VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wānanga o te Ūpoko o te Ika a Māui

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Internet: office@ecs.vuw.ac.nz

PSO for Simultaneous Feature Selection and Weighting in High Dimensional Clustering

Damien O'Neill, ID:300329342

Supervisors: Bing Xue, Mengjie Zhang, Andrew Lensen

Submitted in partial fulfilment of the requirements for
Bachelor of Science with Honours in Computer Science.

Abstract

Clustering is one of the most important unsupervised learning tasks, but it is very challenging on high dimensional data. In particular, a challenge known as the Curse of Dimensionality can make clusters significantly less meaningful as the dimensionality of a dataset increases. To address the Curse of Dimensionality in clustering, this project utilses Particle Swarm Optimisation to perform simultaneous feature selection and feature weighting, which is the first such approach presented in the literature. Two new internal validation measures are proposed to be used as optimisation criteria in the new approach, one of which represents the first clustering specific Bayesian validation measure proposed in the literature. Experiments show that this novel approach can successfully improve clustering performance relative to baseline algorithms while using fewer features. Further, the novel validation measures were demonstrated to able to avoid naive solutions that many distance based validation measures often produce on high dimensional data.

Acknowledgments

I would like to extend my deepest gratitude to my supervisors Bing Xue, Mengjie Zhang, and Andrew Lensen. This work absolutely would not exist without their profound knowledge, endless encouragement, and exceptional guidance.

I would also like to thank my mother, Glenda, and my partner, Rebekah, for their tireless support throughout all areas of my life.

Finally I would like to thank my broader family and friends, who always make time for me, and have been understanding throughout the year when I've been unable to do the same.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unsupervised learning techniques are important in both academic and industry settings due to their ability to work with unlabelled data. As such, their use is common in areas such as pattern discovery and data mining [5]. Within unsupervised learning, clustering is one of the most fundamental tasks [18, 54]. Clustering can be described as the task of partitioning a dataset into groups such that elements within groups are related, and elements between groups are comparatively unrelated [3, 5, 18, 54].

However, there is no strict agreement as to what it formally means for elements to be related or unrelated. This particular difficulty has been summarised aptly by Backer and Jain [3] who write "in cluster analysis a group of objects is split up into a number of more or less homogeneous subgroups on the basis of an often subjectively chosen measure of similarity (i.e., chosen subjectively based on its ability to create "interesting" clusters), such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups".

A measure designed to evaluate a cluster partitioning is named a *validation measure*, and the ambiguity in what it means for a partitioning to be good in cluster analysis has led to the proposal of many cluster validation measures in the literature [53, 1].

Some examples of cluster validation measures include: *compactness* [1], which measures of how 'close together' the data within clusters are; and *separation* [1], which measures how 'far away' different clusters are from each other. Because of the inherent subjectivity of what it means for a partitioning of data to be good, and the abundance of the proposed validation measures in the literature, finding a suitable validation measure for a given clustering algorithm and dataset is challenging.

A further complication when performing clustering relates to the dimensionality of data. High dimensionality in datasets can significantly reduce the ability of clustering algorithms to find meaningful relationships, through a number of characteristics known collectively as the Curse of Dimensionality [6, 32]. Thus, for many high dimensional applications of clustering, dimensionality reduction techniques are used in conjunction with a clustering algorithm in order to address these issues [46]. However, dimensionality reduction techniques can hinder the interpretability of results in some cases, such as when feature construction methods are used, due to the fact that feature construction creates new features to represent the original features, which obscures the role of the original features.

A key class of dimensionality reduction algorithms which improve the interpretability of results is known as feature selection [24]. Feature selection attempts to find a subset of the original features which give similar or improved results over the case when the full feature set is used.

The search space of feature selection consists of all possible subsets of features. For a dataset with dimensionality d, there are a total of 2^d possible feature subsets. This large

search space means that frequently feature selection methods utilise some form of heuristic search throughout the space; utilising sequential forward selection (SFS)[52], sequential backward selection (SBS)[36], or some variation of these methods [41] to find an appropriate feature set. Such methods frequently fail to find optimal feature sets as interactions between features are not well considered in these cases. Specifically it has been shown that no heuristic feature selection search can ever guarantee optimality [51].

To address this concern recent literature has been dedicated to global search methods for feature selection [56]. Promising results have been found by utilising evolutionary computation (EC) methods such as particle swarm optimisation (PSO) [56], although the majority of such work has been performed in a supervised learning setting [56].

By only looking for which features could be included in the final feature set, feature selection algorithms may ignore potential improvements that could be gained from appropriately weighting features, however. Thus the problem of feature weighting should also be considered, driven by a clear intuition that more important features should be weighted more heavily than less important features.

Further, where distance based clustering algorithms are used in prior work, distance functions have been one of only a few standard distance functions (E.g. Euclidean or Manhattan distance). The question of whether or not these distance functions are optimal for a given problem is scarcely addressed, and never addressed empirically.

1.1 Motivations

In the existing literature several competing distance based validation measures for assessing the quality of clusters have been presented [1, 35, 42]. The lack of clear consensus in the literature regarding the value of these different validation measures means that there is benefit to testing these measures in the context of various datasets.

