TÉCNICO LISBOA

Observer biased clustering in wind turbine fault diagnosis

Bernardo Alpoim Filipe Ferreira Pimenta

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. José Luís Valente de Oliveira
Prof. João Miguel da Costa Sousa

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Prof. José Luís Valente de Oliveira
Member of the Committee: Prof. João Filipe Pereira Fernandes

December 2021

Acknowledgments

I would like to thank my supervisors Prof. José Valente de Oliveira and Prof. João Sousa for all the guidance, feedback and availability throughout this dissertation. Also thanks to my family and friends for the continued support.

The work was funded in part by national funds through FCT—Foundation for Science and Technology, I.P., through IDMEC, under LAETA, project UIDB/50022/2020, the Prometeo Project of SENESCYT, Ecuador, and the "Multimodal Deep Learning of Deterioration and Fault Prognosis for Wind Turbine Drivetrains" project, funded by Bilateral cooperation between Portugal (FCT: 441.00) and China, through The National Key Research & Development Program, (MOST: 2016YFE0132200). The experimental work relative to bearings was developed at the GIDTEC research group of UPS, Cuenca, Ecuador. The experimental work relative to the wind turbine was developed at the Chongqing Technology and Business University (CTBU).

Abstract

Over the last two decades wind power has undergone an exponential growth globally. As the demand for wind power systems increases, efficiency is sought to be maximized and operation and maintenance costs reduced. Condition monitoring (CM) systems have become a field of high interest in this industry. The main components of wind turbines constitute the focus of CM, they are responsible for frequent, large repair costs and operational downtime. Within the main components, the gearbox accounts for one of the highest failure rates and is the component that causes the greatest amount of downtime.

The focus is on the application of fuzzy clustering for wind turbine gearboxes fault diagnosis. The Gath-Geva clustering algorithm is explored by applying the observer biased clustering framework. The notion of an observer allows clustering to be an interactive process, providing an intuitive way to control cluster formation and enabling domain knowledge to be incorporated in the process. A domain expert can choose the level of granularity and is able to select a particular region of the data space for a detailed view.

The Gath-Geva with Focal Point algorithm is tested with wind turbine gearbox vibrational data and compared with its unbiased version, the Fuzzy C-Means biased and unbiased algorithms. Two metrics are employed to validate internal clustering: the Xie-Beni index and Kim-Lee index, the latter of which is based on relative degree of sharing. The algorithms are compared by performing several independent runs and using the distribution of the Adjusted Rand Index external validation metric.

Keywords: Fuzzy clustering, Wind turbine fault detection, Observer biased clustering, Focal Point, Gath Geva with Focal Point (GGFP)

Resumo

Nas últimas duas décadas, a energia eólica viu um crescimento exponencial. À medida que aumenta a procura por sistemas de energia eólica, busca-se maximizar a eficiência e reduzir os custos de operação e manutenção. Sistemas de monitoramento de condições tornaram-se num campo de grande interesse nesta área. Dentro dos componentes principais, a caixa de engrenagens é responsável por uma das maiores taxas de falha e é o componente que causa o maior tempo de inatividade .

O foco é a aplicação de agrupamento fuzzy para diagn de falhas em caixas de engrenagens de turbinas eólicas. O algoritmo de agrupamento Gath-Geva é explorado aplicando a estrutura de agrupamento com observador. A noção de um observador permite que o agrupamento seja um processo interativo, fornecendo uma maneira intuitiva de controlar a formação de agrupamentos e permitindo que o conhecimento do domínio seja incorporado no processo. Um especialista do domínio pode escolher o nível de granularidade e é capaz de selecionar uma região específica dos dados para análise detalhada.

O Gath-Geva com ponto focal é testado com dados vibracionais da caixa de engrenagens de uma turbina eólica e comparado com a sua versão original, Fuzzy C-Means com e sem observador. Para validação de agrupamento interno, duas métricas são comparadas, o popular índice Xie-Beni e o índice Kim-Lee com base no grau relativo de partilha. Os algoritmos são comparados fazendo várias execuções independentes e usando a distribuição da validação externa do índice Rand Ajustado.

Palavras-chave: Agrupamento Fuzzy , Deteção de falhas em turbinas eólicas, Agrupamento com observador, Ponto focal, Gath Geva com ponto focal (GGFP)

Contents

List of Tables

List of Figures

Nomenclature

ℓ	Maximum number of iterations
ϵ	Sensitivity threshold parameter
λ_j	Lagrangean multipliers
ζ	Regularization coefficient
A	Positive definite norm-inducing matrix
c	Number of clusters
F	Fuzzy covariance matrix
h	Entropy of data points
J	Objective function
m	Partition matrix coefficient
S	Relative degree of sharing between two fuzzy clusters
U	Partition matrix
u_{ij}	Membership degree of data point j to cluster i
v_i	Prototype of cluster i
X	Multivariate data set
Pr(i)	Priori probability of cluster i

Acronyms

AI Artificial Intelligence

ARI Adjusted Rand Index

CM Condition Monitoring

CVI Clustering Validity Indexes

FCM Fuzzy C-Means

GG Gath-Geva

GGFP Gath-Geva with Focal Point

GWEC Global Wind Energy Council

HAWT Horizontal axis wind turbine

KL Kim-Lee

ML Machine Learning

RI Rand Index

WT Wind Turbines

XB Xie-Beni

Chapter 1

Introduction

1.1 Motivation

The number of wind turbines that are harnessing the natural power of the wind and converting it into electricity is increasing daily on a global level. The wind is a clean, reliable and free source of renewable energy, and thus wind power generation plays a crucial role in the path to achieving a clean, sustainable manner of powering our world. The wind power industry has experienced a rapid growth over the last two decades, a report from by the Global Wind Energy Council (GWEC) shows that 2020 saw new wind power installations surpass 90 gigawatts (GW), an astounding growth of 53% compared to 2019 (figure 1.1), bringing the global total installed capacity to 743 GW, an increase of 14% compared to 2019. The total installed wind energy capacity currently satisfies approximately 7% of the global electricity demand [1].

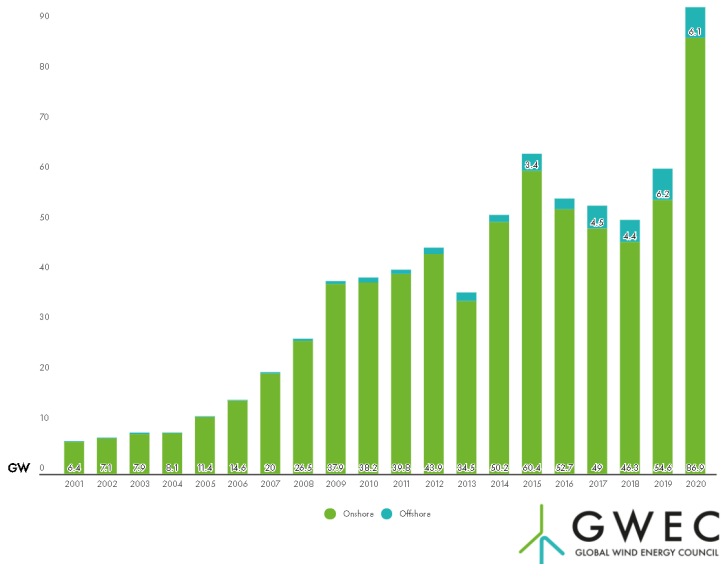


Figure 1.1: Global new wind power capacity per year, from 2001 to 2020. Source: Global Wind Energy Council (GWEC) 2021 [1]

Wind Turbines (WT) allow for the transformation of wind power into energy. In basic terms, the power

of the wind spins the turbine's blades, thus triggering the rotation of the main shaft (low speed shaft), connected to a gearbox within the nacelle. The gearbox, through the high speed shaft, transfers the wind energy to the generator, where it is then converted into electricity. The electricity generated travels to a transformer, where voltage levels are adjusted to the grid requirements. To illustrate this procedure, a diagram of the components for the commonly used Horizontal axis wind turbine (HAWT) with a gearbox can be seen in figure 1.2.

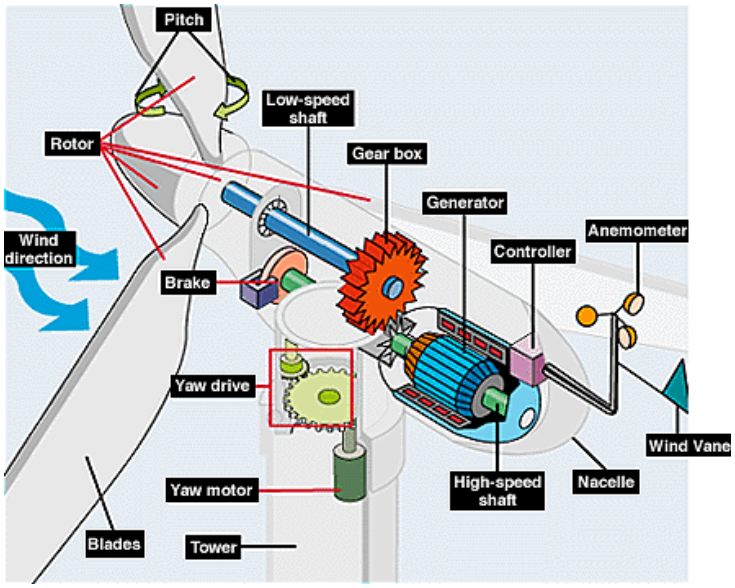


Figure 1.2: Diagram of wind turbine components. Source: National Renewable Energy Laboratory, U.S. Department of Energy. [2]

The drive train on a turbine with a gearbox usually includes the rotor, main bearing, shafts, gearbox, and generator. Due to all the moving parts that exist in the drive train it is considered the area more likely to be subject to failure, leading to undesirable costs and downtime. From all the components within the drive train, the gearbox is responsible for 12% of all failures (the second highest rate) [3] and is also the component whose failure causes the most downtime [4]. These factors make it essential to ensure the healthy and stable operation of this type of equipment. To address this problem, gearbox fault diagnosis has already been the focus of many researchers in the condition monitoring field.

Condition Monitoring (CM) aims to use measured data to predict deterioration and failure of machine components, which has led CM systems to become highly sought after in this industry. A review of different techniques used in CM of wind turbines can be consulted in [5]. This review includes traditional techniques such as vibration signals, acoustic emission, or ultrasonic testing. It has been proven that vibrational analysis is the technique that gives more information about faults in rotating machinery, making vibration sensors widely used in wind turbine applications [6]. The survey [7] gathers an extensive list of approaches based on vibration based condition monitoring for wind turbine gearboxes, ranging from signal processing methods to fault detection methods including Artificial Intelligence (AI) and Machine Learning (ML) based approaches.

Methods based on supervised learning can be used for fault diagnosis after having been trained with known fault training samples. However, only the patterns that are found in the training samples can

be classified. Wrong diagnosis can often occur when dealing with unknown faults, making it hard to use supervised methods effectively. Thus, the application of unsupervised pattern recognition methods, such as clustering, becomes relevant.

This work focuses on the application of fuzzy clustering to fault detection and classification in wind turbines gearboxes using vibrational data. More specifically, the integration of the observer biased framework in the Gath-Geva (GG) clustering algorithm. Fuzzy clustering methods applied to WT gearbox fault detection include the application of the K-means Clustering Method [8] and the unsupervised learning method Kernel C-Means [9]. In comparison to WT gearbox fault detection, applications that involve other machine parts, such as bearings, display a much larger list of different approaches in the literature, ranging from different variations of the Fuzzy C-Means (FCM) algorithm, to the application of the Gustafson-Kessel and Gath-Geva algorithm [10–14]. Motivated by the success of the application of observer biased fuzzy clustering in fault detection and classification for bearing CM, the Gath-Geva with Focal Point (GGFP) clustering algorithm is here applied to new, untested, real world data and compared with other state of the art fuzzy clustering algorithms.

1.2 Contributions

This dissertation aims to achieve the following goals:

- Implement, configure, and deploy the clustering algorithm Gath-Geva in the framework of observer-biased clustering.
- Apply, for the first time, the Gath-Geva with Focal Point algorithm to real world data, with the main purpose to analyze the feature space of vibrational signals from a wind turbine lab test rig.
- Evaluate performance of the Gath-Geva with Focal Point internally, using two different internal validation indexes and establish a comparison to Fuzzy C-Means with Focal Point and its corresponding original versions, using an external validation index.

1.3 Thesis Outline

This dissertation is structured as follows:

- Chapter 2: an overview of clustering is given with focus on the fuzzy algorithms Fuzzy C-Means (FCM) and Gath-Geva.
- Chapter 3: the observer biased framework is presented and exemplified using the FCM.
- Chapter 4: application of the observer biased framework to GG algorithm.
- Chapter 5: results from the application of GGFP to three different data sets (two for verification purposes and one for WT fault detection) are presented
- Chapter 6: main conclusions are drawn from the work developed and future work is proposed

Chapter 2

Fuzzy Clustering

2.1 Clustering Overview

In the data driven world of today, increasingly large amounts of information are stored as data for further analysis and processing. One of the most used tools to analyze data is clustering.

Clustering can be described as the process of separating data points into homogeneous classes or clusters so that points in the same class are as similar as possible and points belonging to separate classes are as different as possible. Alternatively, clustering can be considered a way to compress data, by converting a large number of samples into a smaller number of representative clusters. Different types of similarity measures may be used to identify clusters, depending on the data and the application, the similarity measure controls how the clusters are created [15] . Similarity measures based on distance are the most widely used type of metric. An overview of similarity measure functions commonly used for clustering can be found in [16].

In general terms, clustering can be divided into two major groups: hard clustering and soft clustering, the latter of which includes fuzzy clustering. Within each category there is a diverse number of algorithms, utilizing different approaches for the way partitions are formed.

In hard, or non-fuzzy, clustering data is assigned to clusters such that the degree of membership of each data point to a particular cluster is either 0 or 1. These types of clusters are called crisp clusters. In other words, a given data point belongs to exactly a single cluster. There is an abundance of hard clustering algorithms, among which some of the most well-known are the K-Means algorithm and hierarchical clustering [17].

Utilizing notions of fuzzy sets, data points in soft clustering may belong to more than one cluster. Each data point is then attributed membership values which indicate the likelihood of belonging to different clusters [18]. In this work, the focus is directed to the Fuzzy C-Means algorithm (section 2.2), the Gath-Geva algorithm (section 2.3) and its observer biased variants (chapters 3 and 4).

2.2 Fuzzy C-Means Algorithm

The Fuzzy C-Means Algorithm (FCM) is arguably the most well known fuzzy clustering algorithm. It has been studied extensively and countless variations can be found in the literature. Due to its simplicity and proven effectiveness in distinct applications, the FCM is the perfect candidate for alterations that aim at improving the algorithm. The FCM is here presented as it is the basis of all the clustering algorithms included in this dissertation.

Given an unlabeled multivariate data set, $X = \{x_1, ..., x_n\}$ with $x_j \in \mathbb{R}^d$, where d is the number of features in the data, the goal is to minimize the objective function, J, given by:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m ||x_j - v_i||^2 \tag{2.1}$$

under the constraints:

$$\begin{aligned} &u_{ij} \in [0,1], \qquad i = 1, \ldots, c; j = 1, \ldots, n \\ &\sum_{i=1}^{c} u_{ij} = 1, \qquad j = 1, \ldots, n \end{aligned} \tag{2.2}$$

where c is the number of clusters, n is the number of data points, u_{ij} is the partion matrix that represents the membership of x_j in the i-th cluster, v_i represents the clusters' centers (or prototypes), m is the fuzziness parameter, or fuzzifier, m must have a value larger than 1 and is a user-defined hyper-parameter that controls cluster overlapping, and $||.||$ is the distance norm responsible for the shape of the clusters.

The FCM algorithm utilizes the well known Euclidean norm for the distance metric and achieves the minimization of equation (2.1) by an iterative process where in each step the prototypes and membser-ship values are updated. To find the updating equations, the objective function 2.1 is minimized using Lagrangean Multipliers subject to constraints (2.2):

$$\mathcal{L}_J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m ||x_j - v_i||_A^2 + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1 \right) \tag{2.3}$$

where $||\boldsymbol{x}_j - \boldsymbol{v}_i||_A^2 = (x_j - v_i)^T A (x_j - v_i)$ with A being a $(d \times d)$ positive definite norm-inducing matrix. In the FCM, the distance metric is Euclidean thus A equals the identity matrix.

To obtain the updating expression for the prototypes, u_{ij} is fixed and applying the constraint $\frac{\delta \mathcal{L}}{\delta v_i} = 0$, the resulting expression is:

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.4}$$

To find the updating expression for the partition matrix, $U = [u_{ij}]$, this time v_i is fixed and applying the constraint $\frac{\delta \mathcal{L}}{\delta u_{ij}} = 0$ results in:

$$u_{ij} = [\sum_{k=1}^{c} \left(\frac{||x_j - v_i||^2}{||x_k - v_i||^2} \right)^{\frac{1}{m-1}}]^{-1} \tag{2.5}$$

The steps of the iterative process for the FCM algorithm are gathered in Algorithm 1.

Algorithm 1: FCM Algorithm

Input : Unlabeled multivariate data set: $\boldsymbol{X} \subset \mathbb{R}^d$;
Number of clusters: **c**;
Fuzzifier: **m** > 1;
Sensitivity treshold: ϵ;
Maximum number of iterations: ℓ
Output: Partition matrix: $\boldsymbol{U} = [\boldsymbol{u}_{ij}]$;
Centroids: $\boldsymbol{V} = [\boldsymbol{v}_i]$

Initialize the centroids: $\boldsymbol{v}_i$;
repeat
  for *i = 1* **to** c **do**
    for *j = 1* **to** $|X|$ **do**
      Update $\boldsymbol{u}_{ij}$ with eq. (2.5);
  for *i = 1* **to** c **do**
    Update $\boldsymbol{v}_i$ with eq. (2.4);
until *a termination criterion was met*;

The termination criterion used in algorithm 1, is either when the number of maximum iterations (ℓ) is reached or until algorithm convergence, which occurs when the difference of the objective function between two iterations is smaller than the given sensitivity threshold, ϵ, and is given by the following equation :

$$J_k - J_{k-1} < \epsilon \tag{2.6}$$

where ϵ is an arbitrarily small positive value and k is the current iteration. This stopping criterion will be applied to all the algorithms in this work.

