWM targeting specific aspects of learning and memory. Nevertheless, the basic set-up of the MWM remains the same.

3. Minor to none dependency on appetite, sense motivators (e.g. electrical shocks), body weight and non-spatial (e.g. odour-based) behaviours [Brandeis et al., 1989]. Behavioural experiments conducted with other mazes have been heavily criticized because of the use of a variety of factors which may lead to false assumptions. For example the T-maze and its variations (e.g. multiple T-maze, Y-maze) are based on food reward which can cause implications when the treatment affects the appetite of the animal [Vorhees and Williams, 2014].

4. Easy to distinguish spatial from non-spatial behaviours (e.g. sensory information) [Brandeis et al., 1989].

On the other hand the MWM holds specific disadvantages that have receive significant criticism:

1. The animal is prone to receiving extensive stress due to its sudden water immersion and due to the fact that it is placed into a trapped-like environment [D'Hooge and De Deyn, 2001; Morris, 2008; Vorhees and Williams, 2014].

2. It is sensitive to animal characteristics which leads to species-specific behaviours inside the MWM [D'Hooge and De Deyn, 2001; Vorhees and Williams, 2014].

3. There are still not enough evidence to support that the MWM working memory procedures are superior to the ones used with other mazes (such as the the radial-arm maze) [Vorhees and Williams, 2014].

2.1.3 The light/dark preference task

The light/dark preference task was originally used with rodents (where it is named as light-dark box test) primarily for the study of anxiety [Blumstein and Crawley, 1983; Crawley and Davis, 1982; Crawley and Goodwin, 1980] and the effects of anxiolytic (anxiety reduction) and anxiogenic (anxiety increase) drugs and substances [Blaser and Penalosa, 2011]. The structure of the test has different variations [Bourin and Hascoët, 2003] but it is generally consisted of two areas, one lighted and one dark [HascoëT and Bourin, 1998; Serchov et al., 2016]. With the growing interest of behavioural research on zebrafish [Maximino et al., 2012], the light/dark task was used as a paradigm in many studies using zebrafish models [Blaser and Penalosa, 2011; DePasquale and Leri, 2018; Magno et al., 2015; Steenbergen et al., 2011].

The general procedure of the task is straightforward, the animal is left free to explore the area as its path is recorded. Based on the protocol of [Takao and Miyakawa, 2006] for mice and the recent tank and procedure design protocol of [Liu and Sitaraman, 2019] for zebrafish, a separator is used between the light and dark areas of the experimental area with one (in mice) or multiple (in zebrafish) entrances/exit between the two areas (refer to Figure 2.3). In case of zebrafish larvae, the subjects are placed within round wells in a plate format and then the plate is placed in a room with a light switch [Schnörr et al., 2012]. In our study [Chhabria et al., 2019] (refer to Chapter 4) the half of each well was covered by cellophane films (blue, green and yellow) to create the 'dark' area.

Even though the light/dark preference task has been criticised for its inconsistency and unreliability it still remains a popular behavioural test [Ennaceur, 2014]. When experimenting with rodents, since they are nocturnal animals, the task relies on the fact of natural avoidance of open and well-lit areas by rodents [Champagne et al., 2010] (a behaviour known as scototaxis [Liu and Sitaraman, 2019]). On the other hand, zebrafish are diurnal animals and should prefer bright environments for mating, foraging and predator avoidance [Champagne et al., 2010]. However, the transference of the light/dark preference task to zebrafish still produces inconsistent results among different studies on the animal preference between light or dark compartments of the experimental arena [Liu and Sitaraman, 2019]. Based on the study of [Liu and Sitaraman, 2019] such inconsistencies might be the result of the experimental arena structure (i.e. the material of the walls), the dividers between the light and dark parts as well as the source of lighting.

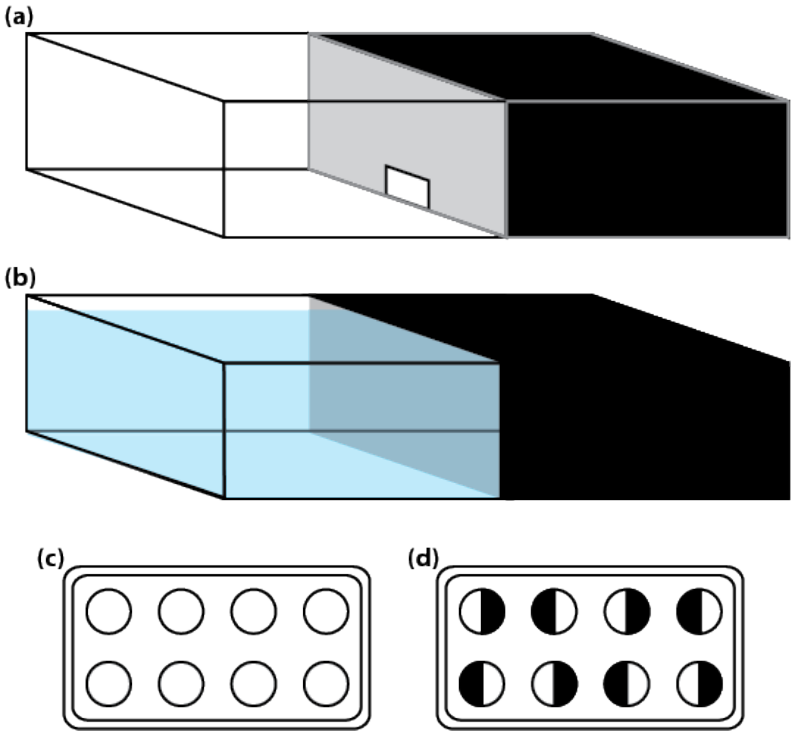


Figure 2.3: Typical configurations of light/dark preference task. **(a)** Commonly two transparent rectangle boxes, one of which is covered, are connected with a single passage in order to create the experimental structure of the light/dark task [Takao and Miyakawa, 2006]. **(b)** When the task is applied to zebrafish a single tank is used and half of it is covered to create the dark area. A separator with one or multiple entrances/exits may be used to separate the two areas [Liu and Sitaraman, 2019]. **(c)** Wells arranged in a plate format are used when the procedure involves zebrafish larvae. In this case each larvae is placed on a well and then the wells are placed in a room with a light switch [Schnörr et al., 2012]. (d) In the study of [Chhabria et al., 2019] half of each well was covered with cellophane films to create the 'dark' compartment [Chhabria et al., 2019].

2.2 Data clustering

Clustering is a branch of machine learning that aims to detect patterns in data. In a usual clustering scenario, given a set of data points a clustering algorithm is employed in order to partition them in distinctive groups, where each group is composed of elements as similar as possible [Aggarwal, 2014].

More specifically, clustering falls under the category of unsupervised learning where there is no prior knowledge of categories, i.e. class labels that specify distinctive groups [Gehring, 2018]. At this point there should be a clear distinction between the concepts of classification and clustering. The number of natural classes in a data set do not necessary equal to the number of natural clusters. As an example, we can consider the classification task between the handwritten digits 0 and 1 shown in Figure 2.4. Clearly we are having two distinctive classes and the classification task is to train a system to detect which digits are 0 and which are 1. However in a clustering framework each class would be split into multiple clusters because the structural similarities (or dissimilarities) of each class elements.

In clustering frameworks three aspects need to be specified: similarity, number of target clusters and clustering evaluation. A way to express similarity are the distances between the data points in a data set [Soler et al., 2013]. Distance is associated with the data attributes or features of the data set, i.e. how far away are

two data points between each other based on a distance metric such as Euclidean or Minkowski [Soler et al., 2013]. Target number of clusters K is a meta-parameter that requires tuning and there are various methods and algorithms that aim on finding an optimal value for K. Clustering evaluation is a method or criterion that is deployed to tune the meta-parameter in order to have the best possible clustering performance. This is usually achieved by running the clustering solution using different values for K and the one that leads to the best clustering is then adopted.

As a comparison, in supervised scenarios, e.g. neural networks classifying digits, we have a lot of meta-parameters that require tuning and a common concern is the overfitting of a classification system. Overfitting occurs when the system has learned to classify only a specific data set but its accuracy is not generalizable to unseen data even from the same source. To avoid overfitting we use the method of cross validation (shown in Figure 2.4). More details on this method will be provided later (refer to section 2.2.8) but in summary, during cross validation, a given data set is separated into training and test sets; the training set is used to train the system and the test set to test its classification quality with unseen data [Arlot et al., 2010]. This process is executed a number of times with different sections of the data as training and test sets.

The concept of cross validation is applicable to the area of Machine Learning that combines unsupervised and supervised learning and it is known as semi-supervised learning [Zhu and Goldberg, 2009]. More information about this type of learning as well as semi-supervised algorithms will be given in section 2.2.4. In general, semi-supervised systems learn by using both labelled (supervised) and unlabelled (unsupervised) data.

Next, there is going to be a discussion on the literature of clustering methods and techniques that are going to be used in this dissertation.

2.2.1 Feature engineering, selection and weighting

Feature engineering is an important process in clustering when domain knowledge of the data is available. Since clustering algorithms are detecting similarities within the data, it is useful to obtain measurements that transform these data in a way that the clustering algorithm can easily detect internal structures. Such structures are usually dense regions in the feature space which are formed by data having the same patterns thus are closer together from other data. Such measurements are the features (or data attributes or data variables) and can be "hand-crafted" based on the nature of the data. In the work of [Gehring et al., 2015] features have been designed from trajectory data recorded during a Morris Water Maze procedure in order to capture similar behavioural motifs. Performance measurements from other similar studies can also be used as features for the same purpose. Chapter 4 contains a section which is dedicated to feature engineering for path data recorded during experimental procedures.

Apart from feature engineering, feature selection is an important step of clustering and it is sometimes embedded inside a clustering algorithm. Because not all features are relevant on finding patterns in the data if all the available features are used during clustering this can have a negative effect on the clustering solution. Selecting the most important features is challenging and again depends on the application or prior knowledge about the data. Usually subsets of features are tested and the ones that led to the best clustering are adopted. A close related concept to feature

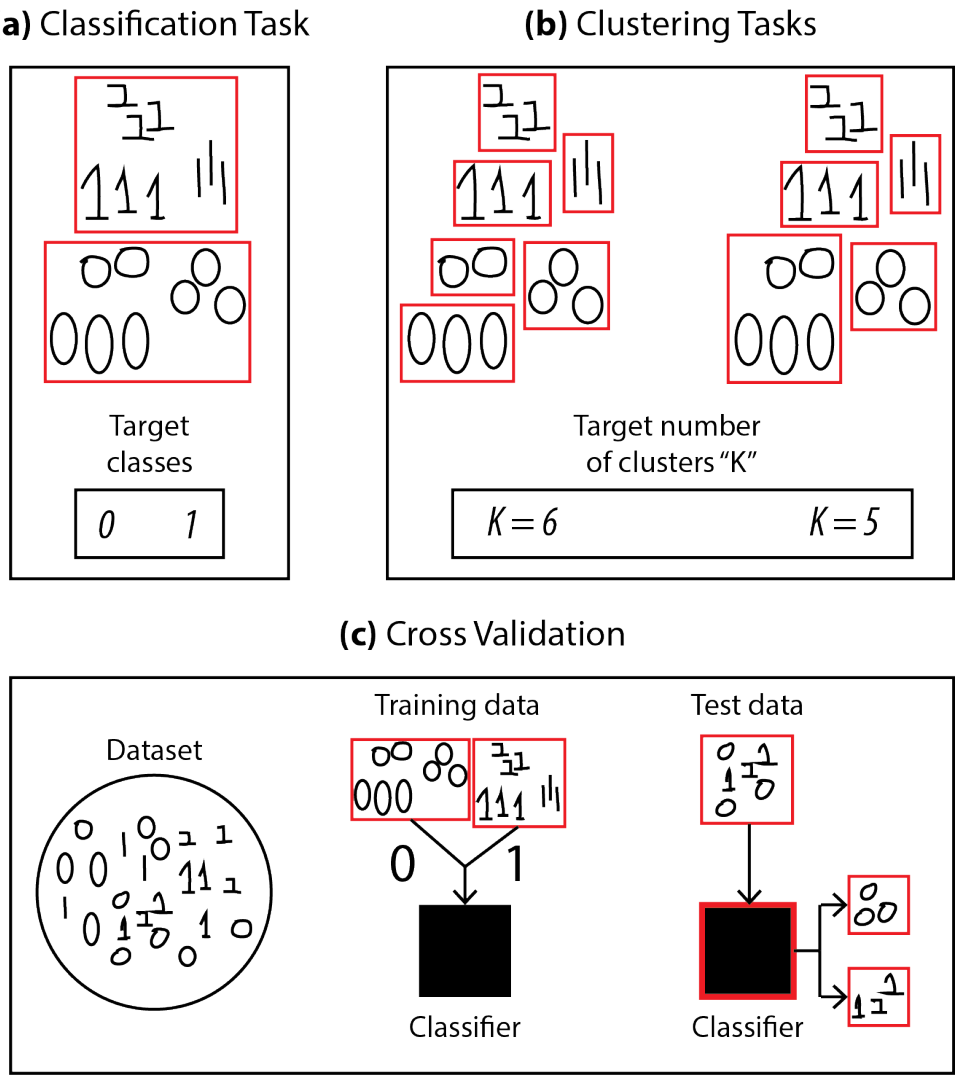


Figure 2.4: Classification and clustering procedures. **(a)** In a classification procedure there is prior knowledge of the target classes, in this case 0 and 1, and the task is to separate our data into the different classes. **(b)** In a clustering procedure there is no knowledge of the target classes and the data are separated into groups based on their similarities or dissimilarities. In such tasks the target number of groups (number of clusters, K) needs to be defined by the user. **(c)** Cross validation is a method to avoid overfitting in classification systems. A given data set is separated into training and test sets, the training set is labelled and is used to train a classification system, the testing set is unlabelled and it is used to assess the classification accuracy of the trained system. Cross validation cannot directly be used for clustering since there are no labels. For assessing the clustering quality there are various methods depending on the application, the clustering system and the nature of the data set.

selection is feature reduction where methods such as Principal Components Analysis (PCA) [Bro and Smilde, 2014] are used to reduce the dimensionality and capture the variance inside the data set by creating linear combinations of the initial features. The new features are likely to be more useful for separating the data into clusters and in the literature there are various examples of studies who used PCA and then fed the resulted, reduced dimensionality data sets to unsupervised algorithms [Ding and He, 2004; Honda et al., 2010; Park et al., 2008]. However, there is no guarantee that the new features will lead to better clustering [Chang, 1983]. Moreover, the new features are not directly interpretable.

Another method which can be considered as a generalization of feature selection

is feature weighting. In this dissertation two cases of feature weighting will be considered: (a) weights are applied to each feature in order to shape the feature space accordingly to achieve better clustering [Bilenko et al., 2004] (this concept is known as metric learning, refer to section 2.2.4); (b) feature assessment. The idea is that each feature receives a weight based on its contribution to the clustering [Modha and Spangler, 2003]. In this concept a completely uninformative feature will receive a weight of 0, thus it will be discarded (for an algorithm performing this procedure refer to section 2.2.3). Feature weighting can have benefits over the dimensionality reduction methods since the interpretation of the features is not lost, and it can rather be used to better describe the clustering outcome.

2.2.2 K-Means clustering

K-Means is the most well-studied clustering method for grouping data based on their similarities [Jain, 2010]. The problem of data grouping can be formulated by equation 2.1,

$$\mathcal{J}_{kmeans} = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 \tag{2.1}$$

where a set of data points is partitioned into K clusters so that equation 2.1 is minimized, i.e. for each cluster the distance (usually squared Euclidean) of every data point $x_{i:}$ to each respective centroid $m_{k:}$ is as small as possible. Equation 2.1 (also known as within-cluster-sum-of-squares, WCSS) specifies a convex function and setting $\frac{\partial \mathcal{J}_{kmeans}}{\partial m_{kj'}} = 0$, in order to minimize it, leading to the centers of the clusters $m_{1:}, \ldots, m_{K:}$ (for all the steps of this proof refer to Appendix A.2):

$$\frac{\partial \mathcal{J}_{kmeans}}{\partial m_{k'j'}} = 0 \Rightarrow \frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 = 0 \Rightarrow$$

$$m_{k'j'} = \frac{1}{n_{k'}} \sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} x_{ij'} \tag{2.2}$$

2.2.2.1 Alternative objective functions for K-Means clustering

In the literature [Witten and Tibshirani, 2010] there are some alternative objective functions for K-Means clustering such as the maximization of the between-cluster-sum-of-squares (BCSS) specified in equation 2.3,

$$\mathcal{J}_{BCSS} = \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - \mu_{1j})^2 - \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 \tag{2.3}$$

where the first term of the equation, $\sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - \mu_{1j})^2$, is a constant specifying the global WCSS (the squared Euclidean distance of all the points from the global centroid). Thus maximizing the function 2.3 is equivalent on minimizing the function 2.1. The maximization of 2.3 will be used latter for the Sparse K-Means algorithm.

In the same study of [Witten and Tibshirani, 2010] the authors also use equation 2.4 instead of equation 2.1,

$$\mathcal{J}'_{kmeans} = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \tag{2.4}$$

which uses the pairwise distances between the data points of each cluster. These two equations are equivalent (for the proof refer to Appendix A.3).

2.2.2.2 Lloyd's K-Means algorithm

The most common algorithm to minimize equation 2.3 is the Lloyd's K-Means algorithm described below [Jain, 2010]:

Lloyd's K-Means algorithm

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method.
2. Assign each data point to cluster c_{k^*} so that,
$$k^* = \underset{k}{argmin} \left\{ \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 \right\}$$
3. Recompute the cluster centroids,
$$m_{kj} = \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} x_{ij}$$
4. Go to step 2 until converge.

The algorithm returns the final clusters (centroids and element assignments).

2.2.2.3 Hartigan-Wong's K-Means algorithm

Hartigan-Wong's K-Means algorithm is an alternative to Lloyd's K-Means and in the study of [Slonim et al., 2013] it is shown that this method has lower probability of converging to a local minima solution compared to Lloyd's method in exchange of extra complexity. The algorithm starts by executing the first two steps of Lloyd's K-means algorithm and then proceeds as follows [Hartigan and Wong, 1979; Slonim et al., 2013]:

Hartigan-Wong's K-Means algorithm

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method.
2. Assign each data point to cluster k' so that,
$$k' = \underset{k}{argmin}\left\{ \sum_{j=1}^{p}(x_{ij} - m_{kj})^2 \right\}$$
3. Recompute the cluster centroids,
$$m_{kj} = \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} x_{ij}$$
4. Set an indicator $s = 0$.
5. For each data point $x_{i':}$:
   (a) Remove it from its cluster $c_{k'}$ and compute the within cluster sum of squares of $c_{k'}$,
   $$wcss_{k'} = \sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} \sum_{j=1}^{p}(x_{ij} - m_{k'j})^2 \tag{2.5}$$
   (b) For each cluster c_t where $t = 1, \cdots, K$ and $t \neq k'$:
      i. Temporarily assign $x_{i':}$ to c_t and compute the $wcss_t$ using equation 2.5 replacing k' with t.
      ii. If $wcss_t < wcss_{k'}$ set $s = t$ and $wcss_{k'} = wcss_t$.
   (c) If $s > 0$, assign $x_{i':}$ to cluster c_s, update the centroids $m_{k':}$ and $m_{s:}$, and set $s = -1$. Else assign $x_{i':}$ to its original cluster $c_{k'}$.
6. If $s \neq 0$, go to step 4. Else terminate.

The algorithm returns the final clusters (centroids and element assignments).

2.2.3 Sparse K-Means clustering

It is possible to define a the sparse clustering algorithm as a weighted version of the $BCSS$ (refer to equation 2.3) subject to certain constraints on the weights. One such method is proposed by Witten and Tibshirani [Witten and Tibshirani, 2010] and is given in equation 2.6,

$$\mathcal{J}_{skmeans} = \sum_{i=1}^{n}\sum_{j=1}^{p} w_j(x_{ij} - \mu_{1j})^2 - \sum_{k=1}^{K}\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k}\sum_{j=1}^{p} w_j(x_{ij} - m_{kj})^2 \Rightarrow \tag{2.6}$$

$$\mathcal{J}_{skmeans} = \sum_{j=1}^{p} w_j\gamma_j \quad , \text{with} \quad \gamma_j = \sum_{i=1}^{n}(x_{ij} - \mu_{1j})^2 - \sum_{k=1}^{K}\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k}(x_{ij} - m_{kj})^2 \tag{2.7}$$

$$\text{subject to} \qquad \sum_{j=1}^{p} w_j^2 \leq 1, \quad \sum_{j=1}^{p}|w_j| \leq s, \quad w_j \geq 0 \quad \forall j$$

where $\sum_{j=1}^{p} w_j^2$ is the L_2 penalty or ridge regression [Hoerl and Kennard, 1970] and $\sum_{j=1}^{p}|w_j|$ is the L_1 penalty or lasso regression [Tibshirani, 1996]. The minimization of the L_1 penalty will result in a constant shrinkage of the weights meaning that some weights will reach 0 (feature selection mechanism, see also Appendix A.7). On the other hand, the minimization of the L_2 penalty will result to proportional shrinkage of the weights meaning that the wights will never reach 0 (feature weighting mechanism, see also Appendix A.7). The parameter s is known as *sparsity* and regulates the amount of sparsity, i.e. how many weights will become 0 (for a graphical representation of L_1 and L_2 refer to the Appendix A.8).

In order to optimise equation 2.6 we can use a two stage optimization [Witten and Tibshirani, 2010]. The first step aims to optimise the BCSS holding the weights fixed and the second stage aims to optimise the weights holding the centroids fixed. The first stage is equivalent on minimising the weighted WCSS since the first term of equation 2.5 regarding the global centroid is fixed. The second stage can take the following form,

$$\underset{w_j}{maximize}\left\{\sum_{j=1}^{p} w_j\gamma_j\right\} \quad \text{subject to} \qquad \sum_{j=1}^{p} w_j^2 \leq 1, \;\; \sum_{j=1}^{p}|w_j| \leq s, \;\; w_j \geq 0 \;\; \forall j \tag{2.8}$$

Based on [Boyd and Vandenberghe, 2004; Witten et al., 2009] the solution to the convex problem of 2.8 is,

$$w_j = \frac{sign(\gamma_j)(|\gamma_j| - \Delta)_+}{\sqrt{\sum_{j'=1}^{p}\left(sign(\gamma_{j'})(|\gamma_{j'}| - \Delta)\right)^2}} \tag{2.9}$$

where the function $x_+ = \mathscr{H}(x) \cdot x$, where $\mathscr{H}$ is the Heaviside function, $x \in \mathbb{R}$ and there are the assumptions that γ_j has a unique maximum and that $1 \leq s \leq \sqrt{p}$ (for the proof of this solution and for an explanation of the upper bound of s refer to the Appendix A.8). There are two possibilities for the variable Δ, it can be either equal to 0 if that results in $\sum_{j=1}^{p}|w_j| \leq s$ or it has to be assigned with a positive value so that $\sum_{j=1}^{p}|w_j| = s$. In the latter case, to find an appropriate value for Δ the Bisection Search algorithm is used (description of this algorithm is provided latter as part of the Sparse K-Means).

An iterative algorithm for maximizing the function 2.6 is given by the Sparse K-Means (SK-Means) algorithm below:

Sparse K-Means algorithm [Witten and Tibshirani, 2010]

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method and the feature weights as $w_1 = \cdots = w_p = \frac{1}{\sqrt{p}}$.

2. Holding the weights fixed, maximize 2.6 with respect to M. This can be achieved by performing K-Means on the scaled data, i.e. multiply each feature j with $\sqrt{w_j}$.

3. Holding M fixed optimize equation 2.6 with respect to the weights applying the proposition given in equation 2.9. Choose $\Delta = 0$ if that

leads to $\sum_{j=1}^{p}|w_j| \leq s$, otherwise find $\Delta > 0$ that results in $\sum_{j=1}^{p}|w_j| = s$ To find Δ the Bisection algorithm can be used.

4. Go to step 2 until the convergence criterion in equation 2.10

$$\frac{\sum_{j=1}^{p}|w_j^r - w_j^{r-1}|}{w_j^{r-1}} < 10^{-4} \quad , \text{if } \; r > 1 \tag{2.10}$$

where r refers to the current iteration, and w_j^{r-1} to the weights of the previous iteration.

The algorithm returns the final clusters (centroids and elements) and the weight of each feature.

Bisection algorithm

1. Assume $lim_1 < \Delta < lim_2$, $lim_1 = 0$ and $lim_2 = max(\gamma_1, ..., \gamma_p)$
2. Compute $\Delta = \frac{lim_1 + lim_2}{2}$ and set

$$\begin{Bmatrix} lim_2 = \Delta & , & \text{if} & \sum_{j=1}^{p}|w_j| < s \\ lim_1 = \Delta & , & \text{if} & \sum_{j=1}^{p}|w_j| \geq s \end{Bmatrix}$$

3. If $lim_2 - lim_1 \geq 10^{-4}$ go to step 2.

2.2.4 Semi-supervised pairwise constrained learning

In the field of semi-supervised learning there are clustering algorithms that incorporate prior knowledge in the form of labels or constraints in order to achieve better clustering solutions [Bar-Hillel et al., 2003; Basu et al., 2003; Bilenko et al., 2004; Xing et al., 2003]. Specifically the study of Bilenko et al. [Bilenko et al., 2004] proposed a semi-supervised algorithms called Metric Pairwise Constrained K-Means (MPCK-Means) which learns a distance metric based on constraints imposed by labelled data points in the dataset. The constraints are imposed between pairs of points and can be either MUST-LINK, i.e. the two points must be in the same cluster or CANNOT-LINK, i.e. the two points must not be in the same cluster [Wagstaff et al., 2001].

The algorithm is a variant of the Pairwise Constrained K-Means (PCK-Means) algorithm [Basu et al., 2004] with metric learning [Bar-Hillel et al., 2003; Xing et al., 2003]. The PCK-Means algorithm incorporates constraints to guide the clustering solution, the constraints are considered as soft meaning that violations are permitted as opposed to its predecessor the COP-KMeans [Wagstaff et al., 2001] algorithm which stops if constraints violation is unavoidable. Metric learning is the adaptation of a distant metric to satisfy the similarity imposed by the pairwise constraints (supervised similarity [Bilenko et al., 2004]). These constraints may not results in separable clusters, thus a metric should be learnt to create distinctive clusters but at the same time satisfy the supervised similarity.

A key concept of both PCK-Means and MPCK-Means algorithms is the centroids

intialisation procedure known as seeding [Basu et al., 2002]. This PhD study will consider separately the centroid initialisation method from the clustering algorithm and description of seeding initialisation will be described in section 2.2.7.7.

2.2.4.1 Pairwise Constrained K-Means (PCK-Means)

Originally, the PCK-Means objective function is defined by equation 2.11 [Basu et al., 2004]

$$\begin{aligned}\mathcal{J}_{pckm} = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \Bigg(& \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 + \\ & \sum_{(x_{i:}) ML (x_{i':})} b_{x_i, x_{i'}} \ \mathbb{1}\left[(x_{i:}) \cancel{ML} (x_{i':})\right] + \\ & \sum_{(x_{i:}) CL (x_{i':})} \bar{b}_{x_i, x_{i'}} \ \mathbb{1}\left[(x_{i:}) \cancel{CL} (x_{i':})\right] \Bigg) \end{aligned} \tag{2.11}$$

where the second and third terms of the equation are two functions that indicate the severity of violating the imposed MUST-LINK and CANNOT-LINK constraints of the i-th element belonging to the k-th cluster; $\mathbb{1}$ is a boolean function that specifies if in case of a MUST-LINK ($(x_{i:}) ML (x_{i':})$) or CANNOT-LINK ($(x_{i:}) CL (x_{i':})$) constraint, this constraints has been violated; $\left[(x_{i:}) \cancel{ML} (x_{i':})\right]$ specifies violation of a MUST-LINK constraint and $\left[(x_{i:}) \cancel{CL} (x_{i':})\right]$ violation of a CANNOT-LINK constraint. The terms $b_{x_i, x_{i'}}$ and $\bar{b}_{x_i, x_{i'}}$ are providing a way of specifying individual costs for each constraint violation. A value of 0 for the cost terms will result to unsupervised K-Means clustering while a large value will lead to Constrained K-Means clustering [Basu et al., 2002], where the clustering process will be forced to respect all the given constraints [Basu et al., 2004]. Finally, an intermediate value will result in a tradeoff between minimizing the total distance of the data points to the cluster centroids and satisfying the constaints [Basu et al., 2004].

Specifying appropriate values for constraint costs can be challenging and requires extensive knowledge about the data set under analysis or the constraints quality. In the later study of [Bilenko et al., 2004] the authors specified two distance functions to calculate the amount of penalty for violating the constraints which can also been incorporated in equation 2.11 resulted in equation 2.12,

$$\begin{aligned}\mathcal{J}_{pckm} = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \Bigg(& \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 + \\ & \sum_{(x_{i:}) ML (x_{i':})} \sum_{j=1}^{p} b_{x_i, x_{i'}} (x_{ij} - x_{i'j})^2 \ \mathbb{1}\left[(x_{i:}) \cancel{ML} (x_{i':})\right] + \\ & \sum_{(x_{i:}) CL (x_{i':})} \sum_{j=1}^{p} \bar{b}_{x_i, x_{i'}} \left((x_{Ij} - x_{I'j})^2 - (x_{ij} - x_{i'j})^2\right) \ \mathbb{1}\left[(x_{i:}) \cancel{CL} (x_{i':})\right] \Bigg) \end{aligned} \tag{2.12}$$

where, based on the second term the severity of the penalty for violating a MUST-LINK constraint between the i-th element and another point $x_{i'j}$ distant from the i-th element is higher than if this pair of point were nearby. Analogously, based

on the third term, where the $x_{I:}$ and $x_{I':}$ is the maximally separated pair of points, the severity of the penalty for violating a CANNOT-LINK constraint between the i-th element and another point near the i-th element is higher than if this pair of points were far from each other. The terms $b_{x_i,x_{i'}}$ and $\bar{b}_{x_i,x_{i'}}$ are still providing a way of specifying individual costs for each constraint violation but, for the rest of this thesis, they will be assumed to have the constant value of 1.

Conceptually, the way of specifying the severity of constraints violation in equation 2.12 is beneficial for metric learning (such as in the case of MPCK-Means algorithm that will be described afterwards) [Bilenko et al., 2004], where we have to estimate how severe modification is needed for a metric to satisfy the constraints. In the case of PCK-Means objective given by equation 2.12, if two data points are far from each other and are having a MUST-LINK constraint or if two data points are close to each other and are having a CANNOT-LINK constraint then in both cases these constraints are likely to be violated within the K-Means iterations thus a huge penalty needs to be applied in order to compensate these violations.

Minimizing equation 2.12 leads to the centers of the clusters similar to K-Means,

$$\frac{\partial \mathcal{J}_{pckm}}{\partial m_{k'j'}} = 0 \Rightarrow \frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 = 0 \Rightarrow$$

$$m_{k'j'} = \frac{1}{n_{k'}} \sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} x_{ij'}$$

An iterative algorithm for minimizing the function 2.12 is given by the algorithm below:

Pairwise Constrained K-Means algorithm

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method.

2. Assign each data point to cluster k^* so that,

$$k^* = \underset{k}{argmin}\Bigg\{\Bigg(\sum_{j=1}^{p}(x_{ij} - m_{kj})^2 + \sum_{(x_{i:})ML(x_{i':})}\sum_{j=1}^{p} b_{x_i,x_{i'}}(x_{ij} - x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big] + \sum_{(x_{i:})CL(x_{i':})}\sum_{j=1}^{p} \bar{b}_{x_i,x_{i'}}\big((x_{Ij} - x_{I'j})^2 - (x_{ij} - x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)\Bigg\} \tag{2.13}$$

3. Recompute the cluster centroids,

$$m_{kj} = \frac{1}{n_k}\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} x_{ij}$$

4. Go to step 2 until converge.

The algorithm returns the final clusters (centroids and elements).

To avoid any confusion with the study of [Bilenko et al., 2004], in the implementation of the latter, PCK-Means uses the semi-supervised seeding method to initialise the cluster centroids.

2.2.4.2 Metric Pairwise Constrained K-Means (MPCK-Means)

Integrating the metric learning to the PCK-Means objective function in equation 2.12 results in the objective function of the MPCK-Means algorithm given by equation 2.14 [Bilenko et al., 2004],

$$\mathcal{J}_{mpckm} = \sum_{k=1}^{K}\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k}\Bigg(\sum_{j=1}^{p} a_j(x_{ij} - m_{kj})^2 - \sum_{j=1}^{p} log(a_j) + \sum_{(x_{i:})ML(x_{i':})}\sum_{j=1}^{p} b_{x_i,x_{i'}} a_j(x_{ij} - x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big] + \sum_{(x_{i:})CL(x_{i':})}\sum_{j=1}^{p} \bar{b}_{x_i,x_{i'}}\big(a_j(x_{Ij} - x_{I'j})^2 - a_j(x_{ij} - x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg) \tag{2.14}$$

where a_j is a weight on the j-th dimension of the features and $\sum_{j=1}^{p} log(a_j)$ is a normalization constant [Xing et al., 2003] that does not allow the weights to grow to large.

In order to minimize equation 2.14 we can use a two stage optimization [Bilenko et al., 2004]. The first stage is the minimization of equation 2.14 with respect to the centroids, which results in the standard K-Means algorithm, and the second stage with respect to the weights. The second optimization step has the following form,

$$a_j = n\Bigg(\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\Bigg((x_{ij}-m_{kj})^2+ \sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij}-x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+ \sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big(a_j(x_{Ij}-x_{I'j})^2-(x_{ij}-x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)\Bigg)^{-1}$$

(for more details on how these equations were derived refer to Appendix A.4).

An iterative algorithm for minimizing the function 2.14 is given by the algorithm below [Bilenko et al., 2004]:

MPCK-Means algorithm

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method and W as a diagonal matrix with values $w_1 = \cdots = w_p = 1$.

2. Assign each data point to cluster k^* so that,

$$k^* = \underset{k}{argmin}\Bigg\{\Bigg(\sum_{j=1}^{p} a_j(x_{ij}-m_{kj})^2 - \sum_{j=1}^{p} log(a_j)+ \sum_{(x_{i:})ML(x_{i':})}\sum_{j=1}^{p} b_{x_i,x_{i'}}a_j(x_{ij}-x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+ \sum_{(x_{i:})CL(x_{i':})}\sum_{j=1}^{p} \bar{b}_{x_i,x_{i'}}\big(a_j(x_{Ij}-x_{I'j})^2 - a_j(x_{ij}-x_{i'j})^2\big) \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)\Bigg\} \quad (2.15)$$

3. Recompute the cluster centroids,

$$m_{kj} = \frac{1}{n_k}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k} x_{ij}$$

4. Update the weights $\forall j$,

$$a_j = n\Bigg(\sum_{k=1}^{K}\sum_{\substack{i=1\\ x_{i:}\in c_k}}^{n_k}\Bigg((x_{ij}-m_{kj})^2+ \sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij}-x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+ \sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big(a_j(x_{Ij}-x_{I'j})^2-(x_{ij}-x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)\Bigg)^{-1} \tag{2.16}$$

5. Go to step 2 until converge. Various criteria can be used for convergence e.g. maximum number of iteration reached or minimum changes in the objective function.

The algorithm returns the final clusters (centroids and elements) and feature weights. The weights correspond to the learnt metric that shapes the feature space accordingly to satisfy the input constraints.

To avoid any confusion with the study of [Bilenko et al., 2004], in the implementation of the latter, MPCK-Means uses the semi-supervised seeding method to initialise the cluster centroids. For this PhD study it is considered that a single metric is used for all clusters which is parameterized by a diagonal matrix. This corresponds to feature weighting where each feature f_j of a $p-$dimensional data set is multiplied with the corresponding element of the diagonal of the matrix A_{jj} (for more information refer to the Appendix A.1).

2.2.5 K-Medians clustering

A variation K-Means objective function shown in equation 2.1 can be the minimisation of the function 2.17 for each dimension j,

$$\mathcal{J}_{kmedians} = \sum_{k=1}^{K}\sum_{\substack{i=1\\ x_{i:}\in c_k}}^{n_k}\sum_{j=1}^{p}|x_{ij}-m_{kj}| \tag{2.17}$$

It can be shown that minimizing equation 2.17 leads to the median of each dimension of the clusters (refer to Appendix A.5). K-Medians corresponds to the L_1-norm (taxicab or Manhattan distances) as opposed to the L_2-norm of K-Means [Aggarwal, 2014]. The benefits of K-Medians is the use of the median to compute the cluster centroids instead of the mean that K-Means uses. The median is a robust to outliers statistic [Feldman and Schulman, 2012] and has a breaking point of 0.5, i.e. even if half of the data set is corrupted by outliers the median of the corrupted data set will be similar to the median of the original data set [Lopuhaa et al., 1991].

2.2.5.1 K-Medians algorithm

An iterative process for minimising equation 2.17 is given by an algorithm similar to Lloyd's K-Means,

K-Medians algorithm

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method.
2. Assign each data point to cluster k^* so that,

$$k^* = \underset{k}{argmin}\left\{ \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 \right\}$$

3. Recompute the cluster centroids by taking the median on each dimension of the data points assigned to them.
4. Go to step 2 until converge.

The algorithm returns the final clusters (centroids and element assignments).

2.2.6 Geometric K-Medians clustering

In the literature there is another variant of K-Means that uses the geometric median instead of the median on each dimension by minimizing the equation 2.18

$$\mathcal{J}_{gkmedians} = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \left| \sum_{j=1}^{p} (x_{ij} - m_{kj}) \right| \tag{2.18}$$

It can be shown that minimizing equation 2.17 leads to the following expression for the clusters centroids, (refer to Appendix A.6 for all the steps),

$$\frac{\partial \mathcal{J}_{gkmedians}}{\partial m_{k'j'}} = \frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \left| \sum_{j=1}^{p} (x_{ij} - m_{kj}) \right| = 0$$

$$\Rightarrow m_{k'j'} = \frac{\sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} \frac{x_{ij'}}{\sqrt{(x_{ij'} - m_{k'j'})^2}}}{\sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} \frac{1}{\sqrt{(x_{ij'} - m_{k'j'})^2}}} \tag{2.19}$$

2.2.6.1 Weiszfeld's algorithm

With 1-dimensional data sets the median is similar to the geometric median [Haldane, 1948]. However, in high dimensions there is no close form for the geometric median [Whelan et al., 2015], thus one can use an iterative form of equation 2.19 known as the Weiszfeld's algorithm. In this algorithm the $m_{kj}^{(l)}$ term is the l-th estimate of m_{kj} and $l = 1, \ldots, n_k$.

Weiszfeld's algorithm

1. Initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method.

2. Assign each data point to cluster k^* so that,

$$k^* = \underset{k}{argmin}\left\{ \sum_{j=1}^{p}(x_{ij} - m_{kj})^2 \right\}$$

3. Recompute the cluster centroids using the Weiszfeld's algorithm,

   (a) For each cluster k and dimension j:

   (b) Initialise the k-th centroid,

$$m_{kj} = \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} x_{ij}$$

   (c) Update the centroid estimation, $m_{kj}^{(l)} = \frac{\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \frac{x_{ij}}{\sqrt{\sum_{j=1}^{p}(x_{ij} - m_{kj}^{(l)})^2}}}{\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \frac{1}{\sqrt{\sum_{j=1}^{p}(x_{ij} - m_{kj}^{(l)})^2}}}$, $l = 1, \ldots, n_k$

4. Go to step 2 until converge.

The algorithm returns the final clusters (centroids and element assignments).

2.2.7 Clustering initialisation methods

In the literature there are various studies regarding the importance of the initial cluster centroids for the performance of the K-Means algorithm [Jain, 2010] and extensive testing on various initialisation techniques [Celebi et al., 2013; Fränti and Sieranoja, 2019]. In this section there is going to be a description of the techniques that will appear in the PhD thesis. Benchmarking on unsupervised and semi-supervised K-Means variations using each one of these methods is available in Chapter 3.

Before proceeding to the initialisation methods we will introduce the following notation. Let $D(x_{i:})$ to denote the distance between data point $x_{i:}$ and the nearest of the selected cluster centroids, $m_{k:}$, $k = 1, \ldots, L$, with L being the number of selected centroids ($L \leq K$):

$$D(x_{i:}) = \underset{k}{min} \sqrt{\sum_{j=1}^{p}(x_{ij} - m_{kj})^2}.$$

2.2.7.1 Random

The initialisation method of [MacQueen et al., 1967] proposes a random selection of data points from the data set which will be the initial centroids. This is one of the earliest clustering initialisation techniques and an improvement of Jancey's method [Jancey, 1966]. The latter study suggested the centroids to be at random locations within the hypersphere of the data set but this might result in empty clusters to be generated after the execution of the K-Means algorithm.

2.2.7.2 K-Means++

K-Means++ [Arthur and Vassilvitskii, 2007] is a standard clustering initialisation technique in many programming languages, such as MATLAB and Python. It has linear complexity $\mathcal{O}(N)$ and it uses a probabilistic approach in order to select initial centroids data points that are well separated. The steps of this algorithm are as follows:

1. Select randomly a data point $x_{i:}$ as the first centroid $m_{1:}$ and set $k = 2$.
2. Choose another data point $x_{i':}$ as the next centroid $m_{k:}$ with probability

$$p(x_{i':}) = \frac{D(x_{i':})^2}{\sum_{i=1}^{n} D(x_{i:})^2}$$

   and set $k = k + 1$.
3. While $k \leq K$ go to step 2.

2.2.7.3 Maximin

The Maximin method of [Gonzalez, 1985] picks data points as cluster centroids that are far apart form each other. The steps in the algorithm are:

1. Select randomly a data point $x_{i:}$ as the first centroid $m_{1:}$ and set $k = 2$.
2. Select as the next centroid $m_{k:} = x_{i':}$ with $i' = \underset{i}{argmax}\{D(x_{i:})\}$ and set $k = k + 1$.
3. While $k \leq K$ go to step 2.

Maximin has linear complexity $\mathcal{O}(N)$. The study of [Katsavounidis et al., 1994] proposed a modification in the first step of the algorithm to select as the first centroid the data point with the maximum Euclidean norm [Celebi et al., 2013]. In this way the method can become deterministic.

2.2.7.4 Kaufman

Kaufman and Rousseeuw [Kaufman and Rousseeuw, 2009] proposed a deterministic method for centroids initialisation. Their method is as follows [Pena et al., 1999]:

1. Select the closest data point to the global centroid of the data set as the first centroid $m_{1:}$ and set $k = 2$.
2. For every two non-selected data points $x_{i:}$ and $x_{i':}$ calculate,

$$C_{i'i} = max\left\{D(x_{i:}) - \sqrt{\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2}, 0\right\}.$$

3. Select as the next centroid $m_{k:} = x_{i^*:}$, with $i^* = \underset{i}{argmax}\{\sum_{i'} C_{i'i}\}$ and set $k = k + 1$.

4. While $k \leq K$ go to step 2.

Kaufman's and Rousseeuw's algorithm has quadratic complexity $\mathscr{O}(N^2)$ because of the computation of the pairwise distances [Celebi et al., 2013].

2.2.7.5 ROBust INitialisation (ROBIN)

ROBIN [Al Hasan et al., 2009] is an initialisation method that is robust to outliers. It uses the Local Outlier Factor (LOF) [Breunig et al., 2000] in order to select as initial centroids data points that are far away from each other and also representative points of dense regions in the data set. In addition it requires one more tuning parameter which is the number of neighboring data points mp to be consider when computing the LOF of each data point. The LOF score of each data point, $LOF(x_{i:}, mp)$, is given by the algorithm below [Al Hasan et al., 2009]: $N(x_{i:}, mp)$ is the set of the mp nearest data points to the $x_{i,:}$ data point, with $|N(x_{i:}, mp)| \geq mp$.

1. Compute the density of each data point $x_{i:}$,

$$density(x_{i:}, mp) = \frac{|N(x_{i:}, mp)|}{\sum_{x_{i':} \in N(x_{i:}, mp)} \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}}, i \neq i'. \tag{2.20}$$

2. Compute the average relative density of each data point $x_{i:}$,

$$ard(x_{i:}, mp) = \frac{density(x_{i:}, mp) \;\; |N(x_{i:}, mp)|}{\sum_{x_{i':} \in N(x_{i:}, mp)} density(x_{i':}, mp)}. \tag{2.21}$$

3. Compute the LOF score of each data point $x_{i:}$,

$$LOF(x_{i:}, mp) = \frac{1}{ard(x_{i:}, mp)}. \tag{2.22}$$

The ROBIN algorithm ($K > 1$) is described below [Al Hasan et al., 2009]:

1. Pick a reference data point $x_{r:}$ from the data set.
2. Sort the data points in decreasing order of their distance from $x_{r:}$.
3. For each $x_{i:}$ in sorted order, pick the first data point $x_{i':}$ for which $LOF(x_{i':}, mp) \approx 1$ as the first centroid $m_{1:}$ and set $k = 2$.
4. Sort the data points in decreasing order based on $D(x_{i:})$.
5. For each $x_{i:}$ in sorted order, pick the first data point $x_{i':}$ for which $LOF(x_{i':}, mp) \approx 1$ as the next centroid $m_{k:}$ and set $k = k + 1$.
6. While $k \leq K$ go to step 4.

The computational cost of this method is dominated by the complexity of sorting, which is $\mathscr{O}(N \; log \; N)$ [Celebi et al., 2013] but for the LOF score calculation we have a choice of algorithms varying from $\mathscr{O}(N)$ to $\mathscr{O}(N^2)$, that can be chosen based on dimensionality-related constraints, see [Breunig et al., 2000]. Regarding the 4^{th} step of the algorithm, in an R implementation (refer to the study of [Brodinová et al., 2017])

the formula $LOF(x_{r':}, mp) < 1.05$ was used but since the LOF score can also be lower than 1, in our experiments, we used the formula $1 - \epsilon < LOF(x_{r':}, mp) < 1 + \epsilon$ where ϵ was set to 0.05. In the original algorithm [Al Hasan et al., 2009] the authors used the algorithm in a deterministic manner by setting the reference point on step 2, x_r : to the origin. In the R implementation of [Brodinová et al., 2017] the reference point is chosen at random. In this study we test both methods, *ROBIN(S)* will refer to the stochastic method of [Brodinová et al., 2017] while *ROBIN(D)* will refer to the deterministic method of [Al Hasan et al., 2009].

2.2.7.6 Density K-Means++ (DK-Means++)

DK-Means++ [Nidheesh et al., 2017] is a deterministic method for centroids initialisation based on data density. It is an improved method of [Lan et al., 2015; Rodriguez and Laio, 2014] since it requires only to define the number of clusters K and utilizes a heuristic to detect dense regions in the data set based on a radius ε in order to select optimal centroids. The radius ε can be computed form the following algorithm [Nidheesh et al., 2017]:

1. Construct the minimum spanning tree of the distance matrix of the data set.
2. Let Λ be the weights of edges of the Minimum Spanning Tree and IQR the Inter Quartile Range. Then,

$$\varepsilon = 3 \cdot IQR(\Lambda) + 75^{th} percentile(\Lambda).$$

The DK-Means++ algorithm is described below [Nidheesh et al., 2017]:

1. Compute the the local density $p(x_{i:})$ of each data point using the formula:

$$p(x_{i:}) = \sum_{x_{i'} \in \varepsilon\text{-}neighbors(x_{i:})} \exp\left(\frac{-\sqrt{\sum_{j=1}^{p}(x_{ij} - x_{i'j})^2}}{\varepsilon}\right).$$

   where $\varepsilon\text{-}neighbors(x_{i:})$ are the data points falling under the hypersphere with centroid $x_{i:}$ and radius ε.

2. Normalize $p(x_{i:})$ using the *min-max* normalization.
3. The first cluster centroid $m_{1:}$ is the data point $x_{i^*:}$ for which $p(x_{i^*:}) = max\{p(x)\}$. Then $m_{1:} = x_{i^*:}$ and $k = 2$.
4. Compute the prospectiveness all data points that are not selected as centers given the formula, $\phi(x_{i:}) = p(x_{i:}) \cdot D(x_{i:})$.
5. The next centroid $m_{k:}$ is the data point with maximum prospectiveness: $m_{k:} = x_{i^*:}$ with $i^* = \underset{i}{argmax}\{\phi(x_{i:})\}$ and $k = k + 1$.
6. While $k \leq K$ go to step 4.

The computation of $\varepsilon\text{-}neighbors$ contributes to the complexity of DK-Means++. It is dominated by the distance matrix computation, which is $\mathcal{O}(N(N-1)/2)$. The computation of the Minimum Spanning Tree depends of the algorithm used to compute it and varies from $\mathcal{O}(N\ log\ N)$ (Kruskal's algorithm) to $\mathcal{O}\big((2N-1)\ log\ N\big)$ (Prim's algorithm) [Moret and Shapiro, 1992].

2.2.7.7 Semi-supervised initialisation: the seeding method

In the studies of [Basu et al., 2004; Bilenko et al., 2004] the PCK-Means and MPCK-Means algorithms are initialized using a semi-supervised procedure based on the MUST-LINK constraints imposed to the data set that specify which elements should belong to the same cluster. This method is known as seeding [Basu et al., 2002].

In more detail, the given set of MUST-LINK constraints is first augmented using the transitive closure [Flaška et al., 2007] that infers additional constraints based on the existing relationship between the data points, e.g. if $x_{1:}$ MUST-LINK $x_{2:}$ and $x_{2:}$ MUST-LINK $x_{3:}$ then $x_{1:}$ MUST-LINK $x_{3:}$. These kind of inferred relationships are discovered using the depth-first search algorithm [Korf, 1985] and form $\Xi = \{\xi_1, \xi_2, \ldots, \xi_\Omega\}$ *neighborhoods*. The given set of CANNOT-LINK constraints is then augmented with additional constraints between pairs of points in neighborhoods that have at least one pair of CANNOT-LINK constraints, e.g if $(x_{1:} \in \xi_1)$ CANNOT-LINK $(x_{1:} \in \xi_2)$ then CANNOT-LINK constraints are created between every pair of points $(x_{i:} \in \xi_1)$ and $(x_{i':} \in \xi_2)$.

After this augmentation step, the centroids of each neighborhood are computed by taking the mean of the elements assigned to each neighborhood to obtain Ω centroids, $m_{1:}, m_{2:}, \ldots, m_{\Omega:}$. If K target number of clusters are required, thus K initial centroids, then MPCK-Means executes one of the following options:

- If $\Omega = K$: Every neighborhood centroid becomes an initial cluster centroid.
- If $\Omega > K$: K neighborhoods are picked based on a weighted farthest-first traversal [Gonzalez, 1985]. The latter is a process where a point is selected arbitrarily and each successive point is selected so that maximum distance between the newly selected point and the previous selected points is achieved. The weighted form of this process is based on the size of the neighborhoods meaning that we are searching for neighborhood centroids that are both far away from each other but also represent large neighborhoods. The weighted distance metric that determines the next neighborhood based on its centroid $m_{\omega:}$ and size n_{m_ω} from a previous neighborhood with centroid $\xi_{(\omega-1):}$ and size $n_{\xi_{\omega-1}}$ is given by the equation 2.23

$$dist = \sqrt{\sum_{j=1}^{p} (m_{\omega j} - m_{(\omega-1)j})^2} \cdot \sqrt{n_{\xi_\omega} \cdot n_{\xi_{\omega-1}}} \tag{2.23}$$

  This heuristic is favoring K-Means-based clustering algorithms since it promotes initial cluster centers that are far apart from each other but also represent large portions of the dataset.
- If $\Omega < K$: Every neighborhood centroid becomes an initial cluster centroid and the rest $K - \Omega$ centroids are generated at random. In the MPCK-Means implementation the number 42 is used as a random seed in order to have deterministic results [Bilenko et al., 2004].

2.2.8 Clustering tuning and performance evaluation

Clustering tuning refers to the selection of values for the meta-parameters of the algorithm. Since different turnings lead to different solutions for the clustering

problem these solutions are then evaluated with one or more validity methods. The solution that yields the best performance is then adopted.

The validity methods fall under two distinctive groups, external validation methods and internal validation methods [Rendón et al., 2011]. External methods are using previous knowledge about the data to assess the clustering solutions and such knowledge is commonly the ground truth of the data set, i.e. the cluster of each element. Internal methods are using only intrinsic to the data information such as clusters compactness or separability to access the clustering solution [Kovács et al., 2005; Rendón et al., 2011].

2.2.8.1 Internal validity methods

2.2.8.1.1 The elbow method

The most simplistic method to validate a clustering solution is to compare the values of the objective function that the algorithm tries to minimize or maximize. In the K-Means algorithm the only meta-parameter is the target number of clusters K. Assuming that the goal is the minimization of the objective function then for a range of different k values the K-Means algorithm WCSS can be plotted against the different k values. The assumption is that a value for the number of clusters far less than the "optimal" number of clusters of the data set will have a significant larger WCSS than a value further away from the "optimal" number of clusters [Hardy, 1994]. However, as we are getting closer to the "optimal" then there is going to be a sudden drop of the WCSS value which will result on the "elbow" point to appear. That value will be a good candidate for the number of target clusters. Due to its simplicity, the elbow method can provide ambiguous results and is not very accurate especially in cases where there is not well-separable cluster as shown in Figure 2.5.

2.2.8.1.2 The silhouette index

The silhouette index [Rousseeuw, 1987] is a value that specifies the degree of similarity between a data point and other data points of the same cluster and the dissimilarity between a data point and other data points in different clusters. The silhouette index of the data point $x_{i:} \in c_k$ is given by the formula 2.24[Starczewski and Krzyżak, 2015]

$$S_{x_{i:}} = \frac{b_{x_{i:}} - a_{x_{i:}}}{max\{b_{x_{i:}}, a_{x_{i:}}\}} \tag{2.24}$$

where $a_{x_{i:}}$ is the average distance of $x_{i:}$ and all the other data points in the cluster that $x_{i:}$ belongs to,

$$a_{x_{i:}} = \frac{1}{n_k - 1} \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2} \tag{2.25}$$

and $b_{x_{i:}}$ is the minimum average distance of $x_{i:}$ and all the other data points in other clusters,

$$b_{x_{i:}} = \min_{k'} \frac{1}{n_{k'}} \sum_{\substack{i'=1 \\ (x_{i':} \in c_{k'})}}^{n_{k'}} \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}, k' \neq k \tag{2.26}$$

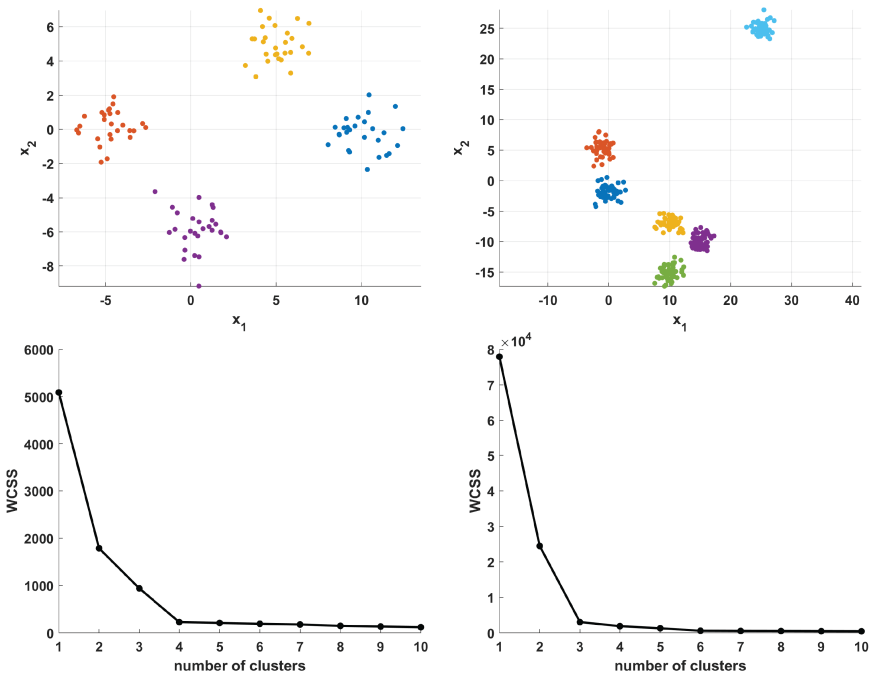


Figure 2.5: The elbow method for tuning the target number of clusters meta-parameter. Two examples of the elbow method. The upper plots illustrate the given data sets where the distinctive clusters are marked with separate colors. The bottom plots illustrate the WCSS values resulted by running, on each data set, the Lloyd's K-Means algorithm using the DK-Means++ initialization procedure and different values for K. Based on the elbow method, the correct number of clusters is the value for K where there is a clear "elbow" in the graphs. In the left-hand scenario the correct number of clusters can be identified correctly as 4 using the elbow method. However in the right-hand scenario the elbow method indicates wrongly that K should be equal to 3 instead on 6 because based on that criterion the clusters are not well-separable.

The silhouette index of each cluster can then be specified as the average silhouette index of the data points that belongs to it,

$$S_{c_k} = \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} S_{x_{i:}} \tag{2.27}$$

Finally there are two different formulas for defining the overall silhouette index of a clustering solution. The first formula is the average silhouette index of the clusters define as,

$$S = \frac{1}{K} \sum_{k=1}^{K} S_{c_k} \tag{2.28}$$

an the second one (which is the original based on [Rousseeuw, 1987]) is the average silhouette of all the data points in the data set,

$$S' = \frac{1}{n} \sum_{i=1}^{n} S_{x_{i:}} \tag{2.29}$$

In the study of [Starczewski and Krzyżak, 2015] the performance of the two formulas for the silhouette index of a clustering solution 2.28 and 2.29 has been tested and the formula 2.29 was empirically superior to 2.28 on finding the right

number of clusters in a data set. An example of the silhouette method is depicted in Figure 2.6. An advantage of the silhouette over other methods is that it is bounded between -1 and 1, where 1 specifies maximum separation of clusters and maximum within cluster density while other indexes, such as the distortion score (or the WCSS) gives only an estimation of the latter.

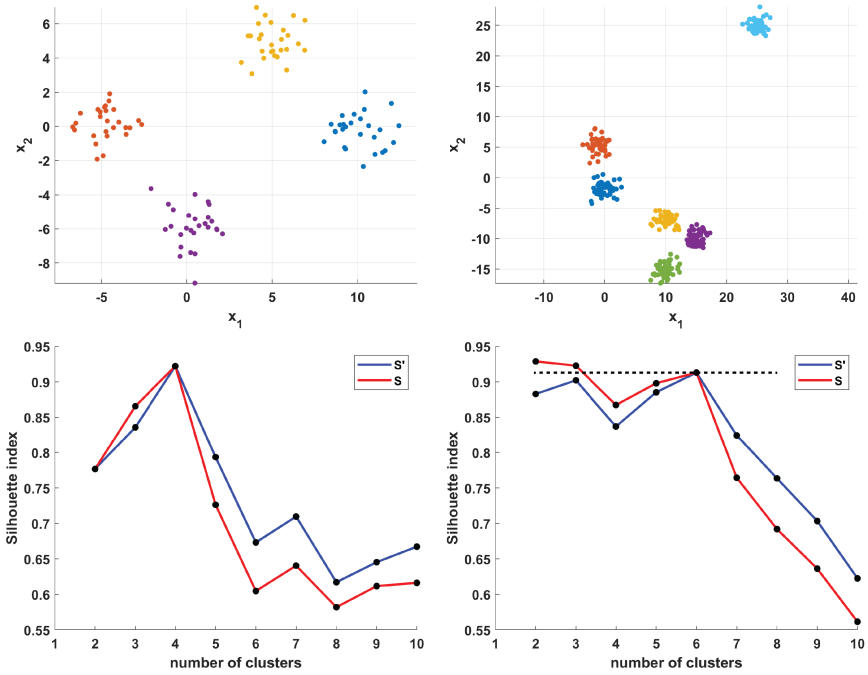


Figure 2.6: The silhouette method for tuning the target number of clusters meta-parameter. Two examples of the silhouette method. The upper plots illustrate the given data sets where the distinctive clusters are marked with separate colors. The bottom plots illustrate the two formulas for the silhouette method, $S = \frac{1}{K}\sum_{k=1}^{K} S_{c_k}$ (2.28) and $S' = \frac{1}{n}\sum_{i=1}^{n} S_{x_i:}$ (2.29). The number of clusters for which the silhouette index is maximized specifies the best clustering model. The superiority of the formula 2.29 as indicated in [Starczewski and Krzyżak, 2015] can be seen in the right hand side where the silhouette method using the formula 2.28 provides a wrong indication for the number of clusters (red line at number of clusters equal to 3 above the dashed line) while the formula 2.29 correctly identifies the number of clusters to be 6 (blue line at number of clusters equal to 3 below the dashed line).

2.2.8.1.3 The gap statistic

The gap statistic compares the within-cluster dispersion with that of a reference distribution [Tibshirani et al., 2001]. For the K-Means clusters the within-cluster dispersion is the within-cluster-sum-of-squares given in Equation 2.1. Here we will assume any algorithm resulting in a solution that optimises the within-cluster dispersion $\mathcal{J}$. For the reference distribution the authors of [Tibshirani et al., 2001] propose two methods:

- Generate each feature of the original data set $\mathcal{X}$ uniformly over the range of the observed values that each feature has.
- Same as above but using the transformed data set $\mathcal{X}^*$ which is generated by centering the data of $\mathcal{X}$ around their mean (resulting to $\hat{\mathcal{X}}$), applying

singular value decomposition on the centered data, $\hat{\mathcal{X}} = USV^T$ and create the transformed data set $\mathcal{X}^* = \hat{\mathcal{X}}V$.

The second method is more complex, but accounts for the shape of the data distribution, and it is expected to be better in practice [Yan and Ye, 2007]. The gap statistic is then given by the formula,

$$Gap(k) = E_B\{\log(\mathcal{J}_k^*)\} - log(\mathcal{J}_k) \tag{2.30}$$

where k is the number of target clusters, $log(\mathcal{J}_k)$ is the logarithmic within-cluster dispersion obtained by running the clustering algorithm on the original data set and $E_B\{\log(\mathcal{J}_k^*)\}$ is the expected logarithmic within-dispersion which is obtained by running the clustering algorithm on B reference data sets, generated using either of the methods explained before, and taking the average of the resulted (logarithmic) within-dispersions,

$$E_B\{\log(\mathcal{J}_k^*)\} = \frac{1}{B}\sum_{b=1}^{B}\log(\mathcal{J}_k^*(b)) \tag{2.31}$$

Equation 2.30 is repeated for different values for k. The choice of k^* is the smallest k such that,

$$Gap(k) \geq Gap(k+1) - \mathcal{S}_{k+1} \tag{2.32}$$

where $\mathcal{S}_k = \sqrt{1+\frac{1}{B}} \cdot \sigma_k\big(E_B\{\log(\mathcal{J}_k^*)\}\big)$ (σ stands for the standard deviation) and accounts for the simulation error.

Overall the gap statistic method aims to determine the appropriate elbow point of the within-cluster dispersion curve (refer to 2.2.8.1.1). This is achieved by comparing the within-cluster dispersion of the clustering algorithm on the original data for some k with that of reference data with the expectation that $log(\mathcal{J}_k)$ will decrease faster for $k \leq k^*$ and more slowly for $k > k^*$. If $log(\mathcal{J}_k)$ falls farthest below the expectation $E_B\{\log(\mathcal{J}_k^*)\}$ then we have the best estimate for $k*$; this implies that $k*$ maximises $Gap(k)$. The stopping criterion in Equation 2.32 applies the 1-standard-error rule which empirically works well according to the authors [Tibshirani et al., 2001].

Some modification of the gap statistic have been proposed. The study of [Yan and Ye, 2007] proposes the weighted gap statistic $\bar{Gap}(k)$ which is applied on the weighted within-cluster dispersion: Let $\mathcal{J}_k = \sum {}^1\mathcal{J}_k \cdots {}^k\mathcal{J}_k$ to be the summation of the within-cluster dispertion of each cluster; the weighted version is then $\bar{\mathcal{J}}_k = \sum \frac{1}{2n_k(n_k-1)}({}^1\mathcal{J}_k \cdots {}^k\mathcal{J}_k)$. The study of [Mohajer et al., 2011] compares the gap statistic definitions with and without the logarithm. Furthermore the study of [Witten and Tibshirani, 2010] proposes a modification of the gap statistic for the tuning of the sparsity parameter s. Since in sparse clustering (see 2.2.3 we aim to maximise the objective function of the algorith, i.e. maximize the between-cluster dispersion $\mathcal{J}'$ then the gap static for s will be given by the formula,

$$Gap(s) = log(\mathcal{J}_k') - E_B\{\log(\mathcal{J}_k'^*)\} \tag{2.33}$$

where,

$$E_B\{\log(\mathcal{J}_k'^*)\} = \frac{1}{B}\sum_{b}^{B}\log(\mathcal{J}_k'^*) \tag{2.34}$$

In this way, and similarly to the gap statistic in 2.2.8.1.3, we choose the s^* that corresponds to the largest value of $Gap(s)$ or, similarly to the stopping criterion in 2.32, s^* is choosen to equal the smallest s such that,

$$Gap(s) \geq Gap(s+1) - \mathcal{S}_{s+1} \tag{2.35}$$

where $\mathcal{S}_s = \sqrt{1+\frac{1}{B}} \cdot \sigma_s\big(E_B\{\log(\mathcal{J}_k'^*)\}\big)$ The authors of sparse clustering [Witten and Tibshirani, 2010] assumed that we k and we want to estimate only s but the latter study of [Brodinová et al., 2017] showed empiricly that k and s needs to be tunes together and for each pair of for different values of these two parameters executed the $Gap(s)$ method.

An advantage of the gap statistic compared to the silhouette is that it can be computed also for target number of clusters equal to 1 giving it the ability to test the null case of $k = 1$ against the alternative case of $k \geq 2$. However, it is computationally far more expensive than other methods since for any value of k to be tested the clustering method under analysis has to be executed $B + 1$ times where B is the number of reference data sets. In addition, it seems to be no direct indication of selecting a value for B. In the experiments of [Brodinová et al., 2017] and [Dudoit and Fridlyand, 2002] $B = 10$ was arbitrary selected and this value seems to be dominant for many third person implementations of the gap criterion; MATLAB [MATLAB, 2019] and R [R Core Team, 2017] have $B = 100$ by default.

