Gonçalo Sancho de Queiroz de Moncada Sousa Mendes

BSc in Computer Science

Mining Extremes through Fuzzy Clustering

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Engineering

Adviser: Susana Nascimento, Assistant Professor,
NOVA University of Lisbon

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2018

Mining Extremes through Fuzzy Clustering

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

I would like to start by expressing my deepest thanks to my adviser, Professor Susana Nascimento for all her support, dedication and patience. Characteristics that pushed me forward and allowed me to grow as a person and scientist.

A special thanks to the PhD student Sérgio Casca for his support, patient and kindness demonstrated during the course of this work, while sharing his knowledge with me. To my college, João Almeida, for his sharp and accurate reviews. And to my good friend, Tiago Costa, for the many years hours he dedicated in helping me, that really helped in improving the quality of this dissertation.

Finally, many thanks to my family and friends, for their support and understanding showed towards me throughout this work, and the years that led to it.

Abstract

Archetypes are extreme points that synthesize data representing "pure" individual types. Archetypes are assigned by the most discriminating features of data points, and are almost always useful in applications when one is interested in extremes and not on commonalities. Recent applications include talent analysis in sports and science, fraud detection, profiling of users and products in recommendation systems, climate extremes, as well as other machine learning applications.

The furthest-sum Archetypal Analysis (FS-AA) (Mørup and Hansen, 2012) and the Fuzzy Clustering with Proportional Membership (FCPM) (Nascimento, 2005) propose distinct models to find clusters with extreme prototypes. Even though the FCPM model does not impose its prototypes to lie in the convex hull of data, it belongs to the framework of data recovery from clustering (Mirkin, 2005), a powerful property for unsupervised cluster analysis. The baseline version of FCPM, FCPM-0, provides central prototypes whereas its smooth version, FCPM-2 provides extreme prototypes as AA archetypes.

The comparative study between FS-AA and FCPM algorithms conducted in this dissertation covers the following aspects. First, the analysis of FS-AA on data recovery from clustering using a collection of 100 data sets of diverse dimensionalities, generated with a proper data generator (FCPM-DG) as well as 14 real world data. Second, testing the robustness of the clustering algorithms in the presence of outliers, with the peculiar behaviour of FCPM-0 on removing the proper number of prototypes from data. Third, a collection of five popular fuzzy validation indices are explored on accessing the quality of clustering results. Forth, the algorithms undergo a study to evaluate how different initializations affect their convergence as well as the quality of the clustering partitions. The Iterative Anomalous Pattern (IAP) algorithm allows to improve the convergence of FCPM algorithm as well as to fine-tune the level of resolution to look at clustering results, which is an advantage from FS-AA. Proper visualization functionalities for FS-AA and FCPM support the easy interpretation of the clustering results.

Keywords: Archetypal analysis; Fuzzy proportional membership; Clustering data recovery; Fuzzy data generator; Fuzzy validation indices.

Resumo

Arquétipos são pontos extremos que sintetizam dados que representam tipos individuais "puros". Arquétipos são constituídos pelas características mais descriminantes dos atributos dos pontos, e são quase sempre uteis em aplicações onde o interesse está em extremos e não em características gerais. Aplicações recentes onde este conceito tem sido aplicado incluem analise de talento em desporto e ciência, detecção de fraude, descrição de perfis de consumidores e produtos em sistemas de recomendação, eventos climáticos extremos, entre outras aplicações de aprendizagem automática.

Tanto a soma mais distante da Análise de Arquétipos (FS-AA) (Mørup e Hansen, 2012) como o Agrupamento Difuso com Pertença Difusa por Proporção (FCPM) (Nascimento, 2005) propõem modelos distintos para encontrar partições com protótipos extremos. Apesar de o modelo do FCPM não impor aos seus protótipos que estejam na fronteira do dados, ele pertence à abordagem de recuperação de dados das partições encontradas (Mirkin, 2005), uma propriedade forte para analise não super-visionada de agrupamento difuso. O modelo base do FCPM, o FCPM-0, encontra protótipos centrais enquanto que a versão menos restringida, o FCPM-2, encontra protótipos extremos, como o AA.

O estudo comparativo entre os algoritmos FS-AA e FCPM realizados nesta dissertação cobre os seguintes aspetos. Primeiro, a análise do FS-AA em recuperar os dados das partições encontradas usando uma coleção de 100 conjuntos de dados de diversas dimensionalidades, gerados através de um gerador de dados próprios (FCPM-DG) e com 14 conjuntos de dados do mundo real. Segundo, testar a robustez dos algoritmos na presença de pontos atípicos, com o comportamento peculiar do FCPM-0 em remover o número correto de protótipos do espaço dos dados. Terceiro, uma coleção de cinco índices de validação difusa populares são explorados para avaliar a qualidade das partições encontradas. Quarto, os algoritmos são sujeitos a um estudo para avaliar como diferentes inicializações afetam a sua convergência assim como a qualidade das partições encontradas. O algoritmo Padrões Anômalos Iterativos não só permite melhorar a convergência do algoritmo do FCPM, como também afinar o nível de resolução para observar as partições encontradas, o que é uma vantagem do FS-AA. Funcionalidades de visualização próprias para o FS-AA e o FCPM suportam a fácil interpretação dos resultados.

Palavras-chave: Análise de Arquétipos; Agrupamento difuso com pertença difusa por proporção; Recuperação de dado das partições encontradas; Geradores de dados difusos; Índices de validação difusa.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Throughout the history of Mankind, scientists have always tried to classify what surrounds us with well-defined proprieties and clear boundaries. Here, an object, emotion, propriety, either belonged to one class, or to another. From this classification, everything was sorted in well-defined classes, and specific names were coined to address each one of them.

However, the world that surround us is not as clear shaped as we would like it to be, and the presence of randomness, and lack of clear classifications affects the distributions in such organized groups.

From such uncertainty, the theory of Fuzzy sets was created by Zadeh, 1965 to accommodate these notions and deal with such imprecise and blurry frontiers. Now, problems where the difficulty in classification is present, might not be credited to random variables, but to the intrinsic nature of the problem, where a sharply defined criteria is absence (Zadeh, 1965). These Fuzzy sets could be used in cluster analysis, or pattern-recognition (Bellman et al., 1966).

The definition of c-partition space (Ruspini, 1969) followed, creating the foundations for what would later be the first fuzzy clustering algorithm. This led to the creation of the fuzzy ISODATA, by Dunn, 1973, that was later generalized by Bezdek, 1981, bringing forth the first Fuzzy c-means (FCM). This algorithm assigns memberships values to each individual in a data set, allowing it to be related to several groups. These groups are represented by a single point, located in center of the group.

However, in some cases, it's more interesting to find "pure types"instead of those central representations. Types that can be seen as the origin of information, the individuals

from which all other ones withdraw their characteristics, Archetypes. The Merriam-Webster dictionary (Archetype, 2018) defines them as "the original pattern or model of which all things of the same type are representations or copies". Such concept is rather useful in many applications, as an "ideal type" can be interpreted as a model, or an extreme of some environment.

Good examples of the usefulness of such types are medical environments, where a data set contains several patients and the description of their illness symptoms. When the archetypes of such data sets are found, it's possible to describe the "true form" of the illnesses present in such group. Here, diseases can be perfectly profiled, without being "contaminated" with symptoms of other illness, as it usually happens when using "central types".

The Mental Disorders data set is one of the best examples of a medical data set, to understand the usefulness of archetypes (Nascimento, 2005). It contains 44 patients, with 17 psychosomatic features describing 4 psychiatric disorders: depressed (D), maniac (M), simple schizophrenic (S_s) and paranoid schizophrenic (S_p). Figure 1.1 contains four archetypes found for this data set. The archetypes can then been seen as the "true form" of the four psychiatric disorders, given the extreme features that each one contain, Figure 1.1a.

a Percentile plot.

b Mixture plot.

Figure 1.1: Demonstration of the usefulness of the archetypes with a data set of mental disorders. The archetypes were found through the implementation provided in Mørup and Hansen, 2012. On left, the percentile plot, showing the extreme values of the features with a red line, for each archetype. On right, the mixture plot, with the data points plotted according to their memberships, *i.e.*, their distance to the archetypes. It's also possible to observe how the archetypes successfully identify the four mental disorders.

The concept of archetypes also allows for benchmarking, as, in this scenario, "ideal types" are the individuals (real or not) from which all the others must be compared to. Other applications such as sports, fraud detection, products recommendation systems, and so on, have found an extraordinary usefulness in such concepts.

Despite the usefulness of such cluster analysis methods, they all have been dominated

by learning from data rather than theoretical based instructions. Indeed, the clusters to be retrieve from data not only depend on the data by itself, but also on the user's goals and on the degree of granularity one wants to analyse the grouping of data. To deeply understand the clustering structure present in data Mirkin, 2005 proposed a data-recovery paradigm where the retrieved clusters must be treated as "ideals" representation of the data. These representations could then be used for recovering the original data back from its "ideal" format. Therefore, not only use the data for finding clusters, but also use the clusters for recovering the original data.

This principle has been incorporated in fuzzy clustering by Nascimento, 2002 with the model of Fuzzy Clustering with Proportional Membership (FCPM) for mining typological structures from data. The fuzzy proportional membership extends the classical fuzzy memberships since it's involved in the reconstruction of the observations from the clusters. The FCPM provides a family of clustering criteria, FCPM-m, with fuzziness parameter ($m = 0, 1, 2$), leading to cluster structures with central prototypes (FCPM-0, FCPM-1), closely matching the FCM, as well as cluster structures with extreme prototypes (FCPM-2), close to the concept of archetypal types, found by the Archetypal Analysis (Cutler and Breiman, 1994).

The greatest experimental contribution of Nascimento, 2002, was the creation of an artificial data generator, based on the model of the FCPM, alongside a Matlab (MATLAB, 2015) based platform. This platform allows for experimentation, with the FCPM model, using the data generator, a visualization tool and post-processing, with well-known clustering indices.

Mørup and Hansen, 2012 proposed an effective AA algorithm, with a variant of the projected gradient for the alternating optimization (AO) algorithm, which guarantees a faster convergence. As in this implementation of the AA, the FCPM also uses a variant of the project gradient method for the AO. As such, both algorithms need careful initializations. For the location of the seeds, Mørup and Hansen, 2012 proposed the Furthest-Sum (FS) method to initialize the AA algorithm. The method finds c pre-defined data points, that are furthest way from the centre of the data, to be used as seeds. This method resembles the Iterative Anomalous Pattern (IAP), that showed good results when applied to the problem of unsupervised segmentation of Sea Surface Temperature (SST) images with the Fuzzy c-Means (Nascimento and Franco, 2009). For the FCPM, the problem of initialization is address by running the algorithm several times, with pseudo-random seeds.

Despite the theoretical and emerging areas of application where archetypal analysis (AA) has shown success, it still lacks a systematic method to correctly validate the number of archetypes to be used, as it still relies on the simplistic elbow method. Also, no study has been conducted on the analysis of data recovery from its clusters, in particular, from the archetypes.

Both algorithms also lack a depth study regarding their behaviour in the presence of

outliers. This is extremely important, as the archetypes are located in the convex hull of the data, and the FCPM-2 has extreme prototypes.

1.2 Objectives

The main goal of this dissertation is four-fold:

1. To systematically analyse the data recovery properties of the archetypal analysis algorithm;
2. To experimentally compare the FCPM with AA clustering using proper synthetic generated data, as well as real-world data in the framework of the data recovery paradigm;
3. To develop an experimental validation protocol for AA, using a fusion strategy of fuzzy validation indices, to overcome the simplistic existing analysis of AA validation, by the elbow method;
4. To study the influence of different initializations in the clustering solutions provided by the algorithms.

1.3 Main Contributions

The main goal of this dissertation is to experimentally compare one version of Archetypal Analysis, the Furthest-Sum Archetypal analysis (FS-AA) algorithm (Cutler and Breiman, 1994) with the Fuzzy Clustering with Proportional Membership (FCPM) (Nascimento, 2002). This way, the main contributions of the dissertation are:

1. To experimentally compare the FCPM with the Furthest-Sum Archetypal Analysis (FS-AA) algorithm in the framework of data recovery. This goal is achieved using synthetic data generated from different space dimensionalities with a proper data generator of the FCPM model, the FCPM-DG. Also, a collection of diverse real-world data had been applied;
2. To develop an experimental validation protocol for AA exploring five premier fuzzy validation indices, to overcome the simplistic existing AA validation scheme with the elbow method;
3. To analyse the robustness of the FS-AA and FCPM algorithms in the presence of outliers;
4. To study the influence of different initialization strategies on the FS-AA and FCPM algorithms respecting the quality of found partitions.

The first, second and third contributions lead to the creation of a paper, published and presented in 19th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2019 (Mendes and Nascimento, 2018)

1.4 Organization

This document is organized in 5 chapters, including this one.

Chapter 2 is dedicated to partitional soft clustering. It serves the purpose of introducing the algorithms that will be used throughout this work. The chapter starts by introducing the FCM, proposed by Bezdek, 1981, that only finds "central types", and walks towards an algorithm that only finds "pure types" (AA), introducing in the middle, one that finds both of them (FCPM). For each model, its method is introduced, followed by a reference to an implementation to solve its clustering criterion. Then, its main characteristics are shown. Finally, a short review on the areas of application where the model has found success.

At the end of the chapter, the three models are compared against each other, highlighting their main differences, and stressing the problems that all of them share.

In 3 chapter the focus is on the use of artificial data sets and on the generation of such sets. Here, its underline the benefits and importance of using artificial data sets. By presenting the problems that occur when a practitioner doesn't use artificial data, and the benefits when it does, it becomes clear why such data sets have an important role in unsupervised learning. Data generators with cluster tendency for the algorithms used in this work will then be introduced.

The 4 chapter starts by setting up the theoretical framework on the need of validation indices in clustering. It then follows to the paradigm of data recovery, and on how to assess it. Then, several validation indices are introduced, together with strategies on how to join them and use them as one. In the end, some visualization techniques to inspect fuzzy partitions and help in the evaluation of the results are introduced.

Chapter 5 presents the results of the experimental study and a discussion on the findings. First, on the capability of the AA on recovering archetypes, using multidimensional artificial data, generated with respect to the FCPM original model, and compared against the FCPM. Then, the data is augmented with outliers, and the sensitivity of the algorithms regarding the augmented data is tested. Third, the algorithms are methodically studied with unsupervised validation fuzzy clustering indices, to evaluate the quality of the found fuzzy partitions, regarding the number of clusters, with real data. In the end, a study comparing how different initializations affect the clustering results, exploring the Furthest Sum and the Iterative Anomalous Pattern (IAP). This comparison was extended to include the Iterative Furthest Prototype (IFP) algorithm, a modification to the IAP.

Chapter 6 presents the conclusions and future work of this thesis.

CHAPTER 2

Partitional Soft Clustering

In this chapter, it's explored the first Fuzzy *c*-Means and two other algorithms, with the same fuzzy framework, that explore the notion of "pure types", the Fuzzy c-Means Via Proportional Membership Model and the Archetypal Analysis. For each of the models that will be presented, it will be given a description, followed by the implementation. Then, its main characteristics and a review of the main applications.

In the end of the chapter, the 3 of them are compared against each other.

2.1 Fuzzy c-Means

The Fuzzy *c*-means (FCM) is introduced in this section and follows the definition of Bezdek, 1981.

2.1.1 Method

Given $X = x_1, x_2, ..., x_n$, a data set, with p attributes, it's possible to partition X into c clusters, with $c \in \{2, ..., n-1\}$, that represent a structure of X. The fuzzy partition space is organized in a $c \times n$ matrix $U = [u_{ik}]$, with u_{ik} $i = 1, ..., c$, $k = 1, ..., n$ denoting the fuzzy membership value of x_k to the c^{th} cluster. This matrix is called the fuzzy partition matrix and satisfies the following constraints:

$$0 \leq u_{ik} \leq 1, \ for\ all\ i = 1, ..., c, \ k = 1, ..., n, \tag{2.1}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \ for\ all\ k = 1, ..., n, \tag{2.2}$$

$$0 < \sum_{k=1}^{n} u_{ik} < n, \ for\ all\ i = 1, ..., c. \tag{2.3}$$

The first constraint (2.1) states that the membership values belongs to the interval $[0,1]$. The second constraint (2.2) implies that the total membership of each entity, x_k, is equal to one, i.e., their membership are exhaustive regarding the c clusters. Finally, constraint (2.3) states that no cluster is empty.

Formally, find $U = u_1, u_2, ..., u_n$, the fuzzy membership matrix, and $V = v_1, v_2, ..., v_c$, the cluster prototypes, that minimize the square-error objective function clustering criterion:

$$J_m(U, V, X) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m d^2(x_k, v_i), \tag{2.4}$$

where $m \in]1, \infty)$ is a parameter which determines the degree of fuzziness of the resulting clusters and $d^2(x_k, v_i)$ is the Euclidean norm. There are several norms (Höppner et al., 1999), but throughout this will only the Euclidean norm will be used, as experiments with others is not in the scope of this work. One of the most famous way to minimize the clustering criteria (2.4) is the Alternating Optimization Algorithm (AO) (Bezdek, 1981).

2.1.2 Algorithm

The problem of minimizing the clustering criteria (2.4) represents a non-linear optimization problem that can be solved using a wide range of methods. In this work, as mentioned before, the focus will be on the alternating optimization algorithm, as it is the most widely used optimization method and the simplest one. Bezdek, 1981 used this method and the implementation of Balasko et al., 2005 follows the same structure (Algorithm 1) and it's distributed in Matlab.

Algorithm 1 The Fuzzy c-means Algorithm

function FCM$(X, c, m, epsilon)$ % $c \in \{2, ..., n-1\}$, $m > 1$, $\epsilon > 0$
  Initialize U randomly.
  repeat for $l = 1, 2, ...$

    Step 1 Compute the cluster prototypes, v_i:
    $v_i^{(l)} = \frac{\sum_{k=1}^{n} (u_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^{n} (u_{ik}^{(l-1)})^m}$, $1 \le i \le c$.

    Step 2 Compute the distances:
    $\|x_k - v_i^{(l)}\|^2$, $1 \le i \le c$, $1 \le k \le n$.

    Step 3 Update the partition matrix:
    $u_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d^2(x_k, v_i)}{d^2(x_k, v_j)} \right)^{2/(m-1)}}$, $1 \le i \le c$, $1 \le k \le n$.

  until $\|U^{(l)} - U^{(l-1)}\| < \epsilon$
end function

2.1.3 Main characteristics

The choice of similarity (or dissimilarity) metric requires that the structure of data is taken into consideration. For instance, using the Euclidean distance as a measure of similarity will tend to produce circular clusters, which may not be in accordance with the data structure. This led to several modifications on the original model (Bezdek et al., 1999; Li and Lewis, 2016). The choice of m is also important, as different values of m leads to different results in the partitions. When $m \to \infty$, the partitions approach $\overline{U} = [1/c]$, that corresponds to entirely fuzzy ones. Contrariwise, when m approaches 1 the partitions become more and more crisp, reducing the algorithm to a hard c-means, when m reaches 1.

The number of clusters is a user dependent parameter and a crucial one as it deeply influences the clustering results.

Starting the algorithm by initializing U, or V, and how they are initialized is very important as it holds a significant impact in the convergence of the algorithm, storage and speed. Although the algorithm is guaranteed to converge to a local minimum (Bezdek, 1981), distinct initializations may lead to different locals.

The prototypes found by the FCM are, usually, "central types", located in the center of the cluster. Such propriety is visible on the first step of the FCM algorithm (1), where each v_i is the weighted mean of the points in each c_i.

Finally, the epsilon, ϵ, needs to be chosen, as it controls the termination of the algorithm and its quality for the final clusters.

2.1.4 Areas of Application

Due to its ease of use and interpretability, as it presents less strict results than hard clustering, the Fuzzy c-Means is very popular and widely spread amongst several industries. These proprieties make the Fuzzy c-Means very useful in the decision-making process of such industries. Some of them are, the businesses world (Tufan and Hamarat, 2003; Stetco et al., 2013; Bose and Chen, 2015; Schafer et al., 2015), the energy sector (Alia, 2014; Sert et al., 2015; Jahromi et al., 2016; Maity et al., 2016), chemistry (Liu et al., 2016), medicine and health care (Fenza et al., 2012; Huang et al., 2014; Ferreira et al., 2015; Karami et al., 2015; Ahmad, 2016), web classification (Ansari et al., 2015; Tsekouras and Gavalas, 2013; Cosma and Acampora, 2016), big data analysis (Găceanu and Pop, 2012; Li et al., 2015; Xianfeng and Pengfei, 2015), machine learning (Wang et al., 2012; Wu et al., 2014), pattern recognition and image classification (John et al., 2015; Majumdar et al., 2015; Khormali and Addeh, 2016), times-series prediction (Yolcu, 2013; Izakian et al., 2015; Peng et al., 2015), robust design (D'Urso et al., 2014), meteorological data (Sun et al., 2010; Li et al., 2011), just to name a few. Li and Lewis, 2016 provide an extensive overview on emerging domains of application of fuzzy clustering.

2.2 Fuzzy c-Means via Proportional Membership Model

Most approaches on fuzzy clustering, specially the Fuzzy c-Means method, previously described, find a membership degree for each entity to express its proximity to each prototype. This framework makes the cluster structure determined from the data but fails to provide a feedback on the generation of the data from the cluster structure.

To tackle this problem, Nascimento, 2005 proposed a framework for mining for typological structures. The definition of typology is stated as "Study of or analysis or classification based on types or categories", according to the Merriam-Webster dictionary (Typology, 2018). The motivation of this approach is to define the underling fuzzy c-partition in such a way that the membership of an entity to a cluster not only expresses the belongingness of the entity to the cluster, but also expresses the proportion of the clusters prototypes present in the entity. This means that, an entity x_i, with a membership of 0.60 to cluster A and 0.40 to cluster B, reflects 60% of the prototype A and 40% of the prototype B. This type of membership function has been coined Fuzzy Clustering with Proportional Membership (Nascimento et al., 2003).

2.2.1 Method

The FCPM model assumes that the data is generated according to the cluster structure:

$$observed\ data\ =\ model\ data\ +\ noise. \tag{2.5}$$

Here, it's assumed the existence of some prototypes which serve as "ideal" patterns to data entities. The meaning of "ideal" patterns is something that the researcher needs to define as an entity that would ideally typify the characteristics of a cluster.

Given data matrix Y, preprocessed from X by shifting the origin to the gravity center of all the entities, and rescaling features by their ranges:

$$y_{kh} = \frac{x_{kh} - a_h}{s_h}, \tag{2.6}$$

with $a_h = \bar{x}_h$ and $s_h = max_k(x_{kh}) - min_k(x_{kh})$, then, $Y = [y_{kh}]$ is a $n \times p$ entity-to-feature data table, with $k = 1,..,n; h = 1,...,p$. Based on the assumption (2.5), a generic proportional membership model was defined, where the membership value u_{ik} is not just a weight, but an expression of the proportion of v_i which is present in y_k, is assumed. This assumption translates to the following model that instantiates the generic model 2.5:

$$y_{kh} = u_{ik}v_{ih} + e_{ikh}, \tag{2.7}$$

where e_{ikh} are the residuals values and as small as possible. From this generic model, a generic Square-Error Criterion, for the clustering criterion was defined. This criterion is defined as fitting each data point to a share of each of the prototypes, represented by the degree of membership. By minimizing all the residual values in the generic model (2.7) via the squared-error, the goal is achieved:

$$E_0(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{h=1}^{p}(y_{kh} - u_{ik}v_{ih})^2, \tag{2.8}$$

with the fuzzy constraints

$$0 \leq u_{ik} \leq 1,\ for\ all\ i = 1,...,c,\ k = 1,...,n, \tag{2.9}$$

and

$$\sum_{i=1}^{c} u_{ik} = 1,\ for\ all\ k = 1,...,n. \tag{2.10}$$

However, as this criterion is too strong and unrealistic sometimes (Nascimento, 2005), an adaptation of the squared error (2.8) was made, creating a smooth version. Here, only meaningful proportions, those with high membership values, are to be taken into account in the assumption (2.5). To smooth this influence, a weight was put on the squared residuals in the squared error (2.8), with a power of m ($m = 0, 1, 2$) of the corresponding u_{ik}, creating the smooth squared error, the FCPM-m:

$$E_m(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{h=1}^{p} u_{ik}^m(y_{kh} - u_{ik}v_{ih})^2, \tag{2.11}$$

also subject to the constraints (2.9) and (2.10). Now, the influence of high residual values, e_{ikh}, are smoothed. In this new clustering criterion, the choice of m highly influences the position of the prototypes. Note that (2.8) is a special case of (2.11), for $m = 0$.

The Alternation Optimization (AO) is adopted to minimize the smooth squared error in (2.11). First, initialize V with pseudo-random values, generated in the data space, and update U from this V. Then, alternate between minimizing the membership matrix, U given the centroids, $\hat{V}$ and minimizing V, given the updated $\hat{U}$. Stop when the algorithm converges. The prototypes feature values are derived by the first order condition of minimizing the clustering criterion 2.11 as:

$$v_{ih}^{(t)} = \frac{\left\langle \left(u_i^{(t)}\right)^{m+1}, y_h \right\rangle}{\left\langle \left(u_i^{(t)}\right)^{m+1}, u_i^{(t)} \right\rangle}. \tag{2.12}$$

The process of finding the membership matrix U is not so simple. Due to the constraints (2.9) and (2.10), the minimization of the clustering criterion (2.11) with respect to U requires an iterative process on its own, as is not analytically derivable. This lead to the development of a new variant of the Gradient Projection Method (GPM). Now, two different iterations are need for each step of the minimization process, a major iteration and a minor one. Each major represents a step in the full process of minimizing the smooth squared error (2.11). Within each major iteration, there is a minor one, to calculate U. Updating U requires several steps, that comes from the gradient projection method. As the theoretical framework of this update is outside of the scope of this work,

the pseudo-code for the minor iteration will not be presented. For a detailed explanation of the foundations of the FCPM algorithm and its variations, please consult Nascimento, 2005.

2.2.2 Algorithm

Algorithm 2 presents the major iteration of the FCPM-m.

Algorithm 2 The FCPM-m Algorithm - The major iteration

function FCPM(Y, c,T_1,T_2, ϵ) % $\epsilon > 0$
  $V^{(0)} \leftarrow \{v_i^{(0)}\}_{i=1}^{c}$ % initialize V
  $U^{(0)}$ % initialize U from the V
  $t_1 \leftarrow 0$
  repeat
    $t_2 \leftarrow 0$
    $U^{(t_2)} \leftarrow U^{(t_1)}$
    repeat
      $t_2 \leftarrow t_2 + 1$
      for $k = 1,...,n$ **do**
        $d_k \leftarrow computeD(V^{(t_1)}, u_k^{(t_2-1)})$
        $u_k^{t_2} \leftarrow ComputeProjection(d_k)$ % Minor iteration
      end for
    until $(|U^{(t_2)} - U^{(t_2-1)}|_{err} < \epsilon \parallel t_2 = T_2)$
    $t_1 \leftarrow t_1 + 1$
    $U^{(t_1)} \leftarrow U^{(t_2)}$
    $V^{(t_1)} \leftarrow computeV(U^{t_1})$ % from (2.12)
  until $(|V^{(t_1)} - V^{(t_1-1)}|_{err} < \epsilon \parallel t_1 = T_1)$
  return $(V^{(t_1)}, U^{(t_1)})$
end function

2.2.3 Main characteristics

As in the FCM, the number of prototypes chosen requires some thought, especially if $m = 0$. A bad choice on this number may lead to the non-convergence of the FCPM algorithm, as it may shift some of the prototypes to infinity. In the conducted experimental study (Nascimento et al., 2003), when the FCPM algorithm did converge, the number of major iterations was quite small. Combining these two characteristics allowed to define another stopping criteria: if the number of major iterations, when $m = 0$, exceeds a large number, it means that the algorithm did not converge. The calculations in the original work suggests a number above 100, for the major iterations and 10000 for the minor iterations. These limits are adopted in this work. Proprieties such as the ϵ, or the initialization also need some careful thought.

The prototypes derived by the FCPM-0, or the non-smoothed model (2.8), are ideal types, since they have extreme subset of features. Each entity contains u_{ik} (a membership

value) of it, plus the residuals. In this view, both the prototypes and the memberships are reflected on the model of the data.

Observing the generic model (2.7), it is possible to see that it can be treated as a device to reconstruct the data from the model. Furthermore, the trivial structure where all the entities are prototypes it's not a solution, as it doesn't minimize the squared error (2.8) to its absolute minimum.