The FCM algorithm is dependent on the initialization of the clusters centers, the typical approach is to initialize the centers randomly which may not always lead to the best results possible. In [19], an alteration to the way centroids are initialized is proposed, the FCM++ variation, the seeding mechanism of the K-Means++ algorithm [20] is employed and is proven to produce efficient results in both the number of iterations needed for convergence and also the quality of the partitions produced. Moving forward, when the FCM algorithm is referenced the initialization is performed using the K-Means++ algorithm.

2.3 Gath-Geva Algorithm

The Gath-Geva (GG) clustering algorithm was originally proposed by Gath and Geva in [21], and is also known as fuzzy maximum-likelihood clustering. In contrast to the FCM algorithm which due to its

Euclidean distance metric imposes the same spherical shape to its clusters, the GG algorithm detects hyper-ellipsoidal clusters with different orientations, sizes, and densities. The GG algorithm aims at minimizing the same objective function as the FCM, equation (2.1), the difference being the adoption of the the Gauss metric:

$$||x_j - v_i||^2 = \frac{|F_i|^{1/2}}{\text{Pr}(i)} \exp[-\frac{1}{2}(x_j - v_i)^T \boldsymbol{F}_i^{-1}(x_j - v_i)] \tag{2.7}$$

where Pr(i) is the priori probability of the i-th cluster, given by:

$$\text{Pr}(i) = \frac{\sum_{j=1}^{n} u_{ij}^m}{\sum_{j=1}^{n}\sum_{\iota=1}^{c} u_{\iota k}^m} \tag{2.8}$$

and $\boldsymbol{F}_i$ and $|\boldsymbol{F}_i|$ are the fuzzy covariance matrix of the i-th cluster (2.9) and its determinant, respectively. Once (2.7) is computed the updates of u_{ij}, and $\boldsymbol{v}_i$ can also be found by (2.5) and (2.4), respectively.

$$F_i = \frac{\sum_{j=1}^{n} u_{ij}^m (x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^{n} u_{ij}^m} \tag{2.9}$$

The GG algorithm follows the same structure as algorithm 1, they differ in the way that the distance metric is computed, and is summarized in algorithm (2).

Algorithm 2: Gath-Geva Algorithm

Input : Unlabeled multivariate data set: $\boldsymbol{X} \subset \mathbb{R}^d$;
Number of clusters: **c**;
Fuzzifier: **m** > 1;
Sensitivity treshold: ϵ;
Maximum number of iterations: ℓ
Output: Partition matrix: $\boldsymbol{U} = [u_{ij}]$;
Centroids: $\boldsymbol{V} = [\boldsymbol{v}_i]$

Initialize the centroids: $\mathbf{v}_i$;
repeat
  for *i = 1* **to** c **do**
    Compute $\boldsymbol{F}_i$ using (2.9);
    for *j = 1* **to** $|X|$ **do**
      Compute the distance plugging $\boldsymbol{F}_i$ in (2.7);
      Update u_{ij} with eq. (2.5);
  for *i = 1* **to** c **do**
    Update $\boldsymbol{v}_i$ with eq. (2.4);
until *a termination criterion was met*;

Similarly to the FCM algorithm, the GG algorithm also possesses the condition of having the clusters' centers initialized. In the FCM the centroids are, standardly, initiliazed randomly. However the GG algorithm is even more dependent on a good initialization of the centroids, the exponential element of the distance function makes the algorithm seek an optimum in a narrower local region making it

converge to a local optimum. A random initialization will, most likely, incur in poor results, as such one has to be conscient of the way centroids are initialized. For many applications using the output of the FCM algorithm in itself to initialize the GG algorithm is a sensible choice. [21]

2.4 Evaluation of clustering results: Validity Measures

Due to the unsupervised nature of clustering, it becomes necessary to find a way to determine if the partition produced is a good representation of the inherent structure found in a given data. To achieve this, there must be systematic measurements that can be employed regardless of the algorithm or the application. These measurements are called Clustering Validity Indexes (CVI) and there is wide range of CVIs that use different criteria for evaluation. The calculation of these indexes is performed in such a way that they can be compared when evaluating different partitions resulting from various algorithms as well as partitions generated by the same algorithm with different parameters.

Clustering validity indexes can be generalized into two categories: internal and external indexes. Internal indexes are independent of any information about the real structure of the data (labels), and metrics such as compactness, level of separation, similarity of the clusters are some examples of criteria used in internal indexes. In contrast, external indexes are used when there is knowledge about the data labels and this information is used in the evaluation process.

There is a wide variety of indexes in the literature that have been extensively explored [22]. In this work, two internal indexes and two external index are investigated. The applied internal indexes are the Xie-Beni (XB) index [23] and the Kim-Lee (KL) index [24], these validity measures were chosen considering the two clustering algorithms studied are the FCM and the GG algorithms (and its variants). As for the internal validity measure, as the data used for testing (in chapter 5) is labeled, the Adjusted Rand Index was the measure chosen.

2.4.1 Xie-Beni Index

Several internal validation measures in relation to the FCM algorithm have been studied in the literature and the XB index has proven to both perform well and be quite reliable [25], making the XB index a good choice to evaluate partitions produced by the FCM. The Xie Beni index focuses on the identification of compact and well-separated clusters, for computational purposes the inverse of the XB index was applied and is given by:

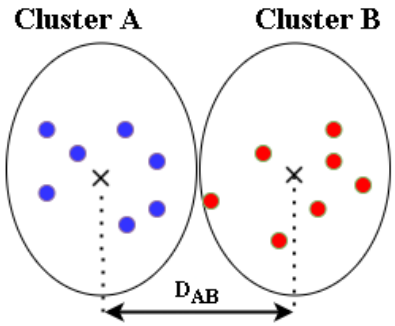
$$\boldsymbol{XB}^{-1} = \frac{n\, min_{i\neq j}||\boldsymbol{v}_i - \boldsymbol{v}_j||^2}{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m ||\boldsymbol{x}_j - \boldsymbol{v}_i||^2} \tag{2.10}$$

This index can be seen as the ratio between degrees of intra-cluster distance and inter-cluster distance where the numerator represents the minimal separation between fuzzy clusters and the denominator is the sum of the compactness of each fuzzy cluster. The optimal partition is then obtained by maximizing equation 2.10.

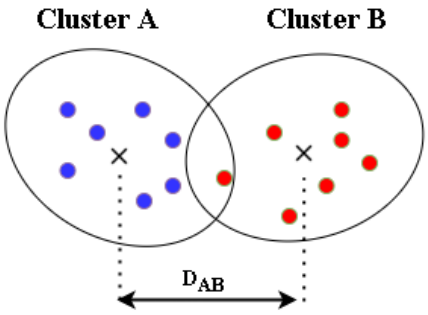
2.4.2 Kim-Lee Index

As mentioned in section (2.3), the shapes and sizes of the clusters in the GG algorithm differs from the spherical and same sized clusters of the FCM. This limits the applicability of indexes, like the Xie-Beni, that rely solely on the clusters' centroid distance to evaluate cluster separation. Since the GG algorithm involves the Gaussian metric, these type of indexes cannot differentiate two different partitions with the same centroid distance and different orientations. This is shown in figure 2.1, in spite of partition 1 resulting in a better division of the data, this cannot be reflected in an index that only relies on centroid distances.

To tackle this issue in [24], an alternative cluster validation index based on relative degree of sharing was applied to the Gustafson-Kessel (GK) algorithm (which also generates hyper-ellipsoidal clusters) and revealed promising results. In view of this similarity in the cluster shapes between the GG and GK algorithms, this index was here applied to the GG algorithm.



(a) Partition 1, $(U^{(1)}, V^{(1)})$

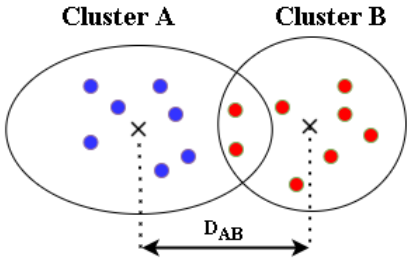


(b) Partition 2, $(U^{(2)}, V^{(2)})$

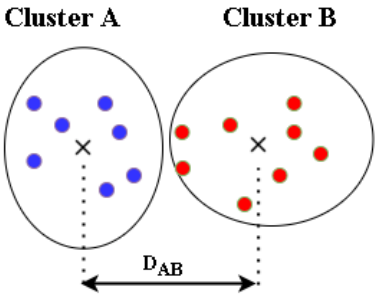
Figure 2.1: Two different partitions of the same data which have the same centroid distance and different cluster orientation, reproduced from [24]

The KL index's goal is to calculate the average overlap between clusters using the definition of the relative degree of sharing between fuzzy clusters. The higher the membership values of a data point to a pair of clusters, the higher the relative degree of sharing between the clusters, which is an indication of cluster overlap. Figure 2.2 shows two different partitions of the same data with the same distance between the clusters' centroids, here partition 2 provides a better representation of structure in the data, again, this also cannot be reflected in centroid distance reliant indexes. To take into account overlapped data points a weighing parameter is introduced and highly overlapped data points are given a bigger

weight over data that are classified clearly.



(a) Partition 1, $(U^{(1)}, V^{(1)})$



(b) Partition 2, $(U^{(2)}, V^{(2)})$

Figure 2.2: Two different partitions of the same data which have the same centroid distance , reproduced from [24]

The relative degree of sharing between two fuzzy clusters is then defined by the weighted sum of the relative degrees of sharing at each data point x_k:

$$\boldsymbol{S(C_i, C_j)} = \sum_{k=1}^{n} [c \cdot [u_{ik} \wedge u_{jk}] h(x_k)] \tag{2.11}$$

where, the fuzzy AND operator [26] is employed : $u_{ik} \wedge u_{jk} = min(u_{ik}, u_{jk})$ and the entropy of data points is used as weighing parameter: $h(x_k) = -\sum_{i=1}^{c} u_{ik} log\, u_{ik}$. The partition entropy attains its maximum value when all data points are ambiguously assigned to clusters, this occurs when the membership degrees of all data points to all clusters are $\frac{1}{c}$, which means that all data points have the same membership value to all clusters and the entropy takes a value near 1. In the case where all data points are clearly divided into all clusters and the amount of uncertainty is minimal, the partition entropy has a value close to 0.

Using the entropy as weighing parameter allows vague (unclearly classified) points to have greater impact, meaning that partitions with highly overlapped data are penalized.

This index has as its only input the partition matrix, u_{ij}, and considers a partitioning optimal when the degree of overlap between clusters is minimal. The KL index is then defined as the average relative degree of sharing for all possible cluster pairs:

$$\boldsymbol{KL} = \frac{2}{c\,(c-1)} \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \sum_{k=1}^{n} [c \cdot [u_{ik} \wedge u_{jk}] h(x_k)] \tag{2.12}$$

The KL index can measure the ambiguity of a certain partitioning as well as the geometrical property

of overlap between clusters. It possesses the advantage over other indexes of calculating the separation between clusters without making use of inter-center distances.

2.4.3 Rand Index and Adjusted Rand Index

For external validation, cluster validity measures Rand Index (RI) and Adjusted Rand Index (ARI), [27], are employed . These indexes measure the similarity of two partitions of a data set $X = (x_1, ..., x_n)$, the ground truth partition, $\mathcal{G}$, and a hypothesis partition $\mathcal{H}$ generated by a clustering algorithm and is given by:

$$\text{RI}(\mathcal{G}, \mathcal{H}) = \frac{a+b}{n_t} \tag{2.13}$$

where a is the number of pairs of elements in X that are clustered together in both partitions, b is the number of elements that are separated in both partitions, n_T is the total number of possible pairs $n_t = \binom{n}{2} = \frac{n(n-1)}{2}$. The RI index takes values in the of 0 to 1, where 0 means the two partitions do not agree on any pair of points and 1 indicating that the partitions are exactly the same. However, the RI of random partitions is not constant and two random partitions can produce high RI values even if they don't represent the structure in the data. In order to deal with these drawbacks, this index is corrected for chance resulting in the more reliable Adjusted Rand Index:

$$\text{ARI}(\mathcal{G}, \mathcal{H}) = \frac{\sum_{i,j}\binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_{\cdot j}}{2} / \binom{n}{2}]}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_{\cdot j}}{2} / \binom{n}{2}]} \tag{2.14}$$

where n_{ij} is the number of elements that cluster together in the subsets $\mathcal{G}_i$ and $\mathcal{H}_j$; n_i,n_j are the number of elements in $\mathcal{G}_i$ and $\mathcal{H}_j$, respectively. It also attains its maximum value at 1 when both partitions completely agree. The ARI is the main external validation index applied throughout this work.

Chapter 3

Observer Biased Fuzzy Clustering and its Application to Fault Detection

3.1 The Observer Biased Clustering Algorithm

The observer biased variant applied to fuzzy clustering was first introduced in [28]. In theory, this approach can be applied to any clustering algorithm that seeks to optimize an objective function. The purpose of the observer biased technique is to give a clustering algorithm the ability to change the position from which the data is observed, and thus allowing for the possibility of viewing different perspectives of the data.

The inspiration for this algorithm surged from a metaphor found in daily life depicted in [29]. It uses the analogy of human perception of objects depending on the point of observation; the closer the location of the observer, the more distinct each object becomes. Conversely, the further away the observer stands from the objects, the less detail is visualized and the group objects begins to merge into a single element. This is akin to the effect of the zoom property found in an optical lens which can cover a wide array of perspectives, ranging from fine details when zoomed in and the ”bigger picture” when zoomed out.

The approach follows a statistics concept known as shrinkage. Usually, shrinkage is applied with the goal of improving an unbiased estimate by adding a regularization term. Clustering algorithms based on shrinkage techniques have been proposed and have proven through experimental results to have advantages in comparison to unbiased traditional algorithms [30, 31]. Although the observer biased approach is based on shrinkage, it differs from previously proposed algorithms due to the regularization coefficient being different from the ones previously suggested, seeing that it originated from a distinct motivation.

The observer approach can be related to fuzzy hierarchical clustering [32]. For instance, by gradually varying the position of the observer, imagining that the starting point is far away from the data and then gradually bringing it closer, a hierarchical cluster tree can be formed with greater efficiency than the available techniques according to [29]. The most significant advantages in relation to hierarchical clustering are the possibility of exploring selected regions of the data in detail and its ability to reassign

data points to clusters when the number of clusters is changed, something that is impossible in traditional hierarchical clustering.

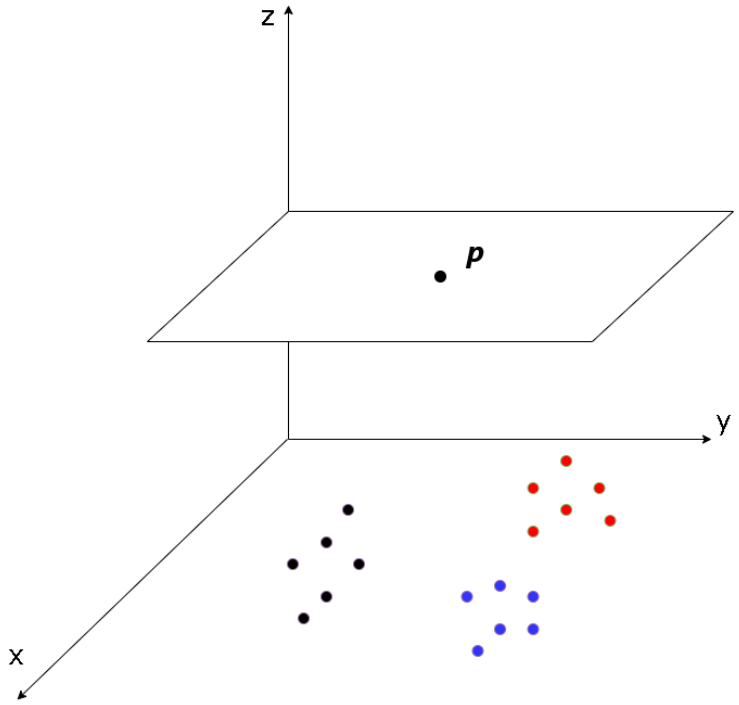


Figure 3.1: Data set in $\mathbb{R}^2$ and focal point (FP) in a higher dimension, $\mathbb{R}^3$

The placement of the observer becomes a matter of significance and the pertinent question that emerges is where the observer should be located. The location of the observer is not constrained, it can be placed anywhere in the data space or even in a different dimensional space. To fulfill the previously mentioned metaphor, the observer is placed in a higher dimensional space than the data, which also has the benefit of easily being able to distinguish the observer point from data points. Figure 3.1 illustrates this, in the case where the data belongs to the two dimensional space, $\mathbb{R}^2$, and focal point is placed in a higher dimension of the data, the tri-dimensional space $\mathbb{R}^3$.

All of the above allows for the user to obtain different reasonable clusters, for a given data, depending on the position of the observer. The term ”reasonable cluster” is used as described in [33]: A reasonable cluster belongs to a certain partition that possesses “reasonably good similarity groups”, validated by a given internal cluster validity index, which is not to be mistaken for the term: ”meaningful cluster” , a partition that is recognized by a domain expert to be representative of the data structure.

In summary, this approach allows for an intuitive way to control the process of cluster formation, by incorporating the knowledge of domain experts which can select the most appropriate level of granularity and the relevance of regions of the data space.