Further, where statistical validation measures have been used in clustering, the validation measures have been applications of broader model selection methods, usually utilising a Bayesian framework [20, 10]. Thus the development of clustering specific statistical validation measures presents a clear opportunity to advance the research field.

Additionally, many validation measures present naive solutions if used as the optimisation criteria during the clustering process, e.g. compactness holds a trivially optimal solution when there is a singular arbitrarily compact cluster. These clusters can be said to violate one of the key criteria in clustering, namely that discovered clusters be "interesting" [3]. Thus, creation of validation measures which do not lead to trivial clustering solutions is not only a fundamental requirement for the current project but also a valuable avenue of research.

Prior work which has aimed to reduce the problems encountered when clustering high dimensional datasets have utilised both feature construction methods and feature selection methods.

Feature construction methods tend to hinder interpretability of the results by obscuring the relationship that any particular feature may have to the partitioning. Given that interpretability of results is a key goal in pattern discovery and data mining tasks, and further is considered a requirement for models used in many 'mission critical' systems [9], this lack of interpretability is a significant limitation for many current dimensionality reduction techniques.

Where feature selection methods, which tend to preserve or improve interpretability, have been used in the context of clustering the application has typically utilised local search methods, which are prone to overlooking interactions between features [51].

Where global search techniques have been used for feature selection they have tended to improve clustering outcomes while improving interpretability of clusters [11, 51], and feature weighting has shown benefit in some unsupervised learning tasks [37], however there is no existing literature regarding simultaneous feature selection and weighting in the context of clustering. The current project is premised on the idea that simultaneous feature selection and weighting may offer a more powerful method to significantly improve clustering outcomes while improving interpretability of results, and seeks to provide cursory tests along these lines. Further, if it is the case that feature selection and weighting do significantly improve clustering outcomes, then extending this method to also find a unique distance function for a dataset, which can be non-uniform in how it addresses the distance between different features, could further improve results.

Finally, there is a diversity in clustering algorithms, with many clustering algorithms operating under very different assumptions [18]. This means that, in general, some clustering algorithms may be significantly more suited for a given dataset. Thus this project seeks to demonstrate any improvements gained from the novel methods on a variety of different clustering algorithms.

1.2 Goals

This project seeks to address issues that arise when clustering high dimensional datasets, while improving the interpretability and quality of resulting clusters and providing insight as to how important different features are to the dataset as a whole.

In order to achieve this, validation measures which do not lead to naive solutions when used as optimisation criteria must be established. This project seeks to develop two novel validation measures, one distance based and the other statistical. A distance based validation measure is being utilised to make use of the existing literature on distance based validation measures, as while many distance based validation measures have been proposed previously, they have not been examined as optimisation criteria for feature selection and weighting.

A statistical validation measure will be proposed due to the success of previous work in using broad model selection criteria as validation measures in clustering [20, 10]. A statistical validation measure which is developed explicitly for clustering has the potential to improve on these existing methods, and will also act as exploratory research, extending the literature by considering how Bayesian probability can be applied specifically to the problem of clustering.

After establishing these measures this project seeks to create a novel approach to feature selection and weighting in clustering utilising PSO. Prior work indicates that both feature selection and weighting can improve clustering results, over the respective cases where the full and unweighted feature sets are used. However, in prior work only one of these two methods have been applied, and thus no prior work has investigated a combined feature selection and weighting method in clustering.

Further, an extension to this method will be considered to also find exponents with which pairwise distance is calculated. This extension is driven by the fact that very little work has been done on constructed distance measures in clustering, meaning that this method will extend the literature.

PSO will be utilised because it has been shown to be a broadly applicable global search technique, especially for continuous problems, which generates highly interpretable solutions.

Once these methods are developed this project seeks to examine the behaviour of the

methods when applied to a variety of different clustering algorithms, specifically aiming to improve statistical measures of the correctness of the returned clusters while using only a subset of the features.

Thus, the overall goal of this research is to develop a broadly applicable method for simultaneous feature selection and feature weighting in clustering using PSO, investigating the use of internal validation measures as optimisation criteria within this framework.

The specific goals of this research are to:

1. Create two novel optimisation criteria, one distance-based and one statistical, for feature selection and weighting in clustering problems,

2. Create a novel PSO method for simultaneous feature selection and weighting in clustering, and

3. Investigate the generality of the novel algorithms and optimisation criteria by applying them to several different classes of clustering algorithms over several non-trivial datasets, and to test whether they can achieve better clustering performance with a smaller number of features.

Further, an extension to the novel PSO method will be investigated, although it does not constitute a primary research goal.

1.3 Major Contributions

This project has a number of major contributions:

1. This project has developed the first method for simultaneous feature selection and feature weighting in clustering. In particular, a novel PSO method was established which performs feature selection and feature weighting as a wrapper approach for a variety of distance based clustering algorithms. Tests performed by clustering widely used ellipsoid datasets show that the novel algorithm significantly increases clustering performance, while using approximately half of the available features, when compared with baseline methods.

2. This project has developed a novel *statistical* internal validation measure, the first measure utilising a statistical formulation of cluster separability. In particular, a clustering specific internal validation measure was derived from Bayesian probability. The measure was shown to be able to avoid naive solutions that many distance based validation measures often produce on high dimensional data, improving clustering performance in the majority of test conditions relative to baseline when used with the novel PSO approach.