In the FCPM model, the data has to be shifted to the origin of the space gravity, that, according to the model, allows for a greater discrimination through attributes (see Fig. 4.3, Nascimento, 2005, p.99).

2.2.4 Areas of Application

Nascimento, 2005 divided the experiments of the FCPM model into two parts. First, a study of the model with artificial data sets, randomly generated wrt the FCPM model, to prove for the underlying assumptions of the model. Then, a study with real-world data sets.

These artificial data sets were constructed from a specific data generator, that builds the data accordingly to the model of the FCPM, with the original prototypes as extremes points. The artificial data served the purpose of studying the performance of the FCPM regarding the following proprieties:

1. To examine how the FCPM was able to recover the original prototypes from which the data had been generated and compare it to the ones retrieved by the FCM;
2. To observe the behaviour of the FCPM-0, while it shifts prototypes to outside the data space, and use it as an index of the number of clusters present in the data;
3. To study the performance of the FCPM when more clusters than those from which data has been generated were specified;
4. To compare the fuzzy partitions retrieved by FCPM against the FCM ones.

The dimensionality of the data sets generated ranged from 5 to 180, and the number of original prototypes, from 3 to 6. Due to the different behaviour that the data sets presented in the experiments, it was possible to partition the sets into 3 different types, regarding their dimensionality, small, medium and high. The experiments with artificial data sets were divided in 2 parts. First, the algorithms had to find the same number of prototypes as the ones that were generated. Then, they had to look for more than the ones generated. Some of the conclusions are presented next.

On the number of clusters found: When the dimensionality is low or intermediate, all the algorithms, FCPM-m ($m = 0, 1, 2$) and FCM found the correct number of clusters. For the higher type of dimensionality, only FCPM-1 and FCPM-2 found the correct number of clusters. For FCM some prototypes converge to the same stationary point and, for FCPM-0 some initial prototypes had been removed from the data cloud.

On the proximity to the original prototypes and the ones found by the FCM: When $m = 0, 1$, the prototypes found were closer to ones found by the FCM, and further from the original ones, as the prototypes for these models tend to be central points. As for the FCPM-2, found the closest to the originals, making it the furthest from the ones found by the FCM. The only exception being when the FCPM-0 shifts the prototypes to infinity. These results are transversal to all the types of dimensionality.

On partition separability: For FCPM-0 and FCPM-1 had partitions more contrasting than the ones the FCM found. The FCPM-2 had the fuzziest partitions.

On the number of iterations: the FCPM-1 and FCPM-2 had less than the FCM. For the FCPM-0, the number did not differ much from that in the FCM. Even so, the time it took for the FCPM algorithms to run was longer due to the minor iterations of the gradient projection method.

The algorithms had then to search for more prototypes than the ones generated, $c' = c_0 + 1$.

For the small dimensional data sets, FCM, FCPM-1 and FCPM-2 found $c' = c_0 + 1$, while the FCPM-0 removes the extra from the data space, $c' = c_0$.

For the intermediate, all the algorithms found the correct number of prototypes, $c' = c_0$. The FCM and FCPM-1 because the extra prototype almost always converges to another one. As for the FCPM-0 and FCPM-2, they remove the extra from the data space.

Finally, for the high dimensional data sets, the FCM and FCPM-0 had "degenerate" solutions. For the FCM, several of the prototypes overlap, and for the FCPM-0, more than one was pushed out of the data space, preventing the algorithm to converge. For FCPM-1 and FCPM-2, they found $c' = c_0 + 1$.

The previous proprieties were also tested with real-world data sets. First, the Mental Disorders data set (Nascimento, 2005), providing some interesting results regarding the capacity of the FCPM-2 to find typological structure, especially for capturing Archetypal Types. This data set becomes particularly interesting due to its nature, "in which cluster prototypes, syndromes of mental conditions, are indeed extreme with regard to patients" (Nascimento, 2005, p.119). In this data set, there is always a subset of features that have extreme values and distinctly separate each class. So, each disease can then be characterized by an 'archetypal patient', that exhibit extreme psychosomatic values, and thus, defining a *syndrome* of mental conditions, or an 'underlying type' (Nascimento, 2005). Not only was the FCPM-2 able to reveal this extremes types (the underlying topology), but was also able to perform such discovery when the data set was modified by adding less expressed cases, i.e., the data set was augmented with artificial patients that exhibit less severe syndromes. These results show how much is the FCPM-2 sensitive to the most "discriminating" features.

Other data sets from the UCI Machine Learn (Lichman, 2013) were tested with results concordant with the artificial data sets results.

2.3 Archetypal Analysis

Sometimes, one wish not to represent a group by its mean, or a prototype that lies in the center of the group, but by some sort of "pure type", an extreme point, based on all the other individuals on the data set. Thus, a more general idea than in the FCPM model, is taken by archetypal analysis, where not only the points are a convex combination of the prototypes, but the prototypes are also a convex combination of the points, creating this "pure types". Such model was proposed by Cutler and Breiman, 1994. They used a statistical method to discover this "pure types", by synthesizing a set of multivariate observations through a few points, which lie on the boundary of the data scatter, *i.e.* on the convex hull.

In Archetypal analysis (AA) each individual is represented as a mixture of "pure points" or, archetypes, and, each one is restricted to be a mixture of the individuals. This method can be used as a dimensionality reduction or as a clustering algorithm, where each archetype is easily interpretable by human experts.

2.3.1 Method

Formally, we want to find a matrix $Z = z_1, z_2, ..., z_c$ of archetypes, given a data set $X = x_1, x_2, ..., x_n$, where X has n observations and p attributes and Z has c archetypes and p attributes.

So, each archetype, z_j, is a convex combination of the data points

$$z_j = \sum_{i=1}^{n} x_i . b_{ij}, \tag{2.13}$$

constrained to

$$b_{ij} \geq 0, \tag{2.14}$$

and

$$\sum_{i=1}^{n} b_{ij} = 1. \tag{2.15}$$

Equation (2.14) makes the archetypes resemble the data and (2.15) is so that the archetypes are convex mixtures of the data.

Then the data are best approximated by a convex combination of the archetypes, minimizing

$$\|x_i - \sum_{j=1}^{c} z_j . a_{ji}\|^2, \tag{2.16}$$

constrained to

$$a_{ji} \geq 0, \tag{2.17}$$

and

$$\sum_{j=1}^{c} a_{ji} = 1. \tag{2.18}$$

Again, the model imposes a restriction of positivity (2.17), making each point a meaningful combination of the archetypes, and (2.18), imposes that each point is a mixture of archetypes. In order to find a suitable choice of archetypes, $z_1, z_2, ..., z_c$, it's necessary to minimize the residual sum of squares (RSS):

$$RSS(c) = \min_{a,b} \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{c} z_j.a_{ji}\|^2 = \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{c} \sum_{k=1}^{n} x_k.b_{kj}.a_{ji}\|^2. \tag{2.19}$$

Sometimes it's simpler to use the matrix notation of the RSS, making the former equation (2.19) as

$$RSS(c) = \|X - ZA\|^2 = \|X - XBA\|^2. \tag{2.20}$$

To minimize the RSS and find the Z matrix So, to find this Z matrix, we need to discover both the A and B matrices which requires an alternating optimization algorithm.

2.3.2 Algorithm

Solving the convex combinations for the archetypes (2.13) and for the data points (2.16), while minimizing the residual sum of squares (2.19) is a non-trivial task, as using a general-purpose constrained non-linear least squares algorithm is only practical for the smallest of the problems, due to its high computational costs (Damle and Sun, 2016; Mørup and Hansen, 2012; Chen et al., 2014; Eugster and Leisch, 2009; Bauckhage and Thurau, 2009).

To solve the clustering criterion (2.19) for optimal coefficients a_{ji} and b_{ij}, Cutler and Breiman, 1994 proposed an alternating constrained least squares algorithm.

This method alternates between finding the best a's for a given set of b's, and finding the best b's for a given set of a's. Each step demands the solution of several convex least squares (CLS) problems of the form:

Given u and $t_1, ..., t_q$, find $w_1, ..., w_q$ to minimize

$$\|u - \sum_{k=1}^{q} w_k t_k\|^2, \tag{2.21}$$

subject to $w_k \geq 0$ for $k = 1, ..., q$ and $\sum_{k=1}^{q} w_k = 1$. With each solution of the CLS, the RSS in (2.19) is reduced. The algorithm stops when a threshold for the reduction has been achieved or enough time has passed.

Given some initialization of the archetypes $Z = z_1, z_2, ..., z_c$, start by finding the best a_{ji}, solving n CLS problems. To find them, is necessary to minimize for the a_{ji} in the

convex combination of the archetypes (2.16) for each i, subject to the constraints (2.17) and (2.18). Each of this CLS problems has n observations and c variables.

Next, recalculate the archetypes, $\tilde{z_1}, \tilde{z_2}, ..., \tilde{z_c}$, from the updated a_{ji}, solving the system of linear equations given from the RSS (2.20) for $\tilde{Z}$.

With the new $\tilde{Z}$, find the best b_{ij} from the convex combination of the data points (2.13), solving c CLS problems, subject to the constraints (2.14) and (2.15), where each of this problems has n variables and p observations.

Update the archetypes $Z = z_1, z_2, ..., z_c$ with: $Z = XB$

Finally, compute the RSS and evaluate the improvement.

To solve the several CLS problems, Cutler and Breiman, 1994 implemented a penalized version of the Non-Negative Least Squares (NNLS) algorithm.. Using this penalized version of the NNLS, $\tilde{u}$ and $\tilde{t_1}, \tilde{t_2}, ..., \tilde{t_k}$ can be found by adding an extra element M to u an to $t_1, ..., t_q$ in the generic CLS model (2.21):

$$\|\tilde{u} - \sum_{k=1}^{p} w_k \tilde{t_k}\|^2 = \|u - \sum_{k=1}^{p} w_k t_k\|^2 + M^2 \|1 - \sum_{k=1}^{p} w_k\|^2, \tag{2.22}$$

that is minimized under non-negativity restrictions. The value M can enforce the equality constraint to be approximately satisfied, by setting it to large values, and thus, dominating the second term, while maintaining the non-negativity constraint. There are several methods to increase the efficiency of the algorithm (Damle and Sun, 2016; Mørup and Hansen, 2012; Chen et al., 2014; Eugster and Leisch, 2009; Bauckhage and Thurau, 2009). In this work, the implementation of Mørup and Hansen, 2012 will be the one used. Besides some modifications to increase the speed, e.g. the *FurthestSum* method to select c points close to the data boundary, and as far from each other as possible, this approach uses a simple projected gradient method to solve the AA problem. In this implementation the authors set the maximum number of iterations to 500. This limit is also adopted in this work.

2.3.3 Main Characteristics

Cutler and Breiman, 1994 demonstrated that for c>1, the archetypes that minimize the RSS (2.19) fall on the convex hull of the data, making the archetypes extremes data-values. For c=1, the sample mean minimizes the RSS. Also, there is no condition that makes the archetypes being observables points, and this can be seen as a drawback (Vinué et al., 2015).

Another interesting propriety of the archetypes is that they do not nest, i.e., as more archetypes are found, the existing ones can change, trying to get a better grasp of the shape of the data.

The convergence of the alternating optimization is also proven, although, without guarantee that it will be to a global minimum. Thus, it's advised that several runs of the algorithm are performed, with different initial seeds.

To select the number of archetypes to use, Cutler and Breiman, 1994 suggested the use of the "elbow criterion". This method consists on running the algorithm several times, for different numbers of archetypes, and use the "flattening"of the curve of the RSS values to choose a proper value.

The presence of outliers can also impose a problem. Usually, in clustering applications, there is always the need to pre-process the data and deal with the outliers, but, due to the imposition on the location of the archetypes, on the convex hull of the data, archetypal analysis can be quite sensitive to them and may need special attention (I. Epifanio, 2013; Cutler and Breiman, 1994; Chen et al., 2014; Eugster and Leisch, 2011). However, archetypes are not outliers (Eugster and Leisch, 2011) as the definition for both of them are profoundly and significantly different. It is important to stress this difference as the misunderstanding is quite easy to make.

2.3.4 Areas of Application

Even though it presents some problems, AA has found its way into several industries, as it presents a singular way to cluster the data, retrieve their representatives as archetypes, and use particular visualization techniques to interpret the found groups. Some of them are the gaming and behaviour analysis (Drachen et al., 2012; Sifa and Bauckhage, 2013; Pirker et al., 2016), sports (Eugster, 2012; Vinué and Epifanio, 2017), physics (Stone and Cutler, 1996; Stone, 2002; Chan et al., 2003), medicine and health care (Huggins et al., 2007; Römer et al., 2012; Thøgersen et al., 2013; Fehrman et al., 2017), benchmarking and profiling (Porzio et al., 2006; Porzio et al., 2008; Eugster, 2012; Seiler and Wohlrabe, 2013; Ragozini and D'Esposito, 2015), banking (Yeh and Lien, 2009), computer vision (Marinetti et al., 2006; Marinetti et al., 2007; Thurau and Bauckhage, 2009; Xiong et al., 2013) and nominal observations (Seth and Eugster, 2016).

2.4 Comparing FCM, FCPM and AA

Although the 3 methods find fuzzy clusters, they have substantial differences, that lead to fuzzy partitions with distinct characteristics, and, consequently, need different interpretation for the results.

First, their aim is different, and that translates to distinct clustering criteria. This means that each one of them have its own way of minimization. As an example, the FCM

uses a simple alternation optimization, where the FCPM-m requires an iterative process of its own in the alternation optimization.

Second, the location of the prototypes. While the FCM construct the prototypes as the mean of the clusters, the FCPM-m pushes them to the frontier of the data cloud. In AA, the points that represents clusters, are not prototypes, but archetypes, and are located, almost exclusively, in the convex hull of the data.

As the value of m for the FCPM-m can be any value from $\{0, 1, 2\}$, the same cannot be said about the FCM, as values different than $m = 2$ may give poor results (Bezdek, 1981).

The interpretation of the results varies according to the model: In the FCM, the membership degree is viewed as the proximity to a cluster center; In the FCPM-m, the membership also says how much of the prototype is expressed in the entity; In the AA, the archetypes are extremes points, characterized by a subset of features of the feature space taking extreme values.

However, they still share some problems:

1. The need to a careful initialization of the algorithm;
2. The choice on the number of clusters;
3. The similarity (or dissimilarity) function to use;
4. The sensibility to outliers;
5. The speed of convergence;

Chapter 3

On Clustering Manifolds

3.1 Generating Data with Cluster Tendency

When introducing a new algorithm, or an improvement of an existing one, the researchers should perform an extensive and systematic study with different types of data. Only then, it's possible to have a concise and clear evaluation of the algorithm. This would enable any researcher who desires to improve an algorithm, or perform a comparative study, to have a simple method of doing it. This is particularly important when there is no ground truth, that is the case of unsupervised learning (Zimmermann, 2015). However, researchers often only use a few, and specific data sets (either artificial or from the real-world), measuring only the times of execution, comparing the number of found clusters and assessing the quality of the found clusters with validation indices. Although this is a valid, and necessary approach, it lacks an exhaustive evaluation regarding the behaviour of the algorithm. Zimmermann, 2015 summarized this problem in three aspects:

1. There is no way of quantitatively evaluate the performance of the algorithms. These algorithms, more often than not, are not reassessed with additional data after their publication, or compared with other algorithms. Proprieties such as transitivity are assumed in most cases.

   Proprieties such as transitivity are assumed in most cases. For instance, improvements to an algorithm are tested against the same portfolio of data sets, which may led to some unjustified generalizations (Zimmermann, 2015). These generalizations may led to the observation of the desired proprieties, with small and restricted artificial data, that do not uphold with real-world data (Zimmermann, 2015).

2. There is no empirical evidence of how to choose good parameters settings. This

leads to a poor understanding about the relationship between the parameters and results. The behaviour of the algorithm might only be known for a small set of settings, making it hard to understand how small changes influence its behaviour.

3. If the algorithm is really mining the generative processes underlying the data, or, if all relationships captured are meaningful.

Without a clear answer to these questions, it's not possible to assess if the algorithm is truly fulfilling the purpose for which it was build. Meaning that, even if the patterns are successfully identified, it lacks the knowledge to know how those patterns relate to the process that generated the data. It becomes ambiguous how to exploit those patterns in the original domains (Zimmermann, 2015).

To systematically answer the previous problems, it's necessary to use several heterogeneous artificial data sets, in which it's possible to control the dimensionality of the data space, the number of initial clusters, the underlying distribution, among other parameters (Pei, Yaling; Zaiane, Osmar, 2006; Albuquerque et al., 2011; Zimmermann, 2015; Adä and Berthold, 2010). Only by means of this variation, can a complete evaluation of an algorithm be provided. This implies that, before an algorithm is confronted with real-world data sets, it needs to be tested against artificial data, therefore, assessing that the algorithm does in fact behave as proven by the theory, and presents the results exhibited with the artificial data. By following this framework, new data sets are easier to approach by knowing which tools to use, the parameters settings, and, most importantly, what conclusions can be drawn from the results (Zimmermann, 2015).

With this idea in mind, artificial data sets will be used to compare the FCPM-*m* and the AA before using real-world data. Note that this is a comparative study, and the exhaustive study of the individual behaviour of each one of them is out of the scope of this work, as both of them have already undergone an individual evaluation (Nascimento, 2005; Madaleno, 2017). Two data generators are considered in this work, one for each model. These data generators will presented in the next sections.

3.1.1 FCPM Data Generator

To evaluate the FCPM, Nascimento, 2002 developed a data generator (Figure 3.1) according to the FCPM model, the FCPM-DG. This generator was build following the assumptions of the underlying FCPM model. (1) That the model of data generated contains a cluster structure; (2) In the mentioned structure, any entity bears a proportion of each prototype, that is a model or ideal point.

In this data generator, the parameters are randomly generated from user defined intervals. First, the minimum and maximum for the dimensionality of the data space (p), [*min_DimP*, *max_DimP*]. Then, the interval for the number of clusters (c_0) to be generated,

[*min_C*, *max_C*]. Finally, from [*min_PtsClt*, *max_PtsClt*], it comes the minimum and maximum for the number of entities to be generated within each cluster $(n_1, n_2, ..., n_{c_0})$.

The data generator was designed as it follows:

1. Define the c_0 clusters directions using the following technique: from a pre-specified hyper-cube with side length [*min_HCube*, *max_HCube*], generate random vectors $o_i \in \mathbb{R}^p (i = 1, ..., c_0)$. Then, their gravity center o is taken as the origin of the space. Each cluster direction is taken as the segment $\overrightarrow{oo_i}$. In the original work, the values for the hyper-cube were $[-100.0; 100.0]$.

2. Define two p-dimensional sampling boxes, for each i $(i = 1, ..., c_0)$. The first box, within bounds $A_i = [(1 - percent\_DSeg).o_i, (1 + percent\_DSeg).o_i]$ (*e.g.* $[0.9.o_i, 1.1.o_i]$) and the other, within $B_i = [o, o_i]$. Then, for each box A_i generate randomly a small percentage of points, *percent_PtsOrgVs* (*e.g.* $0.2n_i$). Generate, also randomly, the remaining points $(1 - percent\_PtsOrgVs).n_i$ (*e.g.* $0.8n_i$) for each box B_i.

3. All data generated (including the c_0 original prototypes) are normalized by centering to the origin and scaling by the range of features.

Besides the dimensionality of the data space, the number of the cluster and the number of entities within each cluster, the length of the cube ([*min_HCube*, *max_HCube*]), the side length of box A_i (*percent_DSeg*) and the percentage of points to generate in box A_i (*percent_PtsOrgVs*) are also user defined parameters. The randomly generated items were withdrawn from a uniform distribution in the interval [0,1]. Figure 3.1 contains a example of a synthetic data set.

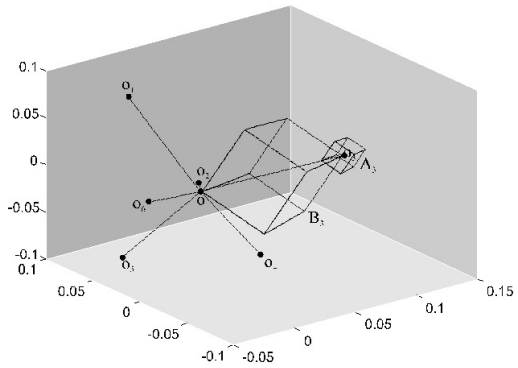


Figure 3.1: Example of the architecture of the FCPM data generator, using the three best principal components, on a 3D projection. This example contains six original prototypes and two illustrative boxes, A_3 and B_3, for prototype 3. Original from Nascimento, 2005

3.1.2 AA Data Generator

There are no references in the literature for a data generator for the underlying model of Archetypal Analysis. Even so, it's possible to somehow build a simple generator (AA-DG), that builds data that resembles the model of AA (Figure 3.2).

One approach is to take in consideration the model of AA (2.20). The data matrix, X, is derived from the following expression,

$$X = ZA, \tag{3.1}$$

in which the matrix A needs to follow the constrains of the AA model, namely, all its values must be positive (2.17), and the columns must sum to one (2.18). Mørup and Hansen, 2012 applied this principle[1], and their idea is followed here.

A matrix $\tilde{A}$ is created, where the constrain 2.18 is relaxed, in a sense that the sum of its columns might not be exactly 1, but the mean of the sums is close to 1. Then, generate c points on the surface of p-sphere, with radius 1, that will represent the archetypes. This generation can be done either with an artificial sampling, or manually selecting the location of the archetypes. There are several ways in the literature on how to sample points on the surface of an m-sphere, here the Marsaglia, 1972 method is used. This creates the $\tilde{Z}$, as the constrains for Z are overlooked. In the end, the data matrix is given as,

$$X = \tilde{Z}\tilde{A}. \tag{3.2}$$

It is also possible to add noise to the data, by multiplying the points with a uniform distribution up to a percentage. Figure 3.2 contains two data sets, one with noise, another without.

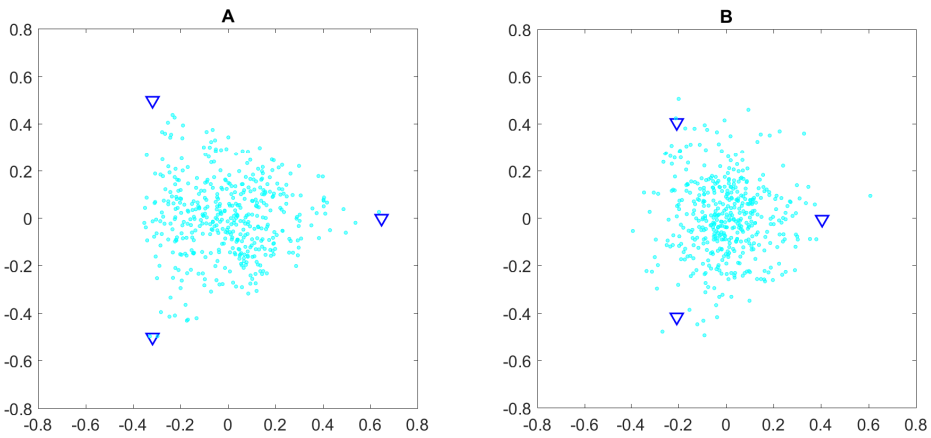


Figure 3.2: Two artificial data sets, with the archetypes equidistant. On the left the data set was generated without noise, and on the right, with 0.2 of noise.

[1] On the matlab code provided by the authors

This data generator, although it allows to observe the behaviour of the algorithm regarding the recovery archetypes, it doesn't allow for much more, and it comes with some problems. First, it is not possible to control the separability of the clusters, or the compactness. Second, generating points on a hyper-sphere it's a non-trivial task, most of the time leading to points to close to each other. Third, it is highly dependent on input parameters, to make the sum of $\tilde{A}$ be close to 1, as these values change with the dimensionality of the space. Finally, and most important, there is no guaranty that data generated follows the archetypal model.

There are other ways to generate this data. One way is to define a set of archetypes and compute its convex hull, then, create the smallest hyper-cube containing the convex hull, generate points with a uniform distribution inside this hyper-cube, retaining only those that are inside the convex hull. Repeat until the number of points inside the convex hull is the desired. This generator is not dependent on the input parameters, as the previous one, making it easier to use. Except for the location of the original archetypes, it fails to consider the underlying model of AA.

This is still a open research problem.

3.2 Initializations Strategies with Extreme Points

As Mørup and Hansen, 2012 state, it's useful to initialize the AA algorithm with extreme points. This section is dedicate to introduce such methods. First, the Furthest Sum. Second, the Iterative Anomalous Pattern, and a modification to the original algorithm to return extreme seeds.

3.2.1 Furthest Sum Algorithm

The Furthest Sum Algorithm, proposed in Mørup and Hansen, 2012, is a method that takes in consideration the location of the archetypes and as such, it selects c points in the convex hull of the data to be used as seeds. It iteratively chooses points further from the center of data, and as far way as possible from each other. The c number is a user defined parameter. The authors also proved that the c selected points are guaranteed to lie in the minimal convex set of unselected data points.

3.2.2 Iterative Furthest Point Algorithm

As demonstrated in Nascimento and Franco, 2009, the Iterative Anomalous Pattern (IAP) presented good results with the FCM, in unsupervised segmentation of Sea Surface Temperature (SST) images. This method, not only serves as an initialization scheme for an algorithm, but it's also capable of acting as an indicator of the number of clusters present in the data.

For IAP algorithm, $X = [x_{kh}]$, a $n \times p$ entity-to-feature data matrix, with $k = 1,..,n$; $h = 1,...,p$, needs to be preprocessed into Y, by shifting X the origin to the gravity center, the

grand mean. The center of Y is the point $O_y = 0_1, 0_2, ..., 0_p$. It then uses the total data scatter of all data points,

$$T(Y) = \sum_{i=1}^{n} \sum_{h=1}^{p} y_{ih}^2, \tag{3.3}$$

and the relative contribution of a cluster (S_t, v_t) to the data scatter as

$$W((S_t, v_t)) = \frac{|S_t| \sum_{h=1}^{p} v_{th}^2}{T(Y)}, \tag{3.4}$$

as measures to evaluate the found anomalous patterns. $|S_t|$ is the cardinality of cluster S_t,

To find t^{th} anomalous pattern, the algorithm initializes the new cluster seed v^*, as the farthest point from O. Then, it defines S_t as the set of entities closer to c^* than to the origin O_y: $S_t = \{y_i \in Y : d(y_i, c^*) < d(y_i, O_y)\}$. The new centroid v is computed as the gravity center of S_t. This new centroid is then compared with the old one. If $v^* \approx v$, S_t is considered as the t^{th} anomalous patter and $v_t = v$ its centroid: (S_t, v_t). Otherwise, define $v^* = v$ and continuously update S_t until the new centroid no longer differs from the previous.

Update the Y by removing the points assigned to the found cluster, $Y_{t+1} = Y_t \backslash S_t$, and repeat this process until one of the following stopping criteria is met: i) All entities have been assigned; ii) The t^{th} cluster has a relative contribution (Eq. (3.4)) to the data scatter (Eq. (3.3)) lower than a pre-specified value, τ; iii) The total contribution of the first t clusters reaches pre-specified threshold, δ; iv) The number of found clusters reaches a pre-specified value, $t == k_{max}$.

In this work, two different settings for the stopping conditions of the IAP were used:

- First: τ, δ and k_{max} were set to large enough values, in order to allow the algorithm to assign all entities to clusters. Then, all clusters with a relative contribution of 5% ($W((S_t, v_t)) > 0.05$) were selected. The threshold value of 0.05 was fixed empirically as a result of running several experiments and observing the relative contributions of the found clusters. Here, not only the algorithm returns the seeds to initialize an algorithm, but also return the number of groups, k. This setting was named IAP ($s \geq 0.05$).

- Second: the number of clusters to retrieved was restricted to the number of labels in the data set: $k_{max} = c_0$. Here, the algorithm returns the first c_0 clusters found, independent of their relative contribution to the data scatter. This setting was named IAP ($k == c_0$).

Since this study focuses on retrieving extremes ideal points and the IAP algorithm returns the seeds as means points of the clusters, the algorithm was modified to return the seeds as extreme points of the clusters. In this version, for each anomalous pattern found, instead of returning v as the gravity center, it returns the initial seed of the cluster,

the farthest point from O_y. For the stopping criteria, it uses the same threshold of 0.05, as in the IAP ($s \geq 0.05$) and it was coined the Iterative Furthest Point, IFP ($s \geq 0.05$).