The observer biased framework has already been successfully applied and tested in fuzzy clustering algorithms , namely the Fuzzy C-Means with a focal point (FCMFP) algorithm and the Gustafson-Kessel with a focal point (GKFP) algorithm. In order to achieve a deeper understanding of the observer based paradigm, in the following sections the FCM algorithm exemplifies the application of the observer biased

framework to a clustering algorithm.

3.2 Fuzzy C-means with Focal Point Algorithm

In this section, the observer biased variant is formulated and expanded in detail, using the FCM algorithm, as proposed in [29]. A term that depends on the focal point ($\boldsymbol{p}$) is added to the FCM objective function (2.1):

$$J_{FCMFP} = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}||x_j - v_i||^2 + \zeta \sum_{i=1}^{c} ||\boldsymbol{p} - v_i||^2 \tag{3.1}$$

where ζ is a user-defined regularization parameter. All the other terms continue having the same meaning as in section 2.2 including the Euclidean distance metric and the constraints (2.2).

The regularization coefficient ζ imposes a balance between the unbiased algorithm (it degenerates to the original FCM objective function when $\zeta = 0$) and a biased estimate when $\zeta > 0$.

The regularization term in equation 3.1 is zero if $\zeta = 0$ or in the case that all prototypes v_i are equal to $\boldsymbol{p}$. If the focal point is at a large enough distance from the data, a prototype that is very near the focal point will have values nearing zero in the partition matrix, u_{ij}, this can be deduced by observing equation 2.5, when the distance $||\boldsymbol{x}_j - \boldsymbol{v}_i||^2$ increases, u_{ij} will tend to zero. These prototypes are considered empty and can be neglected.

In order to calculate the updating equations for the prototypes $V = [v_i]$ and the membership values $U = [u_{ij}]$, similarly to FCM algorithm, optimization of the objective function is performed using Lagrange Multipliers and the following equation is obtained:

$$\mathcal{L}_{J_{FCMFP}} = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}||x_j - v_i||_A^2 + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1 \right) + \zeta \sum_{i=1}^{c} ||\boldsymbol{p} - v_i||_A^2$$

where $||x_j - v_i||_A^2 = (x_j - v_i)^T A (x_j - v_i)$ and $||\boldsymbol{p} - v_i||_A^2 = (\boldsymbol{p} - v_i)^T A (\boldsymbol{p} - v_i)$, with A being a $(d \times d)$ positive definite norm-inducing matrix. If the distance metric is Euclidean then A equals the identity matrix.

By fixing either one of the parameters u_{ij} and v_i in equation (3.1) it is possible to obtain the optimizing expression for the other. The calculation for each parameter is shown in the following subsections.

3.2.1 Estimation of Partition Matrix

To estimate the fuzzy partition matrix $U = [u_{ij}] \in [0,1]^{c \times n}$ the objective function 3.2 is solved in order to u_{ij}, by applying the constraint: $\frac{\delta J_{\mathcal{L}}}{\delta u_{ij}} = 0$. Noticing that the terms that have u_{ij} do not depend on $\boldsymbol{p}$, the resulting expression will be equal to the non-biased one, derived for the FCM algorithm, (2.5), in section 2.2.

3.2.2 Estimation of Cluster Centroids

For obtaining the estimates of centroids $V = [v_i] \in \mathbb{R}^{c \times d}$ the objective function (3.2) is solved in order to v_i by applying the constraint $\frac{\delta J_{\mathcal{L}}}{\delta v_l} = 0$:

$$\frac{\delta J_{\mathcal{L}}}{\delta v_i} = -2 \sum_{j=1}^{n} u_{ij}^m A(\boldsymbol{x}_j - \boldsymbol{v}_i) - 2\zeta A(\boldsymbol{p} - v_i) = 0 \tag{3.2}$$

Simplifying and rearranging in order to v_i:

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n} u_{lj}^m x_j + \zeta \boldsymbol{p}}{\sum_{j=1}^{n} u_{lj}^m + \zeta} \tag{3.3}$$

The equation to update the prototypes is then given by equation (3.3). Analyzing the previous equation it can be seen that ζ determines the degree of attraction of the prototypes to the focal point, when ζ is large the prototype will tend to $\boldsymbol{p}$ and if zeta is arbitrarily small the dependency on $\boldsymbol{p}$ will diminish. Therefore, decreasing the value of ζ results in a finer detailed data observation, which is to say that more clusters are generated, due to less prototypes being attracted to $\boldsymbol{p}$; while increasing ζ will produce the opposite reaction, the observation will have a smaller number of clusters.

3.2.3 The FCMFP algorithm with a focal point in a higher dimension of the data space

Once the focal point and the regularization coefficient $\zeta \geq 0$ are defined, the clusters' prototypes are initialized. After initialization, the algorithm follows an iterative process similar to FCM with consequent updates of the partition matrix and prototypes until the termination criteria is reached. The options for the termination criterion are essentially the same as in the FCM algorithm; the algorithm will stop if there are no significant changes in the partition matrix or prototypes between iterations, causing no improvement in the cost function or if the number of maximum iterations is reached.

As previously mentioned the point of observation is not restricted to the original input space $\mathbb{R}^d$, in fact, its more interesting application is to be placed in a higher dimension w where $w \geq d$. Placing the observer in a higher dimension requires two extra steps, one before the updating of the partition matrix and prototypes and the other after the termination criterion has been reached. The first extra step is the extension of the data and prototypes to the focal point dimensional space. This is easily achieved by adding $(w - d)$ null coordinates for each data point. The final step is projecting the estimated prototypes back to its original feature space, using the intersection of the lines defined by $\boldsymbol{p}$ and cluster v_i with the original data space, this step is illustrated in figure 3.2, where the prototypes C_i estimated in $\mathbb{R}^3$ are projected back to its original dimensional space $\mathbb{R}^2$.

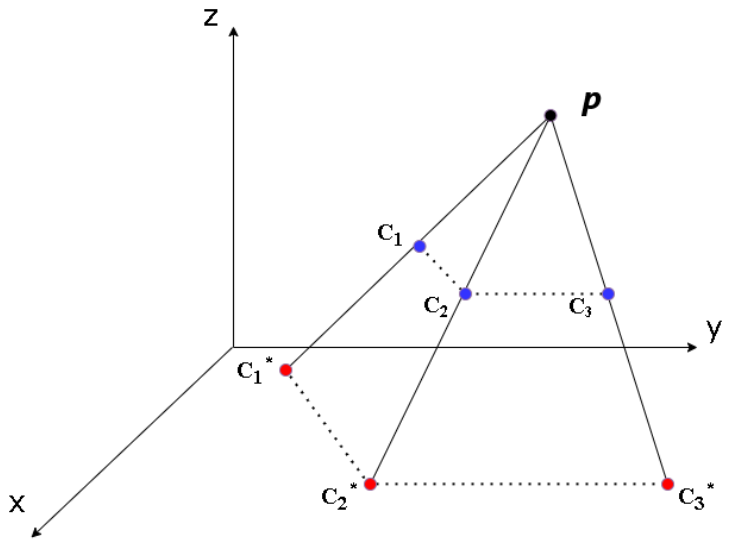


Figure 3.2: Projection of prototypes in $\mathbb{R}^3$ back to original dimension $\mathbb{R}^2$, where C_i corresponds to the computed prototypes and C_i^* to the projected prototypes.

Summarizing the previously described steps in algorithmic form:

Algorithm 3: FCMFP - The FCM clustering algorithm with a focal point in a higher dimension of the data space

Input : Unlabeled multivariate data set: $\boldsymbol{X} \subset \mathbb{R}^d$;
Number of clusters: **c**;
Fuzzifier: **m** > 1;
Focal point: $\boldsymbol{p} \subset \mathbb{R}^w$, $w > d$
Regularization coefficient: ζ
Output: Partition matrix: $\boldsymbol{U} = [u_{ij}]$;
Prototypes: $\boldsymbol{V} = [\boldsymbol{v}_i]$

Initialize the clusters' prototypes ;
Extend $\boldsymbol{X}$ and $\boldsymbol{V}$ to $\mathbb{R}^w$ by adding $(w - d)$ null coordinates per datapoint;
repeat
  for *i = 1* **to** c **do**
    for *j = 1* **to** $|X|$ **do**
      Update u_{ij} using (2.5);
  for *i = 1* **to** c **do**
    update $\mathbf{v}_i$ using (2.4);
until *a termination criterion was met*;
Project prototypes to the original feature space $\mathbb{R}^d$

3.2.4 The Iterated FCM Algorithm with a Focal Point

The number of clusters, c, is considered one of the most critical parameters of the FCM algorithm. Only in the case of c being equal to the number of classes in the data is there a possibility that the outcome of the clustering process is in agreement with the existent data structure.

If there is no prior knowledge about the optimal number of clusters, usually, the selection of the

number of clusters is performed using cluster validity indexes. A common way to ascertain the optimal number of clusters is to compare multiple outcomes of the same algorithm with different initializations for a range of values of c. The value c that optimizes the validity measure is then selected. In [28] an iterative process to find the number of reasonable clusters was proposed for the FCMFP. It consists of running the FCMFP algorithm while increasing ζ in each iteration. It has as its inputs an overestimation of the cluster number (c_{max}), a focal point $\boldsymbol{p}$, the increment of ζ for every iteration ($\Delta\zeta$) and the maximum value for ζ (ζ_{max}). In each iteration the number of candidate clusters is determined and a validity measure is calculated. In increasing ζ some clusters will be attracted to $\boldsymbol{p}$, these clusters become neglectable clusters. A cluster is considered neglectable when it does not possess any typical points belonging to the data set. The concept of typicality is used from [34] , a data point x_j is considered a typical point of a cluster i, if and only if, the partition matrix for that cluster satisfies the inequality: $u_{ij} > u_{kj}, \quad \forall \quad k \neq i$, which is to say that, if for all data points a given cluster does not have a single point with maximum membership value relative to the other clusters, then it is considered negligible. This iterative process is summarized in Algorithm 4.

Algorithm 4: Iterative FMCFP

input : $\boldsymbol{\Delta\zeta}$, $\boldsymbol{\zeta_{max}}$, $\boldsymbol{C_{max}}$
output: The partition that optimizes the internal validity measure
repeat
  1) Run the GGFP algorithm
  2) Remove negligible clusters using the definition of typicality
  3) Calculate internal validity of clusters
  4) Update $\zeta = \zeta + \Delta\zeta$
until $\zeta = \zeta_{max}$;

3.3 Application to Fault Detection

Considering one of the goals of this dissertation is to apply an observer biased clustering algorithm to a fault detection problem it is of interest to verify the work done using observer biased algorithms in the field of fault detection. In [35], the FCM with a Focal Point (FCMFP) was applied to bearing condition monitoring, in [36] the Gustafson-Kessel with a Focal Point (GKFP) was studied for a different data set in the field of bearing fault diagnosis. A comparison of Fuzzy C-Means (FCM), the Gustafson-Kessel (GK) algorithm, FN-DBSCAN, and FCMFP is performed in [37] also for bearing fault diagnosis.

The results from the above-mentioned work revealed that applying the observer bias framework to clustering enhanced the performance of the algorithms although this was not the main objective, it is a side effect of the inherent shrinkage encompassed in the algorithm. Moreover, it was proven that the observer metaphor can be applied successfully, providing different views of the data and allowing to shift the focus of a clustering analysis depending on the end goal.

Chapter 4

Gath Geva Algorithm with Focal Point

The empirical evidence obtained from introducing a focal point in fuzzy C-means (FCM) [29, 35] and to the Gustafson-Kessel (GK) algorithm [36], served as motivation to further explore this idea in the realm of other fuzzy clustering algorithms. In this chapter, the Gath-Geva algorthm with a Focal Point (GGFP) is introduced along with a detailed formulation and derivation of the algorithm. Similarly to the FCMFP, in GGFP the focal point ($\boldsymbol{p}$) can also be located in a higher dimension in relation to the data, if the data dimensional space is $\mathbb{R}^d$ then $\boldsymbol{p}$ may belong to $\mathbb{R}^{d+1}$, much like in figure 3.1 where this idea is illustrated. In the case of the feature space being equally relevant, the focal point can be placed at the barycenter of the data, in placing $\boldsymbol{p}$ in a different position the focus of the analysis will shift to that particular location.

4.1 Formulation

The GGFP shares the same foundational idea as the FCMFP in incorporating a focal point to an otherwise unbiased algorithm. To achive this, here the GG objective function is modified to include an additional term which depends on $\boldsymbol{p}$. The GGFP objective function is then given by:

$$\begin{aligned} J_{GGFP} = &\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \frac{|M_i|^{1/2}}{\mathsf{Pr}(i)} \exp[-\frac{1}{2}(x_j - v_i)M_i^{-1}(x_j - v_i)^{\intercal}] \\ &+ \zeta \sum_{i=1}^{c} \frac{|M_i|^{1/2}}{\mathsf{Pr}(i)} \exp[-\frac{1}{2}(\boldsymbol{p} - \boldsymbol{v}_i)M_i^{-1}(\boldsymbol{p} - \boldsymbol{v}_i)^{\intercal}] \end{aligned} \tag{4.1}$$

where, Pr(i) is given by (2.8) and under the constraints (2.2).

The clustering problem can then be formulated as:

$$\mathbf{U}^*, \mathbf{V}^*, \{\boldsymbol{M}_i^*\}_{i=1}^{c} = \arg \min_{\mathbf{U},\mathbf{V},\{\boldsymbol{M}_i\}_{i=1}^{c}} J_{GGFP}(\boldsymbol{x}; \mathbf{U}, \mathbf{V}, \{\boldsymbol{M}_i\}_{i=1}^{c}) \text{ subject to (2.2)} \tag{4.2}$$

Which is to say that the problem lies in finding the optimal values for the partiton matrix, prototypes and fuzzy covariance matrix, $\mathbf{U}, \mathbf{V}, \boldsymbol{M}_i$, that minimize J for a given data set $X = [x_j]$.

4.2 Derivation

Solving (4.2) can be accomplished using Lagrange multipliers λ_j and minimizing:

$$\begin{aligned}\mathcal{L}_{GGFP} = &\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\frac{|M_i|^{1/2}}{\mathsf{Pr}(i)}\exp[-\frac{1}{2}(x_j-\boldsymbol{v}_i)M_i^{-1}(x_j-v_i)^{\intercal}] \\ &+\sum_{j=1}^{n}\lambda_j\left(\sum_{i=1}^{c}u_{ij}-1\right) \\ &+\zeta\sum_{i=1}^{c}\frac{|M_i|^{1/2}}{\mathsf{Pr}(i)}\exp[-\frac{1}{2}(\boldsymbol{p}-\boldsymbol{v}_i)M_i^{-1}(\boldsymbol{p}-v_i)^{\intercal}]\end{aligned} \tag{4.3}$$

Taking the logarithm of (2.7):

$$\log||\boldsymbol{x}_j-\boldsymbol{v}_i||^2 \propto \log(|\boldsymbol{M}_i|)-(\boldsymbol{x}_j-\boldsymbol{v}_i)^{\intercal}\boldsymbol{M}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{v}_i) \tag{4.4}$$

thus minimizing (4.3) is to minimize:

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}[\log|\boldsymbol{M}_i|-(\boldsymbol{x}_j-\boldsymbol{v}_i)^{\intercal}\boldsymbol{M}_i^{-1}(\boldsymbol{x}_j-\boldsymbol{v}_i)] \\ &+\sum_{j=1}^{n}\lambda_j\left(\sum_{i=1}^{c}u_{ij}-1\right) \\ &+\zeta[\sum_{i=1}^{c}\log|M_i|-(\boldsymbol{p}-v_i)^{\intercal}M_i^{-1}(\boldsymbol{p}-v_i)]\end{aligned} \tag{4.5}$$

where M_i is the to be estimated adaptive matrix from the metric (2.7), for the case of the GG algorithm corresponds to the fuzzy covariance matrix (2.9).

The parameter updating expressions are then computed from the respective necessary conditions which are presented in the following subsections.

4.2.1 Estimation of the Partition Matrix

To calculate the fuzzy partition matrix $U=[u_{ij}]\in[0,1]^{c\times n}$ the objective function 4.3 is solved in order to u_{ij}, by applying the constraint: $\frac{\delta J_{\mathcal{L}}}{\delta u_{ij}}=0$. Noticing that the terms that have u_{ij} do not depend on $\boldsymbol{p}$, the resulting expression will be equal to the non-biased one equation (2.5), derived for the GG algorithm, where the distance metric is (2.7).

4.2.2 Estimation of the Cluster Centroids

For obtaining the estimates of centroids $V=[v_i]\in\mathbb{R}^{c\times d}$ the objective function (4.3) is solved in order to v_i by applying the constraint $\frac{\delta\mathcal{L}}{\delta v_l}=0$. Considering that for any symmetric matrix A and any compatible

vector x: $\frac{\partial x^\intercal Ax}{\partial x} = 2Ax$

$$\frac{\partial \mathcal{L}}{\partial v_i} = \sum_{j=1}^{n} u_{lj}^m M_i^{-1}(x_j - \boldsymbol{v}_i) + \zeta M_i^{-1}(\boldsymbol{p} - v_i) = 0 \tag{4.6}$$

multiplying both terms by M_i and rearranging yields :

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j + \zeta \boldsymbol{p}}{\sum_{j=1}^{n} u_{ij}^m + \zeta} \tag{4.7}$$

It is worth noticing that the above updating expression for the centroids is equal to the one found for the FCMFP algorithm in section and it degenerates into the GG (and FCM) updating expression (2.4) if $\zeta = 0$.