3. This project has developed a novel *distance based* internal validation measure which extends existing distance based validation measures. In particular, empirical results were used to inform the construction of a distance based internal validation measure which combines two existing effective measures. This measure was shown to improve clustering outcomes relative to baseline in all test conditions when used with the novel PSO approach.

4. The existence and interpretation of naive solutions in clustering was investigated. In particular, the behaviour of clustering algorithms in a feature selection and weighting

framework was examined when using existing internal validation measures as optimisation criteria. The investigation demonstrated that existing internal validation measures were not robust to naive solutions in this framework, returning cluster partitions which trivially maximised the optimisation criteria while baring little resemblance to the base truth.

Part of the research done in this project is under preparation to be submitted to the 2018 IEEE World Congress on Computational Intelligence/IEEE Congress on Evolutionary Computation (WCCI/CEC2018).

1.4 Report Organisation

The remainder of this report is organised as follows. Chapter 2 provides background to the current project, as well as detailing methods which will be used throughout the work. Chapter 3 discusses the choice of datasets, details the dataset generation procedure, and demonstrates some characteristics of the datasets. Chapter 4 addresses Goal 1, while Chapters 5 and 6 together address goals 2 and 3. Chapter 7 presents the major conclusions from this project as well as providing several avenues for future research.

Chapter 2

Background

This chapter introduces both the background to the current project, as well as outlining several algorithms which will form much of the basis for the current work. In particular a broad overview of Machine Learning is presented, as well as work relating to specifically to Clustering, the Curse of Dimensionality, Dimensionality Reduction, and Evolutionary Computation. Further, algorithms and methods used directly in this work are presented, and related work is considered. Where related work is considered we note that the current project is novel, as no current method exists for feature selection and weighting in an unsupervised learning context. Thus much related work is related only tangentially.

2.1 Machine Learning

Machine learning [2] refers to a collection of methods in which programs learn from data in order to achieve some tasks, rather than being explicitly coded. The tasks to which machine learning algorithms can be applied are numerous, but three main paradigms are presented in the literature [43]:

- Supervised learning, where both inputs and target outputs are known,
- Reinforcement learning, where programs attempt to find outputs which optimise some reward function, and
- Unsupervised learning, where target outputs are unknown, and the goal is to discover some underlying pattern in a dataset.

Within each of these paradigms there are several notable algorithms. Some key algorithms include artificial neural nets (ANNs), decision trees, Q-Learning, k-means clustering, and Expectation Maximisation [2].

2.1.1 Supervised Learning

Supervised learning consists of algorithms which seek to match data to some known target output [43]. The two broad categories of supervised learning tasks are classification and regression. Classification seeks to map inputs to some known class label, and typically learns by attempting to minimise the error rate of classification [43]. Regression, however, seeks to map inputs to some real valued target output, and learns by minimising an error function such as the mean-squared error [43].

Within different algorithms the notion of learning holds different meanings. For example, ANNs often learn by finding the optimal parameters to a structurally pre-defined

mathematical function, whereas a decision tree is learned by finding both the correct structure and parameters for a model.

A goal for all supervised learning algorithms is to ensure that a model generalises to unseen data. As such, datasets are normally split into training and test sets, with models trained on training sets but assessed on the unseen data in the test set.

2.1.2 Reinforcement Learning

Reinforcement learning is a machine learning paradigm concerned with designing a decision making process such that an agent using that decision making process optimises some reward criteria [48], and is frequently used when modelling agents in games [30]. A canonical algorithm within reinforcement learning is Q-Learning. Traditional Q-Learning considers a series of states, with a set of potential actions which an agent can select that lead to different successor states. It then attempts to learn which is the optimal action for each state representation by assigning values to actions from prior experience [48, 30].

2.1.3 Unsupervised Learning

Unsupervised learning contains algorithms for a number of different tasks, all addressing situations whereby a dataset has no known target output. This difference distinguishes unsupervised learning from supervised learning, where target outputs from the dataset are known [4].

There are numerous data mining tasks which utilise unsupervised learning techniques [4], including text mining [23], approximating latent variable models [7], and clustering [54, 18, 5]. A unifying quality among these tasks is that, as there is no known target output, each of these tasks are seeking to extract meaningful information from a given dataset.

Text mining, for example, seeks to automatically extract novel information from different written resources [23]. Latent variable models, on the other hand, assert that all datapoints are being generated by some unseen latent generative probability distributions, and seek to find these distributions [7].

An example of a latent variable model algorithm is Expectation Maximisation (EM). EM is an unsupervised learning algorithm which, for some assumed number of generating distributions and a given probability function, finds parameters such that the distributions best explain the dataset. These distributions are used to inform users as to the underlying probabilistic structure behind the data without using class labels.

Within these tasks, however, clustering is among the most important [54, 18, 5].

2.2 Clustering

Given an unlabelled dataset clustering seeks to find a partitioning of the data such that datapoints within partitions are related, and datapoints between partitions are comparatively unrelated. Further, clustering aims to find partitionings which are non-trivial [3].

Given that the criteria for what can be considered a good partitioning is only defined vaguely in the literature, several attempts have been made to formalise the assessment of partitions. These assessments are named validation measures.