3.3 Assessing the Quality of Fuzzy Partitions

A fundamental problem in cluster analysis is how to evaluate the clustering results, *i.e.*, given some input parameters, how well the resulting partitions represent the natural or underlying grouping of the data (Dunn, 1973; Bezdek, 1973; Kryszczuk and Hurley, 2010; Arbelaitz et al., 2013; Chouikhi et al., 2015). This is a non-trivial problem and requires careful thought. Otherwise, without a systematic evaluation process, it's not possible to infer conclusions about the results, without being susceptible to bias or inadequate interpretations. From this necessity, indices to evaluate the performance of the clustering algorithms become a well-address problem in the literature (Bezdek, 1973; Dunn, 1973).

Such indices, known as the Clustering Validity Indices (CVI), not only allow the comparison of different algorithms, but also, the results of the same algorithm with different parametrizations. This is especially important for algorithms that are highly dependent on input parameters, *e.g.* the number of clusters, which is rarely known beforehand.

Although there are several CVI's proposed in the literature, none of them is capable of providing a good measurement on its own (Arbelaitz et al., 2013; Chouikhi et al., 2015). It's has become standard to use several CVI's and combining them with a fusion strategy (Yera et al., 2017; Kryszczuk and Hurley, 2010).

Even with evaluation indices, it's not always possible to understand the results of the clustering process, how the data is organized or what is the clustering tendency, specially in data sets with high dimensionality. Although the validation process can assess the goodness of results, and indirectly, how appropriate were the chosen parameters, visualization techniques are a must, as they bring the human insight to the whole process. Since a cluster algorithm always fits the data to the clustering model, this human knowledge becomes even more indispensable to understand the adequateness of the clustering solution.

Also, in the real world, the users of a clustering process, most often than not, are not experts in machine learning, and, as such, the results often require an interpretation and translation of number and metrics to human perception.

This chapter is divided in three distinct parts. First, a discussion on data recovery, as a way to evaluate the clustering result when using artificial data. Then, a brief review on existing CVI's, and fusions strategies to combine them. In the end, a section dedicated to the visualization techniques.

3.3.1 The Clustering Data Recovery

As proposed by Mirkin, 2005, one way of understanding, not only the structure of data, but also the effectiveness of the clustering algorithm, is the data recovery paradigm mentioned in the Introduction. This paradigm is useful to study the AA and FCPM due to the algorithms treating the groups representatives as ideal types.

By using artificial data, where the data points are generated from "ideal types" (*e.g.* the FCPM-DG, from Section 3.1.1), it becomes possible to measure the data recovery of an algorithm, and assess its ability in recovering the original clusters. This is done by measuring the difference between the found prototypes, $V' = \{v'_j\}_{j=1}^{c'}$, and the original ones, $V = \{v_j\}_{j=1}^{c}$, where c represents the number of prototypes generated and c' the number of retrieved. The closer to the original prototypes are the retrieves, the higher is the data recovery capacity of the algorithm.

To compute this distance, Nascimento, 2005 introduced a Dissimilarity Coefficient D, defined as squared relative quadratic mean error between the original prototypes V, and the found ones V',

$$D(V', V) = \frac{\sum_{i=1}^{c}\sum_{h=1}^{p}(v'_{ih} - v_{ih})^2}{\sum_{i=1}^{c}\sum_{h=1}^{p} v_{ih}^2 + \sum_{i=1}^{c}\sum_{h=1}^{p} v_{ih}'^2}. \tag{3.5}$$

When applying D, if the number of found prototypes (c') by an algorithm is smaller than the original ones (c), only (c') "reference" prototypes participate in (3.5). This measure is non-negative and it equals to 0 when $v_{ih} = v'_{ih}$, for all $i = 1,...,c; h = 1,...,p$. When the components of each v_i and v'_i are in the same orthants, then D is not greater than 1.

The Dissimilarity Coefficient requires a matching between the retrieved prototypes and the found ones. This matching is done using a K-NN distances with $K = min(c', c_0)$. In the event of a tie between two prototypes, one of them is matched to its next closest reference prototype.

3.3.2 Five Premier Fuzzy Validation Indices

The CVIs are divided according to the source of information they use to assess the clustering results. If they use the labels of the data set, that is information not contained in the clustering solution, they are external validation indices, otherwise, they are internal validation indices.

Even though it's always good practice to use external validation indices when ground truth is available, the classification boundaries are not well-defined in fuzzy clustering. As such, these indices were not considered for the comparison of the algorithms in this work.

The internal validation indices evaluate the clustering solution by measuring proprieties of the final cluster structure, such as the compactness of the clusters, how well separated are the final clusters, or the (dis)similarities between clusters. The external indices work by comparing the clustering solution to the labels of the data set, assessing

the capacity of the algorithm in finding a cluster structure that relates to the ground truth.

To this date, aside from one (very) recent proposal (Suleman, 2017), there are no mentions in the literature of validation indices for archetypal analysis. Although there are several fuzzy clustering indices (Chouikhi et al., 2015; Arbelaitz et al., 2013), only 5 of them were considered. These indices are implemented in the R language toolbox in Ferraro and Giordani, 2015:

1. Partition Entropy (PE): This index measures the separation of clusters by looking at the information entropy of the memberships values (u_i),

$$PE(c) = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\log_2 u_{ij}. \tag{3.6}$$

It's contained in the interval $0 \leq PE \leq log_2 c$, with c as the number of clusters. The minimal value of PE (0) corresponds to the optimal number of clusters.

2. Partition Coefficient (PC): It measures the "overlap" between clusters by averaging through the squared memberships,

$$PC(c) = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2. \tag{3.7}$$

It's contained in the interval $\frac{1}{c} \leq PC \leq 1$, with c as the number of clusters. The maximal value of PC corresponds to the optimal number of clusters.

3. Modified Partition Coefficient (MPC): It's the PC normalized,

$$MPC(c) = 1 - \frac{c}{c-1}(1 - PC(c)), \tag{3.8}$$

to be contained in $0 \leq MPC \leq 1$. As in the PC, its maximum value corresponds to the optimal number of clusters.

4. Xie-Beni (XB): This index computes the *ratio* between the sum of the squared within-cluster distances weighted by the respective memberships to the power of m (compactness of clusters), and the minimum squared distance between all pairs of prototypes (separation of the clusters), multiply by the number of points (N),

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \|V_i - X_j\|^2}{N \min\limits_{i,j} \|V_i - V_j\|^2}. \tag{3.9}$$

The within-cluster distances are computed with respect to the norm used in the clustering algorithm. This index does not contain a maximum value, but is confined to the

lower bound of 0, that is the optimal value. In this work, $m = 2$

5. Fuzzy Silhouette Index (FSI): This index is an adaptation of the Average Silhouette Width Criterion, or Crisp Silhouette (CS), made for hard clustering algorithms. In the CS, for each point $j \in \{1, 2, ..., N\}$, its silhouette (s_j) it's computed as the difference between the average distance of j to all the other points of the cluster to which j belongs ($a_p j$), and the average distance of j to all the points in the closest neighbouring cluster ($b_{qj}, q \neq p$),

$$s_j = \frac{b_{qj} - a_{pj}}{\max\{a_{pj}, b_{qj}\}}, \tag{3.10}$$

normalized by the maximum between a_j and b_j. In the end, CS is defined by the average of all s_j,

$$CS = \frac{1}{N} \sum_{j=1}^{N} s_j. \tag{3.11}$$

The FSI differs from the CS by not averaging the s_j with an arithmetic mean, but with a weighted average, where the weight of each term is the difference between the first (u_{pj}) and second (u_{qj}) largest elements in the fuzzy matrix U, to the power of α, for the each point,

$$FSI = \frac{\sum_{j=1}^{N} (u_{pj} - u_{qj})^{\alpha} S_j}{\sum_{j=1}^{N} (u_{pj} - u_{qj})^{\alpha}}. \tag{3.12}$$

The FSI is contained in $0 \leq FSI \leq 1$. The higher the FSI value, the better is the clustering solution. In this work $\alpha = 1$.

3.3.3 Visualization of Fuzzy Partitions

To get a sense of the raw data, the features distributions and characteristics, the most common techniques from data analysis will suffice.

To visualize data in low dimensions, two dimensionality reduction techniques were used, Principal Components Analysis (PCA) and Sammon mapping Sammon, 1969.

The first technique (PCA) focus on reducing dimensionality preserving the variances of the data.

Consider a p-dimensional data set X. The quality of the projection of X, into a r-dimensional space Y, $r < n$, obtained from PCA can be reflected by the quantity

$$R = \frac{\sum_{j=1}^{r} (\lambda_j)^2}{\sum_{j=1}^{p} (\lambda_j)^2}, \tag{3.13}$$

with λ as the eigenvalues of the covariance matrix. R measures the ratio of the total variance of the data captured by the r-projection.

Contrary to PCA, the Sammon Mapping, is a non-linear mapping that tries to preserve the interpattern distances, and projecting it in a 2D space. Balasko et al., 2005 provides an implementation for the Sammon mapping. This implementation in 2D, also plots the memberships values with a contour map, of the resulting clustering solutions, for the PCA and Sammon mapping. Figure 3.7 shows an example of both visualizations, with the contour maps.

The discussed clustering algorithms are extremely dependent on input parameters, so, a necessity arises on how to systematically discover a good set of parameters.
One of the most important parameters, is the number of clusters to choose, as it deeply influences the clustering results. From this necessity, the Visual Assessment of Tendency (VAT) was introduced by Bezdek and Hathaway, 2002. This technique serves as a visual heuristic to inspect the cluster tendency of the data and the underlying number of clusters. It uses an ordered dissimilarity matrix (the pair-wise Euclidean distance fo the data) to plot the Ordered Dissimilarity Image (ODI).

Retrieving the number of clusters from the ODI is relatively easy, particularly for data sets with well separated clusters (Hu and Hathaway, 2008). It's only necessary to follow the diagonal of the ODI and count the number of squared shaped dark blocks. A well separated data set results in more noticeable squares, Figure 3.3.

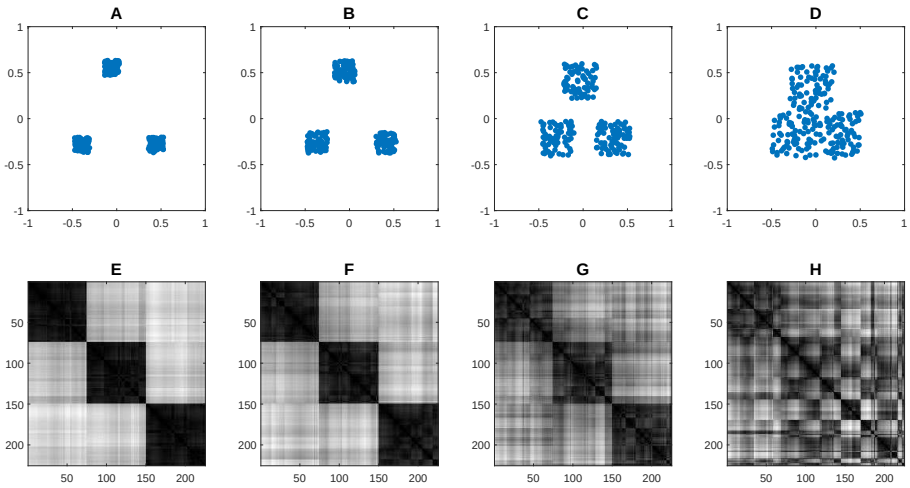


Figure 3.3: Plots for 4 different synthetic data sets and their corresponding ODI, with an increasing difficulty in retrieving the number of clusters. Each data set is composed of 150 points, evenly distributed in 3 clusters, with each cluster having an uniform distribution. From left to right, the clusters become closer, and the corresponding ODI becomes harder to evaluate. In the first ODI (E) it's easy to see the correct number of clusters. In contrast, the last ODI (H) gives no valuable input as the number of clusters present in the data. Even such cases, the ODI tells that the data set does not contain a clear cluster structure.

Even with the previous techniques, is not always easy to determine the correct number of clusters, *e.g.* Figure 3.3. For the AA, Cutler and Breiman, 1994 suggested finding this number *a posterior* by plotting the value of the Residual Sum of Squares (RSS) (Eq. 2.19)

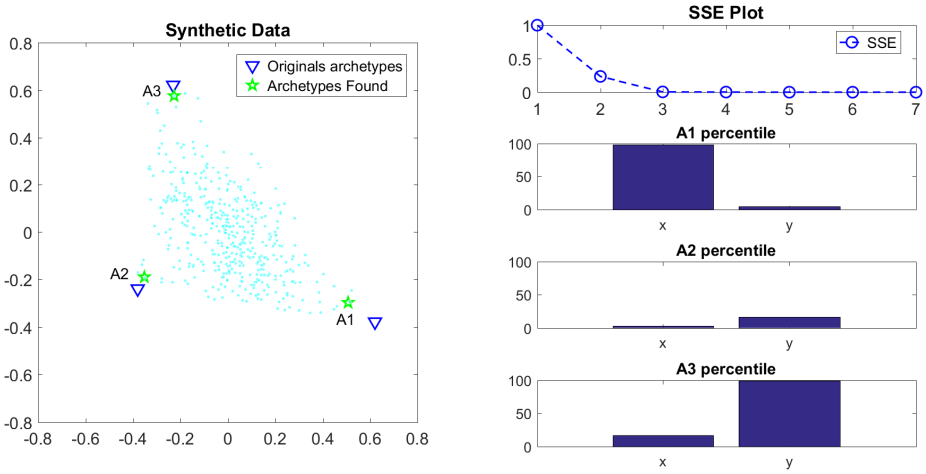


Figure 3.4: On the left, an artificial data set, generated according to the AA-DG. It contains 3 archetypes, that are not equidistant. On the top right, the SSE plot for the artificial data showing the flattening of the curve on the 3^{th} archetype, which is in agreement with the process of generation for the artificial data set. The three bottom right plots correspond to the percentiles plots for each archetype shown in the left plot.

against the number of archetypes, as a heuristic to know how many archetypes are a good fit for the data. As in the "knee" plot used for the PCA, it's also necessary to look for a flattening of the curve. Since the implementation Mørup and Hansen, 2012 was used, the number of optimal clusters was determined by plotting the number of archetypes against the Sum of Square Errors (SSE), Figure 3.4.

To further explore the AA results with visualizations techniques, Cutler and Breiman, 1994 also suggested the use percentile profiles to compare archetypes, and mixtures plots to visualize the fuzzy memberships values.

Since archetypes are "extreme types", it's exceptionally useful to analyse the composition of each individual and how they differ from each other. To that end, a bar plot can be created, with the percentile values of each feature in an archetype as compared to the data, hereinafter referred as the percentile plot (Figure 3.4).

Mixture plots, or simplex visualizations (Seth and Eugster, 2015) are a useful technique to help relate the entities to the archetypes, through the fuzzy memberships values found by the AA. Cutler and Breiman, 1994 only used ternary plots (3 archetypes), but some authors extended it to p archetypes. Seth and Eugster, 2015 provide a detailed explanation about these visualization technique, and how this projections are possible. Some of their approaches are used here and were implemented to add additional features to the plots.

To build these mixtures plots, the archetypes are projected equidistantly on a circle, forming a polygon. Then, the data points are projected as convex combinations of the archetypes, using the fuzzy memberships values provided by the archetypal analysis.

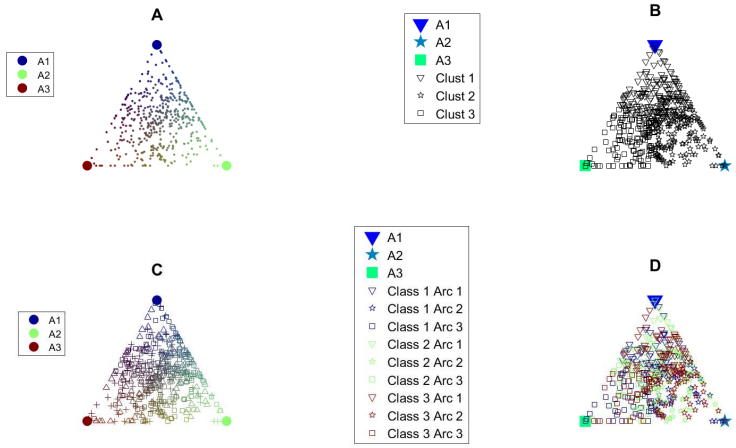


Figure 3.5: Top left: a mixture plot with the points coloured in function of how close they are to the archetypes, *e.g.*, they aim at representing the distribution of the memberships of each point to each archetype; Top Right: A mixture plot with the plots represented according to their higher membership value; Bottom left: a mixture plot with the points coloured, as in the top right, and shaped according to their class; Bottom right: Mixture plot with the points coloured according to their class and shaped according to their highest membership value; The data used for the mixture plots is the same as in Figure 3.4.

In these plots, the points that lie outside of the boundary defined by the polygon are projected to its frontier, Figure 3.5. The last 3 figures in the plot can also be seen as a transformation to crisp clustering, hardening the partition by maximum membership value, *i.e.* points are assigned to the cluster of maximum belongingness. A drawback of the mixture plot is the inability of representing solutions with only two archetypes.

However, the archetypes are not typical equidistant to each other. With the intention of trying to observe this unconformity, Seth and Eugster, 2015 proposed to rearrange the archetypes on a circle according to the distances in the original space. This implies an optimal order of the vertices, according to the distance. To solve this problem, a simple hill climbing algorithm is sufficient to find this optimal order, as normally the number of archetypes is usually small, Figure 3.6. Note that these two plots are the same as in the C and D in Figure 3.5, but with the archetypes rearranged.

The difference between PCA and Sammon mapping is visible in Figure 3.7, where both projections contain the contour map for the fuzzy memberships of a Fuzzy c-Means run.

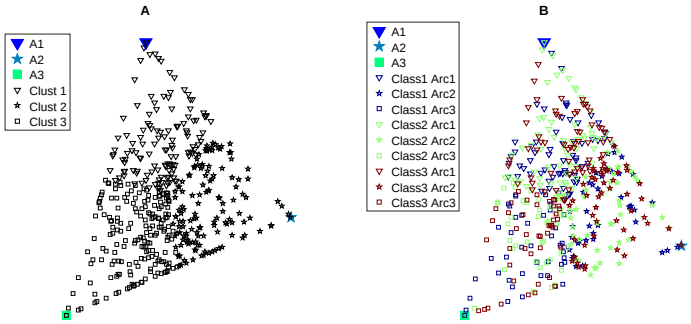
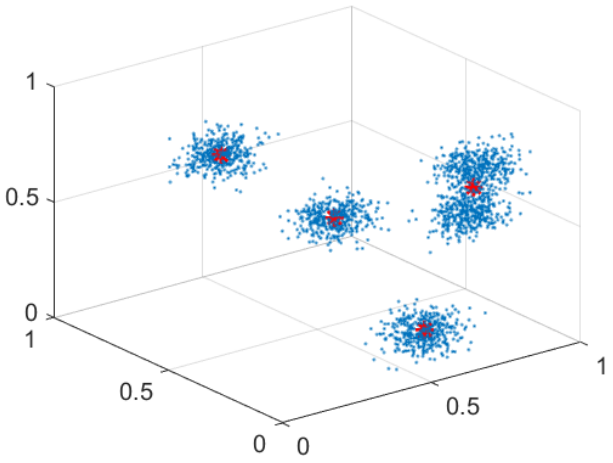
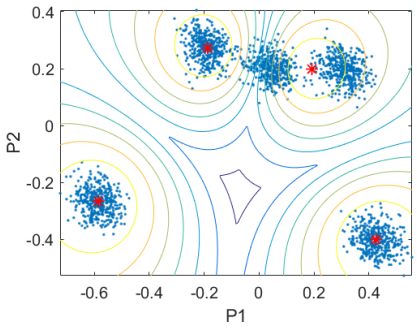


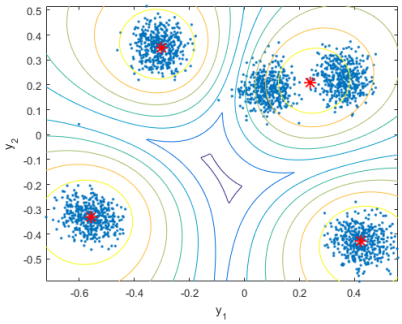
Figure 3.6: On the left, a mixture plot with the distances preserved and the points shaped according to their highest membership value. On the right, a mixture plot with the distances preserved, and with the points coloured according to their class and shaped according to their highest membership value. Remembering how the 1^{st} archetype is further from the 3^{rd} than the 2^{nd} (Figure 3.4), these plots represent this situation with good accuracy.



a Synthetic data with 5 clusters.



b PC projection (R=100%).



c Sammon mapping projection.

Figure 3.7: On top, an artificial data set with 400 points, and 5 clusters. The FCM was run searching for 4 clusters, resulting in a prototype being in the middle of two clouds of points. On the bottom left, a PC projection, where it seems as the 3 clusters are continuous. In the bottom right, the Sammon mapping, that clearly shows this two clouds are isolated from the third one. The clusters centers are in red in all plots.

Chapter 4

Comparing Fuzzy Proportional Membership Algorithm with Archetypal Analysis

To accomplish the goals proposed in the beginning of the dissertation, the algorithms were subject to a manifold of experiments, with different settings and a diverse data collection. This chapter describes those experiments, as well as their results. It's organized in three sections, where each one corresponds to a different study.

The first section focused on using synthetic data, and its divided in two parts. First, a comparative analysis of the data recovery proprieties and efficiency of the AA and the FCPM-*m* algorithms. Second, using the same synthetic data but augmented with outliers, the same analysis was made but as a measure of the of the algorithms in presence of outliers. The FCPM-0 behaviour, in shifting prototypes to outside the space was also studied as a possible indicator of the number of true clusters.

In the second section, the algorithms were run with real-world data and evaluated with fuzzy internal validation indices and visualization techniques. By applying those indices, it was possible to compare the algorithms, and observe which indices are more suitable for each algorithm. The visualization techniques allowed to validate the results for the AA algorithms.

In the final section, it's explored how the initialization of the algorithms can affect their efficacy (quality of the found partitions that it's quantitatively evaluated by validation indices) and efficiency (measure by the iterations needed to converge).

4.1 Comparative Study with Synthetic Data

As already stressed in the previous chapters, when studying an algorithm, it's extremely important to first use synthetic data with known statistical proprieties. Following this approach, the algorithms were analysed with 82 synthetic data sets generated by the

FCPM-DG (Section 3.1.1). These data sets are divided in three dimensionality sets, according to the ratio (r) between the data set dimensionality (p) and number of original prototypes (c_0) (Nascimento, 2005):
- low dimensionality ($r \leq 5$): with 19 data sets, $r = \left\{\frac{5}{3}, \frac{15}{3}\right\}$;
- medium dimensionality ($5 < r < 25$): with 52 data sets, $r = \left\{\frac{20}{3}, \frac{40}{4}, \frac{50}{4}, \frac{100}{5}\right\}$;
- high dimensionality ($r \geq 25$): with 12 data sets, having $r = \left\{\frac{180}{6}\right\}$.

To measure the data recovery proprieties of the algorithms, *i.e.* their ability in retrieving the originals prototypes, the dissimilarity index D (Eq. (3.5) in Section 3.3.1) between the original DG prototypes (V_{Org}) and the retrieved archetypes/prototypes, was applied.

4.1.1 Data Recovery Analysis on Synthetic Data

This experiment has two objectives: to study the ability of the AA algorithm in recovering archetypes on multidimensional synthetic data; to compare the data recovery proprieties of the AA algorithm with the ones of the FCPM-0 and FCPM-2, already obtained in Nascimento, 2005.

The AA algorithm was initialized by the Furthest Sum method (described in Section 2.3.2). Each algorithm was run 5 times, and the results are the average of the dissimilarity index D, of those 5 runs.

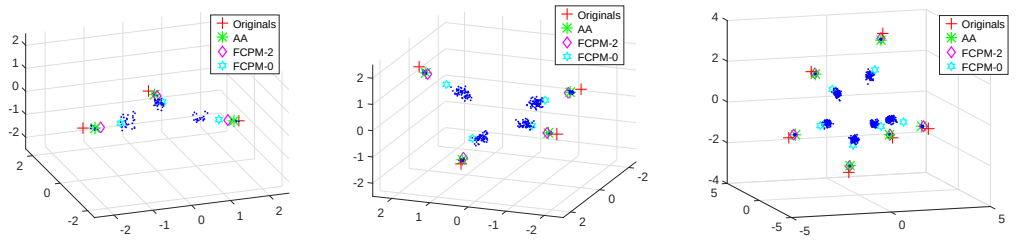
Table 4.1: Average Dissimilarity (D) values of AA archetypes to FCPM-DG originals V_{Org} and to FCPM2, FCPM-0 prototypes.

Dimensionality	AA vs V_{Org}	AA vs FCPM-2	AA vs FCPM-0	FCPM-2 vs V_{Org}	FCPM-0 vs V_{Org}
Small	0.008	0.006	0.109	0.021	0.156
Medium	0.007	0.001	0.167	0.011	0.200
High	0.006	0.000	0.217	0.005	0.228

The first column shows how the AA archetypes closely match the original ones. This is a clear indication of the ability of the AA algorithm to retrieve extreme prototypes, that are ideal points, and to reconstruct the original data from those retrieved ideal clusters. The AA archetypes are also very close to the prototypes found by the FCPM-2 (Figure 4.1). This is natural, as Nascimento, 2005 derived that the FCPM-2 always finds extreme prototypes, matching the FCPM-DG original ones, and the archetypes of the AA are located in the convex hull of the data (Cutler and Breiman, 1994).

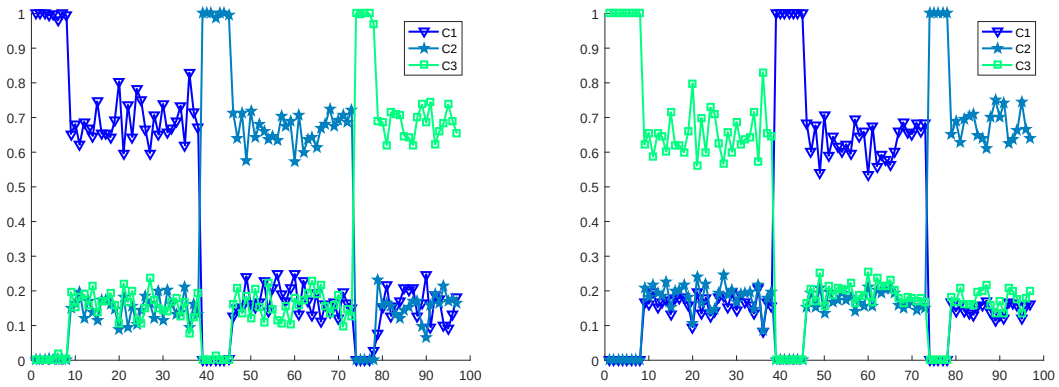
A close inspection to the values of the AA and FCPM-2 (first and fourth column) allows to see that, as the dimensionality increases, the prototypes found by the FCPM-2 became closer to the originals, than the archetypes found by the AA (Figure 4.1c).

For the FCPM-0, where the prototypes are typically central points (Figure 4.1), there is an increase in the dissimilarity, when compared to AA and FCPM-2. Also, in high dimensional spaces, the algorithm sometimes removes one of its prototypes out of the data space, resulting in a higher dissimilarity value.



a b c

Figure 4.1: PC projection for the 3 dimensionalities, showing the archetypes/prototypes found and the V_{Org}, for the data recovery study. In (a) for a small dimensional data set (R=99%) (n=97, p=20, c=3). In (b) for a medium dimensional data set (R=99%) (n=318, p=40, c=4). In (c) for a high dimensional data set (R=74%) (n=799, p=180, c=6). For the high dimensional data sets (c), it's possible to observe how the FCPM-2 prototypes are closer to the originals than the AA archetypes.



a For the AA. b For the FCPM-2.

Figure 4.2: Fuzzy memberships evolution for the partitions found by the AA and FCPM-2, for a medium dimensional data set (n=97, p=20, c=3), showing how both algorithms find solutions that match the data generation process of the FCPM-DG.

The cluster structure and memberships values of the AA solutions match with the generation process of the FCPM-DG, and are very similar to the solutions of the FCPM-2 (Figure 4.2).

Table 4.2: Average number iterations needed for the converge of the 3 algorithms, across the 3 dimensionalities. Where the major and minor iterations are described in Section 2.1.2

Dimensionality	AA	FCPM-0		FCPM-2	
	Iterations	Major	Minor	Major	Minor
Small	196	40	472	10	407
Medium	188	65	733	12	583
High	174	100	1935	34	1281

The FCPM-m algorithms consistently have fewer major iterations than the AA algorithm, Table 4.2. However its running time is higher because the current implementation is not yet optimized. Contrary to the AA, where the number of iterations doesn't seem to depend on the dimensionality, the number of iterations necessary in the FCPM increases with it. The FCPM-0 reaches the maximum number of iterations due to its behaviour of removing extra prototypes out of the data space.