4.2.3 Estimating the Matrix of the Distance Metric

An updating expression for the matrix $\boldsymbol{M}_i$ of the distance metric (4.4) can be obtained from $\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}_i^{-1}} = 0$ as follows. Keeping in mind that for a square non-singular matrix $\boldsymbol{A}$ and any compatible vector $\boldsymbol{x}$, $\frac{\partial |\boldsymbol{A}|}{\partial \boldsymbol{A}^{-1}} = -|\boldsymbol{A}|\boldsymbol{A}$, $\frac{\partial \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{A}} = \boldsymbol{x}\boldsymbol{x}^\intercal$, and from (4.5):

$$\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial \boldsymbol{M}_i^{-1}} = 0 \\
&\sum_{j=1}^{n} u_{ij}^m [(\boldsymbol{x}_j - \boldsymbol{v}_i)(\boldsymbol{x}_j - \boldsymbol{v}_i)^\intercal) - \boldsymbol{M}_i] + \zeta[(\boldsymbol{p} - \boldsymbol{v}_i)(\boldsymbol{p} - \boldsymbol{v}_i)^\intercal - \boldsymbol{M}_i] = 0 \\
&\boldsymbol{M}_i = \frac{\sum_{j=1}^{n} u_{ij}^m (\boldsymbol{x}_j - \boldsymbol{v}_i)(\boldsymbol{x}_j - \boldsymbol{v}_i)^\intercal + \zeta(\boldsymbol{p} - \boldsymbol{v}_i)(\boldsymbol{p} - \boldsymbol{v}_i)^\intercal}{\sum_{j=1}^{n} u_{ij}^m + \zeta}
\end{aligned} \tag{4.8}$$

Thus, $\boldsymbol{M}_i$ can be viewed as the fuzzy covariance matrix regularized by $\boldsymbol{p}$. For $\zeta = 0$ it turns into the fuzzy covariance matrix (2.9).

4.2.4 Regularization of the Estimation of the Covariance Matrices

The Gath-Geva algorithm is a clustering algorithms that relies on a fuzzy covariance matrix for each cluster. The covariance matrix allows for the generation of ellipsoidal clusters with independent orientations and sizes, however their estimation during the algorithm execution is often likely to incur numerical problems, especially if there are a large number of features and the values of the matrix are relatively small due to matrix operations such as the inverse or multiplication .

In order to attenuate these problems, in [38] a Bayesian framework for the estimation of the covariance matrices is proposed. This method is a remarkably simple but effective regularizer, in comparison to existing regularizers. The regularizer in its simpler form can be reduced to the following equation:

$$\boldsymbol{M}_i^{\mathcal{R}} = \alpha \boldsymbol{I} + \boldsymbol{M}_i \tag{4.9}$$

where α is a user-defined positive value and I the identity matrix.

4.3 The Gath-Geva Algorithm with a Focal Point in a Higher Dimension of the Data Space

The GGFP, where the focal point is located in a dimension higher than that of the data space, is summarized in Algorithm 5.

The algorithm follows the same procedure as of the FCMFP algorithm, although here the different distance metric requires the calculation of the fuzzy covariance matrix and its regularization.

Algorithm 5: GGFP - The Gath-Geva clustering algorithm with a focal point in a higher dimension of the data space

Input : Unlabeled multivariate data set: $\boldsymbol{X} \subset \mathbb{R}^d$;
Number of clusters: **c**;
Fuzzifier: **m** > 1;
Focal point: $\boldsymbol{p} \subset \mathbb{R}^w$, $w > d$
Regularization coefficient: $\boldsymbol{\zeta}$
Fuzzy Covariance matrix regularization: $\boldsymbol{\alpha}$
Output: Partition matrix: $\boldsymbol{U} = [u_{ij}]$;
Prototypes: $\boldsymbol{V} = [\boldsymbol{v}_i]$;
Fuzzy covariance matrix : $\boldsymbol{M}_i$

Initialize the clusters' prototypes ;
Extend $\boldsymbol{X}$ and $\boldsymbol{V}$ to $\mathbb{R}^w$ by adding $(w-d)$ null coordinates per data point;
repeat
  for *i = 1* **to** c **do**
    Compute $\boldsymbol{M}_i$ using (4.8);
    Apply regularization $\boldsymbol{M}_i^{\mathcal{R}} \leftarrow \alpha \boldsymbol{I} + \boldsymbol{M}_i$ (4.9) ;
    for *j = 1* **to** $|X|$ **do**
      Compute the distance plugging $\boldsymbol{M}_i^{\mathcal{R}}$ in (2.7);
      Update u_{ij} using (2.5);
  for *i = 1* **to** c **do**
    update v_i using (4.7);
until *a termination criterion was met*;
Project prototypes to the original feature space $\mathbb{R}^d$

As mentioned before for the Gath Geva algorithm, the initialization is an absolutely critical step for this algorithm. In this work, usually the FCM++ algorithm is used to initialize the prototypes and if the FCM++ does not provide acceptable results then the FCMFP algorithm is considered for initialization.

4.4 The Iterated Gath-Geva Algorithm with a Focal Point

The alternative iterative process to the empirical selection of ζ, used for the FCMFP (section 3.2.4) is here applied to the GGFP algorithm. For each value of ζ the resulting clustering partition is evaluated by an internal cluster validation index, due to its hyper-ellipsoidal clusters, both the XB and KL indexes

are employed in order to compare them. A cluster without typical points is viewed as irrelevant and is eliminated. A new partition matrix U is then calculated together with the centroids V and M_i.

Algorithm 6: Iterative GGFP

input : $\boldsymbol{\Delta\zeta}$, $\boldsymbol{\zeta_{max}}$, $\boldsymbol{C_{max}}$
output: The partition that optimizes the internal validity measure
repeat
  1) Run the GGFP algorithm
  2) Remove negligible clusters using the definition of typicality
  3) Calculate internal validity of clusters
  4) Update $\zeta = \zeta + \Delta\zeta$
until $\zeta = \zeta_{max}$;

Chapter 5

Results

The application of the GGFP algorithm to real world, experimentally collected data is presented in this chapter. Three data sets were tested using the GGFP algorithm: the well known Iris plants data set [39], the bearing condition data set from [35] and finally the experimental data collected from a wind turbine gearbox in a laboratory setting.

The first two data sets serve, mainly, as verification for the implementation of the GGFP algorithm, detailed in chapter 4. The Iris data set was chosen due to being, perhaps, the most well known data set in clustering and regularly used for bench-marking, while the bearing condition monitoring data set is tested here due to the extreme relevance of its application in the field of fault detection and multi-fault classification to the topic of the wind turbine data set explored in this dissertation. Furthermore, having already been studied in depth, it also serves as common ground for comparison with the work developed for the FCMFP algorithm.

This chapter is divided into three sections, one for each data set, where there is a presentation of the analysis of the clustering results obtained. In the case of the bearing condition monitoring and the wind turbine data sets both the fault detection and multi-fault classification cases are analyzed. This includes experimental verification on the merits of employing the GGFP algorithm and its iterated version, in providing an intuitive way to control the cluster formation process, and therefore allowing the user to select a suitable level of granularity while searching for meaningful clusters in a specific region of the feature space.

The observer biased algorithms, FCMFP and GGFP, are compared against each other as well as to their corresponding unbiased versions (FCM and GG). This comparison measures the quality of the resulting partitions using the external validity measure Adjusted Rand Index, for this, boxplots of 30 independent runs of the algorithms are obtained (outlier points in the boxplots are represented by circles). Afterwards, there is also the verification for statistical significant differences between the obtained indexes, this is done using non-parametric statistical tests; the Wilcoxon signed-rank test for comparison of pairs of samples and the Friedman test to compare more than two sets of samples [40].

5.1 Iris Dataset

The iris data set is quite possibly one of the best known databases found in the literature [39]. The data set contains 3 classes of 50 instances each, where the classes refer to three different types of iris plants (Setosa, Versicolour and Virginica). One of the classes is linearly separable from the others, while the other two cannot be separated linearly. The four features in this data set are the petal length, petal width, sepal length and sepal length measured in centimeters. Containing 150 samples, each with 4 features, the data set is a list of 150 x 4. This data set is quite simple and is very frequently used for bench-marking purposes.

5.1.1 Clustering results

The analysis starts by applying the iterative algorithm (6) to the iris data set in order to find the reasonable number of clusters, which are validated by using the two different internal validity indexes from section 2.4, the XB and the KL indexes. Figure 5.1 and 5.2 present the results obtained.

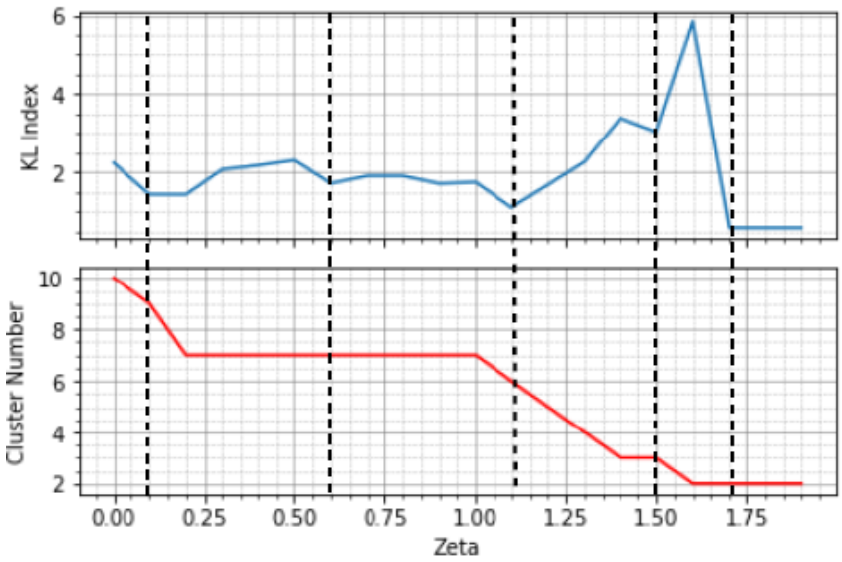


Figure 5.1: Internal validity index KL and cluster number as function of ζ (Objective is to minimize KL index)

Observing figures 5.1 and 5.2, there are two graphs per figure that share the same horizontal axis (they are both in function of ζ), in the upper graph the value calculated for the respective validity index is shown, while the bottom graph shows the evolution of the number of clusters as ζ is progressively increased. By tracing a vertical line through both graphs in the location that the validity index is at a local minimum value (for the KL index) or at a local maximum value (for the XB index) the number of clusters that produce that optimal value is obtained.

For the Iris dataset using the GGFP algorithm the KL index gives as reasonable numbers of clusters, $c = 2, 3, 6, 7$ and 9 and the XB index reveals as reasonable $c = 2$ and 3. Interestingly, from the reasonable clusters both indexes evaluate $c = 2$ as the ”optimal” (minimum and maximum index value for KL and XB respectively) number of clusters, this is due to the structure in the data where one of the clusters

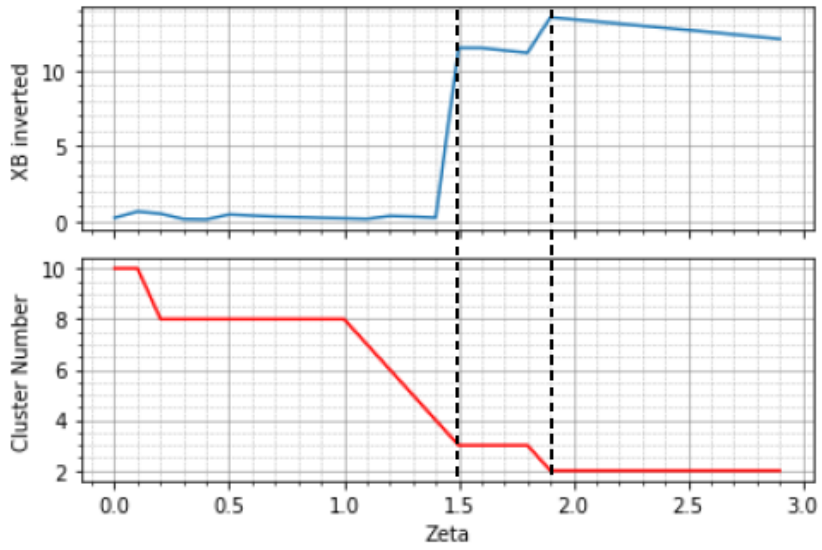


Figure 5.2: Internal validity index XB^{-1} and cluster number as function of ζ (the objective is to maximize XB^{-1})

can clearly be separated from other two. For visualization purposes, Sammon mapping—an algorithm that maps highly dimensional data to lower dimensions while at the same time maintaining the structure of the data—is employed. [41]. A typical run of the GGFP algorithm can be visualized in figure 5.3 and of the FCMFP algorithm in figure 5.4.

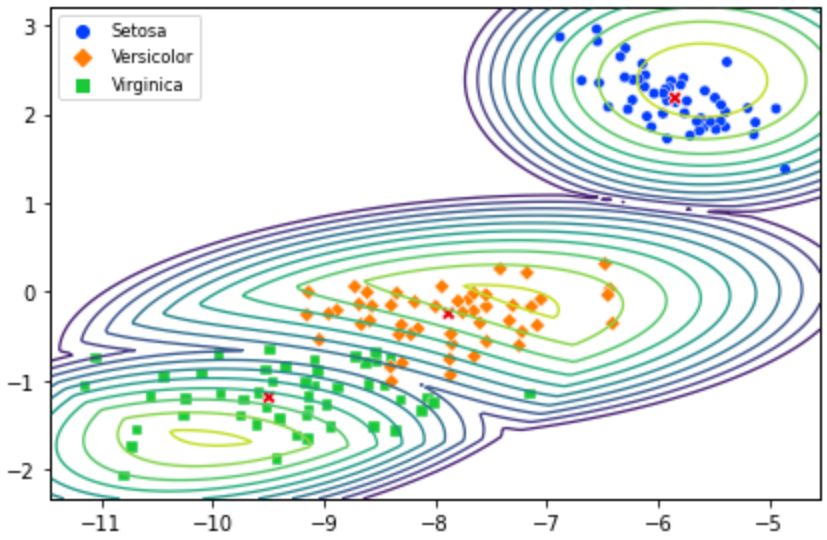


Figure 5.3: GGFP algorithm Sammon projection of the 4-dimensional feature space, of the Iris dataset, into the plane

Around each cluster center (marked with a red 'x') there are ten solid line curves, each representing a contour of equal membership value. The further away a curve is from the center, the smaller the membership value (the color mapping from yellow to dark blue colors signifies high to low membership). Inspection of figures 5.3 and 5.4, makes it possible to see how the different algorithms create different partitions. In the GGFP case, the possibility of having different hyper-ellipsoidal shapes, sizes and

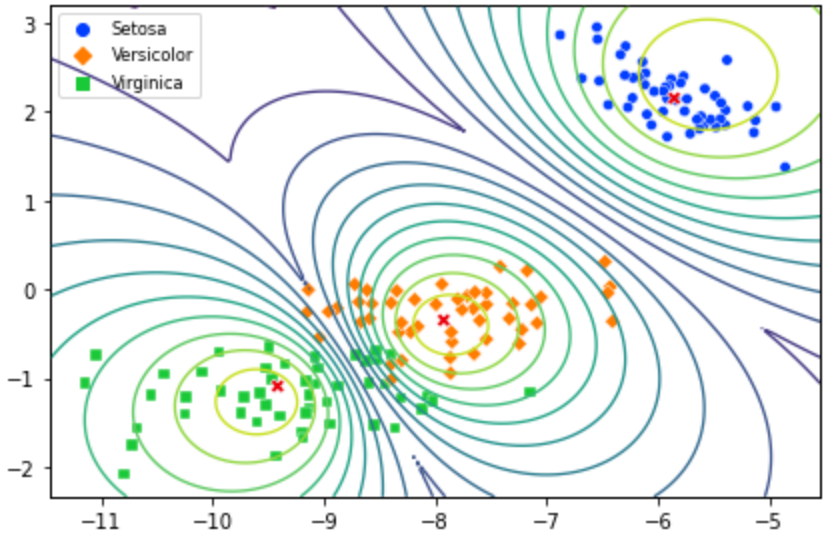


Figure 5.4: FCMFP algorithm Sammom projection of the 4-dimensional feature space, of the Iris dataset, into the plane.

orientation of clusters allows for a better fit of the data structure, while the FCMFP algorithm is limited by imposing the same sized spherical clusters.

5.1.2 Comparison with the corresponding unbiased algorithm and FCM/FCMFP

To compare the algorithms, 30 independent runs of each algorithm were performed and the resulting distributions of the Adjusted Rand Indices displayed in the form of boxplots, for different numbers of clusters.

To evaluate the statistical difference in the results Friedman and Wilcoxon signed-rank tests were used. The significance level considered for both the Friedman and Wilcoxon signed-rank test was $\alpha = 0.05$ which corresponds to a confidence interval of 95%. If $p_{value} < 0.05$ it is considered a statistically significant difference between the distributions, otherwise no statistical significant difference is noted.

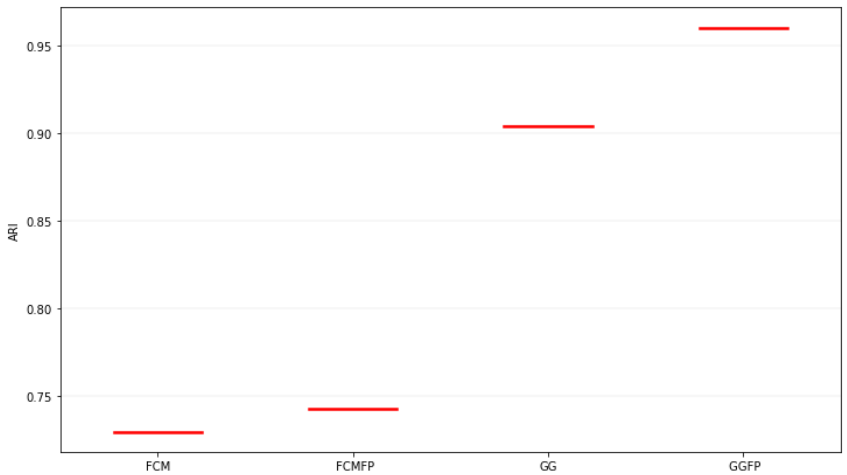


Figure 5.5: Adjusted Rand Index (ARI) boxplot comparison for 3 clusters in Iris data set

An analysis of figure 5.5 demonstrates that the boxplot results appear as a line, meaning that there is little variability in the results over the 30 runs. This can be attributed to a good initialization of the algorithms and that for this simple data set the algorithms converge to the same result. The value obtained by the Friedman test was $\mathrm{p}_{\text{Friedman}} \approx 0$ which reflects a statistical difference in the results. The results obtained for the ARI demonstrate a noticeable improvement that results from using the GG/GGFP, as it clusters this data with a very high accuracy, with a $ARI = 0.9038$ and $ARI = 0.9410$ for the GG and GGFP respectively. Although the difference in using the biased version as opposed to its non biased version is not as large, the effect of the shrinkage aspect of the observer biased algorithm can be observed.