2.2.8.2 External validity methods

2.2.8.2.1 Purity

Assuming that the number of clusters equals the number of class, the purity index estimates how much each cluster contains elements of only one class [Rendón et al., 2011]. For each cluster the purity index is defined as,

$$P_{c_k} = \frac{1}{n_k} \underset{\ell}{max}\{n_\ell^{(k)}\} \tag{2.36}$$

where $\underset{\ell}{max}\{n_{\ell_k}^{(k)}\}$ specifies the dominant class of the k-th cluster. The overall clustering purity index is then computed as,

$$P = \sum_{i=1}^{K} \frac{n_k}{n} P_{c_k} \tag{2.37}$$

The purity index is bounded between (0 1] [Aggarwal, 2014]; larger values of purity correspond to better performance accuracy and a purity of 1 specifies an accuracy of 100% meaning that each cluster has data points from only one class.

2.2.8.2.2 F-score

F-score (or F-measurement) is a method used in the field of information retrieval. Mathematically, it is a harmonic mean of precision and recall [Schütze et al., 2008] using pair of data points (for a full explanation on pair of data points computation refer to [Pfitzner et al., 2009]). In general, if a pair of data points is having the same

class then it should be assigned in the same cluster (true positive, TP) while if a pair of data points is having a different class it should not be assigned to the same cluster (true negative, TN). Alternatively, it is wrong to assign a pair of data points with the same class to different clusters (false negative, FN) and it is also wrong to assign a pair of data points with different class in the same cluster (false positive, FP) [Pfitzner et al., 2009]. Using the above logic then the clustering precision can be defined as,

$$precision = \frac{TP}{TP + FP} \tag{2.38}$$

and clustering recall can be specified as,

$$recall = \frac{TP}{TP + FN} \tag{2.39}$$

Finally, the F-score is given by the formula,

$$F\text{-}score = \frac{2 \times recall \times precision}{precision + recall} \tag{2.40}$$

The F-score is bounded between [0 1]. In some extreme case (refer to the github repository of [Usbeck et al., 2015]) the above formulas can cause a division by 0. In such cases:

- If TP, FP and FN are equal to 0 then precision, recall and F-score are equal to 1.
- If TP is equal to 0 and FP or FN are not equal to 0 then precision, recall and F-score are equal to 0.

2.2.8.3 Cross validation for semi-supervised clustering

Cross validation is a supervised process for avoiding overfitting, i.e. ensure that our system is generic and performs well on classifying unseen data. In this thesis cross validation will refer to the $\Bbbk$-fold cross validation where our data set is split into $\Bbbk$ folds. $\Bbbk - 1$ folds (the training set) are used to train the system and one fold (the testing set) is used to evaluate its performance. This process is executing until all the folds have been used as a test set, i.e. $\Bbbk$ times. The average performance of the system is then used as an index of performance.

In semi-supervised learning the same process can be used is order to evaluate the performance of a semi-supervised system, e.g. how the number and type of given constraints affect its performance. This is achieved using the partial labelled data of our data set, and using only these data to create the folds and proceed with the cross validation process as described before (refer to Figure 2.7). However, some previous studies [Bilenko et al., 2004; Gehring et al., 2015] have chosen a slightly alternative process, where the test data are also used in the training process but unlabelled (refer to Figure 2.8). This kind of route adds a bias in the process since testing data are used for the system training. Nevertheless, in the first study [Bilenko et al., 2004] the authors want to access supervised component of the MPCK-Means algorithm which is dependent only on labelled data. In the second study [Gehring et al., 2015], manual labelling of a small percentage of data points is part of the analysis process

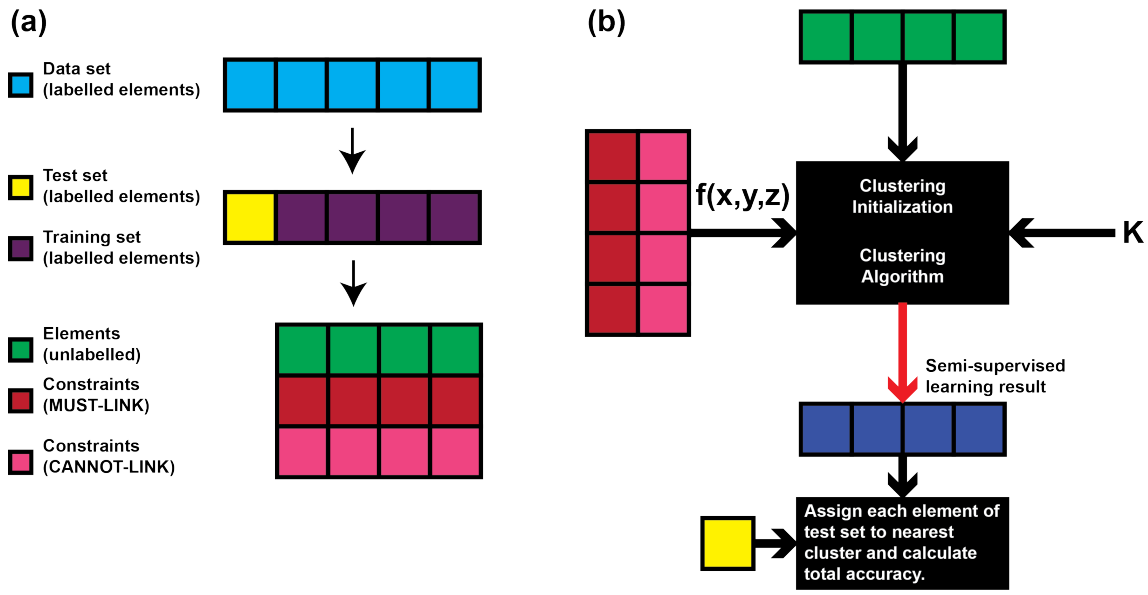


Figure 2.7: Semi-supervisor cross validation technique. Cross validation for semi-supervised learning is similar to the one used in classification. **(a)** The labelled data set is partitioned into test and training sets (e.g. in the 5-fold cross validation, as shown in the figure, we have 5 folds, 1 for testing and 4 for training). Using only the training set the labels are removed from the data and are used to generate MUST-LINK and CANNOT LINK constraints **(b)** The now unlabelled training set is fed into the semi-supervised clustering system (which is composed by a clustering initialisation method and a clustering algorithm one or both of which is semi-supervised, i.e. it uses constrains) among with a given number of clusters K which is equal to the number of labels and a certain number of constraints. Specifically for the constraints, a function is used to specify the type of constraints (z: MUST-LINK and/or CANNOT-LINK) as well as their number (x: number of MUST-LINK constraints, y: number of CANNOT-LINK constraints). After the result of the semi-supervised system is obtained, the part that corresponds in the test set is taken and the performance of the system is tested based only on it. The process is repeated as many times as the number of folds we have specified and each time the testing fold is swapped with a different training fold. The overall performance of the system is the average performance over all the folds. For the testing procedure, each element of the test set is assigned to its nearest cluster (which now represents a class) and the classification accuracy is obtained. The average accuracy over all the folds represent the cross validation metric of the system.

thus the aim is not to create a generic classifier but a classifier which is good only for the specific data set under consideration.

2.3 Data analysis in behavioural experiments

In the literature there are various ways to analyse data from behavioural experiments involving navigation but in general they all fall under two categories: performance measurements and behavioural quantification. The first category involves metrics that express various aspects of performance such as, the time that the subject spent inside an experimental procedure in order to solve a task, while the second category tries to match such metrics to stereotypical motifs of behaviour and then to link such motifs to different stages of learning and memory. The recent developments in the field of machine learning has brought much automation to the latter category and techniques for the automatic recognition of behavioural motifs. In this section a summary is presented with the most common techniques for analysing

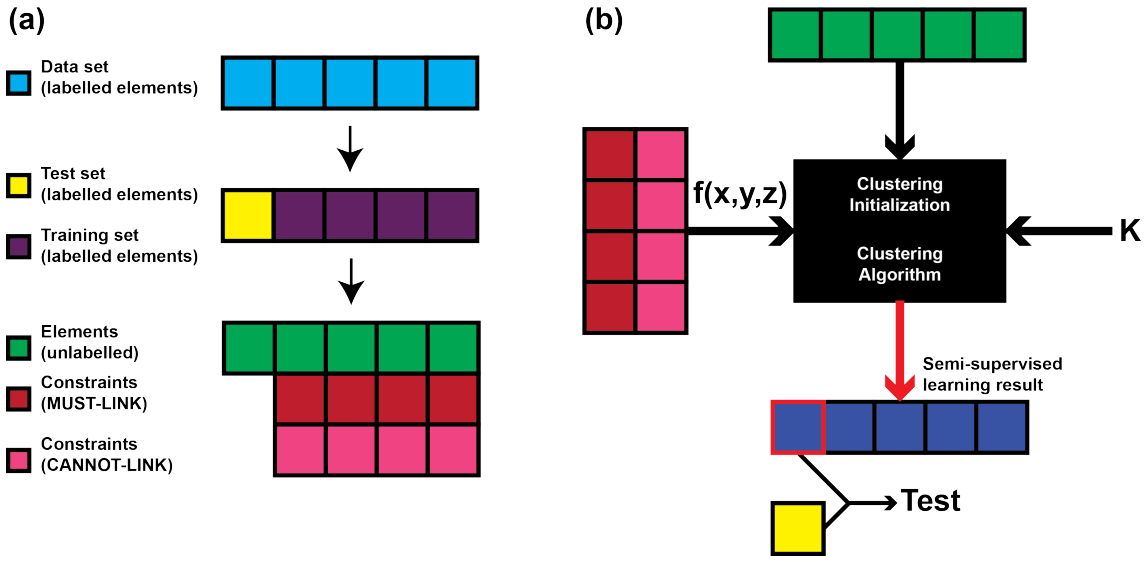


Figure 2.8: An alternative semi-supervisor cross validation technique. (a) The labelled data set is partitioned into test and training sets (e.g. in the 5-fold cross validation, as shown in the figure, we have 5 folds, 1 for testing and 4 for training). Using the training sets the labels are removed from the data and are used to generate MUST-LINK and CANNOT LINK constraints; from the testing set labels are also removed but no constraints are created. **(b)** The now unlabelled data set is fed into the semi-supervised clustering system (which is composed by a clustering initialisation method and a clustering algorithm one or both of which is using semi-supervised, i.e. it uses constrains) among with a given number of clusters K which is equal to the number of labels and a certain number of constraints. Specifically for the constraints, a function is used to specify the type of constraints (z: MUST-LINK and/or CANNOT-LINK) as well as their number (x: number of MUST-LINK constraints, y: number of CANNOT-LINK constraints). After the result of the semi-supervised system is obtained, the part that corresponds in the test set is taken and the performance of the system is tested based only on it. The process is repeated as many times as the number of folds we have specified and each time the testing fold is swapped with a different training fold. The overall performance of the system is the average performance over all the folds. For the testing procedure a number of options are available e.g. the performance can be based of the F-score [Bilenko et al., 2004], the classification accuracy [Gehring et al., 2015] or the purity index.

path data from the experimental procedures of the Morris Water Maze and light/dark preference task described before. The Morris Water Maze contains the most rich bibliography and best highlights the different data analytics approaches mentioned before thus this short review will start from there.

2.3.1 Data analytics in the Morris Water Maze

Most of the studies using the Morris Water Maze experiment utilise several measurements of performance in order to assess learning and memory. Many of these measurements have also been used to ensure that the animal groups have equal skills and abilities (e.g. swimming ability, speed, 'understanding' of the escape mechanism) [Maei et al., 2009; Vorhees and Williams, 2014]. Common measurements include the time that the animal spends inside each quadrant of the pool, the latency of finding the platform in each trial, the directionality and the total swimming distance in each trial [Brandeis et al., 1989; Lindner, 1997; Morris, 1984]. There are also a number of more sophisticated measurements such as the body temperature of the animals

throughout the experiment [Lindner and Gribkoff, 1991] or the cumulative distance to platform, which is the distance between the animal location and the platform location calculated a number of times with a specific sampling rate [Dalm et al., 2000; Gallagher et al., 1993]. The reader is also redirected to the study of [Tucker et al., 2018] which is focused on translational traumatic brain injury research but contains an informative overview and description of various performance measurements.

These simplistic measurements and statistics have been criticised as being insufficient to capture all of the different animal behaviours that are observed during the Morris Water Maze experiments [Dalm et al., 2000; Gehring et al., 2015]. For this reason researchers started to study the various behaviours that the animals were expressing inside the pool, which are known as exploration strategies. Notable are the studies of Wolfer et al., who computed a large amount of measures for each swimming path inside the maze in order to categorise the various strategies [Wolfer and Lipp, 2000; Wolfer et al., 1998] and also developed two softwares, TRACK-ANALYZER [Wolfer and Lipp, 1992] and Wintrack [Wolfer et al., 2001] to make their methods publicly available to the scientific community. Other studies include the automatic classification procedures of Graziano et al. [Graziano et al., 2003] and Garthe et al. [Garthe et al., 2009], both of which specified regions of interest inside the arena. The categorisation method of Graziano et al. was based on a number of path measures while in the work of Garthe et al. a hierarchical classification algorithm was used and the categorisation of each swimming path was primarily based on the amount of time that the animal spent in each region of the arena. The latter method was also used in more recent studies ([Rogers et al., 2017; Yeshurun et al., 2017]). Illouz et al. [Illouz, Madar, Louzon, Griffioen and Okun, 2016] proposed a classification technique based on support vector machines (SVM) [Cortes and Vapnik, 1995]. Based on their method, the SVM was trained on 800 labelled trials where the animal [X, Y] coordinates of the paths had been converted to a set of 11 features. The final classicifation was the result of a series of binary choices that were classifying the trials from generic classes (e.g. long and short trials) to more specific ones (e.g. Thigmotaxis and Chaining). Their method was generic enough to detect these behavioural strategies on a variety of different Morris Water Maze setups and the same analysis procedure was also used latter in another behavioural task the Barnes maze [Illouz, Madar, Clague, Griffioen, Louzoun and Okun, 2016].

2.3.2 Data analytics in light/dark preference task

Common measurements in the light/dark preference task regardless of the subject type (rodent or zebrafish) include the total or percentage of time that the animal remained in the light or dark compartment of the arena [Aulich, 1976; Hascoët et al., 2001; Magno et al., 2015; Rodgers and Shepherd, 1993], the latency to switch between the light and the dark compartments for the first time and the number of entrances/exits to and from the light and dark compartments (number of transitions) [Blumstein and Crawley, 1983; Crawley and Goodwin, 1980; Magno et al., 2015]. Other measurements include total path length of the animals, turning angles and immobility and thigmotactic (amount of time spent near walls) duration [Araujo et al., 2012; Blaser and Penalosa, 2011].

Compared with the Morris Water Maze data analytics methods for the light/dark preference task are limited and there is no direct classification of animal behaviours.

Thigmotaxis, which is a behaviour where the animal moves around the walls of the arena, and exploration as well as light or dark avoidance are measured based on the aforementioned measurements.

Chapter 3

New clustering techniques and benchmarking

This chapter consists of benchmarks aiming at detecting optimal K-Means clustering initialisation methods and comparing a novel semi-supervised algorithm with other existing ones in terms of classification accuracy and feature selection capabilities. The goal is to identify and engineer methods that can be used to analyse data from behavioural neuroscience experiments consisted of biological features with unknown informative power. In more detail:

- Stochastic and deterministic initialisation methods are compared under different unsupervised variations of K-Means clustering in order to access their goodness on clustering and their robustness towards the different algorithms (for relevant material refer to [Vouros et al., 2019]).
- A modification of Sparse K-Means clustering is proposed in order to include pairwise constraints. The new algorithm, called PCSK-Means, is compared against other unsupervised and semi-supervised K-Means variations in the following aspects:
  - classification accuracy,
  - feature selection,
  - robustness of the above under different conditions such as different initialisation methods.

  (for relevant material refer to [Vouros and Vasilaki, 2020]).

3.1 Comparison between stochastic and deterministic centroid initialisation for unsupervised K-Means variations

This section will cover the published material of [Vouros et al., 2019] which aims to benchmark existing initialisation methods for K-Means clustering algorithms as well as common K-Means variations. Compared with the published material the benchmark of this section contains an additional K-Means variation which is the Weiszfeld algorithm.

3.1.1 Introduction

The most well-known algorithm in the field of clustering analysis is the K-Means algorithm. Its simplicity, versatility and efficiency makes it popular in many different research fields [Jain, 2010; Pena et al., 1999]. Despite its reputation and success in many different studies, it has a series of disadvantages such that it can detect only spherical and well-separated clusters, it is sensitive to outliers, highly dependent on the features (dimensions) of the data set and it only converges to local minima [Jain, 2010]. Over the years a number of K-Means variations (Lloyd's K-Means [Slonim et al., 2013], Hartigan-Wong's K-Means [Hartigan, 1975]), K-Means inspired algorithms (K-Medians [Aggarwal, 2014]), and K-Means initialisation methods [Celebi et al., 2013] have been proposed in order to overcome some of these issues. Such methods have also enhanced KMeans with additional properties such as feature selection mechanisms [Kondo et al., 2016; Witten and Tibshirani, 2010] and outliers robustification [Al Hasan et al., 2009; Brodinová et al., 2017].

In the literature there are various studies regarding the importance of the initial selection of cluster centroids for the performance of the K-Means algorithm [Jain, 2010] and extensive testing on various initialisation techniques [Celebi et al., 2013; Fränti and Sieranoja, 2019], but a detailed comparison on the effects on these techniques on common K-Means variations is not available. It is hypothesized that sophisticated initialisation methods alleviate the need for complex clustering and, if deterministic, they could lead to satisfactory solutions within a single execution of the clustering algorithm. Consequently, they would alleviate the need for executing a stochastic method multiple times and picking the best clustering based on some criterion.

In order to investigate this hypothesis a comparison of different clustering initialisation methods, namely Random [MacQueen et al., 1967], K-Means++ [Arthur and Vassilvitskii, 2007], Maximin [Gonzalez, 1985] ROBust INitialisation (ROBIN) [Al Hasan et al., 2009], Kaufman [Kaufman and Rousseeuw, 2009] and Density K-Means++ (DK-Means++) [Nidheesh et al., 2017], and their effects on common K-Means variations, Lloyd's K-Means [Jain, 2010], Hartigan-Wong's K-Means [Hartigan, 1975; Hartigan and Wong, 1979] and K-Medians [Aggarwal, 2014] is performed. It is shown that more sophisticated initialisation methods reduce on average the performance difference among the K-Means implementations and that the deterministic DK-Means++ method can achieve better average performance than stochastic methods. Nevertheless there is a trade-off, simplistic stochastic methods can achieve better clustering performance if executed multiple times due to the potential of discovering better local minima. For very large data sets where execution time is a factor, a single run using a deterministic initialisation method can be competitive compared to multiple runs using stochastic initialisation methods.

A similar study comparing many different intialisation methods has been performed by [Celebi et al., 2013] but it is focused on algorithms of linear complexity without considering various K-Means implementations. Recently, another study [Fränti and Sieranoja, 2019] was performed on stochastic initialization heuristics for K-Means and on how much the algorithm can be improved by repetition. They based their conclusions on a clustering benchmark [Fränti and Sieranoja, 2018, 2019] which contains standalone data sets with different properties and they showed that K-Means performance is in general poor on unbalanced data sets and that the algorithm is not affected by high dimensionality while more iterations can improve its performance

on overlapping clusters. A more extensive benchmarking is performed in the current study that takes into consideration data set generation models as well as standalone data sets. The models gave us the ability to perform hypothesis testing in order to strengthen our conclusions and to account for variability.

The code of the clustering methods, data set model generators, scripts and a standalone application to reproduce this research are available in the GitHub repository https://github.com/avouros/Code-KMeans-benchmark (under the branch PhD-additions).

3.1.2 Methods

This work will make use of the algorithms described in chapter 2. The clustering algorithms are as follows: Lloyd's K-Means (see 2.2.2.2), Hartigan-Wong's K-Means (see 2.2.2.3), K-Medians (see 2.2.5) and Weiszfeld (see 2.2.6.1). The clustering initialisation methods are as follows: Random (MacQueen, see 2.2.7.1), K-Means++ (see 2.2.7.2), Maximin (see 2.2.7.3), Kaufman (see 2.2.7.4), ROBust INitialisation (ROBIN(S) and ROBIN(D), see 2.2.7.5) and Density K-Means++ (DK-Means++) (see 2.2.7.6). For the clustering evaluation the purity and silhouette indexes are used (see 2.2.8.2.1 and 2.2.8.1.2).

3.1.2.1 Benchmark

In the experiments, the synthetic data set models from the studies of Tibshirani et al. (gap statistic) [Tibshirani et al., 2001], Yan and Ye (weighted gap statistic) [Yan and Ye, 2007] and Brodinova et al. [Brodinová et al., 2017] are used. A summary of the models can be found in Table 3.1 grouped by specific properties of the models. For more information refer to the relevant studies and also to Figure 3.1 for a sample visualization of each model. Model 1 from the gap statistic study [Tibshirani et al., 2001] is excluded since it contains only one cluster. The Brodinova et al. [Brodinová et al., 2017] generator was used to generate high-dimensional data sets consisting of informative and non-informative features. No noise injection (attributes with noise contamination) was considered in the current study. To avoid situations of overlapping clusters the minimum Euclidean distance between any two points in different clusters was set to 3 and data sets violating this rule were re-generated. A summary of the models can be found in Table 3.2. Next, novel synthetic data sets models consisted of clusters with mixed properties are considered. These are referred to as *mixed* models (refer to Figure 3.1 for a sample visualization of these models):

- *model 1* generates 3-dimensional clusters, 1 spherical and 2 elongated. The spherical cluster is an 80 points Gaussian cluster at the origin with standard deviation of 0.1. The two elongated clusters have 100 points each and are generated as follows: $x_1 = x_2 = x_3 = t$ with t taking 100 equally spaced values from -1 to 1. Gaussian noise with standard deviation of 0.3 was then added to each dimension. The second dimension of the first elongated cluster was shifted by 2 from the centre of the spherical cluster. Similarly the second dimension of the second elongated cluster was shifted by -2 and the first dimension was rotated by 180^0.

- *model 2* generates 3-dimensional non-Gaussian and normal clusters. It generates: (a) a cluster from an exponential distribution with rate of 1 and truncated

at $[-1\ \ 1]$ containing 80 points, (b) a cluster from an exponential distribution with rate of 1 and and truncated at $[2\ \ 3]$ with 100 points, (c) a Gaussian cluster of 80 points with mean $[0.5, 2.5, 2.5]$ and standard deviation of 0.1 in every dimension and (d) a Gaussian cluster of 100 points with mean $[2.5, 0.5, 0.5]$ and standard deviation of 0.2 in every dimension.

- *model 3* generates 3-dimensional Gaussian clusters with different standard deviations. The first cluster has 80 points with mean at the origin and standard deviation of 0.1 on each dimension. The second cluster has of 100 points with mean $[2, 0, 0]$ and standard deviation of 0.2 on each dimension. The third cluster has of 120 points with mean $[0, 2, 0]$ and standard deviation of 0.3 on each dimension. The forth cluster consists of 140 points with mean $[0, 0, 2]$ and standard deviation of 0.4 on each dimension.

- *model 4* generates 3-dimensional mixed Gaussian clusters. The first cluster consists of 80 points with mean $[0, 0, 0]$ and standard deviations $[0.1, 0.1, 0.2]$. The second cluster consists of 100 points with mean $[2, 0, 0]$ and standard deviations $[0.1, 0.2, 0.3]$. The third cluster consists of 120 points with mean $[0, 2, 0]$ and standard deviations $[0.2, 0.4, 0.6]$. The forth cluster consists of 140 points with mean $[0, 0, 2]$ and standard deviations $[1.0, 0.1, 0.1]$.

Other data sets are the *S-sets* [Fränti and Virmajoki, 2006] and the *A-sets* [Kärkkäinen and Fränti, 2002] obtained from the "clustering basic benchmark" which were used in the studies of [Fränti and Sieranoja, 2018, 2019]. The aforementioned studies were dedicated to the K-Means properties, advantages and disadvantages. Both models contain 2-dimensional data; *S-sets* contains 4 data sets with 5000 data points distributed among 15 Gaussian clusters with different degree of clustering overlap [Fränti and Virmajoki, 2006] and *A-sets* contains 3 data sets with 20, 35 and 50 clusters and 150 data points per cluster [Kärkkäinen and Fränti, 2002]. For more information about these data sets refer to the relevant studies. Finally, a selection of real-world data sets from the UCI repository [Asuncion and Newman, 2007] is considered. These data sets are the following: Iris, Ionosphere, Wine, Breast Cancer, Glass and Yeast. More information about these data sets are shown on Table 3.3.

3.1.3 Results

This study tests the performance of the K-Means variations, Lloyd's [Jain, 2010] Hartigan-Wong's [Hartigan, 1975; Hartigan and Wong, 1979], K-Medians [Aggarwal, 2014] and Weiszfeld [Whelan et al., 2015] initialised using the eight different clustering initialisation methods named: Random [MacQueen et al., 1967], K-Means++ [Arthur and Vassilvitskii, 2007], Maximin(S) [Gonzalez, 1985], ROBIN(S) [Brodinová et al., 2017], Kaufman [Kaufman and Rousseeuw, 2009], ROBIN(D) [Al Hasan et al., 2009], DK-Means++ [Nidheesh et al., 2017] and Maximin(D) [Katsavounidis et al., 1994]. For the ROBIN variations the *mp* parameter specifying the number of neighbor data points was set to 10 as in the original study [Al Hasan et al., 2009]. For the Hartigan-Wong's algorithm NAG's implementation was used [Numerical Algorithms Group (NAG), 2019].

A "sophistication" scale is considered for the initialisation methods based not only on their execution time but also on the complexity of their underlying operators.

Table 3.1: Gap [Tibshirani et al., 2001] and weighted gap statistic [Yan and Ye, 2007] data sets models. Points: the number of data points per cluster, *or* indicates that a random number was selected among the specified numbers for each cluster, *to* indicates that a random number was selected between the specified numbers for each cluster; D: number of features or attributes of the data set (dimensions); C: number of generated clusters. Gaussian models: clusters of low dimensionality generated from Gaussian distributions. 10-D Gaussian models: clusters of higher dimensionality generated from Gaussian distributions. Elongated models: clusters generated by adding Gaussian noise across lines. Unbalanced model: data sets containing Gaussian clusters of very different sizes, exponential: non-Gaussain clusters generated from the exponential distribution. Visualization (when possible) of a data set from each model is available in Figure 3.1.

Gaussian models	**Points**	**D**	**C**
gap model (gap 2)	25,25,50	2	3
gap model 3 (gap 3)	25 or 50	3	4
weighted gap model 1 (wgap 1)	25 to 50	2	6
weighted gap model 6 (wgap 6)	50 each	2	6

Unbalanced model	**Points**	**D**	**C**
weighted gap model 2 (wgap 2)	100,15	2	2

Exponential model	**Points**	**D**	**C**
weighted gap model 3 (wgap 3)	50 each	2	4

10-D Gaussian models	**Points**	**D**	**C**
gap model 4 (gap 4)	25 or 50	10	2
weighted gap model 5 (wgap 5)	25 to 50	10	2

Elongated models	**Points**	**D**	**C**
gap model 5 (gap 5)	100 each	3	2
weighted gap model 4 (wgap 4)	100 each	2	2

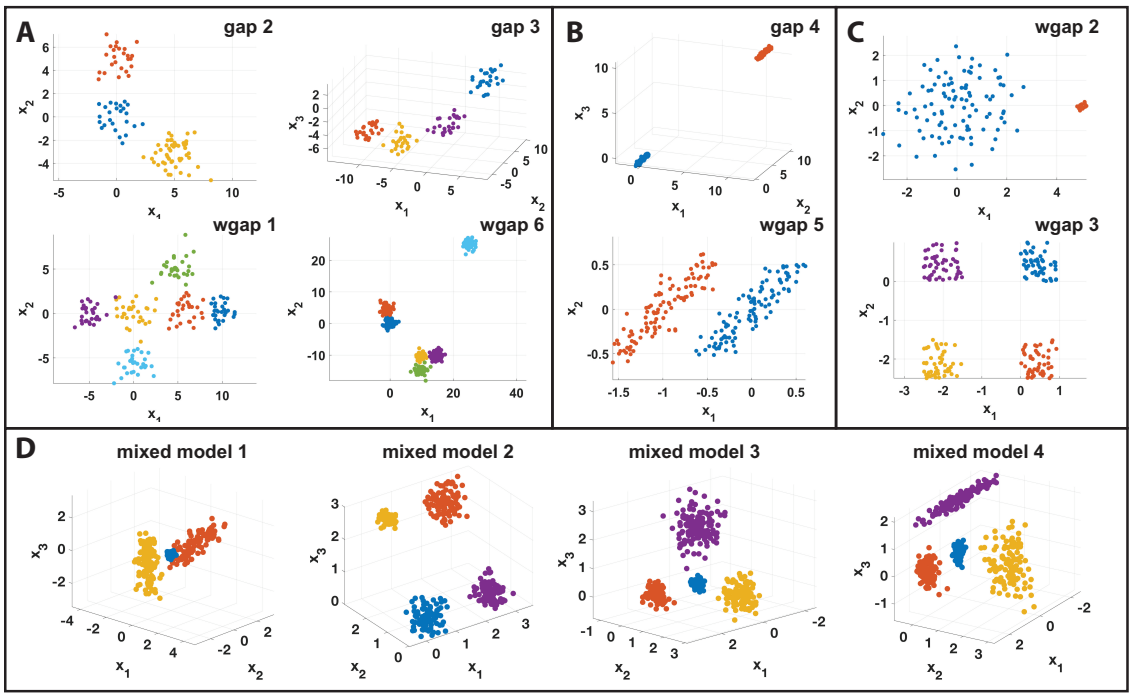


Figure 3.1: Data set models visualization. Examples of data used in this study. Gap (gap) and weighted gap (wgap) models are separated into three categories: **(A)** 4 Gaussian models, **(B)** 2 elongated models and **(C)** one highly unbalanced model (wgap2) and one non-Gaussian model (wgap3) in which the clusters are generated from the exponential distribution. **(D)** Mixed models: these are models proposed in this study that contain clusters with mixed properties such as different sizes (unbalanced) and/or generated from Gaussian and non-Gaussian distributions.

Table 3.2: Brodinova model generator [Brodinová et al., 2017]**.** The minimum allowed Euclidean distance between two data points in different clusters was set to 3 and no noise injection was considered. Name: name of the model; Points: the total number of data points in the data set; D: number of features or attributes of the data set, *Informative (+)* indicates attributes that are required to describe the data set while *Non-informative (-)* indicates variables that should be ignored; C: number of generated clusters. These models are creating high-dimensional Gaussian clusters of different shapes using two different distributions, one for the informative and one for the uninformative variables. *Left table:* Each cluster contains 40 data points. The parameters of these models are selected to test the performance of the clustering algorithm in data sets with different degrees of informative and/or uninformative features *Right table:* The first four models create higher-dimensional balanced clusters (clusters of equal sizes) and the last two higher-dimensional unbalanced clusters each with number of points randomly selected between 50 to 100. The parameters of these models are selected to test the performance of the clustering algorithm in balanced and unbalanced data sets of higher dimensionality with increasing number of clusters. The input space was selected to be sparse, i.e. a few hundred points in 1000 or 1500 dimensions to avoid the slow computation of the Kaufman algorithm.

Name	Points	D		C
		+	-	
brod 1	120	20	0	3
brod 2	400	20	0	10
brod 3	120	15	5	3
brod 4	400	15	5	10
brod 5	120	10	10	3
brod 6	400	10	10	10

Name	Points	D		C
		+	-	
brod 7	120	1000	0	3
brod 8	400	1000	0	10
brod 9	400	1500	0	10
brod 10	1250	1500	0	50
brod 11	50 to 100	1000	0	3
brod 12	50 to 100	1000	0	10

Table 3.3: Real data sets from the UCI repository [Asuncion and Newman, 2007]. Points: the number of data points per cluster; D: number of features or attributes of the data set (dimensions); C: number of generated clusters.

Name	Points	D	C
Iris	50,50,50	4	3
Ionosphere	225,126	34	2
Wine	59,71,48	13	3
Breast Cancer	444,239	9	2
Glass	70,76,17, 19,9,29	9	6
Yeast	463,5,35,44, 51,163,244, 429,20,30	8	10

For example DK-Means++ and ROBIN would be considered more sophisticated than Kaufman since they incorporate more advanced statistics while Kaufman uses only distances and still has a complexity of $O(N^2)$. Our scale is as follows: Random $<$ K-Means++ $<$ Maximin $<$ Kaufman $<$ ROBIN $<$ DK-Means++.

In the experiments the synthetic data sets models from the studies of gap statistic [Tibshirani et al., 2001] and weighted gap statistic [Yan and Ye, 2007] (refer to Table 3.1, 10 sets in total), Brodinova [Brodinová et al., 2017] (refer to Table 3.2, 12 sets in total) and other four custom data sets models (refer to Methods and Figure 3.1, 4 sets in total) are used. From each model 40 data sets are generated and for each data set the stochastic methods are executed 50 times. The "clustering data sets" (*S-sets* [Fränti and Virmajoki, 2006] and *A-sets* [Kärkkäinen and Fränti, 2002]) from the studies of [Fränti and Sieranoja, 2018, 2019] are also used as well as real-world data sets from the UCI repository [Asuncion and Newman, 2007]: Iris, Ionosphere, Wine, Breast Cancer, Glass and Yeast (see Table 3.3). For each of these data sets the same setup of executing the stochastic methods 50 times is considered.

All the hypothesis testing on the data set models is based on the Paired Samples Wilcoxon Test, a non-parametric alternative to paired t-test, For the outcome of the test the following symbols indicate the corresponding level of significance, $*$ for p-value < 0.05; $**$ for p-value < 0.01; $***$ for p-value < 0.001; $****$ for p-value < 0.0001. For the clustering performance measurements there was evaluation on the monotonic relationship of Silhouette and Purity via a large sample of clustering results on the multiple executions of the methods across all the data sets (20000 cases). Spearman's rank correlation coefficient indicated that Purity and Silhouette have a strong monotonic relation (Spearman's Rho 0.97). As a side note, the distortion score, which is essentially the outcome of the objective function of the algorithm (for K-Means clustering this is the WCSS) and has been used before in other studies [Al Hasan et al., 2009; Celebi et al., 2013], had a weaker monotonic relationship with Purity (Spearman's Rho 0.65).

3.1.3.1 Comparison on the average performance among stochastic and among deterministic methods

First this study assesses the average performance of stochastic methods as well as the performance of deterministic methods. For the former the average performance of stochastic methods is assessed based on 50 different runs across 40 different data sets for each one of our 26 models (10 gap and weighted gap, 12 Brodinova, 4 mixed models). Deterministic methods are executed once on the 40 data sets.

Based on Figure 3.2 the average performance of K-Means variations increases by using more sophisticated initialisation methods and ROBIN(S) initialisation provides the best average performance followed by Maximin(S) and K-Means++ while Random initialisation results in the poorest performance. For the deterministic methods, shown on Figure 3.3, it is observed again that the average performance of K-Means variations increases by using more sophisticated initialisation methods. DK-Means++ achieved the best average performance followed by ROBIN(D) and then by Kaufman and Maximin(D). Finally, it is assessed if more sophisticated initialization methods alleviate the need for complex clustering. For this reason comparisons among the K-Means variations initialised with either Random and K-Means++ or Kaufman and DK-Means++ methods are performed. Maximin and ROBIN have both stochastic and deterministic variations of equal sophistication thus they were excluded them. As shown in Table 4, deterministic methods 3.4 deterministic methods that are more sophisticated (as per our definition) reduce performance differences among the different variants of the K-Means algorithms.

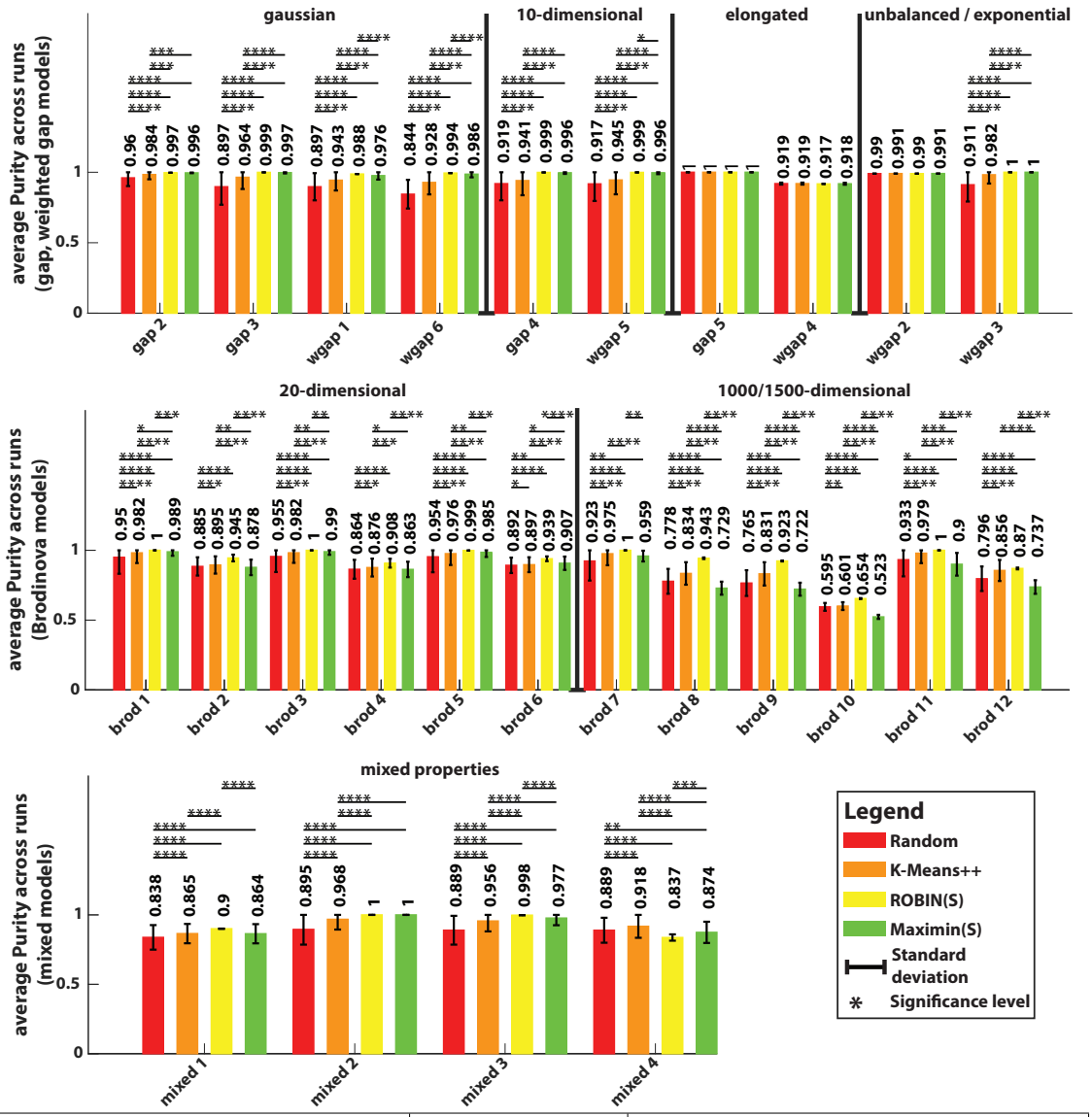
3.1.3.2 Comparison of the average performance between stochastic and deterministic methods

Following on from the previous conclusions, it is assessed if overall deterministic methods provide on average better performance than stochastic methods. For this reason the following comparisons are performed: stochastic and deterministic variations of Maximin and ROBIN as well as the best stochastic performer (ROBIN(S) see Figure 3.2) and the best deterministic performer (DK-Means++ see Figure 3.3). Based on the results in Figure 3.4: (a) Maximin(D) is on average better than Maximin(S); (b) ROBIN(D) and ROBIN(S) are on average equivalent; (c) DK-Means++ is better than ROBIN(S).

3.1.3.3 Comparison of the maximum performance across multiple runs of stochastic and deterministic methods

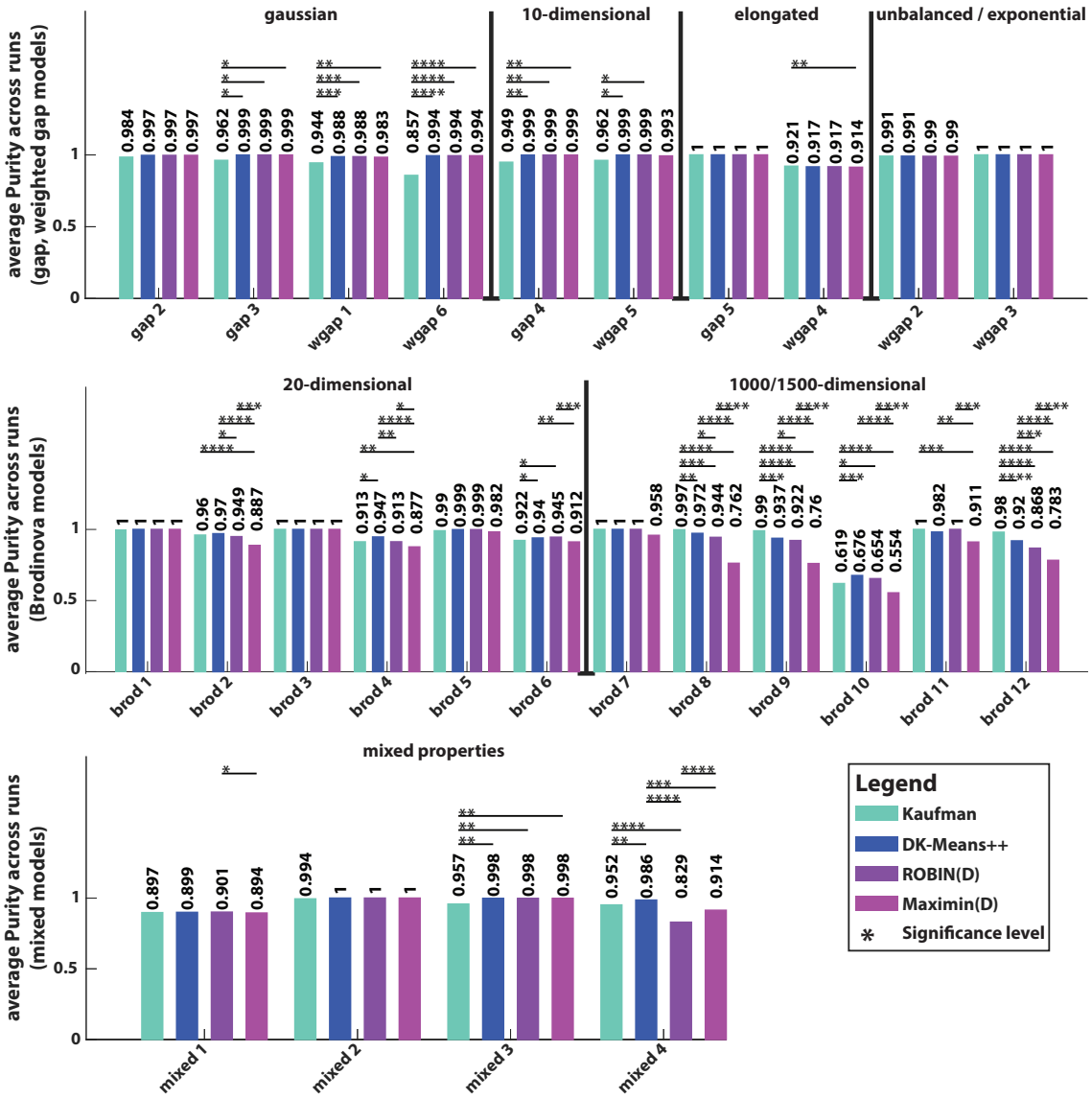
Next, there is an additional comparison between the stochastic and the deterministic methods but based on the maximum performance that the former can achieve on multiple repetitions. Each stochastic method was run 50 times and the best outcome was selected based on the silhouette index. Its corresponding value is reported according to the purity index. It is expected that due to the many repetitions, stochastic methods can find different local minima and potentially result in a better performance at the cost of multiple repetitions.

Firstly, comparison is repeated among the different stochastic methods but based on the maximum performance that they can achieve. Figure 3.5 shows the relevant results and, opposite to our observations on the average performance, stochastic



Initialization method	Total number of instances	Significantly better average performance			
		HW	Ll	KMed	Weis
Random vs K-Means++	26	0 vs 23	0 vs 23	0 vs 24	0 vs 23
Random vs ROBIN(S)	26	1 vs 22	3 vs 22	2 vs 23	2 vs 22
Random vs Maximin(S)	26	6 vs 15	6 vs 16	6 vs 16	0 vs 19
K-Means++ vs ROBIN(S)	26	1 vs 21	3 vs 21	2 vs 22	2 vs 22
K-Means++ vs Maximin(S)	26	8 vs 13	9 vs 14	7 vs 13	6 vs 15
ROBIN(S) vs Maximin(S)	26	17 vs 1	18 vs 3	19 vs 1	17 vs 2

Figure 3.2: The average performance of K-Means variations increases by using more sophisticated stochastic initialisation methods. Each plot shows the performance of the Hartigan-Wong's K-Means clustering solution using the Silhouette index (y-axis) on different data sets models (x-axis) and initialized with different stochastic methods. To calculate performance, the average Purity index across the 50 initial conditions and 40 data sets for each model (gap, weighted gap, Brodinova and mixed) is considered. The errorbars are showing the (average) standard deviation across the 40 data sets. Solid lines on any two bars underline the level of significant difference between the corresponding methods (cases of no significant differences are not showing). The accompanied Table below the figure shows a summary of the comparisons through all the K-Means variations (Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld (Weis)) where there is a significant performance difference between the compared methods. Based on the results ROBIN(S) achieves the best performance followed by Maximin(S), K-Means++ and Random.



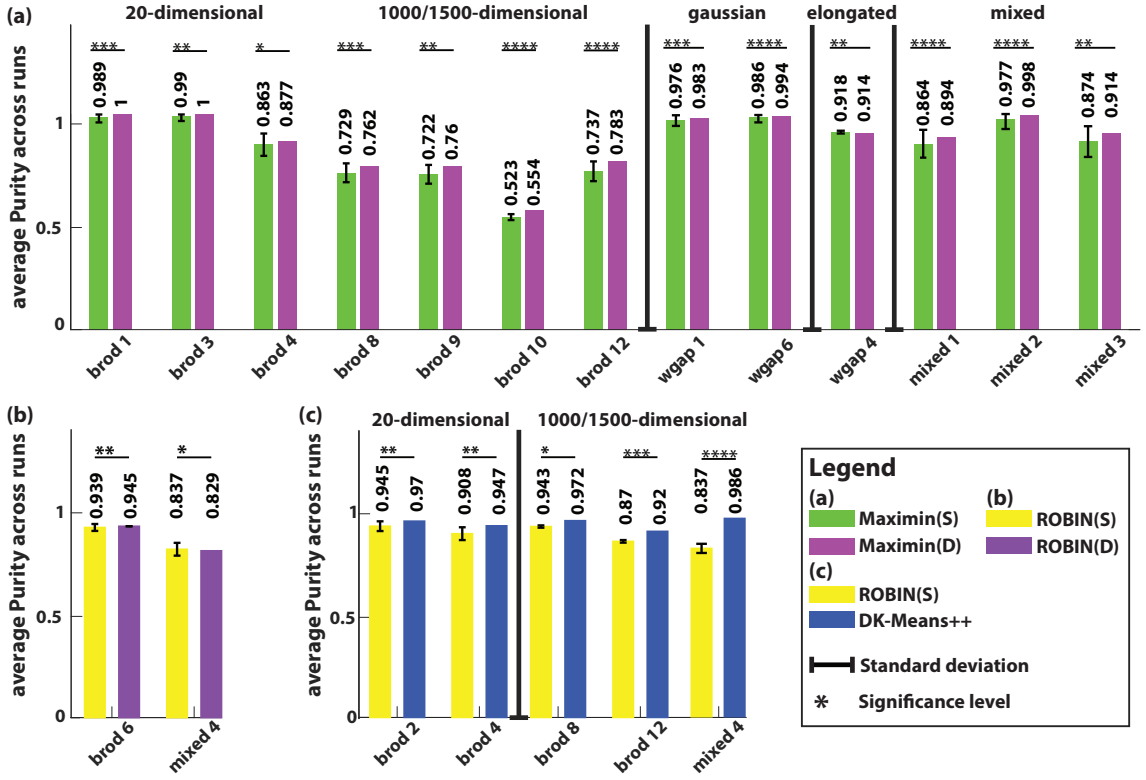
Initialization method	Total number of instances	Significantly better average performance			
		HW	Ll	KMed	Weis
Kaufman vs DK-Means++	26	3 vs 10	3 vs 9	4 vs 8	4 vs 8
Kaufman vs ROBIN(D)	26	4 vs 8	6 vs 9	4 vs 6	5 vs 8
Kaufman vs Maximin(D)	26	8 vs 5	11 vs 5	9 vs 5	9 vs 6
DK-Means++ vs ROBIN(D)	26	6 vs 0	7 vs 0	6 vs 1	3 vs 0
DK-Means++ vs Maximin(D)	26	9 vs 0	11 vs 0	10 vs 1	7 vs 0
ROBIN(D) vs Maximin(D)	26	9 vs 1	10 vs 1	9 vs 1	8 vs 1

Figure 3.3: The average performance of K-Means algorithm increases by using the more sophisticated deterministic initialisation methods. Each plot shows the performance of the Hartigan-Wong's K-Means clustering solution using the Silhouette index (y-axis) on different data sets models (x-axis) and initialized with different stochastic methods. To calculate performance, the average Purity index across the 40 data sets for each model (gap, weighted gap, Brodinova and mixed) is considered. Solid lines on any two bars underline the level of significant difference between the corresponding methods (cases of no significant differences are not showing). The accompanied Table below the figure shows a summary of the comparisons through all the K-Means variations (Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld (Weis)) where there is a significant performance difference between the compared methods. Based on the results DK-Means++ achieves the best performance followed by ROBIN(D), Kaufman and Maximin(D).

Table 3.4: More sophisticated initialisation methods alleviates the need for complex clustering. Each row compares two K-Means variations (Hartigan-Wong's K-Means: HW, Lloyd's K-Means: Ll, K-Medians: KMed and Weiszfeld: Weis) initialised with the same method on 26 occasions (10 gap and weighted gap, 12 Brodinova and 4 mixed models). To calculate performance, the average Purity index across the 50 initial conditions and 40 data sets for each model is considered and the comparison is based on the times that there was significant difference between the two algorithms. The ROBIN and the Maximin variations were excluded from this analysis since their stochastic and deterministic versions have equivalent sophistication. Based on the results using a deterministic method reduces the observed average performance differences among clustering algorithms.

Initialization method	**Total number of instances**	**Significantly better average performance**		
		HW vs Ll	**HW vs KMed**	**Ll vs KMed**
Random	26	13 vs 0	18 vs 4	6 vs 4
K-Means++	26	10 vs 0	13 vs 3	2 vs 5
Total	**52**	**23 vs 0**	**39 vs 12**	**8 vs 9**
Kaufman	26	1 vs 1	2 vs 6	1 vs 6
DK-Means++	26	1 vs 0	2 vs 4	1 vs 5
Total	**52**	**2 vs 1**	**4 vs 10**	**2 vs 11**

Initialization method	**Total number of instances**	**Significantly better average performance**		
		HW vs Weis	**Ll vs Weis**	**KMed vs Weis**
Random	26	16 vs 1	8 vs 4	14 vs 4
K-Means++	26	14 vs 2	6 vs 2	4 vs 2
Total	**52**	**30 vs 3**	**14 vs 6**	**18 vs 6**
Kaufman	26	1 vs 5	2 vs 4	2 vs 1
DK-Means++	26	2 vs 5	1 vs 4	1 vs 1
Total	**52**	**3 vs 10**	**3 vs 8**	**3 vs 2**



Initialization method	Total number of instances	Significantly better average performance			
		HW	Ll	KMed	Weis
Maximin(S) vs Maximin(D)	26	1 vs 12	3 vs 12	0 vs 13	1 vs 12
ROBIN(S) vs ROBIN(D)	26	1 vs 1	1 vs 2	1 vs 0	1 vs 0
ROBIN(S) vs DK-Means++	26	0 vs 5	0 vs 7	1 vs 5	0 vs 3

Figure 3.4: Deterministic methods for K-Means clustering provide, on average, equally good or better performance than stochastic methods. Each plot shows the performance of the Hartigan-Wong's K-Means clustering solution using the Silhouette index (y-axis) on different data sets models (x-axis) and initialized with different stochastic methods. To calculate performance, the average Purity index across the 50 initial conditions and 40 data sets for each model (gap, weighted gap and Brodinova) is considered. The errorbars (on the stochastic methods only) are showing the average standard deviation across the 40 data sets. Solid lines on any two bars underline the level of significant difference between the corresponding methods (cases of no significant differences are not showing). The accompanied Table below the figure shows a summary of the comparisons through all the K-Means variations (Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld (Weis)) where there is a significant performance difference between the compared methods. Based on the results **(a)** Maximin(D) is on average better than Maximin(S); **(b)** ROBIN(D) and ROBIN(S) are on average equivalent; **(c)** the best deterministic method DK-Means++ (see Figure 3.3) is better than the best stochastic method ROBIN(S) (see Figure 3.2).

methods have higher chances of obtaining a better clustering result with multiple execution. K-Means++ is the best method followed by Random while ROBIN(S) and Maximin(S) have almost similar performance.

Afterwards, the maximum performance of stochastic methods with the performance of deterministic methods is compared similarly to our previous experiment (refer to Figure 3.4). The comparison includes the stochastic and deterministic variations of Maximin and ROBIN as well as the best stochastic performer of the current experiment (K-Means++) and the best deterministic performer (DK-Means++ see Figure 3.3). Based on the results in Figure 3.6, stochastic variations of Maximin and ROBIN achieve overall better performance than their deterministic counterparts and K-Means++ is better than DK-Means++.

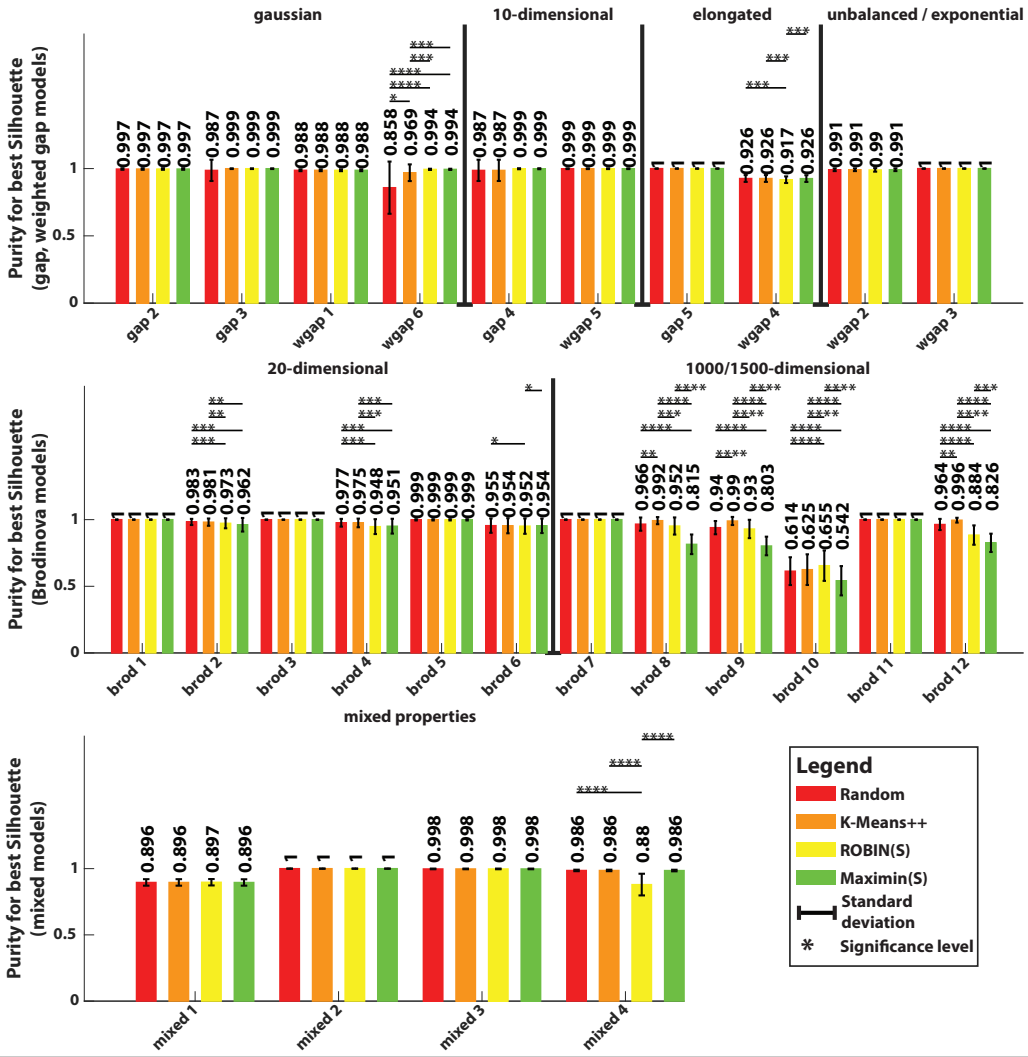
K-Means variations are also compared using different intialisation methods. Based on the result on Table 3.5 K-Medians achieves the best performance followed by Hartigan-Wong's; Lloyd's was the worst performer. Nevertheless, these systematic differences correspond to only 1.5% purity difference.

3.1.3.4 Standalone synthetic and real-world data sets

Standalone data sets were regarded as cases where supervised information is unknown and the assessment of the performance of the algorithms on them was based on the silhouette index. Detailed results for each data set (minimum, maximum, average performance and variance for each K-Means variation) are illustrated in Appendix A.8. It should be noted that DK-Means++ was always able to achieve the best performance of the unsupervised methods while ROBIN(D) failed to achieve the best performance in the cases of A-sets 1, S-Sets 3 and S-Sets 4 when Lloyds and K-Medians were considered; Kaufman and Maximin(D) where the worst performers. From the stochastic methods ROBIN(S) always managed to achieve the maximum performance apart from one case of S-Sets 3 when the Harigan-Wong K-Means was considered; Random was the worst performer. For the real-world data sets most algorithms behaved the same but Maximin(S) outperformed everyone else in the cases of Yeast (all K-Means variations) and Ionosphere (Lloyd's K-Means only). In the case of Glass (all K-Means variations) K-Means++ and Maximin(S) had the best performances. However, with the real data sets it is rare for the number of clusters to equal the number of classes [Gehring et al., 2015]. Thus these data sets might not be the best examples for clustering benchmarking. Also the relatively better performance of Maximin(S) only appears in these few cases where the Silhouette index indicates poor clustering results in general. In such cases comparative conclusions may not be meaningful as these specific results could a product of chance.

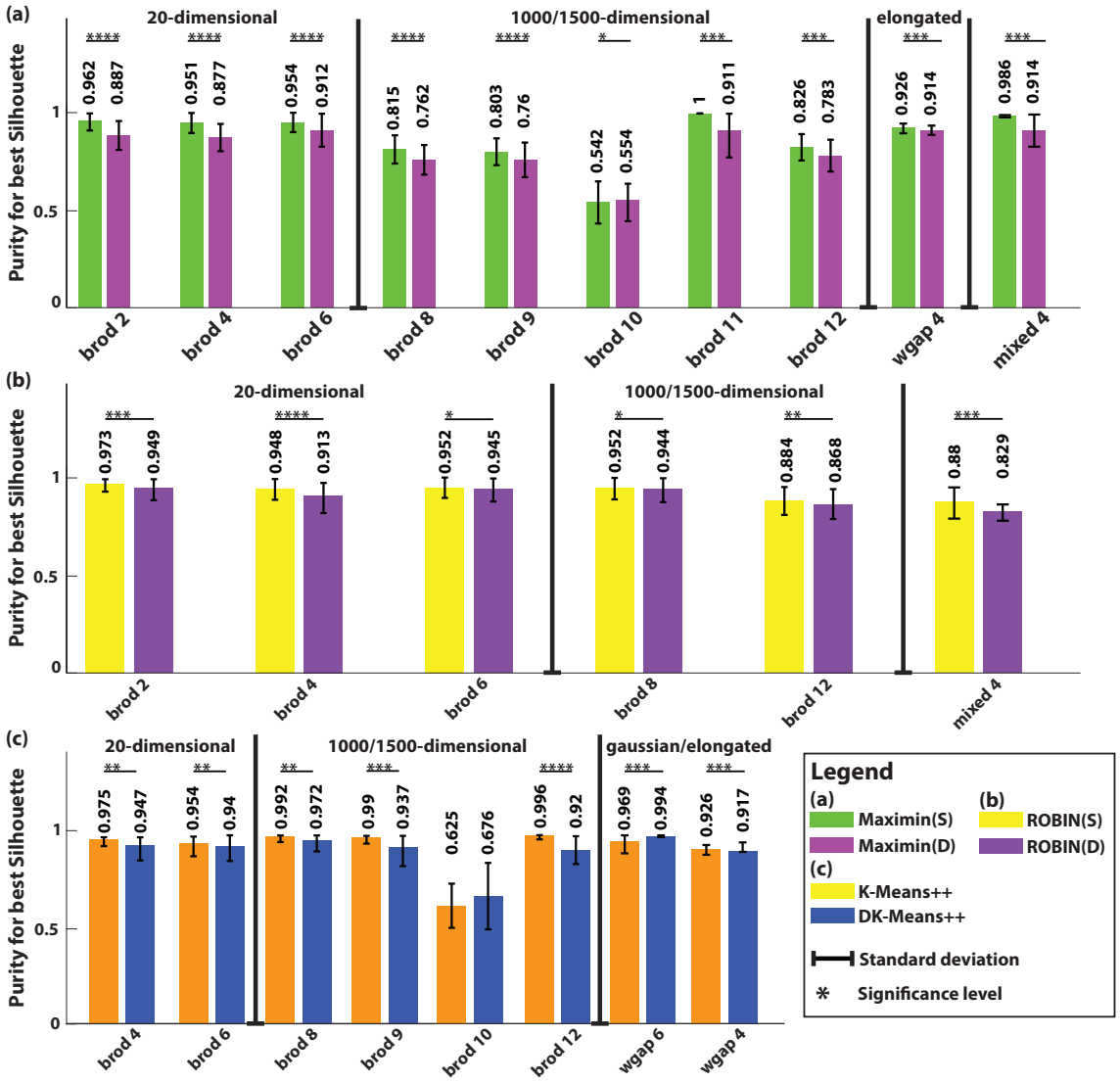
3.1.3.5 Average number of runs for which stochastic methods reach or surpass deterministic methods

The aforementioned experiments consider 50 executions of the clustering algorithm using stochastic methods. On average deterministic methods provide better results than stochastic methods but overall stochastic methods may lead to a better clustering solution. In this analysis it will be quantified how often this happens based on the following estimation: a division of the number of total repetitions (50) by the number of cases where the stochastic method performed better than the deterministic. Table 3.6 summarises the results of this analysis on selected data sets based on their



Initialization method	Total number of instances	Significantly better maximum performance Purity for best Silhouette			
		HW	Ll	KMed	Weis
Random vs K-Means++	26	0 vs 4	0 vs 4	1 vs 4	0 vs 5
Random vs ROBIN(S)	26	6 vs 3	8 vs 2	6 vs 2	6 vs 2
Random vs Maximin(S)	26	6 vs 1	9 vs 1	9 vs 1	6 vs 1
K-Means++ vs ROBIN(S)	26	7 vs 2	9 vs 2	8 vs 2	9 vs 1
K-Means++ vs Maximin(S)	26	6 vs 1	9 vs 1	9 vs 1	8 vs 0
ROBIN(S) vs Maximin(S)	26	4 vs 3	4 vs 4	4 vs 2	5 vs 5

Figure 3.5: The maximum performance of K-Means variations increases by using stochastic initialisation methods. Each plot shows the performance of the Hartigan-Wong's K-Means clustering solution using the purity corresponding to the best silhouette score achieved within 50 different executions (y-axis) on different data sets models (x-axis) and initialized with different stochastic methods. Purity for best silhouette score was averaged over the 40 data sets for each model (gap, weighted gap and Brodinova) The errorbars are showing the standard deviation across the 40 data sets. Solid lines on any two bars underline the level of significant difference between the corresponding methods (cases of no significant differences are not showing). The accompanied Table below the figure shows a summary of the comparisons through all the K-Means variations (Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld (Weis)) where there is a significant performance difference between the compared methods. Based on the results K-Means++ achieves the best solution followed by Random, Maximin(S) and ROBIN(S).



Initialization method	Total number of instances	Significantly better maximum performance Purity for best Silhouette			
		HW	Ll	KMed	Weis
Maximin(S) vs Maximin(D)	26	9 vs 1	11 vs 1	10 vs 1	12 vs 0
ROBIN(S) vs ROBIN(D)	26	6 vs 0	6 vs 0	6 vs 0	5 vs 0
K-Means++ vs DK-Means++	26	6 vs 2	6 vs 2	7 vs 2	9 vs 0

Figure 3.6: Stochastic methods can reach better performance with multiple runs than determinsitc methods. Each plot shows the performance of the Hartigan-Wong's K-Means clustering solution using the purity corresponding to the best silhouette score achieved within 50 different executions (y-axis) on different data sets models (x-axis) and initialized with different stochastic methods. Purity for best silhouette score was averaged over the 40 data sets for each model (gap, weighted gap and Brodinova). The errorbars are showing the standard deviation across the 40 data sets. Solid lines on any two bars underline the level of significant difference between the corresponding methods (only cases with significant difference are showed). The accompanied Table below the figure shows a summary of the comparisons through all the K-Means variations (Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld (Weis)) where there is a significant performance difference between the compared methods. Based on the results **(a)** and **(b)** the stochastic versions of Maximin and ROBIN can surpass the performance of their deterministic versions when executed multiple times; **(c)** K-Means++ the best stochastic method based on the maximum performance it can achieve over multiple runs (see Figure 3.5) surpass the performance of DK-Means++, the best determinstc method (see Figure 3.3).