To run the algorithms for high dimensional data set (n=799, p=180, c=6, Figure°4.1c), in a Personal Computer, with a windows 10, Intel(R) Core (TM) i5-3337U CPU @ 1.80GHz, 6Gb RAM, NVidia 630M (with 2GB dedicated memory), on 64 bit architecture (the results are the mean of 5 runs):

- AA: 1.80 seconds for 179 iterations;

- FCPM-2: 67.41 seconds (one minute and twelve seconds) for 33 major iterations (with 967 minor iterations);

- FCPM-0: 492.97 seconds (eighth minutes and twenty one seconds) for 83 major iterations (with 6666 minor iterations).

4.1.2 Outliers Influence on the Clustering Solutions

After studying the behaviour of the algorithms with synthetic data, it becomes interesting to repeat the study, in the same data, but augmented with outliers. Here, an outlier is defined as a point that does not belong to the original data generated from the FCPM-DG, and is far away from the center of data. The robustness of the algorithms to outliers is measured as their ability in retrieving the original prototypes (V_{Org}), *i.e.*, their capacity in finding cluster solutions as close as possible to the ones found in the previous study. This implies that, not only the outliers shouldn't be retrieved as ideal points, but also reconstructing them from the retrieved ideal clusters should not be possible.

The 82 data sets were augmented first one, then with two outliers. The generation of outliers followed the Interquartile Range (IQR) method (Han et al., 2017, p.554).

The first outlier was created in the following way: for each data set, each feature is computed by summing the mean of the data feature with the corresponding standard deviation multiplied by five. This way, it's guaranteed that the outlier is indeed an extreme point regarding of the considered data set. The second outlier is symmetric to the first one, with respect to the mean, in a sense that the multiplication of the standard deviation by five is subtracted to the mean of the data feature. Figure 4.3b contains an example of a data set with outliers.

The study was conducted with 5 distinct parameter settings, varying the number of outliers and the number of prototypes that the algorithms had to search. In the first two settings the algorithms had to search for the same number of prototypes from which the data was generated, $k = c_0$, first with one outlier, ($out = 1$), then with two outliers ($out = 2$). Then, they had to search for one more, $k = c_0 + 1$, again, with one outlier first, ($out = 1$), and then with two ($out = 2$). In the final setting, they had to search for two

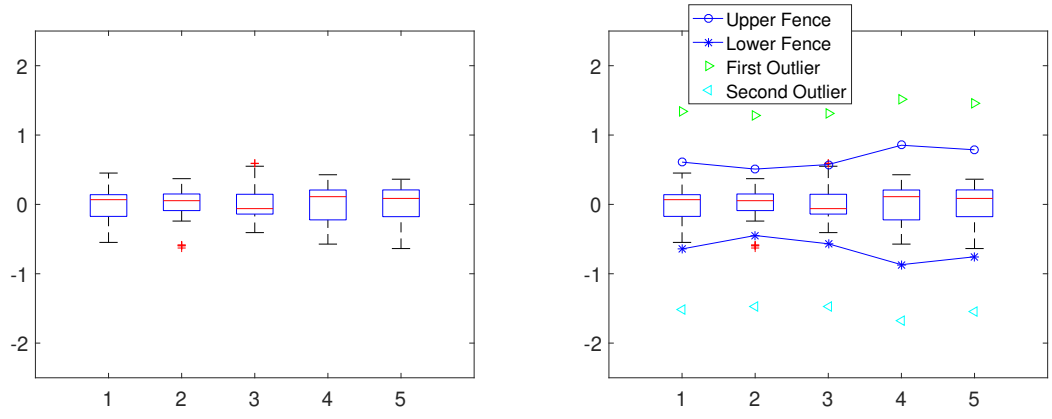


Figure 4.3: a) Boxplot for an artificial data set of small dimensionality (n=37, p=5, c=3). On the x-axis is the indices of every feature and in the y-axis the corresponding feature value. The red crosses indicate features that are above the 75^{th} percentile, or bellow 25^{th} percentile. b) This plot is the same as a), but with the upper and lower fences discriminated. The lower and upper fences are computed from the IQR method and correspond to $1.5 * IQR$, below the 25^{th} percentile and above the 75^{th} percentile, respectively. Two outliers computed from the described method are also displayed, in green the first outlier and in light blue, the second.

more, $k = c_0 + 2$, with the data augmented with two outliers ($out = 2$).

All the algorithms were initialized with the Furthest Sum method and run 5 times each. The results in Table 4.3 contains the Dissimilarity index for each setting and dimensionality. Table 4.4 contains the mode and median for the number of prototypes that the FCPM-0 shifts outside of the data space, for each different setting and dimensionality.

The results in Table 4.3 are also compared to the values in Table 4.1, that contain the data recovery values for the data sets without outliers.

Table 4.3: Average Dissimilarity D values for the outliers experiments, with the respective standard deviation (std).

dim	k=c_0						k=c_0+1						k=c_0+2		
	Out=1			Out=2			Out=1			Out=2			Out=2		
	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2
Small (Mean)	0,344	0,775	0,281	0,434	0,579	0,407	0,013	0,326	0,074	0,310	0,650	0,552	0,013	0,434	0,012
(std)	0,300	0,284	0,251	0,343	0,293	0,341	0,006	0,322	0,136	0,147	0,356	0,091	0,013	0,366	0,006
Medium (Mean)	0,519	0,604	0,461	0,339	0,511	0,610	0,009	0,186	0,010	0,159	0,475	0,580	0,008	0,180	0,006
(std)	0,300	0,254	0,194	0,345	0,282	0,254	0,004	0,173	0,006	0,085	0,332	0,093	0,003	0,166	0,005
High (Mean)	0,475	0,430	0,486	0,265	0,625	0,683	0,006	0,174	0,002	0,048	0,292	0,490	0,006	0,260	0,003
(std)	0,210	0,254	0,025	0,306	0,184	0,021	0,001	0,096	0,001	0,019	0,129	0,028	0,001	0,213	0,001

For the AA: When $k = c_0$, $out = 1$, $k = c_0$, $out = 2$ and $k = c_0 + 1$, $out = 2$ it usually puts the archetype(s) near the outlier(s) (Figures 4.4, A.4), or between an original and an outlier (Figures A.1, 4.5). Sometimes it also stops putting archetypes near the outliers and puts them near the originals (Figure A.3).

In $k = c_0 + 1$, $out = 1$ and $k = c_0 + 2$, $out = 2$, when there as many extra(s) archetype(s)

Table 4.4: Mode (round to unity) of the number of prototypes that the FCPM-0 shifts to outside of data space.

dim	k=c_0		k=c_0+1		k=c_0+2
	Out=1	Out=2	Out=1	Out=2	Out=2
Small (mode)	1	2	1	2	2
Medium (mode)	1	0	1	2	2
High (mode)	0	1	0	2	2

as the number the outliers, it always put the extra(s) archetype(s) in the outlier(s) (Figure A.6, A.7), resulting in D values similar to the ones in Table 4.1.

For the FCPM-2: In $k = c_0,\ out = 1$ and $k = c_0,\ out = 2$ the FCPM-2 always puts the prototype(s) near the outlier(s) (Figure 4.4). In $k = c_0,\ out = 2$, it always put at least one prototype near an outlier (Figure A.3).

In $k = c_0 + 1,\ out = 1$ and $k = c_0 + 2,\ out = 2$, when there as many extra(s) archetype(s) as the number the outliers, it always put the extra(s) prototypes(s) in the outlier(s) (Figures A.6). However, in some small dimensional data sets, it puts two of the prototypes near the same original (Figure A.5), resulting in a high D value.

For the FCPM-0: In small and medium dimensionalities, typically shifts outside of data space as many prototypes as the number outliers. For the $k = c_0,\ out = 1$, $k = c_0,\ out = 2$ and $k = c_0 + 1,\ out = 2$ settings the remaining prototypes are accommodated between the clusters (Figure 4.4).

In the $k = c_0 + 1,\ out = 1$ and $k = c_0 + 2,\ out = 2$ settings, the FCPM-0 finds the true number of clusters by shifting the prototypes.

For the high dimensional data sets, it's only in the $k = c_0 + 1,\ out = 2$ and $k = c_0 + 2,\ out = 2$ settings that it presents the behaviour of shifting as many prototypes as outliers. In the remaining settings, when there is only one outlier, it doesn't shift any prototype (Figure A.2). Finally, in the presence of two outliers, it only shifts one.

In summary, for the settings where the algorithms had to search for the same number of prototypes from which the data was generated (the 1^{st} and 2^{nd}, with $k = c_0$), both the AA and the FCPM-2 were easily influenced by the presence of outliers. From these 3 settings, it's possible to conclude that the AA is more robust in retrieving extreme ideal points in the presence of outliers (generated by the previously described method), than the FCPM-2.

For the 3^{rd} and 5^{th} settings, where the number of prototypes that the algorithms had to search was equal to the number of prototypes from which the data was generated plus the number of outliers, both the AA and the FCPM-2 algorithms had similar behaviours as in the data sets without outliers. These two settings showed that both algorithms are capable of safely retrieving extremes ideal points, as long as they have extra(s) archetype(s)/prototype(s) to put in the outlier(s).

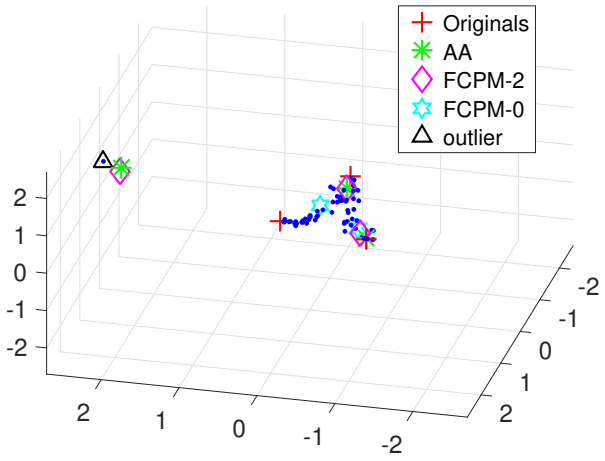


Figure 4.4: Principal components projection (R=99%) for the first setting ($k = c_0$, $out = 1$) of a data set of small dimensionality (n=62, p=5, c=3). It can be observed that both the AA and the FCPM-2 put one archetype/prototype near the outlier. One of the FCPM-0 prototypes is outside of data space and another in the middle of two clusters.

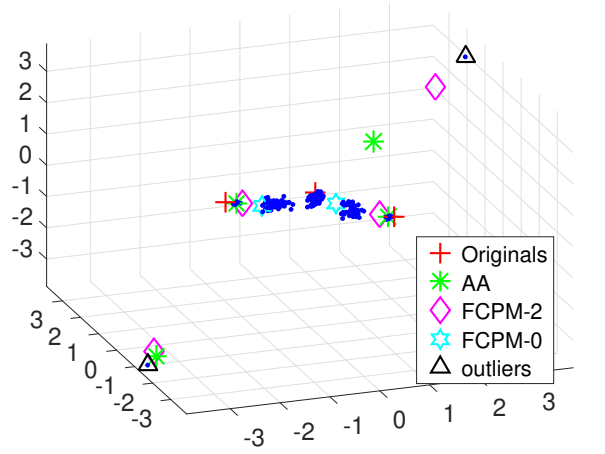


Figure 4.5: Principal components projection (R=99%), for the fourth setting ($k = c_0 + 1$, $out = 2$) of a medium dimensional data set (n=205, p=15, c=3). One of the archetypes is between an outlier and an original. Two FCPM-2 prototypes near both outliers.

The number of iterations necessary for the AA to converge, it's always higher in the presence of outliers, Table 4.5, than in the study without outliers (Table 4.2). Contrary to the previous results, the AA algorithm is now influenced by the increase of dimensionality. This result and the high number of iterations necessary for the AA to converge, indicates that this algorithm is sensible to outliers.

The FCPM-2 iterations are constantly low and in some settings, very close to the values of data without outliers.

The FCPM-0 shows an increase in the number of iterations, in the small and medium

Table 4.5: Average and standard deviations of iterations (round to the closest integer) by setting, for each algorithm, in each dimensionality. Here, only the major iterations of the FCPM are displayed.

dim	k=c_0						k=c_0+1						k=c_0+2		
	Out=1			Out=2			Out=1			Out=2			Out=2		
	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2	AA	fcpm0	fcpm2
Small (Mean)	381	97	18	318	80	28	294	89	13	324	94	19	486	100	18
(std)	136	10	21	168	32	26	126	20	4	125	13	26	41	0	3
Medium (Mean)	477	84	15	445	73	31	162	81	18	381	91	15	490	97	27
(std)	72	25	12	66	29	33	52	23	6	59	14	9	30	6	6
High (Mean)	500	75	27	454	96	20	192	90	33	388	93	33	487	94	37
(std)	0	20	1	43	8	3	20	11	0	53	7	0	19	9	0

dimensionalities. In high dimensional data sets, this algorithm doesn't reach the maximum number of iterations, as it does with the data without outliers.

In all settings, the FCPM-2 not only has a higher efficiency in the number of iterations than the AA, but also, the convergence process does not seem to be influenced by the outliers.

4.2 Comparative Study with Real Data

From the previous studies with synthetic data, it was possible to observe: how the AA algorithm is indeed capable of retrieving extreme prototypes, and how the algorithms behaves in the presence of data augmented with outliers. Using the knowledge acquired from these studies, the algorithms were again compared, but with real-world data. This means that the data generation process, or its clustering structure it's unknown. the only information known it's the labels, that were acquired through observation and expertise of researchers. The comparison of the algorithm is made using 3 different measures: (1) fuzzy internal validation indices; (2) Efficiency in convergence; (3) Data recovery proprieties. Visualization techniques to inspect the clustering results are also applied.

To create a diverse and powerful benchmark to test the algorithms, twelve well-known data sets from the UCI Machine Learning Repository (Lichman, 2013) were selected. It also used the mental disorders data set (Nascimento, 2005). As the mental disorders data set has an extreme tendency, it was counteracted by augmenting the data set with new, less severe cases, creating the Mental disorders augmented datda set (Nascimento, 2005).

Table 4.6 summarizes this benchmark data by number of entities (n), number of features (p), and number of distinct classes (c_0). To run the algorithms, the feature corresponding to the label was removed from all data sets. In all experiences, for each data set, the algorithms were run searching for $k = \{c_0 - 1, c_0, c_0 + 1\}$ archetypes/prototypes. Each algorithm was run 5 distinct times, from the same initial seeds, that were computed using the Furthest Sum method. All data sets were centered and normalized by range.

Table 4.6: Description of the data sets used.

Data set	Number of entities (n)	number of features (p)	number of distinct classes (c_0)
Bank note authentication	1372	4	2
Glass Identification	214	9	6
Indian Liver Patient	579	10	2
Iris	150	4	3/2
Mental Disorders	44	17	4
Mental Disorders augmented	80	17	4
Pima Indians Diabetes	768	8	2
Protein Localization Sites (E. Coli)	366	7	8
Seeds Kernel	210	7	3
Vehicle silhouettes	793	18	4
Wine recognition	178	13	3
Wisconsin Breast Cancer (WBC)	683	9	2
WBC Diagnostic	569	30	2
WBC Prognostic	198	32	2

4.2.1 Assessment of Clustering Solutions

The assessment of the quality of the found clustering partitions was done with the fuzzy internal validation indices described in Section 3.3.2: Partition Entropy (PE, ↓); Partition Coefficient (PC, ↑); Modified Partition Coefficient (MPC, ↑); Xie-Beni (XB, ↓); Fuzzy Silhouette Index (FSI, ↑), with the direction of the arrows (↓/↑) indicating the optimal value of the index. The values of PE index were normalized to be confined to the interval [0,1].

For each data set, Table 4.7 presents, the averages of the five internal validation indices, of the five runs. The best value of each index, for a given data set, across the three algorithms, is highlighted in shade. The second column contains the number of archetypes/prototypes searched by the algorithms. The number of true classes (c_0) is also highlighted in shade. The last row contains the proportion of the number of data sets where a given index is the best, for each algorithm.

When computing the FSI value for the FCPM-0, sometimes the toolbox returns NA. This is due to a division by zero, caused by the FCPM-0 shifting some of its prototypes to outside the data space.

The significant differences between the proportions of each algorithm for each index, in the last row of Table 4.7 indicate that the XB index is the more adequate for the AA algorithm, the FSI for the FCPM-2 and the PE, PC, MPC for the FCPM-0.

These results are concordant with the clustering criterion of each algorithm. The XB index uses the fuzzy cluster's center of each cluster as the representative for that cluster, when evaluating the inter-cluster separation. The FSI uses the average minimum pairwise distance between objects in each fuzzy cluster, as the separation measure. However, both the XB and FSI can be seen as adequate to evaluate algorithms that mine extreme ideal points, as in 11 data sets, for both AA and FCPM-2, these indices are concordant in the number of clusters.

As for the PE, PC, MPC, they only evaluate the membership values, valuing those close to 0 or 1, *i.e*, the less fuzzy the partition is, the better score they achieve.

Table 4.7: Validation indices values for the real-world data and their counts.

Data set	k	AA					FCPM-0					FCPM-2				
		PE(↓)	PC (↑)	MPC(↑)	FSI(↑)	XB(↓)	PE(↓)	PC(↑)	MPC(↑)	FSI(↑)	XB(↓)	PE(↓)	PC(↑)	MPC(↑)	FSI(↑)	XB(↓)
Bank Note	2	0.701	0.670	0.340	0.603	0.120	0.192	0.912	0.823	0.543	0.323	0.739	0.649	0.299	0.615	0.150
	3	0.678	0.535	0.302	0.620	0.105	0.197	0.869	0.803	NA	0.453	0.765	0.507	0.260	0.627	0.113
Glass	5	0.461	0.648	0.560	0.650	0.197	0.438	0.635	0.544	0.277	191	0.886	0.401	0.251	0.772	0.094
	6	0.422	0.628	0.553	0.647	0.193	0.324	0.727	0.672	NA	124	0.986	0.261	0.113	0.587	0.127
	7	0.426	0.566	0.494	0.607	0.304	0.353	0.659	0.602	NA	166	0.948	0.248	0.123	0.554	0.278
Indian Liver Patient	2	0.053	0.985	0.971	0.769	0.155	0.000	1.000	1.000	0.768	0.158	0.688	0.694	0.388	0.786	0.124
	3	0.468	0.654	0.481	0.619	0.139	0.264	0.820	0.730	NA	0.141	0.754	0.522	0.282	0.604	0.222
Iris	2	0.605	0.726	0.452	0.841	0.069	0.106	0.953	0.906	0.839	0.081	0.498	0.775	0.551	0.859	0.068
	3	0.594	0.587	0.380	0.741	0.198	0.228	0.838	0.757	NA	0.109	0.548	0.659	0.489	0.569	0.948
	4	0.589	0.508	0.344	0.422	0.476	0.242	0.782	0.710	0.520	324.639	0.610	0.554	0.406	0.340	1.186
Mental Disorders	3	0.351	0.773	0.660	0.576	0.244	0.020	0.986	0.979	0.408	0.602	0.790	0.495	0.243	0.586	0.199
	4	0.359	0.712	0.615	0.594	0.167	0.162	0.860	0.813	0.522	0.356	0.797	0.416	0.221	0.596	0.133
	5	0.355	0.774	0.661	0.577	0.243	0.023	0.984	0.976	0.444	0.824	0.788	0.497	0.246	0.588	0.199
Mental Disorders Augmented	3	0.530	0.633	0.449	0.496	0.177	0.137	0.910	0.865	0.442	0.472	0.818	0.473	0.209	0.521	0.204
	4	0.493	0.603	0.471	0.527	0.165	0.203	0.812	0.750	0.298	1.234	0.842	0.377	0.169	0.419	0.477
	5	0.481	0.562	0.452	0.499	0.226	0.108	0.888	0.860	0.353	0.798	0.859	0.314	0.142	0.492	0.272
Pima Indians Diabetes	2	0.642	0.708	0.416	0.508	0.175	0.186	0.915	0.829	0.456	0.554	0.843	0.600	0.199	0.505	0.268
	3	0.690	0.527	0.291	0.431	0.152	0.246	0.821	0.731	0.342	0.662	0.878	0.426	0.139	0.410	0.313
Protein Localization E. Coli	7	0,437	0,512	0,431	0,639	0,147	0,416	0,495	0,411	0,316	1,64E+27	0,813	0,293	0,175	0,560	0,682
	8	0,445	0,481	0,406	0,621	0,161	0,427	0,468	0,392	0,066	101120,9	0,812	0,257	0,151	0,480	42,822
	9	0,449	0,455	0,387	0,544	0,207	0,461	0,407	0,333	0,201	79,473	0,818	0,239	0,144	0,457	23,156
Seeds	2	0.636	0.711	0.421	0.794	0.075	0.178	0.918	0.835	0.766	0.115	0.576	0.736	0.473	0.794	0.083
	3	0.621	0.581	0.372	0.690	0.173	0.186	0.880	0.820	NA	0.211	0.642	0.595	0.393	0.683	0.246
	4	0.583	0.523	0.364	0.521	0.424	0.260	0.792	0.722	NA	112.158	0.677	0.501	0.334	0.368	1.793
Vehicle Silhouettes	3	0.633	0.574	0.361	0.584	0.194	0.173	0.889	0.833	0.464	0.643	0.774	0.503	0.255	0.588	0.250
	4	0.602	0.511	0.348	0.495	0.198	0.267	0.781	0.708	NA	1.009	0.821	0.394	0.192	0.563	0.204
	5	0.571	0.474	0.342	0.437	0.287	0.339	0.674	0.593	0.323	7.560	0.836	0.324	0.155	0.482	4.413
Wine Recognition	2	0.557	0.748	0.497	0.545	0.188	0.089	0.959	0.919	0.467	0.479	0.784	0.635	0.269	0.561	0.202
	3	0.511	0.655	0.483	0.552	0.166	0.184	0.875	0.813	0.541	0.418	0.816	0.476	0.214	0.583	0.175
	4	0.516	0.575	0.433	0.539	0.180	0.196	0.843	0.790	0.411	0.541	0.856	0.366	0.155	0.573	0.189
WisconsinBC	2	0.398	0.825	0.649	0.850	0.066	0.092	0.958	0.916	0.795	0.121	0.488	0.785	0.570	0.851	0.075
	3	0.356	0.766	0.650	0.795	0.122	0.021	0.985	0.978	0.602	8.213	0.571	0.644	0.466	0.861	0.945
Wisconsin BC Diagnostic	2	0.615	0.720	0.440	0.706	0.097	0.183	0.916	0.833	0.644	0.252	0.751	0.653	0.306	0.676	0.143
	3	0.597	0.604	0.406	0.685	0.112	0.129	0.908	0.862	0.523	0.656	0.811	0.477	0.216	0.616	0.208
Wisconsin BC Prognostic	2	0,695	0,675	0,350	0,456	0,181	0,040	0,981	0,963	0,291	1,075	0,880	0,577	0,154	0,427	0,288
	3	0,668	0,543	0,315	0,433	0,154	0,148	0,892	0,839	0,288	0,740	0,879	0,424	0,136	0,409	0,258
Count		0/14	0/14	0/14	5/14	10/14	14/14	14/14	14/14	0/14	0/14	0/14	0/14	0/14	10/14	4/14

Table 4.8 summarises the best number of clusters respecting the more adequate Validation Indice: Xie-Beni for AA, FSI for FCPM-2, PE, PC or MPC for FCPM-0. The last row contains the number of times that each algorithm suggested a solution concordant with the number of classes, c_0.

Table 4.8: Suggested number of partitions by the proposed indices, for each algorithm.

Data set	c_0	AA	FCPM-0	FCPM-2
Bank note authentication	2	3	2	3
Glass Identification	6	6	6	5
Indian Liver Patient	2	3	2	2
Iris	3/2	2	2	2
Mental Disorders	4	4	3	4
Mental Disorders augmented	4	4	3	3
Pima Indians Diabetes	2	3	2	2
Protein Localization Sites (E. Coli)	8	7	7	7
Seeds Kernel	3	2	2	2
Vehicle silhouettes	4	3	3	3
Wine recognition	3	3	2	3
Wisconsin Breast Cancer (WBC)	2	2	3	3
WBC Diagnostic	2	2	3	2
WBC Prognostic	2	3	2	2
Number of times $k = c_0$		7/14	6/14	7/14

No algorithm seems capable of systematically find solutions consistent with the number of labels in the data sets, with the suggested validation indices. However, these

numbers by themselves do not allow to conclude anything for two reasons: First, to systematically analyse an algorithm regarding the labels of the data, it's necessary a comprehensive study with external validation indices, this is not done here; Second, there is no imposition regarding the matching of the cluster structure with the labels of the data set. One comes from the "natural" organization of the data, and the other from human expertise.

Given the good fit of each index to the algorithms, there would be no advantage in exploring fusion strategies for the validation of the algorithms.

Table 4.9 contains the iterations that each algorithm needed to converge for each real-world data set. In shade, is a comparison between the AA and FCPM-2, to highlight the algorithm with less iterations. The results presented allow for a relation with the results in Table 4.2. For the FCPM algorithms, the number of iterations is directly related to goodness of results, and always low, whereas in the AA, there doesn't seem to exist a relation.

Table 4.9: Iterations of each algorithm for each data set.

Data set	k	AA	FCPM-0		FCPM-2	
		Iterations	Major	Minor	Major	Minor
Bank Note	2	406	100	5632	18	478
$c_0 = 2$	3	260	100	5516	92	1707
Glass	5	149	100	9498	34	1722
$c_0 = 6$	6	121	100	9509	48	1911
	7	81	100	10000	57	1485
Indian Liver Patient	2	10	4	41	11	428
$c_0 = 2$	3	500	100	704	14	701
Iris	2	182	11	704	9	166
$c_0 = 3/2$	3	202	100	4030	13	646
	4	127	100	10000	23	1622
Mental	3	125	15	1505	15	315
Disorders	4	51	100	8289	29	1343
$c_0 = 4$	5	65	100	8452	90	2534
Mental Disorders	3	140	68	4294	12	235
Augmented	4	62	84	8400	30	1405
$c_0 = 4$	5	61	79	7880	46	2303
Pima Indian Diabetes	2	356	26	1667	21	825
$c_0 = 2$	3	190	69	6900	46	1437
Protein Localization	7	221	100	10000	61	5835
E. Coli	8	76	100	10000	58	2983
$c_0 = 8$	9	127	100	10000	40	2849
Seeds	2	354	8	211	9	200
$c_0 = 3$	3	413	100	2534	19	1479
	4	222	100	8439	20	1562
Vehicle	3	368	84	4627	13	707
Silhouettes	4	500	100	8400	23	1189
$c_0 = 4$	5	235	100	1000	28	1825
Wine	2	291	28	2249	13	230
Recognition	3	79	68	2294	18	479
$c_0 = 3$	4	84	83	5370	18	705
WisconsinBC	2	97	12	120	9	298
$c_0 = 2$	3	462	6	580	21	1020
WBC Diagnostic	2	209	22	183	12	495
$c_0 = 2$	3	155	18	4320	22	915
WBC Prognostic	2	245	11	1060	26	537
$c_0 = 2$	3	185	100	10000	29	1042

4.2.2 Data Recovery Analysis on Real Data

To study the data recovery proprieties of the algorithms with real data, it's first necessary to select the "ideal" points, that will serve as the reference prototypes (V_{Org}) to measure the data recovery ability of the algorithms by dissimilarity index D (Eq. (3.5)).

To find such points, each data set is partitioned according to its labels, creating c_0 distinct groups. According to the type of prototypes generated by each algorithm (extreme points in case of AA and FCPM-2, and central points in case of FCPM-0), the c_0 reference prototypes defined for AA and FCPM-2 are those in each cluster of the ground partition that are furthest way from the grand mean of the data, creating artificial extreme points. For the FCPM-0, they are the centroid of each cluster of the ground partition.

Table 4.10: Data recovery of the real data, using the dissimilarity index D. Each result is the mean of 5 runs.