5.2 Bearing Condition Data set

This data set is studied in [35], for the FCMFP, with the aim to monitor bearing condition, for both fault detection and fault classification. Two bearings are installed in a shaft driven by a controlled motor. Flywheels are mounted on the shaft to exert a load when required. An accelerometer is installed in each of bearing housing to measure the vibration signals that are then collected via an data acquisition card. The experimental setup can be see in figure 5.6. A total of 315 experiments were performed, with different shaft speeds and total applied loads for different bearing conditions . A total of 817 features were computed for each accelerometer distributed as 7 time domain features, 730 frequency domain,and 80 time-frequency features. In the frequency and time-frequency domains, the signals are divided into 80 bands of 20 KHz each, thus features are calculated for each band.

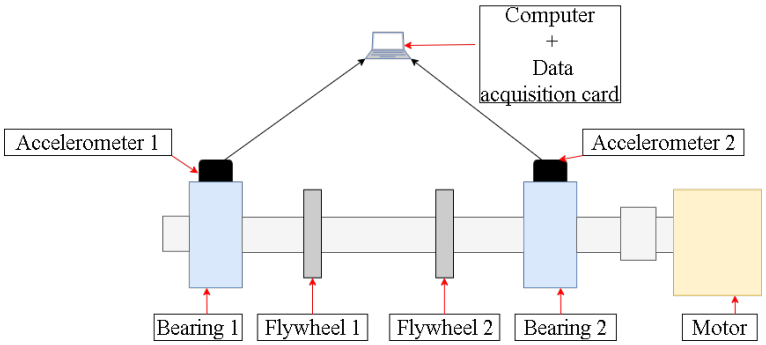
The features considered for the time domain were: the mean (μ), standard deviation (σ), variance (σ^2), root mean square (rms), kurtosis, skewness and crest factor (cf).

The time-domain signals are transformed into frequency signals utilizing Fast Fourier Transform (FFT). A total of 730 frequency domain features are computed for each accelerometer (features include mean, root mean squares, standard deviation, and kurtosis).

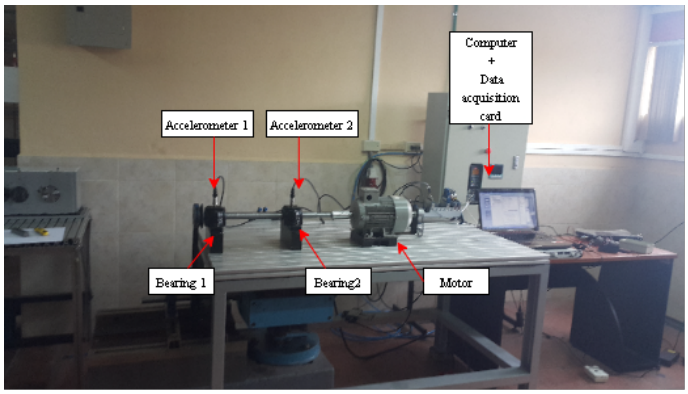
In the time-frequency domain, for each of the 80 bands five wavelets packet transforms are calculated. The list of wavelets employed consists of: Biorthogonal (bior6.8), Coiflets (coif4), Daubechies (db7), Symlets (sym3), Reverse Biorthogonal (rbio6.8). For a detailed description of all the features, the reader is referred to [35].

Table 5.1: Bearing State

ID	Bearing 1	Bearing 2
1	Healthy	Healthy
2	Inner race fault	Healthy
3	Outer race fault	Healthy
4	Ball fault	Healthy
5	Inner race fault	Outer race fault
6	Inner race fault	Ball fault
7	Outer race fault	Ball fault



(a) Diagram of Experimental Setup



(b) Laboratory Setup

Figure 5.6: Bearing Laboratory Test Rig

Feature selection was performed by employing entropy based feature selection [42] which resulted in 12 relevant features. Accelerometer 1 was found to be the most relevant accounting for the capture of 9 out of the 12 selected features. The 12 selected features are distributed as follows: 5 Wavelets or time-frequency domain features, 5 time domain and 3 frequency domain features.

The treated and ready to test data was kindly made available by the authors. The testing ready data set presents 7 classes each with 12 features, there is one class that represents the healthy state of both bearings and the other classes represent faulty states which are combinations of faults in one or both of the bearings and can be seen in table 5.1. The purpose of this section is to evaluate the GGFP algorithm in a previously tested and studied data set , which allows for further verification on a more complex data set than the Iris one.

5.2.1 Clustering results

Following the process as for the Iris data set, the iterative algorithm 6 is applied to the above mentioned 12 features, a wide range of reasonable partitions were revealed in figures 5.7 and 5.8.

The KL index (fig. 5.7) reveals a range of reasonable cluster numbers, $c = 2, 3, 5, 6$ and 7, whereas the XB index (fig. 5.8) gives as reasonable clusters, $c = 2, 3$ and 4. It is worth mentioning that the KL index is able to identify the ground truth number of clusters as a reasonable cluster ($c = 7$), unlike the XB index.

There are ranges ζ values for which the algorithm gives the same amount of clusters and a similar

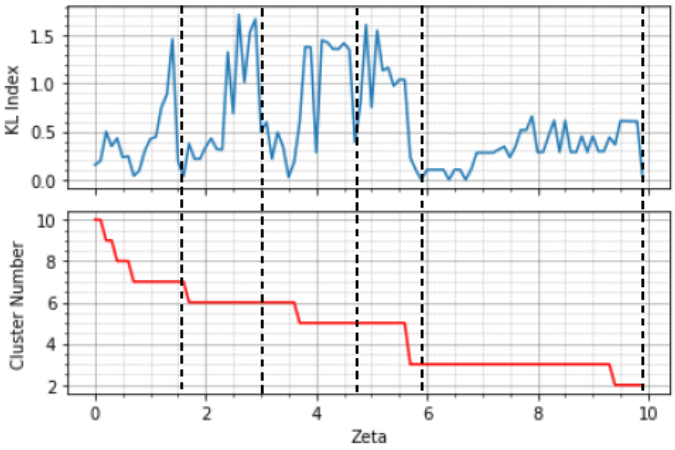


Figure 5.7: Internal validity index KL and cluster number as function of ζ (Objective is to minimize KL index in bearing condition data set

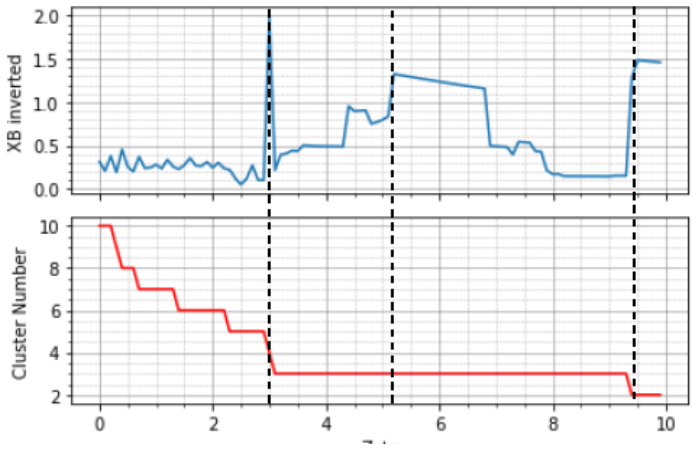


Figure 5.8: Internal validity index XB^{-1} and cluster number as function of ζ (objective is to maximize XB^{-1} in bearing condition data set

validation index. Within this range the values for ARI and RI vary depending on the ζ chosen. Looking more closely at one of those ranges, 30 independent runs with different initial conditions were performed and the performance of the algorithm, in terms of ARI, is tested for the extreme values and two intermediate values of ζ in the range of [1.6, 3.8] that all produce 6 clusters.

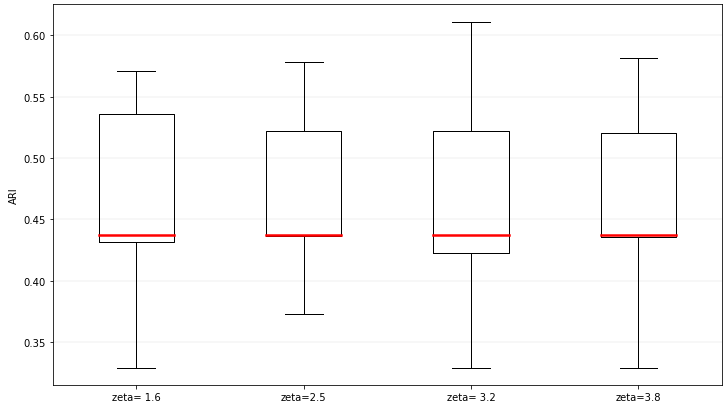


Figure 5.9: Adjusted Rand Index (ARI) boxplot of GGFP in the range of zeta [1.6, 3.8] in bearing condition data set

The Friedman test reveals $\mathrm{p}_{\text{Friedman}} = 0.7022$ meaning that there is no statistical difference in the results of figure 5.9. It is possible to change ζ within this range without affecting the ARI drastically; this highlights the fact that the GGFP algorithm can give different perspectives of the data even for the same focal point and number of clusters without compromising performance.

5.2.2 The healthy vs. faulty case

This section will centre around the fault detection case, i.e., the case where the goal is to verify whether a fault (no matter which) exists. This is equivalent to cluster the data with $c = 2$ clusters. The colored symbols represent the truth classification of a sample. Samples in the same class are represented by the same color and symbol according to the legend in the figures, the healthy state is class 1 represented in green. The 10 contour lines around the cluster centers (red 'x') have the same meaning previously mentioned in section 5.1.1.

Analyzing figure 5.10 it is possible to observe that all the healthy data points (P1 - healthy state) are classified together. Comparing the GGFP with the FCMFP, which has clusters of same size and shape, it is observable that although both algorithms correctly classify the healthy samples, the GGFP algorithm is able to more accurately distinguish between either faulty and healthy states while the FCMFP algorithm classifies more faulty data as healthy. The nature of the GGFP algorithm (clusters of different shapes, sizes and densities) allows data to be clustered more accurately, the drawback being the fact the algorithm is not very robust as it requires a good initialization in order to produce good results.

The multi-fault classification case

From the analysis of the cluster validity index of figures 5.7 it follows that different structural divisions of the data are reasonable. In particular, partitions with number of clusters $c = 2, 3, 5, 6, 7$ provide

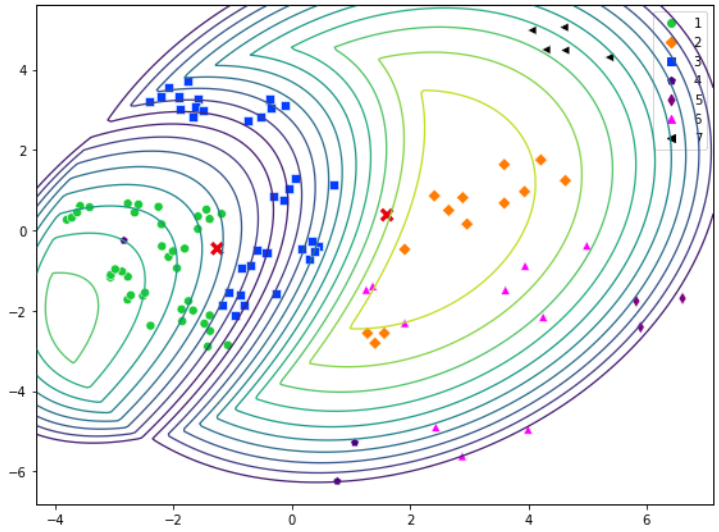


Figure 5.10: GGFP Sammon projection of the 12-dimensional feature space into the plane for the fault detection case ($c = 2$) in bearing condition data set

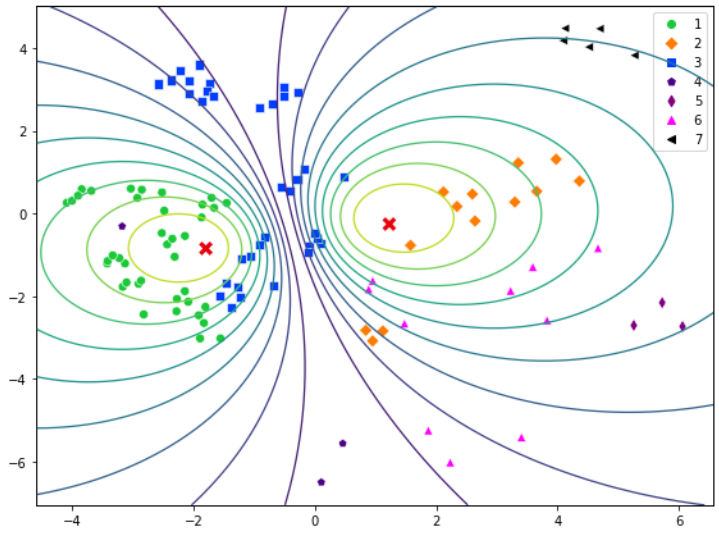
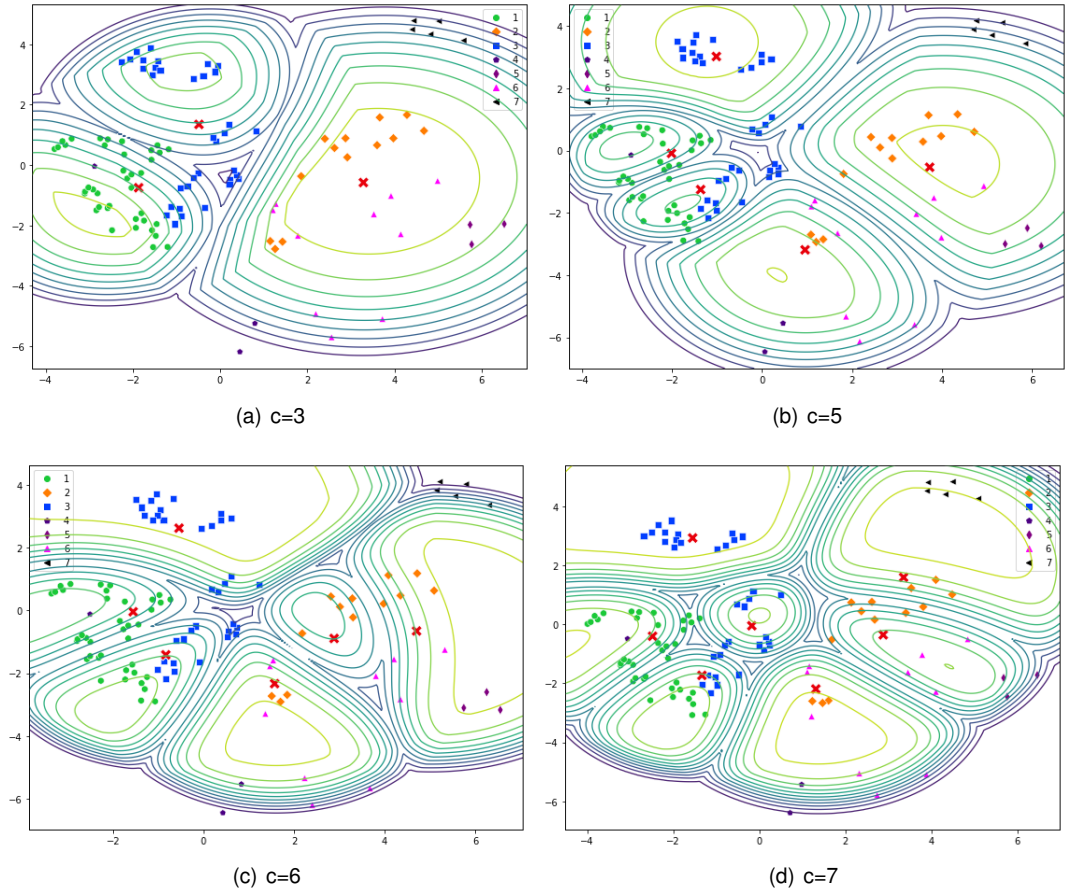


Figure 5.11: FCMFP Sammon projection of the 12-dimensional feature space into the plane for the fault detection case ($c = 2$) in bearing condition data set

reasonable structures. Figure 5.12 demonstrates these different perspectives making use of Sammon projections in a sequence that appears similar to zooming in on the data.

Starting from figure 5.10 , which corresponds to $c = 2$, and transitioning to figure 5.12(a), a cluster corresponding the to the blue state 3 is formed. From 5.12(a) to 5.12(b), the initial cluster corresponding to the healthy state has been split into two. The rest of the figures show this continued trend where the current bigger cluster gets progressively subdivided into consecutive sub-clusters.