Table 3.5: Comparison on K-Means variations using different initialisation methods Each row compares two K-Means variations (Hartigan-Wong's K-Means: HW, Lloyd's K-Means: Ll, K-Medians: KMed and Weiszfeld: Weis) initialised with the same method on 26 occasions (10 gap and weighted gap, 12 Brodinova and 4 mixed models). The comparison is based on the number of times that there was significant difference between the two methods on their maximum performance based on the purity for the best silhouette score. This score was computed by obtaining over 50 executions the best execution of each stochastic method and matching it to its respective purity and then averaging over the 40 data sets of each model (for deterministic methods this is the average purity over the 40 data sets of each model). Based on the results K-Medians appears to be the best performer surpassing the performances of Hartigan-Wong's and Lloyd's K-Means and has almost equivalent performance with Weiszfeld's algorithm; Lloyd's K-Means appears to be the worst performer and between Hartigan-Wong's K-Means and Weiszfeld's algorithm stochastic methods seems to favor the former and deterministic methods the latter. However, it should be highlighted that these differences among the alogirthms add up to 1.5% of purity difference in total.

Initialization method	**Total number of instances**	**Significantly better maximum performance**		
		HW vs Ll	**HW vs KMed**	**Ll vs KMed**
Random	26	4 vs 1	3 vs 7	1 vs 7
K-Means++	26	5 vs 1	2 vs 7	1 vs 4
ROBIN(S)	26	4 vs 0	1 vs 3	1 vs 7
Maximin(S)	26	2 vs 1	3 vs 5	2 vs 3
Total	**104**	**15 vs 3**	**9 vs 22**	**5 vs 21**
Kaufman	26	1 vs 1	2 vs 6	1 vs 6
DK-Means++	26	1 vs 0	2 vs 4	1 vs 5
ROBIN(D)	26	6 vs 0	2 vs 2	1 vs 6
Maximin(D)	26	7 vs 0	2 vs 2	1 vs 6
Total	**104**	**15 vs 1**	**8 vs 14**	**4 vs 23**

Initialization method	**Total number of instances**	**Significantly better maximum performance**		
		HW vs Weis	**Ll vs Weis**	**KMed vs Weis**
Random	26	5 vs 2	1 vs 3	2 vs 1
K-Means++	26	7 vs 2	1 vs 4	2 vs 2
ROBIN(S)	26	5 vs 1	1 vs 7	2 vs 1
Maximin(S)	26	6 vs 0	0 vs 4	0 vs 1
Total	**104**	**23 vs 5**	**3 vs 18**	**6 vs 5**
Kaufman	26	1 vs 5	1 vs 4	2 vs 2
DK-Means++	26	1 vs 5	0 vs 6	1 vs 1
ROBIN(D)	26	0 vs 3	0 vs 4	1 vs 1
Maximin(D)	26	1 vs 3	1 vs 6	2 vs 2
Total	**104**	**3 vs 16**	**2 vs 20**	**6 vs 6**

size, dimensionality and number of clusters among two stochastic (Random and K-Means++) and two deterministic methods (DK-Means++ and ROBIN(D)). Based on the results the number of repetitions required for the clustering method using K-Means++ in order to match or surpass the performance of DK-Means++ and ROBIN(D) are less compared with Random. This was expected given the performance comparison of Random and K-Means++ but an important result is the following: there are cases (Yeast, A-sets 2 and A-sets 3) where these stochastic methods fail to match or surpass the performance of deterministic methods under 50 runs. It is also observed that when the size of the data set surpasses the 1000 data points the number of required repetitions is significantly high. Finally it should be mentioned that these performance differences are minor, in the order of 10^{-3} on average.

3.1.3.6 Execution time analysis

Finally, an execution time analysis was performed on the initialisation methods using a selection of the data sets depending on their size, dimensionality and number of clusters; data sets with equivalent properties were omitted. The analysis was performed as follows: each initialisation methods followed by K-Means clustering (Lloyd's K-Means) was executed 50 times and the average running time was taken into consideration. Regarding this analysis the following aspects should be considered:

- The benchmarking was exclusively performed on a *personal laptop* with the following properties: Dell G7; Intel i7-9750H processor; 16 GB RAM; Windows 10 Pro edition.
- All the algorithms were written in MATLAB but the LOF score for ROBIN was computed using R code (specifically the dbscan package [Hahsler et al., 2019]) because MATLAB's implementation was very slow.
- The running time recording includes only the initialisation methods without the K-Means algorithm. For ROBIN the computation of LOF was included in the execution duration as well as the computation of ε for the DK-Means++

Based on the results in Figure 3.7 Kaufman is the worst method in terms of execution duration and it is affected both by the size, dimensionality and number of clusters. Randoma nd K-Means++ are the fastest methods followed by Maximin(S). DK-Means++ is almost always better than ROBIN(D) in terms of speed for our implementation.

Furthermore, and based on the results of Table 3.6 an additional analysis was performed on the time requirements of the stochastic methods to reach or surpass the performance of deterministic methods with multiple executions of the clustering algorithm using different seeds. Figure 3.8 shows the single run execution time of the stochastic initialisation (plus the clustering overhead) multiplied by the number of iterations required to surpass the deterministic methods (see Table 3.6). Based on the results shown in Figure 3.8 it is observed that in many occasions running DK-Means++ once is better in terms of execution time than repeated runs of the clustering with a stochastic method. Equivalent conclusions can also be obtained from the Maximin(S) initialisation method (see Appendix B, Figure B.3).

The number of iterations required for the clustering algorithm to reach convergence using stochastic and deterministic methods was also investigated. Based on the results

	Rand DKM++	**Rand ROBIN(D)**	**KM++ DKM++**	**KM++ ROBIN(D)**	**size, dimensions, number of clusters**
gap 2	5	5	4	4	100,2,3
wgap 2	6	6	4	4	115,2,2
wgap 4	6	6	5	5	200,2,2
wgap 3	3	3	3	3	200,2,4
gap 5	4	2	4	2	200,3,2
wgap 6	8	8	7	6	300,2,6
gap 3	5	5	4	4	*143,3,4*
gap 4	**10**	**10**	6	6	*158,10,2*
wgap 1	7	7	6	6	*227,2,6*
wgap 5	**14**	**14**	6	6	*141,10,2*
brod 1	4	4	5	5	120,20,3
brod 2	**18**	**17**	**17**	**17**	400,20,3
Iris	7	7	3	3	150,4,3
Wine	3	3	2	2	178,13,3
Glass	7	7	6	6	214,9,6
Ionosphere	3	3	2	2	351,34,2
Breast Cancer	**12**	**12**	7	7	683,9,2
Yeast	N/A	**16**	**27**	**14**	1484,8,10
A-sets 1	**28**	**26**	**26**	**16**	3000,2,15
S-Sets 1	**34**	**34**	**26**	**15**	5000,2,15
A-sets 2	N/A	N/A	**34**	**33**	5250,2,35
A-sets 3	N/A	N/A	N/A	N/A	7500,2,50

Table 3.6: Average number of runs for which stochastic initialisations achieve equivalent or better performance than deterministic initialisations. Each column shows a comparison between clustering initialised with stochastic and deterministic methods (Rand = Random, DKM++ = DK-Means++, KM++ = K-Means++). Each cell value corresponds to the number of executions of the K-Means clustering initialised with the stochastic method to reach or surpass its performance if it was initialised with the deterministic method and executed once. N/A values mean that in these occasions the stochastic clustering was not able to match or surpass the performance of the deterministic clustering under 50 executions. Values equal or more than 10 are marked in bold since in real world situations one would not normally repeat the same procedure more than 5 to 10 times. The data sets are arranged based on their *size*, *dimensionality* and *number of clusters* (see info on last column; numbers in italics correspond to the average number of elements should the model generated data sets of different sizes.

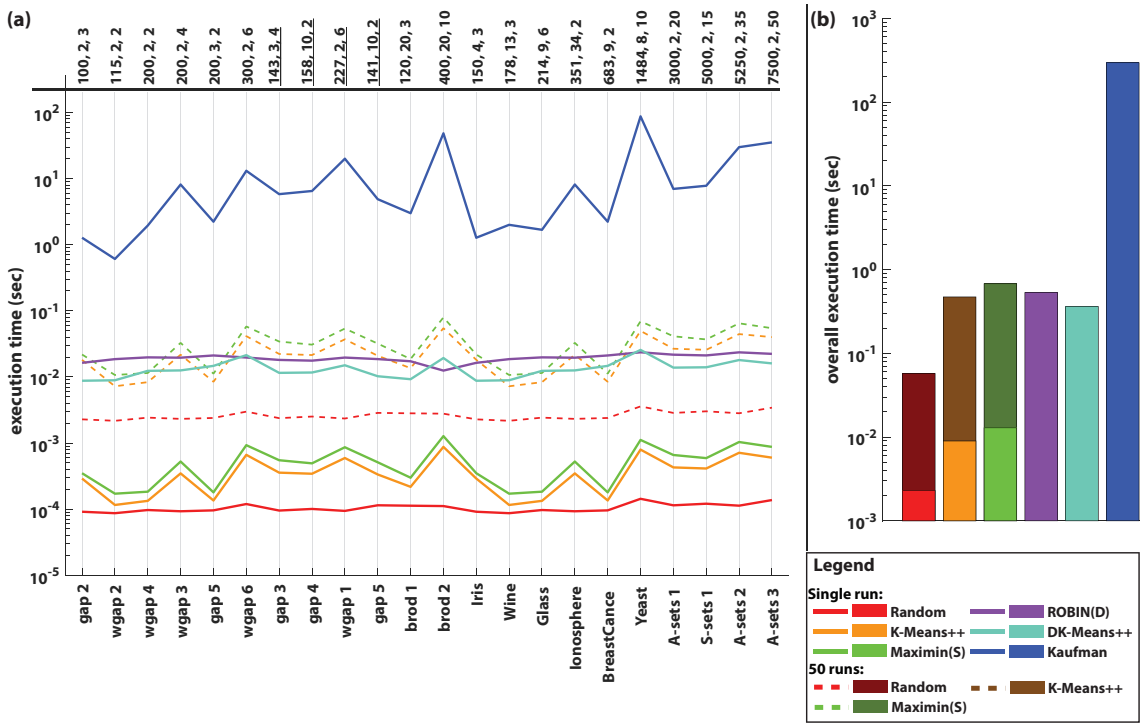


Figure 3.7: Execution time analysis. **(a)** Each line shows the execution duration of an initialisation method on different data sets selected based on size, dimensions and number of clusters. The average execution time is considered over 50 repetitions and, in case of models, across the 40 data sets of each model. The data sets are arranged based on their *size*, *dimensionality* and *number of clusters* (see info on top, underlined numbers means that for these models generate data sets of different sizes). **(b)** Each bar shows the summed execution time across all data sets of (a).

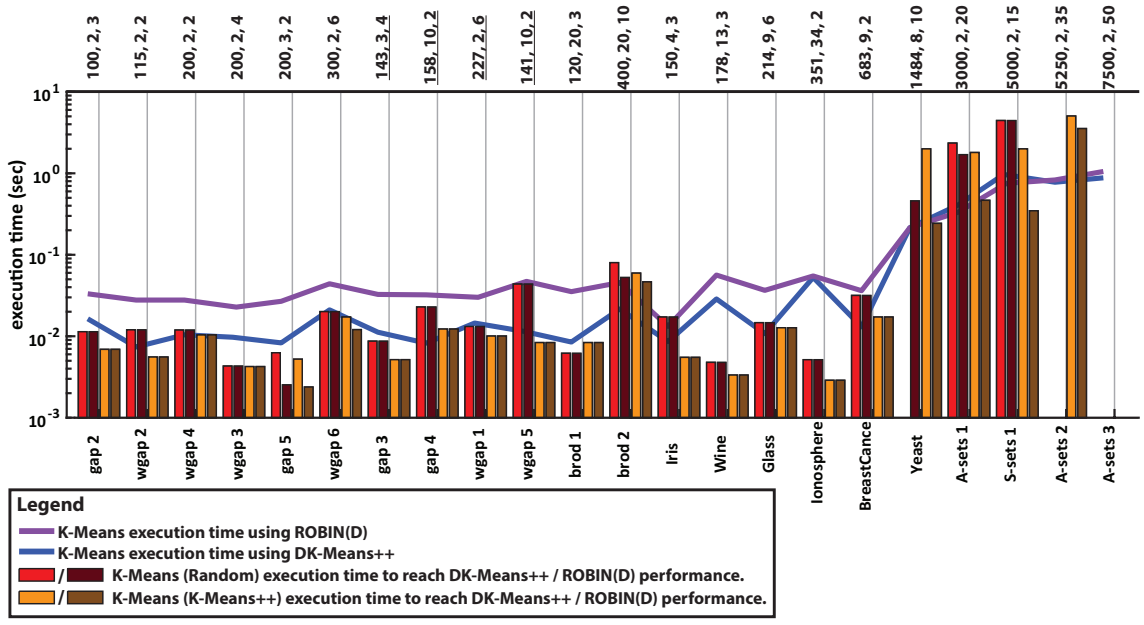


Figure 3.8: Execution time until stochastic methods reach or surpass the performance of deterministic methods. Each bar shows the running time requirements of the clustering algorithm initialised with a stochastic method to reach or surpass the performance of the same algorithm initialised with a deterministic method. The time requirements of the clustering algorithm using a deterministic method are shown as lines for comparison. Cases where bars are not shown mean that, up to 50 runs, the clustering algorithm using a stochastic method was unable to surpass the performance of the deterministic method.

(refer to Table 3.7) stochastic methods overall provide worst initial conditions (with the exception of ROBIN(S) vs Maximin(D)), and as a consequence the clustering algorithm requires more iterations to converge, which adds to the overhead of the stochastic initialisation methods (for detailed results see Appendix B, Figure B.1 for the stochastic methods and Figure B.2 for the deterministic methods).

3.1.4 Discussion

K-Means clustering remains one of the most common clustering techniques in many different research fields and frequently it is used as a component of more complex algorithms (e.g. hierarchical clustering [Jain, 2010]). Following similar benchmark studies on K-Means [Celebi et al., 2013; Fränti and Sieranoja, 2018, 2019], this study compares stochastic and deterministic initialisation methods on K-Means variations. In particular, the methods of ROBIN and DK-Means++ wew investigated since to the best of the author's knowledge they have not been studied as extensively as other initialisation methods. Experimentally this study showed:

- More sophisticated initialisation methods can lead, on average, to better clustering regardless of the K-Means variation (see Table 3.4). From the stochastic methods, ROBIN(S) can achieve the best average performance compared with Random, K-Means++ and Maximin(S) (see Figure 3.2). From the deterministic methods, DK-Means++ can achieve the best performance compared with Kaufman, Maximin(D) and ROBIN(D) (see Figure 3.3). In addition, DK-Means++ can achieve better performance from the average performance of stochastic methods (see Figure 3.4). Overall, deterministic methods have on average less performance variability across the data sets of each tested model and lead to more stable solutions than stochastic methods (see Appendix B) and can surpass the performance of stochastic methods (see Figure 3.4).

- When executed multiple times stochastic methods can achieve better performance than deterministic methods. Opposite the the first point, in that case, less sophisticated methods (such as Random and K-Means++ as opposed to ROBIN(S)) can achieve better performance (see Figure 3.6). K-Means++ with 50 executions achieved the best performance followed by Random (see Figure 3.5). The only deterministic method that can still compete to an extent is DK-Means++ (see Appendix B) where a full list of all comparisons is provided among all the initialisation methods considered in this study).

- As indicated by [Slonim et al., 2013] Hartigan-Wong K-Means is better than Lloyd's K-Means (see Table 3.5) and as shown by [Brusco et al., 2017] (only for one K-Means variant) K-Medians is better than both Hartigan-Wong and Lloyd's K-Means. However these differences add up to performance difference of only 1.5% as measured by the purity index.

- Regarding execution time requirements, Random and K-Means++ are fastest performers in terms of single runs while Kaufman the slowest (see Figure 3.7). Maximin(S) is slightly slower than K-Means++. Nevertheless these methods require multiple executions in order to reach the performance of

Initialization method	Total number of instances	Significantly more iterations for K-Means to converge
Random vs K-Means++	26	23 vs 0
Random vs ROBIN(S)	26	24 vs 0
Random vs Kaufman	26	23 vs 0
Random vs DK-Means++	26	25 vs 0
Random vs ROBIN(D)	26	24 vs 0
Random vs Maximin(S)	26	23 vs 2
Random vs Maximin(D)	26	23 vs 0
K-Means++ vs ROBIN(S)	26	22 vs 0
K-Means++ vs Kaufman	26	21 vs 2
K-Means++ vs DK-Means++	26	24 vs 0
K-Means++ vs ROBIN(D)	26	22 vs 0
K-Means++ vs Maximin(S)	26	21 vs 3
K-Means++ vs Maximin(D)	26	19 vs 1
ROBIN(S) vs Kaufman	26	11 vs 7
ROBIN(S) vs DK-Means++	26	14 vs 0
ROBIN(S) vs ROBIN(D)	26	3 vs 2
ROBIN(S) vs Maximin(S)	26	1 vs 21
ROBIN(S) vs Maximin(D)	26	1 vs 19
Kaufman vs DK-Means++	26	8 vs 4
Kaufman vs ROBIN(D)	26	7 vs 10
Kaufman vs Maximin(S)	26	3 vs 15
Kaufman vs Maximin(D)	26	3 vs 14
DK-Means++ vs ROBIN(D)	26	0 vs 15
DK-Means++ vs Maximin(S)	26	0 vs 20
DK-Means++ vs Maximin(D)	26	1 vs 18
ROBIN(D) vs Maximin(S)	26	1 vs 21
ROBIN(D) vs Maximin(D)	26	1 vs 17
Maximin(S) vs Maximin(D)	26	8 vs 1

Table 3.7: Summary of comparisons for the number of iterations until convergence for the Lloyd's K-Means algorithm using different initialisation methods. Each row shows a comparison between different initialisation methods on the number of times that each method resulted on the Lloyd's K-Means algorithm to have greater number of iterations until convergence. As indicator of performance, a method resulted to lower number of iterations is consider better, thus the lower the score the better the methods. Based on the results, ROBIN(S) is the best stochastic method and results to the less iterations for the K-Means algorithm to reach convergence; DK-Means++ is the best deterministic method and results to the less iterations for the K-Means algorithm to reach convergence. Overall, deterministic methods result to lower number of iterations for the K-Means algorithm.

determinsitic methods (refer to Table 3.6) especially with bigger data sets (number of elements to thousands). Multiple executions of these methods have almost similar requirements as a single run of deterministic methods DK-Means++ and ROBIN(D) (refer to Figure 3.8). This is due to the fact that the clustering algorithm requires more iterations to reach converge when stochastic methods are used (refer to the supplementary material). Between DK-Means++ and ROBIN(D) the former is faster than the latter.

Overall, and from a practical point of view, the stochastic Random and the deterministic Kaufman methods are not advisable. The first method despite being the simplest and the fastest can be replaced with K-Means++ that has better probability of achieving superior performance. The latter method is extremely slow and there are better alternatives such as the DK-Means++ that has both better performance and execution time. Maximin(D) and ROBIN(S) are not advisable either since the former is relatively fast and multiple executions of Maximin(S) can be performed instead while the latter has much more time requirements, small variability on its solutions and when an approximate clustering is required ROBIN(D) can be used instead. DK-Means++ is a good option when determinism is required since with a single run it can achieve better performance compared with other deterministic methods and comparable performance to multiple executions of stochastic methods that would require the same or more running time. In applications where exhausted search of optimal initial centroids needs to be performed K-Means++ should be considered (the study of [Celebi et al., 2013] has also benchmarked a greedy version of this method which is also recommended). In these cases if time requirements are flexible a strategy would be to perform first DK-Means++ which would give an indication about the clustering capabilities of the data set and then multiple executions of K-Means++. It should be added that in the mixed model 4, ROBIN(S) and ROBIN(D) performed significantly low compared with other cases because both where placing two initial centroids on the sides of the most elongated cluster while DK-Means++ were placing correctly a centroid almost in the middle of the cluster. This indicates that the DK-Means++'s heuristic might be more robust to applications than the LOF score of ROBIN for clusters detection. It should be noted that more complex techniques like DK-Means++ can be considered as clustering algorithms themselves since they produce good initial clusters. This observation was mentioned in the study of [Celebi et al., 2013] and in this thesis another study was performed (see Figure 3.7) on the number of iterations until the K-Means algorithm reaches convergence when it is initialized with different methods. Based on the results deterministic methods causes K-Means to converge faster than when it is initialized with stochastic methods. As expected, ROBIN(S) and DK-Means++ were again the best stochastic and deterministic methods on that account.

These conclusions were based on extensive benchmarking considering many different clustering models from other studies: Gaussian, high-dimensional (10 dimensions), elongated, unbalanced, non-Gaussian from the studies of [Tibshirani et al., 2001] and [Yan and Ye, 2007]; high-dimensional (20 dimensions) containing informative and uninformative features and higher-dimensional (1000 and 1500 dimensions) with varying number of clusters (3, 10, 50 clusters) and cluster sizes (50-100 points) [Brodinová et al., 2017]. This study also uses novel models containing clusters with different properties (unbalanced, elongated and Gaussian; unbalanced Gaussian and non-Gaussian; unbalanced, Gaussian with different variability among

their dimensions).

With the use of synthetic data set generators there is the ability to generate multiple data sets and run hypothesis testing to further strengthen the resulted conclusions but standalone data sets were also considered. The "clustering data sets" S-sets [Fränti and Virmajoki, 2006] and A-sets [Kärkkäinen and Fränti, 2002] were selected from the studies of [Fränti and Sieranoja, 2018, 2019] because both are containing more clusters and data points than the generated ones and also because in the case of the S-sets the clusters are having different overlap degrees. The conclusions that are obtained from the data set generators match with the conclusions of the standalone S-sets and A-sets data sets. Specifically the higher dimensional data sets (1000, 1500 dimensions) generated using the Brodinova generator [Brodinová et al., 2017] (see Table 3.2), are having small clusters due to the Kaufman initialization method which requires significant amount of time to be executed. However, data sets with larger clusters (approximately five times bigger) were also generated and the ROBIN(D) and DK-Means++ methods were tested on them. The results (not shown) and conclusions were similar to the ones reported already.

Based on the previous studies [Fränti and Sieranoja, 2018, 2019] the authors have clearly demonstrated that K-Means performs worse as the number of clusters increases (similar to our study the A-set was the data set with the larger number of clusters which were 50) and that dimensionality does not have a direct effect on the performance of the algorithm. In the experiments that use the Brodinova models (see Figures 3.3, 3.6 brod 1 to brod 12) it was observed that indeed the performance of all the methods drops when the number of clusters is increased regardless of the dimensionality, especially in the case of Brodinova brod 10 model where the generated data sets are having 50 clusters. Apart from the last extreme case, it was observed that multiple executions of stochastic methods improve the performance of K-Means. It should also be noted that the deterministic DK-Means++ method achieves (similar to multiple executions of stochastic methods i.e. Random, K-Means++, Maximin(S) and ROBIN(S)) the highest performance on the clustering basic benchmark [Fränti and Sieranoja, 2018, 2019] in all the cases (see Appendix B)) even thought these data sets have high number of clusters (A-sets: 20, 30, 50; S-sets: 15). The same authors [Fränti and Sieranoja, 2018, 2019] also demonstrated that strong cluster unbalances (i.e both dense and sparse clusters) affect negatively the K-Means clustering. In the experiments and superficially for the weighted gap 2 model it was observed that data sets with unbalanced clusters do not cause any particular issues to the maximum performances of the algorithms. For the performance between K-Means and K-Medians, similar to the results of [Brusco et al., 2017], it was found that K-Medians outperforms K-Means on synthetic data set models but on a small difference of 1% of purity and on standalone data sets (both synthetic and real-world) any particular differences among the K-Means variations couldn't be clearly detected.

In order to show application to "real world problems" previous studies have chosen to use standard classification data sets as benchmarks for clustering. While this approach is commonly used, in these data the mapping from classes to clusters is somehow forced: it is possible that data from one class belong to different clusters, and assuming that number of clusters equals number of classes is likely to underestimate the true number of clusters. This can be seen from the low value of the Silhouette index especially in the cases of Ionosphere and Yeast data sets. For this reason the conclusions were based mostly on the benchmark models that allows us to generate

multiple samples and evaluate the statistical significance of the results. In fact, there was consideration of a broad combination of different clusters, in terms of normality (Gaussian, non-Gaussian), shape (spherical, elongated) and size (clusters with different number of data points) including high dimensional data, as found in real world applications such as bioinformatics [Wang et al., 2008].

It should also be noted that many clustering frameworks designed to deal with complex data sets (e.g. sub-clustering [Biswas and Jacobs, 2014], or sparse clustering [Brodinová et al., 2017; Kondo et al., 2016; Witten and Tibshirani, 2010]) are using the K-Means or some variant of it and are dependent on good clustering initialisation. This experimental work revealed that there are deterministic methods (DK-Means++ [Nidheesh et al., 2017]) that lead to a good clustering solution with a single execution of the K-Means algorithm.

A limitation of the current study is that the execution time analysis is subject to the machine that executed it. More powerful machines or code optimisation of the algorithms and initialisation methods can change time analysis results. Nevertheless the rest of the analysis including the number of different seeds for stochastic methods to reach the performance of deterministic is, on average, reproducible. Statistics on average performance comparison are representative since,similar analysis had been also performed on 25 instances of the various data sets models instead of 50 and led to the same conclusions.

3.2 Comparison among K-Means inspired semi-supervised algorithms and the novel PCSK-Means algorithm

This study (refer also to [Vouros and Vasilaki, 2020]) considers the problem of data clustering with unidentified feature quality but with the existence of small amount of labelled data. In the first case a sparse clustering method can be employed in order to detect the subgroup of features necessary for clustering and in the second case a semi-supervised method can use the labelled data to create constraints and enhance the clustering solution. This work proposes a K-Means inspired algorithm that employs these techniques. It is shown that the algorithm maintains the high performance of other semi-supervised algorithms and in addition preserves the ability to identify informative from uninformative features. The study examines the performance of the algorithm on synthetic and real world data sets. A series of scenarios with different number and types of constraints as well as two different clustering initialisation methods is considered.

3.2.1 Introduction

In many learning tasks there is a plethora of unlabelled data in a high-dimensional space consisted of series of features that have some interpretation and a limited number of labelled data since the latter are expensive to be generated. In many cases there is no knowledge of the actual contribution of each features on the learning task and often conditionality reduction methods are employed in order to keep only the most relevant features to the given task. However, many dimensionality reduction methods, such as Principal Component Analysis (PCA) [Bro and Smilde, 2014;

Shlens, 2014], result in a transformation of the original features which limits their interpretability, especially if each feature has specifically designed to have a biological meaning. In unsupervised scenarios a number of authors [Maugis et al., 2009; Pan and Shen, 2007; Raftery and Dean, 2006; Wang and Zhu, 2008; Xie et al., 2008] have proposed clustering algorithms that have the ability to keep the initial features intact and assign a certain weight to them based on their contribution on clustering. These algorithms result in feature selection and sparse clustering.

In the work of [Witten and Tibshirani, 2010] a generic sparse clustering framework is presented which incorporates L_1 (Lasso regression) and L_2 (Ridge regression) penalties in order to eliminate the uninformative feature and weight the rest based on their contribution on clustering [Witten et al., 2009]. Such method requires the tuning of the *sparsity* hyper-parameter which essentially regulates the amount of L_1 application. This framework have been applied with K-Means and Hierarchical clustering but can also be applied to semi-supervised scenarios where pairwise constraints are given as an additional input to the algorithm, such constraints are generated from partly labelled data and indicate which data points should (MUST-LINK) or should not (CANNOT-LINK) belong to the same cluster. Previous work on semi-supervised learning [Bar-Hillel et al., 2003; Basu et al., 2002; Klein et al., 2002; Wagstaff et al., 2001; Xing et al., 2003] has indicated that incorporated constraints can result in superior performance of the learning algorithm. This is achieved by guiding the clustering solution either with the alternation of the objective function of the algorithm to include satisfaction of the constraints [Demiriz et al., 1999] or with the initialisation of the centroids to more appropriate locations of the feature space based on the constraints [Basu et al., 2002]. Another technique is to train a metric that satisfy the constraints as in [Xing et al., 2003] in which pairwise constraints were used to train a Mahalanobis metric.

Based on the previous work of [Witten and Tibshirani, 2010] on sparse K-Means clustering this study proposes a modification to the objective function of the algorithm to incorporate constraints. It is shown that using this method the best of both worlds can be achieved since constraints result to better clustering performance without affecting the sparsity capabilities of the algorithm. This novel algorithm is named as Pairwise Constrained Sparse K-Means (PCSK-Means) and it is tested under different conditions such as different number and kind of constraints (CANNOT-LINK, MUST-LINK or both). In the previous section (refer also to the study [Vouros et al., 2019]) the superiority of the deterministic initialisation method of Density K-Means++ (DK-Means++) was presented and thus this method was selected to initialise all the algorithms along with the seeding method proposed in the study of [Bilenko et al., 2004] and discussed in Chapter 2 (refer to 2.2.7.7). The benchmark includes synthetic data sets from the study of [Brodinová et al., 2017] with known feature quality and real world data sets from the UCI [Asuncion and Newman, 2007] and a real world data set from the behavioural neuroscience study of [Vouros et al., 2018] which contains ten known uninformative features.

The data sets used in this study, including their constraints, and the MATLAB code to run the simulations are available on the GitHub repository https://github.com/avouros/Code-PCSKM.

3.2.2 Methods

This work will make use of the algorithms described in Chapter 2. The clustering algorithms are as follows: Lloyd's K-Means (see 2.2.2.2), Sparse K-Means (see 2.2.3, PCSK-Means (see 2.2.4.1), MPCK-Means (see 2.2.4.2) and PCSK-Means that will be described next. The clustering initialisation methods are as follows: Density K-Means++ (DK-Means++) (see 2.2.7.6) and Seeding (see 2.2.7.7). F-score (see 2.2.8.2.2) was used for evaluation.

3.2.2.1 The Pairwise Constrained Sparse K-Means Algorithm

The MPCK-Means algorithm (refer to 2.2.4.2) is using the pairwise constraints in order to learn a metric capable of shaping the feature space in a way that clusters are formed based on both the similarity of the elements (features) as well as their constraints (which elements should or should not be together). This use of constraints can be incorporated by Sparse K-Means clustering to create a semi-supervised algorithm with feature selection capabilities.

Using the constraints and the sparse clustering the objective function of equation 3.1 is defined,

$$\underset{c_1,\ldots,c_k,w}{maximize}\Bigg\{\sum_{j=1}^{p} w_j\Bigg[\sum_{i=1}^{n}(x_{ij}-\mu_{1j})^2-\Bigg(\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}(x_{ij}-m_{kj})^2+ \sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij}-x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+ \sum_{(x_{i:})CL(x_{i'_:})} \bar{b}_{x_i,x_{i'}}\big((x_{Ij}-x_{I'j})^2-(x_{ij}-x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)\Bigg]\Bigg\} \tag{3.1}$$

$$\text{subject to} \qquad \sum_{j=1}^{p} w_j^2 \le 1, \;\; \sum_{j=1}^{p}|w_j| \le s, \;\; w_j \ge 0 \;\; \forall j$$

where the terms inside the outer parenthesis is the WCSS that K-Means minimizes based on constraints (see J_{pckm} equation 2.12).

Based on the Sparse K-Means problem given in 2.8, the PCSK-Means problem can be specified as,

$$\underset{w_j}{maximize}\Big\{\sum_{j=1}^{p} w_j\gamma'_j\Big\} \quad \text{subject to} \qquad \sum_{j=1}^{p} w_j^2 \le 1, \;\; \sum_{j=1}^{p}|w_j| \le s, \;\; w_j \ge 0 \;\; \forall j \tag{3.2}$$

where,

$$\gamma'_j = \sum_{i=1}^{n}(x_{ij} - \mu_{1j})^2 - \sum_{k=1}^{K}\sum_{\substack{i=1 \\ x_{i:}\in c_k}}^{n_k}(x_{ij} - m_{kj})^2$$
$$\sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij} - x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big] +$$
$$\sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big((x_{Ij} - x_{I'j})^2 - (x_{ij} - x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big] \tag{3.3}$$

Analogously to Sparse K-Means 2.2.3, the solution to the convex problem of 3.2 is,

$$w_j = \frac{sign(\gamma'_j)(|\gamma'_j| - \Delta)_+}{\sqrt{\sum_{j'=1}^{p}\big(sign(\gamma'_{j'})(|\gamma'_{j'}| - \Delta)\big)^2}} \tag{3.4}$$

which follows the same proof as the one in the Appendix A.8 accounting also the constraints. In addition, a similar algorithm used for the Sparse K-Means (refer to Chapter 2, section 2.2.3) can be used for the PCSK-Means with minor modification to include the penalty of the imposed constraints,

Pairwise Constrained Sparse K-Means (PCSK-Means) algorithm

1. Given a dataset, number of clusters K, MUST-LINK and CANNOT-LINK constraints and constraints costs (optional), initialise K initial centroids $M = \{m_{1j}, \ldots, m_{Kj}\}$ using some initialisation method and the feature weights as $w_1 = \cdots = w_p = \frac{1}{\sqrt{p}}$ [Witten and Tibshirani, 2010].

2. Holding the weights fixed, maximize 3.1 with respect to M. This can be implemented by performing an equivalent algorithm as in K-Means where the point assignment to the nearest cluster is given by equation 3.5,

$$k^* = \underset{k}{argmin}\Bigg\{\Bigg(\sum_{j=1}^{p} w_j(x_{ij} - m_{kj})^2 +$$
$$\sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij} - x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big] +$$
$$\sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big((x_{Ij} - x_{I'j})^2 - (x_{ij} - x_{i'j})^2\big) \;\; \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg\} \tag{3.5}$$

3. Step 3 of Sparse K-Means algorithm replacing γ_j with γ'_j.

4. Step 4 of the Sparse K-Means algorithm.

The algorithm returns the final clusters (centroids and elements) and the weight of each feature.

One important note is that K-Means and Sparse K-Means are element order invariant meaning that with the same initialisation method (and the same random

seed if the centroids initialisation method is stochastic) it will produce the same results. This is true for PCSK-Means but only if the elements are processed in the same order due to the constraints. Based on the experimental study using the benchmark described next, shuffling the elements does not have any significant effects on the performance of the algorithm.

3.2.2.2 Benchmark

The benchmark of this study includes the real world data sets fisheriris, ionosphere and digits from the UCI repository [Asuncion and Newman, 2007] which have unknown feature quality. From the original digits data set 10 different sub-sets were created by randomly sampled elements of three classes, 0-4-8 and 3-8-9, 50 elements per class (3-8-9 digits were also used in the study of [Bilenko et al., 2004]). Two synthetic data sets were also added. These data sets were generated based on the generator of [Brodinová et al., 2017] which was used before (refer to section 3.1.2 and the published work [Vouros et al., 2019]). These synthetic data sets consist of known informative and uninformative features without noise. More specifically, they consist of 120 10-dimensional data points spread equally across 3 clusters. The first set has 5 informative and 5 uninformative features while the second 3 informative and 7 uninformative features. Finally, two reduced data sets from the studies of [Gehring et al., 2015; Vouros et al., 2018] were used that consist of 8 features of unknown importance that describe rodent path segments in the Morris Water Maze experiment. The first set (named TT-SC-ST) contains a total of 424 data points and 3 classes (Thigmotaxis: 168, Scanning: 182 and Scanning Target: 74 data points); the second set (named TT-CR-ST) contains a total of 406 data points and 3 classes (Thigmotaxis: 168, Chaining Response: 164 and Scanning Target: 74 data points). The dimensionality of both sets was increased with the addition of 10 features. The 9^{th} feature is the path segment length which is uninformative given the fact that all the segments were created to have approximately the same length and the next 9 features were generated from random shuffling of the segment length feature.

Table 3.8: Data set constraints. Column *Points* shows the total amount of data points of each data set; *CV10 labels* shows the number of labels that can be used for training (90% of the Points since the 10-fold cross validation was used); *CV10 constraints* shows the total number of constraints generated from these labels, i.e. the size of the pool. The last column refers to the number of constraints (minimum and maximum) that were randomly sampled from the pool and used in the training. The minimum corresponds to the 1% and the maximum to the 10% of CV10 constraints. MWM stands for Morris Water Maze.

Data set	Points	CV10 labels	CV10 constraints	CV10 constraints [1% , 10%]
fisheriris	150	135	9045	[90 , 905]
ionosphere	351	316	49738	[497 , 4974]
digits 0-4-8 and 3-8-9	150	135	9045	[90 , 905]
MWM TT-SC-ST	424	382	72618	[726 , 7262]
MWM TT-CH-ST	406	365	66576	[666 , 6658]
Brodinova (2 sets)	120	108	5778	[58 , 578]

3.2.3 Results

The first experiment assesses the performance of PCSK-Means (PCSKM) compared with other unsupervised algorithms (Lloyd's K-Means, LKM and Sparse K-Means, SKM) and semi-supervised algorithms (PCK-Means, PCKM and MPCK-Means, MPCKM). The deterministic initialisation technique of DK-Means++ (DKM++) [Nidheesh et al., 2017] was used for all the algorithms. The semi-supervised algorithms were also tested with different number and types of constraints including, only MUST-LINK, only CANNOT-LINK, and random selection from both MUST-LINK and CANNOT-LINK. Finally all the algorithms initialised with the seeding method of [Basu et al., 2002] (refer to section 2.2.7.7) were tested considering only random selection from both MUST-LINK and CANNOT-LINK constraints in order to have a direct comparison with [Bilenko et al., 2004].

For this experiment the results for the fisheriris, ionosphere and the Morris Water Maze data sets are shown since all the algorithms, as expected, performed equally well on the synthetic data due to their distinctive clusters. Figure 3.1 shows the results of the performance comparison. Our algorithm (PCSKM) was designed for its feature selection property rather than its performance and yet it is demonstrated that it has better performance in most cases and, where its performance is worse, the difference to the best performing algorithm is relatively small. Interestingly the use of only MUST-LINK constraints has a negative effect in the performance of all the semi-supervised algorithms except for the digits data sets. Using either ROBIN or Maximin initialisation leads to similar conclusions (see Appendix B, Figures B.4, B.5, B.6 and B.7).

The performance was tested using a similar evaluation as the one used in [Basu et al., 2004; Bilenko et al., 2004] (see also section 2.2.8.3). The 10-fold cross validation was executed using all the data but splitting the labels into training and test sets. The performance on each fold was assessed based on the F-score, an information retrieval measure, adapted for evaluating clustering by considering same-cluster pairs similar to [Bilenko et al., 2004]. The clustering algorithm was run on the whole data set, but the F-score was calculated only on the test set. Results were averaged over 25 runs (each run with a random selection of constraints) of 10 folds. For the number of given constraints 1% to 10% with step of 0.5% of the constraints generated from the training labels of each fold (Table 3.8) were given as input. This was done so that all the data sets are tested under equal conditions proportional to their sizes. Sparse algorithms (i.e. SKM and PCSKM) have one parameter than needs to be tuned, the sparsity parameter s. A value for this parameter was selected in the following way: The clustering process for $s = 1.1$ to $s = \sqrt{p}$, where p is the dimensionality of the data set with step 0.2 was executed. Among these s values the one that yields the best value of the F-score was selected.

To quantify the overall performance of PCSKM versus other algorithms, for each initialisation method (Seeding, DKM++, ROBIN, Maximin) constraints type (both, MUST-LINK only and CANNOT-LINK only) and data set (6 data sets in total) the average performance over the number of constraints (60 cases in total, since Seeding has only both constraints) was calculated per algorithm (PCSKM, PCKM, MPCKM, KM, SKM). Afterwards the Paired Samples Wilcoxon Test was used hypothesizing that the performances between any algorithm and PCSKM have the same distributions with the same medians. A p-value less than 0.05 in our analyses, lead us to discard the null hypothesis. Based on the results, there is significant

difference between PCSKM and KM ($p < 1^{-10}$), SKM ($p < 1^{-10}$), PCKM ($p < 1^{-10}$), MPCKM ($p < 0.001$) in favor of the PCSKM.

The performance difference among the different types of constraints (both, MUST-LINK only and CANNOT-LINK only) was also quantified in a similar manner as was done with the overall performance. For each initialisation method (DKM++, ROBIN, Maximin), semi-supervised algorithm (PCKM, MPCKM, SKM, PCSKM) and data sets (6 in total excluding the synthetic), the average performance over the number of constraints (72 cases in total) was calculated. Afterwards the Paired Samples Wilcoxon Test was used setting again a p-value less than 0.05 to discard the null hypothesis. Based on the results, there is significant difference between MUST-LINK only and CANNOT-LINK only constraints in favor of the latter (p-value < 0.01) and there is significant difference between MUST-LINK only and both constraints in favor of the latter (p-value < 0.001). No significant difference was detected between the CANNOT-LINK only and both constraints (p-value > 0.1).

The second experiment assesses the feature selection capabilities of the novel algorithm. Two synthetic data sets and the Morris Water Maze data set were used since, for them, there is knowledge towards the features of importance. The results are shown in Figure 3.10 where the feature selection capabilities of MPCKM, SKM and PCSKM are shown. For SKM and PCSK average results are shown over different values of s from 1.1 to $\sqrt{p}$, where p is the dimensionality of the data set, with step 0.2 as in [Brodinová et al., 2017]. This is to demonstrate that the feature selection does not strongly depend on optimally selecting the value of s. The weight values are plotted as a function of number of constraints (where applicable) demonstrating that the feature selection capabilities of PCSKM are not affected by it. Furthermore, the digits data sets were contaminated with 4 uninformative features generated from exponential distributions (see Appendix B, Figure B.8) and again the PCSKM algorithm was able to assign a 0 weight to them. On the contrary, the MPCKM algorithm, which learns a metric, makes use of uninformative features, see Figure 3.10 Seeding MPCKM. In addition, in both contaminated digit sets, the uninformative features have on average higher weights that the original features, and this is true across all initialisation methods (Seeding, DKM++, ROBIN, Maximin).

3.2.4 Discussion

This study proposes a modification on an existing clustering algorithm with a feature selection mechanism to address semi-supervised problems where few labels are available. It was hypothesized that the ability of this algorithm in detecting informative and non-informative features is better than an alternative semi-supervised algorithm with metric learning, while its performance will be at least equal. It is also demonstrated that the constraints improve the performance of our algorithm on classification problems in comparison to its unsupervised version.

The modified algorithm (PCSKM) was tested under different initial conditions. It was found that its performance is almost equivalent to other semi-supervised algorithms and in many cases better, regardless of the initialisation method (DKM++, MPCKM, ROBIN or Maximin), type (MUST-LINK only, CANNOT-LINK only, both) or number of constraints. It is shown that its feature selection mechanism is robust to the initialisation method and the number of constraints, and its weight assignments can be used to indicate informative or uninformative features in all the

data sets that were tested. On the contrary the MPCKM algorithm, which learns a metric, while it performs best or in par with PCSKM when the Seeding initialisation method is used (with the exception of the data sets ionosphere and digits 0-4-8), it makes use of the uninformative features. As a consequence the learned weights cannot be used to evaluate the quality of the features.

It is also shown that the performance of semi-supervised algorithms can be affected by initialisation procedures, similar to unsupervised methods [Vouros et al., 2019], and by the type of constraints. By experimenting with different number of constraints, different initialisation methods and different semi-supervised algorithms conclusions can be drawn about the goodness of each constraints type. Only MUST-LINK constraints have a negative effect on the performance for all the data sets that were tested (apart from the digits). It is speculated that this may be the case because data points belonging to the same class do not necessarily belong to the same cluster, while CANNOT-LINK constraints are more informative: data points belonging to different classes should not belong to the same cluster. Nevertheless, PCSKM could cope with the MUST-LINK constraints much better than MPCKM and PCKM.

In addition, the Seeding initialisation method proposed for the MPCKM in the study of [Bilenko et al., 2004] has mostly a positive effect on that particular algorithm. In the case of Morris Water Maze data sets, the initial 8 features were also engineered to perform best with the MPCKM algorithm [Gehring et al., 2015]. Nevertheless, the proposed PCSKM algorithm maintains a close performance to MPCKM and based on our hypothesis testing, it has on average the best overall performance. This is, perhaps, due to the use of quality features that should have in general a positive effect on the overall clustering performance.

Finally, the proposed PCSKM algorithm, similar to SKM, requires an additional sparsity parameter s. In the study of [Brodinová et al., 2017] both s and the number of clusters K are chosen using the gap statistic. Due to the semi-supervised setting, K can be set equal to the number of classes and the F-score can be used to select the sparsity parameter s avoiding the computationally more expensive gap criterion.

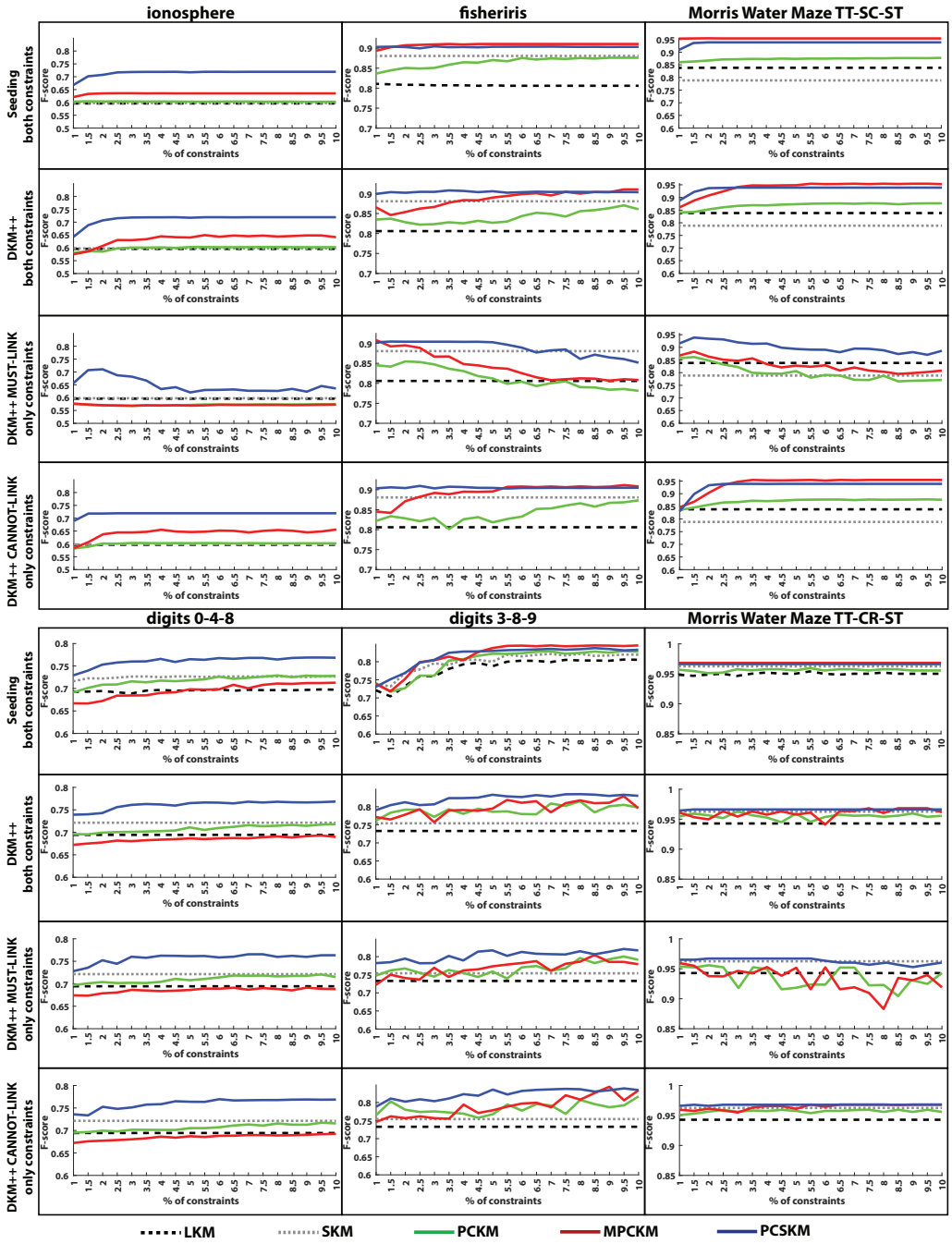


Figure 3.9: Performance of PCSKM as opposed to other unsupervised and semi-supervised algorithms. Each row compares the algorithms on six data sets (ionosphere, fisheriris, digits 0-4-8 and 3-8-9 and Morris Water Maze TT-SC-ST and TT-CH-ST) using different types of constrains. First row (Seeding both constraints): Seeding was used for cluster initialisation and a random selection from all the constraints, both MUST-LINK and CANNOT-LINK. Second row (DKM++ both constraints): similar as before but DKM++ initialisation was used. Third and fourth rows (MUST-LINK, CANNOT-LINK): DKM++ initialisation was used and a random selection of only MUST-LINK or CANNOT-LINK. For the SKM and PCSKM the sparsity value with the best performance was selected. For the ionosphere and digits 0-4-8 data, our algorithm has a clear advantage compared with the other methods. For the fisheriris, digits 3-8-9 and Morris Water Maze TT-SC-ST data, cluster initialisation with Seeding offers an advantage to the MPCKM algorithm compared to the DKM++ initialisation method. For all the data sets apart from the digits, using only MUST-LINK constraints has a negative effect. Clearly, the type of constraints can greatly affect the clustering performance while the initialisation method has less effect apart from the case of the MPCKM algorithm. PCSKM is in general more robust to initial conditions and its performance either surpass or is close to the performance of the other algorithms.

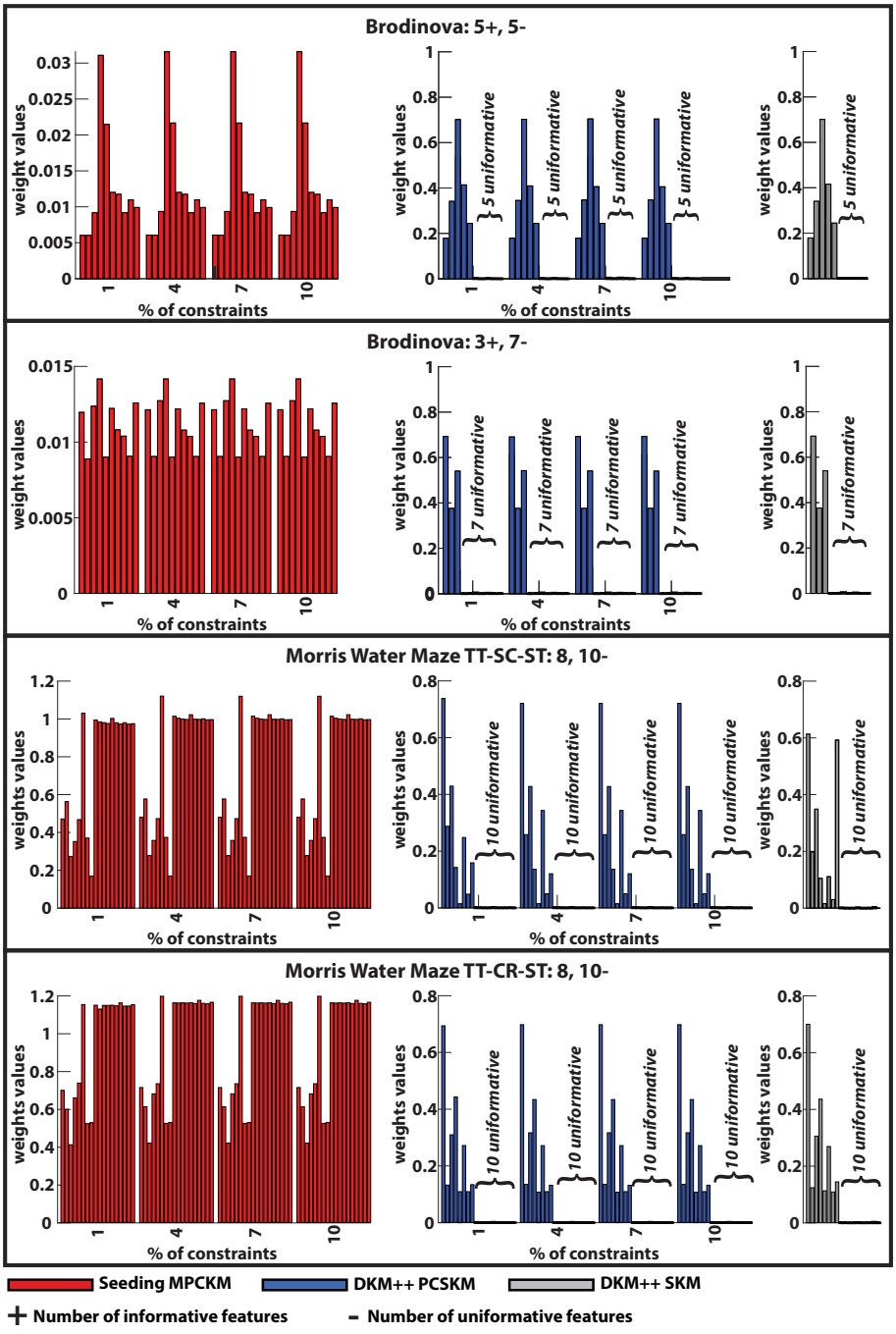


Figure 3.10: PCSKM feature selection capabilities as opposed to other algorithms in synthetic and the Morris Water Maze data sets using both constraints. Red bars: MPKM (semi-supervised), blue bars PCSKM (semi-supervised), gray bars: SKM (unsupervised). For SKM and PCSKM the bars show the average weight value of a feature over different sparsity (s) values (from $s = 1.1$ to $s = \sqrt{p}$, where p is the dimensionality of the data set, with step 0.2). The + and − signs indicate the number of informative and uninformative features (uninformative features are always plotted last). In the case of the Morris Water Maze the quality of the first 8 features is unknown but the last 10 are uninformative. The SKM and PCSKM correctly identifies the known uninformative features regardless of the choice of the s parameter in all the cases. The feature selection mechanism of the PCSKM is not affected by the number of constraints. The weights of the MPCKM algorithm are not indicative of the feature quality and in all the cases the algorithm uses the uninformative features. The plots shows only the case when both type of constraints are used but the same result is observed for the other constraint types cases regardless of the initialisation method.

Chapter 4

Manual behavioural difference detection using path features

This chapter is split into two sections. The first section contains a detailed listing of features that can be extracted from path coordinates and be used to capture animal behaviours or indicate behavioural differences among different animal groups. The second section is devoted to the published work of Chhabria, Vouros et al. [Chhabria et al., 2019] where some of these features were used for the manual detection of behavioural differences in zebrafish larvae inside a light/dark preference task. For more information about this particular experimental procedure refer to 2.1.3 and 2.3.2.

4.1 An overview description of generic path features

In Chapter 2 it was described that path analysis in behavioural experiments is traditionally performed by performance measurements. Such measurements can provide insights of different behavioural motifs or indicate certain classes of behaviour and Machine Learning techniques make use of them to perform automatic classification.

Here, there will be an attempt to list and group some of these measurements which will be referred to as features. These features are generic and applicable to any experimental procedure involving path tracking of animals inside constrained environments and do not require timestamps in order to be computed.

4.1.1 Geometry concepts

Given a set of points in the plane the minimum enclosing ellipsoid is defined as the unique closed ellipse of smallest volume which enclose these points [Gärtner and Schönherr, 1998; Todd and Yıldırım, 2007]. The unique circle which passes through the vertices of a triangle formed by any three points is called circumcircle [Weisstein, n.d.]. Refer to Figure 4.1 for a graphical illustration of the aforementioned concepts.

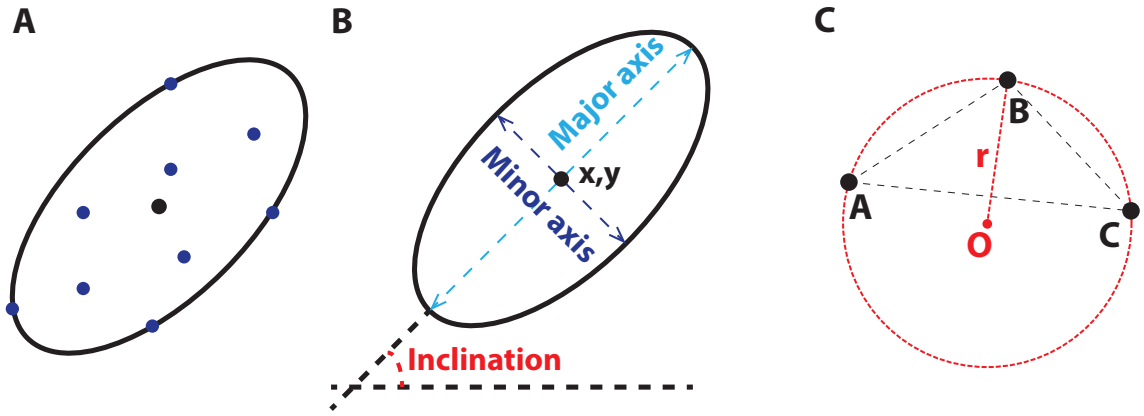


Figure 4.1: *Minimum enclosing ellipse and circumcircle.* **A.** The minimum enclosing ellipsoid over a set of points; blue dots are the points and the black dot corresponds to the center of the ellipse. **B.** Ellipsoid enclosing metrics: center of the ellipse (x,y); major and minor axes of the ellipse; inclination of the ellipse. **C.** Let A, B and C be three points in space which form the triangle $\triangle ABC$ if are used as vertices. The unique circle O with radius r which passes through the vertices A, B and C is then called circumcircle.

4.1.2 Geometric features engineering

These features aims to capture geometric aspects that specify the shape of the paths. These features require only path x and y coordinates in order to be computed and assume consistency on the lengths of the paths under comparison. Refer to Figure 4.2 A for a graphical illustration of the measuremtns used to form these features.

- *Path focus.* A measurement of how concentrated a path is on specific locations. It is defined as $f = 1 - \frac{4A}{\pi l^2}$, where A is the area of the minimum enclosing ellipse if the path and l is the total length of the path. With this definition a focus of 0 means that the path is perfectly circular; larger focus values give an indication of increasingly closed paths [Gehring et al., 2015].

- *Path eccentricity.* A measurement of how elongated are the paths. It is defined as $\epsilon = \sqrt{1 - \frac{b^2}{a^2}}$, where a and b are the semi-major and semi-minor axis of the enclosing ellipse of the path [Gehring et al., 2015]. An eccentricity value of 0 means that the path is perfectly circular while more oval-shaped enclosing ellipse will yield larger eccentricity values.

- *Path loops.* A path loop is the result of self-intersecting sub-segments. To compute a path loop, all pairs of lines defined by two consecutive path points are tested for intersection. Identifying all the path loops can result to the generation of a series of features such as number of loops, average length of loops and length of the longest loop (the latter feature was used in the study of [Gehring et al., 2015]).

- *Path distance to center of the enclosing ellipse.* Statistics such as the average or the variation of the distances of each point of the path to the center of the enclosing ellipse can indicate useful information and capture focus exploration on a certain location or non-strategic movements through an area. The study

of [Gehring et al., 2015] used a formula for the coefficient of variation over the distances, named inner radius variation. It is expected that the distances of more elongated paths will have more spread values thus this family of features can indicate more or less focused paths along with a degree of how much focused they are.

- *Path sinuosity.* Sinuosity is a measurement of the tortuosity of a path. It can simply be the straightness of the path [Almeida et al., 2010], which is defined as the distance between the first and the last point of the path divided by the total length of the path [Benhamou, 2004]. The study of [Benhamou, 2004] suggests that such simplistic measurement is not enough for paths that are completely random or exhibit both random and targeted behaviours and proposes a more sophisticated index of sinuosity can take into account the distribution of the turning angles. In the aforementioned study there are two indexes described by the equations 4.1 and 4.2

$$S = 2 * \sqrt{p * \frac{1+c}{1-c} + b^2} \qquad (4.1) \qquad S = 2 * \sqrt{p * \frac{1+c^2-s^2}{(1-c)^2+s^2} + b^2} \qquad (4.2)$$

where p is an expected step length and b the coefficient of variation of p and s and c are the mean sine and cosine of the turning angles. Given these formulas equation 4.2 is more generic while equation 4.1 assumes a Gaussian distribution of the turning angles since $s = 0$. A way to adjust the given equations for path coordinates could be to set p equal to the average of the lengths between the successive points forming the path and b to the ratio of the standard deviation of the lengths and p.

- *Path curvature.* The curvature is computed between any three points and it is equal to the radius of their circumcircle. When we have a sequence of points then a sequence of curve radii can be computed for each segment formed by two successive points. The curve radius of each segment is then equal to the average of the two curve radii which specify the segment. Finally the curvature of a path as a whole is normally computed by adding the length of the segments whose radius is less than a certain threshold [Franco, 2016] (refer to Figure 4.2 C for a graphical illustration of the circumcircle). Curvature is expected to be a measurement of the straightness of the path, thus it can differentiate random or targeted movements.

4.1.3 Spatial features engineering

Spatial features capture location aspects of the path in relation to a specified experimental area which is normally circular or sometimes square. Similar to geometric features they require only path x and y coordinates in order to be computed. Refer to Figure 4.2 B for a graphical illustration of the measuremtns used to form these features.

- *Path distance to center of the arena.* The distance to the center of the arena is a useful measurement that indicates if the animal spends most of its time next

to the walls or to the more central parts of the arena. Statistics such as the average or the variation of these distances can then be used as features as in the study of [Gehring et al., 2015].

- *Central displacement.* The central displacement is the Euclidean distance of the centre of the minimum enclosing ellipsoid to the centre of the arena, dividing with the arena radius. This measurement may be used to identify concentric paths with the arena [Gehring et al., 2015].

- *Path angle to center.* Using the center of the arena as a reference point the angle between each point and the arena center can be calculated. Statistics such as the average or the variation of these distances can be considered as features.

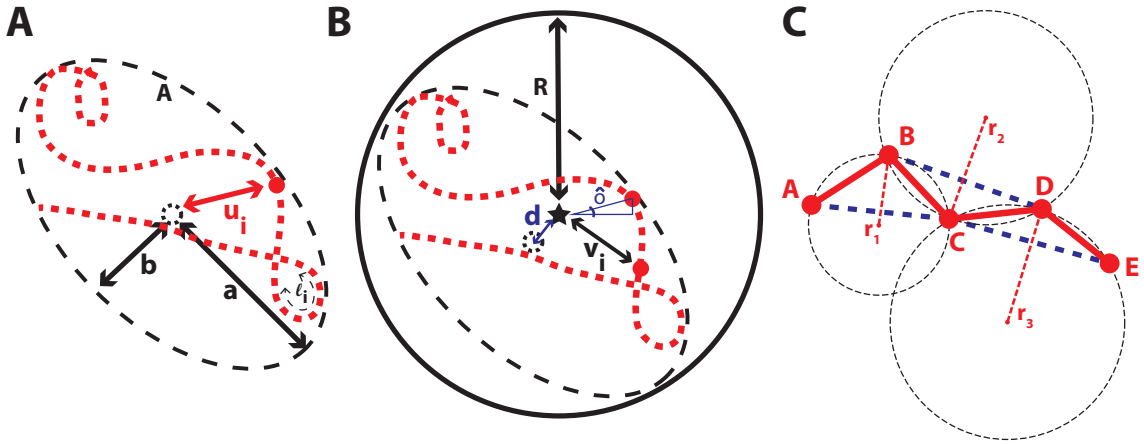


Figure 4.2: A graphical illustration of various metrics for feature engineering. A. The minimum enclosing ellipsoid of a path. Dotted red line indicates a path; dashed black line indicates the minimum enclosing ellipsoid (MEE) of the path with center at the dotted circle, major axis a, minor axis b and area A; u_i is the distance between the center of the MEE to a particular point of the path; ℓ_i indicates a path loop. **B.** Metrics of path in relation to a constrained environment. Solid circle indicates a circular constrained environment, centered at the star, with radius R; d is the distance between the MME of the path and the center of the constrained environment; v_i is the distance between the center of the constrained environment and a particular point of the path; $\hat{o}$ the angle formed between a particular point of the path and the center of the constrained environment. **C.** The curvature of a path. Let the points A to E be coordinates of a path illustrated by red line. For each sequence of three points there is a intersecting circle with radius r and a triangle formed by connecting the first to the last of the three points. The segment formed between each two points is then having a curve radius equal to the average radius of the radii for the triangles it is part of. For example The curve radius of segment BC is equal to $\frac{r_1+r_2}{2}$ and the curve radius of segment CD is equal to $\frac{r_2+r_3}{2}$. In this example the first segment AB and the last segment DE are having curve radii equal to r_1 and r_3 [Franco, 2016].

4.2 Manual behavioural analysis in zebrafish larvae inside the light/dark preference task using Morris Water Maze path features

In this study [Chhabria et al., 2019] a series of features used before for behavioural analysis in the Morris Water Maze with rodents were adopted and used to quantify

behaviour differences among four groups of zebafish larvae in the light/dark task. This research aimed to study the effects of glucose exposure on the neurovascular unit and behaviour of zebrafish.

4.2.1 Introduction

Diabetes is a disease associated with high levels of glucose which is the main source of energy for cells. It is caused when there is poor regulation of insulin which is a hormone produced by the pancreas to allow glucose to enter the cells. Diabetes is linked to dysfunctions of the neurovascular coupling which is the connection between neurons and their energy source and to-date there is no definite treatments to these negative effects [Chhabria et al., 2019]. In the study of [Chhabria et al., 2018] it is reported that the nitric oxide (NO) donor sodium nitroprusside (SNP) prevents the negative effects of glucose on neurovascular coupling in zebrafish and the followed study of [Chhabria et al., 2019] investigates the wider effects of glucose in the neurovascular coupling and behaviour of zebrafish and how SNP treatment prevents them. This dissertation will be focused and present the behavioural analysis and results of the aforementioned study.

4.2.2 Methods

4.2.2.1 Locomotion analysis

The full swimming paths of the animals where segmented based on entrances/exists to/from the light/dark areas of the arena. Small path segments with lengths less than the first percentile of the segments lengths generated as an artefact from light/dark transitions were discarded from further analysis.