2.2.1 Validation Measures

Validation measures are used in the context of clustering to assess the validity of a given set partitioning of a dataset. They can be separated into two categories, internal and ex-

ternal validation measures [53]. Internal validation measures assess the validity of clusters using only the data available to the clustering algorithm, such as spatial characteristics of the clusters, whereas external validation measures make use of information external to the clustering algorithm, such as the base truth. External validation measures are thus used to provide a valuation of a partitioning when the base truth is known, and thus can demonstrate more objectively the performance of a given clustering algorithm.

For this project several internal and external validation measures are researched and implemented in order to assess their performance and to inform the choice for which internal validation measure seems promising as an optimisation criterion for the novel PSO algorithms on the chosen type of datasets. The implemented validation measures are listed and described below.

2.2.1.1 Distance Based Internal Validation Measures

- Silhouette: the silhouette [42] of a given datapoint, i, is defined in terms of the functions $a(i)$ and $b(i)$, where $a(i)$ is the average distance between the datapoint i and all other datapoints in the same cluster, and $b(i)$ finds, for each other cluster, the average distance between a datapoint i and all datapoints within that cluster, returning the minimum of these values. Given these functions the silhouette of a given datapoint is calculated as $sil(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$, and the silhouette for a set of clusters is considered to be the mean of silhouettes of all datapoints, reflecting on average how close points tend to be to other datapoints within their cluster and how far away datapoints tend to be from datapoints in the closest neighbouring cluster. The range of values possible for this measure are $[-1, 1]$ where higher values indicate better clusters.

- Connectedness: the connectedness [1] of a given datapoint i belonging to a cluster C is defined using a given number, n, of closest neighbours, and a specified maximum absolute distance m. Specifically connectedness is defined by the equation

$$conn(i) = \sum_{k=1}^{n} \begin{cases} min(\frac{1}{d(i,k)}, m) & if\ k \in C \\ -min(\frac{1}{d(i,k)}, m) & if\ k \notin C \end{cases}$$

  Thus connectedness looks at the closeness of neighbours of a given datapoint, assigning a high positive value to neighbours which are close and within the same cluster, and a high negative value to neighbours which are close but are not part of the same cluster. The value m provides a practical limit on these values such that no one distance can dominate the overall sum. The connectedness of a set of clusters is the mean connectedness of all datapoints. The possible values for this measure are $[-m * n, m * n]$, with higher values indicating a more appropriate partitioning on a local level. For this work parameters $n = 5$ and $m = 10$ were found to be suitable, giving outputs in the range $[-50, 50]$.

- Sparsity: the sparsity [1] of a cluster is a measure of how isolated datapoints are within the cluster. Ideally within a cluster all points are close to at least one other datapoint in the cluster. Thus, sparsity is defined as the maximum distance over a datapoint and their nearest neighbour within a cluster, and a lower sparsity is considered beneficial. Formally, given a cluster C and distance function $d(x, y)$,

$$\begin{aligned} Sparsity(C) \quad &= \quad max(min(d(p_1, p_2))), \\ where \quad p_1,\ p_2 \in C, \quad &p_1 \neq p_2 \end{aligned}$$

  The sparsity of a set of clusters is considered to be the mean sparsity over all clusters.

- Separation: the separation [1] of a cluster is a measure of how far away its boundary is from all other clusters. Separation is formally defined as the minimum distance between datapoints within a given cluster and all datapoints within other clusters, and a higher separation is considered beneficial. Formally, given a cluster C, points belonging to all other clusters $U \backslash C$, and a distance function $d(x, y)$,

$$\begin{aligned} Separation(C) \quad &= \quad min(min(d(p_1, p_2))), \\ where \quad p_1 \in C, \quad &p_2 \in U \backslash C \end{aligned}$$

  The separation of a set of clusters is considered to be the mean separation over all clusters.

2.2.1.2 Statistical Internal Validation Measures

While no specific statistical internal validation measures have been made for clustering tasks, general model selection criteria have previous been used as internal validation measures in clustering with some success [20, 10]. In particular, the Bayesian approach known as the Bayesian Information Criterion (BIC) has been used to compare different partitionings under a clustering framework [20, 10]. BIC is utilised when selecting between a finite number of models, and is defined as:

$$BIC = ln(\hat{L}) - \frac{ln(n)k}{2}$$

Where $\hat{L}$ is the likelihood of data given the model, n is the number of datapoints, and k the number of parameters on which the model depends [45]. Thus, when finding a model maximising the BIC, one is finding a model which explains the data well but avoids excessive complexity. However, this measure ignores an important aspect of clustering, namely that datapoints should not only be explained well by the cluster they are assigned to, but also not be explained well by neighbouring clusters.

2.2.1.3 External Validation Measures

External validation measures are used to evaluate how well cluster partitions relate to the base truth. Thus, external validation measures provide a more objective way of measuring the performance of clustering algorithms, removing the subjectivity in cluster analysis noted by Backer and Jain [3]. However, they are only usable on datasets where the base truth is known, such as on classification datasets and synthetic clustering datasets. Two key external validation measures used in cluster analysis are:

- Purity: the purity [1] of a cluster is defined as what fraction of points within a cluster belong to the majority label for that cluster. Ideally clusters contain only datapoints from one class, thus having a purity of 1. Formally given a cluster C and a set true labels L, which itself contains a set of points for each label, this can be defined by $Purity(C) = \frac{1}{|C|} max_{l \in L}(C \bigcup l)$. The purity of a set of clusters is defined as the mean purity over all clusters.