(a) c=3 (b) c=5

(c) c=6 (d) c=7

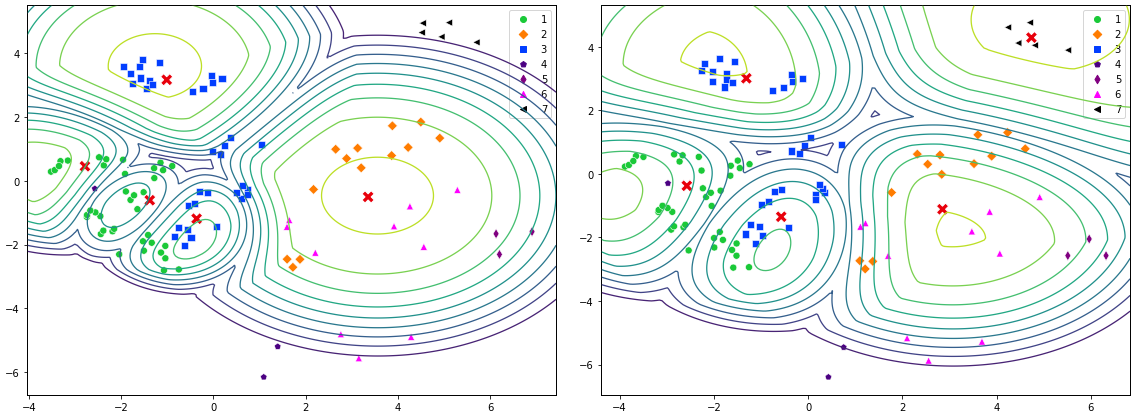
Figure 5.12: GGFP Sammon projections of the 12-dimensional feature space into the plane for fault classification for bearing condition data set

In order to see the effect of the observer metaphor, the feature space is explored in figure 5.13. This is accomplished by placing the focal point in the location where the features obtain their minimum and maximum values (instead of the barycenter of the data) for a different number of clusters. Even though the same number of clusters is visualized in each horizontal pair of subfigures, different levels of detail can be seen depending on the placement of the focal point.

5.2.3 Fuzzy parameter sensitivity analysis

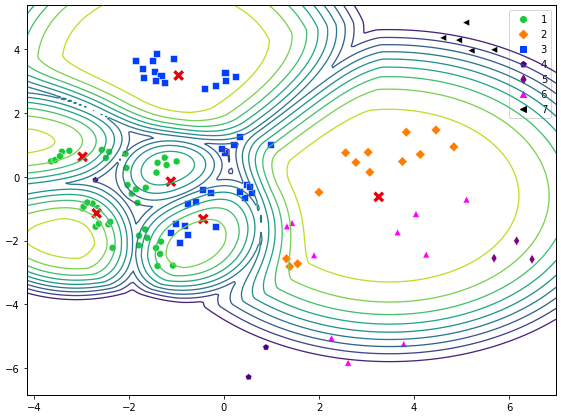
The fuzzy parameter,m, relates to the degree of fuzziness of the partition. Typically for large values of m, the classes will tend to blend together and the clusters will share more datapoints.

During testing it was observed that the GGFP algorithm would perform differently when this param-

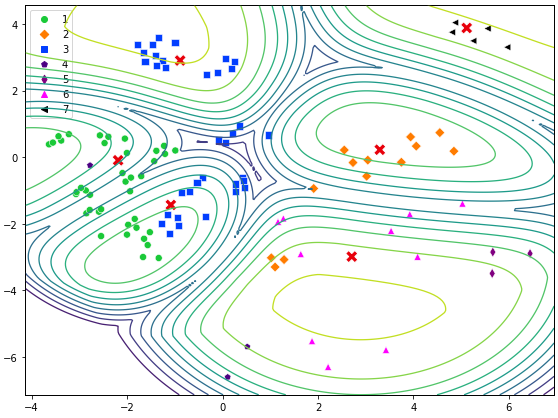


(a) $\boldsymbol{p}_{min}$, c=5

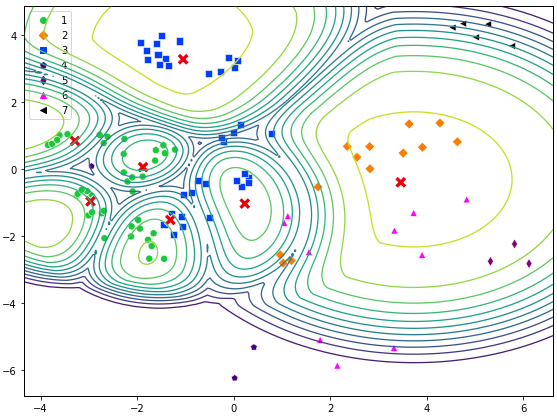
(b) $\boldsymbol{p}_{max}$, c=5



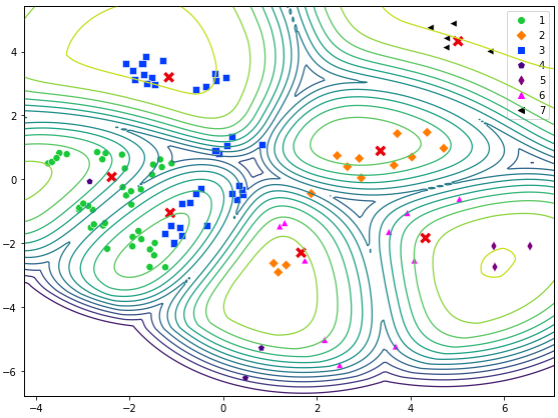
(c) $\boldsymbol{p}_{min}$, c=6



(d) $\boldsymbol{p}_{max}$, c=6



(e) $\boldsymbol{p}_{min}$, c=7

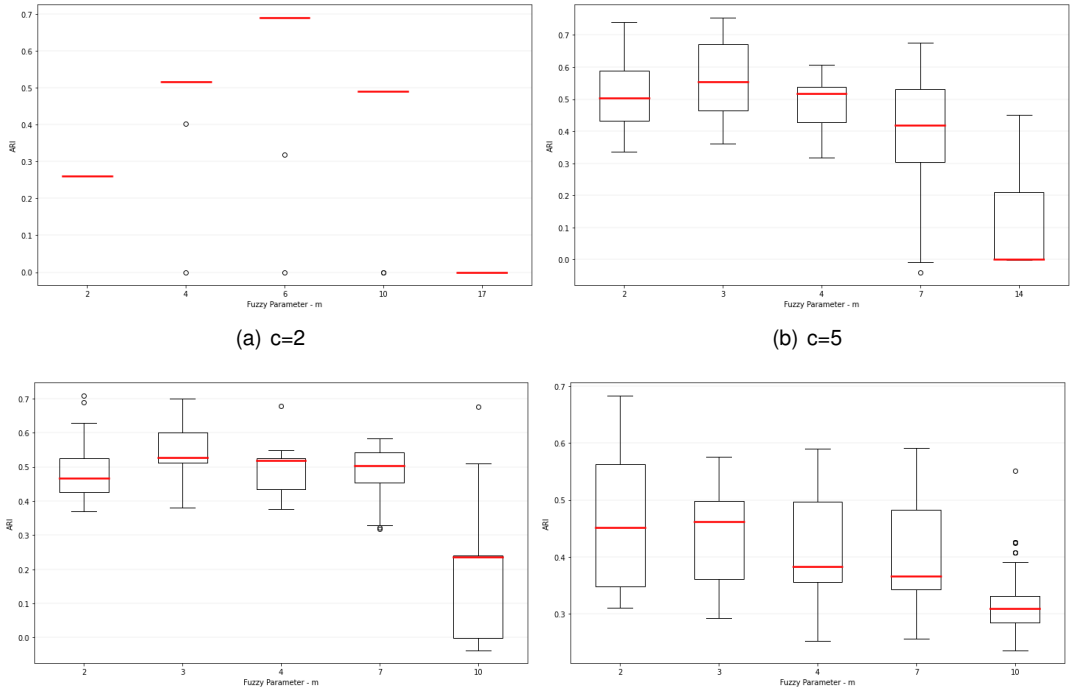


(f) $\boldsymbol{p}_{max}$, c=7

Figure 5.13: Sammon projections with more detailed view of different regions of the data space in bearing condition data set. Left side sub-figures correspond to the focal point placed in the region where features attain their minimum values ($\boldsymbol{p}_{min}$). Right side sub-figures correspond to the focal point placed where features attain their maximum value ($\boldsymbol{p}_{max}$).

eter was changed, but unlike the FCMFP that when increased above $m = 2$ would produce a single cluster due to cluster overlap, this parameter was more sensitive in the GGFP algorithm. That is to say, increasing it to certain values would produce better results in terms of ARI.

An analysis was performed on the sensitivity of the fuzzy parameter in the GGFP algorithm, the results of which are shown in figure 5.14 for different numbers of cluster. It can be extracted that depending on the number of clusters there is an optimal value of the fuzzy parameter, m. This value was found to be experimental and trial and error was employed. This can be explained due to the exponential nature of the distance metric in the GGFP algorithm; the distance between clusters can be sometimes extremely large and thus by fuzzifying the partition can help the algorithm find better partitions [43].



(a) c=2 (b) c=5

(c) c=6 (d) c=7

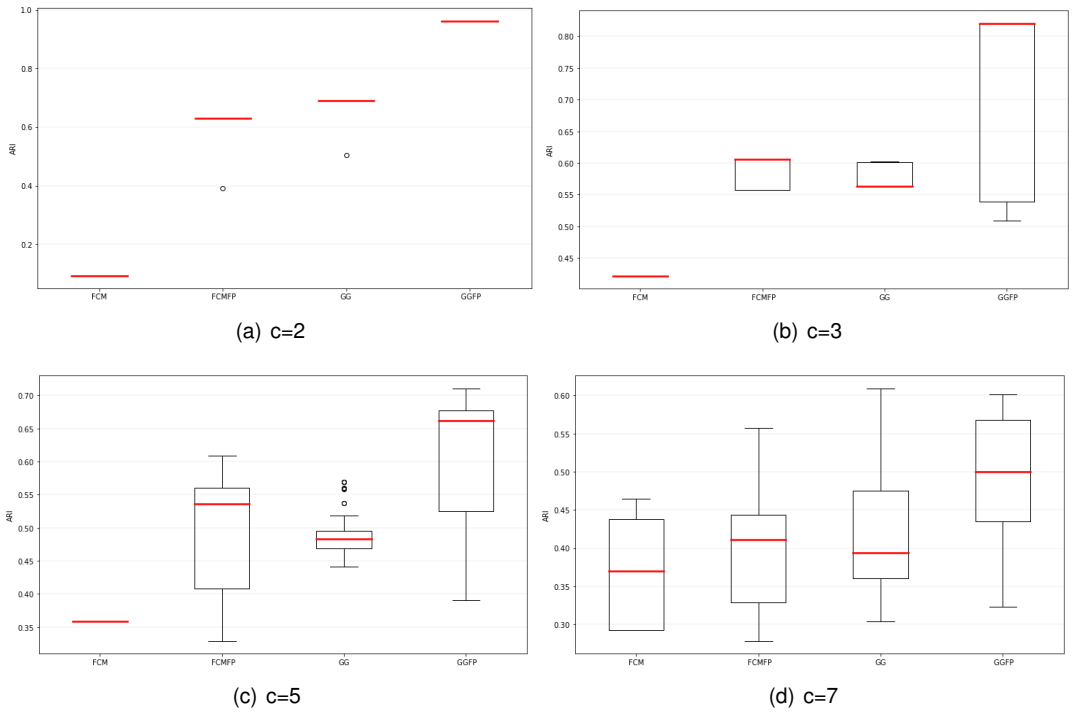
Figure 5.14: GGFP fuzzy parameter boxplot comparison for different numbers of clusters in bearing condition data set

5.2.4 Comparison with the corresponding unbiased algorithm and FCM/FCMFP

In this section the goal is to assess the quality of the final results produced by the GGFP, the FCMFP, and their respective unbiased versions. Momentarily, the ability of the observer biased algorithms to explore different regions of the feature space is ignored and the parameters that produce better results are selected for all algorithms. In this scenario, the question of how the GGFP compares to its unbiased version and to the FCM/FCMFP algorithm is posed.

To answer this question, the algorithms are compared resorting to boxplot graphs of the distributions of the Adjusted Rand Index over 30 independent runs for each algorithm for a different number of clusters. The statistical difference tests used and their significance levels are the same as in section 5.1 and the results are shown in figure 5.15. It is observable that the FCM (with kmeans++ initialization) results

in a low ARI, especially for $c = 2$ and $c = 3$, which turned out to be a very poor initialization for the Gath-Geva algorithm. Hence, the FCMFP due to its improved performance over the FCM was used to initialize the GG/GGFP. The GGFP algorithm outperforms its unbiased version as well as the FCM and FCMFP for most of the reasonable clusters given by figure 5.7 and the values of the Friedman test ($p_{\text{Friedman}} \approx 0$) reveal that there is statistical difference in the results obtained. The most noticeable improvements are observed for smaller numbers of clusters, the biggest being in the fault detection case.



(a) c=2 (b) c=3

(c) c=5 (d) c=7

Figure 5.15: Adjusted Rand Index (ARI) boxplot comparison for different numbers of clusters in bearing condition data set

5.3 Wind Turbine Fault Diagnosis Application

Experiments were performed with the planetary gear box of a RCVA-300 vertical axis wind turbine. The experimental setup consists of a blower positioned in front of the wind turbine, a PCB 3-axis accelerometer mounted on top of the gearbox, and a SQI data acquisition system as shown in the diagram 5.16(a). The PCB accelerometer is responsible for the measurement of the vibrational data, followed by transmission of acquired data to the SQI data acquisition system analog to digital conversion with 100 kHz of sampling rate and a sampling interval of 20 seconds. 10 repetitions per experiment were performed and 155520 data points with 787 features were obtained.

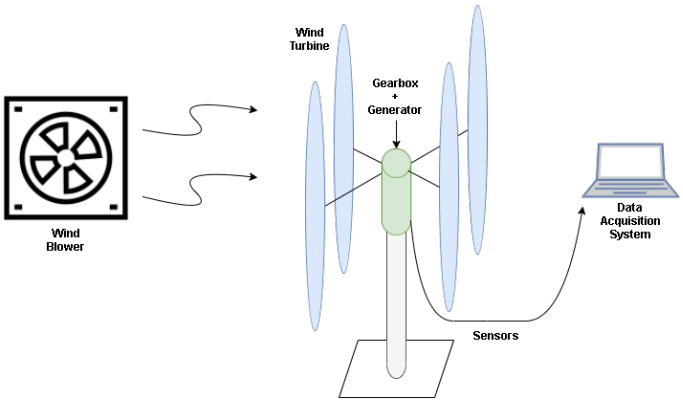
A total of 787 features were computed: 11 time domain features, 687 frequency domain and 89 time-frequency features. For the time domain the 11 considered features were: the mean, root mean square (rms), standard deviation, kurtosis, signal peak, crest factor, rectified average, form factor, impulse factor, variance and the signal mininum.

To obtain the frequency signals, Fast Fourier Transform (FFT) is applied to the time domain signal

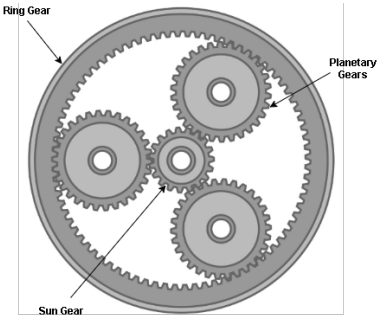
and for 89 bands a total of 687 frequency domain features are computed.

In the time-frequency domain, the same five wavelets packet transforms as for the bearing data (bior6.8, coif4, db7, sym3 and rbio6.8). They are calculated for all of the 89 bands.

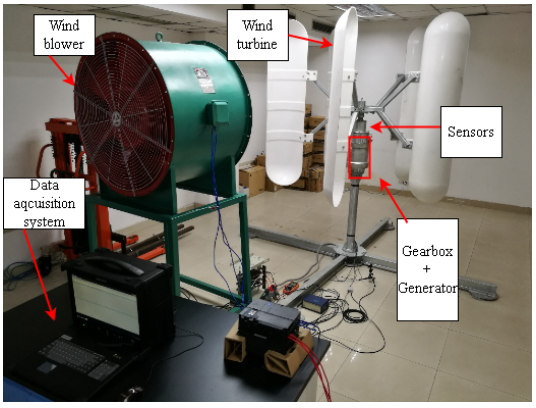
The vibrational data was acquired under nominal conditions as well as under different types of faults in components of the gearbox, namely: in the ring gear, the planetary gear and the sun gear. Figure 5.16(b) shows the structure of the gearbox .



(a) Diagram of Experimental Setup



(b) Gearbox Structure



(c) Laboratory setup

Figure 5.16: Laboratory Test Rig

Random Forest feature selection was employed and the number of 787 features was reduced to 33.

The data set ready for clustering possesses 17280 data points each with 33 features. there are four classes which are divided as follows: 1) No fault - Healthy state , 2) Ring gear fault, 3) Sun gear fault and 4) Planetary gear fault as shown in table 5.2.

Table 5.2: Condition of Gearbox

ID	Ring Gear	Sun Gear	Planetary Gears
1	Healthy	Healthy	Healthy
2	Fault	Healthy	Healthy
3	Healthy	Fault	Healthy
4	Healthy	Healthy	Fault

Due to the above-mentioned large number of samples and high number of dimensions, this data

set presents itself as more complex and computationally demanding compared to the previously tested datasets, and thus gives rise to the problem of representing the data in two dimensions for visualization. The Sammon projection highly depends on the number of samples, and data sets with a high number of data points cause the algorithm to become computationally expensive [44]. Due to the large amount of samples in this dataset it was found to be unfeasible to employ Sammon Mapping as a visualization technique. The more recent UMAP visualization technique provides several advantages over Sammon Mapping, in particular when it comes to handling large data sets [45]. This was the was chosen visualization method from this point forward.

5.3.1 Clustering results

Following the same procedure as in the previous sections, algorithm 6 is applied in order to obtain a range of reasonable partitions for the data.