For each segment, a number of features (refer to Figure 4.2) inspired from the Morris Water Maze with rodents [Gehring et al., 2015; Vouros et al., 2018] that capture geometrical and positional aspects of the segments, were computed. In addition to these features the experiment specific feature of the number of light/dark transitions is also considered.

The analysis was performed for both the segments and the whole swimming paths, in blocks of 15 minutes in order to obtain a more detailed insight of drug-induced changes.

A prior analysis of some extra attributes of the animals swimming paths is included. These attributes are, the percentage of time spent in the light and the percentage of time spent in low and high speed locomotion in light and dark region of the arena. Speed thresholds were set as high $\text{speed} > 6.4mm/s$, low $\text{speed} \in [3.3\ \ 6.3]\ \ mm/s$ and inactive $< 3.3\ \ mm/s$.

4.2.2.2 Light/dark task and data properties

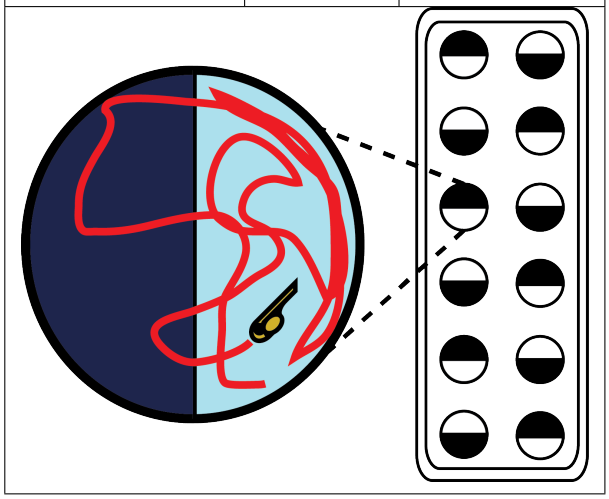
Analysis of light/dark preference of the larval zebrafish was designed on a similar principle to that of the adult zebrafish light/dark preference test [Blaser and Penalosa, 2011]. A 12-well plate was modified by adhering three cellophane films (blue, green and yellow) to half of each well to create a dark side that allowed the camera to track larvae movement by infrared (IR) (refer to Figure 2.3 (d) for a sample visualisation

of this setup). Larvae from different treatment groups were placed on the light side of the well. The plate was placed inside a Viewpoint Zebrabox system.

The data set was consisted of four different groups of zebrafish larvae, two of the groups were consisted of zebrafish larvae incubated at embryonic period in mannitol while the others in glucose. One mannitol and one glucose group was also inducted in SNP co-treatment during their incubation (refer to Table 4.1 for details about the number of animals per group). The mannitol group without treatment was considered the control group. The zebrafish larvae were left in the wells for 1 hour in total. For more detailed specifications refer to [Chhabria et al., 2019].

Table 4.1: Visualization of the experimental procedure and animal counting per group for the light/dark preference task.

	mannitol	glucose
no treatment	50	44
SNP treatment	45	56



4.2.2.3 Statistical analysis

Statistical comparisons were executed on GraphPad Prism [Jolla, 2016]. All inter-group comparisons were performed using two-way ANOVA with post-hoc multiple-comparison tests (Sidak's test [Šidák, 1967]), where appropriate. P-values < 0.05 were considered to be statistically significant; degrees of significant are as follows: ns or no stars $P \geq 0.05$, $*P < 0.05$, $**P < 0.01$, $***P < 0.001$, $****P < 0.0001$. Data are shown as mean ± standard error of mean (s.e.m.).

4.2.3 Results and discussion

Based on common locomotion analysis of the zebrafish larvae swimming paths the two mannitol exposed larvae groups showed a significant preference for the light as opposed with the untreated glucose exposed group (metric: time spent in light area). The latter group also showed significant increased time in both low and high speed locomotion while in light compared with the two mannitol exposed groups (metric: low/high speed locomotion in light area). There was no significant locomotion differences in the dark area. These behavioural differences were prevented with co-treatment with SNP. For relevant results refer to [Chhabria et al., 2019].

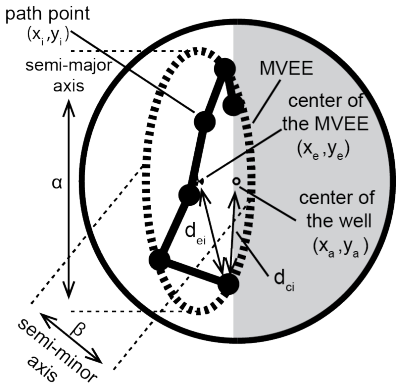
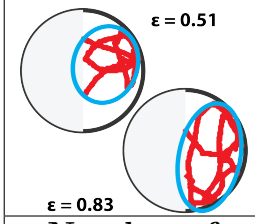
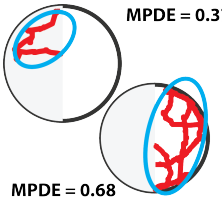
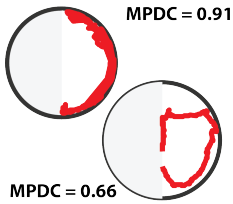
Based on the features (refer to Table 4.2) analysis, there was a significant increase in the number of transitions between the untreated glucose exposed group and the two mannitol exposed larvae groups and this increase was prevented by co-treatment with SNP (refer to Figure 4.3 (A)). In addition, the untreated glucose exposed group was showing a significant increase in both exploratory behaviours (eccentricity and MPDE) and thigmotaxis (MPDC) in the light side of the well as opposed to the mannitol exposed larvae groups (refer to Figure 4.4). Again these differences were prevented with co-treatment with SNP.

The same analysis was performed in blocks of 15 minutes. Based on the results, the glucose exposed group showed an increased number of transitions on the 30 minutes time stamp (refer to Figure 4.3 (B)) but no statistical significant different could be established between this group and the other three groups. For the rest of the features (refer to Figure 4.5) there was a more evenly increase on path eccentricity, MPDE and MPDC for the glucose-exposed zebrafish indicating significant difference in exploration (light area: eccentricity 15 minutes and 30 minutes timestamps, MPDE 30 minutes timestamp; dark area: eccentricity 30 minutes and 45 minutes timestamps, MPDE 45 minutes timestamp) and thigmotactic behaviour (light area: MPDC 30 minutes and 45 minutes timestamps; dark area: MPDC 30 minutes timestamp) compared with the other three groups. Again, SNP prevented the effect of glucose on these aspects of behaviour.

Previous studies have described larval zebrafish behavioural differences with anxiolytic or anxiogenic treatments [Egan et al., 2009; Richendrfer et al., 2012] as well as increased exploration and thigmotaxis [Blaser et al., 2010; Egan et al., 2009]. This is the first study that characterise the effect of hyperglycemia on geometrical and positional aspects of zebrafish locomotion. Glucose exposure resulted in an increase in both exploration (as measured by the geometric features, eccentricity and MPDE) and thigmotaxis (as measured by an increase in the positional feature, MPDC). The latter point may indicate an association of glucose exposure and diabetes to anxiety-related brain activation but this requires further investigation.

Finally, this study shows the importance of path features in the quantification of various behavioural motifs that are observed throughout experimental procedures. Here, the feature selection was performed manually by careful consideration and experimentation of various geometrical and positional aspects of the paths. Later (refer to Chapter 6) a new behavioural analysis framework based on semi-supervised sparse clustering will be proposed for automatic feature selection.

Table 4.2: Manual feature selection for the light/dark experimental procedure with zebrafish larvae (source: [Chhabria et al., 2019]).

<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td>Eccentricity (ϵ)
$\epsilon = \sqrt{1 - \frac{\beta^2}{\alpha 2}}$</td><td>Mean point distance from ellipsoid (MPDE)
$MPDE = \frac{\frac{\sum\sqrt{(x_i - x_e)^2 + (y_i - y_e)^2}}{n}}{R}$</td><td>Mean point distance from center (MPDC)
$MPDC = \frac{\frac{\sum\sqrt{(x_i - x_\alpha)^2 + (y_i - y_\alpha)^2}}{n}}{R}$</td></tr>
<tr><td>A geometric feature measuring the absolute elongation of a path [Gehring et al., 2015]. More elongated paths have higher eccentricity indicating more exploration. For a complete circular path, $\epsilon = 0$. α and β are the semi-major and semi-minor axes of the minimum enclosing ellipsoid indicated by a blue circle.</td><td>A geometric feature measuring the average elongation of a path. It is defined as the average Euclidean distance between every point of the path (x_i,y_i) and center of the minimum enclosing ellipsoid (x_e,y_e), normalized over the well radius, R. Larger values of MPDE indicate paths that are more scattered across the given area of the well.</td><td>A positional feature measuring the position of the path in relation to the center of the well. It is defined as the average Euclidean distance between every point of the path (x_i,y_i) and center of the well (x_α,y_α), normalized over the well radius R. Paths closer to the walls of the well (thigmotaxis) have a higher value of MPDC.</td></tr>
<tr><td> 
</td><td> 
</td><td> 
</td></tr>
<tr><td>Number of light/dark transitions</td><td>Number of times the zebrafish larvae transitioned between light and dark regions of the well.</td><td></td></tr>
</table>

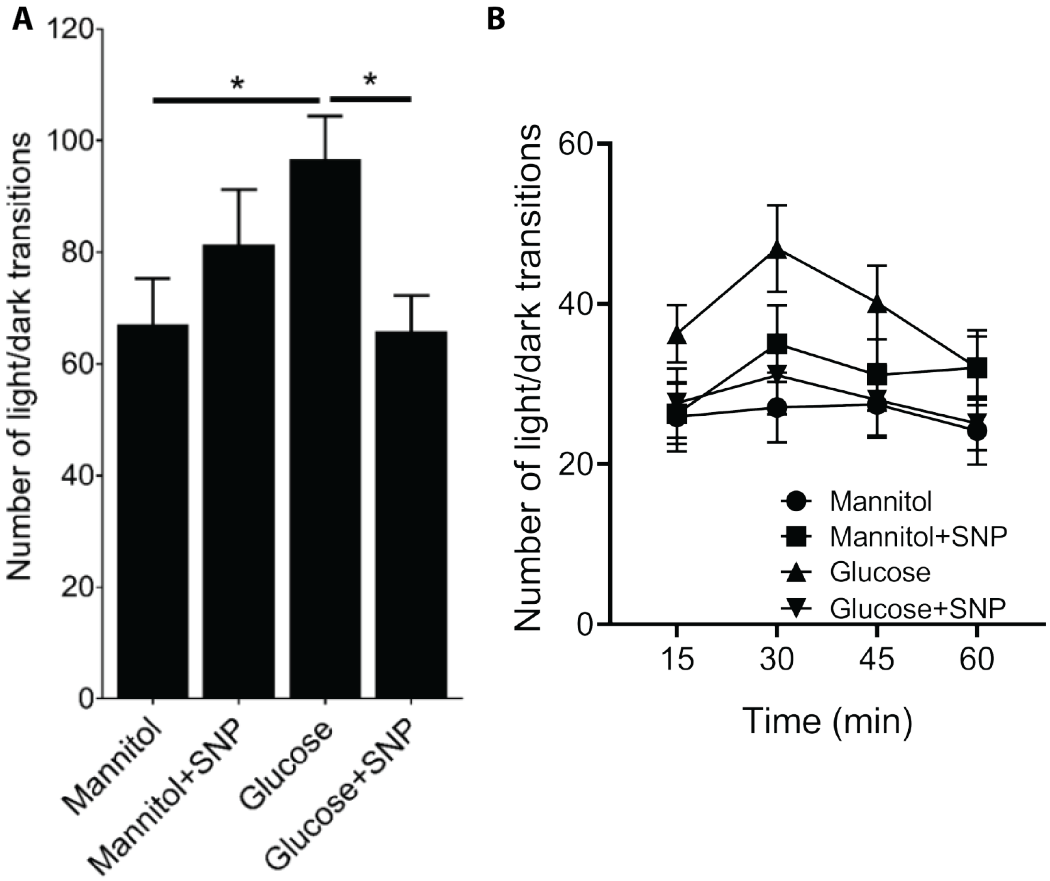


Figure 4.3: Effect of mannitol/glucose treatment with/without sodium nitroprusside (SNP) on larval zebrafish light/dart behaviour (source: [Chhabria et al., 2019]). Quantification of number of transitions into the light/dark regions for the zebrafish larvae. **(A)** Overall number of light/dark transitions for each zebrafish larvae group. **(B)** Number of light/dark transitions within 15 minutes time intervals.

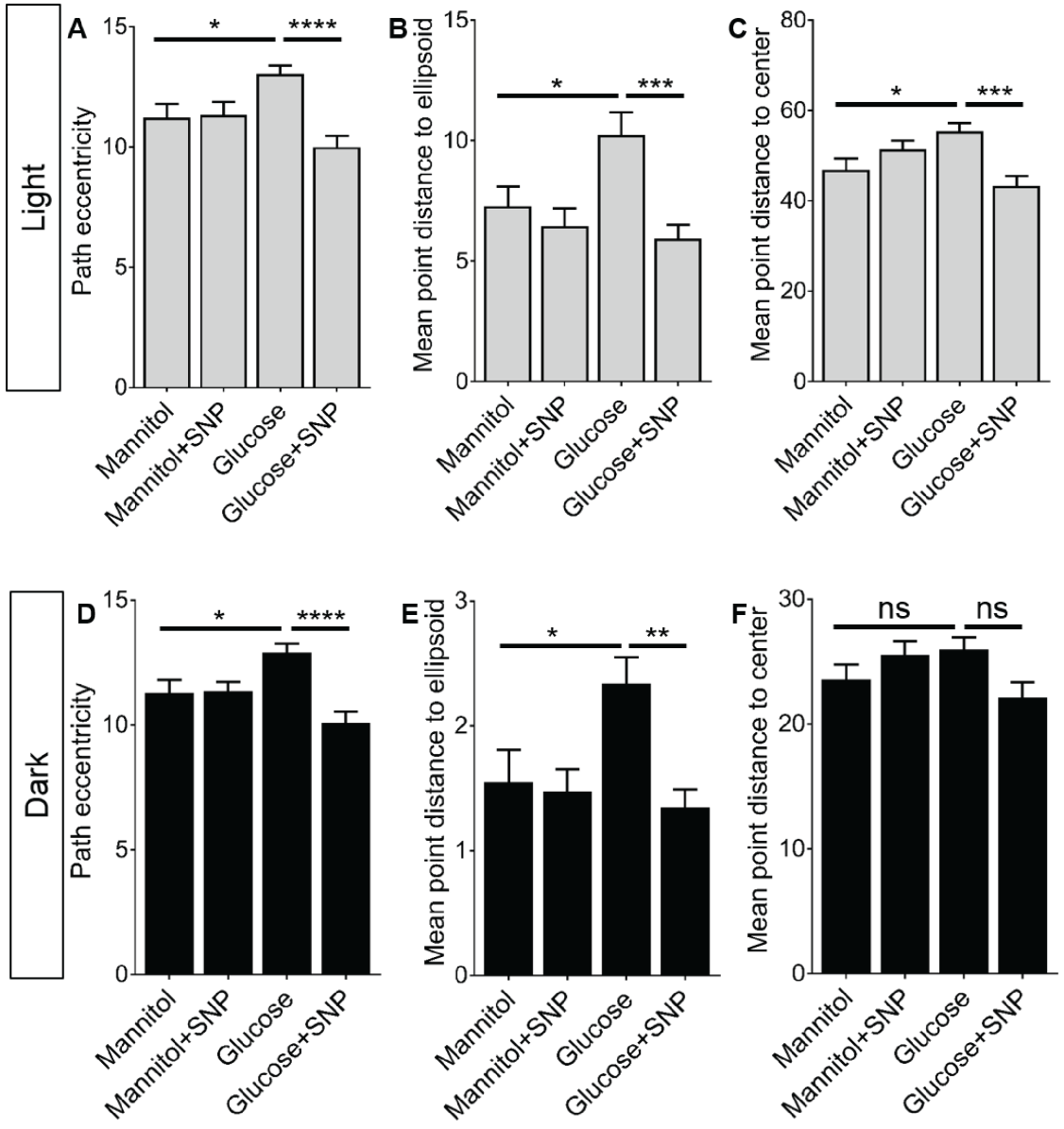


Figure 4.4: Effect of mannitol/glucose treatment with/without sodium nitroprusside (SNP) on various features of zebrafish locomotion (source: [Chhabria et al., 2019]). Quantification of mean frequency of eccentricity (A, D), mean point distance to ellipsoid (MPDE; B, E) and mean point distance to centre (MPDC; C, F) for the swimming path segments of the zebrafish larvae in the light (A-C) and dark (D-F) regions of the well.

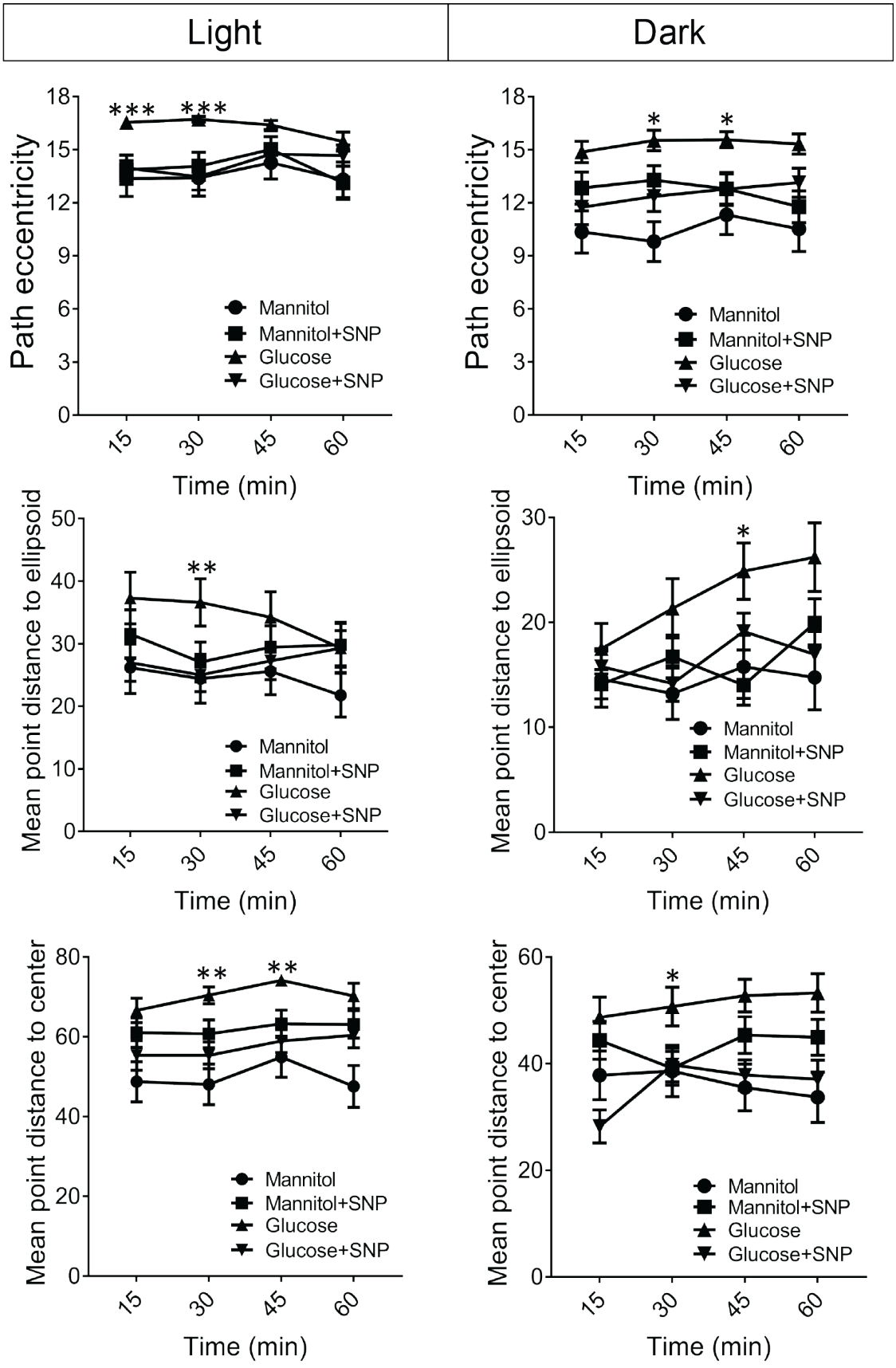


Figure 4.5: Effect of mannitol/glucose with/without within 15 minute time intervals on various features of larval zebrafish behaviors; eccentricity, MPDE and MPDC in corresponding light and dark sides of the well (source: [Chhabria et al., 2019]).

Chapter 5

A generalized framework for detailed classification of swimming paths inside the Morris Water Maze

This chapter describes the published work of Vouros et al. [Vouros et al., 2018] and its further application in the work of Huzard et al. [Huzard et al., 2019]. This study generalises the method of Gehring et al. [Gehring et al., 2015] and proposes a framework and software for detailed behavioural analysis inside the Morris Water Maze procedure (MWM). For more information about this experimental procedure refer to Chapter 2. This framework uses the concept of semi-supervised learning with the MPCK-Means algorithm described in Chapter 2.

5.1 Introduction

The Morris Water Maze is commonly used in behavioural neuroscience for the study of spatial learning with rodents. Over the years, various methods of analysing rodent data collected during this task have been proposed. These methods span from classical performance measurements to more sophisticated categorisation techniques which classify the animal swimming path into behavioural classes known as exploration strategies. Classification techniques provide additional insight into the different types of animal behaviours but still only a limited number of studies utilise them. This is primarily because they depend highly on machine learning knowledge. In the work of [Gehring et al., 2015] there has been a demonstration that the animals implement various strategies and that classifying entire trajectories can lead to the loss of important information.

In this work, an automatic boosted classification procedure based on majority voting, which improves on the classification error, and a validation framework which leads to conclusions with a high degree of confidence are presented. Majority voting refers to the fact that more than one classifier are used in order to assign a swimming path segment into a class. This framework has been implemented into a fully working software capable of performing all of the analyses, without requiring machine learning knowledge from the user. This software is called RODA (ROdent Data Analytics) [Vouros et al., 2017] and is focused on the MWM experiment. It provides

an easy to use graphical user interface (GUI) for loading the data and defining the experimental specifications. It also supports automatic segmentation and semi-automatic classification, and produces quality figures which can be exported into various image formats. RODA's framework is applied to two MWM experimental procedures ([Huzard et al., 2019; Vouros et al., 2018]) focused on the effects of stress in learning and memory.

5.2 Methods

5.2.1 Analysis overview

In the proposed analysis method, the swimming paths of the animals inside the Morris Water Maze are divided into segments of approximately equal length and a fixed overlap percentage. For each segment a set of eight features is computed (refer to Section 5.2.2) and then used in the classification procedure. Finally, a small portion of the segments needs also to be assigned manually to a specific strategy (labelling); this information is used as prior knowledge to guide the classification procedure.

The classification procedure, which assigns segments to classes of behaviour, is based on a semi-supervised clustering algorithm called Metric Pairwise Constrained K-Means (MPCK-Means) [Bilenko et al., 2004]. This algorithm incorporates the two main approaches of semi-supervised clustering: metric learning (the measuring of similarity, 'distance', between data) and constrained-based learning (the use of labels or constraints that produce a better grouping of the data). To turn the algorithm into a classifier, the labelled data were used not only to guide the clustering process but also to assign clusters to classes based on the labbelled elements of each cluster (refer to Section 5.2.3).

A common issue with many clustering algorithms, including MPCK-Means, is that a predefined number of target clusters needs to be provided; this number indicates the number of clusters into which the data will be partitioned. Determining the optimal number of target clusters is challenging and, although many different quality measures were proposed over time [Kovács et al., 2005], this value will depend on the specific clustering method and data at hand.

In this work, instead of searching for an optimal number of clusters and attempting to generate an optimal classifier, a pool of 'strong' classifiers whose 'goodness' is assessed based on the 10-fold cross validation error is generated. The strong classifiers are then used to form an 'ensemble' which uses majority voting to reach a classification decision. The two conditions of having both strong and diverse classifiers are essential in majority voting in order to reach an optimal classification solution [Sharkey and Sharkey, 1997; Zhou et al., 2002]. This will be discussed in more detail later. In order to assess the labelling procedure (if enough and consistent labels have been provided) the criterion of having a minimum of 40 strong classifiers has been added prior to majority voting. Finally, the classification result of the ensemble is expected to have a low percentage of unclassified segments (less than 7%) because, since the classifiers are diverse, they will have different errors or will fail to classify different segments. Thus if they work together and form an ensemble, the individual errors will be compensated by the correct responses of the other members of the ensemble [Sharkey and Sharkey, 1997]. A diagram of the procedure is illustrated in Figure 5.1.

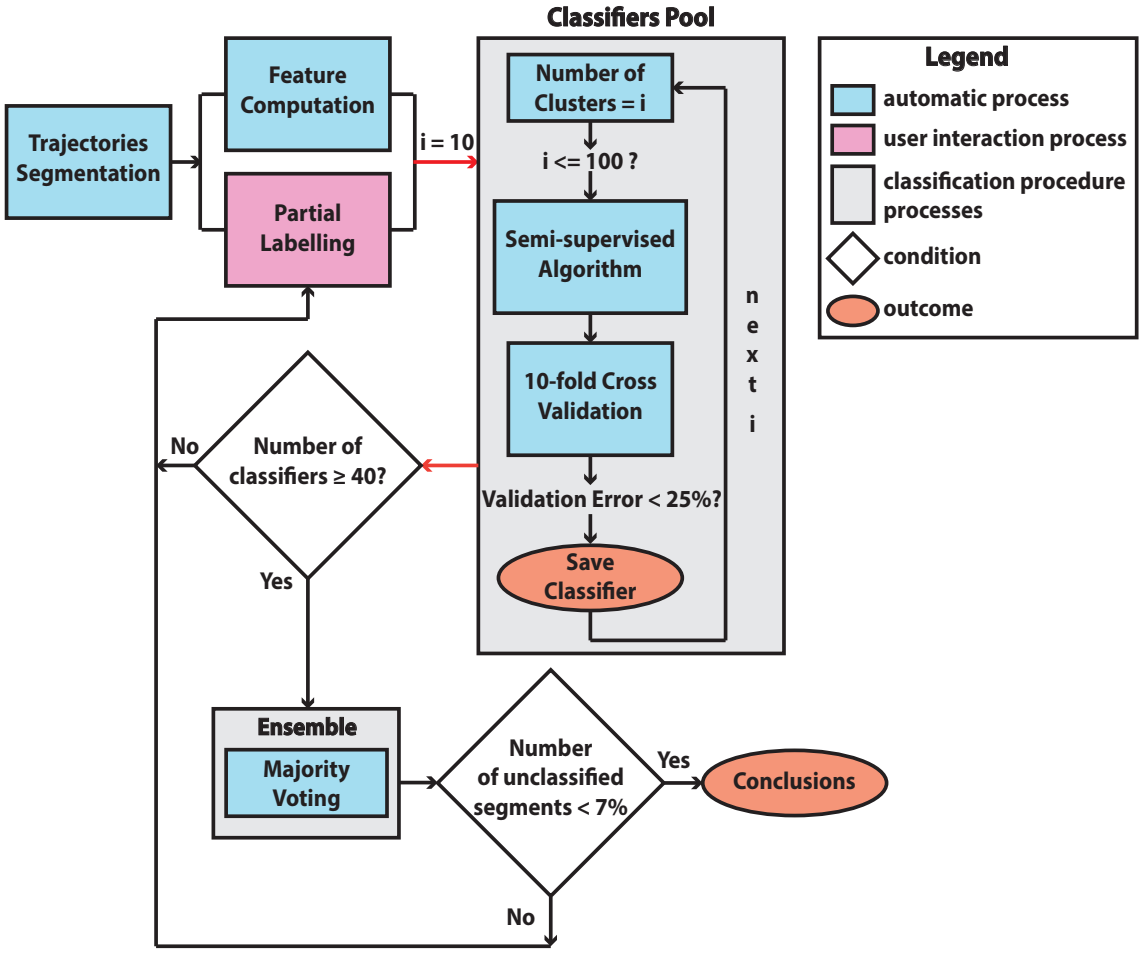


Figure 5.1: Workflow diagram illustrating the analysis procedure. After the trajectory segmentation, eight trajectory features for each segment are computed and a certain number of segments are manually labelled. Afterwards, a pool of classifiers is generated using different number of initial clusters ranging for 10 to 100. 'Strong' classifiers (cross validation error $< 25\%$) are then selected from the pool and work together (majority voting) as a team (ensemble) to produce the classification results. It is expected that the ensemble is large enough (number of classifiers ≥ 40). Throughout the process the labelling quality is constantly assessed and in case of weak classification results (small ensemble or 7% or more of the segments remained unclassified) the user is directed back to the labelling stage.

5.2.2 Trajectories segmentation, features computation and partial labelling

To assign one trajectory to multiple classes, the earlier work of [Gehring et al., 2015] proposed the division of the full animal swimming paths into segments. Based on this method, each segment overlaps significantly with its previous one (percentages of 70% and 90% have been performed on this analysis) to make sure that important information is not lost due to an unfavourable segmentation. In this study focus was given on the segment length under which there is consistency in the analysis conclusions thus multiple segmentations with different tunings where performed (refer to Table 5.1). If the segment length is too short it might be difficult to identify to which class segments belong; if it's too long it might happen that more than one class of behaviour is represented. The latter case can be seen in the results section, where the large segment length (3 times the arena radius) causes some classes to be overshadowed by the more common classes (refer to Figure 5.9).

For each segment a set of 8 features (refer to Table 5.2) was computed. These features were adopted from the earlier study of [Gehring et al., 2015] and are able to capture geometrical and positional aspects of the segments. The features, along with manually labelled data that will be described next, were used during the classification procedure.

In this study, nine predefined strategies were adopted (see section 5.2.9). Empirically is was found that the amount of data that needs to be labelled should be roughly between 8% to 12% of the total segment number but the exact value depends greatly on the dataset under investigation. As a rule of thumb, if fewer labels are provided then the classification results will be poor in the sense that a lot of segments will remain unclassified or fall under the wrong class. Since the labelling procedure is prone to error and subjectivity a number of validation criteria have been implemented throughout the proposed analysis.

	Segmentation I	Segmentation II	Segmentation III	Segmentation IV
Segment Length (cm)	300 $(3 \cdot R)$	250 $(2.5 \cdot R)$	250 $(2.5 \cdot R)$	200 $(2 \cdot R)$
Segment Overlap	70%	70%	90%	70%
Number of Segments	8847	10388	29476	13283
Number of Segments Labelled	988 (12%)	1261 (12%)	2445 (8%)	1227 (9%)
Total Number of Labels	1022	1313	2568	1232

Table 5.1: Parameters for the classification of four different segmentation configurations with variable segment lengths and overlaps. For each segmentation a percentage of segments (between 8% and 12%) was manually labeled (Number of Segments Labelled). Multiple labels could be given to each segment; in this study no more than two labels were given simultaneously to a segment (Total Number of Labels). The segment length was selected to be proportional to the arena radius (R), which was equal to 100cm. The segment overlap was used to avoid any unfavourable segmentation (see Methods)

5.2.3 Semi-supervised classification

The classification procedure is the one described in the work of [Gehring et al., 2015] which is based on the Metric Pairwise Constrained K-Means (MPCK-Means) clustering algorithm implemented by Bilenko et al. [Bilenko et al., 2004] (for more information about the algorithms refer to 2.2.4.2).

With the use of labelled data it is possible not only to guide the clustering procedure (with the creation of MUST-LINK and CANNOT-LINK constraints as described in 2.2.4.2) but to also combine clusters together and form larger groups (classes) which are actually the categories of the labelled data. This mapping of clusters into classes is done by converting a cluster into a class based on the number of labelled segments within the cluster and its size (based on the number of points within the cluster) as it is shown in the equation below:

$$m_i \equiv \lceil n_i * p_{min,i} \rceil, \text{ where }, p_{min,i} \equiv max(n_i^{-\gamma}, p_{min}) \text{ and} \tag{5.1}$$

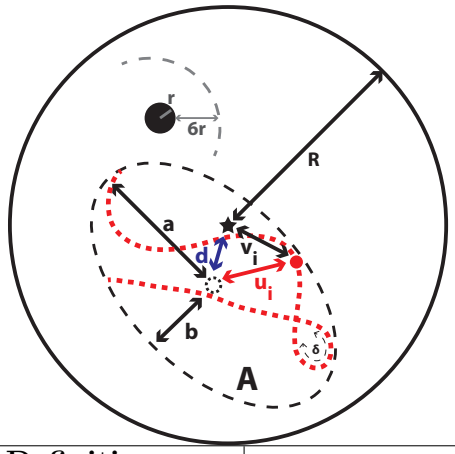
$m_i \equiv$ minimum amount of labels $, n_i \equiv$ cluster size, $\gamma = 0.75$, $p_{min} = 0.01$ (or 1%)

Based on equation 5.1 smaller clusters require more labelled data in order to be assigned to a class while larger clusters require less labelled data. p_{min} acts as a threshold in the sense that is must always be 1% of labelled data available within the cluster in order to be assigned to a class disregarding its size. For example if a cluster has 100 points then $p_{min} = 0.03$ and the minimum number of labelled data needs to be $m_i = 3$. In case that the criterion described by equation 5.1 fails or if multiple labels of different classes are present within the cluster then the cluster is marked as undefined.

Regarding the constraints, a MUST-LINK constraint is generated between two datapoints with the same label and a CANNOT-LINK constraint is generated between two datapoints with a different label. Multilabelled datapoints are considered distinctive meaning that if for example a datapoint is labelled as *thigmotaxis* and *incursion* then a MUST-LINK constraint will be generated only with datapoints that are also labelled as *thigmotaxis* and *incursion*. In addition, a constraint is created only between relatively close datapoints, i.e. if the Euclidian distance between the two labelled datapoints is less that 0.25 (the features of the datapoints are normalized between [0 1]). The last rule has been implemented in the previous work of [Gehring et al., 2015] to limit the number of generated constraints since too many constraints can create computational issues.

In order to improve the classification quality the 'two-stage clustering' of [Gehring et al., 2015] was performed. In the first stage the data were clustered using only the CANNOT-LINK constraints and then clusters that could not be mapped to a class (ambiguous clusters) are sub-divided by another clustering step, this time, however, both CANNOT-LINK and MUST-LINK constraints are used [Gehring et al., 2015]. Moreover multiple target number of clusters are tried in succession from 2 up to two times the initial number of clusters used in the first clustering. A sub-portioning is considered correct if one of the sub-clusters could be classified. The stages of this process are shown in Figure 5.2.

Table 5.2: List of features used during the classification procedure. For the figure: **Outer (solid) circle:** the Morris Water maze arena with radius $\boldsymbol{R}$ and center at **star**; **black (filled) circle:** the hidden platform with radius $\boldsymbol{r}$; **red dots:** sample of points forming a trajectory segment; **dashed ellipse:** the minimum enclosing ellipse (MEE) to the trajectory segment with center at the **dotted circle**, the MME is defined as the smallest enclosing ellipsoid of a set of points [Gärtner and Schönherr, 1997]; $\boldsymbol{A}$, $\boldsymbol{a}$ and $\boldsymbol{b}$: area of the MEE, the semi-major and semi-minor axes of the MEE; $\boldsymbol{d}$**:** distance between the center of the arena and the center of the MEE; $\boldsymbol{u_i}$**:** distance between the center of the MEE to the segment point $\boldsymbol{i}$, $\boldsymbol{U}$ is the set of all these distances. $\boldsymbol{v_i}$**:** distance between the center of the arena to the segment point $\boldsymbol{i}$, $\boldsymbol{V}$ is the set of all these distances.



Feature name	Definition	Purpose
Eccentricity	$\epsilon = \sqrt{1 - \frac{b^2}{a^2}}$	Measure the elongation of the segments.
Focus	$f = 1 - \frac{4A}{\pi l^2}$	Measure how much the animal is searching a specific area of the arena.
Inner radius variation	$IRV = \frac{IQR(U)}{median(U)}$	Measures the relative dispersion of points related to a circle. The specific formula was used to increase the robustness and stability against outliers.
Maximum loop length	$MLL = \frac{\delta}{l}$, where δ is the length of the longest self-intersecting loop and l is the total length of the segment. If no intersection is present, the value 0 is assigned.	It is used mainly as a measurement of the self-oriented movements
Central displacement	$CD = \frac{d}{R}$	It is used to identify concentric paths with the arena.
Median distance to center	$DCm = \frac{median(V)}{R}$	It is used to identify the amount of time that the animal spends next to the walls of the arena.
IQR distance to center	$DCiqr = \frac{IQR(V)}{R}$	It is used to identify the spread of the time during which the animal is moving next to the walls of the arena.
Target proximity	Percentage of the path lying within an area centered at the platform and $6 \cdot r$.	It is used to identify if the animal spends time actively searching for the platform or to capture random crosses through or close to the platform.

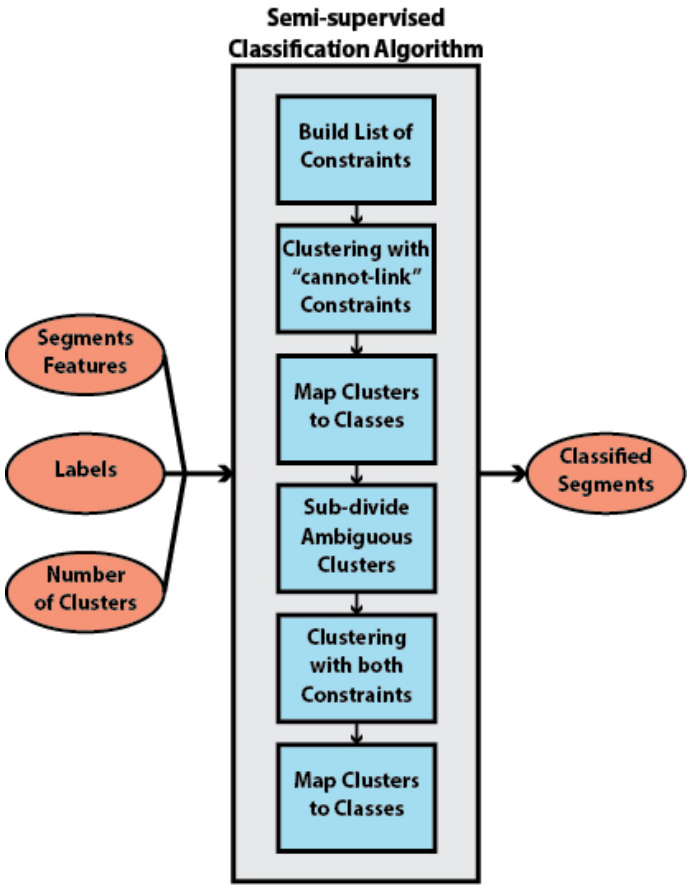


Figure 5.2: Stages of the semi-supervised classification algorithm [Gehring et al., 2015]. As inputs the computed features of the segments along with a partial set of labels of the segments are provided. In addition, a predefined number of target clusters needs to be provided, which specifies the number of clusters that the algorithm needs to detect. As output the algorithm provides the class in which each segment falls into. The labels are used to formed the list of constraints of which data should not be (CANNOT-LINK) or should be (MUST-LINK) in the same clusters. In the first-stage clustering only the CANNOT-LINK constrains are used to guide the clustering procedure and then clusters that could not be mapped into classes (ambiguous clusters) are sub-divided and a second clustering stage begins this time with both CANNOT-LINK and MUST-LINK constraints.

5.2.4 Classification boosting with majority voting

The classification boosting is an ensemble technique that is based on the idea that many weak learners can be converted to a strong learner [Kearns and Valiant, 1993]. In machine learning terms an ensemble of weak classifiers (classifiers that make mistakes) can be used to form a strong classifier (classifier that makes fewer mistakes) by combining each individual's opinion [Gerecke et al., 2003; Jurek et al., 2011]. This approach has been used in various classification tasks (see Oza et al. [Oza and Tumer, 2008] for a survey) and in addressing complex real-world problems, when single algorithmic classification solutions are unable to achieve high performance [Acharya et al., 2011].

One way to perform classification boosting is through majority voting [Gerecke et al., 2003]: many classifiers form an ensemble, vote for the class of each datapoint and the class with the most votes wins. The output of the ensemble is expected to have improved accuracy since individual errors of each classifier are compensated by

the correct responses of the other members of the ensemble [Sharkey and Sharkey, 1997]. In order to achieve such an outcome, the classifiers need to at least be diverse in the sense that they should not share the same errors [Gerecke et al., 2003; Schapire, 1990]. It should be noted, however, that diversity alone is insufficient to ensure that randomly selected, arbitrarily weak, classifiers will achieve high classification accuracy [Sharkey and Sharkey, 1997; Zhu, 2015]. Individual classifiers have to also be strong meaning that they should be sufficiently accurate on their own ([Ruta and Gabrys, 2002; Zhu, 2015] indicate an accuracy of at least 50%).

5.2.4.1 Majority voting implementation

In the proposed framework, the need is to classify different trajectory segments into animal behavioural classes (strategies) having only a partial set of labelled data. The classification is parameterised by the target number of clusters of the clustering algorithm, a value that is difficult to estimate in advance. In order to overcome this problem a number of classifiers were generated by providing different numbers of target clusters in succession; At the end of this process a pool of classifiers is generated. 10-fold cross validation [Varma and Simon, 2006] process is then used, as proposed in [Bilenko et al., 2004; Gehring et al., 2015], to evaluate different numbers of target clusters from 10 to 100 (for more information about the cross validation procedure refer to 2.2.8.3). More number of clusters can be tested but this will result on a significant rise of the classification time requirements and, empirically, it can be avoided with better labeling. Only classifiers with a validation error lower than 25% are used to form an ensemble. The minimum number of required classifiers that fulfill this criteria was set to 40. The reasoning behind this requirement is to ensure that there is a satisfactory sample size of 'strong' classifiers since this will be used later to provide a degree of confidence to the quality of the results (refer to 5.2.7). For the majority voting, a simple scheme was adopted where the vote of each classifier has the same weight [Bouziane et al., 2011; Liaw and Wiener, 2002] and that in case of a tie the data point (segment) is marked as undefined.

5.2.5 Framework validation

The new framework was validated thoroughly by assessing all the procedures in terms of robustness and results consistency. Overall, four different segmentations were performed with the aim to find the bounds (error margins) for the segment length between which there are consistent analysis conclusions. As it is discussed in the results section, it is expected that the segmentation length affects the results and it is shown that consistency for the MWM can be achieved with segmentation lengths between 2 times and 2.5 times the arena radius. For more information refer to Figure 5.9 where it is shown that longer segment lengths fails to capture the difference between the two groups of the Chaining Response behavioural strategy.

For each of the four different segmentations the performance of the classifiers, the ensemble and multiple ensembles formed by random sample of 'strong' classifiers are compared. Table 5.3 shows the relevant results of the last stage of analysis, where the overlapped segments have been mapped back to the original swimming paths. For the latter the smoothing function is applied on the segments (refer to section 5.2.6) and this detail is important because the smoothing procedure increases the performance of the classifiers (for the statistical analysis prior to the smoothing

function refer to the Appendix C). As expected, ensembles have higher accuracy, a lower percentage of unclassified segments and a higher percentage of agreement among them in comparison to individual classifiers. However, since cross validation was used for both tuning and testing, additionally the error of the ensembles was manually assessed on two out of the four segmentations (see Table 5.4 for the manual error estimation).

	Segmentation I	Segmentation II	Segmentation III	Segmentation IV
Number of generated Classifiers	42	78	91	64
	Performance: Classifiers			
Average Error (%) [min-max]	16.8 [5.4 24.9]	17.5 [3.7 25.0]	13.9 [1.8 21.5]	18.0 [7.3 24.9]
Unclassified (%) Segments	2.5	2.5	1.3	3.7
Agreement (%)	58.7	61.0	59.6	56.3
	Performance: Ensemble(s)			
Error (%)	< 0.01	0.2	< 0.01	< 0.01
Unclassified (%) Segments	0.0	0.0	0.0	0.1
Agreement (%)	83.4	82.6	82.3	80.0

Table 5.3: Classification statistics (average) for the four segmentation configurations of Table 5.1 and benefits of majority voting. (1) Number of generated classifiers: based on each segmentation, only classifiers with cross-validation error lower than 25% were selected to take part in the classification analysis procedures (ensemble and binomial confidence intervals). As a rule of the thumb a minimum number of 40 'strong' classifiers is required to be generated in order to trust the classification results. **(2) Error:** the 10-fold cross validation was used in order to select 'strong' classifiers based on their validation error. 10-fold cross validation was also used to compute the average accuracy of the 'strong' classifiers and the accuracy of the ensemble (in case of the ensemble, the same folds used by the classifiers were re-used). The ensemble significantly benefits the classification accuracy. Because the cross validation was used for both tuning and testing the error of the ensembles was manually assessed on two out of the four segmentations (see Appendix C). **(3) The percentage of unclassified segments** was computed separately (again based on cross-validation); since the classifiers are 'strong' only a few segments remain unclassified, nevertheless the ensemble almost totally nullifies the unclassified segments. **(4) The average agreement** between the classifiers was computed by first calculating the percentage of agreement within each pair (agreement is formed when two classifiers have assigned the same label on a particular segment) and then averaging all the agreements together (refer to Validity Measurements for more information). In order to perform the same statistical measurement in the ensemble domain, 21 ensembles were created by picking a random sample of 11 'strong' classifiers from the pool. The agreement between the classifiers is better than moderate and, as expected, the agreement of the ensembles is high. A sample smaller than 40 was chosen to avoid a large overlapping of classifiers across ensembles.

5.2.5.1 Classifier diversity

To evaluate the diversity of the classifiers, the percentage of their agreement is assessed for the class of each segment. The result is a symmetric matrix with rows and columns representing the classifiers where each element shows the percentage of segments for which two classifiers agree on the assigned class. The diagonal values of this matrix equal to 100 as each classifier is in 100% agreement with itself (refer to

	Segmentation II	Segmentation IV
TT error	1.4%	1.0%
IC error	8.0%	1.5%
SC error	6.6%	4.1%
FS error	6.0%	3.7%
CR error	11.5%	12.6%
SO error	11.0%	8.1%
SS error	9.6%	3.2%
ST error	2.3%	1.0%
average error	5.7%	4.4%
total error	6.3%	2.8%

Table 5.4: Manual estimation of ensemble error. The ensemble error was manually assessed for the segmentations II and IV. The table shows both the total error and the error among the different classes (including the average). The total manually estimated error of the ensembles is still significantly lower than the average error of the classifiers (6.3% vs 17.5% for Segmentation II and 2.8% vs 18.0% for Segmentation IV). The results of the ensembles were manually assessed for two reasons: (i) to estimate the overfitting, which is likely to be caused because the same data were used for both tuning and testing, and (ii) because the testing set was very small since a limited amount of labels were provided, the error estimation is likely to be overly optimistic.

the Appendix C for an example of an agreement matrix). An overall agreement can be computed by averaging the upper or lower triangular of the matrix. In addition, the average cross validation error (accuracy) over the classifiers is considered. In order for the classifiers to be both diverse and strong it is expected that they should have an average percentage of agreement well below 100% (in this case around 60%) and low cross validation error (refer to Table 5.3).

As previously reported [Sharkey and Sharkey, 1997], ensembles have far less variance in comparison with individual classifiers thus it is expected to have much higher agreement. To demonstrate this observation, a number of ensembles is generated by picking classifiers at random from the pool. Afterwards the same statistical measurement of agreement is performed for the ensembles, similar to the one described for the classifiers. In contrast to the classifiers, the ensembles have high agreements among them (more than 80%) and nearly nullify the cross validation error of the classifiers (see Table 5.3). However, since in this particular occasion the cross validation was used for both tuning and testing [Gehring et al., 2015], additionally manually assessment of the error was performed on the ensembles in two out of the four segmentations (see Table 5.4 for the manual error estimation).

5.2.5.2 Percentage of unclassified segments

A useful measure for the quality of the classification is the percentage of unclassified segments. For certain segments, it is expected that none of the classifiers in the ensemble will be able to determine a class, or that there could be a draw for segments that transit between classes (refer to Table 5.5). This, however, does not have an impact on the consistency of results.

	Segmentation I	Segmentation II	Segmentation III	Segmentation IV
Thigmotaxis	27.7%	24.0%	24.6%	22.5%
Incursion	19.0%	18.9%	20.6%	17.0%
Scanning	10.2%	12.3%	10.5%	11.9%
Focused Search	9.2%	8.9%	8.2%	10.0%
Chaining Response	4.5%	5.8%	5.5%	9.8%
Self Orienting	7.1%	8.8%	8.2%	8.4%
Scanning Surroundings	17.4%	15.8%	16.8%	12.9%
Target Scanning	4.9%	5.6%	5.6%	7.4%
Unclassified	0.0%	0.0%	0.0%	0.1%

Table 5.5: Percentage of segments falling under each class for the four segmentation configurations of Table 5.1. Some differences among the four segmentations are visible although based on the results of Figure 5.9 consistency on the conclusions is preserved in segmentations II, III and IV. Regarding segmentation I, where there is no indication of any difference between the two animal groups based on the Chaining Response strategy, more segments are identified as Thigmotaxis and Scanning Surroundings. This indicates the possibility that some segments which transit between Chaining Response and one of these strategies are classified either as Thigmotaxis or Scanning Surroundings.

5.2.6 Mapping segment classes to the full swimming paths

The classification has been performed on overlapping segments of the animals' swimming paths, thus they need to be mapped back to the whole trajectories.

As a first approach, the classified segments are considered as continuous parts of the trajectories ignoring the overlap percentage. This method provides consistent results on the significant differences of the strategies but fails to detect differences on strategy transition between groups (refer to the Appendix C for the relevant result). The reason for this is that sparse segments within each swimming path fall under different classes thus viewing them as a sequence leads to an overestimation of transitions (a transition occurs when a segment falls under a different class after a sequence of segments that fall under the same class).

To address this limitation, a smoothing technique is implemented with parameters independent of the segmentation choice. This was done for two reasons: (i) to avoid subjective conclusions based on a specific segmentation configuration and (ii) to be able to directly compare different segmentations. In more detail, given that R equals to the radius of the arena, the swimming paths are now divided into intervals of length R. Each of the intervals is assigned to a certain class based on a weighed voting of all the overlapping segments. The mathematical expression for this operation is shown in equation 5.2,

$$C_{T_i} \equiv arg_{c_k} max \sum_{\substack{S_j \in c_k \\ T_i \cap S_j \neq \varnothing}} w_k \cdot e^{-\frac{d_{ij}^2}{2 \cdot \sigma^2}} \tag{5.2}$$

where T_i is the i_{th} interval, $d_{i,j}$ is the distance from the centre of the j_{th} segment (S_j) overlapping with the i_{th} interval to the centre of the i_{th} interval, c_k is the k_{th} segment class and w_k is a class weight normalised so that $\sum w_k = 1$. The sum is to be taken over the segments intersecting with the interval T_i, belong to class c_k (unclassified segments are excluded) and fulfill the threshold requirement $e^{-\frac{d_{ij}^2}{2 \cdot \sigma^2}} >= 0.14$, where

σ is the variance of the Gaussian and the value 0.14 is obtained when $d_{ij} = 2 \cdot \sigma$. The reason for the latest requirement is to create a cutoff for the segments that are too far away from the centre of the interval. The parameter σ controls the weight of the vote of each segment based on its distance from the interval and in this analysis it was set equal to R in order to achieve proportionality with the arena dimensions (other values have also been tested, refer to the Figure 5.3). Finally, the class weight w_k was defined as $w_k = \frac{1}{P(c_k)}$, where $P(c_k)$ is the percentage of segments belonging to class k. The intuition for setting the class weights inversely proportional to the amount of segments that fall under each class was to prevent rare classes from being overshadowed by common ones. To prevent having too small or too large class weights the bounds of [0.01 0.5] were set, which means that if less than 1% or more than 50% of the segments fall under a certain category then this class will receive weight equal to 0.5 or 0.01 respectively.

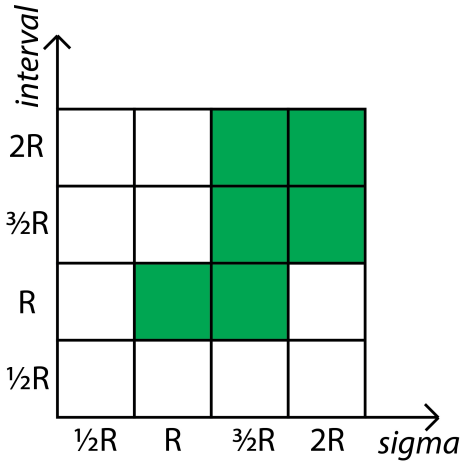


Figure 5.3: Empirically defined area of tuning for the smoothing function. R refers to the arena radius (in cm); x-axis (sigma) refers to a particular value of σ (variance of the Gaussian); y-axis (interval) refers to a particular value of the length of the interval; green boxes indicate areas under which the smoothing function (refer to section 2.7 Mapping Segment Classes to the Full Swimming Paths) yields consistent results for every segmentation (excluding Segmentation I where the segments length is too large). Interval of length $2 \cdot R$ is at the limit and from this point onwards consistency cannot be sustained.

5.2.7 Statistics

The non-parametric Friedman test [Siegel, 1956] was used for the analysis of variance of each strategy between the two animal groups. This test was selected because the data are not normally distributed and because of its ability to control the variability among subjects over the different observations [Theodorsson-Norheim, 1987].

For the analysis the null hypothesis is that there in no difference between the two animal groups (stressed and control) over each one of the strategies (refer to section 5.2.9) as well as over the number of times that the animals change their behaviour within single trials (strategy transitions). Small p-values (< 0.05) generated by the Friedman test lead us to discard the null hypothesis that the results are identical and that any differences are only due to chance (random sampling). When the test is used the Friedman test p-value and the Friedman's chi-square statistic (Q) [Hollander

and Wolfe, 1999] are reported. Since the comparison is between two animal groups, stress and control, there are $k = 2$ variables and the degrees of freedom are equal to $df = 1$.

In addition to the Friedman test, the 95% confidence intervals of a binomial distribution [Wallis, 2013] are being used, where the significance of a specific classification, as judged by each of the classifiers that form the ensemble, is viewed as a random process generating one (significant differences) or zero (non-significant differences). In more detail, the confidence intervals indicate the degree of confidence that the classifiers forming the ensemble are on average pointing to the same conclusion as the ensemble (i.e. the majority agrees that there is significant difference over strategies or strategies transitions). Given that the Friedman test can have two outcomes, it is hypothesised that the outcomes are the result of a binomial distribution. It is required that the 95% confidence intervals to be clearly above 0.5 (or 50%) in order to be confident that the result in not due to chance [Brown et al., 2001; Peck, 2012].

5.2.8 The RODA software

RODA [Vouros et al., 2017] consists of a series of graphical user interfaces (GUIs) which offer straightforward analysis of trajectory data extracted from the Noldus Ethovision System [Noldus et al., 2001]. Every stage of the process can be tuned to meet the user's needs. The generated figures can be exported into a variety of different image formats (JPEG, TIFF, etc.) while the numerical data depicted in the figures are also saved in Comma Separated Values (CSV) file format in case the user wishes to generate the figures using a different software (e.g. Microsoft Excel).

The software is entirely written in MATLAB [MATLAB, 2016*b*] and uses a modified version of the WEKA library [Frank et al., 2016] written in Java which is known as WekaUT (for more information refer to http://www.cs.utexas.edu/users/ml/risc/code/).

The code of RODA is open-source and available on the github repository https://github.com/Rodent-DataAnalytics/mwm-ml-gen. The code requires the MATLAB's Statistics Toolbox [MATLAB, 2016*a*] to be installed. Compiled versions of the software are also available for Windows and MAC OS (see the releases tab of the repository https://github.com/RodentDataAnalytics/mwm-ml-gen/releases).

5.2.9 Classes of behaviour and strategy transitions

The choice of the classes of behaviours (strategies) in this analysis is motivated by previous studies (e.g. [Graziano et al., 2003; Wolfer and Lipp, 1992; Wolfer et al., 1998]) which have observed and reported stereotypical animal behaviours inside the MWM (for an example of each strategy refer to Figure 5.4).

- **Thigmotaxis (TT).** The animal moves exclusively on the periphery of the arena and most of the time it touches the walls of the arena.
- **Incursion (IC).** The animal starts to distant itself from the arena periphery with visible inward movements.
- **Scanning (SC).** A behaviour associated with random searches focused in the centre of the pool. Another characteristic of this behaviour is that the animal rapidly turns away from the arena walls if it touches them [Graziano et al., 2003].

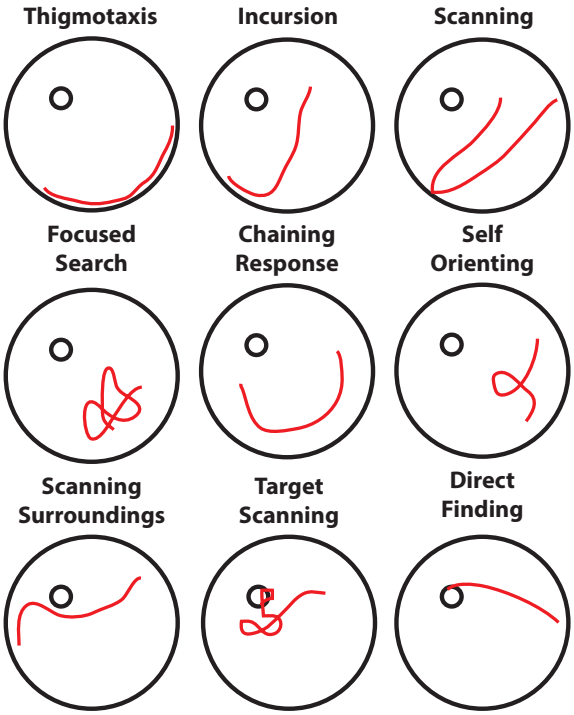


Figure 5.4: Stereotypical classes of behaviour. Each figure shows an example of a trajectory segment falling under each behavioural class. Throughout the experiment, the animals implement different strategies in order to solve the maze. By detailed analysis of each trial trajectory data into segments the interchange of these stereotypical animal behaviours becomes visible.

- **Focused Search (FS).** This behaviour is also associated with random searches but here the animal actively searches a particular small region of the arena.

- **Chaining Response (CR).** A behaviour first observed in the study of Wolfer et al. [Wolfer and Lipp, 2000] where the animal appears to have memorised the distance to the platform from the arena wall and swims circularly in order to find it.

- **Self Orienting (SO).** The animal performs a loop and orients itself inside the arena [Graziano et al., 2003].

- **Scanning Surroundings (SS).** The animal crosses a region very close to the platform of the arena but moves away [Gehring et al., 2015].

- **Scanning Target (ST).** The animal actively searches for the arena by swapping paths around it.

- **Direct Finding (DF).** The animal navigates straight to the platform.

- **Strategy Transitions (tr).** In addition to the behavioural strategies, analysis on the number of times that the animals change their behaviour within single trials was also performed.

5.2.10 Morris Water Maze experimental procedure and data properties

The data have been collected from experiments performed at the Laboratory of Behavioural Genetics, EPFL at Lausanne, Switzerland. All procedures were conducted in conformance with the Swiss National Institutional Guidelines on Animal Experimentation and approved by a license from the Swiss Cantonal Veterinary Office Committee for Animal Experimentation.

The water maze had a diameter of $200cm$ with a submerged platform of diameter $12cm$. The recordings of the animals trajectories were performed by using the tracking software, Noldues EthoVision [Noldus et al., 2001] version 3.1. The data set contains 57 rats, 30 of which were inducted into stress at peripubertal age [Márquez et al., 2013] and 27 of which were the control group. In the previous study of [Gehring et al., 2015], 3 stress animals were removed due to missing trials and low levels of stress based on the animal speeds. A total of 12 trials were performed per animal divided into 3 consecutive days with 4 trials per day. The timeout of each trial was 90 seconds and if the animal failed to find the platform within the time limit it was guided to it. The inter-trial interval between the trials of the same day was only a few minutes. The starting position of the animals was altered between trials.

The data are available in the same GitHub repository that hosts RODA (https://github.com/RodentDataAnalytics/mwm-ml-gen). RODA has a demo function embedded for importing the data and reproducing the results of this work (see the Wiki section of the repository).

5.3 Results

5.3.1 Trajectory Segmentation Analysis (TSA) & the RODA software

The generic framework allows Morris Water Maze trajectory segmentation analysis that requires little input from the user. Trajectories are divided into overlapping segments, a percentage of which (8% to 12%) are labelled by an expert user as belonging to one of eight different behavioural strategies. Multiple labels can also be used for a segment (see Methods for more information about the behavioural classes). The remaining segments are automatically classified via a semi-supervised clustering algorithm to one of the user-defined strategies, and via a smoothing procedure are mapped back to the full trajectories. This procedure allows the identification of multiple strategies in a single trial.

The user, in addition to providing labels, needs to define the segmentation length and overlap. For the segmentation parameters, appropriate regimes have been identified for the MWM with dimensions from 2 up to 2.5 times the arena radius (see Results: Robustness across different segmentation configurations).

In order to reduce the required tuning from the user (an issue of the previous work of Gehring et al. [Gehring et al., 2015]) and improve the objectivity of the classification, ensembles of classifiers were employed that vote to assign the segment to a strategy according to a simple majority voting rule.

A software, called RODA [Vouros et al., 2017] (shown in Figure 5.5), has been developed in order for the proposed framework to be available for usage

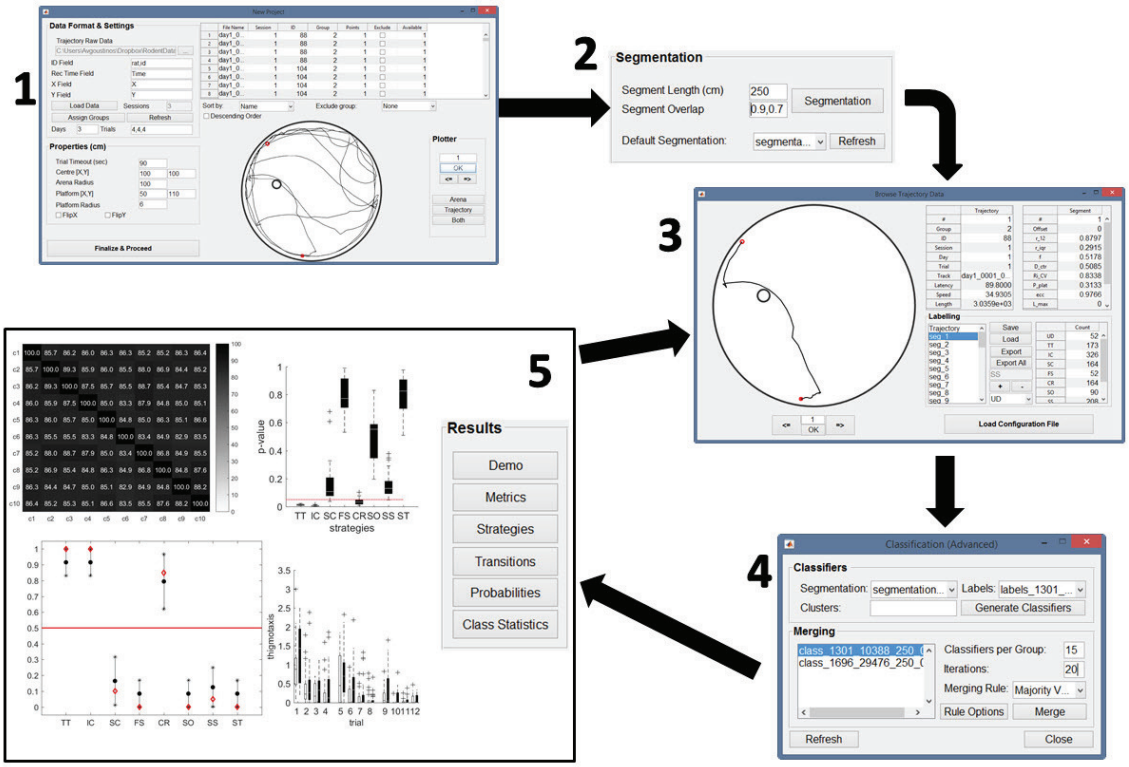


Figure 5.5: Screenshots of the software RODA. Each window is numbered to denote a separate stage of the workflow, which consists of: **(1)** the data input GUI, which is used to load the trajectory data extracted from Ethovision and select the specific tracks that will be used in the analysis; **(2)** the segmentation panel, which offers full control over the segmentation options; **(3)** the labelling GUI, which offers visualisation of entire trajectories and their segments allowing easy labelling of the segments; **(4)** the classification GUI, which contains options to tune various parts of the classification process (a default option is also available); **(5)** the results panel; which generates the analysis results. The results are generated in both graphical and textual formats. The user also has control over the output format of the image files as well as the elements of the generated figures such as text size, line width, etc. The arrow connecting **(5)** with **(3)** indicates that if the analysis results are not consistent then there is the need to go back to the labelling stage and provide additional or improved labels.

by the scientific community. The software is available on the github repository https://github.com/RodentDataAnalytics/mwm-ml-gen under the GNU General Public License version 3 (GPL-3.0). A manual of the corresponding software can be found under the wiki section of the repository (https://github.com/RodentDataAnalytics/mwm-ml-gen/wiki) details on the methodology and technical information about RODA can be found under Materials and Methods.

5.3.2 Advantages of Trajectory Segmentation Analysis (TSA)

The proposed methodology finds quantitative behavioural differences beyond those identified by standard metrics on the full swimming paths of the animals. It is able to detect additional significant differences between the behavioural strategies employed by the two or more animal groups in comparison to the categorisation of the whole animals trajectories. In more detail, from a manual behavioural analysis of the whole swimming paths of the animals the strategies thigmotaxis, incursion, scanning, self oriented, target scanning and direct finding are detected and analysed.

By using TSA the additional behavioural classes of focused search, chaining response and scanning surroundings were able to be identified and analysed (for more details on the aforementioned classes of behaviour refer to Methods and Figure 5.4).