- F-Score: the F-Score of a cluster is defined as the square root of the product of precision and recall, and tends to be preferred over purity because the consideration of recall prevents trivially high values when the number of clusters approaches the number of instances. Specifically, in clustering, pairwise comparisons are made between every point in the dataset in order to find the number of True Positives (TP), False Positives

(FP), and False Negatives (FN). For each pairwise comparison a TP is said to be where two points share a label and are also in the same cluster, a FP is when two points are within the same cluster but do not share the same label, and a FN is when two points share a label but are not in the same cluster [33]. We then define precision and recall as is standard, $recall = \#TP/(\#TP + \#FN)$, $precision = \#TP/(\#TP + \#FP)$. Giving us our final definition of the F-Score as defined by Fowlkes and Mallows [19], $F = \sqrt{precision * recall}$.

The F-Score is used as the key external validation measure in the current project, and informs claims as to the objective performance of clustering algorithms.

2.3 Distance Metrics

Distance metrics are used in clustering to formalise the notion of distance (or dissimilarity) between two points. Intuitively, points with low distance between each other should normally lie within the same cluster, and points which have comparatively high distance between them should normally lie within different clusters, because they have values which are less similar.

Distance metrics, formally, are functions mapping two vectors of equal length to a single non-negative real-valued output, formally notated $d : X \times X \rightarrow [0, \infty)$, such that the following properties hold:

- $d(x, x) = 0$,
- $d(x, y) = d(y, x)$, and
- $d(x, y) \leq d(x, z) + d(z, y)$

Examples of commonly used distance metrics in clustering include the Manhattan distance ($\sum_{i=1}^{n} |p_i - q_i|$) and Euclidean distance ($\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$), where p and q are vectors of dimension n.

2.4 The Curse of Dimensionality

The Curse of Dimensionality is a term applied to group of problems that arise when working with high dimensional data. One aspect of the Curse of Dimensionality defined with respect to distance metrics in higher dimensional space has been formalised by Beyer et al.[6] which is the aspect of the Curse of Dimensionality which this project hopes to address.

The formalisation of this property put forward by Beyer et al. [6] hinges on the following observation, which was proven in the original work: for many distributions of dimension d, given an arbitrary point in the dataset x and finding a point y which satisfies $argmax_y(d(x, y))$ and a point z which satisfies $argmin_z(d(x, z))$, the $lim_{d\rightarrow\infty} \frac{dist(x,y)-dist(x,z)}{dist(x,z)} \rightarrow 0$. This is to say, while all distances tend towards infinity as dimensionality tends to infinity, the maximum and minimum distance between all points tends towards equidistance. It was also found that strong evidence for this behaviour can be seen by the time $d = 10$ for many synthetic and real-world datasets [6].

This causes a problem in the context of clustering because it implies that distance functions cease to provide meaningful insight about how related two datapoints actually are in clustering problems as sufficient dimensionality. Research has supported this implication, showing that this aspect of the Curse of Dimensionality significantly impacts the effectiveness of clustering algorithms [47].

2.5 Evolutionary Computation

Evolutionary Computation (EC) denotes a group of population based algorithms broadly inspired by biological behaviours, which are widely used for global optimisation and tend to make very few assumptions about the properties of the data on which they operate, while being robust to local optima.

Two major paradigms present themselves in EC literature [56], namely evolutionary algorithms and swarm based algorithms, although many algorithms which fit into neither of these paradigms have been proposed, such as learning classifier systems [56].

Evolutionary algorithms tend to create a random initial population of candidate solutions for the target problem and then perform the following operations until some termination criteria is met [56]:

- Elitism, where individuals are created by copying members from the previous population, with a bias given to copying 'high fitness' members,
- Mutation, where individuals have characteristics randomly changed, in order to explore the search space of solutions more widely, and
- Crossover, where two individuals 'breed', producing two new members which share attributes of both parents.

The most widely used evolutionary algorithms are genetic algorithms (GAs) and genetic programming (GP). While both of these methods utilise a framework broadly agreeing with the above, they contain very different representations of individuals in the population. GA treats candidate solutions as bit-strings, which are interpreted according to some user defined function, whereas GP evolves program trees, which take inputs and returns some value. This distinction leads to many conceptual and practical differences between GA and GP, a key point being that GA is defined on bit-strings of a fixed length, while GP can create program trees of various sizes.

Swarm based algorithms, on the other hand, tend also to begin with a random population of candidate solutions, but instead of the three operators used in evolutionary algorithms, they at each iteration of the algorithm update each member of the candidate population directly [56]. Normally these updates are to push each candidate solution closer to where promising solutions have been found by other members, but some stochastic movement is also frequently used to ensure sufficient exploration of the search space.

2.5.1 Particle Swarm Optimisation

Particle Swarm Optimisation [8] (PSO) is an EC, swarm based, algorithm inspired by social behaviour in animals, namely bird flocking and fish schooling. Specifically it creates a population of candidate solutions, called the swarm, where each candidate solution, denoted a particle, is a vector. PSO then seeks to find an optimal solution in the given search space by balancing stochastic movement, local knowledge of the best solution so far, and global knowledge of the best solution so far.