Data set	c_0	AA	FCPM-2	FCPM-0
Bank Note	2	0,369	0,425	0,463
Glass Data	6	0,066	0,001	0,522
Indian Liver Patient	2	0,745	0,773	0,841
Iris Data	3/2	0,369	0,293	0,731
Mental Disorders	4	0,144	0,147	0,187
Mental Disorders Augmented	4	0,249	0,222	0,224
Pima Indian Diabetes	2	0,946	0,977	0,475
Protein Localization E. Coli	8	0,177	0,474	0,550
Seeds	3	0,056	0,061	0,979
Vehicle Silhouettes	4	0,436	0,167	0,978
Wine Recognition	3	0,351	0,334	0,610
WisconsinBC	2	0,198	0,228	0,007
WBC Diagnostic	2	0,568	0,677	0,042
WBC Prognostic	2	0,885	0,963	0,552

Table 4.10 present the results of applying the dissimilarity index to the select referenced prototypes. It's easy to see as the FCPM-0 and FCPM-2 algorithms are better in the data recovery with real data than the AA algorithm.

In the Mental Disorders Augmented, it's also interesting to observe how, despite its extreme tendency being contradicted, the FCPM-0 and FCPM-2 find the best results.

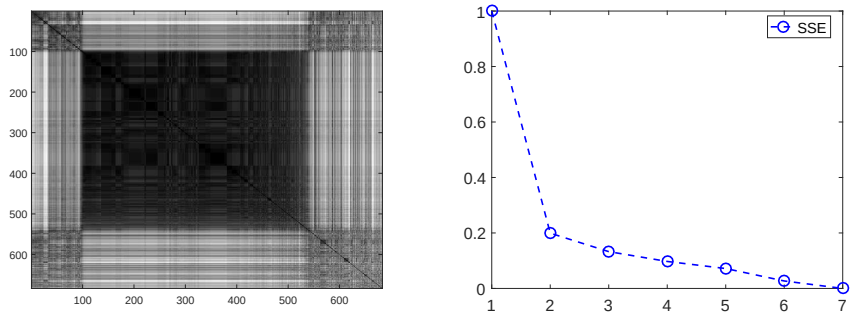
4.2.3 Visualization and Interpretation of Clustering Results

The visualization techniques presented in Section 3.3.3 are explored here to inspect the clustering results and visualize the AA clustering solutions of the 14 data sets. Due to the high number of data sets, only three are presented here, the Wisconsin Breast Cancer, the Mental Disorders and the Seeds Identification data set. The remaining can be visualized in Appendix B.

The Wisconsin Breast Cancer ($n = 683$, $p = 9$, $c_0 = 2$) aims at tumour classification and has two classes: benign and malign. From Table 4.8: the XB in AA points to 2 clusters;

the PE, PC, MPC in the FCPM-0 and the FSI in the FCPM-2, to 3 clusters.

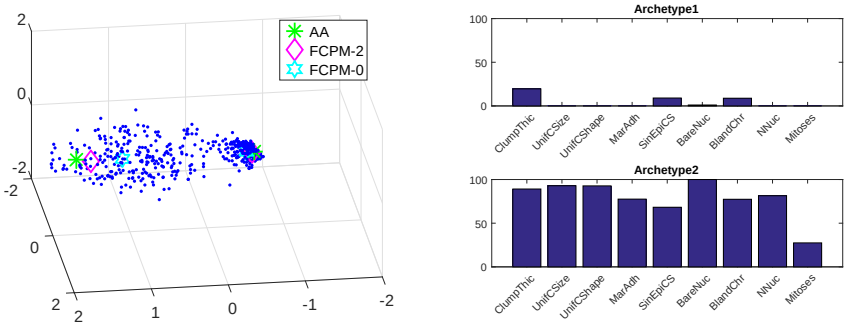
The SSE plot (Figure 4.6) is concordant with the suggested number of clusters for the AA, and the VAT with the suggested number for the FCPM-0 and FCPM-2.



a VAT plot. b SSE plot.

Figure 4.6: Plots for the Wisconsin Breast Cancer data set to inspect the number of clusters present. a) the VAT plot indicating the presence of one big cluster, and two smalls ones. b) the SSE plot indicating 2 clusters.

The archetypes found by the AA algorithm, for $k = 2$ represent clear profiles of the tumours, Figure 4.7b.

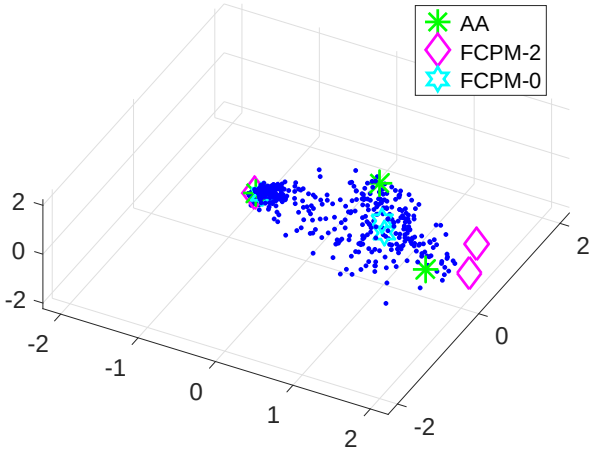


a PC projection. b Percentile plot.

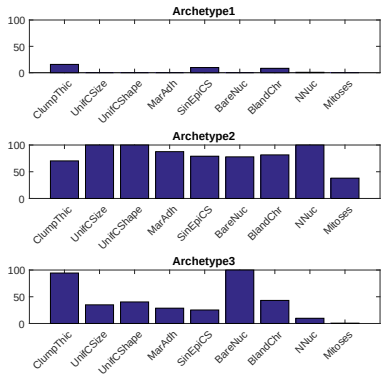
Figure 4.7: Plots for the Wisconsin Breast Cancer data set after the clustering process with k=2. On the left, a principal components projection (R=98%) with the archetypes/prototypes found. It shows how close the AA archetypes and the FCPM-2 prototypes are. On the right, the percentile plots for the 2 archetypes, with the first tumour as a benign tumour and the second as a malign one

For $k = 3$ the algorithm finds a new profile, however, its true meaning is unknown, Figure 4.8b. It's definitely a representation for a malign tumour, but only an expert could interpret its significance.

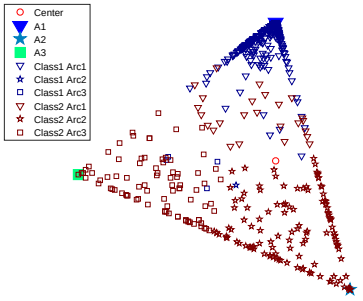
In this example, it's easy to understand how the proprieties of the AA can be of importance for medical diagnosticians. The archetypes are faithful representations of patients with, or without tumours. And each point, being combination of those two profiles, represented by its fuzzy membership, allows the physician to closely relate



a Principal components projection.



b Percentile plot.



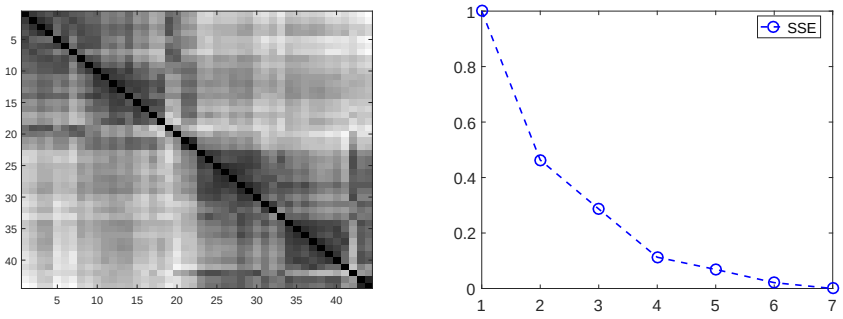
c Mixture plot.

Figure 4.8: Plots for the Wisconsin Breast Cancer data set after the clustering process with k=3. The top plot is a principal components projection (R=98%) with the archetypes/prototypes found, with two of the FCPM-2 prototypes close to each other. On bottom left, the percentile plots for the 3 archetypes. It contains a new profile, when compared with k=2, alongside the two profiles that were already discovered for $k = 2$. On the bottom right, the mixture plot with the archetypes distances preserved and the labels: Blue for benign and red for malign. The 2^{nd} and 3^{rd} successfully identifying almost all malign patients.

its patients to the profiles. More importantly, the set of features most discriminant in each archetype gives the possibility to extract valuable information, such as the stage of the tumour, its aggressiveness, or its classification, thus allowing the physician to craft detailed treatment plans for each patient.

The Mental Disorders data set ($n = 44$, $p = 17$, $c_0 = 4$), contains 44 patients, with 17 psychosomatic features ($h_1 - h_{17}$), evenly distributed in 4 distinct mental disorders: depressed (D), maniac (M), simple schizophrenic (S_s) and paranoid schizophrenic (S_p). Using the suggested indices, the AA and FCPM-2 algorithms points to 4 clusters, and the

FCPM-0 to 3. The SSE plot (Figure 4.9) is concordant with the AA and FCPM-2.



a VAT plot. b SSE plot.

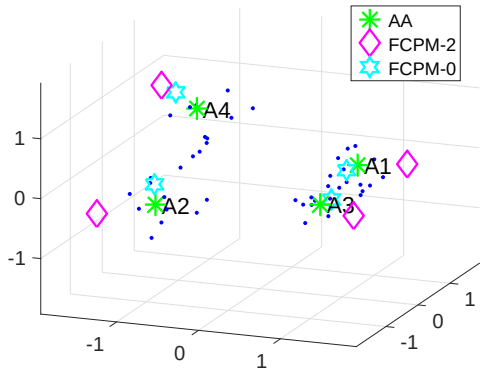
Figure 4.9: Plots for the Mental Disorders data set to inspect the number of clusters present. On the left, the VAT plot with the ODI very blurry. However it's still possible to identify two faint boxes. On the right, the SSE plot clearly indicating 4 clusters.

In Nascimento, 2005 is proven that this data set contains a cluster structure, where "each disease is characterised by 'archetypal patients' that show a pattern of extreme psychosomatic features values defining a *syndrome* of mental conditions (...)"(Nascimento, 2005, p.120). These "archetypal patients" and extreme features are now analysed in the context of AA (Figure 4.10). From the percentile plot it's possible to identify the features that define each subset: $D - \{h5, h9, h13\}; M - \{h8, h17\}; S_s - \{h3, h16\}; S_p - \{h8, h11, h15\}$. Even with the data set augmented with less severe cases (Figure B.9d), these subset of features are still present, although less pronounced.

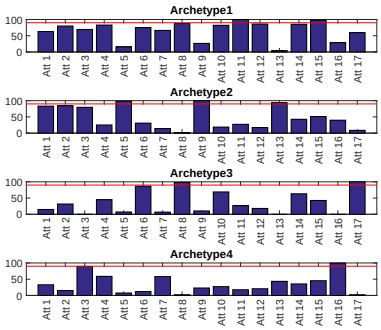
The Seeds Identification Kernel data set ($n = 366$, $p = 7$, $c_0 = 3$), contains kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian. All three algorithms indicate the presence of 2 clusters in the data. None of the algorithms are concordant with the suggested number by the VAT and SSE (Figure 4.11).

The archetypes for $k = 2$ are two distinct profiles for the seeds (Figure 4.12b).

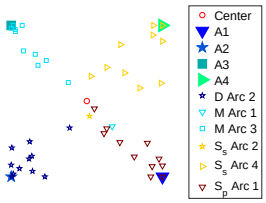
Although the XB index indicates 2 clusters, the solution with 3 archetypes also proves to be good (Figure 4.13c)



a Principal components projection.

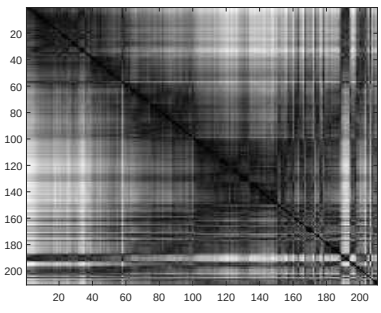


b Percentile plot.

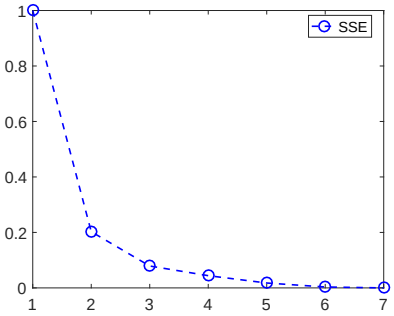


c Mixture plot.

Figure 4.10: Plots for the Mental Disorders data after the clustering process, with k=4. On the top, the principal components projection (R=98%) with the archetypes/prototypes found. On the bottom left, the percentile plot with a threshold on the 90^{th} percentile. On the bottom right, the mixture plot for the archetypes for k=4, indicating how the AA successfully identified the patients condition.

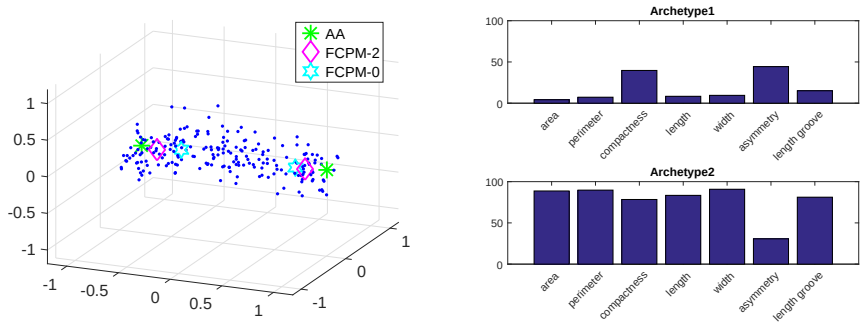


a VAT plot.



b SSE plot.

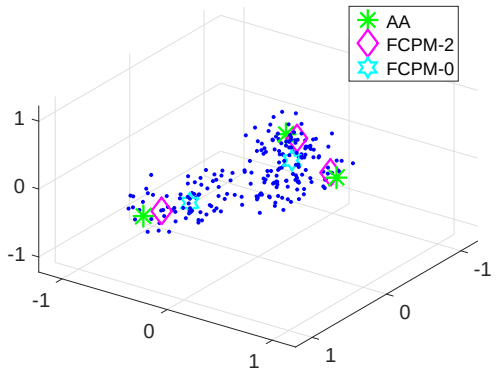
Figure 4.11: Plots for the Seeds Kernel data set to inspect the number of clusters present. On the left, the VAT, that is very blurry, not containing well-defined blocks. This suggests the lack of a cluster structure in the data, *i.e.*, the clusters are not well separated. On the right, the SSE suggesting 3 clusters



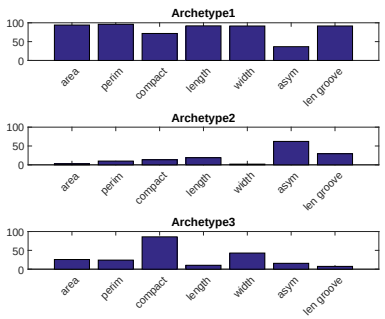
a Principal components projection.

b Percentile plot.

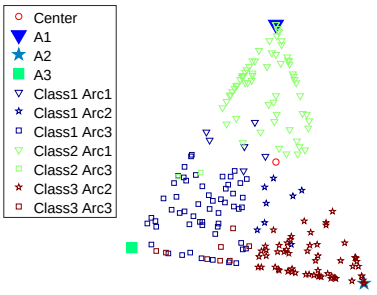
Figure 4.12: Plots for the Seeds Kernel data with k=2. On the left, a principal components projection (R=99%) with the archetypes/prototypes found for k=2. On the right, the percentile plot k=2. One of the archetypes has almost all the features in 90^{th} percentile and the other with almost all below the 20^{th}, showing that they are opposites. This is expected, as the archetypes must lie on the convex hull of the data.



a PC projection.



b Percentile plot3.



c Mixture plot.

Figure 4.13: Plots for the Seeds Kernel data with k=3. On the top, the PC projection (R=99%) with the archetypes/prototypes found for k=3. Here, one of the archetype shifts, to accommodate the new one. On the bottom left, the percentile plot for k=3. The first and second archetypes (Figure 4.13b) are very similar to the founds with $k = 2$, but with the second archetype being less pronounced in its features. On the bottom right, the mixture plot for k=3, with the archetypes distances preserved and the labels: Blue for Kama, green for Rosa, and red for Canadian. It shoows the good results of the AA for $k = 3$ in profiling the seeds and identifying the correct labels.

4.3 Comparing Initialization Strategies for AA and FCPM

In order to further explore the proprieties of an algorithm, it's necessary to study its behaviour with different initializations, in term of efficacy (values of indices) and efficiency (iterations necessary to converge). The methods use to initialize the algorithms are the ones described in 3.2.

The dissimilarity index D, is used to measure the distance of the found archetypes/prototypes between the solutions of two different initializations methods. For the glass data set, where $c_0 = 6$, and the IAP found 3 clusters, the algorithms had to be run again for $k = 3$.

The results of this experimental study are organized as follows:

1. One section for each algorithm. In each section, the two versions of the IAP and the IFP are compared, individually with the FS;

2. In each section, there are four tables. One for each type of initialization, to be compared with the FS, and a last one with a summary of all three;

3. Each of the first three tables contains the name of the data set, the dissimilarity index D, the number of iterations and the validation indices values for both initializations. For the IAP ($s \geq 0.05$) and IFP ($s \geq 0.05$), the second column contains the number of found clusters by the IAP. For the IAP ($k == c_0$), this number is the number of classes of the data set. The highlighted values correspond to the best values in the respective data set. In the last row, is the proportion of the best value for each measure. For the IAP ($s \geq 0.05$) and the IFP ($s \geq 0.05$), last row also contains the proportion of data sets where the k discovered by the IAP or IFP matches the number of labels of the data set, c_0. For each algorithm, the evaluation is done with the indices proposed on Section 4.2.1;

4. The fourth table contains the last row of each the previous 3 tables, with proportions of the best values.

4.3.1 Comparing initializations on AA

Table 4.11 contain the results for the comparison between the FS and IAP ($s \geq 0.05$). Except for the Mental Disorders Augmented, all the solutions are of equal efficacy. This is also proven by the D value, that is constantly null. The FS initialization has a higher efficiency, as in 8 of the data sets it had fewer iterations.

The analysis of Table 4.12 (FS *vs* IAP ($k == c_0$)), is similar to the previous setting. Both initializations have equal efficacy, again, proven by the D measure. The only exceptions being on the Glass identification and Protein location data sets, where the FS achieves significant lower values, and the distance (D) between the solutions is at its maximum. The initialization by FS also has a higher efficiency, as in 9 of the data sets it had fewer

Table 4.11: Comparing the FS with IAP ($s \geq 0.05$) in the AA algorithm. The k represents the number of suggested clusters by the IAP.

Data Set	IAP k	D	FS		IAP ($s \geq 0.05$)	
			Iterations	XB (↓)	Iterations	XB (↓)
Bank ($c_0 = 2$)	3	0	260	0.105	183	0.105
Glass ($c_0 = 6$)	3	0	324	0.118	439	0.118
Indian Liver ($c_0 = 2$)	3	0	500	0.139	500	0.138
Iris ($c_0 = 3/2$)	2	0	182	0.069	438	0.069
Mental ($c_0 = 4$)	4	0	51	0.167	68	0.166
Mental Aug ($c_0 = 4$)	3	0.182	140	0.177	110	0.192
Pima indian ($c_0 = 2$)	2	0	356	0.175	497	0.178
Protein location ($c_0 = 8$)	4	0	119	0.125	181	0.126
Seeds ($c_0 = 3$)	2	0	354	0.075	252	0.075
Vehicle ($c_0 = 4$)	3	0	140	0.194	341	0.194
Wine ($c_0 = 3$)	3	0	79	0.166	84	0.166
WisconsinBC ($c_0 = 2$)	2	0	97	0.066	99	0.066
WBC Diag ($c_0 = 2$)	2	0	209	0.097	205	0.097
WBC Prog ($c_0 = 2$)	3	0	185	0.154	160	0.154
Count	6		8	4	4	1

iterations.

Table 4.12: Comparing the FS with IAP ($k == c_0$) in the AA algorithm.

Data Set	k	D	FS		IAP ($k == c_0$)	
			Iterations	XB (↓)	Iterations	XB (↓)
Bank ($c_0 = 2$)	2	0	406	0.120	441	0.120
Glass ($c_0 = 6$)	6	1	121	0.193	166	0.310
Indian Liver ($c_0 = 2$)	2	0	10	0.155	18	0.155
Iris ($c_0 = 3/2$)	3	0	202	0.198	133	0.198
Mental ($c_0 = 4$)	4	0	51	0.167	68	0.166
Mental Aug ($c_0 = 4$)	4	0	62	0.165	62	0.165
Pima indian ($c_0 = 2$)	2	0	356	0.175	497	0.178
Protein location ($c_0 = 8$)	8	1	76	0.161	243	0.213
Seeds ($c_0 = 3$)	3	0	413	0.173	195	0.173
Vehicle ($c_0 = 4$)	4	0	62	0.198	500	0.198
Wine ($c_0 = 3$)	3	0	79	0.166	88	0.166
WisconsinBC ($c_0 = 2$)	2	0	97	0.066	83	0.066
WBC Diag ($c_0 = 2$)	2	0	209	0.097	205	0.097
WBC Prog ($c_0 = 2$)	2	0	245	0.181	303	0.181
Count			9	3	5	1

For the FS *vs* $IFP(s \geq 0.05)$ (Table 4.13), the results are different from the preceding settings. The efficacy is very similar, with the FS performing slightly better on three data sets. However, in two of those data sets, the difference is of the order of 0,001, and therefore is non-significant. In this setting, both initializations have equal efficiency, with each initialization being better than the other one in 7 data sets.

Table 4.13: Comparing the FS with $IFP(s \geq 0.05)$ in the AA algorithm. The k represents the number of suggested clusters by the IFP.

Data Set	IFP k	D	FS		IFP $(s \geq 0.05)$	
			Iterations	XB (↓)	Iterations	XB (↓)
Bank ($c_0 = 2$)	3	0	260	0.105	284	0.105
Glass ($c_0 = 6$)	3	0	324	0.118	397	0.118
Indian Liver ($c_0 = 2$)	3	0	500	0.139	479	0.139
Iris ($c_0 = 3/2$)	2	0	182	0.069	183	0.069
Mental ($c_0 = 4$)	4	0	51	0.167	45	0.167
Mental Aug ($c_0 = 4$)	3	0.182	140	0.177	106	0.192
Pima indian ($c_0 = 2$)	2	0	356	0.175	331	0.175
Protein location ($c_0 = 8$)	4	0	119	0.125	133	0.126
Seeds ($c_0 = 3$)	2	0	354	0.075	265	0.075
Vehicle ($c_0 = 4$)	3	0	140	0.194	399	0.195
Wine ($c_0 = 3$)	3	0	79	0.166	89	0.166
WisconsinBC ($c_0 = 2$)	2	0	97	0.066	84	0.066
WBC Diag ($c_0 = 2$)	2	0	209	0.097	203	0.097
WBC Prog ($c_0 = 2$)	3	0	185	0.154	203	0.154
Count	6		7	3	7	0

Table 4.14 contains the summary of the proportions of the best values, for each measure. In most settings, the difference between the solutions is minimal, with most of the times having $D = 0$. When there is a difference in the solutions, the FS initialization leads to better XB values. Regarding the efficiency, in the first two settings, where the IAP returns the seeds as averages of the clusters, the FS always has fewer iterations. However, in the last setting, where the seeds are extremes, there is a tie between them. This is a strong suggestion on the advantage of using points located on the boundary of the data as seeds to improve the efficiency, without influencing the efficacy, for this algorithm. Another evidence is how the IFP $(s \geq 0.05)$ (Table 4.14), in 7 of the data sets has less iterations, contrasting to the 4 data sets in IAP $(s \geq 0.05)$ (Table 4.14). These results are expected given the location of the archetypes in the convex hull.

Table 4.14: Summary of the counts from the previous tables for the AA.

FS		IAP		
Iterations	XB (↓)	Iterations	XB (↓)	IAP type
8	1	4	0	IAP $(s \geq 0.05)$
9	3	5	1	AP $(k == c_0)$
7	3	7	0	IFP $(s \geq 0.05)$

4.3.2 Comparing initializations on FCPM-2

For the FCPM-2 initialized with the AP $(s \geq 0.05)$ (Table 4.15), the solutions are also very similar, with the mental disorders augmented being the exception. For the efficacy, there isn't a significant difference between the solutions. In the 6 of the data sets with different FSI values, that difference its only by 0.001 in 5 of those cases. Regarding the efficiency,

the IAP is better, as in 10 of the data sets it has fewer major iterations, and in 11, fewer minor.

Table 4.15: Comparing the FS with IAP ($s \geq 0.05$) in the FCPM-2 algorithm. The k represents the number of suggested clusters by the IAP.

Data Set	IAP k	D	FS			IAP ($s \geq 0.05$)		
			Major	Minor	FSI (↑)	Major	Minor	FSI (↑)
Bank ($c_0 = 2$)	3	0.001	92	1707	0.627	92	2462	0.626
Glass ($c_0 = 6$)	3	0	19	855	0.589	13	518	0.588
Indian Liver ($c_0 = 2$)	3	0	14	701	0.604	14	533	0.604
Iris ($c_0 = 3/2$)	2	0	9	166	0.859	7	156	0.859
Mental ($c_0 = 4$)	4	0	29	1343	0.596	16	464	0.596
Mental Aug ($c_0 = 4$)	3	1	12	235	0.521	34	701	0.510
Pima indian ($c_0 = 2$)	2	0	21	825	0.505	11	346	0.505
Protein location ($c_0 = 8$)	4	0	26	1401	0.699	17	542	0.699
Seeds ($c_0 = 3$)	2	0	9	200	0.794	7	170	0.794
Vehicle ($c_0 = 4$)	3	0	12	235	0.588	14	856	0.589
Wine ($c_0 = 3$)	3	0	18	479	0.583	14	**354**	0.583
WisconsinBC ($c_0 = 2$)	2	0	9	298	0.851	8	234	0.850
WBC Diag ($c_0 = 2$)	2	0	12	495	0.676	10	294	0.675
WBC Prog ($c_0 = 2$)	3	0	29	1042	0.409	15	497	0.408
Count	6		2	3	6	10	11	1

For the FS *vs* IAP ($k == c_0$) (Table 4.16), the later has better results both in efficacy and efficiency. With 6 data sets having better FSI values, and 10 with less major and minor iterations. Also, in this setting there is more divergence between the solutions, with 4 data sets having high D values, where the IAP ($k == c_0$) constantly obtains better FSI values, with a significant difference from the FS.

As in the IAP ($s \geq 0.0$)5 (Table 4.15), the solutions between the IFP($s \geq 0.05$) (Table 4.17) and the FS are very similar, with the Augmented Mental Disorders data set having again, $D \neq 0$. Almost all data sets have the same values for the FSI in both initializations. In two of the data sets, the FS achieves higher values by a significant margin. It's also noteworthy, in the Glass data set, a peculiar behaviour of different FSI values, but $D = 0$.

The IAP initialization is also more efficient, with 7 and 8 data sets having fewer iterations, for the major and minor, respectively.

As shown in Table 4.18 all 3 IAP versions had a higher efficiency, by always having less iterations. As for the efficacy, the FS was better than IAP ($s \geq 0.05$) and IFP($s \geq 0.05$), despite their solutions being very similar. However, the FCPM-2 had better FSI values when initialized with the IAP ($k == c_0$), than the FS. This suggests that, for the FCPM-2, the IAP is more suitable when the algorithm must find as many clusters as the labels of

Table 4.16: Comparing the FS with IAP ($k == c_0$) in the FCPM-2 algorithm.

Data Set	k	D	FS			IAP ($k == c_0$)		
			Major	Minor	FSI (↑)	Major	Minor	FSI (↑)
Bank ($c_0 = 2$)	2	0	18	478	0.615	12	382	0.615
Glass ($c_0 = 6$)	6	1	48	1911	0.587	65	3450	0.654
Indian Liver ($c_0 = 2$)	2	0	11	428	0.786	10	251	0.786
Iris ($c_0 = 3/2$)	3	1	13	646	0.569	45	4089	0.759
Mental ($c_0 = 4$)	4	0	29	1343	0.596	16	464	0.596
Mental Aug ($c_0 = 4$)	4	0.159	30	1405	0.419	27	939	0.529
Pima indian ($c_0 = 2$)	2	0	21	825	0.505	11	346	0.505
Protein location ($c_0 = 8$)	8	1	58	2983	0.480	42	3035	0.530
Seeds ($c_0 = 3$)	3	0.001	19	1479	0.683	27	2001	0.689
Vehicle ($c_0 = 4$)	4	0.002	30	1405	0.563	46	998	0.580
Wine ($c_0 = 3$)	3	0	18	479	0.583	13	465	0.583
WisconsinBC ($c_0 = 2$)	2	0	9	298	0.851	10	325	0.850
WBC Diag ($c_0 = 2$)	2	0	12	495	0.676	10	294	0.675
WBC Prog ($c_0 = 2$)	2	0	26	538	0.427	20	400	0.427
Count			5	5	1	9	9	6

Table 4.17: Comparing the FS with $IFP(s \geq 0.05)$ in the FCPM-2 algorithm. The k represents the number of suggested clusters by the IFP.