The framework was applied to the data set of Gehring et al. [Gehring et al., 2015] composed of two rodent groups (stressed and control rats). The same data was selected as a benchmark to show the improved method of this study and to demonstrate its robustness and generality. The two animal groups differ on the strategies of Thigmotaxis (Friedman test p-value = 0.004, $Q = 8.516$, $k = 2$), Incursion (Friedman test p-value = 0.009, $Q = 6.811$, $k = 2$) and Chaining Response (Friedman test p-value = 0.007, $Q = 7.220$, $k = 2$) in favour of the stressed group meaning that stressed animals implement these strategies more often than the control group. In addition, stressed animals tend to transit between different strategies more often than the control animals (Friedman test p-value = 0.037, $Q = 4.340$, $k = 2$). For relevant results refer to Figure 5.6.

Commonly used measurements of learning (animal speed, escape latency and path length) suggest that there is a significant difference among the two animal groups in the sense that the stressed animals are faster (Friedman test p-value = $2\text{x}10^{-8}$, $Q = 31.510$, $k = 2$) and swap longer paths (Friedman test p-value = 0.002, $Q = 9.836$, $k = 2$) within the trials but they still fail to find the platform in less time than the control animals (Friedman test p-value = 0.154, $Q = 2.030$, $k = 2$). For relevant results refer to Figure 5.7 for the relevant results). Manual classification of the full swimming paths to different behavioural strategies was performed due to the small amount of data; this analysis suggests that the reason for this phenomenon is because stressed animals tend to use the low level strategy of Thigmotaxis more than the control group (Friedman test p-value = 0.015, $Q = 5.888$, $k = 2$), which lowers their chances of finding the platform since they spent most of the time close to the arena periphery (refer to Figure 5.8 for the relevant results). TSA agrees on that conclusion but it is also able to detect that stressed animals tend to use *a series* of low level strategies, both Thigmotaxis (Friedman test p-value = 0.004, $Q = 8.516$, $k = 2$) and Incursion (Friedman test p-value = 0.009, $Q = 6.811$, $k = 2$, refer to Figure 5.6), which lower their chances of finding the platform since they spent most of the time on *or close to* the arena periphery. In addition, stressed animals implement the Chaining Response strategy more often than the control animals (Friedman test p-value = 0.007, $Q = 7.220$, $k = 2$, refer to Figure 5.6), which implies that they haven't memorised the location of the platform but its distance to the wall [Wolfer and Lipp, 2000]; so they swim at that distance in hope to find it by chance; a behaviour that is again, on average, time consuming. Furthermore, TSA allows the detection and quantification of behavioural switching which shows that stressed animals change their behaviour inside the arena more often than the control animals (Friedman test p-value = 0.037, $Q = 4.340$, $k = 2$, refer to Figure 5.6). These results are relevant to studies such as [Aston-Jones et al., 2000; Luksys et al., 2009; Luksys and Sandi, 2011] which suggest that high levels of stress lead to weak attention and frequent behavioural switches.

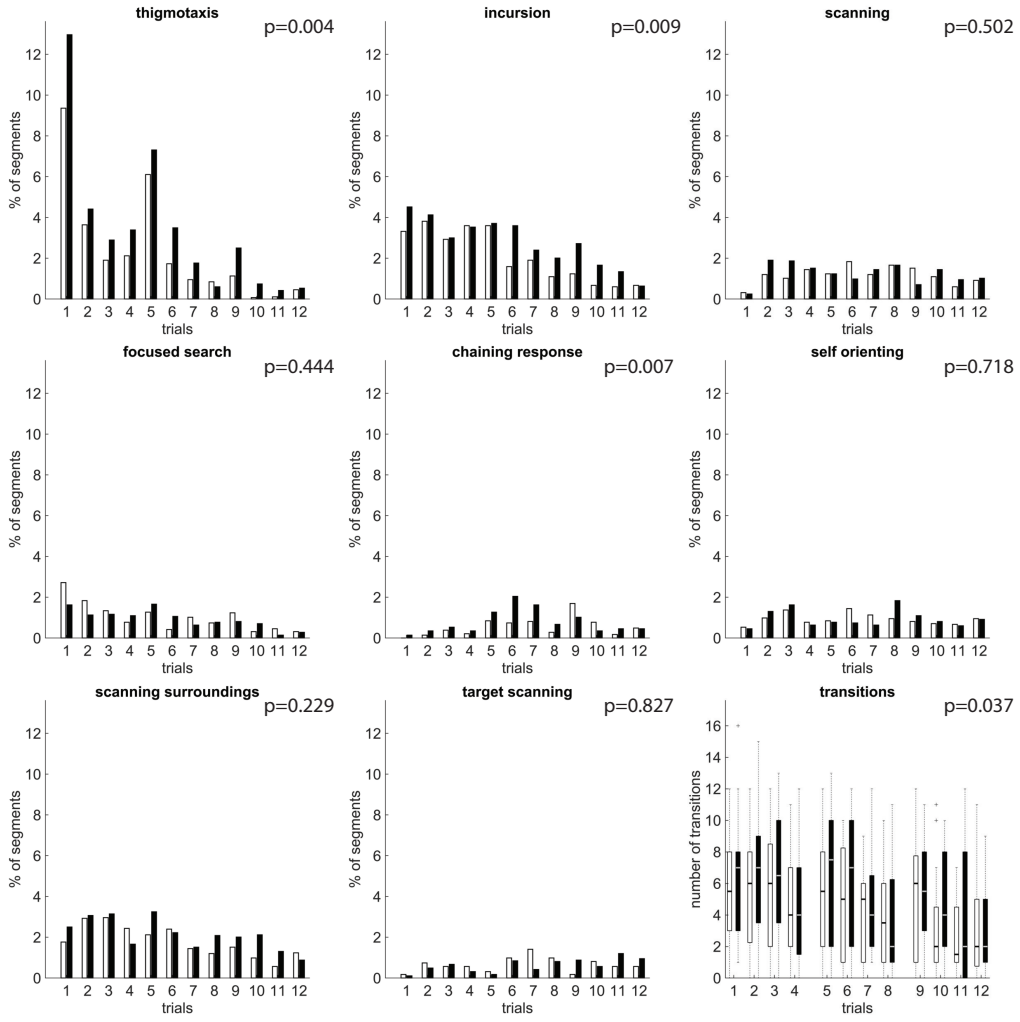


Figure 5.6: Percentage of segments falling under each strategy for the stressed (black) and control (white) animal groups over each trial. All the animals were tested for a set of 12 trials divided in to 3 sessions (days). Each segment (path interval; see Methods) is considered to be of length equal to the length of the arena radius (100cm). For the transitions: bars represent the first and third quartiles of the data; the black (control group) or white (stressed group) horizontal lines are the medians, crosses are the outliers and whiskers indicate the minimum and the maximum values. These results were generated by using a segmentation length of 2.5 times the arena radius (250cm) and 90% overlap; for the classification an ensemble of classifiers was created by using classifiers with validation error less than 25%. The Friedman test p-value (shown on the top right) was used to compare both animal groups for the complete set of trials. According to the plots Thigmotaxis and Incursion strategies show a clear difference in favour of the stressed groups (Friedman test p-value = 0.004, $Q = 8.516$, $k = 2$ and p-value = 0.009, $Q = 6.811$, $k = 2$) along with Chaining Response (Friedman test p-value = 0.007, $Q = 7.220$, $k = 2$). The number of transitions between strategies shows that the stressed animals change their behaviour more often than control animals within single trials (Friedman test p-value = 0.037, $Q = 4.340$, $k = 2$). Segmentation analysis is able to distinguish more behavioural differences between the two groups in comparison with the classification of the full swimming paths (see Figure 5.8), which are then consistent with the performance measurements (see Figure 5.7); stressed animals, despite running faster and sweeping longer swimming paths, require the same amount of time to detect the arena because they implement a series of inefficient strategies (i.e. Thigmatoxis and Incursion) or less effective strategies (i.e. Chaining Response). Furthermore they are switching behaviours (transitions) more often than the control animals indicating a loss of focus of finding the platform. The Direct Finding class was excluded from this figure because for this class the statistical analysis gives quantitatively the same results as in Figure 5.8)

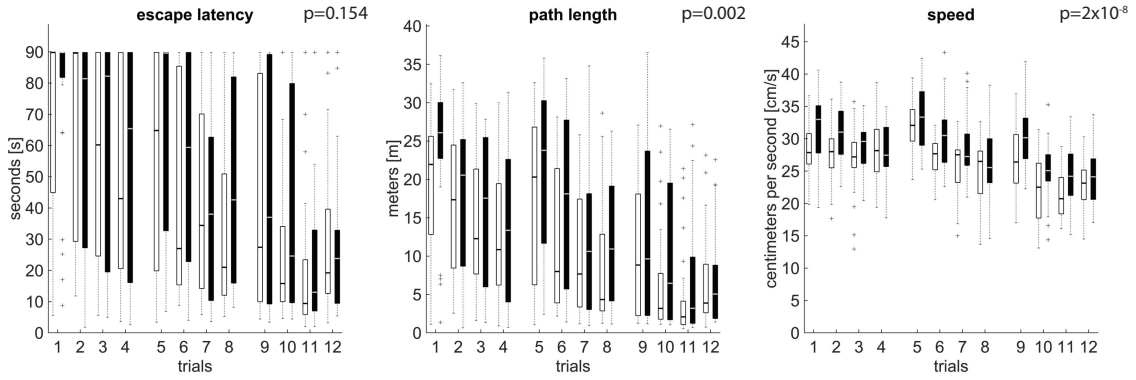


Figure 5.7: Full swimming path standard metrics for the stressed (black) and control (white) animal groups. All the animals were tested for a set of 12 trials divided in 3 sessions (days). Bars represent the first and third quartiles of the data; the grey line that splits the bars represents the median, crosses are the outliers and whiskers indicate the minimum and the maximum values. The Friedman test p-value over the trials is shown on the top right of each plot. Stressed animals find the platform as fast as the control group (escape latency, p-value = 0.154, $Q = 2.030$, $k = 2$) even though they run faster (path length, p-value = $2\text{x}10^{-8}$, $Q = 31.510$, $k = 2$) and sweep (on average) longer swimming paths (speed, p-value = 0.002, $Q = 9.836$, $k = 2$) within the trials than the control group.

5.3.3 Robustness across different segmentation configurations

It is expected that the segmentation length affects the results, i.e. a full trajectory will not reveal more than one strategy or a very small segment will not have enough information for mapping it onto a strategy. Therefore focus was given on segmentation lengths between 2 times and 3 times the arena radius in order to investigate the robustness of the process.

The animal swimming paths inside the maze were segmented using four different segmentation configurations (different segment length and/or segment overlap). For each segmentation, labels were provided to approximately 10% of the segments (refer to Table 5.1 for a summary of the different configurations) and the framework was used to classify the rest. The conclusions were based on both the ensemble classification result as well as the percentage of classifiers in an ensemble that agree to this result (95% binomial confidence intervals clearly above 50%). Three out of four segmentation configurations (with segment lengths 2 and 2.5 times the arena radius) led to the conclusion that the two animal groups (stressed and control) have significant difference on the strategies of Thigmotaxis, Incursion and Chaining Response and strategy transitions (Friedman test p-value < 0.05 and 95% binomial confidence intervals clearly above 50%, see Figure 5.9 for detailed statistics) in favour of the stressed group meaning that stressed animals implement these strategies and transit between different strategies more often than the control animals. One out of four segmentations (segment length of 3 times the arena radius) failed to capture significant difference in the Chaining Response strategy and a probable reason is that the segment length is too large, thus strategies that are rarer and significantly smaller are overshadowed by more common ones (e.g., Chaining Response may be overshadowed by Scanning Surrounding or Thigmotaxis, refer to Table 5.5). This is an issue introduced already during the labelling procedure. For example, in

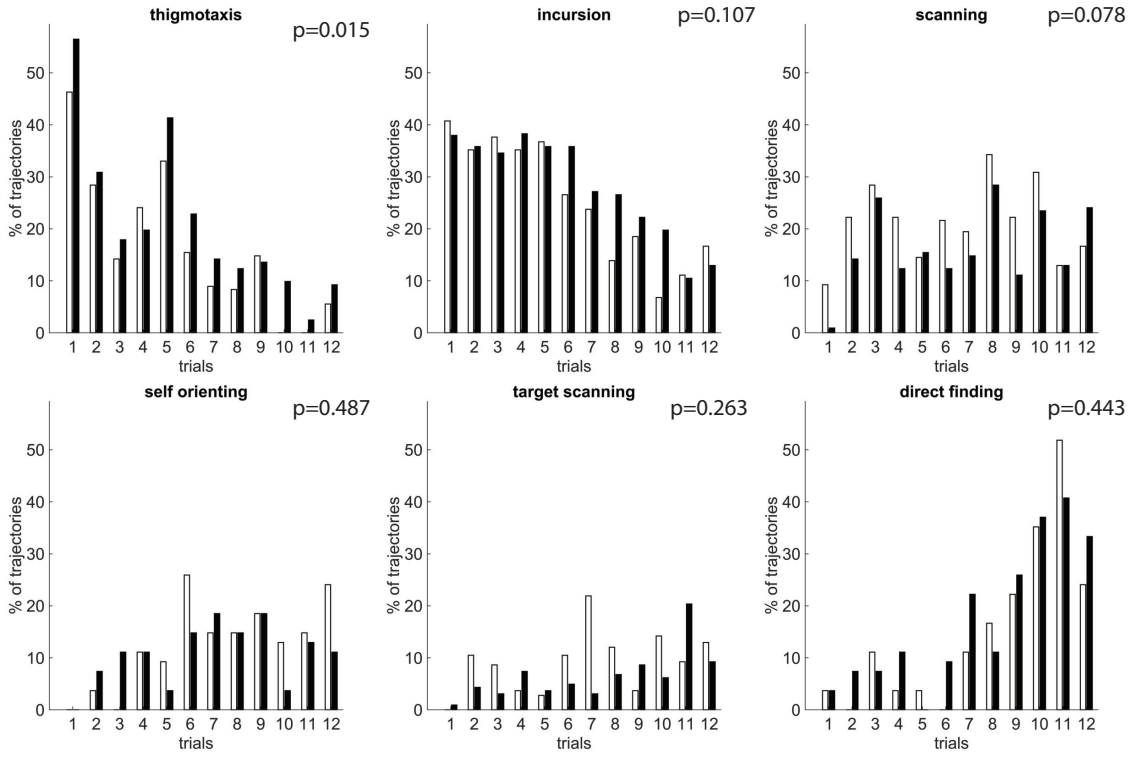
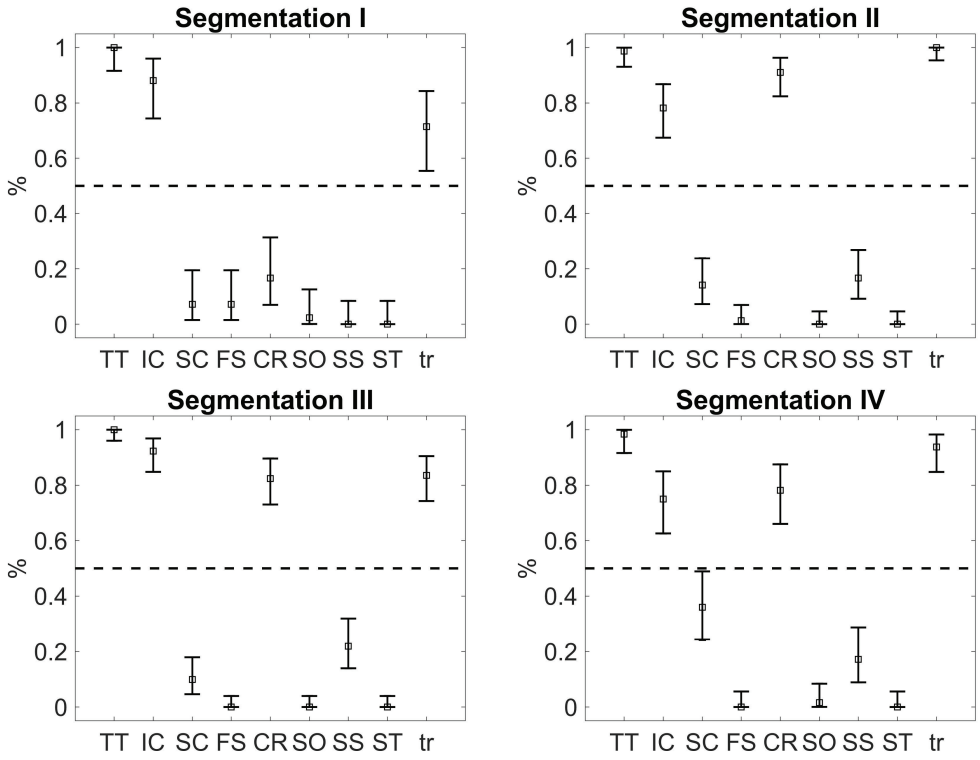


Figure 5.8: Manual classification of the full swimming paths. White bars: control group; Black bars: stressed group; the two groups were compared over the complete set of trials using the Friedman test (shown on the top right corner of each graph). In the manual classification of the full swimming paths, certain behavioural classes (Focused Search, Chaining Response and Scanning Surroundings) couldn't be identified. Significant difference (Friedman test p-value = 0.015, $Q = 5.888$, $k = 2$) was detected only for the Thigmotaxis strategy in favour of the stressed animal group, indicating that stressed animals are implementing it more often than the control animals and have less chances of detecting the platform. This is relevant to the performance measurements (see Figure 5.7) where stressed animals run faster and sweep longer swimming paths, but still fail to find the platform in less time than the control group. For more information about each behavioural strategy refer to Methods.

Segmentation 1 only 0.67% of the samples were single-labelled as *chaining response* vs 1.58%, 0.72%, 1.06% in the Segmentations 2 to 4 correspondingly. The larger segment makes it more difficult for the human expert to distinguish rare classes that are adjoint to frequent ones.

5.4 Discussion

Methodologies that classify swimming paths in MWM to behavioural classes can reveal different stages of learning in animal groups. However, up to now, there are very few examples of earlier research that have made use of machine learning techniques to automatically detect animal behaviours. Most of them have proposed methods that are difficult to generalise and require machine learning knowledge. In the previous study of [Gehring et al., 2015] the limitations of the previous techniques were addressed by focusing on the fact that forcing whole swimming paths into a single class of behaviour can be suboptimal as each trajectory incorporates a number of different behaviours. The methodology of detailed trajectory classification



Ensemble Result

Friedman test p-values and *Q* per strategy and transitions (k=2)

p-val	TT	IC	SC	FS	CR	SO	SS	ST	tr
I	**0.008**	**0.011**	0.450	0.205	0.156	0.960	0.271	0.571	**0.035**
II	**0.005**	**0.013**	0.157	0.278	**0.004**	0.639	0.190	0.345	**0.019**
III	**0.004**	**0.009**	0.502	0.444	**0.007**	0.718	0.229	0.827	**0.037**
IV	**0.004**	**0.005**	0.156	0.821	**0.008**	0.749	0.436	0.389	**0.038**
Q	***TT***	***IC***	***SC***	***FS***	***CR***	***SO***	***SS***	***ST***	***tr***
I	***6.996***	***6.442***	*0.571*	*1.607*	*2.015*	*0.003*	*1.213*	*0.320*	***4.425***
II	***8.003***	***6.187***	*1.998*	*1.176*	***8.537***	*0.221*	*1.718*	*0.891*	***5.516***
III	***8.516***	***6.811***	*0.451*	*0.585*	***7.220***	*0.131*	*1.445*	*0.048*	***4.340***
IV	***8.236***	***8.008***	*2.013*	*0.051*	***7.128***	*0.102*	*0.607*	*0.742*	***4.291***

Figure 5.9: Conclusive results from the classification of each segmentation configuration (see Table 5.1). Each plot shows the 95% binomial confidence intervals for the classifiers of each segmentation regarding their agreement on the significant difference between the two animal groups for each strategy and the strategy transitions. Squares indicate the mean of the classifiers; errorbars represent the 95% confidence intervals; the dashed line indicates the threshold of interest (0.5 or 50%). Confidence intervals clearly above 0.5 (or 50%) confirm that there is indeed a significant difference between the the two animal groups on the strategies and the strategy transitions. The table below the plots shows the Friedman test p-values (upper table) and the equivalent Friedman's chi-square statistic (lower table) for the classification result of the ensemble; in all cases $k = 2$, control and stress columns. Segmentation configurations are arranged in columns and strategies in rows; each element has the relevant p-value and chi-square statistic and bold cells indicate significant difference, i.e. p-value < 0.05. Abbreviations: Thigmotaxis (TT), Incursion (IC), Scanning (SC), Focused Search (FS), Chaining Response (CR), Self Orienting (SO), Scanning Surroundings (SS), Target Scanning (ST), Strategy Transitions (tr) (refer to Methods for more information on each behavioural strategy). It is shown that in three cases (Segmentations II, III, IV) the two animal groups show significant differences in the strategies of Thigmotaxis (TT), Incursion (IC) and Chaining Response (CR) and transition between strategies (tr). It is observed that while Segmentations II, III, IV agree that there is significant differences on the Thigmotaxis, Incursion, Chaining Response and transitions, Segmentation I fails to capture the significant difference on the Chaining Response because of the lengthy segments which caused this strategy to be overshadowed by other strategies and disappear (refer also to Table 5.5).

can reveal additional behavioural differences between two groups of animals and can be used even when small amount of trajectory data are available since the segmentation process, due to overlapping, typically creates a significant amount of data. Nevertheless, the previously proposed method of segmented trajectories classification required a certain degree of machine learning knowledge to be used correctly, and allowed an amount of subjectivity when choosing classifiers.

This work addresses these issues by proposing to improve the robustness of the technique via majority voting. The results are no longer based on a single classification tuning (classifier) but on the agreement of many. This technique alleviated the subjective assignment of the swimming path segments to classes since, in practice, many classifiers that seemingly perform equally well in validation, have relatively high disagreement, and how to best chose among them might be unclear. Here, different segmentation configurations are systematically investigated to identify the bounds under which the method produces meaningful results. The bounds refer to the minimum and maximum segmentation length and the number of labels that needs to be provided. Furthermore, the binomial confidence intervals on the ensemble of the classifiers are informative regarding the quality of the results.

The data set from the previous work of [Gehring et al., 2015] was used as a benchmark of the new methodology and also as a way to demonstrate its robustness and generality. There is a similarity of the results of this study and the study of [Gehring et al., 2015] but with one difference; in this study significant difference for the scanning strategy is not detected as it was the case in [Gehring et al., 2015] based on the result of a single classifier. This is due to a number of factors: (i) the use of only one classifier, which results in higher error (see also the confidence intervals in Figure 5.9), (ii) the merging of three different segmentations that resulted in classifications that did not fully agree with each other. Here the conclusions are based on the majority voting of many classifiers that are shown to have an improved performance versus the simple classifiers, and therefore lead to more reliable results.

One important point that should be mentioned is that despite the fact that for each segmentation the ensemble formed has extremely low to zero error percentage, the largest segmentation failed to indicate difference on the Chaining Response strategy. The cause of this issue was identified to be the difficulty involved when labelling large segments; in this case the Chaining Response can be masked by more dominant classes such as Thigmotaxis. It is worth noting that the smoothing function, which is used to map the segments back to the whole trajectories, again do not affect the conclusions formed based on the strategies. Even without the smoothing function, again, three segmentations agree on the differences between the two animal groups on the Thigmotaxis, Incursion and Chaining Response strategies while the segmentation with the more lengthy segments cannot capture the difference on Chaining Response (refer to the Appendix C for the non-smoothed classification results). For this reason, the criterion for correct classification cannot be based on the classification error alone. Consistent results within a reasonable variation of the segmentation length is also a requirement, in this particular case the variation was between 200 and 250cm, i.e., $2R$ and $2.5R$, with R being the radius of the maze. These segmentation parameter arranges are verified and directly applicable to other Morris Water Maze experiments (refer to [Huzard et al., 2019] or section 5.5 of this dissertation).

To facilitate the use of this methodology by the scientific community, a complete

software incorporating of the framework is provided which includes a Graphical User Interface (GUI) to guide the user throughout all the analysis stages and allows for the manual configuration of each procedure.

Although the proposed framework is able to detect behavioural information in much detail it should highlighted that it has a substantial limitation; it is not able to detect behavioural strategies of lengths shorter than $2 \cdot R$. Thus if groups have different lengths, e.g. if only one group is having lengths more than $2 \cdot R$ then the current method is still applicable but it will not provide much information about behavioural differences because most of the segments will be classified as Direct Finding. Main reasons for this limitation are the following: (a) it is difficult to put manual labels to segments with lengths below $2 \cdot R$ and (b) trajectories with lengths below the length of the segmentation tuning will be automatically classified as Direct Finding. To alleviate this limitation the user has the ability to provide labels to trajectories shorter than the specified segmentation tuning. Nevertheless, this happens only after the classification procedure meaning that these trajectories are not taking part in the semi-supervised clustering and classification; this implies that if the majority or all of the path lengths are shorter than $2 \cdot R$ the proposed framework is reduced to completely manual behavioural classification. In such cases traditional performance measurements would potentially be more effective.

A proposed future application of the current framework is to address differences in sequence of behavioural strategies. As it was suggested in the literature [Hamilton et al., 2004; Whishaw and Mittleman, 1986], strategies within one trial occur in reliable sequences. Since the output of the method are sequences of strategies, one could potentially apply a Markovian analysis [Gagniuc, 2017] on the sequences and detect differences between animal groups on the probabilities of transition between behavioural strategies. A more detailed analysis on the different kinds of transition has the potential to reveal additional differences among animal groups. Such analysis can be viewed as a Markov model where each behavioural strategy is a state and when the animal is in a particular state it can either remain in the same state, i.e. repeat the same behaviour, or transit to another behaviour.

Finally, it should be noted that the work that is presented here can generalise to other species of rodents inside the MWM (e.g. mice) as well as other experiments similar to the MWM (e.g. open field tasks, place avoidance). Two main significant changes to be made are the strategy definitions and the trajectory features. In the unpublished work of [Gehring et al., 2017] the issue of pre-defined strategies is addressed by using a fully unsupervised procedure to find patterns of behaviour in the active allothetic place avoidance task. In that experiment there is no previous knowledge of animal behaviours thus supervised or semi-supervised techniques cannot be applied. However, it is mentioned that the classification of this study depends on the trajectory features that are used. A combined work of the classification boosting technique, an unsupervised methodology [Gehring et al., 2017], and the engineering of trajectory features that not linked to a specific experiment has the potential to lead to a robust generalised framework of trajectory analysis for many different animal species used in experimental procedures (e.g. octopus [Boal et al., 2000] and zebrafish [Gerlai, 2017]).

5.5 Further application

The RODA procedure described before has been further applied in the study of [Huzard et al., 2019]. The purpose of this study was to investigate whether endogenous differences in glucocorticoid responses influenced spatial learning, long-term memory, and reversal learning abilities inside the Morris Water Maze (MWM) at early aging.

In more detail, it has been reported that stress modulates the navigation in animals [Schwabe et al., 2010; Schwabe and Wolf, 2010] and humans [Van Gerven et al., 2016] during spatial tasks and rats that were inducted to stress during early stages of their life are showing impaired navigation in the MWM by performing low level strategies such as Thigmotaxis and Incursion and adapting non-spatial strategies such as Chaining Response [Gehring et al., 2015; Vouros et al., 2018]. The inverted U-shape relationship between stress and spatial memory proposition of [Yerkes et al., 1908] suggests that cognitive performance is best under optimal stress based on the difficulty of the task and levels of stress above or below the optimal will result to impaired performances [Salehi et al., 2010]. Based on this proposition the hypothesis is that rats with low level of corticosterone response would perform worse in navigation tasks such as the MWM in comparison with rats with medium level of corticosterone response. Rats with specifically modulated higher levels of corticosterone response to match the stress challenge of the MWM would potentially be the best performers.

5.5.1 Experimental procedure properties

The maze consisted of a black circular pool (diameter of $200cm$ and height of $45cm$) filled with $30cm$ of water at 23 ± 1^oC and virtually divided into four equivalent quadrants: northeast (NE), northwest (NW), southeast (SE), and southwest (SW). A circular hidden platform (diameter of $10cm$ and distance between platform center point and pool wall was $30cm$) was submerged 1 to $2cm$ below the water surface. The testing room was illuminated (50 ± 10 lux) by lights placed below the pool to avoid light reflections. To monitor the animals, a camera was mounted to the ceiling above the center of the pool. The water maze was surrounded by extra-maze cues of different shape, size, and color.

The data were consisted of male rats with Low, Intermediate and High lines of corticosterone response (for more details on the selected breeding refer to [Huzard et al., 2019]). Each group had 10 subjects but one rat from the High line was excluded from the experiment, in accordance with the Swiss Animal Experimentation Guidelines, due to critical health issue at 15 months of age (tumor growth on forelimb). The experiment setup (shown in Figure 5.10) was chronologically the following:

- 23 training trials between days 1 to 5 as follows: 5 trials the first three days and 4 trials the last two days. This subset of data will be refereed to as *train1*.
- 1 probe trial on the 5th day. This subset of data will be refereed to as *probe1*.
- 1 probe trial on the 17th day. This subset of data will be refereed to as *probe2*.
- 3 training trials on the 17th day. This subset of data will be refereed to as *train2*.

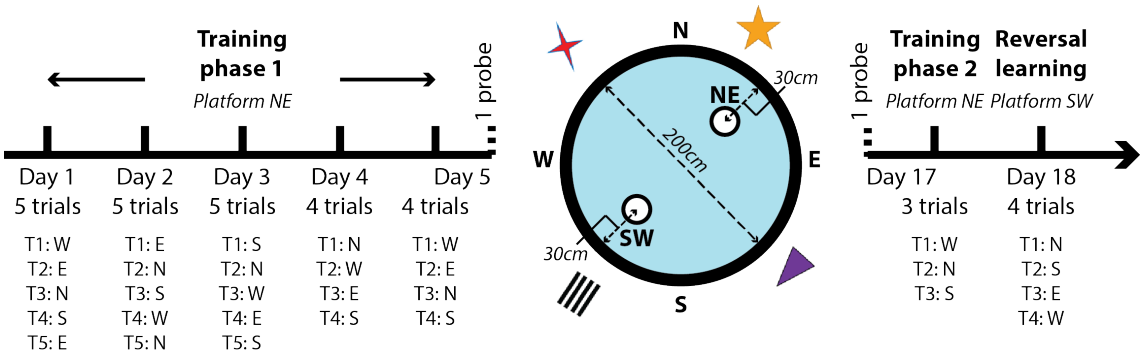


Figure 5.10: Schematic representation of the experimental protocol. During day 1 to day 5 (*train1*), rats were training to escape the maze by reaching the platform in the northeast (NE) quadrant. After training on day 5, a first probe trial was performed *probe1*. On Day 17, a second probe trial was performed *probe2* followed by 3 re-training trials (*train2*). On day 18, the platform was placed in the southwest (SW) quadrant and rats were trained for 4 trials (*reverseT*). Starting positions followed a semi-random sequence and are indicated below each training days. The pool was surrounded by visual cues of different shape, size and color.

- 4 reversal trials on the 18th day. This subset of data will be refereed to as *reverseT*.

5.5.2 RODA tuning and data analytics

The swimming paths of the animals were firstly analysed using the RODA software software version 4.0.2 [Vouros et al., 2017, 2018] which implements an updated analysis procedure based on the previous study of Gehring et al. [Gehring et al., 2015]. RODA has the ability to segment the swimming paths and classify each segment allowing detailed analysis on the different behaviours within each experimental trial. Because the segmentation procedure is based on overlapping segments, the segments are then mapped to the original trajectories considering specific path intervals. These intervals, which essentially represent the different behaviours of the animal during a specific trial, were collected for more sophisticated statistical analysis than the one offered in RODA. Tuning of RODA was performed based on [Vouros et al., 2017, 2018] which is as follows:

- **Segmentation:** The length of the segments was selected to be 2.5 times the arena radius ($2.5 \cdot 100cm$) with 70% overlap.
- **Labelling:** Labels were manually provided to the subsets *probe1, probe2, train2, reverseT* of the data set.
- **Intervals:** The intervals are of length equal to the arena radius ($100cm$).

The analysis procedure was also performed for path segments of 2.7 times the arena radius ($2.7 \cdot 100cm$) with 70% overlap (results not shown). There were no qualitatively differences between the two analyses. Manual assessment on all the classified segments was also performed. There were two statistical analyses performed on the data leading to the same conclusions. The published analysis (see [Huzard et al., 2019]) was performed in EPFL and presents the findings in a more concise way. It was based on the analysis that is described in this dissertation.

5.5.2.1 Statistical analysis methods

Statistical analysis and comparison among the animal groups was performed on the path intervals (each interval has length equal the arena radius) falling under 9 different classes (refer to Figure 5.4). When the compared animal groups had equal number of animals but contained missing values (missing trials) statistical comparison was performed by trial using the Skillings-Mack test [Chatfield and Mander, 2009] instead of the Friedman test that RODA offers. Skillings-Mack test is a generalization of the Friedman test when there are randomly missing data, for a relevant study regarding this test refer to [Lai et al., 2012]. In this particular case some animals performed 22 instead of 23 trials thus the Friedman test was not appropriate. Skillings-Mack test was used only on the *train1* subset since it was the only one containing missing values. The rest of the data were analysed using the Friedman test [Siegel, 1956]. Statistical analysis testing was not performed for the probe trials since the Friedman test requires at least three different occasion measurements on each group, but for complicity strategies distribution graphs for the probe trials are placed into the Appendix C.4.

For the analysis the null hypothesis is that there in no difference between any two animal groups (Low, Intermediate, High) over each one of the strategies as well as over the number behavioural transitions within single trials (strategy transitions). P-values lower than 0.05 generated by either the Skillings-Mack or the Friedman test lead us to discard the null hypothesis that the strategies distributions are identical and that any differences are only due to chance (random sampling). These two tests are non-parametric because the data are not normally distributed and because they control the variability among subjects over the different observations [Theodorsson-Norheim, 1987].

In addition to these tests, 95% confidence intervals of a binomial distribution [Wallis, 2013] were also used in the following case; when the High animal group was compared with either the Low or the Intermediate group. Because the High animal group has one animal less from the other two groups, in order to perform the hypothesis testing both groups under comparison had to be equalised thus one animal was excluded from either the Low or the Intermediate group. However, in order to avoid effecting the results on a particular exclusion each animal was excluded in a rotational basis. This process was executed 10 times in total (because Low and Intermediate groups are both having 10 animals while High group is having 9) thus different p-values were generated. Each p-value can indicate significant difference if it is below 0.05 or not significant difference otherwise, thus the process was considered as binomial and it was concluded that the result was significant only if the binomial confidence intervals were well-above 50% indicating that the result was not due to chance [Brown et al., 2001; Peck, 2012]. A similar process has also been used in the study of [Vouros et al., 2018], and here, since the process is executed 10 times, 9 out of 10 p-values need to be below 0.05 in order for the intervals to be well-above 50%.

5.5.3 Results and discussion

The three animal groups were compared and this section provides a report of the percentage of strategies distributions over trials for each experimental procedure. For each figure, green bars correspond to the Low corticosterone response animal group; blue bars correspond to the Intermediate corticosterone response animal

group; red bars correspond to the High corticosterone response animal group. The five experimental procedures, in chronological order as indicated in the Methods sections, are: the *train1* on days 1 to 5 (Figures 5.11, 5.14, 5.17); the *probe1* on day 5; the *probe2* on day 17; the *train2* on day 17 (Figures 5.12, 5.15, 5.18); the *reverseT* on day 18 (Figures 5.13, 5.16, 5.19). Statistical analysis testing was not performed for the probe trials since the Friedman test requires at least three different occasion measurements on each group (for complicity strategies distribution graphs for the probe trials are placed into the Appendix C).

If the path intervals are now the path segments then the strategy percentages are calculated by counting the number of path segments falling under each strategy, for each group and for each trial, multiplied by 100 and divided by the total number of segments. Thus for each figure with regards to each experimental procedure the sum of all the bars of all the behavioural strategy plots will equal to 100. This was done so that the visual inspection on the differences between the two animal groups under comparison can be performed both among the trials for each strategy and among the strategies of the experimental procedure as a whole. The plot for the strategy transitions is separate and illustrates the number of transitions between behavioural strategies on each trial between the two animal groups under comparison.

In order to compare the Low or the Intermediate corticosterone response group with the High corticosterone response group, 1 animal was excluded from the Low or the Intermediate group in rotation and the hypothesis test was executed 10 times. Then the significant difference based on the multiple p-values was assessed using the 95% binomial confidence intervals where a $p - value < 0.05$ indicates that the null hypothesis should be discarded while a $p - value > 0.05$ indicates that it should not. The intervals are required to be clearly 50% in order to discard the null hypothesis that there is no significant difference between the two groups.

Overall, the results of this analysis are as follows:

- During *train1* there are the following significant differences:
    - In favor of the Intermediate group there is more Thigmotactic behaviour than the other two groups (Figure 5.11, Thigmotaxis, $p - value = 0.003$ and Figure 5.17, Thigmotaxis, $p - value < 0.05$) and more Incursion behaviour than the Low group (Figure 5.11, Incursion, $p - value = 0.048$).
    - In favor of the Low group there is more Self Orienting behaviour than the Intermediate group (Figure 5.11, Self Orienting, $p - value = 0.002$)

  These results suggest that the Intermediate animals perform poorly during training when compared with both the Low and the High animals because they are using Low level behavioral strategies. The Low animals are better than the Intermediate since the results suggest that they try to navigate inside the maze using orient themselves based on the surroundings.

- During *train2* there are the following significant differences:
    - In favor of the Intermediate group there is more Direct Finding behaviour than the low group (Figure 5.12, Direct Finding, $p - value = 0.022$) and more Chaining Response than the High group (Figure, Chaining Response, 5.18, $p - value < 0.05$).

- In favor of the Low group there is more Self Orienting behaviour than the other two groups (Figure 5.12, Self Orienting, $p-value < 0.03$ and Figure 5.12, Self Orienting, $p-value < 0.05$) and this group transit between different behaviours more often than the High group (Figure 5.15, Transitions, $p-value < 0.05$).
- In favor of the High group there is more Direct Finding behaviour than the Low group (Figure, Direct Finding, 5.15, $p-value < 0.05$).

These results suggest that Low animals have long-term memory memory deficits and they have to re-learn the task by orienting themselves inside the arena (Self Orienting behaviour). On the other hand, the other two groups directly navigate to the location of the arena (Direct Finding). The Chaining Response behaviour of the Intermediate animals suggest that this group even though it performs better than the Low group it does not learning the exact location of the platform rather its distance from the wall of the arena. Thus these animals swim around this distance in hopes to find to find the platform. On average, the Intermediate group would require more time to reach the platform, but based on performance measurements (refer to *Results: Spatial learning in the water maze* of [Huzard et al., 2019]) it is the fastest group and speed favors a behaviour such as Chaining Response. The High group remembers the exact location and directly navigates to it (Direct Finding). It should be mentioned is that the hypothesis testing did not show any significant difference between the Low and the High animal groups on the Chaining Response strategy even though the High group does not perform any Chaining Response. Probably this is because of the statistical power of the Friedman test which requires more evidence to indicate significant difference.

- During *reverseT* there are the following significant differences:
  - In favor of the Low group there is more Thigmotaxis behaviour than both the other two groups (Figure 5.13, Thigmotaxis, $p-value = 0.038$ and Figure 5.16, Thigmotaxis, $p-value < 0.05$), more Self Orienting and Scanning Target behaviours than the Intermediate group (Figure 5.13, Self Orienting, $p-value = 0.038$ and Figure 5.13, Scanning Target, $p-value = 0.006$) and more Incursion behaviour as well as more often transitions between different strategies than the High group (Figure 5.16, Incursion, $p-value < 0.05$ and Figure 5.16, transitions, $p-value < 0.05$)
  - In favor of the Intermediate group there is more Chaining Response behaviour than both the other two groups (Figure 5.13, Chaining Response, $p-value = 0.025$ and Figure 5.19, Chaining Response, $p-value < 0.05$)

These results further quantify the observations of procedure *train2*. Based on performance measurements (refer to *Results: Reversal learning* of [Huzard et al., 2019]) Low animals spent more time to reach the platform and this is indicated by their behaviours. They cover more distance inside the maze in order to find the platform because they re-learn the task using Thigmotaxis and Self Orienting. After they have memorized the platform location they show better cognitive abilities than the Intermediate group since they actively searching the platform (Scanning Target) while the Intermediate animals having

only memorize the distance between the platform and the walls of the arena they swim around this distance (Chaining Response). Overall both Low and Intermediate groups exhibit lower cognitive abilities in comparison with the High animals who perform less Incursion and behavioural transitions than the Low and less Chaining Responses than the Intermediate animals.

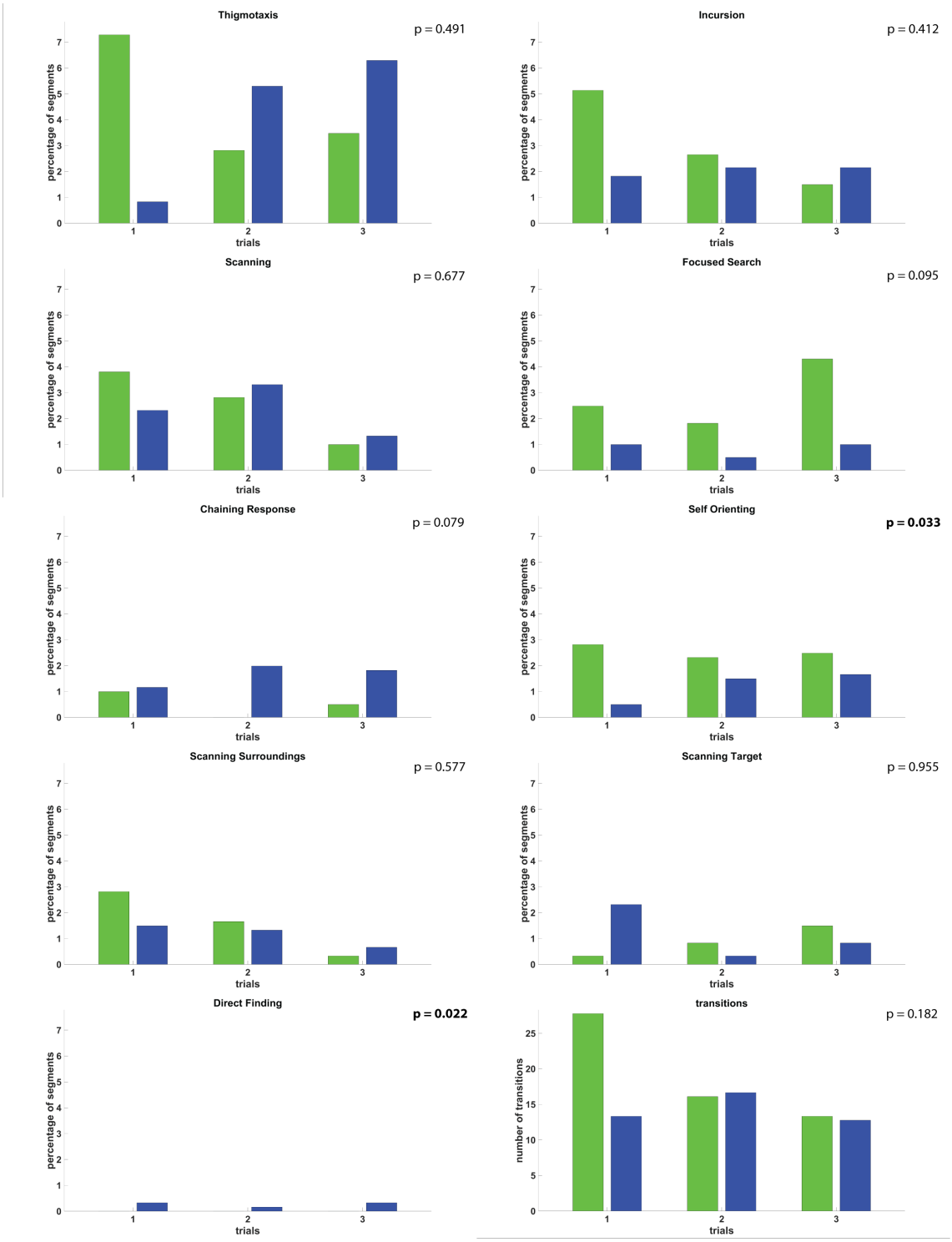


Figure 5.12: *train2* experimental procedure; comparison between Low (green bars) and Intermediate (blue bars) animal groups. Significant differences based on the Friedman test are in Self Orienting (favor of the Low) and Direct Finding (favor of the Intermediate).

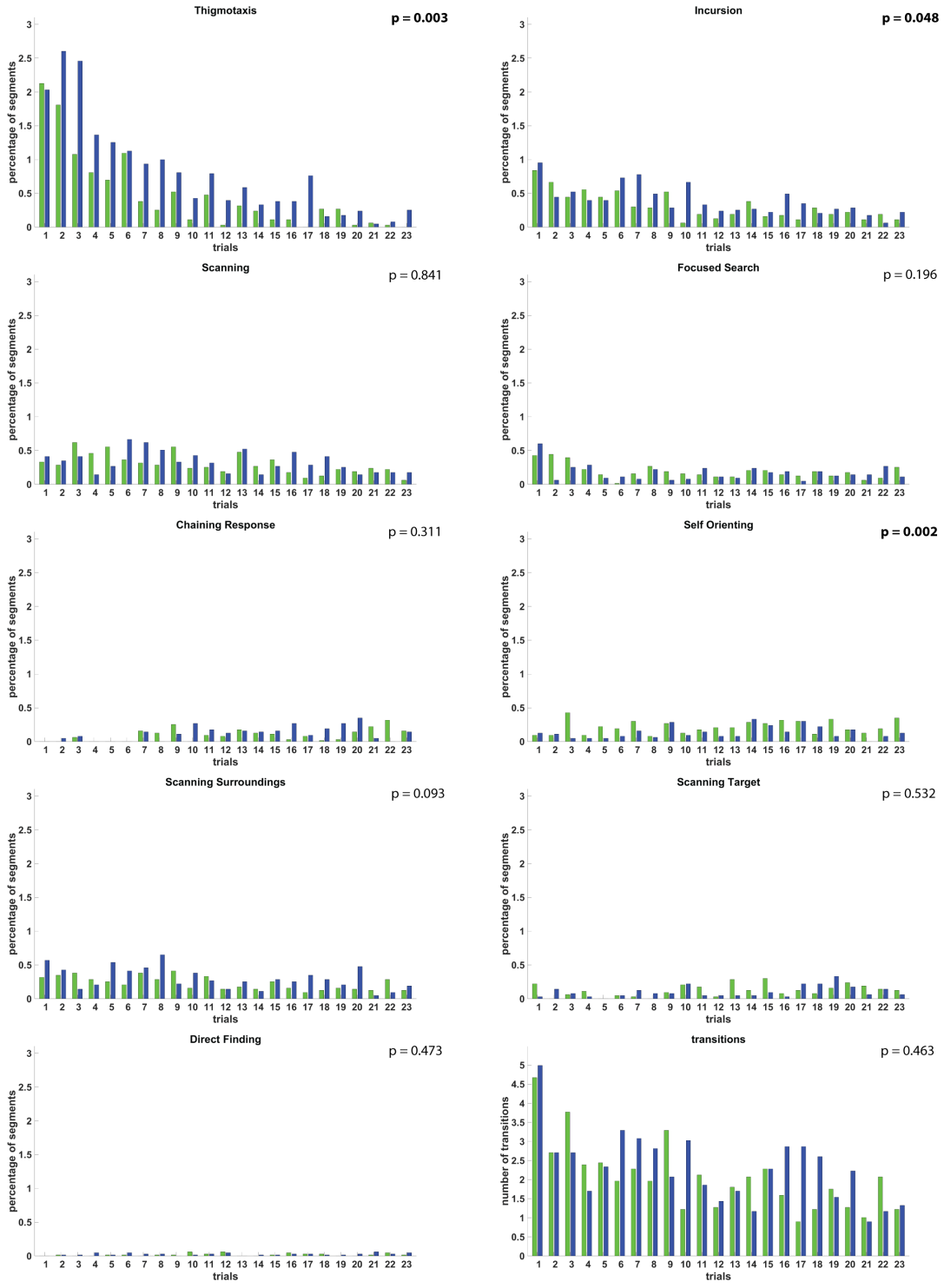


Figure 5.11: *train1* experimental procedure; comparison between Low (green bars) and Intermediate (blue bars) animal groups. Significant differences based on the Skillings-Mack test are in Thigmotaxis (favor of the Intermediate), Incursion (favor of the Intermediate) and Self Orienting (favor of the Low).

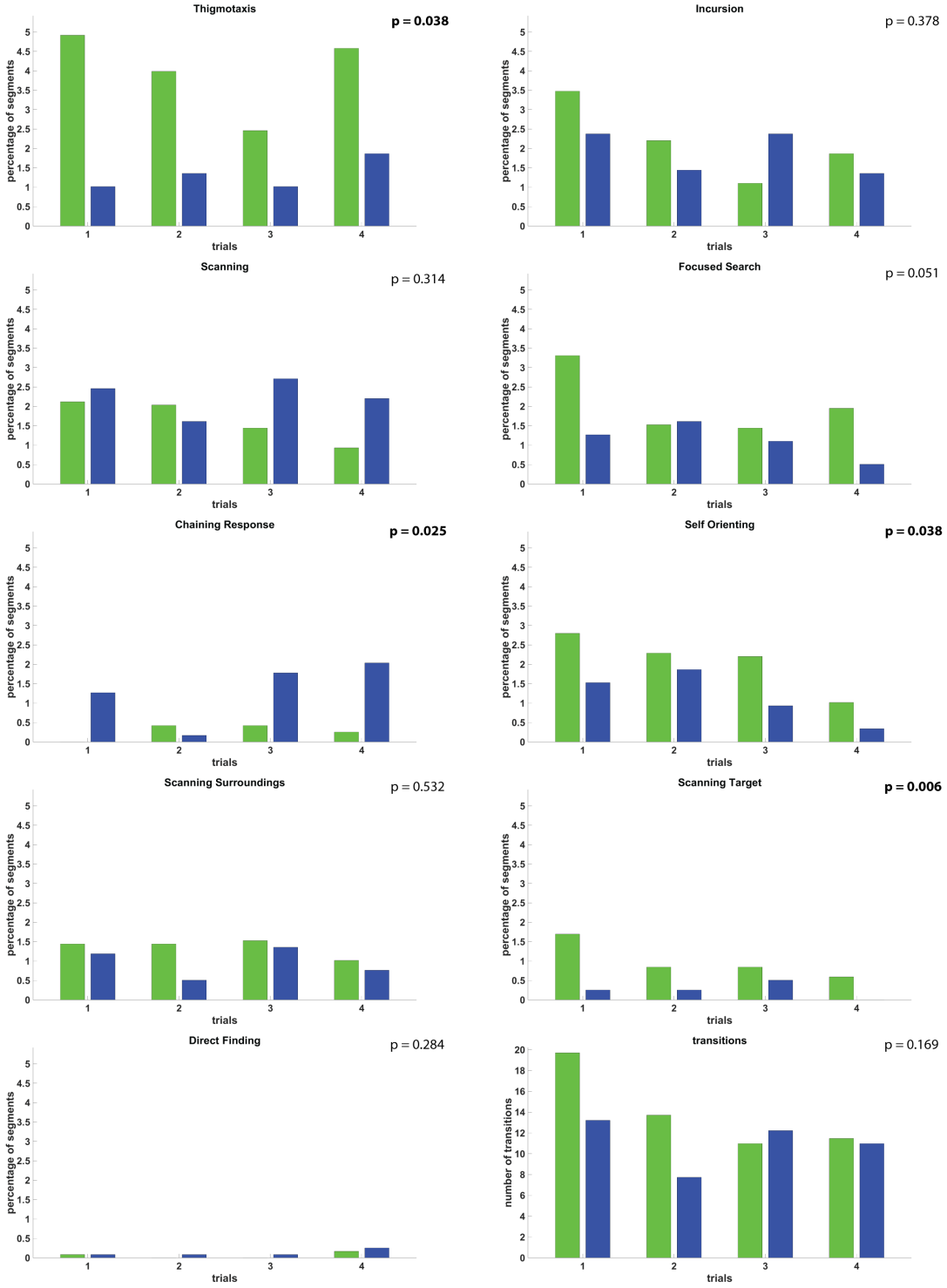


Figure 5.13: *probe1* experimental procedure; comparison between Low (green bars) and Intermediate (blue bars) animal groups. Significant differences based on the Friedman test are in Thigmotaxis (favor of the Low), Self Orienting (favor of the Low), Scanning Target (favor of the Low) and Chaining Response (favor of the Intermediate).

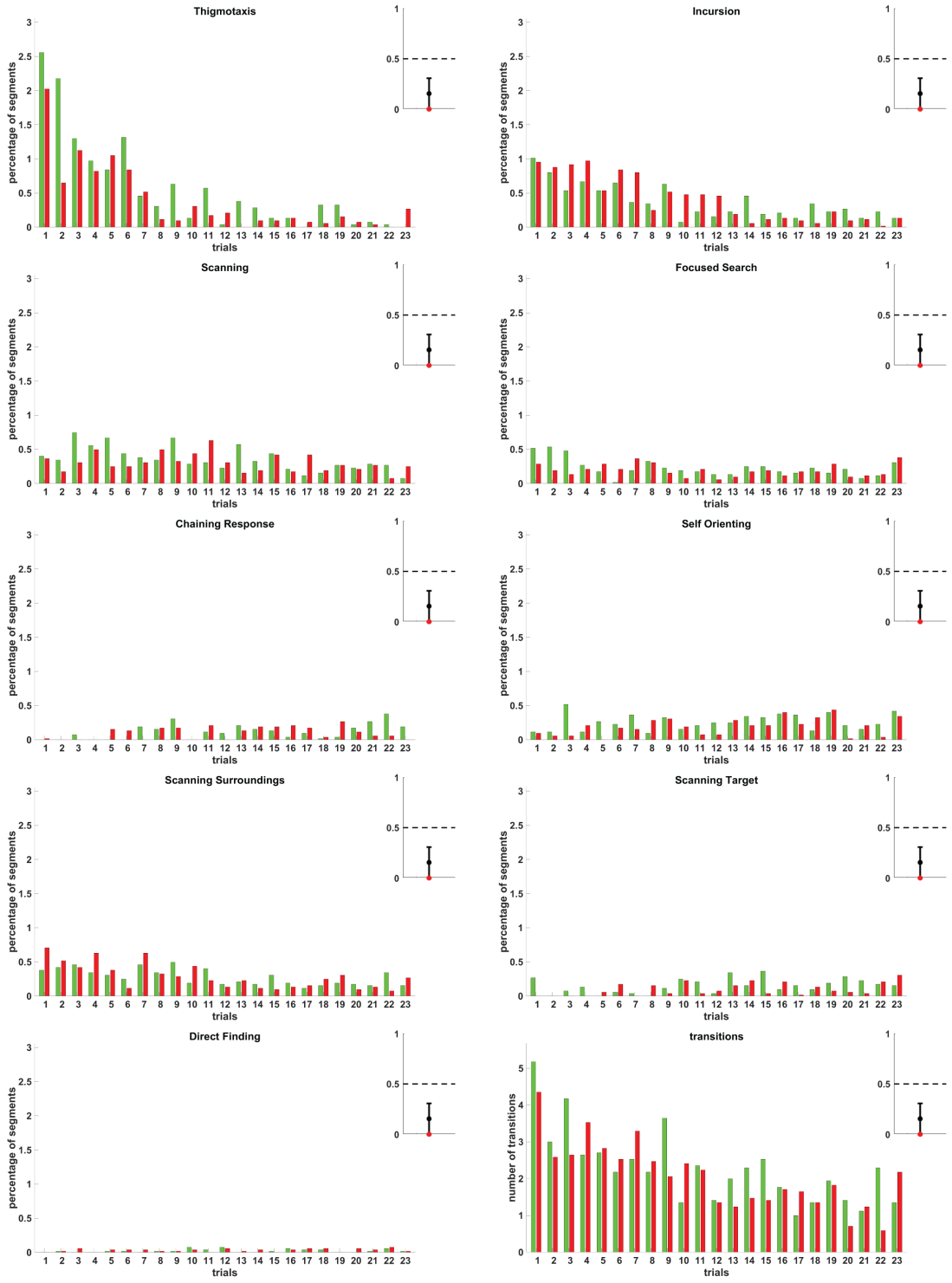


Figure 5.14: *train1* experimental procedure; comparison between Low (green bars) and High (red bars) animal groups. There are no significant differences between the two groups. To equalize the groups, 1 animal was excluded on rotational basis from the Low group and then the Skillings-Mack was executed. On the top right of each plot are the 95% binomial confidence intervals, indicating the number of successes i.e p-values < 0.05 with a red dot. The intervals are required to be clearly above 0.5 in order to drop the null hypothesis that there is no significant difference between the two groups.

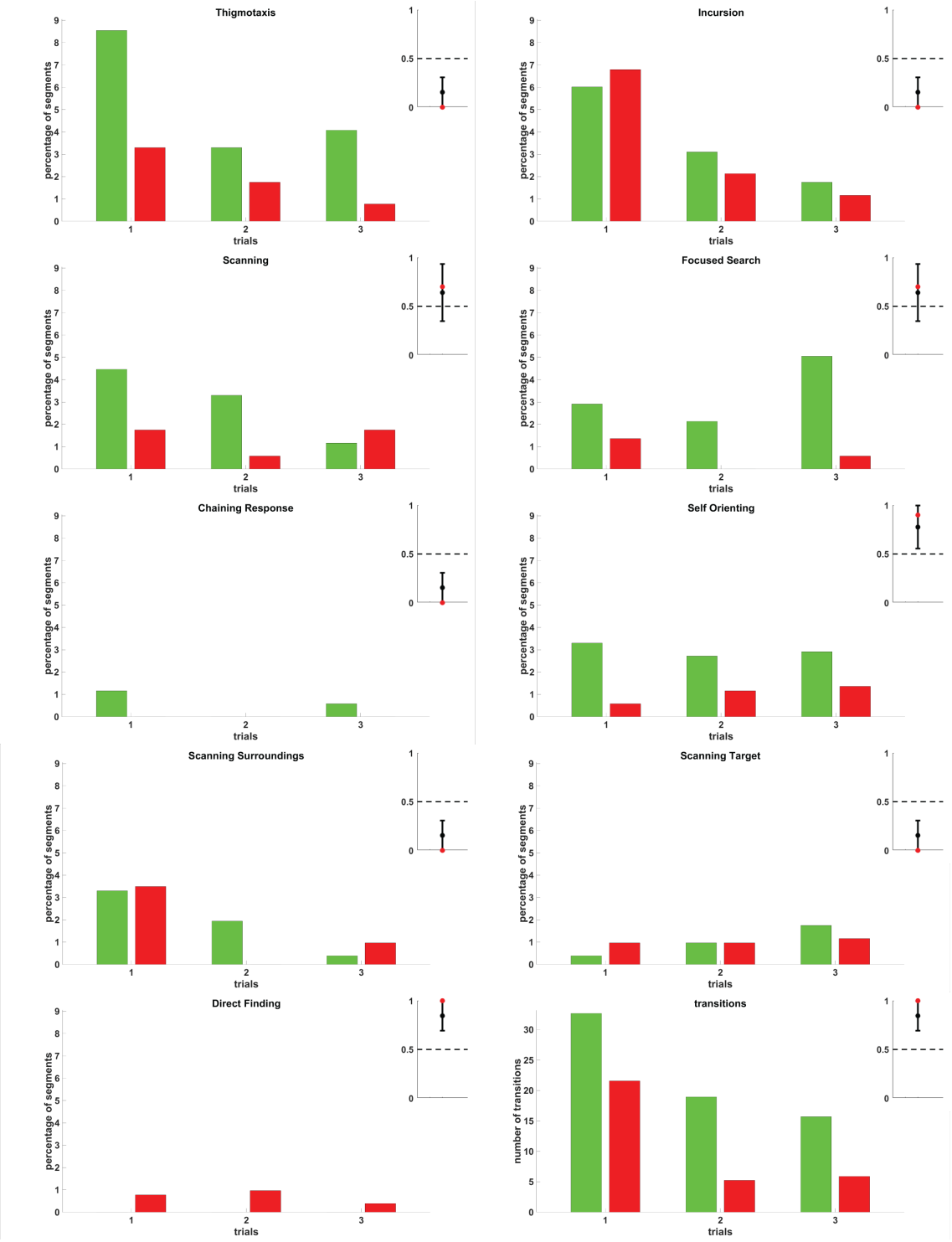


Figure 5.15: *train2* experimental procedure; comparison between Low (green bars) and High (red bars) animal groups. There are significant differences on Self Orienting (favor of the Low), Direct Finding (favor of the High) and transitions (favor of the Low). To equalize the groups, 1 animal was excluded on rotational basis from the Low group and then the Friedman test was executed. On the top right of each plot are the 95% binomial confidence intervals, indicating the number of successes i.e p-values < 0.05 with a red dot. The intervals is required to be clearly above 0.5 in order to drop the null hypothesis that there is no significant difference between the two groups.

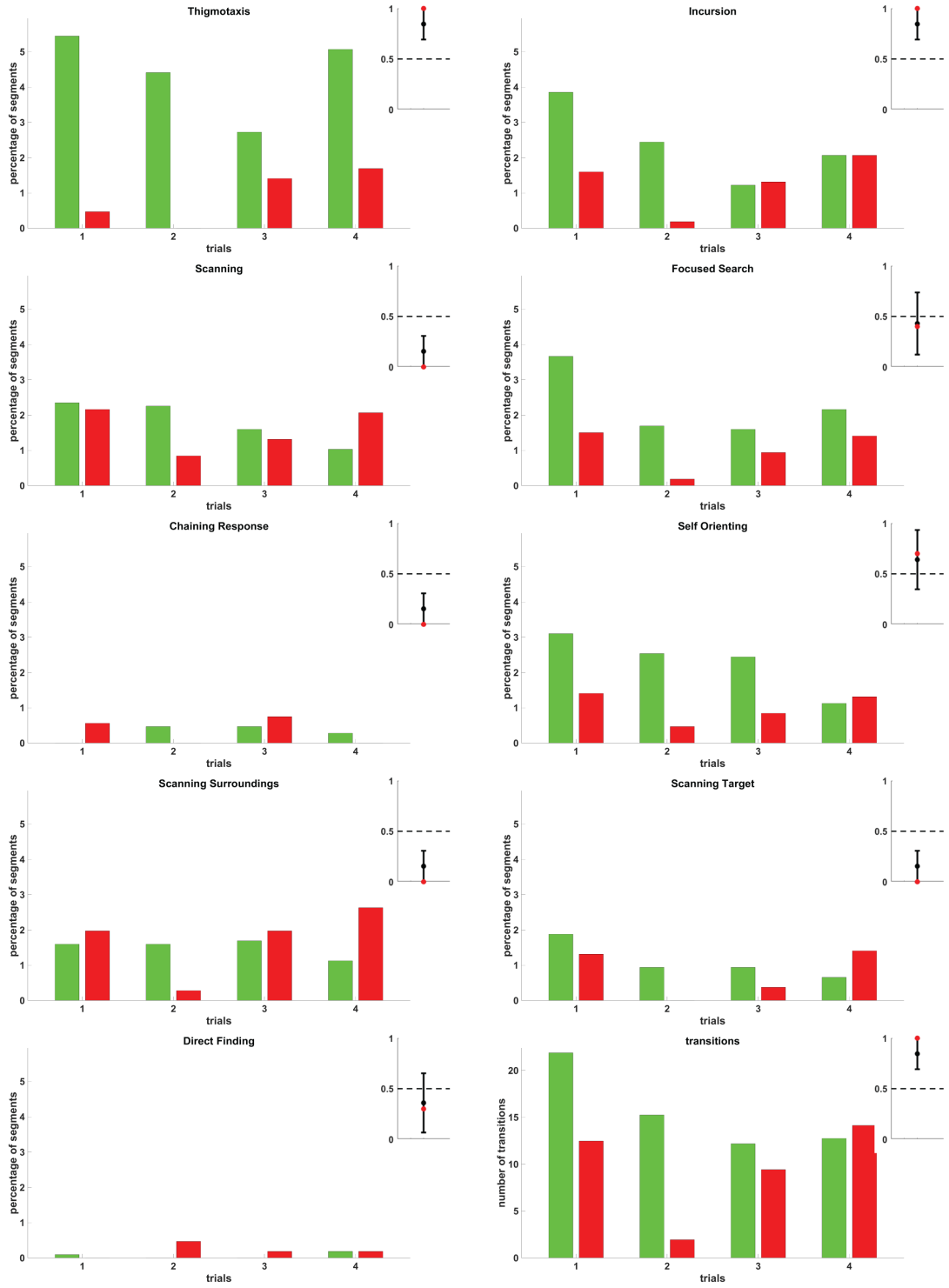


Figure 5.16: *reverseT* experimental procedure; comparison between Low (green bars) and High (red bars) animal groups. There are significant differences on Thigmotaxis, Incursion and transitions (all in favor of the Low). To equalize the groups, 1 animal was excluded on rotational basis from the Low group and then the Friedman test was executed. On the top right of each plot are the 95% binomial confidence intervals, indicating the number of successes i.e p-values < 0.05 with a red dot. The intervals is required to be clearly above 0.5 in order to drop the null hypothesis that there is no significant difference between the two groups.