Specifically, a random d dimensional particle $x_i \in \mathbb{R}^d$ is first assigned a random velocity $v_i \in [-v_{max}, v_{max}]$, where v_{max} is a user specified maximum velocity. Throughout the algorithm each particle maintains a record of its previous best position *pbest* and has access to the recorded global best position *gbest*. For each iteration of the PSO algorithm the following updates are made, given a user specified inertia weight ω, a user specified acceleration coefficient for *pbest* and *gbest*, denoted c_1 and c_2 respectively, and a function r which returns a uniform random value in $[0, 1]$:

$$v_i^{t+1} = \omega v_i^t + r_1 c_1 (pbest_i - x_i^t) + r_2 c_2 (gbest_i - x_i^t) \tag{2.1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2.2}$$

Once this update is complete for all particles in a given iteration values for *pbest* and *gbest* are updated.

After the maximum number of iterations have been reached, or some other termination criterion has been met, the value for *gbest* is returned as the solution.

2.6 Dimensionality Reduction

To address the curse of dimensionality in both machine learning and broader data analysis several algorithms have been developed which reduce the dimensionality of a dataset.

2.6.1 Feature Selection

Feature selection methods are a broad class of methods utilised to reduce the dimensionality of datasets [24]. Feature selection refers to the task of finding a subset of features from the original feature set in such a way that the dimensionality of the dataset is reduced, with the goal of improving results for a given task using this subset relative to using the full set of features.

The search space of feature selection methods consists of all possible subsets of features. For a dataset with dimensionality d this means that there are a total of 2^d possible feature subsets. Accordingly, many early feature selection methods utilised local search to find a good subset of features. The two main approaches for local feature selection search are sequential forward selection (SFS)[52], sequential backward selection (SBS)[36]. SFS begins with an empty feature set and, at each iteration of the algorithm, adds another feature to this subset according to a pre-defined measure until the algorithm terminates. SBS mirrors this approach but starts from the full feature sets and removes one feature at each iteration. In both of these cases the runtime is reduced from the exhaustive $O(2^d)$ to $O(d^2)$, or $O(dn)$ where a maximum number of iterations n is specified, making sequential feature selection a tractable approach to feature selection.

Further developments of feature selection have been demonstrated as extensions of the above ideas. In particular floating search methods [41] were demonstrated to generally outperform SFS, SBS, as well as some other extensions to SFS and SBS [44]. Floating search methods, as with SFS and SBS, start from either the set of all features or an empty set of features, however they contain steps at which they can both add beneficial features and remove the least beneficial features from the feature subset. Thus these algorithms incorporate elements of both forward and backward sequential selection [41].

However, local searches, even floating search methods, are prone to finding suboptimal feature subsets, as they naturally omit many interactions between features. Further, it has been shown that performing feature selection via local search can never guarantee optimality in the general case [51]. Thus much recent literature on feature selection methods have shifted to global search techniques, the most promising of which derive from Evolutionary Computation (EC).

EC methods for feature selection are typically used in a supervised learning context [56]. Specifically, the use of EC methods to perform dimensionality reduction in classification tasks has been well established in the literature, with the most common methods being

GAs and PSO [56]. Further, within the literature regarding the use of GAs and PSO to perform feature selection in classification, these techniques are predominantly used as wrapper methods [56].

The standard representation of candidate feature sets in both GAs and PSO is a binary string, with the length of this binary string equal to the total number of features. For each bit in the bit string, a value of one indicates that the corresponding feature has been selected, whereas a value of 0 indicates that a feature has been excluded.

While both GAs and PSO, when used in this context, tend to improve classifier accuracy while using a reduced feature set, by intuition GAs are likely to outperform PSO when a dataset has groups of interacting features. Thus is due to the building blocks in GAs being suited to combinatorial problems, whereas PSO tends to be more suited to problems in which the search space of solutions is continuous [56].

PSO, when used as a feature selection method, has improved outcomes and interpretability on classification tasks in complex biological datasets [11] and also on widely used machine learning datasets [51].

2.6.2 Feature Construction

Feature construction is another key method used to reduce the dimensionality of datasets. Feature construction refers to methods which construct a feature or features as a function of the original features [39]. These can reduce the dimensionality of a dataset by creating new features for the data which are combinations of only a subset of the original feature set, and then using only these constructed features to perform the original task, e.g. classification. A canonical example of a feature construction technique is Principle Component Analysis (PCA). PCA is a foundational statistical feature construction method used to reduce the dimensionality of the feature set of a dataset while retaining as much information in the dataset as possible. PCA works by constructing a new basis for the dataset according to which weighted combinations of features most account for variance in the dataset [40]. PCA has a long history of use as a data preprocessing technique, but significantly reduces the interpretability of results, as the new basis for the dataset fundamentally changes the feature space without providing a clear idea of the relationship between specific original features and the outcome. This means that PCA is less useful in unsupervised learning tasks where interpretability is a goal, such as pattern discovery and data mining. Further, as PCA tries to find a basis such that each principle component accounts for as much variance in the data as possible it can encounter difficulties where some dimensions of the data hold high variance noise.

Further the search space for constructed features can be arbitrarily large, and thus local optima can be problematic for feature construction methods in general.

2.6.3 Filter, Wrapper, and Embedded Methods

All dimensionality reduction algorithms can be categorised broadly into three overall methodologies: filter, wrapper, and embedded. This categorisation reflects how feature subsets or constructed features are being assessed during selection or construction.