Data Set	IFP k	D	FS			IFP ($s \geq 0.05$)		
			Major	Minor	FSI (↑)	Major	Minor	FSI (↑)
Bank ($c_0 = 2$)	3	0	92	1707	0.627	87	1571	0.627
Glass ($c_0 = 6$)	3	0	19	855	0.589	16	1073	0.306
Indian Liver ($c_0 = 2$)	3	0	14	701	0.604	20	878	0.604
Iris ($c_0 = 3/2$)	2	0	9	166	0.859	9	165	0.859
Mental ($c_0 = 4$)	4	0	29	1343	0.596	18	450	0.596
Mental Aug ($c_0 = 4$)	3	1	12	235	0.521	34	670	0.510
Pima indian ($c_0 = 2$)	2	0	21	825	0.505	24	1181	0.505
Protein location ($c_0 = 8$)	4	0	26	1401	0.699	21	781	0.699
Seeds ($c_0 = 3$)	2	0	9	200	0.794	7	173	0.794
Vehicle ($c_0 = 4$)	3	0	12	235	0.588	17	946	0.589
Wine ($c_0 = 3$)	3	0	18	479	0.583	11	345	0.583
WisconsinBC ($c_0 = 2$)	2	0	9	298	0.851	6	291	0.851
WBC Diag ($c_0 = 2$)	2	0	12	495	0.676	13	529	0.675
WBC Prog ($c_0 = 2$)	3	0	29	1042	0.409	21	685	0.409
Count	6		5	6	3	7	8	1

the data set.

4.3.3 Comparing initializations on FCPM-0

In the FCPM-0, for the FS *vs* IAP ($s \geq 0.05$) (Table 4.19), the FS has a higher efficacy, as in 6 data sets for the PE, and 5 for the PC and MP, it has better values. For the efficiency, the IAP has fewer iterations, both in major and minor, in 7 and 12 data sets respectively.

Table 4.18: Summary of the counts from the previous tables for the FCPM-2.

FS			IAP			
Major	Minor	FSI (↑)	Major	Minor	FSI (↑)	IAP type
2	3	6	10	11	1	IAP ($s \geq 0.05$)
5	5	1	9	9	6	IAP ($k == c_0$)
5	6	3	7	8	1	IFP ($s \geq 0.05$)

In this setting, contrary to the previous experiments, several solutions are different, with $D \neq 0$ in 9 data sets.

Table 4.19: Comparing the FS with IAP ($s \geq 0.05$) in the FCPM-0 algorithm. The k represents the number of suggested clusters by the IAP.

Data Set	IAP k	D	FS					IAP ($s \geq 0.05$)				
			Major	Minor	PE (↓)	PC (↑)	MPC (↑)	Major	Minor	PE (↓)	PC (↑)	MPC (↑)
Bank ($c_0 = 2$)	3	0.051	100	5516	0.197	0.869	0.803	100	3458	0.185	0.877	0.815
Glass ($c_0 = 6$)	3	0.601	51	8209	0.159	0.893	0.839	14	1360	0.172	0.872	0.808
Indian Liver ($c_0 = 2$)	3	0.004	100	704	0.264	0.820	0.730	100	523	0.280	0.806	0.709
Iris ($c_0 = 3/2$)	2	0	11	704	0.106	0.953	0.906	7	303	0.106	0.953	0.906
Mental ($c_0 = 4$)	4	0.214	100	8289	0.162	0.860	0.813	100	10000	0.141	0.873	0.831
Mental Aug ($c_0 = 4$)	3	0.480	68	4294	0.137	0.910	0.865	100	1367	0.139	0.914	0.870
Pima indian ($c_0 = 2$)	2	0	26	1667	0.186	0.915	0.829	10	134	0.186	0.915	0.829
Protein location ($c_0 = 8$)	4	1	56	5640	0.181	0.840	0.787	100	10000	0.209	0.828	0.771
Seeds ($c_0 = 3$)	2	0	8	211	0.178	0.918	0.835	8	200	0.178	0.918	0.835
Vehicle ($c_0 = 4$)	3	0.635	68	4294	0.173	0.889	0.833	100	2379	0.157	0.904	0.855
Wine ($c_0 = 3$)	3	0.600	68	2294	0.184	0.875	0.813	18	1760	0.196	0.859	0.788
WisconsinBC ($c_0 = 2$)	2	0	12	120	0.092	0.958	0.916	8	87	0.092	0.958	0.916
WBC Diag ($c_0 = 2$)	2	0	22	183	0.183	0.916	0.833	6	70	0.183	0.916	0.833
WBC Prog ($c_0 = 2$)	3	0.057	100	10000	0.148	0.892	0.839	48	4560	0.213	0.845	0.768
Count	6		3	2	6	5	5	7	12	3	4	4

When the FCPM-0 is initialized with IAP ($k == c_0$) (Table 4.20), it has a higher efficacy, in 6 data sets. For the major iterations, the FS has 4 data sets with fewer iterations, against the 3 of the IAP. Regarding the minor iterations, it's the opposite situation, with the IAP having 7 data sets with less iterations, and the FS only 6. Again, most of the solutions are different, with 10 data sets having $D \neq 0$.

The solutions of the $IAP\_M\ S \geq 0.05$ (Table 4.21) initializations are usually better. In efficacy, for more than half of data sets it obtains better values for the validation indices. Regarding the efficiency, the IAP obtains fewer major iterations in 5 data sets, and in 7 data sets, fewer minor. Again, almost all data sets have different solutions.

For the FCPM-0, as in the FCPM-2, the IAP ($k == c_0$) is the more suitable choice when there is the need to find as many clusters as the labels of the data set (Table 4.22). Otherwise, the $IFP(s \geq 0.05)$ is a better choice. Not only has a higher efficacy and efficiency than the FS, but also has a higher efficacy then its original version, IAP ($s \geq 0.05$). Given the tendency of the FCPM-0 in finding more central prototypes, these results were unexpected.

Table 4.20: Comparing the FS with IAP ($k == c_0$) in the FCPM-0 algorithm.

Data Set	k	D	FS					IAP ($k == c_0$)				
			Major	Minor	PE (↓)	PC (↑)	MPC (↑)	Major	Minor	PE (↓)	PC (↑)	MPC (↑)
Bank ($c_0 = 2$)	2	0.003	100	5632	0.192	0.912	0.823	100	5786	0.246	0.886	0.772
Glass ($c_0 = 6$)	6	1	100	9509	0.324	0.727	0.672	100	9893	0.280	0.730	0.676
Indian Liver ($c_0 = 2$)	2	0	4	41	0.000	1.000	1.000	4	121	0.000	1.000	1.000
Iris ($c_0 = 3/2$)	3	1	100	4030	0.228	0.838	0.757	100	1697	0.139	0.917	0.875
Mental ($c_0 = 4$)	4	0.214	100	8289	0.162	0.860	0.813	100	10000	0.141	0.873	0.831
Mental Aug ($c_0 = 4$)	4	0.653	84	8400	0.203	0.812	0.750	100	5755	0.185	0.848	0.797
Pima indian ($c_0 = 2$)	2	0	26	1667	0.186	0.915	0.829	10	134	0.186	0.915	0.829
Protein location ($c_0 = 8$)	8	1	100	10000	0.427	0.468	0.392	100	10000	0.454	0.451	0.372
Seeds ($c_0 = 3$)	3	0.006	100	2534	0.186	0.880	0.820	100	1826	0.182	0.883	0.824
Vehicle ($c_0 = 4$)	4	1	84	8400	0.267	0.781	0.708	100	10000	0.355	0.676	0.568
Wine ($c_0 = 3$)	3	0.626	68	2294	0.184	0.875	0.813	73	7340	0.154	0.891	0.836
WisconsinBC ($c_0 = 2$)	2	0	12	120	0.092	0.958	0.916	11	87	0.092	0.958	0.916
WBC Diag ($c_0 = 2$)	2	0	22	183	0.183	0.916	0.833	6	70	0.183	0.916	0.833
WBC Prog ($c_0 = 2$)	2	0.304	11	1060	0.040	0.981	0.963	21	396	0.251	0.883	0.766
Count			4	6	4	4	4	3	7	6	6	6

Table 4.21: Comparing the FS with IFP ($s \geq 0.05$) in the FCPM-0 algorithm. The k represents the number of suggested clusters by the IFP.

Data Set	IFP k	D	FS					IFP ($s \geq 0.05$)				
			Major	Minor	PE (↓)	PC (↑)	MPC (↑)	Major	Minor	PE (↓)	PC (↑)	MPC (↑)
Bank ($c_0 = 2$)	3	0.228	100	5516	0.197	0.869	0.803	100	5857	0.203	0.864	0.796
Glass ($c_0 = 6$)	3	1	51	8209	0.159	0.893	0.839	19	1883	0.122	0.910	0.864
Indian Liver ($c_0 = 2$)	3	0.005	100	704	0.264	0.820	0.730	100	865	0.248	0.834	0.751
Iris ($c_0 = 3/2$)	2	0.015	11	704	0.106	0.953	0.906	9	564	0.102	0.954	0.909
Mental ($c_0 = 4$)	4	0.608	100	8289	0.162	0.860	0.813	100	8241	0.141	0.873	0.830
Mental Aug ($c_0 = 4$)	3	0.477	68	4294	0.137	0.910	0.865	100	1679	0.140	0.913	0.870
Pima indian ($c_0 = 2$)	2	0	26	1667	0.186	0.915	0.829	17	839	0.186	0.915	0.829
Protein location ($c_0 = 8$)	4	1	56	5640	0.181	0.840	0.787	100	10000	0.211	0.827	0.770
Seeds ($c_0 = 3$)	2	0.003	8	211	0.178	0.918	0.835	12	577	0.165	0.924	0.848
Vehicle ($c_0 = 4$)	3	0.640	68	4294	0.173	0.889	0.833	100	2011	0.155	0.905	0.857
Wine ($c_0 = 3$)	3	0.600	68	2294	0.184	0.875	0.813	35	3580	0.156	0.887	0.830
WisconsinBC ($c_0 = 2$)	2	0	12	120	0.092	0.958	0.916	12	129	0.092	0.958	0.916
WBC Diag ($c_0 = 2$)	2	0	22	183	0.183	0.916	0.833	13	159	0.183	0.916	0.833
WBC Prog ($c_0 = 2$)	3	0.037	100	10000	0.148	0.892	0.839	100	10000	0.188	0.864	0.796
Count	6		4	6	4	3	3	5	7	7	8	8

Table 4.22: Summary of the counts from the previous tables for the FCPM-0.

FS					IAP					
Major	Minor	PE (↓)	PC (↑)	MPC (↑)	Major	Minor	PE (↓)	PC (↑)	MPC (↑)	IAP type
3	2	6	5	5	7	12	3	4	4	IAP ($s \geq 0.05$)
4	6	4	4	4	3	7	6	6	6	IAP ($k == c_0$)
4	6	4	3	3	5	7	7	8	8	IFP ($s \geq 0.05$)

4.3.4 Summary

It's now clear how the behaviour of the algorithms varies according to different initializations. For the AA, it was seen a significant benefit in initializing the algorithm using extreme points as seeds. This was somewhat expected due to the intrinsic definition of an archetype. Regarding the FCPM-m, it was observed that different objectives benefits from distinct initializations, *i.e.*, if the desire is to match the number of clusters with the the number of labels, is better to use the IAP solution. It's also possible to conclude that the FCPM-0 also benefits from extreme points as seeds.

CHAPTER 5

Conclusions and Future Work

This work is an attempt to experimentally analyse a version of Archetypal Analysis, FS-AA, in the framework of fuzzy clustering, considering the Fuzzy Proportional Membership and the following aspects:

(i) Analysis of Cluster Structure Recovery - this study is conducted assuming no probabilistic distribution on the data sets. The diverse collection of synthetic data sets taken from the FCPM Data Generator covering small, intermediate, and high dimensional data on data recovery is, to the best of our knowledge, the first study with a proper data generator showing how AA is able to reconstruct the archetypes working well with high dimensional data. This also shows how FS-AA is compatible with FCPM-2 model. The dissimilarity data recovery index D proved to be effective also for AA.

(ii) Assessing the quality of AA partitions - the popular fuzzy validation indices selected in this study contribute to archetypal analysis to substitute the traditionally used elbow method that does not work well in real world data with a non proper clear-cut cluster structure. Specifically, the Xie-Beni index that just considers the clusters prototypes for evaluating inter-cluster separations shown to be appropriate for FS-AA when using the archetypes. The FSI, and the collection PC, PE, MPC are more appropriate for FCPM family because these algorithms provide more clear-cut fuzzy partitions. However, it must be pointed out that, as is well documented in the literature, there is no best validation index for all real data sets. One must select suitable indexes for different kinds of data sets.

(iii) Robustness to the presence of outliers – the conducted study was the first one conducted on FS-AA as well as for the FCPM algorithms. Despite being a study that requires extension, it had shown how AA and FCPM-2 are quite robust to the presence of outliers in finding a good partition, and emphasise the singular behaviour of FCPM-0 of non-convergence and exploding as many prototypes as the number of archetypes present

in data.

(iv) Visualizing data Clustering Tendency – The visualization functionalities provided respond to different proposes. The VAT visualization allows to analyse the intrinsic cluster tendency (or not) of data sets. The projections on the space of PCA's as well as measure R to measure the variance captured by the PCA projection [Nascimento, 2005] very much fit the FCPM model. Mainly, the percentile and mixture plots are very much appealing to have a picture of the archetypes respecting their most discriminating features.

(v) Initialization Strategies - on comparing the furthest-sum (FS) algorithm against the Iterative Anomalous Pattern (IAP) initializations strategies one sees that FS is the most appropriate for AA. This seems natural since the seeds generated by this process are guaranteed to lie in the minimal convex set of the data points [REF to Neuro-Computing 2011]. The FCPM algorithms, on the contrary, benefit from the IAP initialization, not only by improving, in general, the rate of convergence of the algorithms, as well as allowing the user to fine-tune the level of resolution to look at data.

As future work we propose:

- To improve the validation protocol for AA and FCPM algorithms not only by using other family of popular unsupervised validation indices as well as using the data not with the whole set of features but only taking those features that better characterize each archetype / FCPM ideal point.

- To extend the study on synthetic data with fuzzy internal validation indices.

- To better study stop condition of IAP algorithm.

- To apply the algorithms to a real world application where the concept of archetypes be useful for clustering analysis.

Bibliography

Adä, I. and M. R. Berthold (2010). "The new iris data." In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD*. ACM Press. DOI: 10.1145/1835804.1835858.

Ahmad, A. (2016). "Evaluation of Modified Categorical Data Fuzzy Clustering Algorithm on the Wisconsin Breast Cancer Dataset." In: *Scientifica* 2016, pp. 1–6. DOI: 10.1155/2016/4273813.

Albuquerque, G., T. Lowe, and M. Magnor (2011). "Synthetic Generation of High-Dimensional Datasets." In: *IEEE Transactions on Visualization and Computer Graphics* 17.12, pp. 2317–2324. DOI: 10.1109/tvcg.2011.237.

Alia, O. M. (2014). "A Decentralized Fuzzy C-Means-Based Energy-Efficient Routing Protocol for Wireless Sensor Networks." In: *The Scientific World Journal* 2014, pp. 1–9. DOI: 10.1155/2014/647281.

Ansari, Z., M. F. Azeem, A. V. Babu, and W. Ahmed (Sept. 1, 2015). "A Fuzzy Clustering Based Approach for Mining Usage Profiles from Web Log Data." In: *International Journal of Computer Science and Information Security, pp. 70-79 Vol. 9, No. 6, June 2011. (ISSN 1947-5500, IJCSIS Publications, United State)* 9.6, pp. 70–79. arXiv: 1509.00693v1 [cs.DB].

Arbelaitz, O., I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona (2013). "An extensive comparative study of cluster validity indices." In: *Pattern Recognition* 46.1, pp. 243–256. DOI: 10.1016/j.patcog.2012.07.021.

Archetype (Jan. 2018). Merriam-Webster.com. https://www.merriam-webster.com/dictionary/archetype.

Balasko, B., J. Abonyi, and B. Feil (2005). *Fuzzy Clustering and Data Analysis Toolbox*. http://www.abonyilab.com/software-and-data/fclusttoolbox. Accessed in January 4, 2018. URL: http://www.abonyilab.com/software-and-data/fclusttoolbox.

Bauckhage, C. and C. Thurau (2009). "Making Archetypal Analysis Practical." In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 272–281. DOI: 10.1007/978-3-642-03798-6_28.

Bellman, R, R Kalaba, and L Zadeh (1966). "Abstraction and pattern classification." In: *Journal of Mathematical Analysis and Applications* 13.1, pp. 1–7. DOI: 10.1016/0022-247x(66)90071-0.

Bezdek, J. C. (1973). "Cluster Validity with Fuzzy Sets." In: *Journal of Cybernetics* 3.3, pp. 58–73. DOI: 10.1080/01969727308546047.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer US. DOI: 10.1007/978-1-4757-0450-1.

Bezdek, J. C., J. Keller, R. Krisnapuram, and N. R. Pal (1999). *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer US. DOI: 10.1007/b106267.

Bezdek, J. and R. Hathaway (2002). "VAT: a tool for visual assessment of (cluster) tendency." In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN 02 (Cat. No.02CH37290)*. IEEE. DOI: 10.1109/ijcnn.2002.1007487.

Bose, I. and X. Chen (2015). "Detecting the migration of mobile service customers using fuzzy clustering." In: *Information & Management* 52.2, pp. 227–238. DOI: 10.1016/j.im.2014.11.001.

Chan, B. H. P., D. A. Mitchell, and L. E. Cram (2003). "Archetypal analysis of galaxy spectra." In: *Monthly Notices of the Royal Astronomical Society* 338.3, pp. 790–795. DOI: 10.1046/j.1365-8711.2003.06099.x.

Chen, Y., J. Mairal, and Z. Harchaoui (2014). "Fast and Robust Archetypal Analysis for Representation Learning." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. DOI: 10.1109/cvpr.2014.192.

Chouikhi, H., M. Charrad, and N. Ghazzali (2015). "A comparison study of clustering validity indices." In: *2015 Global Summit on Computer & Information Technology (GSCIT)*. IEEE. DOI: 10.1109/gscit.2015.7353330.

Cosma, G. and G. Acampora (2016). "A computational intelligence approach to efficiently predicting review ratings in e-commerce." In: *Applied Soft Computing* 44, pp. 153–162. DOI: 10.1016/j.asoc.2016.02.024.

Cutler, A. and L. Breiman (1994). "Archetypal Analysis." In: *Technometrics 36 (4)* 36.4, pp. 338–347. ISSN: 338–347. DOI: 10.2307/1269949.

Damle, A. and Y. Sun (2016). "A Geometric Approach to Archetypal Analysis and Nonnegative Matrix Factorization." In: *Technometrics* 59.3, pp. 361–370. DOI: 10.1080/00401706.2016.1247017.

Drachen, A., R. Sifa, C. Bauckhage, and C. Thurau (2012). "Guns, swords and data: Clustering of player behavior in computer games in the wild." In: *2012 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE. DOI: 10.1109/cig.2012.6374152.

Dunn, J. C. (1973). "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters." In: *Journal of Cybernetics* 3.3, pp. 32–57. DOI: 10.1080/01969727308546046.

D'Urso, P., L. D. Giovanni, and R. Massari (2014). "Trimmed fuzzy clustering for interval-valued data." In: *Advances in Data Analysis and Classification* 9.1, pp. 21–40. DOI: 10.1007/s11634-014-0169-3.

Eugster, M. J. A. (2012). "Performance Profiles based on Archetypal Athletes." In: *International Journal of Performance Analysis in Sport* 12.1, pp. 166–187. DOI: 10.1080/24748668.2012.11868592.

Eugster, M. J. A. and F. Leisch (2009). "From Spider-Man to Hero - Archetypal Analysis in R." In: *Journal of Statistical Software* 30.8. DOI: 10.18637/jss.v030.i08.

Eugster, M. J. and F. Leisch (2011). "Weighted and robust archetypal analysis." In: *Computational Statistics & Data Analysis* 55.3, pp. 1215–1225. DOI: 10.1016/j.csda.2010.10.017.

Fehrman, E., A. K. Muhammad, E. M. Mirkes, V. Egan, and A. N. Gorban (2017). "The Five Factor Model of Personality and Evaluation of Drug Consumption Risk." In: *Data Science*. Springer International Publishing, pp. 231–242. DOI: 10.1007/978-3-319-55723-6_18.

Fenza, G., D. Furno, and V. Loia (2012). "Hybrid approach for context-aware service discovery in healthcare domain." In: *Journal of Computer and System Sciences* 78.4, pp. 1232–1247. DOI: 10.1016/j.jcss.2011.10.011.

Ferraro, M. B. and P. Giordani (2015). "A toolbox for fuzzy clustering using the R programming language." In: *Fuzzy Sets and Systems* 279, pp. 1–16. DOI: 10.1016/j.fss.2015.05.001.

Ferreira, M. C., C. M. Salgado, J. L. Viegas, H. Schafer, C. S. Azevedo, S. M. Vieira, and J. M. C. Sousa (2015). "Fuzzy modeling based on Mixed Fuzzy Clustering for health care applications." In: *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. DOI: 10.1109/fuzz-ieee.2015.7338028.

Găceanu, R. D. and H. F. Pop (2012). "A fuzzy incremental clustering approach to hybrid data discovery." In: *Acta Electrotechnica et Informatica* 12.2. DOI: 10.2478/v10198-012-0010-x.

Han, J., M. Kamber, and J. Pei (Aug. 11, 2017). *Data Mining: Concepts and Techniques 3rd ed.* Elsevier LTD, Oxford. ISBN: 0123814790. URL: https://www.ebook.de/de/product/14641128/jiawei_han_micheline_kamber_jian_pei_data_mining_concepts_and_techniques.html.

Höppner, F., F. Klawonn, R. Kruse, and T. Runkler (1999). *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition (Wiley IBM PC Series)*. Wiley. ISBN: 978-0-471-98864-9.

Hu, Y. and R. J. Hathaway (2008). "An Algorithm for Clustering Tendency Assessment." In: *WSEAS Transactions on Mathematics* 7.7, pp. 441–450. URL: https://digitalcommons.georgiasouthern.edu/math-sci-facpubs/28.

Huang, C.-W., K.-P. Lin, M.-C. Wu, K.-C. Hung, G.-S. Liu, and C.-H. Jen (2014). "Intuitionistic fuzzy $$c$$ c -means clustering algorithm with neighborhood attraction in segmenting medical image." In: *Soft Computing-A Fusion of Foundations, Methodologies, and Applications* 19.2, pp. 459–470. DOI: 10.1007/s00500-014-1264-2.

Huggins, P., L. Pachter, and B. Sturmfels (2007). "Toward the Human Genotope." In: *Bulletin of Mathematical Biology* 69.8, pp. 2723–2735. DOI: 10.1007/s11538-007-9244-7.

I. Epifanio G. Vinue, S. A. (2013). "Archetypal analysis: Contributions for estimating boundary cases in multivariate accommodation problem." In: *Computers and Industrial Engineering* 64(3), pp. 757–765. DOI: 10.1016/j.cie.2012.12.011.

Izakian, H., W. Pedrycz, and I. Jamal (2015). "Fuzzy clustering of time series data using dynamic time warping distance." In: *Engineering Applications of Artificial Intelligence* 39, pp. 235–244. DOI: 10.1016/j.engappai.2014.12.015.

Jahromi, A. T., M. J. Er, X. Li, and B. S. Lim (2016). "Sequential fuzzy clustering based dynamic fuzzy neural network for fault diagnosis and prognosis." In: *Neurocomputing* 196, pp. 31–41. DOI: 10.1016/j.neucom.2016.02.036.

John, V., S. Mita, Z. Liu, and B. Qi (2015). "Pedestrian detection in thermal images using adaptive fuzzy C-means clustering and convolutional neural networks." In: *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. IEEE. DOI: 10.1109/mva.2015.7153177.

Karami, A., A. Gangopadhyay, B. Zhou, and H. Karrazi (2015). "FLATM: A fuzzy logic approach topic model for medical documents." In: *2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)*. IEEE, pp. 1–6. DOI: 10.1109/nafips-wconsc.2015.7284190.

Khormali, A. and J. Addeh (2016). "A novel approach for recognition of control chart patterns: Type-2 fuzzy clustering optimized support vector machine." In: *ISA Transactions* 63, pp. 256–264. DOI: 10.1016/j.isatra.2016.03.004.

Kryszczuk, K. and P. Hurley (2010). "Estimation of the Number of Clusters Using Multiple Clustering Validity Indices." In: *Multiple Classifier Systems*. Springer Berlin Heidelberg, pp. 114–123. DOI: 10.1007/978-3-642-12127-2_12.

Li, H. F., F. L. Wang, S. J. Zheng, and L. Gao (2011). "An Improved Fuzzy C-Means Clustering Algorithm and Application in Meteorological Data." In: *Advanced Materials Research* 181-182, pp. 545–550. DOI: 10.4028/www.scientific.net/amr.181-182.545.

Li, J. and H. W. Lewis (2016). "Fuzzy Clustering Algorithms — Review of the Applications." In: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*. IEEE. DOI: 10.1109/smartcloud.2016.14.

Li, Y., G. Yang, H. He, L. Jiao, and R. Shang (2015). "A study of large-scale data clustering based on fuzzy clustering." In: *Soft Computing* 20.8, pp. 3231–3242. DOI: 10.1007/s00500-015-1698-1.

Lichman, M. (2013). *UCI Machine Learning Repository*. URL: http://archive.ics.uci.edu/ml.

Liu, L., S. Z. Sun, H. Yu, X. Yue, and D. Zhang (2016). "A modified Fuzzy C-Means (FCM) Clustering algorithm and its application on carbonate fluid identification." In: *Journal of Applied Geophysics* 129, pp. 28–35. DOI: 10.1016/j.jappgeo.2016.03.027.

Madaleno, N. J. M. (Mar. 2017). *Archetypal Analysis: Retrieving Extreme Prototypes from Data*. Undergraduate Research Opportunity Program in the course of Computer Science of Faculty of Science and Technology. supervisor: Professor Susana Nascimento.

Maity, S. P., S. Chatterjee, and T. Acharya (2016). "On optimal fuzzy c-means clustering for energy efficient cooperative spectrum sensing in cognitive radio networks." In: *Digital Signal Processing* 49, pp. 104–115. DOI: 10.1016/j.dsp.2015.10.006.

Majumdar, D., A. Ghosh, D. K. Kole, A. Chakraborty, and D. D. Majumder (2015). "Application of Fuzzy C-Means Clustering Method to Classify Wheat Leaf Images Based on the Presence of Rust Disease." In: *Advances in Intelligent Systems and Computing*. Springer International Publishing, pp. 277–284. DOI: 10.1007/978-3-319-11933-5_30.

Marinetti, S., L. Finesso, and E. Marsilio (2006). "Matrix factorization methods: Application to thermal NDT/E." In: *NDT & E International* 39.8, pp. 611–616. DOI: 10.1016/j.ndteint.2006.04.008.

— (2007). "Archetypes and principal components of an IR image sequence." In: *Infrared Physics & Technology* 49.3, pp. 272–276. DOI: 10.1016/j.infrared.2006.06.017.

Marsaglia, G. (1972). "Choosing a Point from the Surface of a Sphere." In: *The Annals of Mathematical Statistics* 43.2, pp. 645–646. DOI: 10.1214/aoms/1177692644.

MATLAB (2015). *version 8.5.0 (R2015a)*. Natick, Massachusetts: The MathWorks Inc.

Mendes, G. S. and S. Nascimento (2018). "A Study of Fuzzy Clustering to Archetypal Analysis." In: *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Springer International Publishing, pp. 250–261. DOI: 10.1007/978-3-030-03496-2_28.

Mirkin, B. (2005). *Clustering for Data Mining: A Data Recovery Approach*. Chapman and Hall/CRC. DOI: 10.1201/9781420034912.

Mørup, M. and L. K. Hansen (2012). "Archetypal analysis for machine learning and data mining." In: *Neurocomputing* 80, pp. 54–63. DOI: 10.1016/j.neucom.2011.06.033.

Nascimento, S. (2005). *Fuzzy Clustering via Proportional Membership Model (Frontiers in Artificial Intelligence and Applications)*. IOS Press. ISBN: 1586034898. URL: https://www.amazon.com/Clustering-Proportional-Membership-Intelligence-Applications/dp/1586034898?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1586034898.