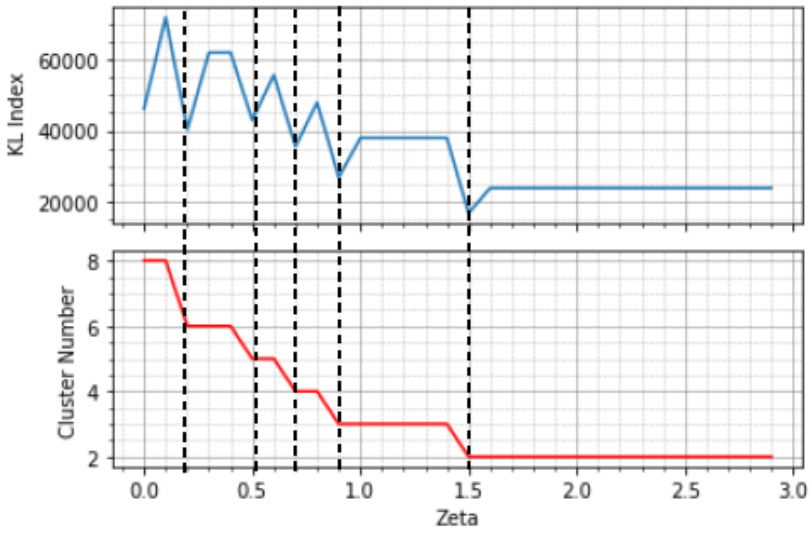


Figure 5.17: Internal validity index KL and cluster number as function of ζ (Objective is to minimize KL index) in WT data set

Observing figures 5.17 and 5.18, the KL index reveals as reasonable partitions of data the cluster numbers, $c = 2, 3, 4, 5$ and 6 while the XB index reveals as reasonable only $c = 2$. In this data set the difference in using the KL index is more noticeable than in the previous data sets. The KL index is able to evaluate as reasonable more partitions of the data and also identifies the ground truth partition as reasonable ($c = 4$). Interestingly, looking at the maximum index value for XB^{-1} and the minimum value for KL it is clear that the partition that optimizes both indexes is the same, $c = 2$; the reason for this can be deduced observing a visual representation of the data, such as figure 5.20, as the structure in the data seems to be divided in two major parts.

There are ranges of ζ values for which the algorithm gives the same amount of clusters and a similar validation index. Within this range the values for ARI and RI vary depending on the ζ chosen. Looking more closely at one of those ranges, 30 independent runs with different initial conditions were performed and the performance of the algorithm, in terms of ARI, is tested for the extreme values and

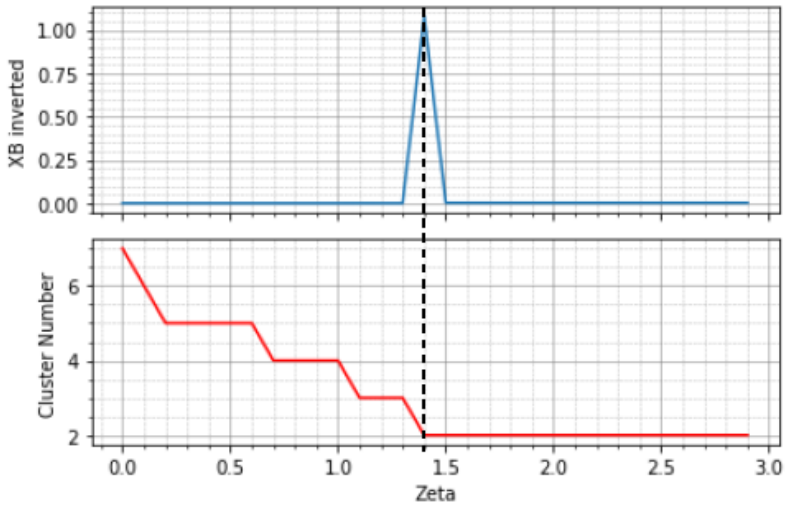


Figure 5.18: Internal validity index XB inverted and cluster number as function of ζ (Objective is to maximize XB inverted index) in WT data set

two intermediate values of ζ in the range of [0.2, 0.4] that all produce 6 clusters, the results can be seen in figure 5.19. The Friedman test gives $\mathrm{p}_{\text{Friedman}} = 0.6041$ revealing that there are no statistical differences

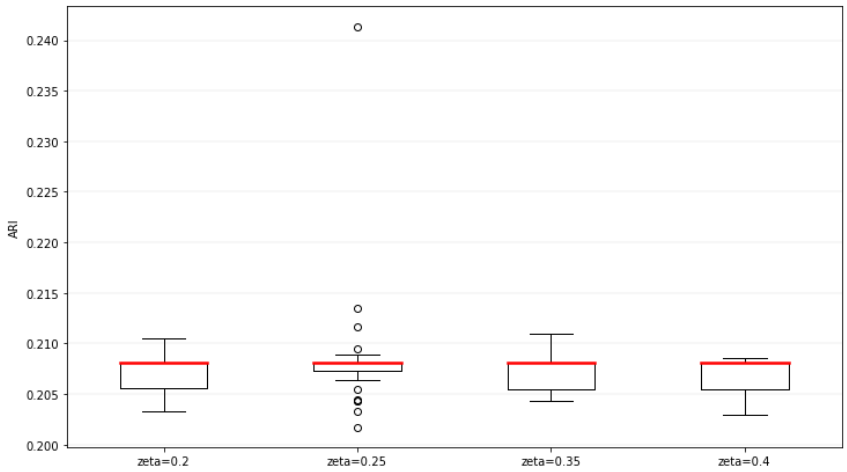


Figure 5.19: Adjusted Rand Index (ARI) boxplot of GGFP in the range of zeta [0.7, 1.1] in WT data set

in the data thereby drawing the conclusion that it is possible to change ζ within this range or any other range that produces the same amount of clusters without great change in the ARI values. Similarly to the previous data set, it is verified here that the GGFP algorithm can give different perspectives of the data without sacrificing performance.

5.3.2 The healthy vs. faulty case

The data is clustered with $c = 2$ clusters in order to analyse the fault detection case. Figures 5.20 and 5.21 show a typical run of the GGFP and FCMFP algorithm for the fault detection case. The different symbols and colors represent the four classes according to the legend in the following figures and the red colored 'x' the cluster centers. The 10 solid curves around the cluster centers maintain the same

meaning as in the previous data sets.

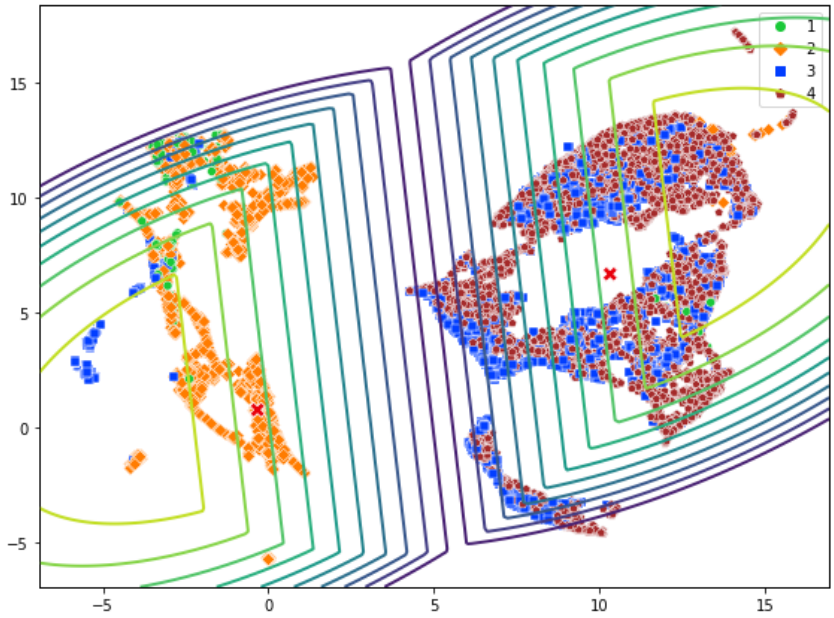


Figure 5.20: GGFP algorithm UMAP projection of the 33-dimensional feature space into the plane for the fault detection case ($c = 2$) for WT data set

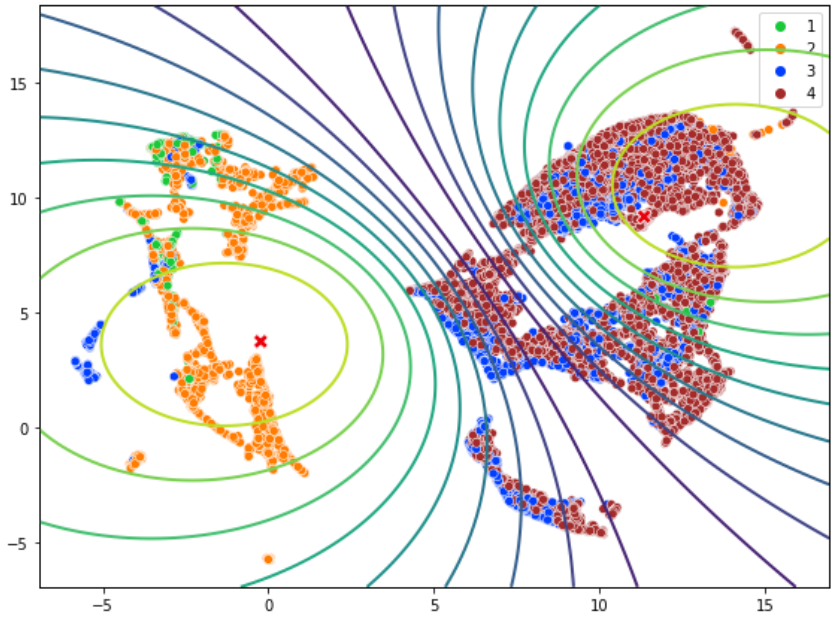


Figure 5.21: FCMFP algorithm UMAP projection of the 33-dimensional feature space into the plane for the fault detection case ($c = 2$) for WT data set

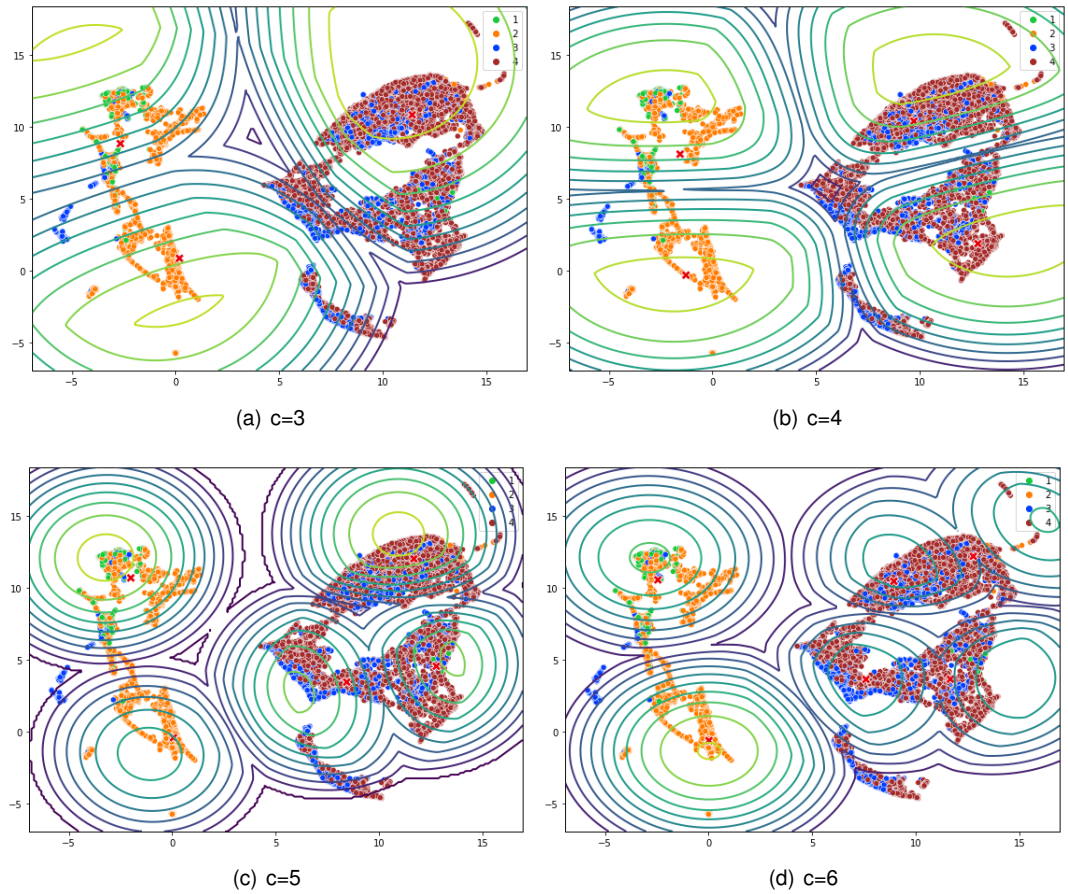
Upon analyzing figure figure 5.20 it can be observed that the majority of the healthy data points (labeled as class 1, represented in green) are clustered together, although there are some mislabeled healthy points.

Most of the faulty points belonging to class 2 and some from class 3 were clustered together with the healthy samples. It is clear that, for this data set, fault classification is a difficult task, and that the

structure in the data—especially as it pertains to classes 3 and 4—is not easily distinguished between classes. It can be said that if data belongs to the cluster in the right (in figure 5.20) then it is very likely that it is not a healthy sample. When comparing the two algorithms visually, it becomes clear that the GGFP algorithm managed to cluster most of the healthy samples together and separated more faulty points from the healthy cluster than the FCMFP.

5.3.3 The multi-fault classification case

Inspecting the cluster validity KL index analysis, in figure 5.17, partitions with number of clusters $c = 2, 3, 4, 5$ and 6 provide reasonable structural alternatives. Figure 5.22 show the different partitions resorting to UMAP projections in a sequence that appears similar to zooming in on the data. Starting from figure 5.20, which corresponds to $c = 2$, the cluster where the healthy samples belong gets subdivided into two in figure 5.22(a), and the trend of the bigger cluster getting subdivided into consecutive sub-clusters is continued in the following figures 5.22(b) to 5.22(d).



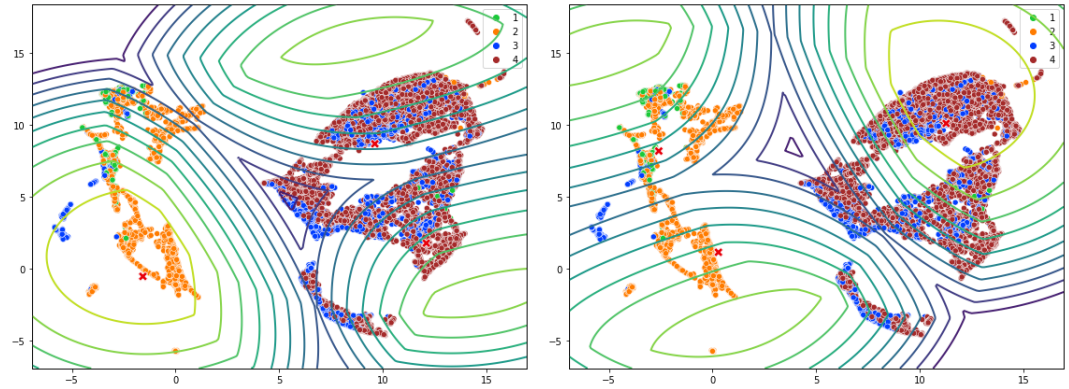
(a) c=3 (b) c=4

(c) c=5 (d) c=6

Figure 5.22: UMAP projections of the 33-dimensional feature space into the plane for fault classification

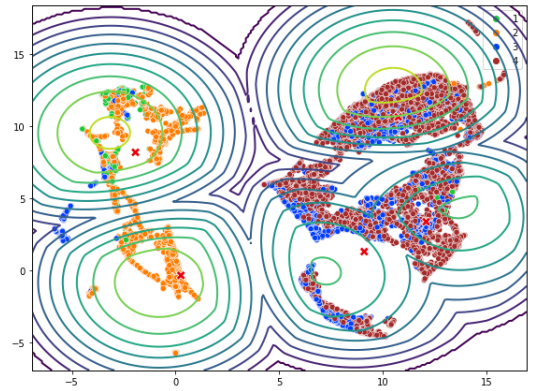
In order to substantiate the observer metaphor in this data set, the feature space is explored. In figure 5.23, the focal point is placed in the location where the features obtain their minimum and maximum values for a different number of clusters. The same number of clusters is visualized in each horizontal pair of sub-figures and it is possible to see that different levels of detail, of a certain region, can be

obtained depending on the placement of the focal point.