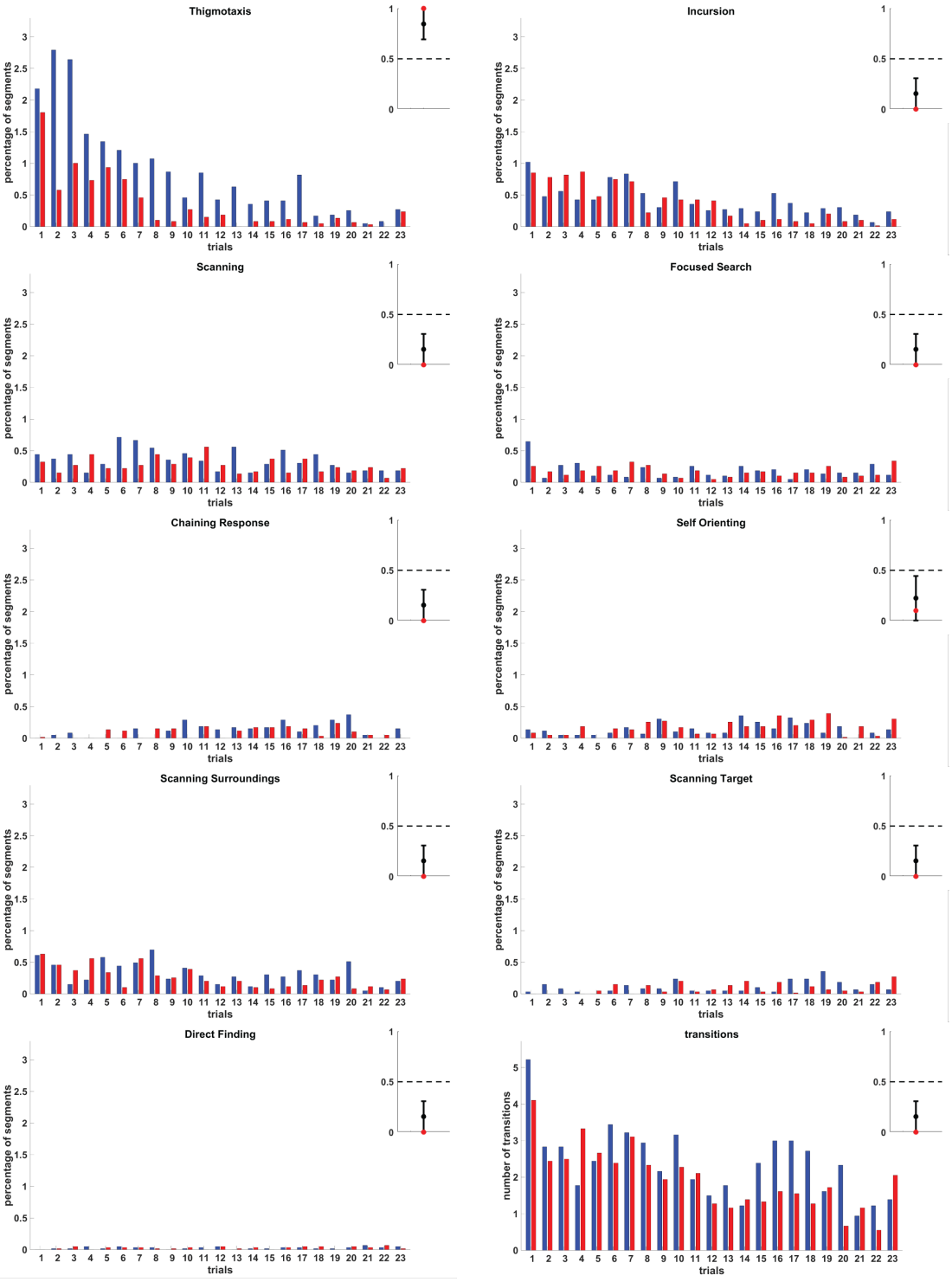


Figure 5.17: *train1* experimental procedure; comparison between Intermediate (blue bars) and High (red bars) animal groups. There is a significant difference on Thigmotaxis (favor of the Intermediate). To equalize the groups, 1 animal was excluded on rotational basis from the Low group and then the Friedman test was executed. On the top right of each plot are the 95% binomial confidence intervals, indicating the number of successes i.e p-values < 0.05 with a red dot. The intervals is required to be clearly above 0.5 in order to drop the null hypothesis that there is no significant difference between the two groups.

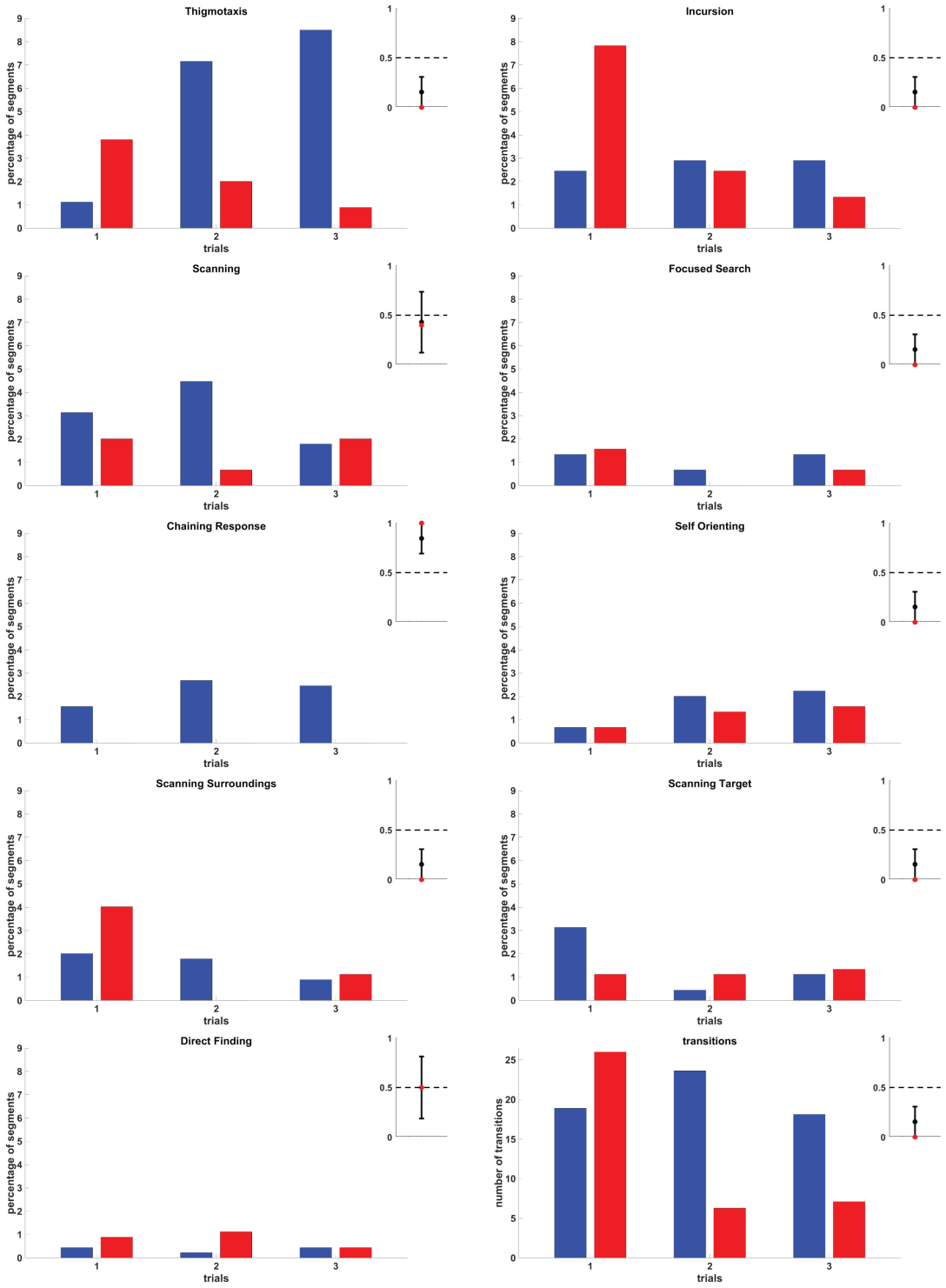


Figure 5.18: *train2* experimental procedure; comparison between Intermediate (blue bars) and High (red bars) animal groups. There is a significant difference on Chaining Response (favor of the Intermediate). To equalize the groups, 1 animal was excluded on rotational basis from the Low group and then the Friedman test was executed. On the top right of each plot are the 95% binomial confidence intervals, indicating the number of successes i.e p-values < 0.05 with a red dot. The intervals is required to be clearly above 0.5 in order to drop the null hypothesis that there is no significant difference between the two groups.

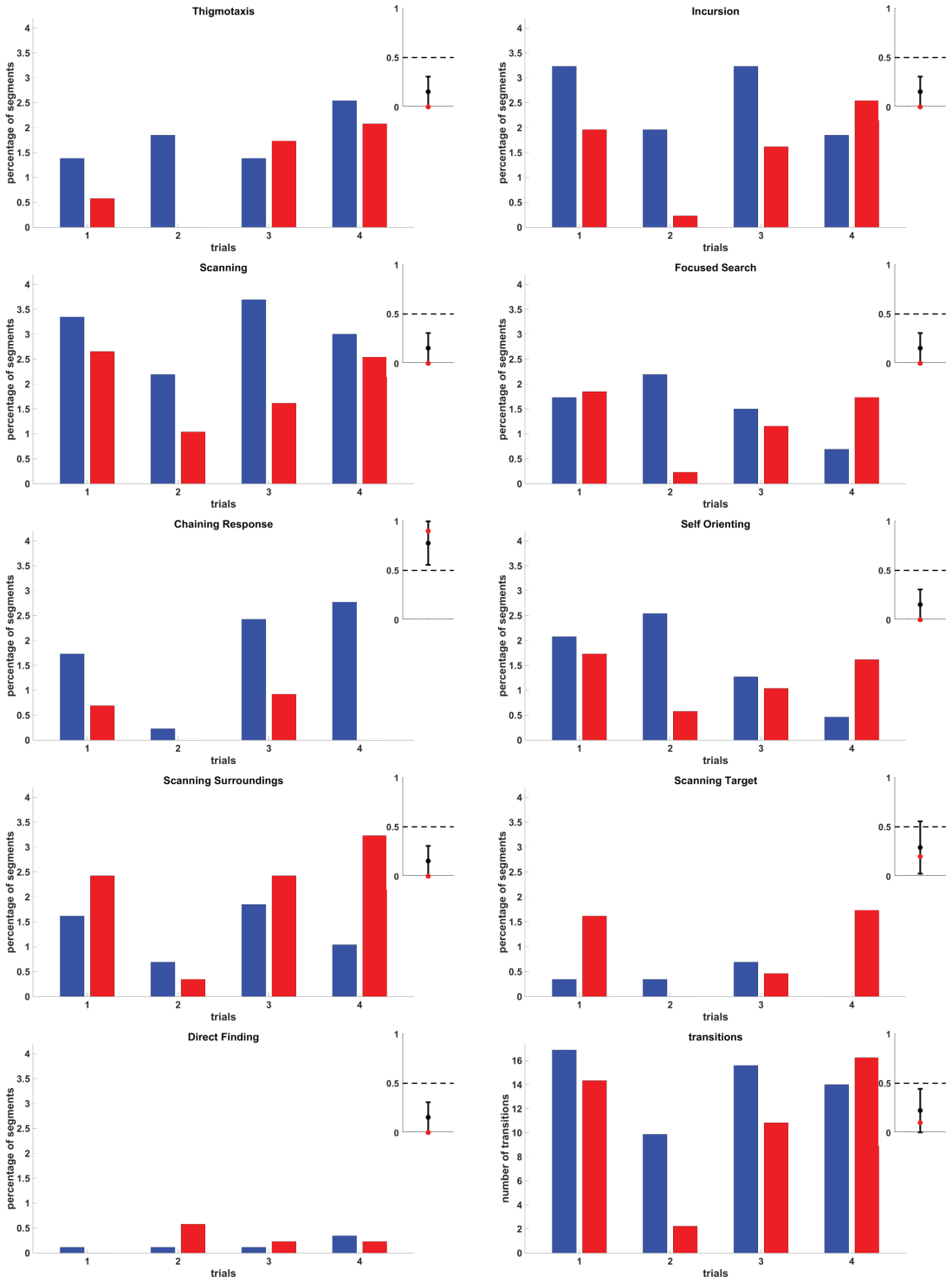


Figure 5.19: *reverseT* experimental procedure; comparison between Intermediate (blue bars) and High (red bars) animal groups. There is a significant difference on Chaining Response (favor of the Intermediate). To equalize the groups, 1 animal was excluded on rotational basis from the Low group and then the Friedman test was executed. On the top right of each plot are the 95% binomial confidence intervals, indicating the number of successes i.e p-values < 0.05 with a red dot. The intervals is required to be clearly above 0.5 in order to drop the null hypothesis that there is no significant difference between the two groups.

5.5.4 Conclusions

Three groups of male rats with different levels of corticosterone response, Low, Intermediate and High were compared. The comparison was performed based on the different behavioural strategies that the animals were implementing in order to solve the Morris Water Maze task. Firstly, the RODA software [Vouros et al., 2017, 2018] was used to perform detailed classification of the animal paths inside the maze leading to the generation of path segments, classified under 9 different behaviours. Afterwards, these segments were used to quantify behavioural differences between the three groups using statistical analysis based on hypothesis testing. The statistical analysis was validated based on the one performed by Huzard et al. [Huzard et al., 2019] as well as with performance measurements performed in the aforementioned study.

- The Low group animals utilizes more consistent learning by gradually evolving their behaviours. The dominant strategy of the Low animals is the Self Orienting which allows them to orient themselves inside the maze and gradually to evolve their behavioural strategies from Low level (Thigmotaxis and Incursion) to medium (Focused Search, Scanning) and then High level (Scanning Target). However, this animal group suffers from long-term memory deficits and it requires to re-learn the location of the platform as we observe from its pattern of behaviours during the different experimental procedures (days 17-18). Nevertheless, once the location has again been memorized this group shows improved performance and implements High level strategies (Scanning Target) to navigate to the location of the platform.

- The Intermediate group utilizes weak learning (Thigmotaxis) and ultimately it adapts to less intelligent behavioural strategies (Chaining Response). The dominant strategy of the Intermediate animals is the Chaining Response which means that these animals have memorized the distance from the walls of the arena to the platform and swim around this distance hoping to find the platform. Evidence suggests that these animals have better long-term memory during latter trials (day 5 and days 17-18) even though less effective learning. Nevertheless, these animals have been adapted to a behaviour (Chaining Response) which is beneficial only because the platform was located on the same distance from the wall during all the procedures, excluding the probe trials, and since they are significantly faster than Low and High animals they show improved performance measurements during long-term memory procedures. The Low animals, even through they are slower in both learning and speed they have developed better behavioural strategies and this is shown during the reversal trials where they implemented High level strategies (Scanning Target) to find the platform while the Intermediate animals remain focused on the Chaining Response strategy which implies no cognitive behaviour.

- The High group is supported by enough evidence to be have the best performers. High animals essentially follow the same behavioural strategies evolution of the Low group but they learn faster and at the same time they maintain better memory of the platform location.

Chapter 6

A semi-supervised algorithm with feature selection mechanism for the Morris Water Maze task

This Chapter builds on Chapter 5 and replaces the MPCK-Means component of the RODA framework [Vouros et al., 2017, 2018] with the PCSK-Means algorithm presented in Chapter 3. The aim is to add a feature selection capability in the framework which would allow to in-detailed study of the features used in the classification task.

6.1 Introduction

As described in Chapter 5, to analyse rodents trajectories from the Morris Water Maze (MWM) experiments the initial animal paths were split into segments of a certain length and a percentage of overlap. A set of eight features is then computed for each segment and a percentage of data is then manually labeled. Finally a classification framework based on MPCK-Means and classification boosting is using the features and the partially labeled data to classify the rest.

An extension of the previously described pipeline is to also provide information about the effect of each feature on the classification task i.e. how much each feature contributed to the separation of the classes. Since the features are engineered to have a biological interpretation, such information can be of importance and favor further researched on targeted subset of features. In the study of [Chhabria et al., 2019] (refer to Chapter 4) such work had to be done manually in order to link features to animal behaviours.

In this study the classification framework described of [Vouros et al., 2018] (refer also to Chapter 5) will be modified in order to incorporate the Pairwise Constrained Sparse K-Means (PCSK-Means) algorithm described and tested in [Vouros and Vasilaki, 2020] (refer also to Chapter 3). As it is shown in the aforementioned study, PCSK-Means has a feature selection mechanism and can be used to extract information about the importance of each feature. It will be shown that by changing only the semi-supervised algorithm the framework for detailed classification of swimming paths inside the Morris Water Maze would then be capable of outputting information about the importance of each feature.

6.2 Methods

6.2.1 Two-stage classification using PCSK-Means algorithm

The same two-stage classification framework with majority boosting as the one described in Chapter 5 (see also the published work of [Vouros et al., 2018]) was used for this study. The only difference is that the clustering was performed using the PCSK-Means algorithm (see 3.2.2.1) instead of the MPCK-Means (see 2.2.4.2). The algorithm was initialised using the seeding method (see 2.2.7.7) as was the case with the MPCK-Means algorithm since based on the benchmark of Chapter 3 PCSK-Means is robust to the initialisation method.

6.2.2 PCSK-Means algorithm tuning

The PCSK-Means algorithm requires the tuning of two parameters the target number of clusters K and the sparsity S. The first parameter is auto-tunable based on the majority voting method implemented and described in Chapter 5 (see also [Vouros et al., 2018]) under which different numbers of target clusters are tested and the ones that result to the best classifiers based on the 10-fold cross validation procedure are adapted and form an ensemble. The results of the selected classifiers and the ensemble are then used to form the final analysis conclusions.

For the tuning of the second parameter S, which specifies how many features will receive 0 weight (amount of sparseness), the studies of [Brodinová et al., 2017; Witten and Tibshirani, 2010] propose a modification of the gap statistic [Tibshirani et al., 2001] (see section 2.2.8.1.3 for more information). Another option would have been the 10-fold cross validation across different values of S. However, both of these methods require a lot of computations thus for this study the silhouette method (see 2.2.8.1.2) was used to identify an appropriate value for S between the interval $[1 \ \sqrt{p}]$ (p is the dimensionality of the data set) with a step of 0.2 (for an explanation of these specific bound of S refer to Appendix B). Using the silhouette method did not cause any difference in the classification results which match that of the original study described in Chapter 5.

6.2.3 Morris Water Maze data properties

The data set used in this study is one of the sets that contain segmented animal trajectories from two animal groups, stress and control, in the Morris Water Maze procedure from the study of [Vouros et al., 2018] (refer also to Chapter 5). Specifically it is the segmentation with segment length of 250cm (2.5 times the arena radius) and overlap 70% (refer to Table 5.1). It contains 10388 data points 1261 of which are labeled and 8 behavioural classes. Apart from the 8 original features computed for each trajectory segment (named *Median distance to center*, *IQR distance to center*, *Focus*, *Central displacement*, *Inner radius variation*, *Target proximity*, *Eccentricity*, *Maximum loop length*; refer to Table 5.2 for more information) a 9^{th} feature was added in the data set which is the *Segment length*. This feature is known to be uninformative since all the segments have approximately the same length. Random permutations of this feature were also generated (features 10^{th} to 13^{th}) and added to the data set resulted to a total of 13 features.

6.3 Results

A comparison between the conclusions of the study described in Chapter 5 using the original classification of RODA [Vouros et al., 2017, 2018] and the proposed one is performed. Some differences are visible among the two classifications (refer to Table 6.1) but based on the comparison in Figure 6.1 both frameworks result to the same conclusions.

	PCSK Classification		**MPCK Classification**	
Number of generated classifiers	64		78	
	Classifiers (average)	**Ensemble**	**Classifiers (average)**	**Ensemble**
Thigmotaxis	23.7%	24.0%	23.0%	24.0%
Incursion	18.9%	18.2%	19.3%	18.9%
Scanning	11.8%	12.2%	12.2%	12.3%
Focused Search	6.3%	7.4%	7.4%	8.9%
Chaining Response	9.0%	8.0%	7.8%	5.8%
Self Orienting	7.4%	8.5%	7.6%	8.8%
Scanning Surroundings	14.6%	16.6%	14.8%	15.8%
Target Scanning	5.0%	5.1%	5.4%	5.6%

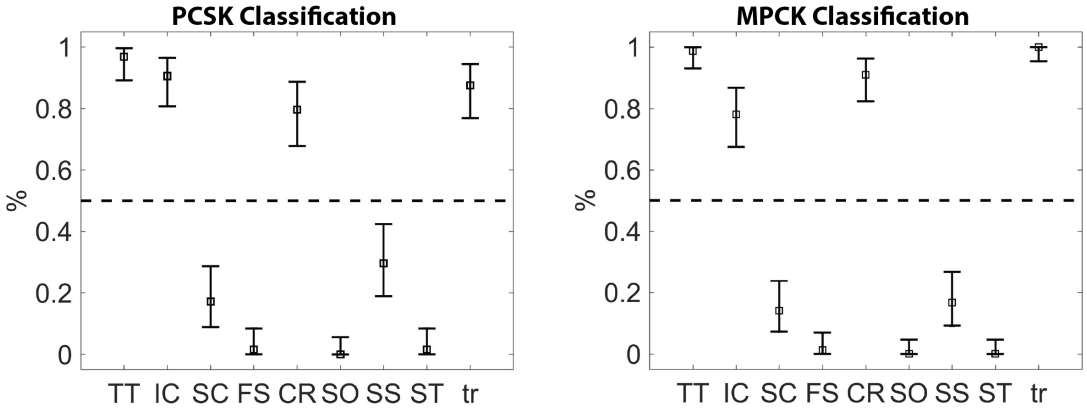
Table 6.1: Percentage of segments falling under each class for the PCSK and MPCK classification frameworks. For each classification (the original using the MPCK-Means algorithm, refer to Chapter 5 and the proposed using the PCSK-Means algorithm) the number of generated classifiers to form an ensemble is presented. The statistics for each class are shown separately for the classifiers (average) and the ensemble. Some differences between the classifiers and the ensemble as well as the two classification procedures are visible but based on the results of Figure 6.1 consistency on the conclusions is preserved.

Afterwards, the capabilities of the new framework are assessed by studying the feature weights assigned after the classification procedure. It should be noted that MPCK-Means cannot be used for feature assessment as shown in the experimental work of Chapter 3, rather it learns a metric that best fit the data to the constraints. Table 6.2 presents the weight of each feature based on the classifiers that were used to form the ensemble. Based on the results, the 9^{th} to 13^{th} features have correctly been identified as uninformative. For the rest of the features, which are also used in the previous study of [Gehring et al., 2015], the *Inner radius variation*, *Target proximity* and *Eccentricity* are also having low feature weights.

6.4 Discussion

In this study a potential update for the RODA framework [Vouros et al., 2017, 2018] is proposed and tested. This new framework has feature selection and assessment capabilities to provide more information about the underlying value of each feature used during the classification of the animal trajectory segments inside the Morris Water Maze experimental procedure.

Based on the results (see Table 6.2 and Figure 6.1) the new framework does not alter the conclusions of the previous study (refer to Chapter 5 and the published study



Ensemble Result (Friedman test p-values per strategy and transitions)									
	TT	**IC**	**SC**	**FS**	**CR**	**SO**	**SS**	**ST**	**tr**
PCSK	**0.014**	**0.012**	0.104	0.692	**0.018**	0.604	0.054	0.528	**0.009**
MPCK	**0.005**	**0.013**	0.157	0.278	**0.003**	0.638	0.190	0.345	**0.019**

Figure 6.1: Comparison between MPCK-Means (MPCK) and PCSK-Means (PCSK) on the conclusive results from the classification of the data set. Each plot shows the 95% binomial confidence intervals for the classifiers of each segmentation regarding their agreement on the significant difference between the two animal groups of the data set for each strategy and the strategy transitions. Squares indicate the mean of the classifiers; errorbars represent the 95% confidence intervals; the dashed line indicates the threshold of interest (0.5 or 50%). Confidence intervals clearly above 0.5 (or 50%) confirm that there is indeed a significant difference between the the two animal groups on the strategies and the strategy transitions. The table below the plots shows the Friedman test p-values for the classification result of the ensemble for each classification (PCSK and MPCK). Each element of the table has the relevant p-value and gray-marked cells indicate significant difference, i.e. p-value < 0.05. Abbreviations: Thigmotaxis (TT), Incursion (IC), Scanning (SC), Focused Search (FS), Chaining Response (CR), Self Orienting (SO), Scanning Surroundings (SS), Target Scanning (ST), Strategy Transitions (tr) (refer to section 5.2.9 for more information on each behavioural strategy). We see that using the PCSK-Means instead of the MPCK-Means algorithm in the classification framework of RODA [Vouros et al., 2017, 2018] (refer to Chapter 5, section 5.2.3) does not alter the conclusions of the study.

Feature name	Weight value			
	min	**max**	**mean**	**var**
Median distance to center	0.32	0.37	0.36	0.01
IQR distance to center	0.41	0.45	0.44	0.01
Focus	0.56	0.61	0.58	0.01
Central displacement	0.23	0.30	0.29	0.01
Inner radius variation	0.02	0.07	0.05	0.01
Target proximity	0.04	0.18	0.10	0.05
Eccentricity	0	0.08	0.04	0.02
Maximum loop length	0.48	0.54	0.49	0.01
Segment length	*0*	*0.01*	*0*	*0*
Segment length (permutation)	*0*	*0.01*	*0*	*0*
Segment length (permutation)	*0*	*0.01*	*0*	*0*
Segment length (permutation)	*0*	*0.01*	*0*	*0*
Segment length (permutation)	*0*	*0.01*	*0*	*0*

Table 6.2: Feature weight values for the Morris Water Maze. The Table shows the weight of each feature used in the Morris Water Maze studies of [Gehring et al., 2015] and [Vouros et al., 2018] as assigned by the new classification framework proposed in this study using all the classifiers that formed the ensemble. Five extra uninformative feature were added which are the length of each segment (Segment length) and four random permutations of this feature. The uninformative feature has correctly being assigned the minimum weight value. Other features that also have been assigned with low weight values are marked in gray.

[Vouros et al., 2018]) and it is able to identify correctly the known uninformative feature of segment length among its permutations by assigning a weight of 0 to them (see Table 6.2, Segment length and permutations). In the current work the features: *Inner radius variation*, *Target proximity* and *Eccentricity* used in the studies of [Gehring et al., 2015; Vouros et al., 2018] have also been identified to be uninformative based on their low weight value.

Eccentricity, which was also used in the study of [Chhabria et al., 2019] (refer to Chapter 4), was expected to capture long straight paths, when the animal traverses through the arena without being focused to certain areas. *Target proximity* was expected to capture behaviours related to the platform, when the animal searches or crosses close to it. *Inner radius variation* is a more abstract feature but the expectation was to capture differences between focal and eccentric paths since the more focal to an area the path is (thus the minimum enclosing ellipsoid to the path would tend to be circular) the more this feature value tends to 0.

Nevertheless, the remaining features are enough to differentiate path aspects described by features indicated as informative by the algorithm: *distance to center* metrics and *central displacement* can be used to indicate thigmotactic behaviours; *focus* can be used to indicate when the animal searches particular areas of the arena. These features are enough to differentiate strategies like Thigmotaxis, Incursion, Scanning, Focused Search and Chaining. Given also the partial labeling, platform related strategies like Scanning Target and Scanning Surrounding can also be identified. Finally *maximum loop length* is identified as important since it is the only feature capturing patterns indicating Self Orientated behaviours.

Chapter 7

Conclusions and future work

7.1 PhD contribution

Behavioural neuroscience uses a variety of experimental procedures involving locomotion. Analysing animal paths inside constrained environments can provide valuable information about how certain factors such as diseases and drugs can alter the natural behaviour and how such alternations affect certain brain regions and brain functionalities such as the development of learning and memory. Over the years different machine learning methods have been designed to automate the process of drawing intelligence from animal data involving behavioural task but such methods are experiment-specific and require domain-specific machine learning knowledge in order to be deployed and used. This PhD work is focused on integrating different analysis techniques together in a unified machine learning framework in order to design pipelines capable of extracting as much information as possible from any experimental procedure. The researched and engineered machine learning methods are built around pre-existing knowledge of animal behaviours to nullify the need of domain-specific machine learning knowledge. When such knowledge is not available then application is still possible in order to yield information about which aspects of the animal pathing are important thus direct further research towards them. In general, this PhD contributions are directed towards the fields of machine learning and behavioural neuroscience.

More specifically, this PhD study contributed to field of machine learning in the following ways (Chapter 3, [Vouros et al., 2019; Vouros and Vasilaki, 2020]):

1. There has been an extended benchmark study on unsupervised and semi-supervised K-Means initialisation methods and variations with detailed listing of their performance. This study includes:
   (a) the K-Means initialisation methods: Random, K-Means++, Maximin, Kaufman, ROBIN and DK-Means++;
   (b) the K-Means unsupervised variations: Lloyd's K-Means, Hartigan-Wong's K-Means, K-Medians and Weiszfeld's algorithm;
   (c) Sparse K-Means and the semi-supervised variations of Pairwise Constrained K-Means and Metric Pairwise Constrained K-Means which are initialised by both the unsupervised DK-Means++ and the semi-supervised Seeding methods;

and showed that, regardless of the K-Means clustering algorithm used, DK-Means++ can achieve the best performance on average compared with the other methods, while if exhaustive search is possible K-Means++ can achieve the best performance. K-Medians is, marginally, the best K-Means variation followed by Hartigan-Wong's K-Means.

2. A semi-supervised K-Means algorithm named Pairwise Constrained Sparse K-Means (PCSK-Means) was engineered. This algorithm brings together Sparse K-Means clustering and Pairwise Constrained K-Means clustering with the goal to achieve automatic feature selection and assessment and increased classification performance when data labels are available. Experimental work showed that the feature selection mechanism is unaffected by the type of constraints, number of constraints or initialisation method used. For classification tasks PCSK-Means can achieve better performance that the unsupervised Sparse K-Means and almost the same performance of other semi-supervised K-Means algorithms (MPCK-Means). This makes it an ideal candidate in scenarios when feature selection is important and should not detriment classification performance.

In the neuroscience field there has been research and collection of universal path features that can analyse animal paths in a constrained open area regardless of the animal subject or the experimental procedure. Using such features behavioural information can be extracted manually or semi-automatically with the use of a framework (RODA) that requires few examples of animal behaviours to be given as input and then classifies the rest listing also information about the importance of each feature during the classification task. Application of manual features analysis in the light/dark preference task using zebrafish larva animal models for diabetes (Chapter 4, [Chhabria et al., 2019]) identified that:

1. Hyperglycemia results in an increase to both exploration and thigmotactic behaviours of the subjects.

2. Such behaviours are return to normal with SNP treatment.

3. SNP treatment does not cause any behavioural alternations to non-diabetic subjects.

Application of the RODA framework [Vouros et al., 2018] and software [Vouros et al., 2017] on Morris Water Maze studies in rodents aiming to explore the effects of stress in learning and memory (Chapters 5 [Huzard et al., 2019; Vouros et al., 2018] and 6) showed that:

1. Stressed animals exhibit impaired learning and utilise low level strategies (Thigmotaxis, Incursion) for a longer duration than control animals. They adapt to sub-optimal strategies (Chaining response) to solve the task. They also transit between different behaviours more often than control animals which is not a beneficial learning process.

2. Corticosterone response level affects learning and memory in the following ways:

   (a) Low corticosterone response levels result to consistent but slow learning accompanied with long-term memory deficits.

(b) High corticosterone response levels result to fast learning and reduce the long-term memory deficits.

(c) Intermediate levels of corticosterone response result to the adaptation of sub-optimal behaviours (Chaining response) which benefit from the increased animal speed.

3. Identifying the sub-optimal strategy of memorising the distance from the walls of the arena to the hidden platform and navigating in a circular path passing through the hidden platform (Chaining response) is not possible using traditional metrics such as the amount of time to solve the task or other analysis methods that classify the whole animal path during a trial to one stereotypical behaviour. The proposed framework though, is able to distinguish different strategies during each trial resulting to the identification of the Chaining response strategy and its disadvantages in learning.

4. Only a subset of features is important for classifying the different animal behaviours inside the Morris Water Maze task.

7.2 Disadvantages, limitations and future work

The work presented in this PhD study aims to design and propose ways of extracting detailed behavioural information from experimental procedures involving navigational tasks. To achieve this aim, a segmentation methodology of the animal paths is proposed. This methodology is first reported in the study of [Gehring et al., 2015] and further researched and improved (refer to Chapter 5 and the publication of [Vouros et al., 2018]). The segmentation is based on overlapping, where an animal path is separated into pieces which overlap with a certain percentage. A disadvantage of this method is the generation of a large amount of data which are used as an internal step to extract useful information. One could argue that a certain feature could be used to segment the data without the overlapping procedure to reduce the computational processing as in the study of [Gehring et al., 2017]. However, such feature is difficult to be detected and it would be subjected to a specific analysis, temporal (e.g. latency of performing a specific action), positional (e.g. entrances/exits to/from specific areas of the experimental arena) or spatiotemporal (e.g. sudden changes of speed) and might be affected from aspects such as the recording frequency and resolution [Benhamou, 2014]. In the field of animal movement ecology there are various studies regarding segmentation and behavioural analysis of animals pathing [Barraquand and Benhamou, 2008; Edelhoff et al., 2016; Thiebault and Tremblay, 2013] based on various criteria but the translation of such methods in the constrained environment of experimental procedures is not straight forward. In the latter cases, conclusions are drawn from the full animal pathing during a specific trial (e.g. in the Morris Water Maze a trial ends when the animal reaches the hidden platform [Morris, 1984]) or specific aspects of the experimental arena (e.g. in the light/dark task the amount of pathing in the light area is separated from the one in the dark area [Maximino et al., 2012]).

Other important points are the assumptions on the data quality and the specifications of the experimental procedure. All the features and the methods presented on this dissertation were designed based on experiments involving navigation tasks

inside constrained open arenas such as the Morris Water Maze [Morris, 1984], the light/dark preference task [Araujo et al., 2012] and the Allothetic Place Avoidance task [Stuchlik et al., 2004]. Such procedures exclude constrained mazes [Olton, 1979] and selection-based arenas such as the T-maze [Graeff et al., 1998], the Arm Radial maze [Juraska et al., 1984] and the double-H maze [Pol-Bodetto et al., 2011]. Furthermore, for the Morris Water Maze studies the features that were engineered and fed to the RODA framework [Vouros et al., 2017] are based on spatial or positional aspects of the animal paths. More features may be considered that take into consideration timing information such as speed, acceleration, duration of inactivity or duration until sudden changes of the animal directionality. In addition, the features were designed for 2-D animal paths which are composed of a series of coordinates extracted from tracking software such as Ethovision [Noldus et al., 2001]. These paths are assumed to be continues and smoothed. A potential future work would be to investigate the minimum sampling rate requirements for the features in order for the latter to be informative; this can greatly reduce the feature computation workload which, throughout this PhD study, is executed per sampling point.

In the machine learning domain, a disadvantage of the classification framework presented in Chapter 5 and extented in Chapter 6 for the Morris Water Maze procedure (refer to [Vouros et al., 2018]) is that it does not aim to create a generic classification procedure but a specific one based on the data under analysis. This is due to the fact that an amount of labels needs to be provided for every new data set. Compared with other studies [Cooke et al., 2019; Illouz, Madar, Louzon, Griffioen and Okun, 2016; Wolfer et al., 2001] that do automatic classification this can be considered as a serious limitation, but such techniques are limited only to specific classes and re-training is required on any new category. This is also another difference between the previous work and [Vouros et al., 2018], by using the RODA software [Vouros et al., 2017] the user is free to provide custom categories and assess the classification based on the 8 features as described in the aforementioned study. To this end, and based on the extension of the RODA software described in Chapter 6 and the path features of Chapter 3, the users can perform (a) custom classification and feature assessment from a database of potential features for the Morris Water Maze and (b) excluding the features specifically designed for the Morris Water Maze (such as target proximity), they can have classification on any constrained experimental environment.

It should be mentioned that in case that no prior knowledge of the animal behaviours is available (in the form of labelled data) and the aim of the analysis is pattern recognition and informative feature detection, then expertise in the field of machine learning is required. In the proposed framework of this PhD study in Chapter 5 [Vouros et al., 2018] and Chapter 6, it is expected that the user is familiar in the field of behavioural neuroscience and able to manually access the animal behaviours in order to partially label the data; after the partial labelling, the other procedures such as the features computation and the classification are automatic. However, in the case of pattern detection using the sparse clustering methodology presenting in Chapter 3 there should be expertise on assessing the clustering quality since some clusters might belong to the same categories. Unsupervised pattern detection and feature assessment were investigated in the benchmark of the semi-supervised sparse clustering presenting in Chapter 3 (see also [Vouros and Vasilaki, 2020]) but it has a number of applications. For example a collection of path features from Chapter 3

can be used along with sparse clustering to analyse experimental procedures without prior knowledge of class labels such as the Allothetic Place Avoidance task [Stuchlik et al., 2004]. A potential difficulty of such application is that, if the overlapping segmentation methodology of [Gehring et al., 2015; Vouros et al., 2018] is used, it would create a natural continuum to the clusters, i.e. clusters might not be clearly separatable. An experiment-specific criterion can be used instead to segment the paths without overlapping as was the case in Chapter 4 (see also [Chhabria et al., 2019]) where the entrances/exits to/from the light/dark areas of the arena were used as criterion of segmentation. For the Allothetic Place Avoidance task such criterion can be sudden spikes of angular speed [Gehring et al., 2017].

Finally, this PhD study has left some room for future work towards benchmarking regarding the sparse clustering K-Means methods. The proposed semi-supervised Sparse K-Means (PCSK-Means) algorithm is indicated that can be tuned by less computationally intense methods than its predecessor the Sparse K-Means. Both these algorithms require the tuning of two parameters, the number of clusters (K) and the degree of spareness (S). Based on the work of [Brodinová et al., 2017] both these parameters can be optimised using the modified gap statistic used in the original study of sparse K-Means clustering [Witten and Tibshirani, 2010]. This method (refer to section 2.2.8.1.3) is computationally expensive and it might be avoided with the use of silhouette. A preliminary analysis (results not conclusive) showed that silhouette can indicate an optimal values for (S) for each (K) but fails to tune both these parameters together. Nevertheless, a potential tuning method would have been to run the silhouette index for multiple value of (S) over the same values of (K) and for each (K) to identify the optimal (S). Then for these values of (S) to execute the modified gap statistic over the values of (K). This procedure will not nullify the need of the modify gap statistic but it will greatly reduce its additional executions. For semi-supervised learning, the $\mathbb{k}$-fold cross validation 2.2.8.3 can be used along with the silhouette index as in Chapter 6.

Apart from the tuning of the sparse clustering procedures two more expansions can be proposed for the PCSK-Means algorithm. In the study of [Witten and Tibshirani, 2010] sparse clustering is extended to hierarchical clustering [Johnson, 1967], another common method with both clustering and classification applications in different fields such as environmental [Govender and Sivakumar, 2020] and biological studies [Seo and Shneiderman, 2002]. The PCSK-Means algorithm can possibly be applied to the same hierarchical clustering framework. In addition, the study of [Brodinová et al., 2017] proposes a method to incorporate the LOF score of the ROBIN initialisation procedure (refer to section 2.2.7.5) to the sparse clustering framework of [Witten and Tibshirani, 2010] in order to identify outliers in the data. It is envisaged that such method is directly applicable to the PCSK-Means algorithm.

To conclude, the application of Sparse K-Means clustering and its proposed semi-supervised version on behavioural experiments requires further testing. Chapter 6 showed that the feature selection capabilities of the PCSK-Means algorithm produces interpretable and logical results but are the conclusions applicable to any similar experimental procedure towards stressed animal capabilities inside the Morris Water Maze or they are affected by the specific animals and data? Also, in a possible collection of generic path features (e.g. the ones listed in Chapter 3) will their application to different experiments be consistent towards identification of possible patterns and biomarkers or will they be dependent on the specific data under analysis?

Further research is required to answer these questions but the methods developed in this PhD have the potential to be engineered and improved or be attached to various pipelines thus they are directly applicable to a variety of experimental procedures requiring path analysis.

7.3 Alternative machine learning methods

The machine learning methods that were described in this PhD study (refer to Chapter 2 for a literature review), researched and developed (refer to Chapter 3 for the new methods and benchmarking) and finally applied to behavioural experiments (refer to Chapters 5 and 6) are based on clustering and specifically the K-Means clustering. Another machine learning approach that is popular in many fields for classification are the artificial neural networks (ANN) [Prieto et al., 2016]. Deep learning networks, which are essentially multilayered ANNs, have been used extensively for image processing task in the fields of medicine [Litjens et al., 2017] and bioinformatics [Min et al., 2017]. In the study of [Higaki, Mogi, Iwanami, Min, Bai, Shan, Kukida, Kan-no, Ikeda, Higaki et al., 2018] a deep learning ANN was used in to analyse mice performance inside the Morris Water Maze during the first few days of the task in order to predict their performance on the final day. In this way the authors propose that the duration of the experimental procedure can be reduced. The same authors also developed a deep neural network capable of classifying behavioural classes inside the Morris Water Maze [Higaki, Mogi, Iwanami, Min, Bai, Shan, Kan-no, Ikeda, Higaki and Horiuchi, 2018]. Their network performance was above 90% accuracy when classifying between 2 and 3 behavioural classes but dropped to 65% with 6. This is one of the issues of deep learning ANNs, that they require large amount of data in order to be trained for classification compared with other "shallow" methods such as support vector machines (SVM) [Illouz, Madar, Clague, Griffioen, Louzoun and Okun, 2016; Illouz, Madar, Louzon, Griffioen and Okun, 2016]. The amount of data produced during a common experimental procedure is usually not that great but with the overlapping segmentation method of [Gehring et al., 2015], which was furthered researched in this PhD study (refer also to [Vouros et al., 2018]), more data can be generated from the original animal paths. Thus it can potentially be beneficial for classification with ANNs. Artificially created data is also another way to deploy ANN solutions. Such data can be created by determining best fit distributions on real data [Scott and Wilkins, 1999].

Furthermore, and regarding deep learning ANNs, there are some additional issues. First of all, ANNs require extensive development process for determining the proper network structure and hyperparameter tuning. Moreover, deep learning ANNs are suffering from interpretability issues regarding on how they use the features to establish a solution. In this PhD work, much effort was made in order to develop pipelines for feature assessment, behavioural clustering and classification in a clear interpretable manner. ANNs ability to perform well comes from the automatic selection of feature quantities [Higaki, Mogi, Iwanami, Min, Bai, Shan, Kan-no, Ikeda, Higaki and Horiuchi, 2018] which might not have a clear interpretation. Towards unsupervised learning, autoencoders [Baldi, 2012] are dedicated ANNs for reconstructing the original data using compressed information from their features. This, however results in losing the original features and, as explained before, this might be a disadvantage since the new features might not a direct biological interpretation

as the original. Sparse autoencoders are also available [Luo et al., 2017] but still they are more complex methods that the ones developed in this PhD study.

Apart from ANN and related to clustering there are other approaches beyond K-Means such as spectral [Ng et al., 2002] and kernel [Shawe-Taylor et al., 2004] methods. These methods were not researched in this study mainly because they are using data transformation. Spectral methods transform the data projecting them to a lower dimensionality based on similarity matrices while kernel methods project the data to a higher dimensionality using a kernel function such as a polynomial, a gaussian or a sigmoid. Both methods are performing well on identifying clusters that are non-linearly separable and they can both be followed K-Means clustering on the transformed data. However, the new features forming the transformed data are not maintaining the biological interpretation of the original features thus more work needs to be done afterwards to reconstruct the biological information that the new features represent. With Sparse K-Means clustering [Witten and Tibshirani, 2010] as well as the new semi-supervised Sparse K-Means algorithm developed in this PhD study (refer to Chapter 3 and [Vouros and Vasilaki, 2020]) both pattern recognition performance as well as feature assessment and selection are directly possible.

An alternative approach to analyse path data given their currently low dimensionality could have been to detect and examine local correlations [Xie et al., 2013]. Such correlations may occur for a certain duration as the animals navigate inside the constrained environment of the experimental procedure and indicate behavioural patters (clusters [Papadimitriou et al., 2003]) and outlying observations [Papadimitriou et al., 2003], ultimately resulting to behavioural differences recognition among animal grounds. This kind of analysis would also include timing information [Papadimitriou et al., 2006] which has not been utilized by the current study.

Finally, another approach for unsupervised behavioural motifs detection using timing information would have been a probabilistic annotation such as the one implemented in the study of [Szigeti et al., 2015]. A probabilistic annotation is essentially the association of a single or multiple groups of features to specific patterns [Del Carratore et al., 2019]. The method developed by [Szigeti et al., 2015] is free of threshold values which are usually used to define sliding windows that segment time series data (such as the position of animals at specific points of time) [Brown et al., 2013]. It also offers quantification of uncertainty, e.g. how much discrete or stereotypical are the detected animal motifs.

Appendix A

A.1 Metric parameterization

Assume the matrix $\mathcal{X}$ containing a data set consisting of n observations and p features and A a square $p-$by$-p$ matrix parameterizing a metric.

- If A is a diagonal matrix then $\mathcal{X}A$ results in feature weighting since,

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & \\ \vdots & & \ddots & \\ x_{n1} & & & x_{np} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & & \\ \vdots & & \ddots & \\ 0 & & & a_{pp} \end{bmatrix} = \begin{bmatrix} x_{11}a_{11} & x_{12}a_{22} & \cdots & x_{1p}a_{pp} \\ x_{21}a_{11} & x_{22}a_{22} & & \\ \vdots & & \ddots & \\ x_{n1}a_{11} & & & x_{np}a_{pp} \end{bmatrix} =$$

$$= \mathcal{X}A = \mathcal{X}diag(A)^T$$

- If A is a full matrix then $\mathcal{X}A$ results in feature generation, i.e. the resulted features are linear combinations of the original ones, since,

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & & \\ \vdots & & \ddots & \\ x_{n1} & & & x_{np} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{p1} & & & a_{pp} \end{bmatrix} =$$

$$= \begin{bmatrix} x_{11}a_{11} + x_{12}a_{21} + \cdots + x_{1p}a_{p1} & \cdots & \cdots & x_{11}a_{1p} + x_{12}a_{2p} + \cdots + x_{1p}a_{pp} \\ \vdots & \ddots & & \\ \vdots & & \ddots & \\ x_{n1}a_{11} + x_{n2}a_{21} + \cdots + x_{np}a_{p1} & \cdots & \cdots & x_{n1}a_{1p} + x_{n2}a_{2p} + \cdots + x_{np}a_{pp} \end{bmatrix}$$

This case is not considered in this PhD study.

A.2 K-Means cluster centers

At convergence, $\frac{\partial \mathcal{J}_{kmeans}}{\partial m_{k'j'}} = 0 \Rightarrow$

$$\frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 = 0 \Rightarrow$$

$$2 \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} (x_{ij'} - m_{k'j'})(-1) = 0 \Rightarrow$$

$$\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} x_{ij'} = \sum_{i}^{n_{k'}} m_{k'j'} \Rightarrow$$

$$\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} x_{ij'} = n_{k'} m_{k'j'} \Rightarrow$$

$$m_{k'j'} = \frac{1}{n_{k'}} \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} x_{ij'}$$

Note: $\mathcal{J}_{kmeans}$ is convex since,

$$\frac{\partial^2 \mathcal{J}_{kmeans}}{\partial^2 m_{k'j'}} = \frac{\partial}{\partial m_{k'j'}} 2 \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} (x_{ij'} - m_{k'j'})(-1) = 2 \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} 1 > 0$$

A.3 Equivalent expressions for WCSS

We will show that:

$$WCSS = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \quad (1)$$

Starting from the right hand side of equality (1)

$$\sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj} + m_{kj} - x_{i'j})^2 =$$

$$= \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} \big((x_{ij} - m_{kj})^2 + (m_{kj} - x_{i'j})^2 + 2(x_{ij} - m_{kj})(m_{kj} - x_{i'j}) \big) =$$

$$= \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{i'=1}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 + \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{i=1}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} (m_{kj} - x_{i'j})^2 +$$

$$+ \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} 2(x_{ij} - m_{kj})(m_k - x_{i'j}) =$$

$$= \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 \sum_{i'}^{n_k} 1 + \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{i'j} - m_{kj})^2 \sum_{i=1}^{n_k} 1 +$$

$$+ \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} 2(x_{ij} m_{kj} - x_{ij} x_{i'j} - m_{kj} m_{kj} + m_{kj} x_{i'j}) =$$

$$= \sum_{k=1}^{K} \frac{1}{2\cancel{n_k}} \sum_{\substack{i=1 \\ x_{ij} \in c_k}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 \cancel{n_k} + \sum_{k=1}^{K} \frac{1}{2\cancel{n_k}} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} (m_{kj} - x_{i'j})^2 \cancel{n_k} +$$

$$+ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{i'=1}^{n_k} \sum_{j=1}^{p} x_{ij} m_{kj} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} x_{ij} x_{i'j} +$$

$$+ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{\substack{i'=1 \\ x_{i':} \in c_k}}^{n_k} \sum_{j=1}^{p} m_{kj} x_{i'j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{i'=1}^{n_k} \sum_{j=1}^{p} m_{kj} m_{kj} =$$

$$= \sum_{k=1}^{K} \frac{1}{2} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 + \sum_{k=1}^{K} \frac{1}{2} \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sum_{j=1}^{p} (m_{kj} - x_{i'j})^2 +$$

$$+ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{ij} m_{kj} \sum_{i'=1}^{n_k} 1 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{ij} \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{i'j} +$$

$$+ \sum_{k=1}^{K} \frac{1}{n_k} \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sum_{j=1}^{p} m_{kj} x_{i'j} \sum_{i=1}^{n_k} 1 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{i'=1}^{n_k} \sum_{j=1}^{p} m_{kj} m_{kj} =$$

$$= \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 + \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{ij} m_{kj} - \sum_{k=1}^{K} \sum_{j=1}^{p} \frac{1}{n_k} n_k m_{kj} n_k m_{kj} +$$

$$+ \sum_{k=1}^{K} \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sum_{j=1}^{p} m_k x_{i'j} - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{p} m_k m_k n_k n_k =$$

$$= \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 + \sum_{k=1}^{K} \sum_{j=1}^{p} m_{kj} m_{kj} n_k -$$

$$- \sum_{k=1}^{K} \sum_{j=1}^{p} n_k m_{kj} m_{kj} + \sum_{k=1}^{K} \sum_{j=1}^{p} m_{kj} m_{kj} n_k - \sum_{k=1}^{K} \sum_{j=1}^{p} n_k m_{kj} m_{kj} =$$

$$= \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} (x_{ij} - m_{kj})^2 = WCSS, \qquad \text{Q.E.D.}$$

We note that we make use of the following equalities,

$$\sum_{i=1}^{n_k} 1 = \sum_{i'=1}^{n_k} 1 = n_k$$

and

$$\sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{ij} = \sum_{\substack{i'=1 \\ (x_{i':} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{i'j} = \sum_{j=1}^{p} n_k m_{kj} \quad \text{, since} \quad \sum_{j=1}^{p} m_{kj} = \frac{1}{n_k} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} x_{ij}$$

(refer to Appendix A.2).

A.4 MPCK-Means algorithm

We define,

$$J_{mpckm} = \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \Bigg(\sum_{j=1}^{p} a_j (x_{ij} - m_{kj})^2 - \sum_{j=1}^{p} log(a_j) + \sum_{(x_{i:}) ML(x_{i':})} \sum_{j=1}^{p} b_{x_i, x_{i'}} a_j (x_{ij} - x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:}) \cancel{ML}(x_{i':})\big] + \sum_{(x_{i:}) CL(x_{i':})} \sum_{j=1}^{p} \bar{b}_{x_i, x_{i'}} \big(a_j (x_{Ij} - x_{I'j})^2 - a_j (x_{ij} - x_{i'j})^2 \big) \;\; \mathbb{1}\big[(x_{i:}) \cancel{CL}(x_{i':})\big] \Bigg)$$

At convergence, $\frac{\partial \mathcal{J}_{mpckm}}{\partial m_{k'j'}} = 0 \Rightarrow$

$$\frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sum_{j=1}^{p} a_j (x_{ij} - m_{kj})^2 = 0 \Rightarrow$$

$$2 a_{j'} \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} (x_{ij'} - m_{k'j'})(-1) = 0 \Rightarrow$$

$$\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} x_{ij'} = \sum_{i=1}^{n_{k'}} m_{k'j'} \Rightarrow$$

$$\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} x_{ij'} = n_{k'} m_{k'j'} \Rightarrow$$

$$m_{k'j'} = \frac{1}{n_{k'}} \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} x_{ij'}$$

and, $\frac{\partial J_{mpckm}}{\partial a_{j'}} = 0 \Rightarrow$

$$\frac{\partial}{\partial a_{j'}} \Bigg(\sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \Bigg(a_j (x_{ij} - m_{kj})^2 - log(a_j) + \sum_{(x_{i:}) ML(x_{i':})} b_{x_i, x_{i'}} a_j (x_{ij} - x_{i'j})^2 \;\; \mathbb{1}\big[(x_{i:}) \cancel{ML}(x_{i':})\big] + \sum_{(x_{i:}) CL(x_{i':})} \bar{b}_{x_i, x_{i'}} \big(a_j (x_{Ij} - x_{I'j})^2 - a_j (x_{ij} - x_{i'j})^2 \big) \;\; \mathbb{1}\big[(x_{i:}) \cancel{CL}(x_{i':})\big] \Bigg) \Bigg) = 0 \Rightarrow$$

$$\frac{\partial}{\partial a_{j'}}\Bigg(\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k} a_j(x_{ij}-m_{kj})^2-\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k} log(a_j)+$$
$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}a_j(x_{ij}-x_{i'j})^2\ \ \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+$$
$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big(a_j(x_{Ij}-x_{I'j})^2-a_j(x_{ij}-x_{i'j})^2\big)\ \ \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)=0\Rightarrow$$

$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}(x_{ij'}-m_{kj'})^2-\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\frac{1}{a_{j'}}+$$
$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij'}-x_{i'j'})^2\ \ \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+$$
$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big((x_{Ij'}-x_{I'j'})^2-(x_{ij'}-x_{i'j'})^2\big)\ \ \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]=0\Rightarrow$$

$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}(x_{ij'}-m_{kj'})^2+$$
$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij'}-x_{i'j'})^2\ \ \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+$$
$$\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big((x_{Ij'}-x_{I'j'})^2-(x_{ij'}-x_{i'j'})^2\big)\ \ \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]=n\frac{1}{a_{j'}}\Rightarrow$$

$$a_{j'}=n\Bigg(\sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}\Bigg((x_{ij'}-m_{kj'})^2+$$
$$\sum_{(x_{i:})ML(x_{i':})} b_{x_i,x_{i'}}(x_{ij'}-x_{i'j'})^2\ \ \mathbb{1}\big[(x_{i:})\cancel{ML}(x_{i':})\big]+$$
$$\sum_{(x_{i:})CL(x_{i':})} \bar{b}_{x_i,x_{i'}}\big(a_{j'}(x_{Ij'}-x_{I'j'})^2-(x_{ij'}-x_{i'j'})^2\big)\ \ \mathbb{1}\big[(x_{i:})\cancel{CL}(x_{i':})\big]\Bigg)\Bigg)^{-1}$$

A.5 K-Medians cluster centers

At convergence, $\frac{\partial \mathcal{J}_{kmedian}}{\partial m_{k'j'}} = 0 \Rightarrow$

$$\frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ (x_{i:} \in c_k)}}^{n_k} \sum_{j=1}^{p} |x_{ij} - m_{kj}| = 0 \tag{2}$$

We note that

$$\frac{\partial}{\partial |x|} = \left\{ \begin{matrix} 1 & , x > 0 \\ -1 & , x < 0 \end{matrix} \right\} = sign(x)$$

Therefore equation 2 becomes,

$$\sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} sign(x_{ij'} - m_{k'j'}) = 0$$

Assuming that for $i = 1, \ldots, z \quad x_{ij'} \leq m_{k'j'}$ and for $i = z+1, \ldots, n_{k'} \quad x_{ij'} > m_{k'j'}$, then:

$$\sum_{\substack{i=1 \\ (x_{i:} \in c_{k'})}}^{z} (-1) + \sum_{\substack{i=z+1 \\ (x_{i:} \in c_{k'})}}^{n_{k'}} (1) = 0 \Rightarrow -z + (n_{k'} - z) = 0 \Rightarrow z = \frac{n_{k'}}{2}$$

which imposes that we have the same amount of data points above and below $m_{k'j'}$, thus $m_{k'j'}$ is the median.

A.6 Geometric K-Medians cluster centers

At convergence, $\frac{\partial \mathcal{J}_{gkmedians}}{\partial m_{k'j'}} = 0 \Rightarrow$

$$\frac{\partial}{\partial m_{k'j'}} \sum_{k=1}^{K} \sum_{\substack{i=1 \\ x_{i:} \in c_k}}^{n_k} \sqrt{\sum_{j=1}^{p} (x_{ij} - m_{kj})^2} = 0 \Rightarrow$$

$$\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \frac{1}{2} \Big((x_{ij'} - m_{k'j'})^2 \Big)^{-\frac{1}{2}} 2(x_{ij'} - m_{k'j'})(-1) = 0 \Rightarrow$$

$$- \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \Big((x_{ij'} - m_{k'j'})^2 \Big)^{-\frac{1}{2}} (x_{ij'} - m_{k'j'}) = 0 \Rightarrow$$

$$- \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \frac{x_{ij'} - m_{k'j'}}{\sqrt{(x_{ij'} - m_{k'j'})^2}} = 0 \Rightarrow$$

$$\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \frac{x_{ij'}}{\sqrt{(x_{ij'} - m_{k'j'})^2}} = \sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \frac{m_{k'j'}}{\sqrt{(x_{ij'} - m_{k'j'})^2}} = 0 \Rightarrow$$

$$m_{k'j'} = \frac{\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \frac{x_{ij'}}{\sqrt{(x_{ij'} - m_{k'j'})^2}}}{\sum_{\substack{i=1 \\ x_{i:} \in c_{k'}}}^{n_{k'}} \frac{1}{\sqrt{(x_{ij'} - m_{k'j'})^2}}}$$

A.7 Minimizing a function with L_1 and L_2 penalties

We will show that the L_1 penalty is not affected by the magnitude of w while L_2 does. Let the function $f(x)$ and a constant $\lambda > 0$.

Applying L_1 to the function and minimizing the penalty leads to a fixed penalization:

$$\frac{\partial}{\partial L_1}(f(x) + \lambda L_1) = 0 \Rightarrow \frac{\partial}{\partial w_i}(f(x) + \lambda \sum_i |w|) = 0 \Rightarrow$$
$$\frac{\partial}{\partial w_i} f(x) + \lambda \ \ sign(w_i) = 0 \Rightarrow \frac{\partial}{\partial w_i} f(x) = -\lambda sign(w_i)$$

i.e. the gradient is independent of the magnitude of w.

Applying L_2 to the function and minimizing the penalty leads to a penalization proportional to the weight:

$$\frac{\partial}{\partial L_2}(f(x) + \lambda L_2) = 0 \Rightarrow \frac{\partial}{\partial w_i}(f(x) + \lambda \sum_i w_i^2) = 0 \Rightarrow$$
$$\frac{\partial}{\partial w_i} f(x) + 2\lambda w_i = 0 \Rightarrow \frac{\partial}{\partial w_i} f(x) = -2\lambda w_i$$

i.e. the gradient is dependent of w.

A.8 Sparse clustering optimization with L_1 and L_2 constraints

In section 2.2.3 we define the problem,

$$\underset{w_j}{maximize}\Big\{\sum_{j=1}^{p} w_j\gamma_j\Big\} \quad \text{subject to} \quad \sum_{j=1}^{p} w_j^2 \leq 1,\ \sum_{j=1}^{p}|w_j| \leq s,\ w_j \geq 0\ \ \forall j \tag{3}$$

where in the case of Sparse K-Means clustering,

$$\gamma_j = \sum_{i=1}^{n}(x_{ij} - \mu_{1j})^2 - \sum_{k=1}^{K}\sum_{\substack{i=1\\(x_{i:}\in c_k)}}^{n_k}(x_{ij} - m_{kj})^2 \tag{4}$$

Here, the full solution of this problem is presented (for the notations refer to the beginning of this dissertation).
Using Lagrange multipliers we can rewrite 3 as,

$$f(w_j) = \sum_{j=1}^{p} w_j\gamma_j - \lambda\big(\sum_{j=1}^{p} w_j^2 - 1\big) - \Delta\big(\sum_{j=1}^{p}|w_j| - s\big)\ \ ,\lambda > 0,\ \ \Delta > 0,\ \ \forall j \tag{5}$$

We differentiate with respect to w_i and set the derivative to 0,

$$\frac{\partial f(w)}{\partial w_i} = \gamma_i - 2\lambda w_i - \Delta\frac{\partial|w_i|}{\partial w_i} = 0. \tag{6}$$

where,

$$\Gamma_i = \frac{\partial|w_i|}{\partial w_i} = \frac{\partial\sqrt{w_i^2}}{\partial w_i} = \frac{\partial(w_i^2)^{\frac{1}{2}}}{\partial w_i} = 2w_i\frac{1}{2}(w_i^2)^{-\frac{1}{2}} = \frac{w_i}{|w_i|} \tag{7}$$

For $w_i \neq 0$, $\Gamma_i = sign(w_i)$ else for $w_i = 0$, $\Gamma_i \in [-1, 1]$ (proof at the end of this section) Since our constraints also have inequalities we use the Karush-Kuhn-Tucker conditions (an extension of the Lagrange multipliers) [Boyd and Vandenberghe, 2004]

[1] $\gamma_i - 2\lambda w_i - \Delta\Gamma_i = 0$ (6).

[2] $\lambda(\sum_{j=1}^{p} w_j^2 - 1) = 0$ (L_2 constraint).

[3] $\Delta(\sum_{j=1}^{p}|w_j| - s) = 0$ (L_1 constraint).

We have from condition [1] above,

$$w_i = \frac{\gamma_i - \Delta\Gamma_i}{2\lambda} \Rightarrow w_i = \frac{sign(\gamma_i)|\gamma_i| - \Delta\Gamma_i}{2\lambda} \tag{8}$$

- If $\Gamma_i = 1$ i.e. $w_i \geq 0$ then,

$$(8) \Rightarrow w_i = \frac{sign(\gamma_i)|\gamma_i| - \Delta\cdot 1}{2\lambda} \Rightarrow w_i = \frac{\gamma_i - \Delta}{2\lambda}$$
$$\text{and} \quad w_i \geq 0 \quad \text{thus} \quad \gamma_i - \Delta \geq 0 \Rightarrow \gamma_i \geq \Delta \tag{9}$$

- If $\Gamma_i = -1$ for $w_i \leq 0$ then,

$$(8) \Rightarrow w_i = \frac{sign(\gamma_i)|\gamma_i| - \Delta \cdot (-1)}{2\lambda} \Rightarrow w_i = \frac{\gamma_i + \Delta}{2\lambda}$$
$$\text{and} \quad w_i \leq 0 \quad \text{thus} \quad \gamma_i + \Delta \leq 0 \Rightarrow \gamma_i \leq -\Delta \tag{10}$$

Hence from (9) and (10) we can only have two cases, $\gamma_i \geq \Delta$ and $\gamma_i \leq -\Delta$. These can be combined as, $sign(\gamma_i)(|\gamma_i| - \Delta)_+ \geq 0$ which leads to equation (11),

$$(8) \Rightarrow w_i = \frac{sign(\gamma_i)(|\gamma_i| - \Delta)_+}{2\lambda} \tag{11}$$

Defining $S(\gamma_i, \Delta) = sign(\gamma_i)(|\gamma_i| - \Delta)_+$,

$$(11) \Rightarrow w_i = \frac{S(\gamma_i, \Delta)}{2\lambda} \tag{12}$$

From condition [2] and equation (12),

$$\sum_{i=1}^{p} w_i^2 = 1 \Rightarrow \sum_{i=1}^{p} \frac{\left[S(\gamma_i, \Delta)\right]^2}{(2\lambda)^2} = 1 \Rightarrow$$
$$(2\lambda)^2 = \sum_{i=1}^{p} \left[S(\gamma_i, \Delta)\right]^2 \Rightarrow 2\lambda = \sqrt{\sum_{i=1}^{p} \left[S(\gamma_i, \Delta)\right]^2} \tag{13}$$

Thus equation (12) can be written as,

$$w_i = \frac{sign(\gamma_i)(|\gamma_i| - \Delta)_+}{\sqrt{\sum_{j=1}^{p} \left(sign(\gamma_j)(|\gamma_j| - \Delta)_+\right)^2}} \tag{14}$$

Finally for the s parameter in condition [3], figure A.1 shows geometrically that the upper bound for the parameter s is $\sqrt{p}$, where p is the dimensionality of the problem.

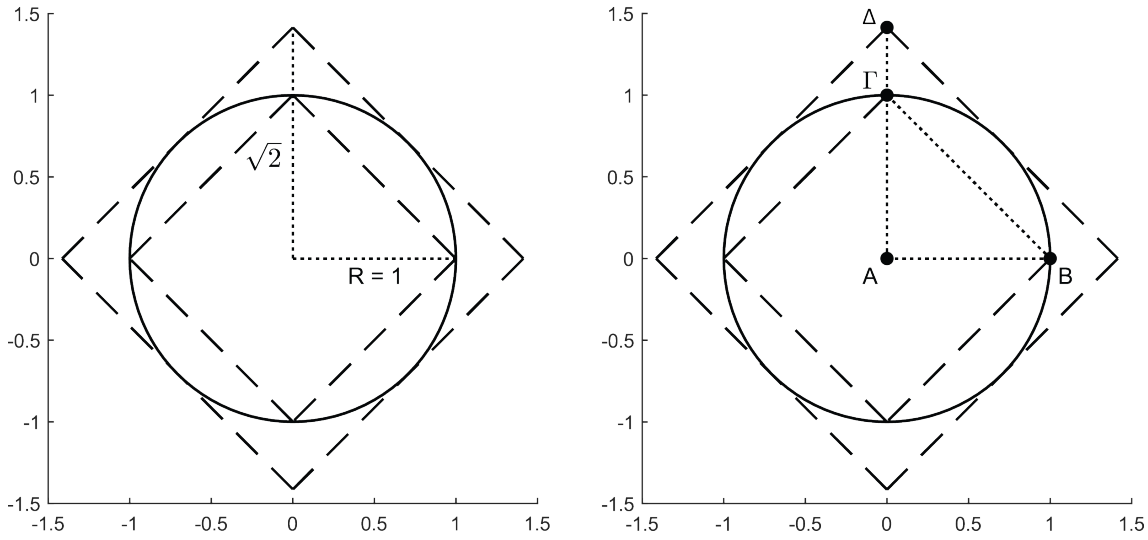


Figure A.1: A graphical representation of L_1 and L_2 constraints for sparse clustering. In both plots, the solid circle with radius $R = 1$ illustrates the $L_2 = \sum_{j=1}^{p} w_j^2 \leq 1$ and the two dashed squares the $L_1 = \sum_{j=1}^{p} |w_j| \leq s$ for $s = 1$ (inner square) and $s = \sqrt{2}$ (outer square). In this 2-dimensional scenario, for both constraints to be active s needs to be between 1 and $\sqrt{2}$. In higher dimensional problems s upper bound is $\sqrt{p}$ where p is the number of dimensions. This upper bound can be explained from the right plot where, $AB = A\Gamma$ and $B\Gamma^2 = AB^2 + A\Gamma^2 \Rightarrow B\Gamma = \sqrt{2} = A\Delta$. For p dimensions $B\Gamma = \sqrt{p}$.