Nascimento, S., B. Mirkin, and F. Moura-Pires (2003). "Modeling proportional membership in fuzzy clustering." In: *IEEE Transactions on Fuzzy Systems* 11.2, pp. 173–186. DOI: 10.1109/tfuzz.2003.809889.

Nascimento, S. (2002). "Fuzzy Clustering Via Proportional Membership Model." Doctoral dissertation. UNL.

Nascimento, S. and P. Franco (2009). "Segmentation of Upwelling Regions in Sea Surface Temperature Images via Unsupervised Fuzzy Clustering." In: *Intelligent Data Engineering and Automated Learning - IDEAL 2009*. Springer Berlin Heidelberg, pp. 543–553. DOI: 10.1007/978-3-642-04394-9_66.

Pei, Yaling; Zaiane, Osmar (2006). "A Synthetic Data Generator for Clustering and Outlier Analysis." In: pp. –. DOI: 10.7939/r3b23s.

Peng, H.-W., S.-F. Wu, C.-C. Wei, and S.-J. Lee (2015). "Time series forecasting with a neuro-fuzzy modeling scheme." In: *Applied Soft Computing* 32, pp. 481–493. DOI: 10.1016/j.asoc.2015.03.059.

Pirker, J., S. Griesmayr, A. Drachen, and R. Sifa (2016). "How Playstyles Evolve: Progression Analysis and Profiling in Just Cause 2." In: *Entertainment Computing - ICEC 2016*. Springer International Publishing, pp. 90–101. DOI: 10.1007/978-3-319-46100-7_8.

Porzio, G. C., G. Ragozini, and D. Vistocco (2006). "Archetypal Analysis for Data Driven Benchmarking." In: *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer Berlin Heidelberg, pp. 309–318. DOI: 10.1007/3-540-35978-8_35.

— (2008). "On the use of archetypes as benchmarks." In: *Applied Stochastic Models in Business and Industry* 24.5, pp. 419–437. DOI: 10.1002/asmb.727.

Ragozini, G. and M. R. D'Esposito (2015). "Archetypal Networks." In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM 15*. ACM Press. DOI: 10.1145/2808797.2808837.

Römer, C., M. Wahabzada, A. Ballvora, F. Pinto, M. Rossini, C. Panigada, J. Behmann, J. Léon, C. Thurau, C. Bauckhage, K. Kersting, U. Rascher, and L. Plümer (2012). "Early drought stress detection in cereals: simplex volume maximisation for hyperspectral image analysis." In: *Functional Plant Biology* 39.11, p. 878. DOI: 10.1071/fp12060.

Ruspini, E. H. (1969). "A new approach to clustering." In: *Information and Control* 15.1, pp. 22–32. DOI: 10.1016/s0019-9958(69)90591-9.

Sammon, J. (1969). "A Nonlinear Mapping for Data Structure Analysis." In: *IEEE Transactions on Computers* C-18.5, pp. 401–409. DOI: 10.1109/t-c.1969.222678.

Schafer, H., J. L. Viegas, M. C. Ferreira, S. M. Vieira, and J. M. C. Sousa (2015). "Analysing the segmentation of energy consumers using mixed fuzzy clustering." In: *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. DOI: 10.1109/fuzz-ieee.2015.7338120.

Seiler, C. and K. Wohlrabe (2013). "Archetypal scientists." In: *Journal of Informetrics* 7.2, pp. 345–356. DOI: 10.1016/j.joi.2012.11.013.

Sert, S. A., H. Bagci, and A. Yazici (2015). "MOFCA: Multi-objective fuzzy clustering algorithm for wireless sensor networks." In: *Applied Soft Computing* 30, pp. 151–165. DOI: 10.1016/j.asoc.2014.11.063.

Seth, S. and M. J. A. Eugster (2015). "Probabilistic archetypal analysis." In: *Machine Learning* 102.1, pp. 85–113. DOI: 10.1007/s10994-015-5498-8.

Seth, S. and M. J. Eugster (2016). "Archetypal Analysis for Nominal Observations." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.5, pp. 849–861. DOI: 10.1109/tpami.2015.2470655.

Sifa, R. and C. Bauckhage (2013). "Archetypical motion: Supervised game behavior learning with Archetypal Analysis." In: *2013 IEEE Conference on Computational Inteligence in Games (CIG)*. IEEE. DOI: 10.1109/cig.2013.6633609.

Stetco, A., X. jun Zeng, and J. Keane (2013). "Fuzzy Cluster Analysis of Financial Time Series and Their Volatility Assessment." In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, pp. 91–96. DOI: 10.1109/smc.2013.23.

Stone, E. (2002). "Exploring archetypal dynamics of pattern formation in cellular flames." In: *Physica D: Nonlinear Phenomena* 161.3-4, pp. 163–186. DOI: 10.1016/s0167-2789(01)00361-x.

Stone, E. and A. Cutler (1996). "Archetypal analysis of spatio-temporal dynamics." In: *Physica D: Nonlinear Phenomena* 90.3, pp. 209–224. DOI: 10.1016/0167-2789(95)00244-8.

Suleman, A. (2017). "Validation of archetypal analysis." In: *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. DOI: 10.1109/fuzz-ieee.2017.8015385.

Sun, Z., L. Gao, S. Wei, and S. Zheng (2010). "A Fuzzy C-Means Clustering Algorithm and Application in Meteorological Data." In: *2010 Second International Conference on Modeling, Simulation and Visualization Methods*. IEEE. DOI: 10.1109/wmsvm.2010.24.

Thøgersen, J., M. Mørup, S. Damkiær, S. Molin, and L. Jelsbak (2013). "Archetypal analysis of diverse Pseudomonas aeruginosa transcriptomes reveals adaptation in cystic fibrosis airways." In: *BMC Bioinformatics* 14.1, p. 279. DOI: 10.1186/1471-2105-14-279.

Thurau, C. and C. Bauckhage (2009). "Archetypal Images in Large Photo Collections." In: *2009 IEEE International Conference on Semantic Computing*. IEEE. DOI: 10.1109/icsc.2009.34.

Tsekouras, G. E. and D. Gavalas (2013). "An effective Fuzzy Clustering Algorithm for Web Document Classification: A Case Study in Cultural Content Mining." In: *International Journal of Software Engineering and Knowledge Engineering* 23.06, pp. 869–886. DOI: 10.1142/s021819401350023x.

Tufan, E. and B. Hamarat (2003). "Clustering of Financial Ratios of the Quoted Companies Through Fuzzy Logic Method." In: *SSRN Electronic Journal* 1.2, pp. 123–140. DOI: 10.2139/ssrn.461700.

Typology (Jan. 2018). Merriam-Webster.com. https://www.merriam-webster.com/dictionary/typology.

Vinué, G. and I. Epifanio (2017). "Archetypoid analysis for sports analytics." In: *Data Mining and Knowledge Discovery* 31.6, pp. 1643–1677. DOI: 10.1007/s10618-017-0514-1.

Vinué, G., I. Epifanio, and S. Alemany (2015). "Archetypoids: A new approach to define representative archetypal data." In: *Computational Statistics & Data Analysis* 87, pp. 102–115. DOI: 10.1016/j.csda.2015.01.018.

Wang, Z., M. Jiang, Y. Hu, and H. Li (2012). "An Incremental Learning Method Based on Probabilistic Neural Networks and Adjustable Fuzzy Clustering for Human Activity Recognition by Using Wearable Sensors." In: *IEEE Transactions on Information Technology in Biomedicine* 16.4, pp. 691–699. DOI: 10.1109/titb.2012.2196440.

Wu, Z., H. Zhang, and J. Liu (2014). "A fuzzy support vector machine algorithm for classification based on a novel PIM fuzzy clustering method." In: *Neurocomputing* 125, pp. 119–124. DOI: 10.1016/j.neucom.2012.07.049.

Xianfeng, Y. and L. Pengfei (2015). "Tailoring Fuzzy C-Means Clustering Algorithm for Big Data Using Random Sampling and Particle Swarm Optimization." In: *International Journal of Database Theory and Application* 8.3, pp. 191–202. DOI: 10.14257/ijdta.2015.8.3.16.

Xiong, Y., W. Liu, D. Zhao, and X. Tang (2013). "Face Recognition via Archetype Hull Ranking." In: *2013 IEEE International Conference on Computer Vision*. IEEE. DOI: 10.1109/iccv.2013.78.

Yeh, I.-C. and C. hui Lien (2009). "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients." In: *Expert Systems with Applications* 36.2, pp. 2473–2480. DOI: 10.1016/j.eswa.2007.12.020.

Yera, A., O. Arbelaitz, J. L. Jodra, I. Gurrutxaga, J. M. Pérez, and J. Muguerza (2017). "Analysis of several decision fusion strategies for clustering validation. Strategy definition, experiments and validation." In: *Pattern Recognition Letters* 85, pp. 42–48. DOI: 10.1016/j.patrec.2016.11.009.

Yolcu, O. C. (2013). "A Hybrid Fuzzy Time Series Approach Based on Fuzzy Clustering and Artificial Neural Network with Single Multiplicative Neuron Model." In: *Mathematical Problems in Engineering* 2013, pp. 1–9. DOI: 10.1155/2013/560472.

Zadeh, L. (1965). "Fuzzy sets." In: *Information and Control* 8.3, pp. 338–353. DOI: 10.1016/s0019-9958(65)90241-x.

Zimmermann, A. (2015). "The Data Problem in Data Mining." In: *ACM SIGKDD Explorations Newsletter* 16.2, pp. 38–45. DOI: 10.1145/2783702.2783706.

Appendix A

Plots for Synthetic Data

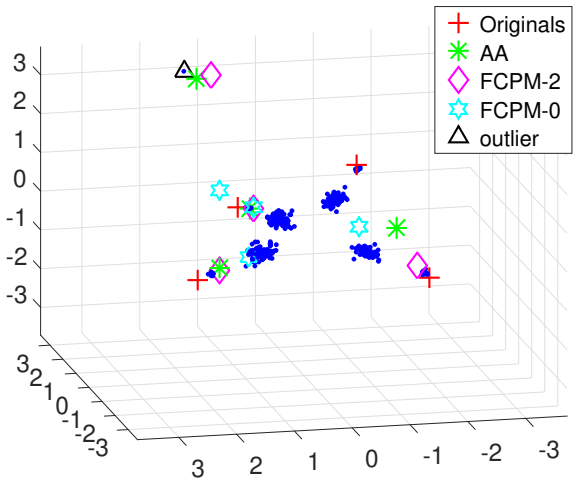


Figure A.1: Principal components projection (R=71%), for the first setting, $k = c_0$, $out = 1$ of a medium dimensionality data set (n=416, p=40, c=4). One of the archetypes is between two clusters and one FCPM-2 prototype and one archetype in the outlier.

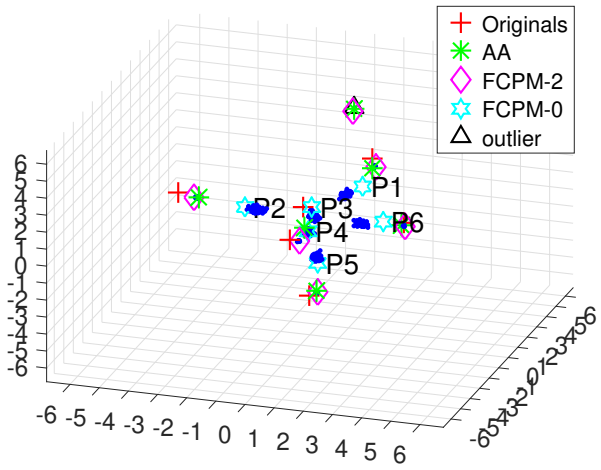


Figure A.2: Principal components projection (R=71%), for the first setting, $k = c_0$, $out = 1$ of a high dimensionality data set (n=624, p=180, c=6). The FCPM-0 prototypes are marked. All of FCPM-0 prototypes are inside the data space.

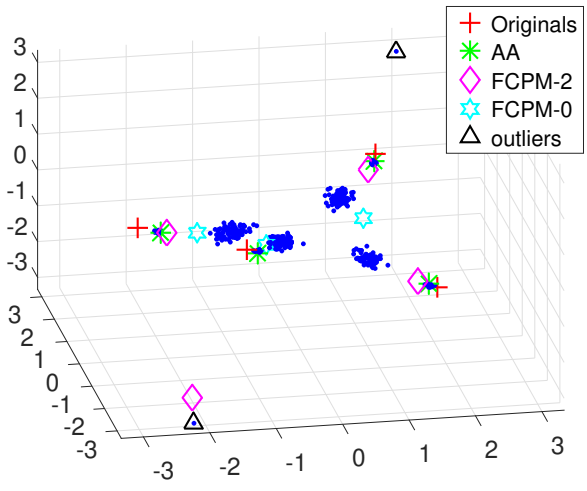


Figure A.3: Principal components projection (R=96%), for the second setting ($k = c_0$, $out = 2$) of a medium dimensional data set (n=440, p=40, c=4). One of the FCPM-2 prototypes near an outlier, and the other in the data space near an original. All archetypes are near the originals. One of the FCPM-0 prototypes is outside of the space.

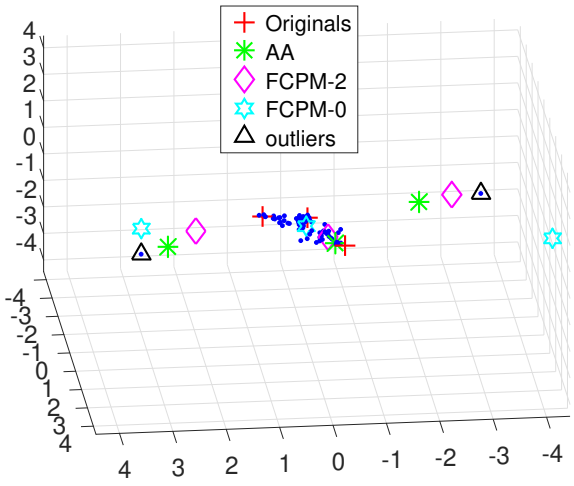


Figure A.4: Principal components projection (R=99%) for second setting, $k = c_0$, $out = 2$, for a small dimensional data set (n=70, p=5, c=3). One archetype and one FCPM-2 prototype are near each of the outliers. One FCPM-0 prototype is outside of the data space and another near one outlier.

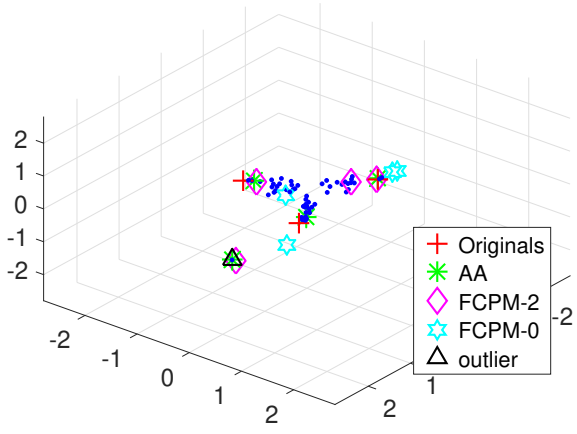


Figure A.5: Principal components projection (R=99%) for third setting, $k = c_0 + 1$, $out = 1$, of a small dimensional data set (n=62, p=5, c=3). The extra archetype and FCPM-2 prototype near the outlier and two of FCPM-0/FCPM-2 prototypes near the same original.

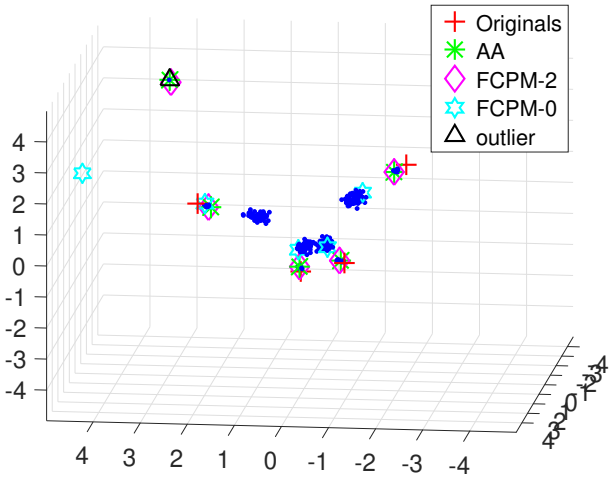


Figure A.6: Principal components projection (R=87%) for third setting, $k = c_0 + 1$, $out = 1$, of a medium dimensional data set (n=276, p=50, c=4). The extra archetype and FCPM-2 prototype are near the outlier. One FCPM-0 prototype outside of the data space and all the others inside, indicating the true number os clusters, 4.

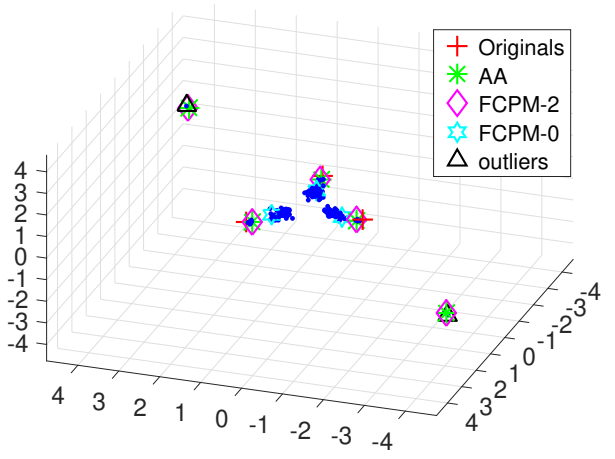
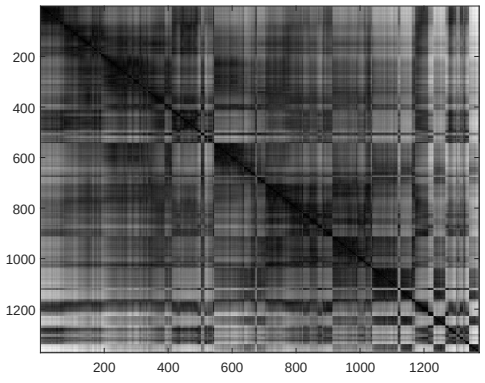


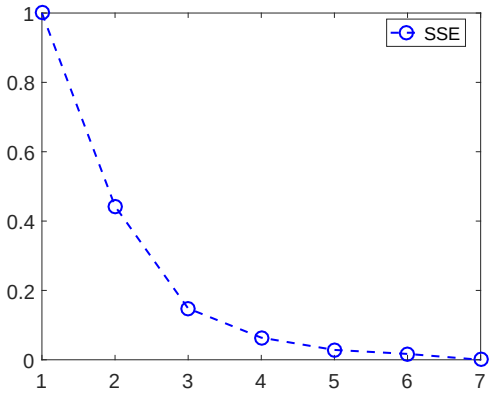
Figure A.7: Principal components projection (R=99%) of medium dimensional (n=199, p=15, c=3) data set for the fifth setting ($k = c_0 + 2$, $out = 2$). It shows the extras archetypes/prototypes in the outliers. Two FCPM-0 prototypes are outside of the data space.

Appendix B

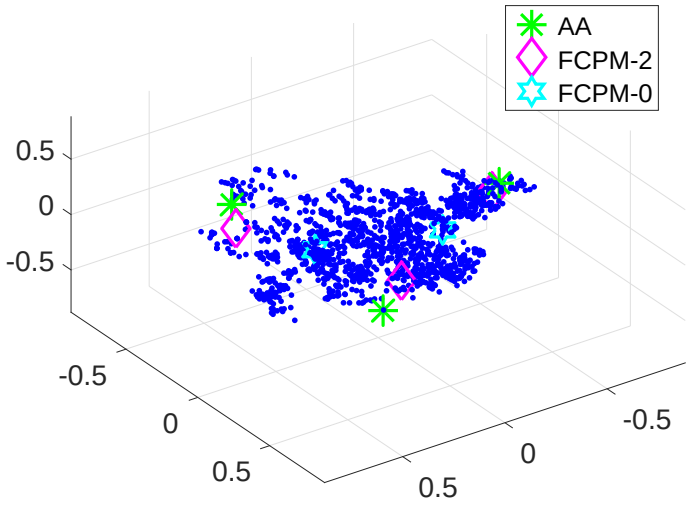
Plots for Real World Data



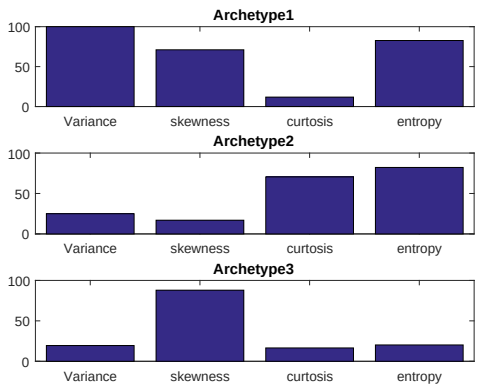
a VAT plot with no clear indication of the number of clusters.



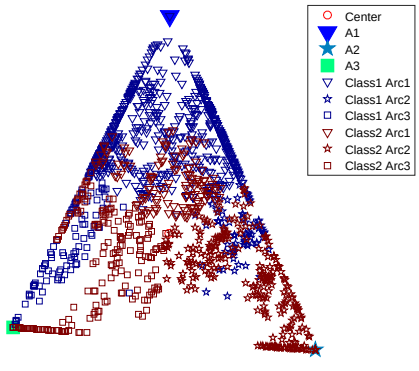
b SSE plot indicating 3 clusters.



c PC projection (R=99%) with the archetypes/prototypes found for k=3. One FCPM-0 prototype is outside of data space.

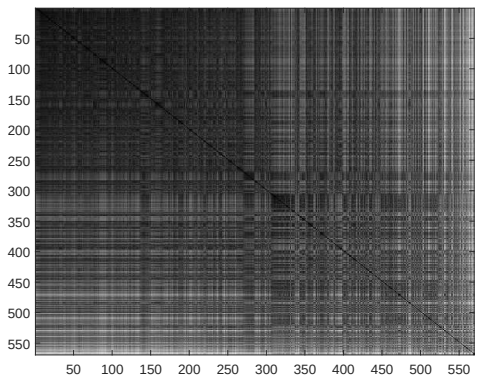


d Percentile plot for k=3

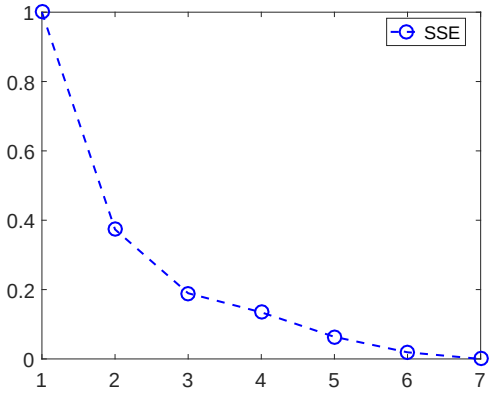


e Mixture plot for the AA solution with the distances preserved and the labels.

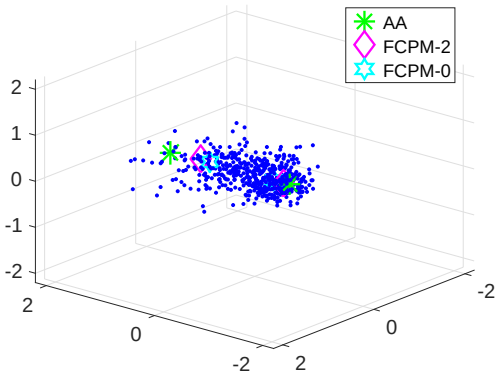
Figure B.1: Plots for the bank authentication data.



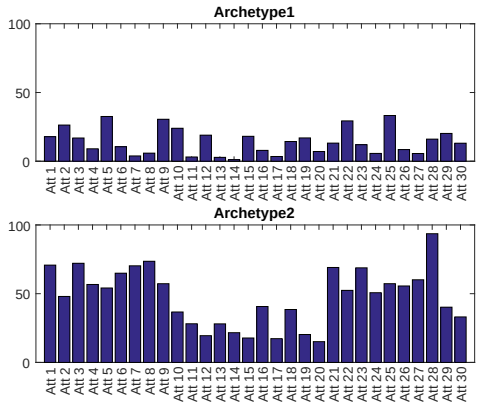
a VAT plot with no clear indication on the number of clusters.



b The SSE plot indicating 3 clusters.

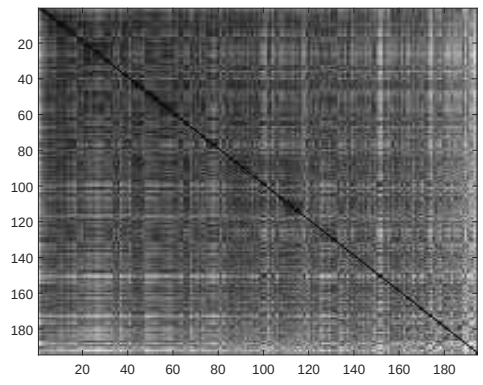


c A 3D projection (R=97%) with the archetypes/prototypes found for k=2.

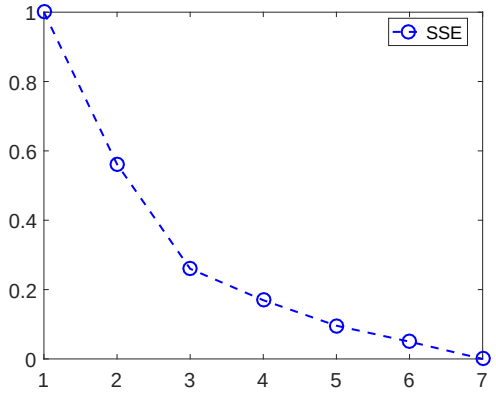


d Percentile plot for k=2.

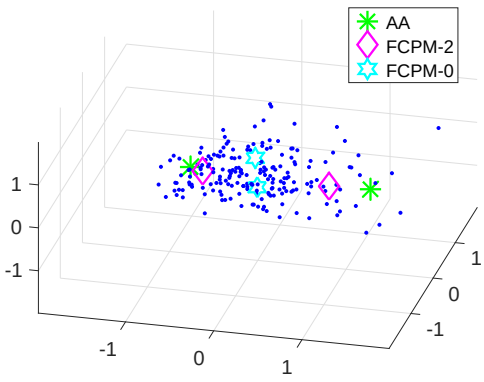
Figure B.2: Plots for Wisconsin Breast Cancer Diagnostic data.



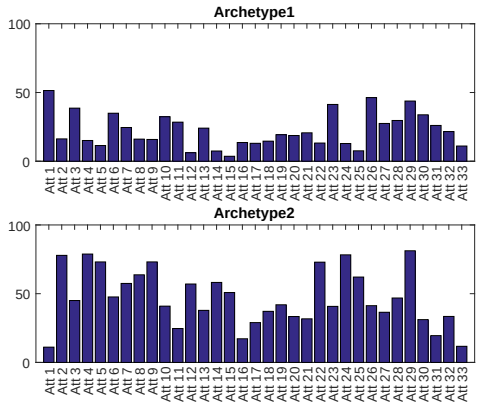
a VAT plot with no clear indication on the number of clusters.



b The SSE plot indicating 3 clusters.

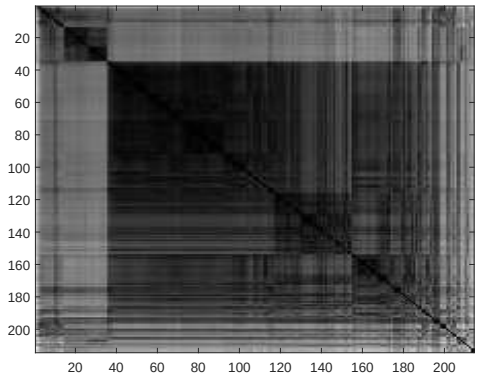


c PC projection (R=92%) with the archetypes/prototypes found for k=2.

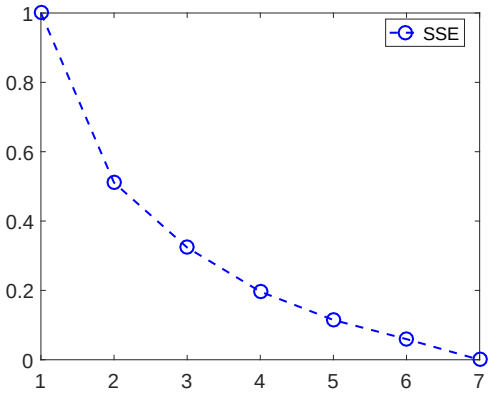


d Percentile plot for k=2.