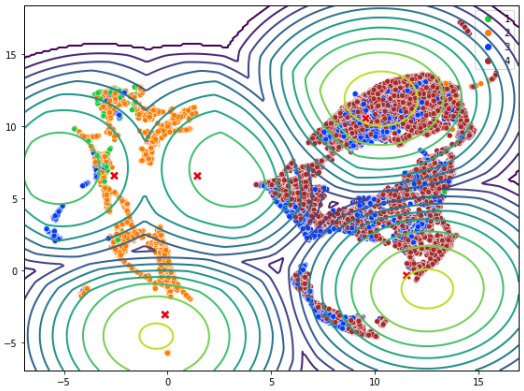


(a) $\boldsymbol{p}_{min}$, c=3

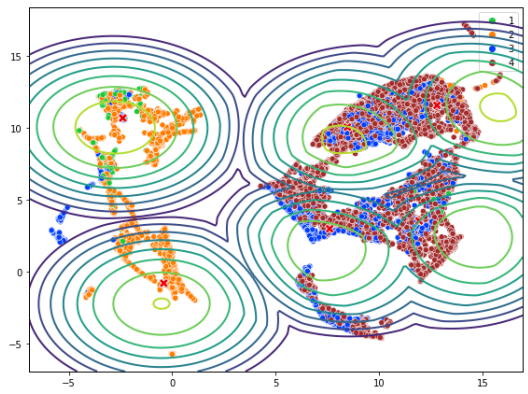
(b) $\boldsymbol{p}_{max}$, c=3



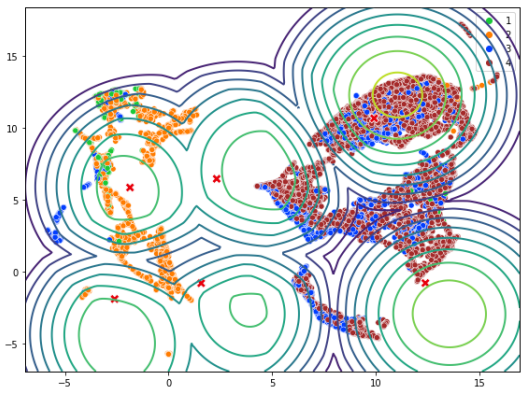
(c) $\boldsymbol{p}_{min}$, c=5



(d) $\boldsymbol{p}_{max}$, c=5



(e) $\boldsymbol{p}_{min}$, c=6



(f) $\boldsymbol{p}_{max}$, c=6

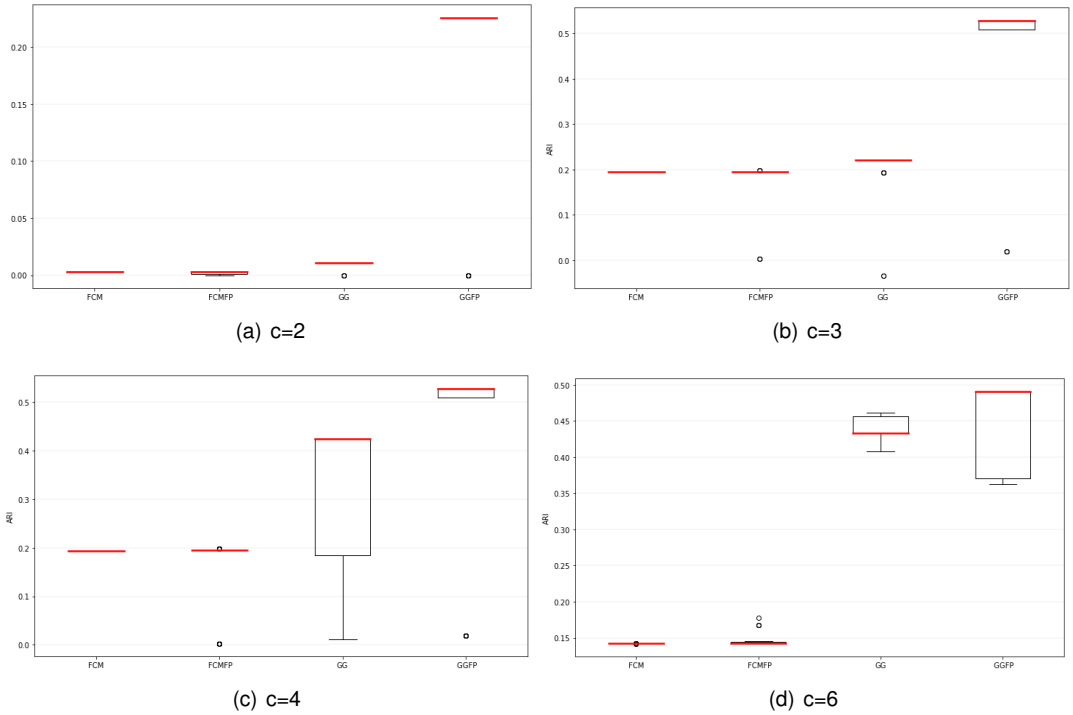
Figure 5.23: UMAP projections with more detailed view of different regions of the data space in WT data set. Left side sub-figures correspond to the focal point placed in the region where features attain their minimum values ($\boldsymbol{p}_{min}$). Right side sub-figures correspond to the focal point placed where features attain their maximum value ($\boldsymbol{p}_{max}$).

In figure 5.23, the benefits of being able to control the cluster formation are displayed, for example, if the goal is to analyze the specific region of the data where the healthy samples lie, this could be achieved by placing the focal point in the location where the features obtain their maximum value, allowing the focus to be on a more detailed visualization of the desired area. For example, for the case of $c = 3$, figures 5.23(a) and 5.23(b) illustrate this idea: in the right hand side figure there is more detail in the

region of the healthy samples when compared with the left-side figure.

5.3.4 Comparison with the corresponding unbiased algorithm and FCM/FCMFP

In this section, just as with the bearing dataset, the algorithms are compared using boxplots of the distributions of the Adjusted Rand index over 30 independent runs of each algorithm for different numbers of clusters. The significance level considered was the same that was used in previous sections, $\alpha = 0.05$ which corresponds to a confidence interval of 95%.



(a) c=2 (b) c=3

(c) c=4 (d) c=6

Figure 5.24: Adjusted Rand Index boxplot comparison for different numbers of clusters in the WT data set

The Friedman statistical test revealed a value of $\mathrm{p}_{\text{Friedman}} \approx 0$ for all clusters analyzed in figure 5.24, signifying that there is statistical difference in the data. Performing the Wilcoxon test among the pairs of algorithms revealed that there is no statistical difference in the data produced from the FCM/FCMFP for $c = 2$ and 3 clusters (with $\mathrm{p}_{\text{Wilcox}} = 0.0527, \mathrm{p}_{\text{Wilcox}} = 0.355$, respectively).

For this data set the FCM and FCMFP algorithms performed at about the same level, either of them could be used as initialization for the GG/GGFP. The initialization was then performed with the FCMFP algorithm.

Analyzing figure 5.24, the GGFP algorithm is able to outperform its unbiased version, as well as the FCM and FCMFP for the reasonable clusters given in figure 5.17. For the fault detection case, c=2, the FCM, FCMFP and GG the ARI values are nearly zero, while the GGFP manages to obtain, approximately an ARI of 0.22.

The GGFP always seems to give a better partition as evaluated by the ARI metric. It can be observed that the ARI values are higher for the GGFP in comparison to FCM/FCMP for all the analyzed clusters;

when compared to its unbiased version, GG, the gap shortens as the number of clusters is increased.

The shrinkage effect of the observer biased framework is again verified and for this data set the benefits of employing the GGFP over the FCMFP or FCM are clear. As for the comparison with its unbiased version, the results show that there is always a slight increase in performance which is more noticeable when the number of clusters in question are lower. Similarly to the bearing data set, the fault detection case is where the application of the GGFP more notably outperforms the other algorithms.

These results showcase the benefits of employing the GGFP for this type of application, especially coupled with its inherent ability to explore the feature space. The GGFP proved to be a powerful tool to be considered when tackling fault detection and classification problems.

Chapter 6

Conclusions and Future Work

In this dissertation, the problem of wind turbine gearboxes fault detection and classification was addressed and observer biased fuzzy clustering was explored for the classification of the data. The observer biased paradigm permits users to interactively select different levels of granularity in the search for clusters that accurately represent the structure embedded in the data. Using the, already successful, implementation of the observer biased framework in the FCM and GK algorithms as motivation to explore other fuzzy clustering algorithms, the Gath-Geva with a Focal Point (GGFP) was implemented and the benefits of employing this algorithm in comparison to the existent observer biased clustering algorithms were evaluated.

The GGFP algorithm, which had never been implemented and tested, was the focus of the analysis. The GGFP can be regarded as a generalization of the GG and FCMFP algorithms. In applying the iterative process to find reasonable numbers of clusters to GGFP, it was possible to see how the two employed internal validations differ for an algorithm that allows its clusters to have different shapes and sizes. The XB index revealed less reasonable partitions for the bearing and WT data sets than the KL index, and only for the Iris data (the simplest of the three tested data sets) the XB index gave as a reasonable cluster number the ground truth number of clusters (c=3). This leads to the conclusion that although the XB index may be used for the GG/GGFP when considering data sets with a structure that allows the clusters to present a spherical shape. It cannot be generalized for all data sets to produce optimal results. Therefore, utilizing the KL index for the GG/GGFP was proven to be a good alternative to internal indexes that only rely on the distance between cluster centroids.

For the bearing and WT data sets, when performing the iterative process to find reasonable numbers of clusters, it was noted that for some ranges of ζ the number of clusters would remain the same for varying internal validity indexes values. A closer look at these ranges (in c=6 for both bearing and WT data sets) showed that the effect of changing the ζ coefficient within these ranges was minimal in performance as evaluated by the Adjusted Rand Index (ARI). Changing ζ signifies changing the view of the data and so this allowed the confirmation of the capability of the GGFP to generate different perspectives of the data without compromising the quality of the partitions.

With the aid of projection techniques to visualize high dimensional data, the differences in using the

FCM and GG based algorithms was shown across the three data sets. The hyper-ellipsoidal cluster shapes generally allowed for a better fit to the data structure including in the fault detection and classification cases.

An analysis on the sensitivity of the fuzzifier parameter, m, was performed which it revealed that depending on the data and the number of clusters there are values for m that produce better results when using the GG/GGFP algorithm, unlike in the FCM/FCMFP where increasing $m > 2$ will usually result in an overlap of clusters. This threshold to encounter cluster overlap was noted to be larger in the GGFP and in between $m > 1$ and this threshold, an optimal value for m can be found.

Furthermore, GGFP was able to produce statistically significant better results,in all three data sets, in comparison to the FCM,FCMFP and GG using as a metric the external validity measure Adjusted Rand Index. The WT gearbox data was of difficult analysis due to the structure of the data which made fault classification a complex task, despite this the GGFP managed to be superior to the other algorithms for all numbers of clusters. It is not unreasonable to think that even better results could be achieved by further processing the data; one option could start by reducing the number of features or scaling the data set to a lower, but still representative, number of samples.

More importantly, the application of the observer biased framework successfully allowed for the exploration of the feature space, making the GGFP a very interesting exploratory analysis tool. The performance increase that the GGFP revealed comes with a price, the fact that the algorithm needs to have a good initialization for the prototypes in order to achieve optimal results makes it a less a robust alternative to the FCMFP, as well as the fact that due to its more complex distance metric, the computational time increases considerably. There is a clear trade-off between performance and efficiency and it its up to the user to balance the pros and cons of employing either of the algorithms, according to needs of the problem at hand.

For future work, the design of a simple and interactive interface could be developed in order to facilitate the exploration of the data space, where the user could be allowed to easily tune parameters such as the location of the focal point or the regularization coefficient. The study of observer-biased clustering with multi-focal points could also be considered and an evaluation on any merits, if any, in applying it over the current version of the observer-biased algorithm.

Bibliography

[1] Global Wind Energy Council (GWEC). Global wind report 2021.

[2] Energy Administration Information (EIA). Wind explained - Electricity generation from wind. URL `https://www.eia.gov/energyexplained/wind/electricity-generation-from-wind.php`. (Accessed: 22/10/2021).

[3] E. Wiggelinkhuizen, H. Braam, T. Verbruggen, and L. Rademakers. Condition monitoring for offshore wind farms. *EWEC2007 conference 7-10 May, Milan, Italy*, 2003.

[4] B. Hahn, M. Durstewitz, and K. Rohrig. Reliability of wind turbines. In *Wind Energy*, pages 329–332. Springer, Berlin, Heidelberg, 2007.

[5] F. P. García Márquez, A. M. Tobias, J. M. Pinar Pérez, and M. Papaelias. Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178, 2012. ISSN 0960-1481.

[6] Z.Hameed, Y.S.Hong, Y.M.Cho, S.H.Ahn, and C.K.Song. Condition monitoring and fault detection of wind turbines and related algorithms: a review. *Renewable and Sustainable Energy Reviews*, pages 1–39, 2009.

[7] T. Wang, Q. Han, F. Chu, and Z. Feng. Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review. *Mechanical Systems and Signal Processing*, 126: 662–685, 2019. ISSN 0888-3270.

[8] L.-M. Wang and Y.-M. Shao. Crack fault classification for planetary gearbox based on feature selection technique and k-means clustering method. *Chinese Journal of Mechanical Engineering volume*, 31(4), 2018.

[9] Z. Li, R. Jiang, Z. Ma, and Y. Liu. Fault diagnosis of wind turbine gearbox based on kernel fuzzy c-means clustering. *International Conference on Renewable Power Generation*, 2015.

[10] S. Lotfan, N. Salehpour, H. Adiban, and A. Mashroutechi. Bearing fault detection using fuzzy c-means and hybrid c-means-subtractive algorithms. *IEEE International Conference on Fuzzy Systems*, 2015.

[11] C. L. Liu, X. M. Huang, and X. J. Luo. Roller bearing fault diagnosis based on elmd and fuzzy c-means clustering algorithm. *Applied Mechanics and Materials*, 602:1698–1700, 2014.

[12] H. Wang, F. Wu, and L. Zhang. Fault diagnosis of rolling bearings based on improved empirical mode decomposition and fuzzy c-means algorithm. *Traitement du Signal*, 38(2):395–400, 2014.

[13] L. Zhang, P. Li, M. Li, S. Zhang, and Z. Zhang. Fault diagnosis of rolling bearing based on itd fuzzy entropy and gg clustering. *Yi Qi Yi Biao Xue Bao/Chinese Journal of Scientific Instrument*, 35: 2624–2632, 11 2014.

[14] K. Yu, T. Ran, L. Ji, and W. Tan. A bearing fault diagnosis technique based on singular values of eemd spatial condition matrix and gath-geva clustering. *Applied Acoustics*, 121(2):33–45, 2017.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3): 264–323, Sept. 1999.

[16] J. Irani, N. Pise, and M. Phatak. Clustering techniques and the similarity measures used in clustering: A survey. *International Journal of Computer Applications*, 134:9–14, 01 2016.

[17] D. Xu and Y. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2, 08 2015.

[18] A. Ben Ayed, M. Ben Halima, and A. Alimi. Survey on clustering methods : Towards fuzzy clustering for big data. 2015.

[19] A. Stetco, X.-J. Zeng, and J. Keane. Fuzzy c-means++: Fuzzy c-means with effective seeding initialization. *Expert Systems with Applications*, 42(21):7541–7548, 2015. ISSN 0957-4174.

[20] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[21] I. Gath and A. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–780, 1989.

[22] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2012.07.021. URL `https://www.sciencedirect.com/science/article/pii/S003132031200338X`.

[23] X. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.

[24] Y.-I. Kim, D.-W. Kim, D. Lee, and K. Lee. A cluster validation index for gk cluster analysis based on relative degree of sharing. *Information Sciences*, 168:225–242, 12 2004.

[25] N. Pal and J. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems*, 3(3):370–379, 1995.

[26] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. ISSN 0019-9958.

[27] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[28] P. Fazendeiro and J. V. de Oliveira. A fuzzy clustering algorithm with a variable focal point. In *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, pages 1049–1056, 2008.

[29] P. Fazendeiro and J. V. de Oliveira. Observer-biased fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 23(1):85–97, 2015.

[30] J. Gao and D. Hitchcock. James-stein shrinkage to improve k-means cluster analysis. *Computational Statistics & Data Analysis*, 54:2113–2127, 09 2010.

[31] K. Pelckmans, J. D. Brabanter, J. Suykens, and B. D. Moor. Convex clustering shrinkage. In *Workshop on Statistics and Optimization of Clustering Workshop (PASCAL)*, pages 2113–2127, London, UK, 2005.

[32] A. Devillez, P. Billaudel, and G. Lecolier. A fuzzy hybrid hierarchical clustering method with a new criterion able to find the optimal partition. *Fuzzy Sets and Systems*, 128:323–338, 06 2002.

[33] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[34] D. Dubois and H. Prade. Fuzzy sets-a convenient fiction for modeling vagueness and possibility. *IEEE Trans. Fuzzy Syst.*, 2:16–21, 1994.

[35] C. Li, J. Valente de Oliveira, M. Cerrada, F. Pacheco, D. Cabrera, V. Sanchez, and G. Zurita. Observer-biased bearing condition monitoring: From fault detection to multi-fault classification. *Engineering Applications of Artificial Intelligence*, 50:287–301, 2016. ISSN 0952-1976.

[36] C. Li, L. Ledo, M. Delgado, M. Cerrada, R.-V. Sánchez, D. Cabrera, and J. V. De Oliveira. Gkfp: A new fuzzy clustering method applied to bearings diagnosis. In *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pages 1295–1300, 2018. doi: 10.1109/PHM-Chongqing.2018.00227.

[37] C. Li, , M. Cerrada, F. Pacheco, D. Cabrera, V. Sanchez, G. Ulutagay, and J. Valente de Oliveira. A comparison of fuzzy clustering algorithms for bearing fault diagnosis. *Journal of Intelligent & Fuzzy Systems*, 34:1–16, 06 2018.

[38] C. Li, L. Ledo, D. Cabrera, R.-V. Sanchez, M. Cerrada, L. Garcia-Hernandez, and J. Valente de Oliveira. A bayesian regularization for the fuzzy covariance matrix of ellipsoidal clusters. In preparation.

[39] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7:179–188, 1936.

[40] N. R. Smalheiser. Chapter 12 - nonparametric tests. In N. R. Smalheiser, editor, *Data Literacy*, pages 157–167. Academic Press, 2017. ISBN 978-0-12-811306-6.

[41] J. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[42] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010. ISSN 0167-8655. URL `https://www.sciencedirect.com/science/article/pii/S0167865510000954`.

[43] B. Balasko, J. Abonyi, and B. Feil. Fuzzy clustering and data analysis toolbox for use with matlab. 07 2014.

[44] H. Kwasnicka and P. Siemionko. Improved sammon mapping method for visualization of multidimensional data. volume 7654, pages 39–48, 11 2012. ISBN 978-3-642-34706-1.

[45] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv e-prints*, Feb. 2018.