Convexity of $|x|$

Let the function f defined on a real interval $[I_1, I_2]$ and $x, y \in [I_1, I_2]$. f is convex if,

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

Let $f(x) = |x|$ then,

$$|\lambda x + (1-\lambda)y| \leq |\lambda x| + |(1-\lambda)y| \Rightarrow \lambda|x| + (1-\lambda)|y| \leq \lambda|x| + (1-\lambda)|y|$$

which is true based on the triangle inequality: $|a+b| \leq |a| + |b|$. Q.E.D.

Subdifferential of $|x|$ at 0

The subdifferential of $|x|$ at 0 is the interval $[-1, 1]$ since $|x|$ is convex and,

$$\frac{\partial |x|}{\partial x} = \lim_{\delta x \to 0} \frac{|x + \delta x| - |x|}{\delta x}$$

where, $\lim_{\delta x \to 0^+} \frac{|\delta x|}{\delta x} = 1$ and $\lim_{\delta x \to 0^-} \frac{|\delta x|}{\delta x} = -1$

Appendix B

Table B.1: Detailed comparison on the maximum performance of stochastic methods with the performance of deterministic methods. Each row compares a stochastic with a deterministic method over the Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld's (Weis) algorithms on 26 occasions (10 gap and weighted gap, 12 Brodinova and 4 mixed models). The comparison is based on the times that there was significant difference between the two methods on their maximum performance. Based on the results stochastic methods are better that deterministic on achieving the best performance but the more sophisticated the stochastic method is the less performance difference it achieves compared with determinstc methods for which the opposite is observed. This Table accompanies Figure 3.5 and Figure 3.6 which show the maximum performance of the initialisation methods and show comparisons only for the Hartigan-Wong's algorithm.

Initialization method	**Total number of instances**	**Significant better maximum performance Purity for best Silhouette**			
		HW	**Ll**	**KMed**	**Weis**
Random vs Kaufman	26	9 vs 3	9 vs 4	9 vs 4	9 vs 3
Random vs DK-Means++	26	5 vs 3	4 vs 2	5 vs 3	5 vs 1
Random vs ROBIN(D)	26	6 vs 3	8 vs 2	7 vs 2	8 vs 2
Random vs Maximin(D)	26	10 vs 2	12 vs 1	11 vs 1	12 vs 1
K-Means++ vs Kaufman	26	10 vs 0	12 vs 0	10 vs 0	11 vs 1
K-Means++ vs DK-Means++	26	6 vs 2	6 vs 2	7 vs 2	9 vs 0
K-Means++ vs ROBIN(D)	26	8 vs 2	9 vs 2	8 vs 2	9 vs 1
K-Means++ vs Maximin(D)	26	10 vs 1	12 vs 1	11 vs 1	13 vs 0
ROBIN(S) vs Kaufman	26	9 vs 4	8 vs 6	8 vs 4	9 vs 6
ROBIN(S) vs DK-Means++	26	0 vs 2	1 vs 3	2 vs 2	1 vs 2
ROBIN(S) vs ROBIN(D)	26	6 vs 0	6 vs 0	6 vs 0	5 vs 0
ROBIN(S) vs Maximin(D)	26	8 vs 0	9 vs 1	9 vs 0	9 vs 0
Maximin(S) vs Kaufman	26	9 vs 4	9 vs 4	8 vs 4	11 vs 4
Maximin(S) vs DK-Means++	26	2 vs 4	2 vs 5	3 vs 5	4 vs 4
Maximin(S) vs ROBIN(D)	26	4 vs 4	6 vs 4	4 vs 4	6 vs 4
Maximin(S) vs Maximin(D)	26	9 vs 1	11 vs 1	10 vs 1	12 vs 0

Table B.2: Summary of comparisons on average performance of stochastic and deterministic methods over different K-Means variations on synthetic data set models. In the first part of the table, each row compares two different methods over the Hartigan-Wong's K-Means (HW), Lloyd's K-Means (Ll), K-Medians (KMed) and Weiszfeld's (Weis) algorithms on 26 occasions (10 gap and weighted gap, 12 Brodinova and 4 mixed models). The comparison is separate among the stochastic and deterministic methods and based on the times that there was significant difference between the two methods over the 40 data sets of each model. Based on the results ROBIN(S) is the best performer of stochastic methods and DK-Means the best performer of deterministic methods, both over all the clustering algorithms. The second part of the table groups all the stochastic and deterministic methods together and counts the overall percentage of observed significant differences. Based on the results the performance differences among the deterministic methods are less compared to the stochastic methods suggesting less performance variability. This Table accompanies Figure 3.2 and Figure 3.3 of the main manuscript which show the average performance of the initialisation methods based on Silhouette.

Initialization method	**Total number of instances**	**Significant better average performance Silhouette (Purity)**			
		HW	**Ll**	**KMed**	**Weis**
Random vs K-Means++	26	0 vs 23 (0 vs 21)	0 vs 23 (0 vs 22)	0 vs 23 (0 vs 24)	0 vs 23 (0 vs 21)
Random vs ROBIN(S)	26	1 vs 22 (1 vs 22)	3 vs 22 (3 vs 22)	2 vs 23 (2 vs 23)	0 vs 22 (2 vs 22)
Random vs Maximin(S)	26	0 vs 17 (6 vs 15)	1 vs 19 (6 vs 16)	1 vs 19 (6 vs 16)	0 vs 19 (2 vs 17)
K-Means++ vs ROBIN(S)	26	1 vs 22 (1 vs 21)	3 vs 22 (3 vs 21)	2 vs 23 (2 vs 22)	2 vs 22 (2 vs 21)
K-Means++ vs Maximin(S)	26	5 vs 14 (8 vs 12)	5 vs 16 (9 vs 14)	5 vs 15 (7 vs 13)	6 vs 15 (8 vs 13)
ROBIN(S) vs Maximin(S)	26	17 vs 1 (17 vs 1)	17 vs 3 (18 vs 3)	18 vs 1 (19 vs 1)	17 vs 2 (18 vs 2)
Kaufman vs DK-Means++	26	3 vs 8 (3 vs 9)	3 vs 9 (3 vs 10)	3 vs 7 (4 vs 8)	4 vs 8 (5 vs 10)
Kaufman vs ROBIN(D)	26	5 vs 8 (4 vs 8)	6 vs 8 (6 vs 6)	5 vs 6 (4 vs 6)	5 vs 8 (6 vs 8)
Kaufman vs Maximin(D)	26	8 vs 5 (8 vs 5)	10 vs 6 (10 vs 6)	9 vs 5 (9 vs 5)	3 vs 7 (4 vs 8)
DK-Means++ vs ROBIN(D)	26	2 vs 1 (4 vs 0)	5 vs 1 (6 vs 0)	4 vs 1 (6 vs 1)	3 vs 0 (5 vs 0)
DK-Means++ vs Maximin(D)	26	7 vs 0 (9 vs 0)	9 vs 0 (11 vs 0)	7 vs 0 (10 vs 1)	7 vs 0 (9 vs 0)
ROBIN(D) vs Maximin(D)	26	8 vs 1 (8 vs 1)	9 vs 1 (10 vs 1)	8 vs 1 (9 vs 1)	8 vs 1 (9 vs 1)
Initialization methods	**Total number of instances**	**Observed significant performance differences on average performance Silhouette (Purity)**			
		HW	**Ll**	**KMed**	**Weis**
Stochastic: Random, K-Means++, ROBIN(S), Maximin(S)	156	78.8% (80.1%)	86.0% (87.8%)	84.6% (86.5%)	82.1% (82.1%)
Deterministic: Kaufman, DK-Means++, ROBIN(D), Maximin(D)	156	36.0% (37.8%)	42.9% (44.2%)	36.0% (41.0%)	34.6% (41.7%)

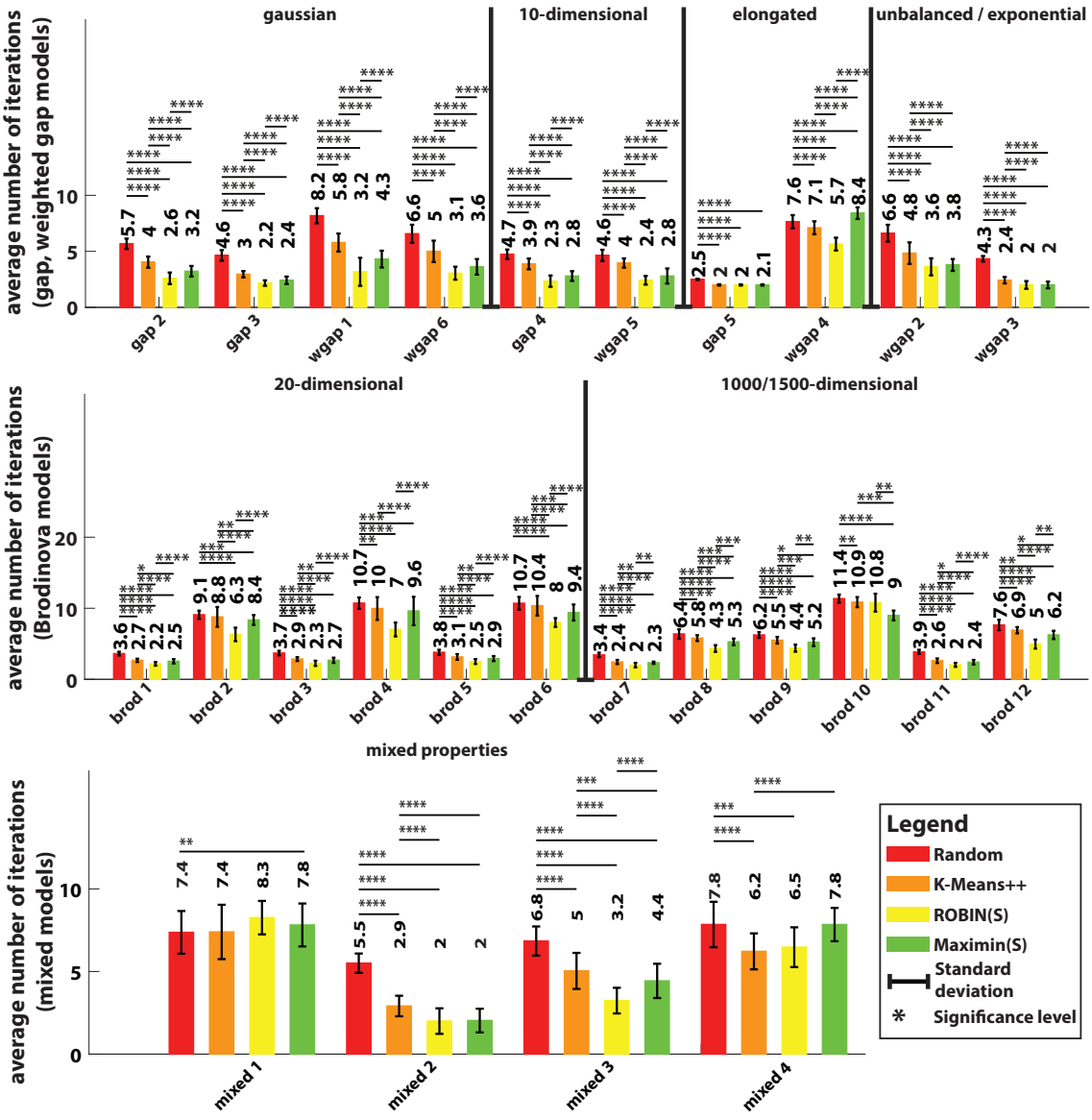


Figure B.1: Average number of iterations until convergence for the stochastic methods. Each plot shows the number of iteration of the Lloyd's K-Means algorithm (y-axis) until it reaches convergence using different stochastic initialisation methods on different data sets models (x-axis). To calculate the average number of iterations, we averaged the number of iterations across the 25 runs on the 40 data sets for each model (gap, weighted gap, Brodinova and mixed). The standard deviation corresponds to the average standard deviation over the 25 runs of each data set. Solid lines on any two bars underline the level of significant difference between the corresponding methods (cases of no significant differences are not shown).

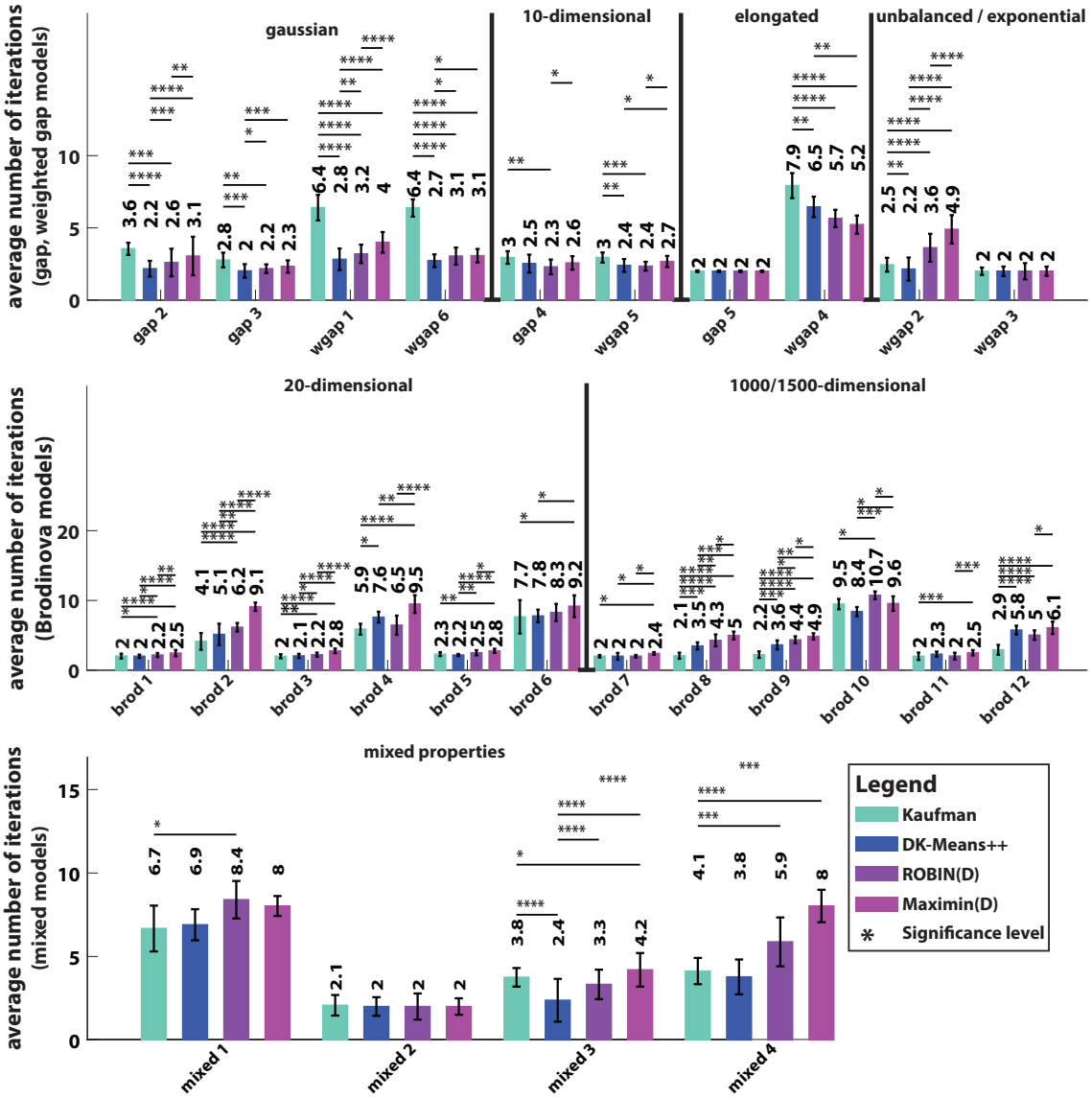


Figure B.2: Average number of iterations until convergence for the stochastic methods. Each plot shows the number of iteration of the Lloyd's K-Means algorithm (y-axis) until it reaches convergence using different deterministic initialisation methods on different data sets models (x-axis). To calculate the average number of iterations, we averaged the number of iterations across the 25 runs on the 40 data sets for each model (gap, weighted gap, Brodinova and mixed). The standard deviation corresponds to the average standard deviation over the 25 runs of each data set. Solid lines on any two bars underline the level of significant difference between the corresponding methods (cases of no significant differences are not showing). Table 3.7 shows a summary of the comparisons among all the different initialisation methods.

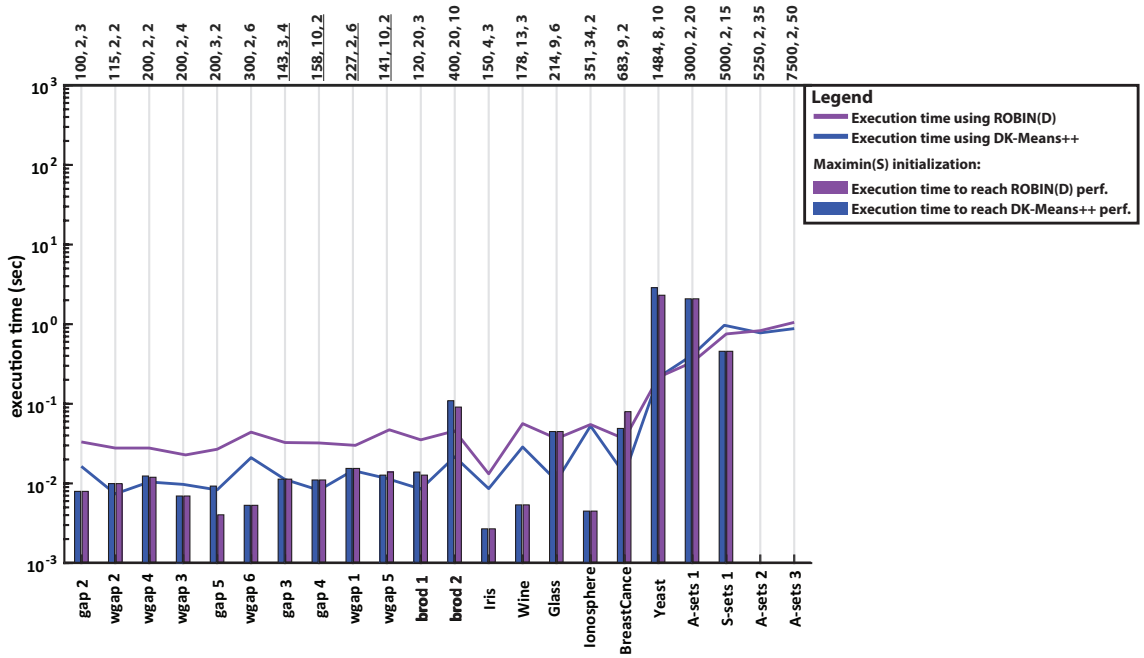


Figure B.3: Execution time analysis for K-Means clustering with Maximin(S) initialisation to reach the performance of DK-Means++ and ROBIN(D). Each bar shows the execution duration of K-Means clustering algorithm executed multiple times with the Maximin(S) initialisation method until reaching the performance of the clustering algorithm running once using DK-Means++ and ROBIN(D) methods. Lines indicate the execution time of K-Means clustering running once using the deterministic methods ROBIN(D) and DK-Means++. The clustering algorithm was given 50 execution repetitions with the stochastic Maximin(S) method to reach an equal or better solution than deterministic methods. Data sets without a bar means that no equal or better solution compared to the respective clustering with a deterministic method was found. The data sets are arranged based on their *size*, *dimensionality* and *number of clusters* (see info on top, underlined numbers means that for these models the generated data sets had different sizes). Results were averaged over 40 data sets for the data set models.

Table B.3: Comparison of the initialisation methods on standalone clustering data sets based on Silhouette index. Each stochastic method (Random, K-Means++, ROBIN(S) and Maximin (S)) was executed 50 times and the minimum, maximum and mean performance is shown followed by the performance variation for four K-Means variations. For the maximum performance the cases where a method has achieved the maximum performance is shown in bold.

		K-Means (Hartigan-Wong)				K-Means (Lloyd)				K-Medians				Weiszfeld			
		min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std
A-sets 1	Random	0.446	**0.595**	0.521	0.029	0.475	**0.595**	0.518	0.031	0.434	0.57	0.507	0.036	0.462	0.569	0.514	0.031
	K-Means++	0.482	**0.595**	0.548	0.03	0.484	**0.595**	0.548	0.024	0.469	**0.595**	0.531	0.029	0.5	**0.595**	0.548	0.021
	ROBIN(S)	0.567	**0.595**	0.586	0.013	0.568	**0.595**	0.58	0.014	0.567	**0.595**	0.585	0.013	0.567	**0.595**	0.586	0.013
	Maximin(S)	0.52	**0.595**	0.56	0.019	0.519	**0.595**	0.56	0.023	0.499	**0.595**	0.553	0.03	0.512	**0.595**	0.553	0.02
	Kaufman	0.567	0.567	0.567	0	0.567	0.567	0.567	0	0.567	0.567	0.567	0	0.568	0.568	0.568	0
	DK-Means++	0.595	**0.595**	0.595	0	0.595	**0.595**	0.595	0	0.595	**0.595**	0.595	0	0.595	**0.595**	0.595	0
	ROBIN(D)	0.567	0.567	0.567	0	0.568	0.568	0.568	0	0.567	0.567	0.567	0	0.567	0.567	0.567	0
	Maximin(D)	0.556	0.556	0.556	0	0.556	0.556	0.556	0	0.538	0.538	0.538	0	0.546	0.546	0.546	0
A-sets 2	Random	0.475	0.568	0.523	0.022	0.444	0.551	0.515	0.022	0.433	0.569	0.506	0.027	0.475	0.535	0.499	0.019
	K-Means++	0.505	**0.598**	0.543	0.02	0.483	0.584	0.547	0.024	0.473	0.58	0.531	0.021	0.488	0.582	0.54	0.023
	ROBIN(S)	0.58	**0.598**	0.591	0.008	0.581	**0.598**	0.59	0.008	0.581	**0.597**	0.59	0.008	0.581	**0.598**	0.589	0.008
	Maximin(S)	0.504	0.574	0.54	0.019	0.5	0.58	0.536	0.022	0.49	0.575	0.536	0.022	0.487	0.571	0.541	0.021
	Kaufman	0.565	0.565	0.565	0	0.565	0.565	0.565	0	0.538	0.538	0.538	0	0.564	0.564	0.564	0
	DK-Means++	0.598	**0.598**	0.598	0	0.598	**0.598**	0.598	0	0.598	**0.598**	0.598	0	0.597	**0.597**	0.597	0
	ROBIN(D)	0.598	**0.598**	0.598	0	0.598	**0.598**	0.598	0	0.597	**0.597**	0.597	0	0.598	**0.598**	0.598	0
	Maximin(D)	0.555	0.555	0.555	0	0.555	0.555	0.555	0	0.56	0.56	0.56	0	0.561	0.561	0.561	0
A-sets 3	Random	0.478	0.556	0.519	0.019	0.465	0.551	0.512	0.018	0.457	0.552	0.502	0.021	0.487	0.543	0.512	0.016
	K-Means++	0.514	0.588	0.547	0.017	0.506	**0.601**	0.548	0.017	0.485	0.576	0.533	0.019	0.506	0.575	0.542	0.017
	ROBIN(S)	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0
	Maximin(S)	0.525	0.585	0.556	0.016	0.525	0.589	0.558	0.015	0.52	0.586	0.559	0.017	0.532	0.571	0.554	0.012
	Kaufman	0.53	0.53	0.53	0	0.53	0.53	0.53	0	0.529	0.529	0.529	0	0.529	0.529	0.529	0
	DK-Means++	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0
	ROBIN(D)	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0	0.601	**0.601**	0.601	0
	Maximin(D)	0.588	0.588	0.588	0	0.588	0.588	0.588	0	0.588	0.588	0.588	0	0.588	0.588	0.588	0
S-sets 1	Random	0.52	**0.711**	0.616	0.037	0.545	0.663	0.612	0.035	0.497	0.662	0.587	0.047	0.495	0.66	0.594	0.046
	K-Means++	0.58	**0.711**	0.654	0.039	0.529	**0.711**	0.655	0.044	0.517	**0.711**	0.651	0.044	0.579	**0.711**	0.665	0.034
	ROBIN(S)	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0
	Maximin(S)	0.611	**0.711**	0.676	0.036	0.575	**0.711**	0.66	0.038	0.58	**0.711**	0.648	0.038	0.591	**0.711**	0.672	0.04
	Kaufman	0.711	**0.711**	0.711	0	0.638	0.638	0.638	0	0.654	0.654	0.654	0	0.592	0.592	0.592	0
	DK-Means++	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0
	ROBIN(D)	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0	0.711	**0.711**	0.711	0
	Maximin(D)	0.651	0.651	0.651	0	0.651	0.651	0.651	0	0.652	0.652	0.652	0	0.652	0.652	0.652	0
S-sets 2	Random	0.464	**0.626**	0.555	0.034	0.486	**0.626**	0.571	0.036	0.407	**0.626**	0.53	0.055	0.416	0.595	0.529	0.045
	K-Means++	0.516	**0.626**	0.586	0.031	0.505	**0.626**	0.577	0.032	0.485	**0.626**	0.566	0.035	0.532	**0.626**	0.584	0.028
	ROBIN(S)	0.575	**0.626**	0.617	0.02	0.575	**0.626**	0.61	0.024	0.568	**0.626**	0.606	0.028	0.572	**0.626**	0.611	0.025
	Maximin(S)	0.533	**0.626**	0.595	0.033	0.546	**0.626**	0.577	0.022	0.503	**0.626**	0.569	0.027	0.501	**0.626**	0.562	0.032
	Kaufman	0.57	0.57	0.57	0	0.57	0.57	0.57	0	0.571	0.571	0.571	0	0.571	0.571	0.571	0
	DK-Means++	0.626	**0.626**	0.626	0	0.626	**0.626**	0.626	0	0.626	**0.626**	0.626	0	0.626	**0.626**	0.626	0
	ROBIN(D)	0.626	**0.626**	0.626	0	0.626	**0.626**	0.626	0	0.626	**0.626**	0.626	0	0.626	**0.626**	0.626	0
	Maximin(D)	0.529	0.529	0.529	0	0.526	0.526	0.526	0	0.521	0.521	0.521	0	0.526	0.526	0.526	0
S-sets 3	Random	0.412	0.492	0.461	0.019	0.427	**0.493**	0.463	0.018	0.395	**0.493**	0.455	0.022	0.413	**0.493**	0.459	0.02
	K-Means++	0.431	0.492	0.467	0.018	0.429	**0.493**	0.465	0.017	0.409	**0.493**	0.459	0.02	0.428	**0.493**	0.46	0.018
	ROBIN(S)	0.431	0.466	0.462	0.01	0.452	0.467	0.464	0.005	0.423	0.468	0.457	0.015	0.422	0.465	0.458	0.014
	Maximin(S)	0.431	0.492	0.469	0.018	0.427	0.492	0.469	0.019	0.422	**0.493**	0.469	0.021	0.438	**0.493**	0.463	0.019
	Kaufman	0.492	0.492	0.492	0	0.492	0.492	0.492	0	0.493	**0.493**	0.493	0	0.493	**0.493**	0.493	0
	DK-Means++	0.493	**0.493**	0.493	0	0.493	**0.493**	0.493	0	0.493	**0.493**	0.493	0	0.493	**0.493**	0.493	0
	ROBIN(D)	0.466	0.466	0.466	0	0.467	0.467	0.467	0	0.464	0.464	0.464	0	0.464	0.464	0.464	0
	Maximin(D)	0.457	0.457	0.457	0	0.464	0.464	0.464	0	0.471	0.471	0.471	0	0.468	0.468	0.468	0
S-sets 4	Random	0.426	**0.48**	0.467	0.012	0.434	**0.48**	0.465	0.012	0.4	**0.479**	0.45	0.021	0.431	**0.48**	0.461	0.017
	K-Means++	0.431	**0.48**	0.469	0.012	0.433	**0.48**	0.469	0.012	0.412	**0.479**	0.456	0.017	0.43	**0.48**	0.461	0.013
	ROBIN(S)	0.457	**0.48**	0.468	0.007	0.435	0.47	0.458	0.012	0.45	0.466	0.461	0.006	0.439	0.459	0.452	0.007
	Maximin(S)	0.443	**0.48**	0.47	0.007	0.456	0.471	0.468	0.004	0.438	**0.479**	0.456	0.014	0.44	**0.48**	0.463	0.011
	Kaufman	0.48	**0.48**	0.48	0	0.48	**0.48**	0.48	0	0.458	0.458	0.458	0	0.48	**0.48**	0.48	0
	DK-Means++	0.48	**0.48**	0.48	0	0.48	**0.48**	0.48	0	0.479	**0.479**	0.479	0	0.48	**0.48**	0.48	0
	ROBIN(D)	0.48	**0.48**	0.48	0	0.435	0.435	0.435	0	0.466	0.466	0.466	0	0.439	0.439	0.439	0
	Maximin(D)	0.47	0.47	0.47	0	0.469	0.469	0.469	0	0.462	0.462	0.462	0	0.457	0.457	0.457	0

Table B.4: Comparison of the initialisation methods on real-world data sets based on Silhouette index. Each stochastic method (Random, K-Means++, ROBIN(S) and Maximin (S)) was executed 50 times and the minimum, maximum and mean performance is shown followed by the performance variation for four K-Means variations. For the maximum performance the cases where a method has achieved the maximum performance is shown in bold.

		K-Means (Hartigan-Wong)				K-Means (Lloyd)				K-Medians				Weiszfeld			
		min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std
Iris	Random	0.517	**0.553**	0.549	0.012	0.5	**0.553**	0.543	0.016	0.467	**0.551**	0.542	0.016	0.41	**0.51**	0.5	0.013
	K-Means++	0.517	**0.553**	0.551	0.009	0.5	**0.553**	0.548	0.011	0.526	**0.551**	0.55	0.006	0.5	**0.51**	0.5	0.009
	ROBIN(S)	0.553	**0.553**	0.553	0	0.551	0.551	0.551	0	0.551	**0.551**	0.551	0	0.51	**0.51**	0.51	0
	Maximin(S)	0.553	**0.553**	0.553	0	0.551	**0.553**	0.552	0.001	0.551	**0.551**	0.551	0	0.43	**0.51**	0.493	0.03
	Kaufman	0.553	**0.553**	0.553	0	0.551	0.551	0.551	0	0.551	**0.551**	0.551	0	0.51	**0.51**	0.51	0
	DK-Means++	0.553	**0.553**	0.553	0	0.551	0.551	0.551	0	0.551	**0.551**	0.551	0	0.51	**0.51**	0.51	0
	ROBIN(D)	0.553	**0.553**	0.553	0	0.551	0.551	0.551	0	0.551	**0.551**	0.551	0	0.51	**0.51**	0.51	0
	Maximin(D)	0.553	**0.553**	0.553	0	0.553	**0.553**	0.553	0	0.551	**0.551**	0.551	0	0.51	**0.51**	0.51	0
Ionosphere	Random	0.296	**0.296**	0.296	0	0.246	0.368	0.296	0.013	0.231	0.334	0.283	0.013	0.287	0.291	0.289	0.002
	K-Means++	0.296	**0.296**	0.296	0	0.266	0.296	0.295	0.006	0.246	**0.408**	0.286	0.015	0.258	0.291	0.289	0.007
	ROBIN(S)	0.296	**0.296**	0.296	0	0.296	0.296	0.296	0	0.284	0.284	0.284	0	0.291	0.291	0.291	0
	Maximin(S)	0.296	**0.296**	0.296	0	0.295	**0.408**	0.314	0.038	0.284	**0.408**	0.311	0.05	0.287	**0.408**	0.31	0.043
	Kaufman	0.296	**0.296**	0.296	0	0.295	0.295	0.295	0	0.284	0.284	0.284	0	0.287	0.287	0.287	0
	DK-Means++	0.296	**0.296**	0.296	0	0.296	0.296	0.296	0	0.284	0.284	0.284	0	0.291	0.291	0.291	0
	ROBIN(D)	0.296	**0.296**	0.296	0	0.296	0.296	0.296	0	0.284	0.284	0.284	0	0.291	0.291	0.291	0
	Maximin(D)	0.296	**0.296**	0.296	0	0.296	0.296	0.296	0	0.284	0.284	0.284	0	0.291	0.291	0.291	0
Wine	Random	0.548	**0.571**	0.57	0.005	0.54	**0.571**	0.57	0.005	0.566	**0.571**	0.568	0.002	0.548	**0.572**	0.568	0.009
	K-Means++	0.548	**0.571**	0.562	0.01	0.548	**0.571**	0.566	0.007	0.566	**0.571**	0.57	0.002	0.548	**0.572**	0.559	0.012
	ROBIN(S)	0.571	**0.571**	0.571	0	0.571	**0.571**	0.571	0	0.566	0.566	0.566	0	0.572	**0.572**	0.572	0
	Maximin(S)	0.553	**0.571**	0.558	0.008	0.553	**0.571**	0.561	0.005	0.571	**0.571**	0.571	0	0.548	**0.572**	0.555	0.011
	Kaufman	0.571	**0.571**	0.571	0	0.571	**0.571**	0.571	0	0.571	**0.571**	0.571	0	0.571	0.571	0.571	0
	DK-Means++	0.571	**0.571**	0.571	0	0.571	**0.571**	0.571	0	0.571	**0.571**	0.571	0	0.571	0.571	0.571	0
	ROBIN(D)	0.571	**0.571**	0.571	0	0.571	**0.571**	0.571	0	0.566	0.566	0.566	0	0.572	**0.572**	0.572	0
	Maximin(D)	0.548	0.548	0.548	0	0.56	0.56	0.56	0	0.571	**0.571**	0.571	0	0.548	0.548	0.548	0
Breast Cancer	Random	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	K-Means++	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	ROBIN(S)	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	Maximin(S)	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	Kaufman	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	DK-Means++	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	ROBIN(D)	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
	Maximin(D)	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.597	**0.597**	0.597	0	0.596	**0.596**	0.596	0
Glass	Random	0.192	0.456	0.43	0.056	0.194	0.452	0.345	0.09	0.137	0.442	0.268	0.077	0.181	0.417	0.24	0.057
	K-Means++	0.271	**0.587**	0.457	0.053	0.278	**0.585**	0.454	0.064	0.211	**0.592**	0.432	0.089	0.233	**0.593**	0.571	0.093
	ROBIN(S)	0.447	0.452	0.447	0.001	0.43	0.448	0.443	0.005	0.257	0.397	0.384	0.028	0.254	0.388	0.376	0.037
	Maximin(S)	0.555	**0.587**	0.582	0.009	0.43	**0.585**	0.563	0.048	0.433	**0.592**	0.576	0.03	0.401	**0.593**	0.576	0.039
	Kaufman	0.452	0.452	0.452	0	0.448	0.448	0.448	0	0.238	0.238	0.238	0	0.257	0.257	0.257	0
	DK-Means++	0.447	0.447	0.447	0	0.431	0.431	0.431	0	0.435	0.435	0.435	0	0.194	0.194	0.194	0
	ROBIN(D)	0.447	0.447	0.447	0	0.444	0.444	0.444	0	0.392	0.392	0.392	0	0.388	0.388	0.388	0
	Maximin(D)	0.584	0.584	0.584	0	0.583	0.583	0.583	0	0.58	0.58	0.58	0	0.58	0.58	0.58	0
Yeast	Random	0.142	0.186	0.166	0.011	0.136	0.189	0.164	0.011	0.117	0.175	0.144	0.012	0.13	0.177	0.148	0.01
	K-Means++	0.15	0.217	0.177	0.012	0.148	0.192	0.173	0.01	0.119	0.184	0.157	0.013	0.141	0.181	0.165	0.01
	ROBIN(S)	0.18	0.183	0.181	0.001	0.178	0.19	0.183	0.006	0.161	0.172	0.164	0.005	0.163	0.176	0.167	0.007
	Maximin(S)	0.189	**0.224**	0.202	0.01	0.173	**0.225**	0.204	0.014	0.166	**0.213**	0.194	0.018	0.176	**0.22**	0.2	0.017
	Kaufman	0.161	0.161	0.161	0	0.159	0.159	0.159	0	0.151	0.151	0.151	0	0.154	0.154	0.154	0
	DK-Means++	0.155	0.155	0.155	0	0.156	0.156	0.156	0	0.14	0.14	0.14	0	0.147	0.147	0.147	0
	ROBIN(D)	0.183	0.183	0.183	0	0.19	0.19	0.19	0	0.172	0.172	0.172	0	0.176	0.176	0.176	0
	Maximin(D)	0.192	0.192	0.192	0	0.191	0.191	0.191	0	0.175	0.175	0.175	0	0.182	0.182	0.182	0

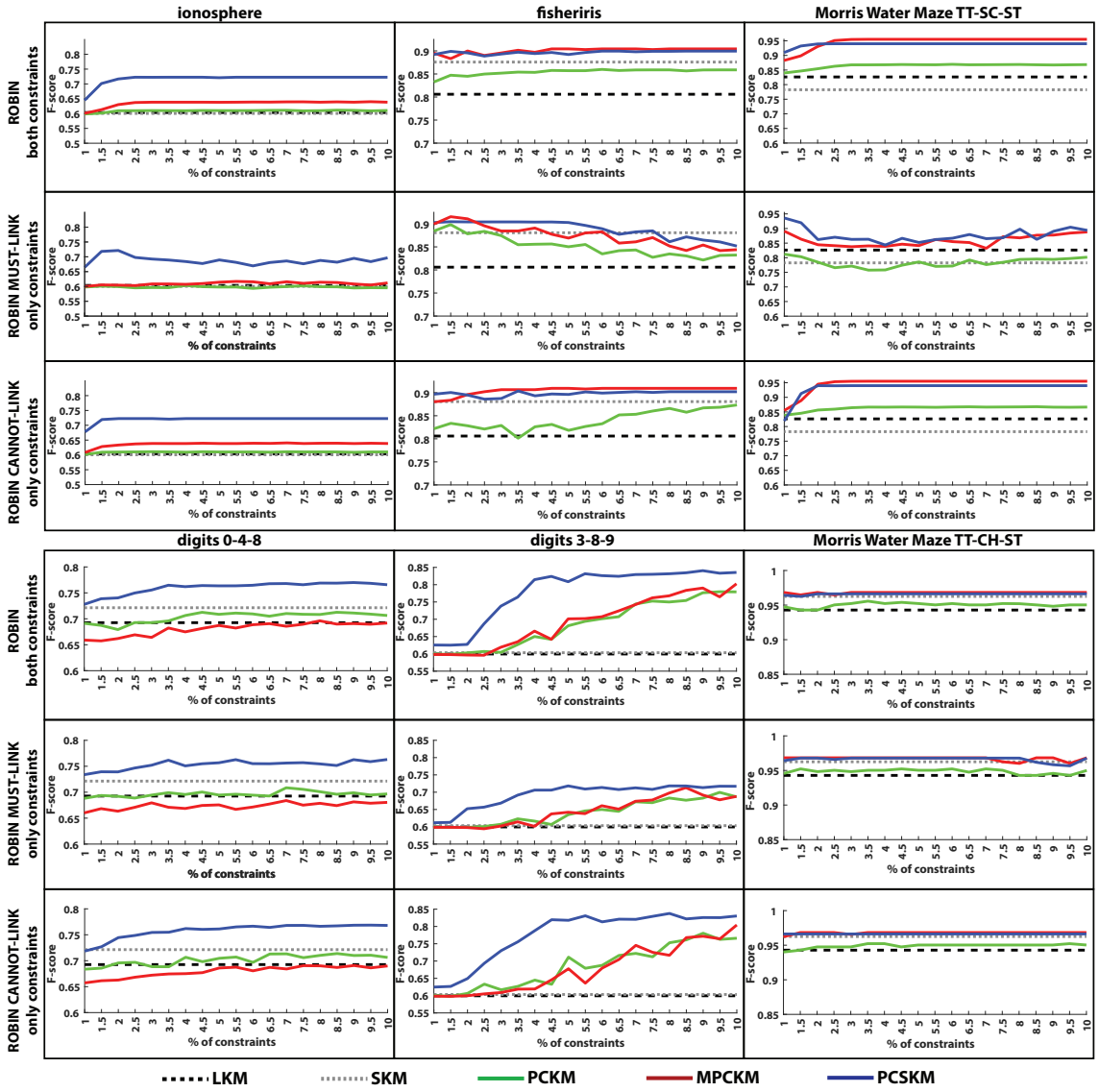


Figure B.4: Performance of PCSKM as opposed to other unsupervised and semi-supervised algorithms using the ROBIN initialisation method. Each row compares the algorithms over six data sets (ionosphere, fisheriris, digits 0-4-8 and 3-8-9 and Morris Water Maze TT-SC-ST and TT-CH-ST) using different types of constrains. First row (ROBIN both constraints): ROBIN was used for clustering initialisation and there has been a random selection from all the constraints, both MUST-LINK and CANNOT-LINK. Second and third rows (MUST-LINK, CANNOT-LINK): ROBIN initialisation was used and there was a random selection of only MUST-LINK or CANNOT-LINK. For the SKM and PCSKM the sparsity value with the best performance was selected.

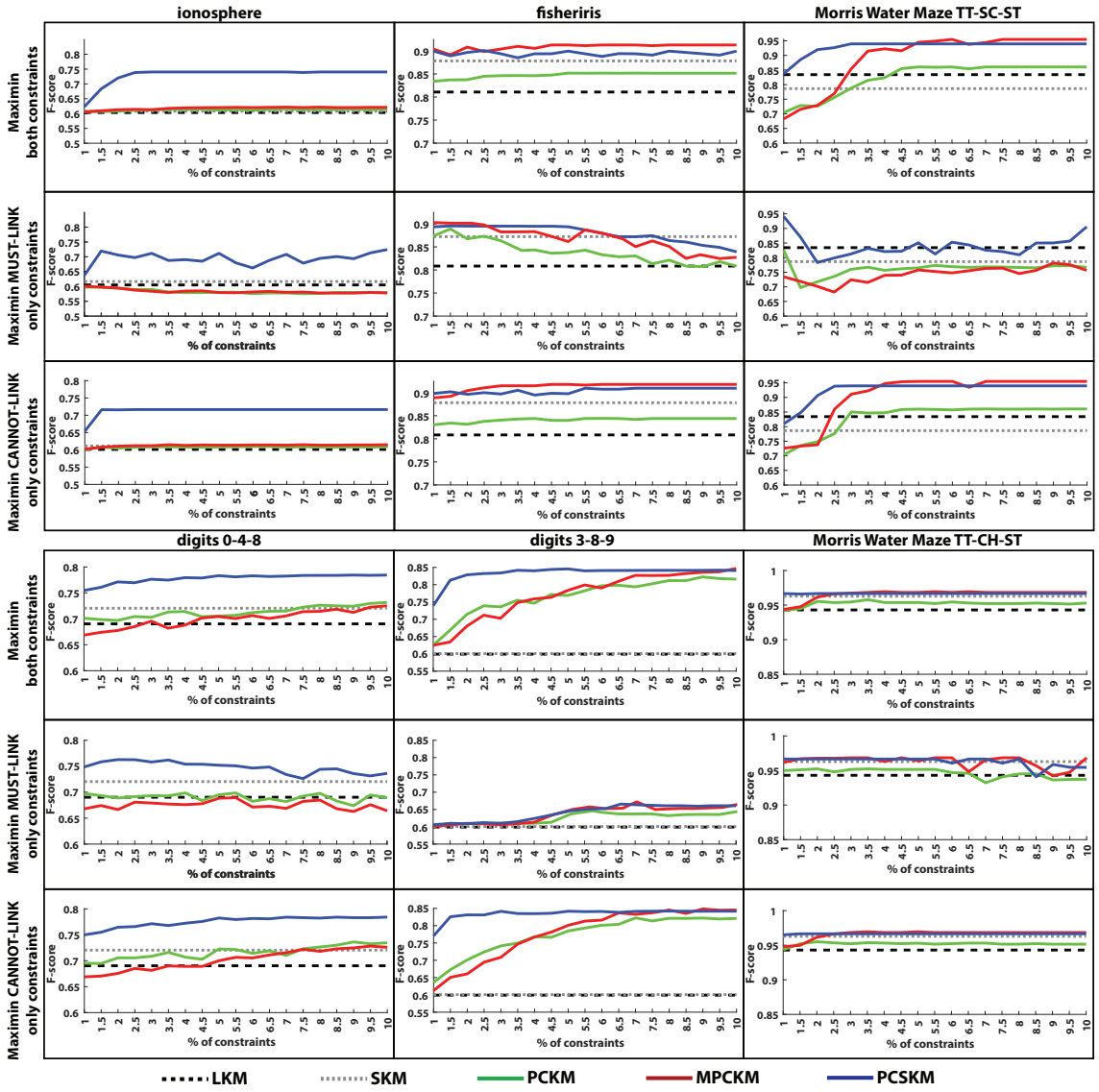


Figure B.5: Performance of PCSKM as opposed to other unsupervised and semi-supervised algorithms using the Maximin initialisation method. Each row compares the algorithms over six data sets (ionosphere, fisheriris, digits 0-4-8 and 3-8-9 and Morris Water Maze TT-SC-ST and TT-CH-ST) using different types of constrains. First row (ROBIN both constraints): ROBIN was used for clustering initialisation and there has been a random selection from all the constraints, both MUST-LINK and CANNOT-LINK. Second and third rows (MUST-LINK, CANNOT-LINK): Maximin initialisation was used and there was a random selection of only MUST-LINK or CANNOT-LINK. For the SKM and PCSKM the sparsity value with the best performance was selected.

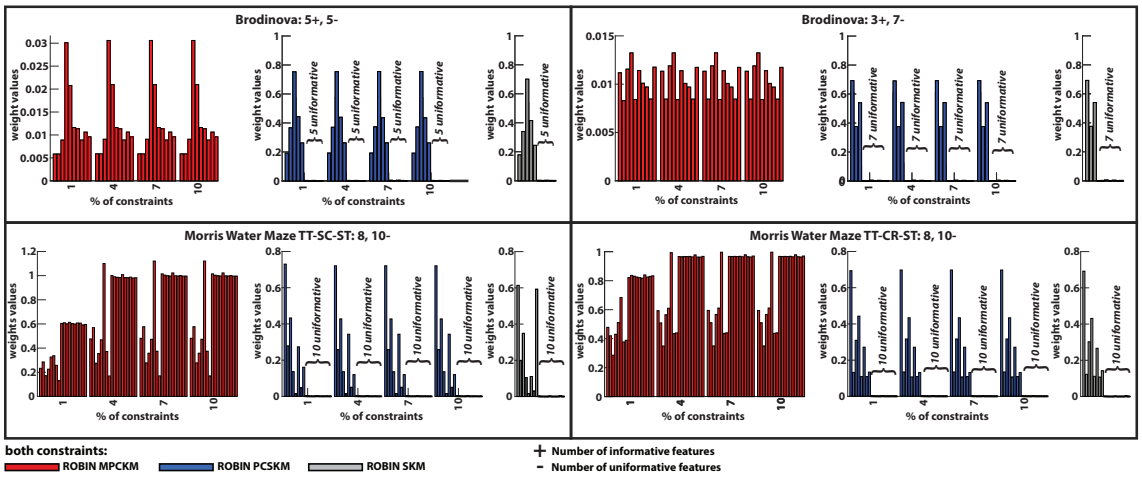


Figure B.6: PCSKM feature selection capabilities as opposed to other algorithms in synthetic and the Morris Water Maze data sets using the ROBIN initialisation method. Each bar plot shows the value of each one of the features of the data set over the number of constraints. Red bars: MPKM (semisupervised), blue bars PCSKM (semi-supervised), gray bars: SKM (unsupervised). Red bars: MPKM (semi-supervised), blue bars PCSKM (semi-supervised), gray bars: SKM (unsupervised). For SKM and PCSKM the bars show the average weight value of a feature over different sparsity ($\boldsymbol{s}$) values (from $\boldsymbol{s} = \mathbf{1.1}$ to $\boldsymbol{s} = \sqrt{\boldsymbol{p}}$, where $\boldsymbol{p}$ is the dimensionality of the data set, with step 0.2). The $+$ and $-$ signs indicate the number of informative and uninformative features (uninformative features are always plotted last). In the case of the Morris Water Maze the quality of the first 8 features is unknown but the last 10 are uninformative. The SKM and PCSKM correctly identifies the known uninformative features regardless of the s parameter value in all the cases. Specifically for the PCSKM the feature selection mechanism is not affected by the constraints. The MPCKM algorithm fails to show any indication about the feature quality based on the feature weights and in all the cases it uses the uninformative features. In the plots we show only the case when both type of constraints are used but we observe the same result for the other constraint types cases regardless of the used initialisation method.

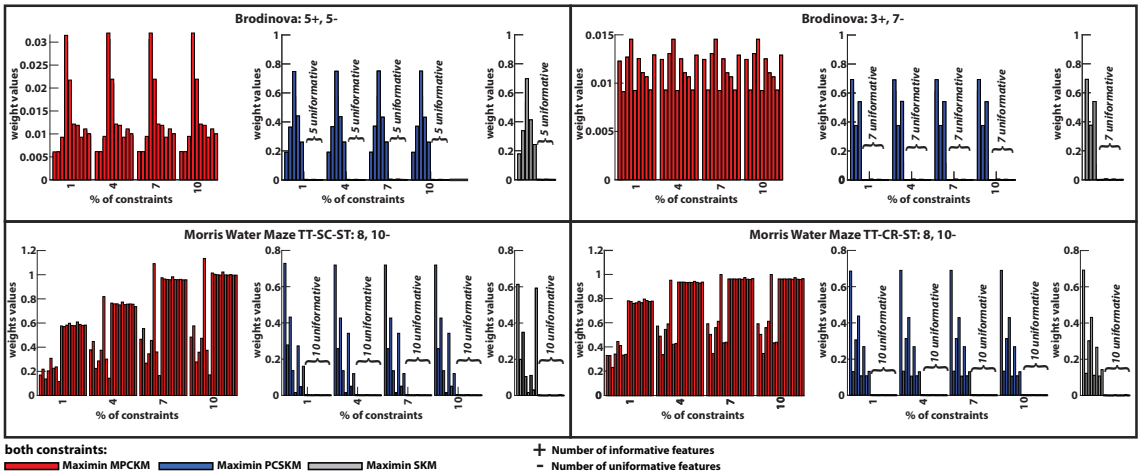


Figure B.7: PCSKM feature selection capabilities as opposed to other algorithms in synthetic and the Morris Water Maze data sets using the Maximin initialisation method. Each bar plot shows the value of each one of the features of the data set over the number of constraints. Red bars: MPKM (semisupervised), blue bars PCSKM (semi-supervised), gray bars: SKM (unsupervised). Red bars: MPKM (semi-supervised), blue bars PCSKM (semi-supervised), gray bars: SKM (unsupervised). For SKM and PCSKM the bars show the average weight value of a feature over different sparsity (s) values (from $s = 1.1$ to $s = \sqrt{p}$, where p is the dimensionality of the data set, with step 0.2). The $+$ and $-$ signs indicate the number of informative and uninformative features (uninformative features are always plotted last). In the case of the Morris Water Maze the quality of the first 8 features is unknown but the last 10 are uninformative. The SKM and PCSKM correctly identifies the known uninformative features regardless of the s parameter value in all the cases. Specifically for the PCSKM the feature selection mechanism is not affected by the constraints. The MPCKM algorithm fails to show any indication about the feature quality based on the feature weights and in all the cases it uses the uninformative features. In the plots we show only the case when both type of constraints are used but we observe the same result for the other constraint types cases regardless of the used initialisation method.

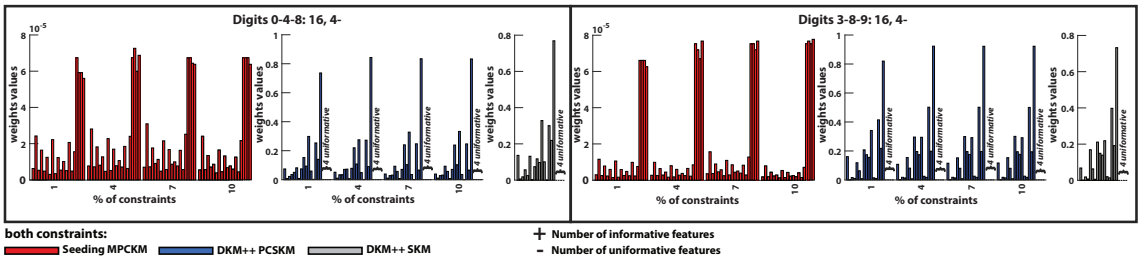


Figure B.8: PCSKM feature selection capabilities as opposed to other algorithms in the digits data sets contaminated with 4 uninformative features.

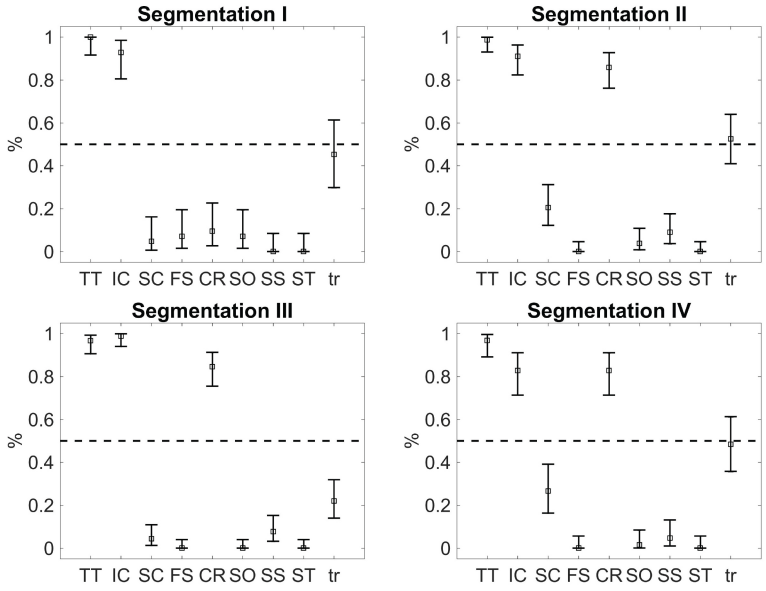
Appendix C

C.1 Agreement matrix

Classifiers																											
Classifiers	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	...	91
1	100	61.39	36.62	46.22	39.30	34.95	44.93	49.03	37.07	45.36	48.75	44.32	44.49	34.82	47.64	45.19	44.35	41.10	41.10	40.28	38.69	35.68	38.46	45.94	...	39.01
2	61.39	100	39.87	44.28	40.06	33.67	41.38	50.01	35.47	43.82	48.02	44.63	43.07	33.10	49.25	42.49	41.29	41.26	41.26	41.93	38.74	37.26	38.27	43.07	...	40.78
3	36.62	39.87	100	50.63	54.55	53.17	41.05	37.23	49.45	38.75	41.19	35.26	38.91	46.95	42.06	40.79	41.68	40.79	40.79	41.44	48.07	47.87	43.24	42.61	...	36.66
4	46.22	44.28	50.63	100	46.25	47.10	51.14	45.10	44.65	51.04	49.85	42.63	46.79	40.52	48.84	49.48	48.56	48.77	48.77	49.98	40.38	41.40	47.61	49.14	...	44.46
5	39.30	40.06	54.55	46.25	100	54.55	45.15	43.33	51.98	39.77	41.82	37.60	38.03	48.39	41.72	40.15	36.52	40.51	40.51	41.72	46.59	48.51	43.48	39.68	...	37.48
6	34.95	33.67	53.17	47.10	54.55	100	46.23	38.06	47.88	44.68	38.89	34.48	36.26	47.70	40.21	39.81	38.73	39.61	39.61	43.10	45.54	45.86	46.72	39.36	...	37.57
7	44.93	41.38	41.05	51.14	45.15	46.23	100	43.67	40.53	47.15	43.83	40.69	43.72	40.01	45.29	45.94	46.14	46.70	46.70	48.72	40.62	42.03	47.26	44.40	...	40.97
8	49.03	50.01	37.23	45.10	43.33	38.06	43.67	100	37.31	46.71	53.86	46.17	46.87	36.05	51.12	48.57	46.12	42.14	42.14	44.88	40.99	38.67	39.14	43.34	...	44.01
9	37.07	35.47	49.45	44.65	51.98	47.88	40.53	37.31	100	48.30	38.78	32.52	32.55	45.65	40.49	39.47	40.45	42.00	42.00	42.40	44.15	49.29	42.26	40.27	...	36.47
10	45.36	43.82	38.75	51.04	39.77	44.68	47.15	46.71	48.30	100	52.48	43.91	42.74	40.83	51.29	54.69	54.55	52.60	52.60	52.86	44.93	42.87	51.15	54.85	...	48.60
11	48.75	48.02	41.19	49.85	41.82	38.89	43.83	53.86	38.78	52.48	100	53.24	50.19	39.53	53.56	46.66	46.36	44.93	44.93	49.03	38.78	35.48	42.94	47.80	...	44.48
12	44.32	44.63	35.26	42.63	37.60	34.48	40.69	46.17	32.52	43.91	53.24	100	54.00	45.74	45.45	46.31	43.91	42.40	42.40	45.14	36.21	34.57	42.39	43.54	...	42.74
13	44.49	43.07	38.91	46.79	38.03	36.26	43.72	46.87	32.55	42.74	50.19	54.00	100	39.25	50.85	48.34	47.99	44.33	44.33	46.63	39.24	37.39	39.74	44.58	...	44.21
14	34.82	33.10	46.95	40.52	48.39	47.70	40.01	36.05	45.65	40.83	39.53	45.74	39.25	100	37.91	40.45	40.05	40.16	40.16	44.44	44.85	44.83	42.67	39.30	...	39.93
15	47.64	49.25	42.06	48.84	41.72	40.21	45.29	51.12	40.49	51.29	53.56	45.45	50.85	37.91	100	52.08	53.64	49.56	49.56	51.47	44.87	43.78	48.13	52.37	...	47.13
16	45.19	42.49	40.79	49.48	40.15	39.81	45.94	48.57	39.47	54.69	46.66	46.31	48.34	40.45	52.08	100	70.70	60.17	60.17	58.03	48.61	49.53	47.45	54.12	...	47.43
17	44.35	41.29	41.68	48.56	36.52	38.73	46.14	46.12	40.45	54.55	46.36	43.91	47.99	40.05	53.64	70.70	100	58.80	58.80	58.74	52.54	50.80	49.96	58.37	...	50.39
18	41.10	41.26	40.79	48.77	40.51	39.61	46.70	42.14	42.00	52.60	44.93	42.40	44.33	40.16	49.56	60.17	58.80	100	100.0	60.64	47.34	48.82	51.50	53.96	...	50.62
19	41.10	41.26	40.79	48.77	40.51	39.61	46.70	42.14	42.00	52.60	44.93	42.40	44.33	40.16	49.56	60.17	58.80	100.0	100	60.64	47.34	48.82	51.50	53.96	...	50.62
20	40.28	41.93	41.44	49.98	41.72	43.10	48.72	44.88	42.40	52.86	49.03	45.14	46.63	44.44	51.47	58.03	58.74	60.64	60.64	100	49.62	49.92	55.04	55.79	...	51.98
21	38.69	38.74	48.07	40.38	46.59	45.54	40.62	40.99	44.15	44.93	38.78	36.21	39.24	44.85	44.87	48.61	52.54	47.34	47.34	49.62	100	64.70	51.90	50.11	...	42.72
22	35.68	37.26	47.87	41.40	48.51	45.86	42.03	38.67	49.29	42.87	35.48	34.57	37.39	44.83	43.78	49.53	50.80	48.82	48.82	49.92	64.70	100	48.68	49.35	...	41.14
23	38.46	38.27	43.24	47.61	43.48	46.72	47.26	39.14	42.26	51.15	42.94	42.39	39.74	42.67	48.13	47.45	49.96	51.50	51.50	55.04	51.90	48.68	100	60.08	...	47.97
24	45.94	43.07	42.61	49.14	39.68	39.36	44.40	43.34	40.27	54.85	47.80	43.54	44.58	39.30	52.37	54.12	58.37	53.96	53.96	55.79	50.11	49.35	60.08	100	...	52.84
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	100	...
91	39.01	40.78	36.66	44.46	37.48	37.57	40.97	44.01	36.47	48.60	44.48	42.74	44.21	39.93	47.13	47.43	50.39	50.62	50.62	51.98	42.72	41.14	47.97	52.84	...	100

Figure C.1: Agreement matrix for the classifiers of Segmentation III. The classifier of each column is being compared with the classifier of each row. The comparison is based on the percentage of segments which both classifiers agree belong to the same class. The diagonal values of the matrix indicate 100% agreement since each classifier is compared with itself.

C.2 Results of each segmentation without the smoothing function



Ensemble Result (Friedman test p-values per strategy and transitions)									
	TT	**IC**	**SC**	**FS**	**CR**	**SO**	**SS**	**ST**	**tr**
I	**0.008**	**0.007**	0.469	0.196	0.159	0.724	0.240	0.375	0.061
II	**0.004**	**0.007**	0.164	0.355	**0.006**	0.756	0.201	0.284	0.066
III	**0.008**	**0.006**	0.335	0.946	**0.012**	0.844	0.220	0.584	0.161
IV	**0.012**	**0.014**	0.178	0.899	**0.005**	0.895	0.237	0.301	**0.035**

Figure C.2: Conclusive pre-smoothing results from the classification of each segmentation configuration. Considering segments as continuous parts of the trajectories ignoring the overlapping provides consistent results when differences between the implemented strategies of the groups are being investigated but creates an overestimation on the number of transitions between strategies. Each plot shows the 95% binomial confidence intervals for the classifiers of each segmentation regarding their agreement if there is significant difference between the two animal groups (i.e Friedman test p-value < 0.05) on each strategy and strategy transitions or not. Squares indicate the mean of the classifiers that shows that there is a significant difference in this particular case; errorbars are the 95% confidence intervals; the dashed line indicates the threshold of interest (0.5 or 50%). The table below the plots shows the Friedman test p-values (upper table) and the equivalent Friedman's chi-square statistic (lower table) for the classification result of the ensemble; in all cases $k = 2$, control and stress columns. Segmentation configurations are arranged in columns and strategies in rows; each element has the relevant p-value and chi-square statistic and bold cells indicate significant difference, i.e. p-value < 0.05. In order to be confident that there is indeed a significant difference between the two animal groups on each strategy and the strategy transitions the confidence intervals should be clearly above 0.5 (or 50%). Compared to the results in the main manuscript, we see that the smoothing function which maps the segments to the full swimming paths is actually beneficial for revealing the animal transitions between strategies. Other than that, the results lead to the same conclusions.

	Segmentation I	Segmentation II	Segmentation III	Segmentation IV
	Classifiers			
Unclassified (%) Segments	24.8	24.3	30.0	29.0
Agreement (%)	53.2	55.5	48.8	52.1
	Ensemble(s)			
Unclassified (%) Segments	1.2	0.7	0.8	1.1
Agreement (%)	84.7	83.3	79.8	80.0

Table C.1: Classification statistics for the four segmentation configurations prior to smoothing. In comparison with the results of the main manuscript we see that the percentage of unclassified segments among the classifiers is higher and the agreement between them lower. However, the ensemble (or ensembles in case of the agreement) again nearly nullifies the unclassified segments and significantly boosts the agreement percentage.

C.3 Ensemble results of each segmentation

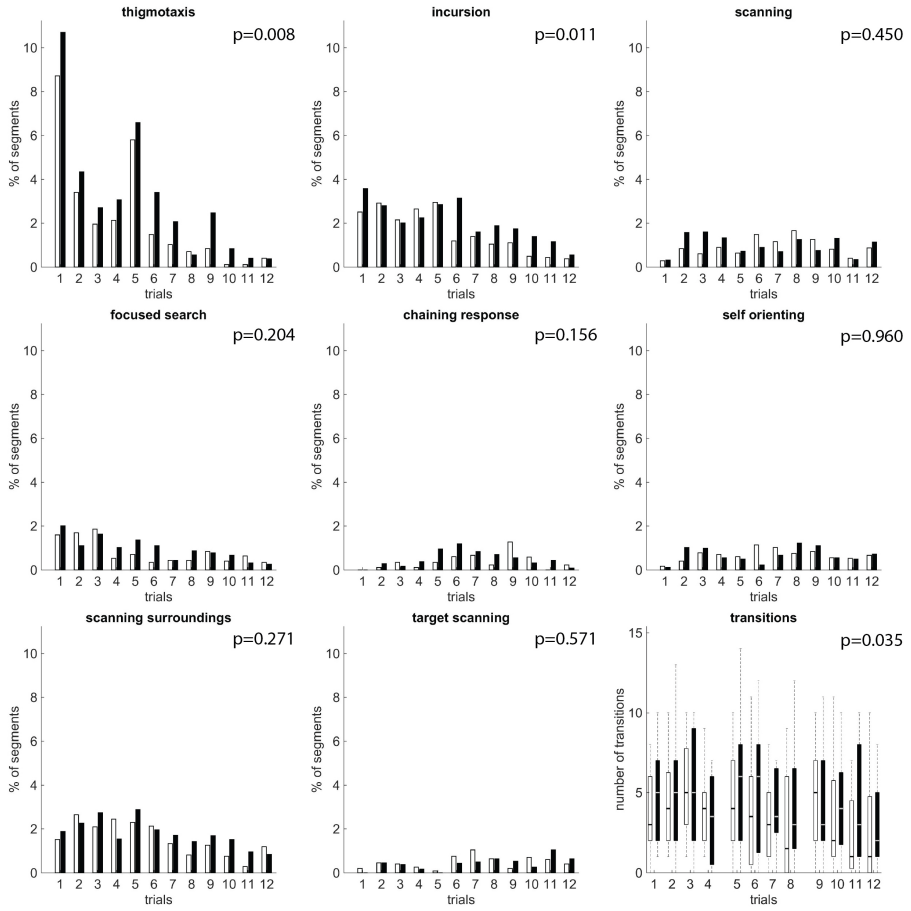


Figure C.3: Percentage of segments falling under each strategy for the stressed (black) and control (white) animal groups over each trial for the Segmentation I. All the animals were tested for a set of 12 trials divided in 3 sessions (days). Each segment is considered to be of a length equal to the arena radius (100cm). For the transitions: bars represent the first and third quartiles of the data; the black (control group) or white (stressed group) horizontal lines denotes the median, crosses are the outliers and whiskers indicate the minimum and the maximum values. The Friedman test p-value (shown on the top right) was used to compare both animal groups for the complete set of trials. According to the plots, stressed animals produce longer paths since the average number of strategy implementations is higher than in the control group. Thigmotaxis and Incursion strategies show a clear difference in favor of the stressed group along with the strategy transitions. This Segmentation configuration fails to reveal significant differences on the Chaining Response because of the segment length which causes some rarer strategies to disappear.