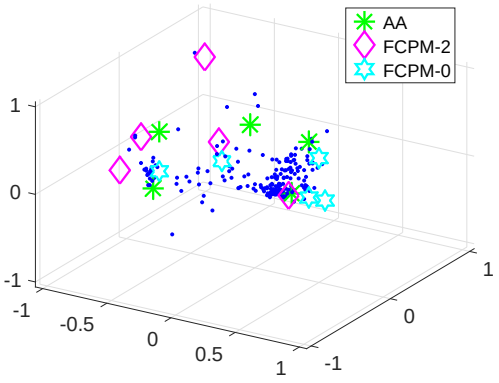
Figure B.3: Plots for the Wisconsin Breast Cancer Prognostic data.



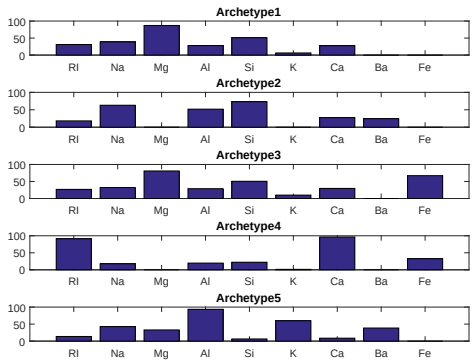
a VAT plot indicating 2 clusters.



b SSE plot indicating 2 clusters.

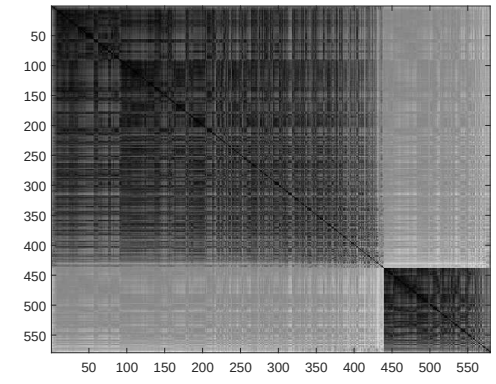


c PC projection (R=95%) with the archetypes/prototypes found for k=5.

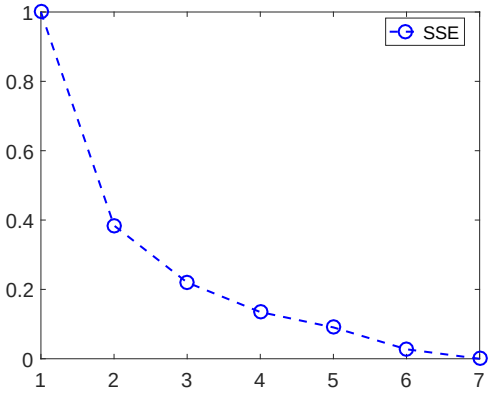


d Percentile plot for k=5.

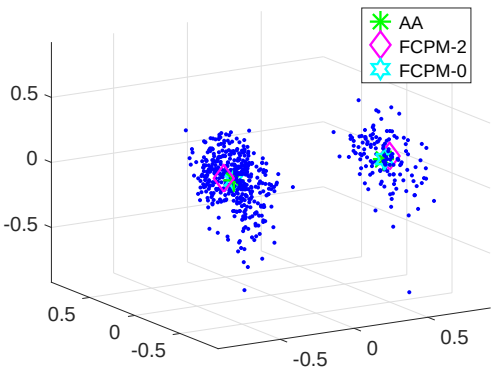
Figure B.4: Plots for the Glass identification data.



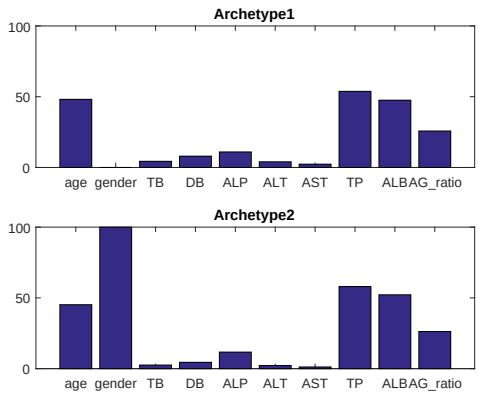
a VAT plot indicating 2 clusters.



b SSE plot indicating 2 clusters.

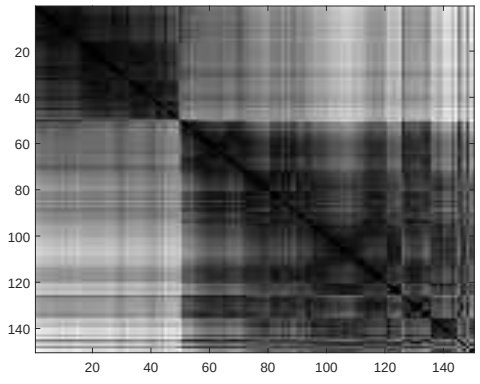


c PC projection (R=97%) with the archetypes/prototypes found for k=2.

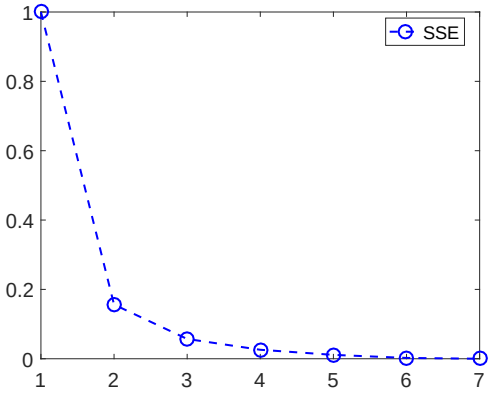


d Percentile plot for k=2.

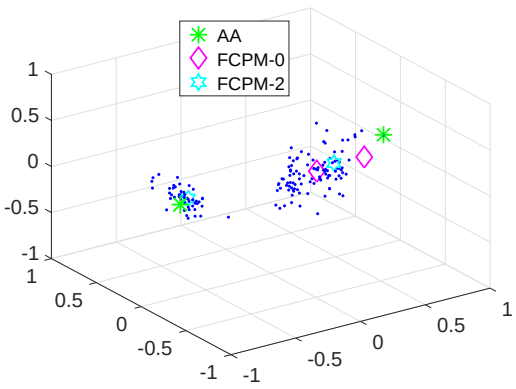
Figure B.5: Plots for the Indian Liver Patient data.



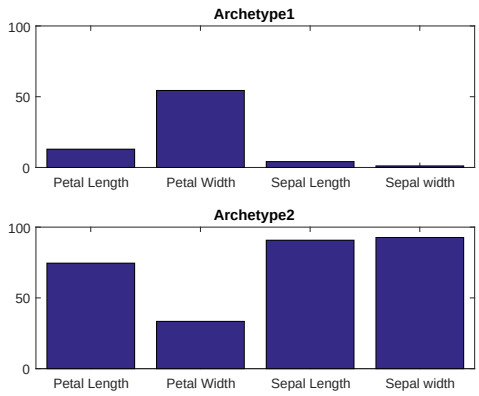
a VAT plot indicating 2 clusters.



b The SSE plot indicating 2 clusters.

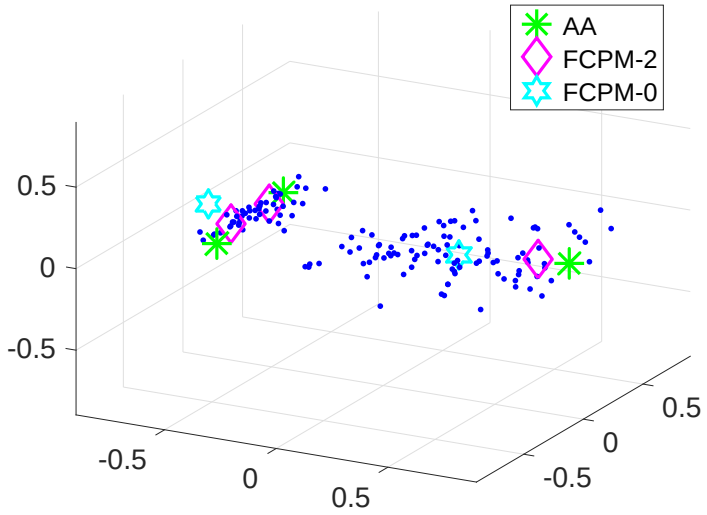


c PC projection (R=99%) with the archetypes/prototypes found for k=2.

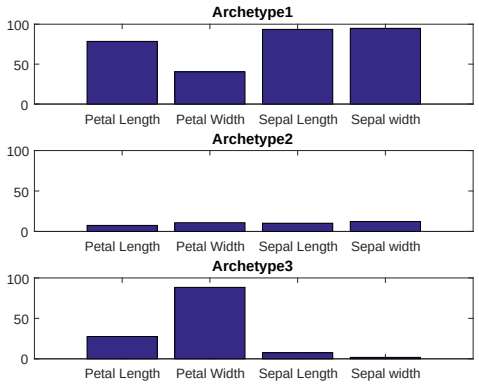


d Percentile plot.

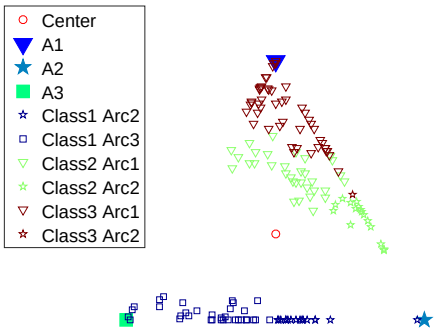
Figure B.6: Plots for the Iris data.



a PC projection (R=99%) with the archetypes/prototypes found for k=3. One FCPM-0 prototype is outside data space.

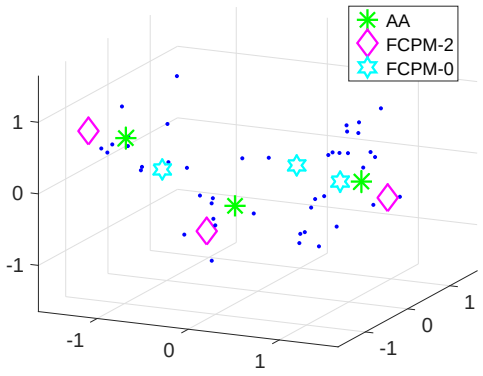


b Percentile plot for k=3.

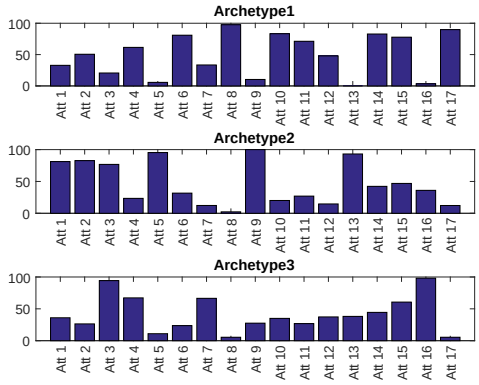


c Mixture plot for the 3 archetypes with the labels.

Figure B.7: Plots for the Iris data (cont.).

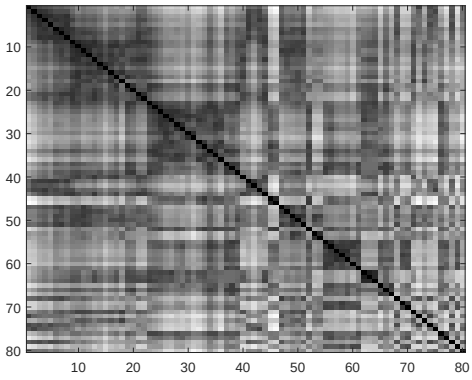


a PC projection (R=98%) with the archetypes/prototypes found for k=3.

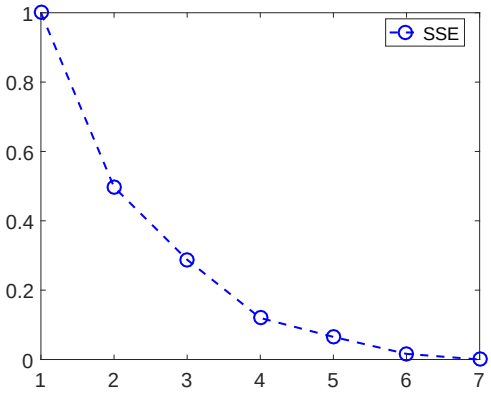


b Percentile plot for k=3.

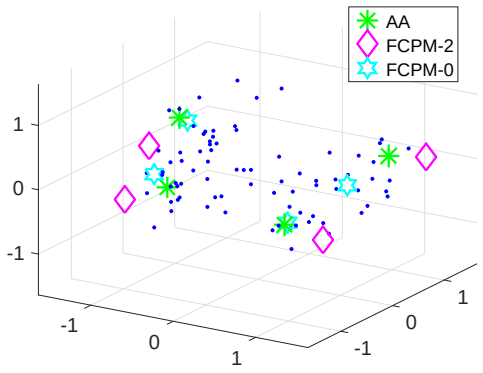
Figure B.8: Plots for the Mental Disorders data for $k = 3$.



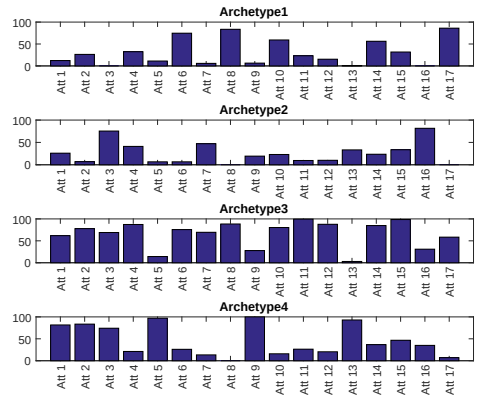
a VAT plot with no clear indication of the number of cluster.



b SSE plot indicating 4 clusters.

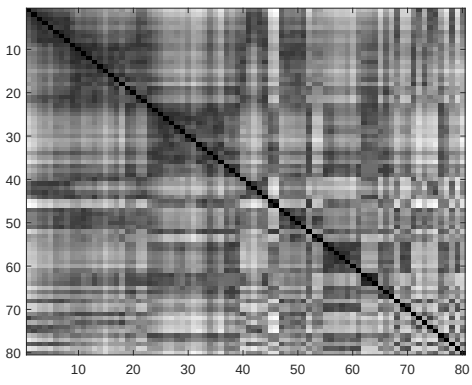


c PC projection (R=98%) with the archetypes/prototypes found for k=4.

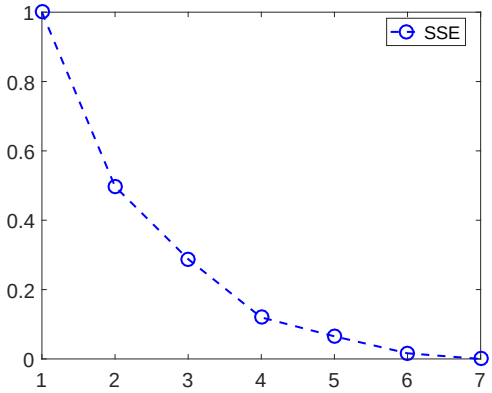


d Percentile plot for k=4.

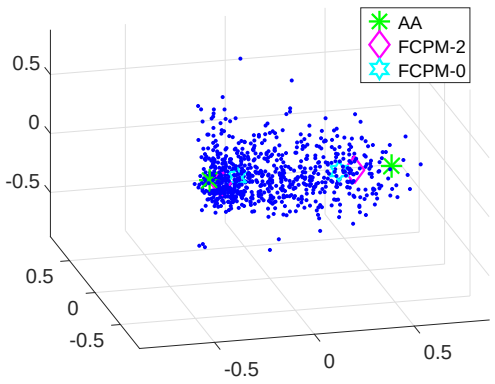
Figure B.9: Plots for Mental disorders augmented data.



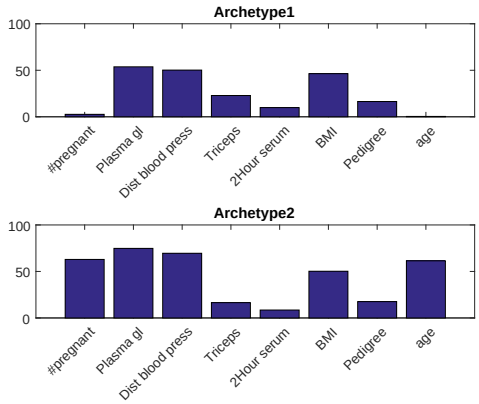
a VAT plot with no clear indication of the number of cluster.



b SSE plot indicating 4 clusters.

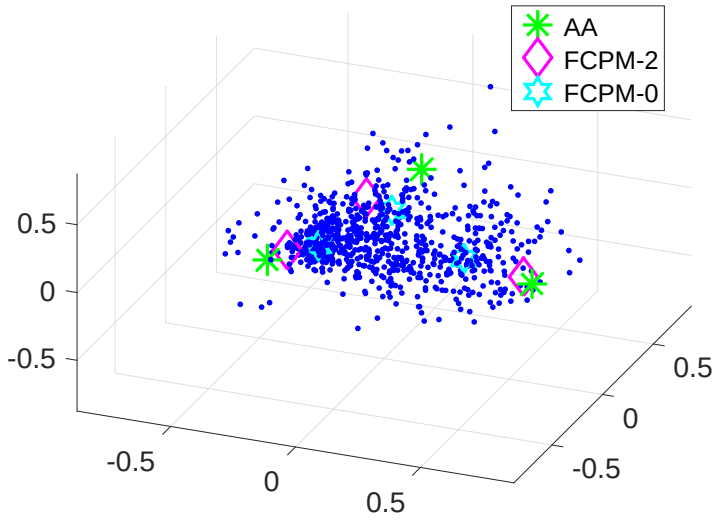


c PC projection (R=84%) with the archetypes/prototypes found for k=2.

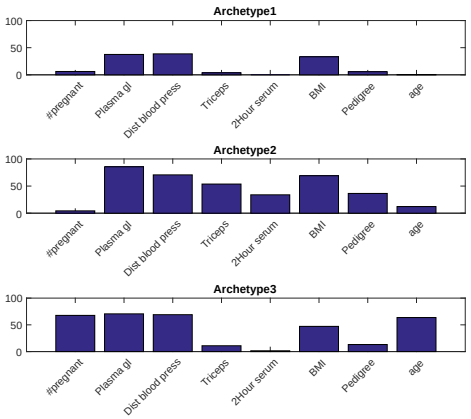


d Percentile plot for k=2.

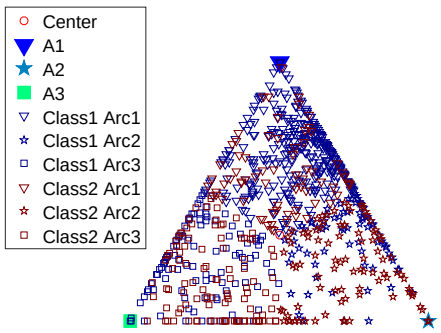
Figure B.10: Plots for the Pima Indians Diabetes data.



a PC projection (R=85%) with the archetypes/prototypes found for k=3.

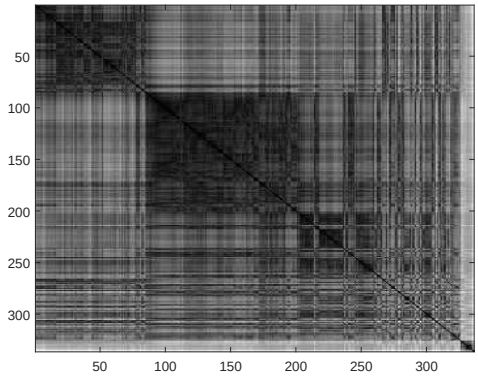


b Percentile plot for k=3.

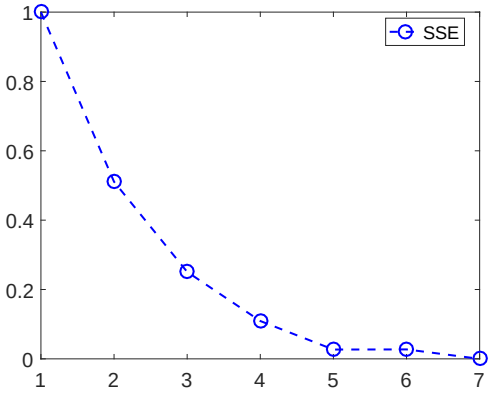


c Mixture plot for k=3.

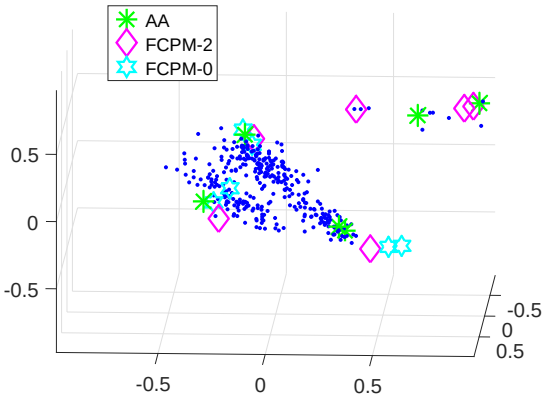
Figure B.11: Plots for the Pima Indians Diabetes data (cont.).



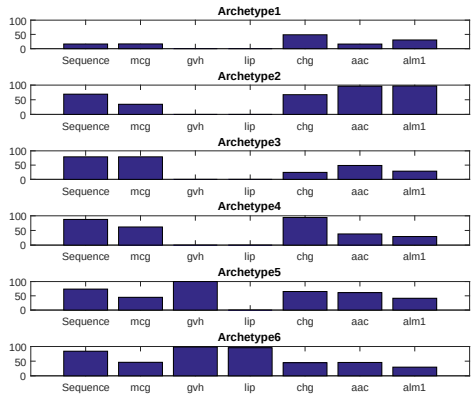
a VAT plot indicating at least 3 clusters.



b SSE plot indicating 5 clusters.

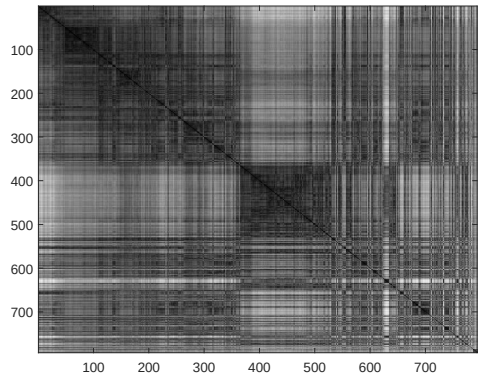


c PC projection (R=96%) with the archetypes/prototypes found for 6.

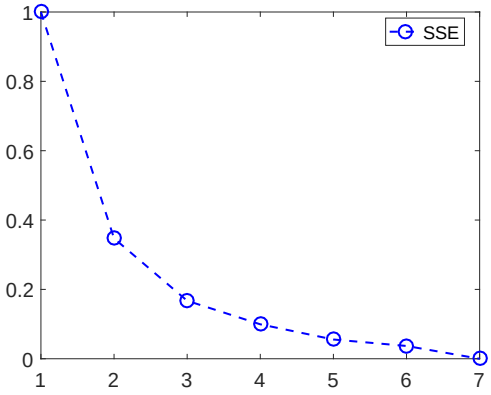


d Percentile plot for k=6.

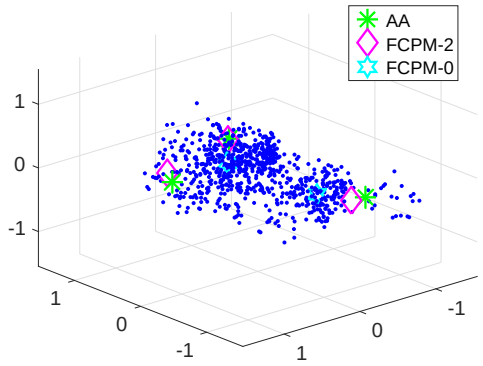
Figure B.12: Plots for the Protein location site data (E. Coli) .



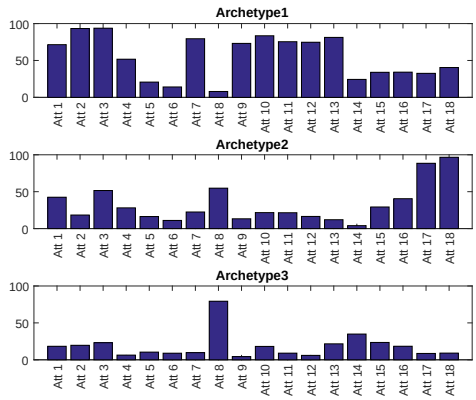
a VAT plot indicating at least 2 clusters.



b SSE plot indicating 3 clusters.

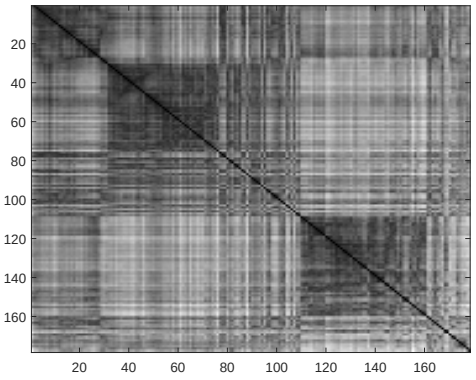


c PC projection (R=98%) with the archetypes/prototypes found for k=3. One FCPM-0 prototype is outside of data space.

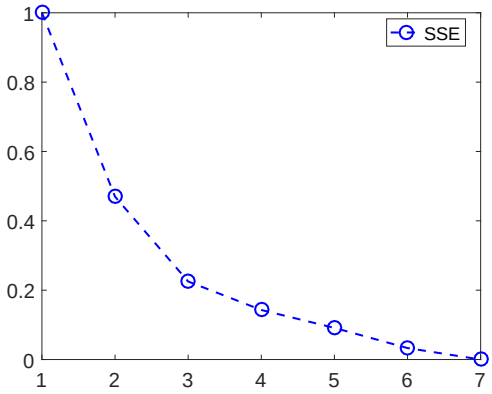


d Percentile plot for k=3.

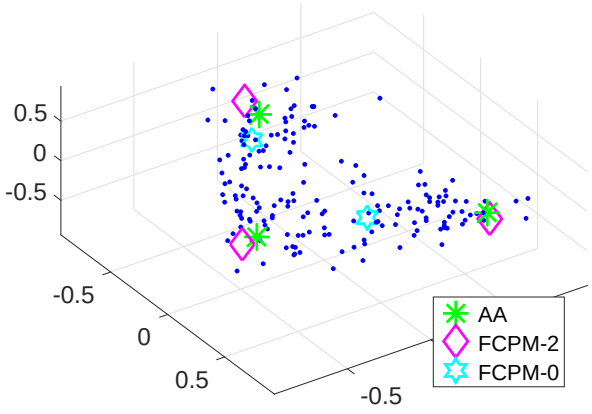
Figure B.13: Plots for the Vehicle Silhouettes data.



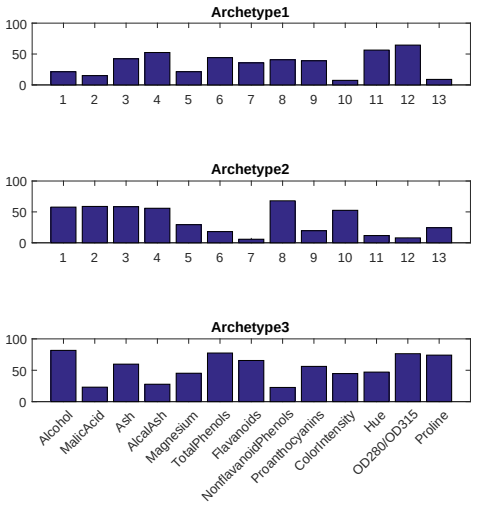
a VAT plot indicating at least 2 clusters.



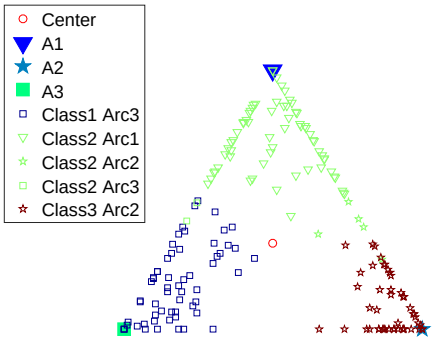
b SSE plot indicating 3 clusters.



c A 3D projection (R=98%) with the archetypes/prototypes found for k=3. One FCPM-0 prototype is outside of data space.



d Percentile plot for k=3.



e Mixture plot for k=3.

Figure B.14: Plots for the Wine Recognition data.