Filter methods are dimensionality reduction techniques which apply dimensionality reduction to the dataset according solely to properties of that dataset [56]. Filter methods tend to be general, producing feature sets that are applicable to many different algorithms, and comparatively computationally inexpensive [56]. However, this generality means that filter methods do not normally create feature sets which are optimal for a given task, because

they are unable, by definition, to assess how the selected or constructed features will interact with specific algorithms.

Wrapper methods, however, apply dimensionality reduction according to the outcomes of feature sets when they are actually used by a chosen algorithm for a given task. Thus, wrapper methods tend to be specific, producing feature sets which are optimised for a specific algorithm and task. Further, wrapper methods tend to be computationally expensive, requiring that each potential feature set be utilised for the task at hand in order to assess its performance. However, wrapper methods, accordingly, tend to outperform filter methods, due to selecting or creating feature sets which best suit a given environment.

Embedded methods refer algorithms which, as part of their design, automatically perform dimensionality reduction. Because of the variety in algorithms which are said to utilise embedded dimensionality reduction methods it is difficult to ascribe specific performance characteristics to this category, however it has been indicated [44], that these methods tend to produce feature sets with middling generality, with middling optimality, and at middling computational cost. An example of such an algorithm is genetic programming in classification tasks, which naturally selects a subset of features when constructing solutions.

2.7 Related Work

This section describes a number of important clustering algorithms, and considers existing work which is important to the current project.

2.7.1 Clustering Methods

In the literature there are numerous clustering algorithms, both non-EC based and EC based [28, 18]. This subsection introduces a relevant taxonomy for non-EC clustering algorithms, which allows the current work to ensure that testing done utilising the novel method does so using a variety of established clustering algorithms, as per Goal 3. Existing methods regarding EC based clustering algorithms are also introduced, due to the relevancy of existing work done on encoding schemes for these EC algorithms.

2.7.1.1 Non-EC Clustering Methods

In order to demonstrate a general methodology as per Goal 3, several distance based clustering algorithms need to be considered. Fahad et al. [18] propose a taxonomy for clustering algorithms consisting of the following five overall classes:

- Partitioning-Based,
- Hierarchical-Based,
- Density-Based,
- Grid-Based, and
- Model-Based

Grid-based and model-based are statistical clustering techniques [18], and thus are not applicable to this project, of which a key motivation is addressing issues relating to distance on high dimensional datasets. Further, the current work proposes the addition of an appropriate graph-based clustering algorithm for additional comparison. We thus propose the use of the following four algorithms:

1. Partitioning-based: Affinity Propagation [21]
2. Hierarchical-based: Complete linkage agglomerative clustering [12]
3. Density-based: DBSCAN [17]
4. Graph-based: KNN-neighbour clustering [50]

Each of these will be described in detail below.

Affinity Propagation

Affinity Propagation is a medoid-based clustering algorithm. Medoids are datapoints which are treated as exemplars for a cluster, and datapoints are assigned to clusters based on similarity to medoids. Affinity Propagation automatically selects both the number of medoids to choose for a dataset, and which datapoints will act as medoids. Affinity Propagation has been demonstrated to improve clustering outcomes relative to K-Means on several complicated datasets, including high dimensional biological datasets, computer vision datasets, and routing datasets [21].

Affinity Propagation finds clusters based on similarity between points, rather than distance, but the convention of using negative squared Euclidean distance as similarity is well established, making this a suitable algorithm for this project.

The specific method for Affinity Propagation, given a similarity matrix S, is:

1. Two square matrices R and A are initialised to arrays of zeroes, where $R_{x,y}$ indicates how appropriate datapoint x is to act as a medoid for datapoint y, and $A_{x,y}$ indicates how appropriate it is for datapoint x to pick y as a medoid considering the appropriateness of y as a medoid for other points,
2. Until convergence criteria are met, for each iteration and all pairwise combinations of datapoints the following updates take place:
   (a) $R_{x,y} \leftarrow S_{x,y} - max_{z \neq x}\{A_{x,z} + S_{x,z}\}$
   (b) $A_{x,x} \leftarrow \sum_{y \neq x} max(0, r(x,y))$
   (c) For $A_{x,y}$ where $x \neq y$, $A_{x,y} \leftarrow min(0, r(y,y) + \sum_{z \notin \{x,y\}} max(0, R_{z,y}))$
3. Finally medoids are extracted from the matrices R and A, where a datapoint x is considered a medoid where $R_{x,x} + A_{x,x} > 0$, and points are assigned to a cluster corresponding to their nearest medoid.

The significance of the term $R_{x,x} + A_{x,x} > 0$ is that it indicates that it is sufficiently appropriate for a datapoint x to act as a medoid for itself, and sufficiently appropriate for x to assign itself to a cluster for which it is the medoid.

KNN-Clustering

The KNN clustering algorithm is a graph-based clustering algorithm which performs the following steps given a user specified $K \in \mathbb{N}$ [50]:

1. Each point is connected to the K points which are closest to it, according to some distance metric, via an undirected edge, and
2. Clusters are then created by assigning points to clusters such that for each two points if a path exists between them then they are assigned to the same cluster.