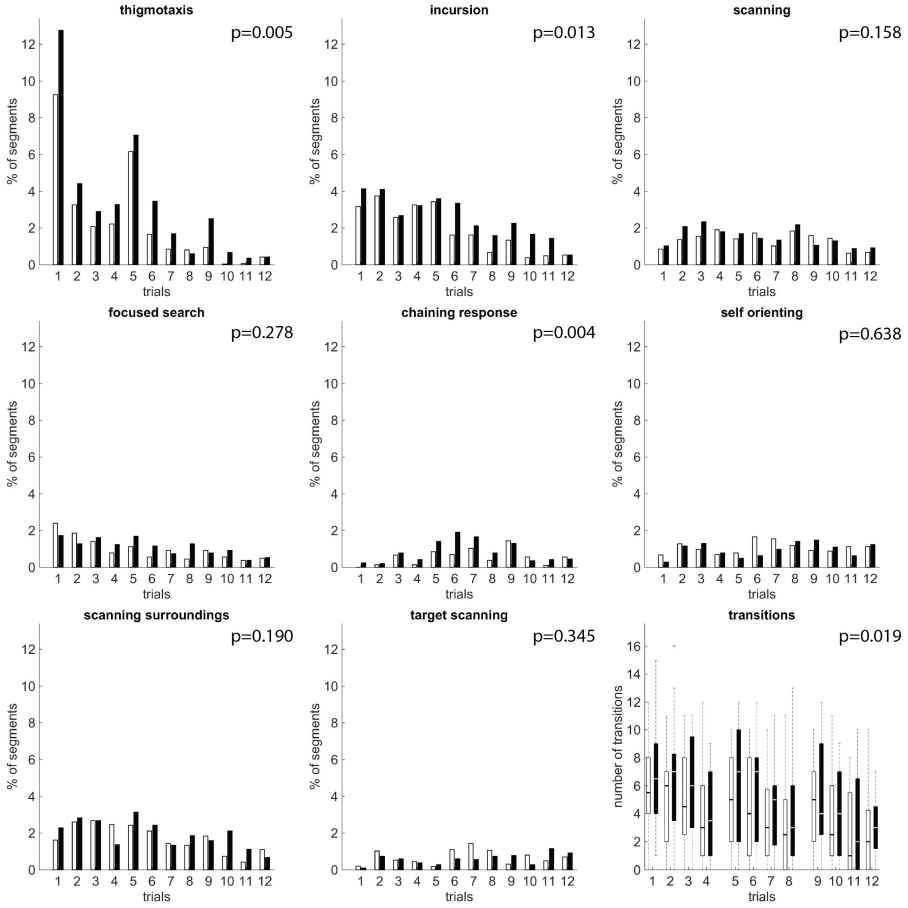


Figure C.4: Percentage of segments falling under each strategy for the stressed (black) and control (white) animal groups over each trial for the Segmentation II. All the animals were tested for a set of 12 trials divided in 3 sessions (days). Each segment is considered to be of a length equal to the arena radius (100cm). For the transitions: bars represent the first and third quartiles of the data; the black (control group) or white (stressed group) horizontal lines denotes the median, crosses are the outliers and whiskers indicate the minimum and the maximum values. The Friedman test p-value (shown on the top right) was used to compare both animal groups for the complete set of trials. According to the plots, stressed animals produce longer paths since the average number of strategy implementations is higher than in the control group. Thigmotaxis and Incursion strategies show a clear difference in favor of the stressed group along with Chaining Response. The number of transitions between strategies shows that the stressed animals change their behaviour more often within single trials.

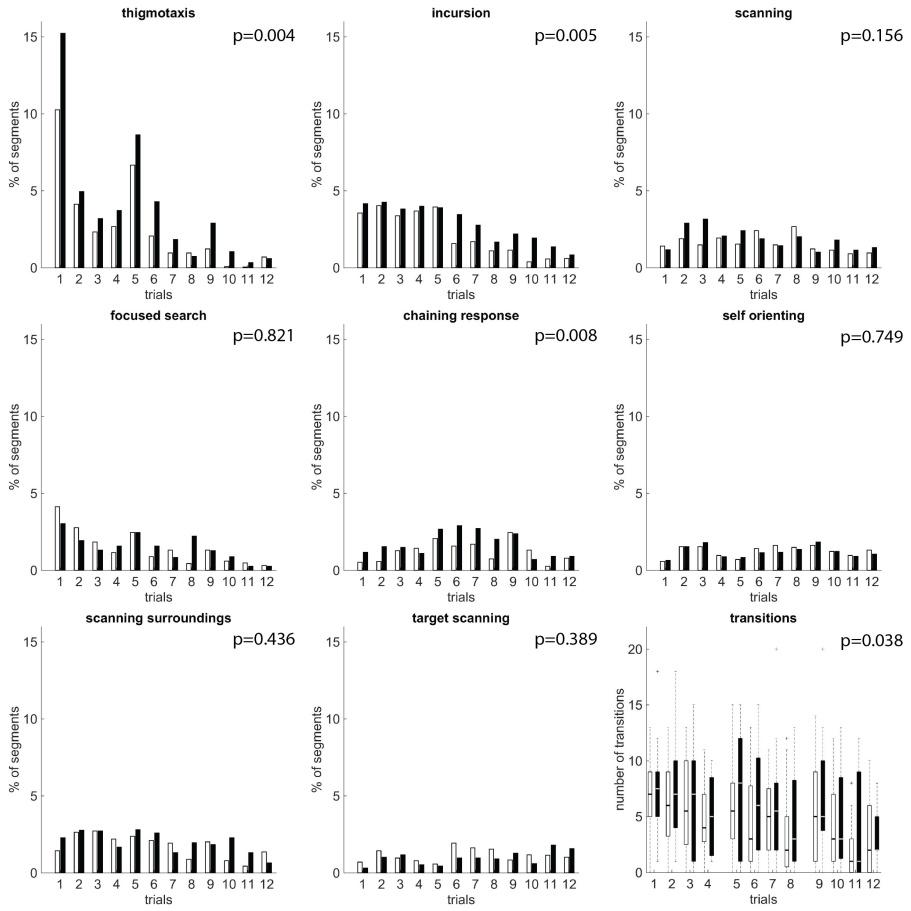


Figure C.5: Percentage of segments falling under each strategy for the stressed (black) and control (white) animal groups over each trial for the Segmentation IV. All the animals were tested for a set of 12 trials divided in 3 sessions (days). Each segment is considered to be of a length equal to the arena radius (100cm). For the transitions: bars represent the first and third quartiles of the data; the black (control group) or white (stressed group) horizontal lines denotes the median, crosses are the outliers and whiskers indicate the minimum and the maximum values. The Friedman test p-value (shown on the top right) was used to compare both animal groups for the complete set of trials. According to the plots, stressed animals produce longer paths since the average number of strategy implementations is higher than in the control group. Thigmotaxis and Incursion strategies show a clear difference in favor of the stressed group along with Chaining Response. The number of transitions between strategies shows that the stressed animals change their behaviour more often within single trials.

C.4 Further application: strategy distributions on the probe trials

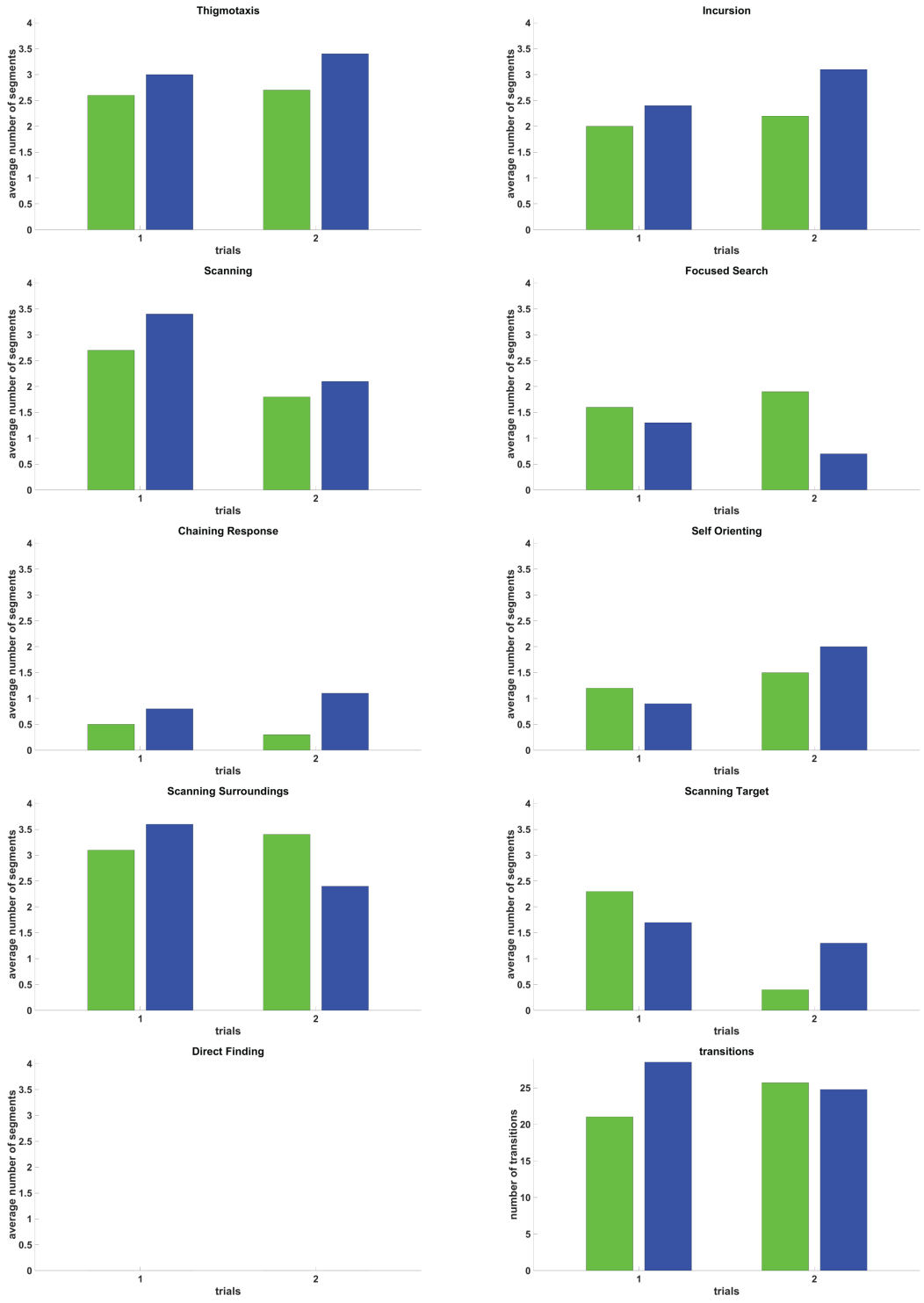


Figure C.6: Comparison between Low (green bars) and Intermediate (blue bars) animal groups on the *probe1* (trial 1) and *probe2* (trial 2) experimental procedures. Statistical analysis testing wasn't performed for the probe trials since the Friedman test requires at least three different occasion measurement on each group.

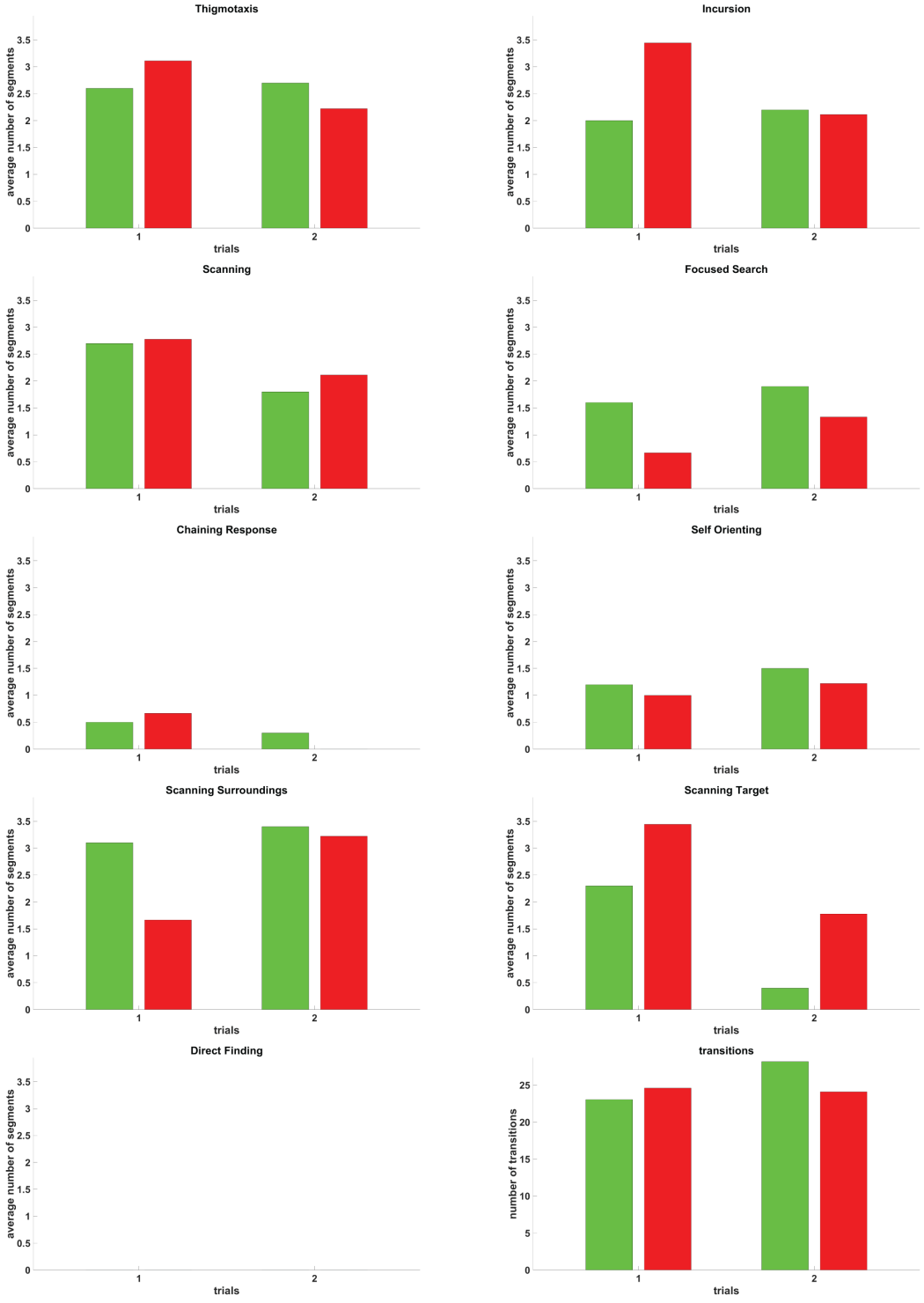


Figure C.7: Comparison between Low (green bars) and High (red bars) animal groups on the *probe1* (trial 1) and *probe2* (trial 2) experimental procedures. Statistical analysis testing wasn't performed for the probe trials since the Friedman test requires at least three different occasion measurement on each group.

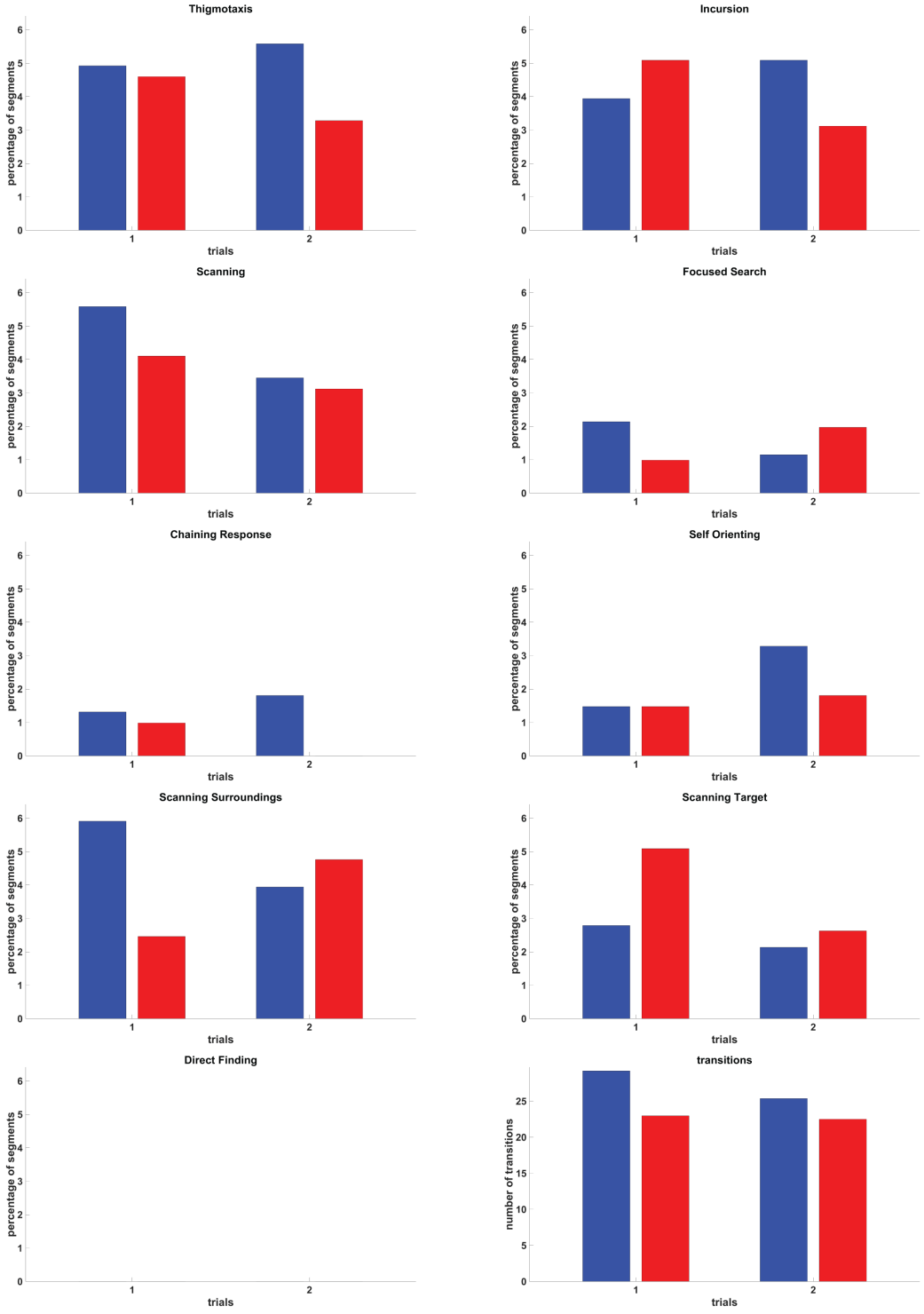


Figure C.8: Comparison between Intermediate (blue bars) and High (red bars) animal groups on the *probe1* (trial 1) and *probe2* (trial 2) experimental procedures. Statistical analysis testing wasn't performed for the probe trials since the Friedman test requires at least three different occasion measurement on each group.

Bibliography

Acharya, A., Hruschka, E. R., Ghosh, J. and Acharyya, S. [2011], C 3e: A framework for combining ensembles of classifiers and clusterers, *in* 'International Workshop on Multiple Classifier Systems', Springer, pp. 269–278.

Aggarwal, C. C. [2014], *Data classification: algorithms and applications*, CRC Press.

Al Hasan, M., Chaoji, V., Salem, S. and Zaki, M. J. [2009], 'Robust partitional clustering by outlier and density insensitive seeding', *Pattern Recognition Letters* **30**(11), 994–1002.

Almeida, P. J., Vieira, M. V., Kajin, M., Forero-Medina, G. and Cerqueira, R. [2010], 'Indices of movement behaviour: conceptual background, effects of scale and location errors', *Zoologia* **27**(5).

Araujo, J., Maximino, C., de Brito, T. M., da Silva, A. W. B., Oliveira, K. R. M., Batista, E. d. J. O., Morato, S., Herculano, A. M. and Gouveia, A. [2012], Behavioral and pharmacological aspects of anxiety in the light/dark preference test, *in* 'Zebrafish protocols for neurobehavioral research', Springer, pp. 191–202.

Arlot, S., Celisse, A. et al. [2010], 'A survey of cross-validation procedures for model selection', *Statistics surveys* **4**, 40–79.

Armañanzas, R. and Ascoli, G. A. [2015], 'Towards the automatic classification of neurons', *Trends in neurosciences* **38**(5), 307–318.

Arthur, D. and Vassilvitskii, S. [2007], k-means++: The advantages of careful seeding, *in* 'Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms', Society for Industrial and Applied Mathematics, pp. 1027–1035.

Aston-Jones, G., Rajkowski, J. and Cohen, J. [2000], 'Locus coeruleus and regulation of behavioral flexibility and attention', *Progress in brain research* **126**, 165–182.

Asuncion, A. and Newman, D. [2007], 'Uci machine learning repository'.

Aulich, D. [1976], 'Escape versus exploratory activity: An interpretation of rats' behaviour in the open field and a light-dark preference test', *Behavioural Processes* **1**(2), 153–164.

Avdesh, A., Martin-Iverson, M. T., Mondal, A., Chen, M., Askraba, S., Morgan, N., Lardelli, M., Groth, D. M., Verdile, G. and Martins, R. N. [2012], 'Evaluation of color preference in zebrafish for learning and memory', *Journal of Alzheimer's Disease* **28**(2), 459–469.

Baldi, P. [2012], Autoencoders, unsupervised learning, and deep architectures, *in* 'Proceedings of ICML workshop on unsupervised and transfer learning', pp. 37–49.

Bar-Hillel, A., Hertz, T., Shental, N. and Weinshall, D. [2003], Learning distance functions using equivalence relations, *in* 'Proceedings of the 20th International Conference on Machine Learning (ICML-03)', pp. 11–18.

Barraquand, F. and Benhamou, S. [2008], 'Animal movements in heterogeneous landscapes: identifying profitable places and homogeneous movement bouts', *Ecology* **89**(12), 3336–3348.

Basu, S., Banerjee, A. and Mooney, R. [2002], Semi-supervised clustering by seeding, *in* 'In Proceedings of 19th International Conference on Machine Learning (ICML-2002', Citeseer.

Basu, S., Banerjee, A. and Mooney, R. J. [2004], Active semi-supervision for pairwise constrained clustering, *in* 'Proceedings of the 2004 SIAM international conference on data mining', SIAM, pp. 333–344.

Basu, S., Bilenko, M. and Mooney, R. J. [2003], Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering, *in* 'Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining', Citeseer, pp. 42–49.

Benhamou, S. [2004], 'How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension?', *Journal of theoretical biology* **229**(2), 209–220.

Benhamou, S. [2014], 'Of scales and stationarity in animal movements', *Ecology letters* **17**(3), 261–272.

Bilenko, M., Basu, S. and Mooney, R. J. [2004], Integrating constraints and metric learning in semi-supervised clustering, *in* 'Proceedings of the twenty-first international conference on Machine learning', ACM, p. 11.

Biswas, A. and Jacobs, D. [2014], 'Active subclustering', *Computer Vision and Image Understanding* **125**, 72–84.

Blaser, R., Chadwick, L. and McGinnis, G. [2010], 'Behavioral measures of anxiety in zebrafish (danio rerio)', *Behavioural brain research* **208**(1), 56–62.

Blaser, R. and Penalosa, Y. [2011], 'Stimuli affecting zebrafish (danio rerio) behavior in the light/dark preference test', *Physiology & behavior* **104**(5), 831–837.

Blumstein, L. and Crawley, J. [1983], 'Further characterization of a simple, automated exploratory model for the anxiolytic effects of benzodiazepines', *Pharmacology Biochemistry and Behavior* **18**(1), 37–40.

Boal, J. G., Dunham, A. W., Williams, K. T. and Hanlon, R. T. [2000], 'Experimental evidence for spatial learning in octopuses (octopus bimaculoides).', *Journal of Comparative Psychology* **114**(3), 246.

Bourin, M. and Hascoët, M. [2003], 'The mouse light/dark box test', *European journal of pharmacology* **463**(1-3), 55–65.

Bouziane, H., Messabih, B. and Chouarfia, A. [2011], 'Profiles and majority voting-based ensemble method for protein secondary structure prediction', *Evolutionary bioinformatics online* **7**, 171.

Boyd, S. and Vandenberghe, L. [2004], *Convex optimization*, Cambridge university press.

Brandeis, R., Brandys, Y. and Yehuda, S. [1989], 'The use of the morris water maze in the study of memory and learning', *International Journal of Neuroscience* **48**(1-2), 29–69.

Braun, A. A., Graham, D. L., Schaefer, T. L., Vorhees, C. V. and Williams, M. T. [2012], 'Dorsal striatal dopamine depletion impairs both allocentric and egocentric navigation in rats', *Neurobiology of learning and memory* **97**(4), 402–408.

Breunig, M. M., Kriegel, H.-P., Ng, R. T. and Sander, J. [2000], Lof: identifying density-based local outliers, *in* 'ACM sigmod record', Vol. 29, ACM, pp. 93–104.

Bro, R. and Smilde, A. K. [2014], 'Principal component analysis', *Analytical Methods* **6**(9), 2812–2831.

Brodinová, Š., Filzmoser, P., Ortner, T., Breiteneder, C. and Rohm, M. [2017], 'Robust and sparse k-means clustering for high-dimensional data', *Advances in Data Analysis and Classification* pp. 1–28.

Bromley-Brits, K., Deng, Y. and Song, W. [2011], 'Morris water maze test for learning and memory deficits in alzheimer's disease model mice', *JoVE (Journal of Visualized Experiments)* (53), e2920–e2920.

Brown, A. E., Yemini, E. I., Grundy, L. J., Jucikas, T. and Schafer, W. R. [2013], 'A dictionary of behavioral motifs reveals clusters of genes affecting caenorhabditis elegans locomotion', *Proceedings of the National Academy of Sciences* **110**(2), 791–796.

Brown, L. D., Cai, T. T. and DasGupta, A. [2001], 'Interval estimation for a binomial proportion', *Statistical science* pp. 101–117.

Brusco, M. J., Shireman, E. and Steinley, D. [2017], 'A comparison of latent class, k-means, and k-median methods for clustering dichotomous data.', *Psychological methods* **22**(3), 563.

Bryda, E. C. [2013], 'The mighty mouse: the impact of rodents on advances in biomedical research', *Missouri medicine* **110**(3), 207.

Buccafusco, J. J. [2000], *Methods of behavior analysis in neuroscience*, CRC Press.

Burešová, O., Krekule, I., Zahalka, A. and Bureš, J. [1985], 'On-demand platform improves accuracy of the morris water maze procedure', *Journal of neuroscience methods* **15**(1), 63–72.

Celebi, M. E., Kingravi, H. A. and Vela, P. A. [2013], 'A comparative study of efficient initialization methods for the k-means clustering algorithm', *Expert systems with applications* **40**(1), 200–210.

Champagne, D. L., Hoefnagels, C. C., de Kloet, R. E. and Richardson, M. K. [2010], 'Translating rodent behavioral repertoire to zebrafish (danio rerio): relevance for stress research', *Behavioural brain research* **214**(2), 332–342.

Chang, W.-C. [1983], 'On using principal components before separating a mixture of two multivariate normal distributions', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **32**(3), 267–275.

Chatfield, M. and Mander, A. [2009], 'The skillings–mack test (friedman test when there are missing data)', *The Stata Journal* **9**(2), 299.

Chhabria, K., Plant, K., Bandmann, O., Wilkinson, R. N., Martin, C., Kugler, E., Armitage, P. A., Santoscoy, P. L., Cunliffe, V. T., Huisken, J. et al. [2018], 'The effect of hyperglycemia on neurovascular coupling and cerebrovascular patterning in zebrafish', *Journal of Cerebral Blood Flow & Metabolism* p. 0271678X18810615.

Chhabria, K., Vouros, A., Gray, C., MacDonald, R. B., Jiang, Z., Wilkinson, R. N., Plant, K., Vasilaki, E., Howarth, C. and Chico, T. [2019], 'Sodium nitroprusside prevents the detrimental effects of glucose on the neurovascular unit and behaviour in zebrafish', *bioRxiv* p. 576942.

Cooke, M. B., O'Leary, T. P., Harris, P., Brown, R. E. and Snyder, J. S. [2019], 'Pathfinder: open source software for analyzing spatial navigation search strategies', *F1000Research* **8**(1521), 1521.

Cortes, C. and Vapnik, V. [1995], 'Support-vector networks', *Machine learning* **20**(3), 273–297.

Crawley, J. and Davis, L. [1982], 'Baseline exploratory activity predicts anxiolytic responsiveness to diazepam in five mouse strains', *Brain research bulletin* **8**(6), 609–612.

Crawley, J. and Goodwin, F. K. [1980], 'Preliminary report of a simple animal behavior model for the anxiolytic effects of benzodiazepines', *Pharmacology Biochemistry and Behavior* **13**(2), 167–170.

Dalm, S., Grootendorst, J., De Kloet, E. R. and Oitzl, M. S. [2000], 'Quantification of swim patterns in the morris water maze', *Behavior Research Methods, Instruments, & Computers* **32**(1), 134–139.

Daugherty, A. M., Bender, A. R., Yuan, P. and Raz, N. [2015], 'Changes in search path complexity and length during learning of a virtual water maze: Age differences and differential associations with hippocampal subfield volumes', *Cerebral Cortex* p. bhv061.

Del Carratore, F., Schmidt, K., Vinaixa, M., Hollywood, K. A., Greenland-Bews, C., Takano, E., Rogers, S. and Breitling, R. [2019], 'Integrated probabilistic annotation: a bayesian-based annotation method for metabolomic profiles integrating

biochemical connections, isotope patterns, and adduct relationships', *Analytical chemistry* **91**(20), 12799–12807.

Demiriz, A., Bennett, K. P. and Embrechts, M. J. [1999], 'Semi-supervised clustering using genetic algorithms', *Artificial neural networks in engineering (ANNIE-99)* pp. 809–814.

DePasquale, C. and Leri, J. [2018], 'The influence of exercise on anxiety-like behavior in zebrafish (danio rerio)', *Behavioural processes* **157**, 638–644.

D'Hooge, R. and De Deyn, P. P. [2001], 'Applications of the morris water maze in the study of learning and memory', *Brain research reviews* **36**(1), 60–90.

Ding, C. and He, X. [2004], K-means clustering via principal component analysis, *in* 'Proceedings of the twenty-first international conference on Machine learning', ACM, p. 29.

Drysdale, A. T., Grosenick, L., Downar, J., Dunlop, K., Mansouri, F., Meng, Y., Fetcho, R. N., Zebley, B., Oathes, D. J., Etkin, A. et al. [2017], 'Resting-state connectivity biomarkers define neurophysiological subtypes of depression', *Nature medicine* **23**(1), 28.

Dudoit, S. and Fridlyand, J. [2002], 'A prediction-based resampling method for estimating the number of clusters in a dataset', *Genome biology* **3**(7), research0036–1.

Edelhoff, H., Signer, J. and Balkenhol, N. [2016], 'Path segmentation for beginners: an overview of current methods for detecting changes in animal movement patterns', *Movement ecology* **4**(1), 21.

Egan, R. J., Bergner, C. L., Hart, P. C., Cachat, J. M., Canavello, P. R., Elegante, M. F., Elkhayat, S. I., Bartels, B. K., Tien, A. K., Tien, D. H. et al. [2009], 'Understanding behavioral and physiological phenotypes of stress and anxiety in zebrafish', *Behavioural brain research* **205**(1), 38–44.

Ennaceur, A. [2014], 'Tests of unconditioned anxiety—pitfalls and disappointments', *Physiology & behavior* **135**, 55–71.

Feldman, D. and Schulman, L. J. [2012], Data reduction for weighted and outlier-resistant clustering, *in* 'Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms', Society for Industrial and Applied Mathematics, pp. 1343–1354.

Flaška, V., Ježek, J., Kepka, T. and Kortelainen, J. [2007], 'Transitive closures of binary relations. i.', *Acta Universitatis Carolinae. Mathematica et Physica* **48**(1), 55–69.

Franco, A. [2016], 'Curvature', https://github.com/adamfranco/curvature/wiki.

Frank, E., Hall, M. A. and Witten, I. H. [2016], *The WEKA Workbench*, Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition.

Fränti, P. and Sieranoja, S. [2018], 'K-means properties on six clustering benchmark datasets', *Applied Intelligence* **48**(12), 4743–4759.

Fränti, P. and Sieranoja, S. [2019], 'How much can k-means be improved by using better initialization and repeats?', *Pattern Recognition* **93**, 95–112.

Fränti, P. and Virmajoki, O. [2006], 'Iterative shrinking method for clustering problems', *Pattern Recognition* **39**(5), 761–765.
URL: *http://dx.doi.org/10.1016/j.patcog.2005.09.012*

Gagniuc, P. A. [2017], *Markov chains: from theory to implementation and experimentation*, John Wiley & Sons.

Gallagher, M., Burwell, R. and Burchinal, M. R. [1993], 'Severity of spatial learning impairment in aging: development of a learning index for performance in the morris water maze.', *Behavioral neuroscience* **107**(4), 618.

Garthe, A., Behr, J. and Kempermann, G. [2009], 'Adult-generated hippocampal neurons allow the flexible use of spatially precise learning strategies', *PloS one* **4**(5), e5464.

Gärtner, B. and Schönherr, S. [1997], 'Smallest enclosing ellipses: fast and exact'.

Gärtner, B. and Schönherr, S. [1998], 'Exact primitives for smallest enclosing ellipses', *Information Processing Letters* **68**(1), 33–38.

Gehring, T. V. [2018], Automated classification of behavioural and electrophysiological data in Neuroscience, PhD thesis, University of Sheffield.

Gehring, T. V., Luksys, G., Sandi, C. and Vasilaki, E. [2015], 'Detailed classification of swimming paths in the morris water maze: multiple strategies within one trial', *Scientific reports* **5**, 14562.

Gehring, T. V., Wesierska, M. J., Wójcik, D. K. and Vasilaki, E. [2017], 'Analysis of behaviour in the active allothetic place avoidance task based on cluster analysis of the rat movement motifs', *bioRxiv* p. 157859.

Gerecke, U., Sharkey, N. E. and Sharkey, A. J. [2003], 'Common evidence vectors for self-organized ensemble localization', *Neurocomputing* **55**(3), 499–519.

Gerlai, R. [2017], 'Zebrafish and relational memory: Could a simple fish be useful for the analysis of biological mechanisms of complex vertebrate learning?', *Behavioural Processes* .

Gillner, S. and Mallot, H. A. [1998], 'Navigation and acquisition of spatial knowledge in a virtual maze', *Journal of cognitive neuroscience* **10**(4), 445–463.

Gonzalez, T. F. [1985], 'Clustering to minimize the maximum intercluster distance', *Theoretical Computer Science* **38**, 293–306.

Govender, P. and Sivakumar, V. [2020], 'Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)', *Atmospheric Pollution Research* **11**(1), 40–56.

Graeff, F. G., Netto, C. F. and Zangrossi Jr, H. [1998], 'The elevated t-maze as an experimental model of anxiety', *Neuroscience & Biobehavioral Reviews* **23**(2), 237–246.

Graziano, A., Petrosini, L. and Bartoletti, A. [2003], 'Automatic recognition of explorative strategies in the morris water maze', *Journal of neuroscience methods* **130**(1), 33–44.

Hahsler, M., Piekenbrock, M., Arya, S. and Mount, D. [2019], *dbscan: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms.*
URL: *https://github.com/mhahsler/dbscan*

Haldane, J. [1948], 'Note on the median of a multivariate distribution', *Biometrika* **35**(3-4), 414–417.

Hamilton, D. A., Rosenfelt, C. S. and Whishaw, I. Q. [2004], 'Sequential control of navigation by locale and taxon cues in the morris water task', *Behavioural brain research* **154**(2), 385–397.

Hammer, M. and Menzel, R. [1995], 'Learning and memory in the honeybee', *The Journal of Neuroscience* **15**(3), 1617–1630.

Han, S., Taralova, E., Dupre, C. and Yuste, R. [2018], 'Comprehensive machine learning analysis of hydra behavior reveals a stable basal behavioral repertoire', *elife* **7**, e32605.

Hardy, A. [1994], An examination of procedures for determining the number of clusters in a data set, *in* 'New approaches in classification and data analysis', Springer, pp. 178–185.

Hartigan, J. A. [1975], 'Clustering algorithms'.

Hartigan, J. A. and Wong, M. A. [1979], 'Algorithm as 136: A k-means clustering algorithm', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1), 100–108.

HascoëT, M. and Bourin, M. [1998], 'A new approach to the light/dark test procedure in mice', *Pharmacology Biochemistry and Behavior* **60**(3), 645–653.

Hascoët, M., Bourin, M. and Dhonnchadha, B. Á. N. [2001], 'The mouse ligth-dark paradigm: a review', *Progress in Neuro-Psychopharmacology and Biological Psychiatry* **25**(1), 141–166.

Higaki, A., Mogi, M., Iwanami, J., Min, L.-J., Bai, H.-Y., Shan, B.-S., Kan-no, H., Ikeda, S., Higaki, J. and Horiuchi, M. [2018], 'Recognition of early stage thigmotaxis in morris water maze test with convolutional neural network', *PloS one* **13**(5).

Higaki, A., Mogi, M., Iwanami, J., Min, L.-J., Bai, H.-Y., Shan, B.-S., Kukida, M., Kan-no, H., Ikeda, S., Higaki, J. et al. [2018], 'Predicting outcome of morris water maze test in vascular dementia mouse model with deep learning', *PloS one* **13**(2).

Hoerl, A. E. and Kennard, R. W. [1970], 'Ridge regression: Biased estimation for nonorthogonal problems', *Technometrics* **12**(1), 55–67.

Hollander, M. and Wolfe, D. A. [1999], 'Nonparametric statistical methods', *Wiley Series in Probability and Statistics* .

Honda, K., Notsu, A. and Ichihashi, H. [2010], 'Fuzzy pca-guided robust k-means clustering', *IEEE Transactions on Fuzzy Systems* **18**(1), 67–79.

Hong, W., Kennedy, A., Burgos-Artizzu, X. P., Zelikowsky, M., Navonne, S. G., Perona, P. and Anderson, D. J. [2015], 'Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning', *Proceedings of the National Academy of Sciences* **112**(38), E5351–E5360.

Horton, P., Nicol, A., Kendrick, K. and Feng, J. [2007], 'Spike sorting based upon machine learning algorithms (soma)', *Journal of neuroscience methods* **160**(1), 52–68.

Huzard, D., Vouros, A., Monari, S., Astori, S., Vasilaki, E. and Sandi, C. [2019], 'Constitutive differences in glucocorticoid responsiveness are related to divergent spatial information processing abilities', *Stress* pp. 1–13.

Illouz, T., Madar, R., Clague, C., Griffioen, K. J., Louzoun, Y. and Okun, E. [2016], 'Unbiased classification of spatial strategies in the barnes maze', *Bioinformatics* **32**(21), 3314–3320.

Illouz, T., Madar, R., Louzon, Y., Griffioen, K. J. and Okun, E. [2016], 'Unraveling cognitive traits using the morris water maze unbiased strategy classification (must-c) algorithm', *Brain, behavior, and immunity* **52**, 132–144.

Ingram, D. K., Spangler, E. L., Iijima, S., Ikari, H., Kuo, H., Greig, N. H. and London, E. D. [1994], 'Rodent models of memory dysfunction in alzheimer's disease and normal aging: moving beyond the cholinergic hypothesis', *Life sciences* **55**(25-26), 2037–2049.

Jain, A. K. [2010], 'Data clustering: 50 years beyond k-means', *Pattern recognition letters* **31**(8), 651–666.

Jain, V., Seung, H. S. and Turaga, S. C. [2010], 'Machines that learn to segment images: a crucial technology for connectomics', *Current opinion in neurobiology* **20**(5), 653–666.

Jancey, R. [1966], 'Multidimensional group analysis', *Australian Journal of Botany* **14**(1), 127–130.

Johnson, S. C. [1967], 'Hierarchical clustering schemes', *Psychometrika* **32**(3), 241–254.

Jolla, L. [2016], 'Graphpad prism 7.01'.
URL: *www.graphpad.com*

Juraska, J. M., Henderson, C. and Müller, J. [1984], 'Differential rearing experience, gender, and radial maze performance', *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology* **17**(3), 209–215.

Jurek, A., Bi, Y., Wu, S. and Nugent, C. [2011], Classification by cluster analysis: A new meta-learning based approach, *in* 'International Workshop on Multiple Classifier Systems', Springer, pp. 259–268.

Kärkkäinen, I. and Fränti, P. [2002], Dynamic local search algorithm for the clustering problem, Technical Report A-2002-6, Department of Computer Science, University of Joensuu, Joensuu, Finland.

Katsavounidis, I., Kuo, C.-C. J. and Zhang, Z. [1994], 'A new initialization technique for generalized lloyd iteration', *IEEE Signal processing letters* **1**(10), 144–146.

Kaufman, L. and Rousseeuw, P. J. [2009], *Finding groups in data: an introduction to cluster analysis*, Vol. 344, John Wiley & Sons.

Kearns, M. J. and Valiant, L. G. [1993], Cryptographic limitations on learning boolean formulae and finite automata, *in* 'Machine Learning: From Theory to Applications', Springer, pp. 29–49.

Klein, D., Kamvar, S. D. and Manning, C. D. [2002], From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering, Technical report, Stanford.

Kondo, Y., Salibian-Barrera, M., Zamar, R. et al. [2016], 'Rskc: an r package for a robust and sparse k-means clustering algorithm', *Journal of Statistical Software* **72**(5), 1–26.

Korf, R. E. [1985], 'Depth-first iterative-deepening: An optimal admissible tree search', *Artificial intelligence* **27**(1), 97–109.

Korthauer, L., Nowak, N., Frahmand, M. and Driscoll, I. [2017], 'Cognitive correlates of spatial navigation: Associations between executive functioning and the virtual morris water task', *Behavioural brain research* **317**, 470–478.

Kovács, F., Legány, C. and Babos, A. [2005], Cluster validity measurement techniques, *in* '6th International symposium of hungarian researchers on computational intelligence', Citeseer, p. 35.

Lai, D., Bai, A., Chang, K.-C., Wei, H. and Luo, L. [2012], 'Nonparametric analysis of the shenzhen stock market: The day of the week effect', *Mathematical and Computer Modelling* **55**(3), 1186–1192.

Lan, X., Li, Q. and Zheng, Y. [2015], Density k-means: A new algorithm for centers initialization for k-means, *in* 'Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on', IEEE, pp. 958–961.

Liaw, A. and Wiener, M. [2002], 'Classification and regression by randomforest', *R news* **2**(3), 18–22.

Lindner, M. D. [1997], 'Reliability, distribution, and validity of age-related cognitive deficits in the morris water maze', *Neurobiology of learning and memory* **68**(3), 203–220.

Lindner, M. D. and Gribkoff, V. K. [1991], 'Relationship between performance in the morris water task, visual acuity, and thermoregulatory function in aged f-344 rats', *Behavioural brain research* **45**(1), 45–55.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., Van Der Laak, J. A., Van Ginneken, B. and Sánchez, C. I. [2017], 'A survey on deep learning in medical image analysis', *Medical image analysis* **42**, 60–88.

Liu, X. and Sitaraman, D. [2019], 'A standardized tank design for the light dark task in zebrafish', *Pharmacology Biochemistry and Behavior* .

Lopuhaa, H. P., Rousseeuw, P. J. et al. [1991], 'Breakdown points of affine equivariant estimators of multivariate location and covariance matrices', *The Annals of Statistics* **19**(1), 229–248.

Luksys, G., Gerstner, W. and Sandi, C. [2009], 'Stress, genotype and norepinephrine in the prediction of mouse behavior using reinforcement learning', *Nature neuroscience* **12**(9), 1180–1186.

Luksys, G. and Sandi, C. [2011], 'Neural mechanisms and computations underlying stress effects on learning and memory', *Current opinion in neurobiology* **21**(3), 502–508.

Luo, W., Li, J., Yang, J., Xu, W. and Zhang, J. [2017], 'Convolutional sparse autoencoders for image classification', *IEEE transactions on neural networks and learning systems* **29**(7), 3289–3294.

MacQueen, J. et al. [1967], Some methods for classification and analysis of multivariate observations, *in* 'Proceedings of the fifth Berkeley symposium on mathematical statistics and probability', Vol. 1, Oakland, CA, USA, pp. 281–297.

Maei, H. R., Zaslavsky, K., Teixeira, C. M. and Frankland, P. W. [2009], 'What is the most sensitive measure of water maze probe test performance?', *Frontiers in integrative neuroscience* **3**, 4.

Magno, L. D. P., Fontes, A., Gonçalves, B. M. N. and Gouveia Jr, A. [2015], 'Pharmacological study of the light/dark preference test in zebrafish (danio rerio): waterborne administration', *Pharmacology Biochemistry and Behavior* **135**, 169–176.

Márquez, C., Poirier, G. L., Cordero, M. I., Larsen, M. H., Groner, A., Marquis, J., Magistretti, P. J., Trono, D. and Sandi, C. [2013], 'Peripuberty stress leads to abnormal aggression, altered amygdala and orbitofrontal reactivity and increased prefrontal maoa gene expression', *Translational psychiatry* **3**(1), e216.

MATLAB [2016*a*], 'Matlab statistics toolbox'.

MATLAB [2016*b*], *version 9.0 (R2016a)*, The MathWorks Inc., Natick, Massachusetts.

MATLAB [2019], *version 9.6.0 (R2019a)*, The MathWorks Inc., Natick, Massachusetts.

Maugis, C., Celeux, G. and Martin-Magniette, M.-L. [2009], 'Variable selection for clustering with gaussian mixture models', *Biometrics* **65**(3), 701–709.

Maximino, C., de Oliveira, D. L., Rosemberg, D. B., Batista, E. d. J. O., Herculano, A. M., Oliveira, K. R. M., Benzecry, R. and Blaser, R. [2012], 'A comparison of the light/dark and novel tank tests in zebrafish', *Behaviour* **149**(10-12), 1099–1123.

Menzel, R. and Erber, J. [1978], 'Learning and memory in bees', *Scientific American* **239**(1), 102–111.

Min, S., Lee, B. and Yoon, S. [2017], 'Deep learning in bioinformatics', *Briefings in bioinformatics* **18**(5), 851–869.

Modha, D. S. and Spangler, W. S. [2003], 'Feature weighting in k-means clustering', *Machine learning* **52**(3), 217–237.

Mohajer, M., Englmeier, K.-H. and Schmid, V. J. [2011], 'A comparison of gap statistic definitions with and without logarithm function', *arXiv preprint arXiv:1103.4767* .

Moret, B. M. and Shapiro, H. D. [1992], 'An empirical assessment of algorithms for constructing a minimum spanning tree.', *Computational Support for Discrete Mathematics* **15**, 99–117.

Morris, R. [1984], 'Developments of a water-maze procedure for studying spatial learning in the rat', *Journal of neuroscience methods* **11**(1), 47–60.

Morris, R. G. [1981], 'Spatial localization does not require the presence of local cues', *Learning and motivation* **12**(2), 239–260.

Morris, R. G. [2008], 'Morris water maze', *Scholarpedia* **3**(8), 6315.

Morris, R., Hagan, J. and Rawlins, J. [1986], 'Allocentric spatial learning by hippocampectomised rats: a further test of the "spatial mapping" and "working memory" theories of hippocampal function', *The Quarterly journal of experimental psychology* **38**(4), 365–395.

Ng, A. Y., Jordan, M. I. and Weiss, Y. [2002], On spectral clustering: Analysis and an algorithm, *in* 'Advances in neural information processing systems', pp. 849–856.

Nidheesh, N., Nazeer, K. A. and Ameer, P. [2017], 'An enhanced deterministic k-means clustering algorithm for cancer subtype prediction from gene expression data', *Computers in biology and medicine* **91**, 213–221.

Noldus, L. P., Spink, A. J. and Tegelenbosch, R. A. [2001], 'Ethovision: a versatile video tracking system for automation of behavioral experiments', *Behavior Research Methods, Instruments, & Computers* **33**(3), 398–414.

Numerical Algorithms Group (NAG) [2019], 'The nag toolbox for matlab ®'.
URL: *www.nag.com*

Olton, D. S. [1979], 'Mazes, maps, and memory.', *American psychologist* **34**(7), 583.

Oza, N. C. and Tumer, K. [2008], 'Classifier ensembles: Select real-world applications', *Information Fusion* **9**(1), 4–20.

Pan, W. and Shen, X. [2007], 'Penalized model-based clustering with application to variable selection', *Journal of Machine Learning Research* **8**(May), 1145–1164.

Panakhova, E., Burešová, O. and Bure, J. [1984], 'The effect of hypothermia on the rat's spatial memory in the water tank task', *Behavioral and neural biology* **42**(2), 191–196.

Papadimitriou, S., Kitagawa, H., Gibbons, P. B. and Faloutsos, C. [2003], Loci: Fast outlier detection using the local correlation integral, *in* 'Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)', IEEE, pp. 315–326.

Papadimitriou, S., Sun, J. and Philip, S. Y. [2006], Local correlation tracking in time series, *in* 'Sixth International Conference on Data Mining (ICDM'06)', IEEE, pp. 456–465.

Park, S., Lee, J.-J., Yun, C.-B. and Inman, D. J. [2008], 'Electro-mechanical impedance-based wireless structural health monitoring using pca-data compression and k-means clustering algorithms', *Journal of Intelligent Material Systems and Structures* **19**(4), 509–520.

Paul, C.-M., Magda, G. and Abel, S. [2009], 'Spatial memory: Theoretical basis and comparative review on experimental methods in rodents', *Behavioural brain research* **203**(2), 151–164.

Peck, R. [2012], *Statistics : the exploration and analysis of data*, Brooks/Cole, Cengage Learning, Australia United States.

Pena, J. M., Lozano, J. A. and Larranaga, P. [1999], 'An empirical comparison of four initialization methods for the k-means algorithm', *Pattern recognition letters* **20**(10), 1027–1040.

Pfitzner, D., Leibbrandt, R. and Powers, D. [2009], 'Characterization and evaluation of similarity measures for pairs of clusterings', *Knowledge and Information Systems* **19**(3), 361.

Piber, D., Schultebraucks, K., Mueller, S. C., Deuter, C. E., Wingenfeld, K. and Otte, C. [2016], 'Mineralocorticoid receptor stimulation effects on spatial memory in healthy young adults: A study using the virtual morris water maze task', *Neurobiology of Learning and Memory* **136**, 139–146.

Pol-Bodetto, S., Jeltsch-David, H., Lecourtier, L., Rusnac, N., Mam-Lam-Fook, C., Cosquer, B., Geiger, K. and Cassel, J.-C. [2011], 'The double-h maze test, a novel, simple, water-escape memory task: acquisition, recall of recent and remote memory, and effects of systemic muscarinic or nmda receptor blockade during training', *Behavioural brain research* **218**(1), 138–151.

Prieto, A., Prieto, B., Ortigosa, E. M., Ros, E., Pelayo, F., Ortega, J. and Rojas, I. [2016], 'Neural networks: An overview of early research, current frameworks and new challenges', *Neurocomputing* **214**, 242–268.

R Core Team [2017], *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Raftery, A. E. and Dean, N. [2006], 'Variable selection for model-based clustering', *Journal of the American Statistical Association* **101**(473), 168–178.

Rauch, T., Welch, D. and Gallego, L. [1989], 'Hypothermia impairs performance in the morris water maze', *Physiology & behavior* **46**(2), 315–320.

Rendón, E., Abundez, I., Arizmendi, A. and Quiroz, E. M. [2011], 'Internal versus external cluster validation indexes', *International Journal of computers and communications* **5**(1), 27–34.

Richendrfer, H., Pelkowski, S., Colwill, R. and Creton, R. [2012], 'On the edge: pharmacological evidence for anxiety-related behavior in zebrafish larvae', *Behavioural brain research* **228**(1), 99–106.

Roberts, A. C., Bill, B. R. and Glanzman, D. L. [2013], 'Learning and memory in zebrafish larvae', *Frontiers in neural circuits* **7**, 126.

Rodgers, R. and Shepherd, J. [1993], 'Influence of prior maze experience on behaviour and response to diazepam in the elevated plus-maze and light/dark tests of anxiety in mice', *Psychopharmacology* **113**(2), 237–242.

Rodriguez, A. and Laio, A. [2014], 'Clustering by fast search and find of density peaks', *Science* **344**(6191), 1492–1496.

Rogers, J., Churilov, L., Hannan, A. J. and Renoir, T. [2017], 'Search strategy selection in the morris water maze indicates allocentric map formation during learning that underpins spatial memory formation', *Neurobiology of learning and memory* **139**, 37–49.

Rousseeuw, P. J. [1987], 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of computational and applied mathematics* **20**, 53–65.

Ruta, D. and Gabrys, B. [2002], 'A theoretical analysis of the limits of majority voting errors for multiple classifier systems', *Pattern Analysis and Applications* **5**(4), 333–350.

Sahgal, A. [1993], *Behavioural neuroscience : a practical approach*, IRL Press, Oxford New York.

Salehi, B., Cordero, M. I. and Sandi, C. [2010], 'Learning under stress: the inverted-u-shape function revisited', *Learning & memory* **17**(10), 522–530.

Schapire, R. E. [1990], 'The strength of weak learnability', *Machine learning* **5**(2), 197–227.

Schnörr, S., Steenbergen, P., Richardson, M. and Champagne, D. [2012], 'Measuring thigmotaxis in larval zebrafish', *Behavioural brain research* **228**(2), 367–374.

Schütze, H., Manning, C. D. and Raghavan, P. [2008], Introduction to information retrieval, *in* 'Proceedings of the international communication of association for computing machinery conference', p. 260.

Schwabe, L., Schächinger, H., de Kloet, E. R. and Oitzl, M. S. [2010], 'Stress impairs spatial but not early stimulus–response learning', *Behavioural brain research* **213**(1), 50–55.

Schwabe, L. and Wolf, O. T. [2010], 'Learning under stress impairs memory formation', *Neurobiology of learning and memory* **93**(2), 183–188.

Scott, P. D. and Wilkins, E. [1999], 'Evaluating data mining procedures: techniques for generating artificial data sets', *Information and software technology* **41**(9), 579–587.

Seo, J. and Shneiderman, B. [2002], 'Interactively exploring hierarchical clustering results [gene identification]', *Computer* **35**(7), 80–86.

Serchov, T., van Calker, D. and Biber, K. [2016], 'Light/dark transition test to assess anxiety-like behavior in mice', *Bio-protocol* **6**(19), e1957.

Sharkey, A. and Sharkey, N. [1997], 'Diversity, selection, and ensembles of artificial neural nets', *Neural Networks and their Applications (NEURAP'97)* pp. 205–212.

Shawe-Taylor, J., Cristianini, N. et al. [2004], *Kernel methods for pattern analysis*, Cambridge university press.

Shlens, J. [2014], 'A tutorial on principal component analysis', *arXiv preprint arXiv:1404.1100* .

Šidák, Z. [1967], 'Rectangular confidence regions for the means of multivariate normal distributions', *Journal of the American Statistical Association* **62**(318), 626–633.

Siegel, S. [1956], 'Nonparametric statistics for the behavioral sciences', *Nonparametric Statistics for the Behavioral Sciences* .

Slonim, N., Aharoni, E. and Crammer, K. [2013], Hartigan's k-means versus lloyd's k-means-is it time for a change?, *in* 'IJCAI', pp. 1677–1684.

Soler, J., Tencé, F., Gaubert, L. and Buche, C. [2013], Data clustering and similarity, *in* 'The Twenty-Sixth International FLAIRS Conference'.

Stackman, R. W., Lora, J. C. and Williams, S. B. [2012], 'Directional responding of c57bl/6j mice in the morris water maze is influenced by visual and vestibular cues and is dependent on the anterior thalamic nuclei', *Journal of Neuroscience* **32**(30), 10211–10225.

Starczewski, A. and Krzyżak, A. [2015], Performance evaluation of the silhouette index, *in* 'International Conference on Artificial Intelligence and Soft Computing', Springer, pp. 49–58.

Steenbergen, P. J., Richardson, M. K. and Champagne, D. L. [2011], 'Patterns of avoidance behaviours in the light/dark preference test in young juvenile zebrafish: a pharmacological study', *Behavioural brain research* **222**(1), 15–25.

Stuchlik, A., Rezacova, L., Vales, K., Bubenikova, V. and Kubik, S. [2004], 'Application of a novel active allothetic place avoidance task (aapa) in testing a pharmacological model of psychosis in rats: comparison with the morris water maze', *Neuroscience letters* **366**(2), 162–166.

Szigeti, B., Deogade, A. and Webb, B. [2015], 'Searching for motifs in the behaviour of larval drosophila melanogaster and caenorhabditis elegans reveals continuity between behavioural states', *Journal of the Royal Society Interface* **12**(113), 20150899.

Takao, K. and Miyakawa, T. [2006], 'Light/dark transition test for mice', *JoVE (Journal of Visualized Experiments)* (1), e104.

Theodorsson-Norheim, E. [1987], 'Friedman and quade tests: Basic computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples', *Computers in biology and medicine* **17**(2), 85–99.

Thiebault, A. and Tremblay, Y. [2013], 'Splitting animal trajectories into fine-scale behaviorally consistent movement units: breaking points relate to external stimuli in a foraging seabird', *Behavioral Ecology and Sociobiology* **67**(6), 1013–1026.

Tibshirani, R. [1996], 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 267–288.

Tibshirani, R., Walther, G. and Hastie, T. [2001], 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(2), 411–423.

Todd, M. J. and Yıldırım, E. A. [2007], 'On khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids', *Discrete Applied Mathematics* **155**(13), 1731–1744.

Tonkiss, J., Shultz, P. and Galler, J. R. [1994], 'An analysis of spatial navigation in prenatally protein malnourished rats', *Physiology & behavior* **55**(2), 217–224.

Tucker, L. B., Velosky, A. G. and McCabe, J. T. [2018], 'Applications of the morris water maze in translational traumatic brain injury research', *Neuroscience & Biobehavioral Reviews* **88**, 187–200.

Usbeck, R., Röder, M., Ngonga Ngomo, A.-C., Baron, C., Both, A., Brümmer, M., Ceccarelli, D., Cornolti, M., Cherix, D., Eickmann, B. et al. [2015], Gerbil: general entity annotator benchmarking framework, *in* 'Proceedings of the 24th international conference on World Wide Web', International World Wide Web Conferences Steering Committee, pp. 1133–1143.

Van Gerven, D. J., Ferguson, T. and Skelton, R. W. [2016], 'Acute stress switches spatial navigation strategy from egocentric to allocentric in a virtual morris water maze', *Neurobiology of Learning and Memory* **132**, 29–39.

Varma, S. and Simon, R. [2006], 'Bias in error estimation when using cross-validation for model selection', *BMC bioinformatics* **7**(1), 91.

Vorhees, C. V. and Williams, M. T. [2006], 'Morris water maze: procedures for assessing spatial and related forms of learning and memory', *Nature protocols* **1**(2), 848–858.

Vorhees, C. V. and Williams, M. T. [2014], 'Assessing spatial learning and memory in rodents', *ILAR Journal* **55**(2), 310–332.

Vouros, A., Gehring, T. V., Croucher, M. and Vasilaki, E. [2017], *RodentDataAnalytics/mwm-ml-gen: Version 4.0.2-beta*, Zenodo. https://github.com/RodentDataAnalytics/mwm-ml-gen/releases/tag/v4.0.2.

Vouros, A., Gehring, T. V., Szydlowska, K., Janusz, A., Tu, Z., Croucher, M., Lukasiuk, K., Konopka, W., Sandi, C. and Vasilaki, E. [2018], 'A generalised framework for detailed classification of swimming paths inside the morris water maze', *Scientific reports* **8**(1), 15089.

Vouros, A., Langdell, S., Croucher, M. and Vasilaki, E. [2019], 'An empirical comparison between stochastic and deterministic centroid initialisation for k-means variations', *arXiv preprint arXiv:1908.09946* .

Vouros, A. and Vasilaki, E. [2020], 'A semi-supervised sparse k-means algorithm', *arXiv preprint arXiv:2003.06973* .

Vu, M.-A. T., Adalı, T., Ba, D., Buzsáki, G., Carlson, D., Heller, K., Liston, C., Rudin, C., Sohal, V. S., Widge, A. S. et al. [2018], 'A shared vision for machine learning in neuroscience', *Journal of Neuroscience* **38**(7), 1601–1607.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S. et al. [2001], Constrained k-means clustering with background knowledge, *in* 'Icml', Vol. 1, pp. 577–584.

Waller, D., Knapp, D. and Hunt, E. [2001], 'Spatial representations of virtual mazes: The role of visual fidelity and individual differences', *Human Factors* **43**(1), 147–158.

Wallis, S. [2013], 'Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods', *Journal of Quantitative Linguistics* **20**(3), 178–208.

Wang, S. and Zhu, J. [2008], 'Variable selection for model-based high-dimensional clustering and its application to microarray data', *Biometrics* **64**(2), 440–448.

Wang, Y., Miller, D. and Clarke, R. [2008], 'Approaches to working in high-dimensional data spaces: gene expression microarrays', *British journal of cancer* **98**(6), 1023.

Wei, H., Li, L., Song, Q., Ai, H., Chu, J. and Li, W. [2005], 'Behavioural study of the d-galactose induced aging model in c57bl/6j mice', *Behavioural brain research* **157**(2), 245–251.

Weisstein, E. W. [n.d.], 'Circumcircle', http://mathworld.wolfram.com/Circumcircle.html.

Weitzner, D. S., Engler-Chiurazzi, E. B., Kotilinek, L. A., Ashe, K. H. and Reed, M. N. [2015], 'Morris water maze test: Optimization for mouse strain and testing environment', *Journal of visualized experiments: JoVE* (100).

Whelan, C., Harrell, G. and Wang, J. [2015], Understanding the k-medians problem, *in* 'Proceedings of the International Conference on Scientific Computing (CSC)', The Steering Committee of The World Congress in Computer Science, Computer ..., p. 219.

Whishaw, I. Q. and Mittleman, G. [1986], 'Visits to starts, routes, and places by rats (rattus norvegicus) in swimming pool navigation tasks.', *Journal of Comparative Psychology* **100**(4), 422.

Witten, D. M. and Tibshirani, R. [2010], 'A framework for feature selection in clustering', *Journal of the American Statistical Association* **105**(490), 713–726.

Witten, D. M., Tibshirani, R. and Hastie, T. [2009], 'A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis', *Biostatistics* **10**(3), 515–534.

Wolfer, D. P. and Lipp, H.-P. [1992], 'A new computer program for detailed off-line analysis of swimming navigation in the morris water maze', *Journal of neuroscience methods* **41**(1), 65–74.

Wolfer, D. P. and Lipp, H.-P. [2000], 'Dissecting the behaviour of transgenic mice: is it the mutation, the genetic background, or the environment?', *Experimental physiology* **85**(06), 627–634.

Wolfer, D. P., Madani, R., Valenti, P. and Lipp, H.-P. [2001], 'Extended analysis of path data from mutant mice using the public domain software wintrack', *Physiology & Behavior* **73**(5), 745–753.

Wolfer, D. P., Stagljar-Bozicevic, M., Errington, M. L. and Lipp, H.-P. [1998], 'Spatial memory and learning in transgenic mice: fact or artifact?', *Physiology* **13**(3), 118–123.

Xie, B., Pan, W. and Shen, X. [2008], 'Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables', *Electronic journal of statistics* **2**, 168.

Xie, Q., Shang, S., Yuan, B., Pang, C. and Zhang, X. [2013], Local correlation detection with linearity enhancement in streaming data, *in* 'Proceedings of the 22nd ACM international conference on Information & Knowledge Management', pp. 309–318.

Xing, E. P., Jordan, M. I., Russell, S. J. and Ng, A. Y. [2003], Distance metric learning with application to clustering with side-information, *in* 'Advances in neural information processing systems', pp. 521–528.

Yan, M. and Ye, K. [2007], 'Determining the number of clusters using the weighted gap statistic', *Biometrics* **63**(4), 1031–1037.

Yerkes, R. M., Dodson, J. D. et al. [1908], 'The relation of strength of stimulus to rapidity of habit-formation', *Punishment: Issues and experiments* pp. 27–41.

Yeshurun, S., Rogers, J., Short, A. K., Renoir, T., Pang, T. Y. and Hannan, A. J. [2017], 'Elevated paternal glucocorticoid exposure modifies memory retention in female offspring', *Psychoneuroendocrinology* .

Zhou, Z.-H., Wu, J. and Tang, W. [2002], 'Ensembling neural networks: many could be better than all', *Artificial intelligence* **137**(1-2), 239–263.

Zhu, F., Liu, Q., Fu, Y. and Shen, B. [2014], 'Segmentation of neuronal structures using sarsa (λ)-based boundary amendment with reinforced gradient-descent curve shape fitting', *PloS one* **9**(3), e90873.

Zhu, M. [2015], 'Use of majority votes in statistical learning', *Wiley Interdisciplinary Reviews: Computational Statistics* **7**(6), 357–371.

Zhu, X. and Goldberg, A. B. [2009], 'Introduction to semi-supervised learning', *Synthesis lectures on artificial intelligence and machine learning* **3**(1), 1–130.