For the datasets used in this research a K value of 3 was found to create the best clusters according to the validation measures used, and so the $3NN$ clustering algorithm acts as the graph-based clustering algorithm for this project.

Density-based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN [17] is a popular density based clustering algorithm which can find arbitrarily shaped clusters of relatively uniform density given a user specified $\epsilon \in \mathbb{R}^+$, $minNeighbours \in \mathbb{N}$. Specifically using a notion of core points, which are datapoints with at least $minNeighbours$ datapoints within ϵ distance of them, the following method is used:

1. All datapoints within ϵ of a core point are said to be directly reachable from that core point,
2. Any datapoint which is directly reachable from a core point is said to be in the same cluster as that core point, and
3. Any datapoint which is not within ϵ distance of a core point is said to be an outlier, and is not treated as belonging to any cluster.

DBSCAN is unique among the algorithms presented here for the final property that datapoints can remain unlabelled, which requires special consideration in the context of our validation measures. Specifically, with regards to the F-Score, datapoints designated as outliers by DBSCAN contribute to the False Negative count, but not the False Positive count.

Agglomerative

Agglomerative clustering methods are a standard clustering method where, given a specified number of clusters, $K \in \mathbb{N}$, the following method is performed:

1. At the start of the algorithm each datapoint is treated as a singleton cluster, and
2. Clusters are gradually merged according to some criteria until the number of clusters is K.

The specific criterion used in this work is complete linkage, whereby the clusters merged at each step are those with the least maximum distance between all datapoints in the clusters. Agglomerative clustering using this criterion can find arbitrarily shaped clusters, but is highly reliant on a priori knowledge regarding the true number of clusters.

2.7.1.2 EC Based Clustering Methods

The clustering literature also presents several cases where EC techniques are used to perform clustering directly. The most common EC techniques used to cluster directly are GAs and PSO [16, 28]. A survey put forward by Hruschka et al. [28] describe three common encoding schemes which are used in most of these algorithms:

- Centroid based encoding schemes, where each candidate solution represents the coordinates of the centroids which are used to partition the dataset according to which centroid is the closest to each datapoint. These encoding schemes require the number of clusters, K, to be predefined, and each candidate solution is then a vector of size Kd, where d is the dimensionality of the dataset. For example, on a two dimensional dataset, with $K = 2$, the candidate solution $[1.0, 0.8, -2.0, -1.0]$ indicates two centroids, one based at position $(1.0, 0.8)$ and the other at position $(-2.0, -1.0)$. Points

would then be partitioned into two clusters according to which of these centroids they were closest to, normally using Euclidean distance.

- Medoid based encoding schemes, where each candidate solution represents whether or not given instances have been selected as medoids. Thus, for a dataset with n instances, each candidate solution is a vector of length n. Where these representations are continuous an interpretation of a candidate solution involves defining some value above which points are treated as medoids. Where representations are binary this normally corresponds to a value of one representing a point being a medoid. After this interpretation, the dataset is partitioned according to which medoid datapoints are closest to, usually according to Euclidean distance, with medoids themselves trivially being their own nearest neighbour.

- Labelling based encoding systems again use candidate solutions of length n, where n is the number of instances in the dataset, but each value in the vector corresponds to an integer cluster label for a point.

These works, while distinct in aim from the current project, provide evidence of the versatility of encoding schemes in EC algorithms. We note, however, that the non-EC clustering algorithms presented are both more researched at the present time and have presented strong results across a variety of datasets [18, 54]. Further, the encodings here do not improve the interpretability of partitions and do not address fundamental concerns regarding the Curse of Dimensionality, which is the second Goal of the current project. We further note that these encoding schemes, if one assumes that $n >> d$, represent very large search spaces, which can be reduced by encoding the EC algorithm directly for dimensionality reduction and using it as a wrapper approach for non-EC clustering methods.

2.7.2 Feature Selection in Clustering

Feature selection in clustering is a currently a small domain of research relative to feature selection in classification [56], and a survey presented in 2005 could find no uses of EC for feature selection in clustering [34]. However, there is some recent work into the use of EC for feature selection in clustering.

This section outlines prior work regarding non-EC methods for feature selection in clustering, EC methods for feature selection in clustering, and an existing work on feature weighting in clustering. Providing a brief overview of limitations in the works and stating how those limitations are addressed by the current project.

2.7.2.1 Feature Selection in Clustering

Dy and Brodley [14] investigated the use of Sequential Forward Selection (SFS) as a wrapper method for an Expectation Maximisation (EM) algorithm, assigning data to the most likely cluster when computing external validity. Class error was used as an external valuation metric. Class error can be considered an inverse to purity, namely it is optimal when all clusters contain only one class, in which case $class\ error = 0.0$. Class error, like purity, is trivially optimal when each datapoint is assigned to a singleton cluster.

During the feature selection process, Dy and Brodley [14] compared two different feature selection criteria: one which selected features to maximise separability, and the other which selected features to maximise the likelihood of the data given the resulting partition. Experiments were performed on synthetic and real-world datasets. The synthetic datasets consisted of three two-dimensional spherical Gaussian datasets, consisting of two, three,

and four Gaussian clusters. The real-world datasets consisted of an ionosphere dataset (dimensionality 34) and a pre-processed HRCT-lung dataset [15] (dimensionality 125).

On the synthetic datasets Dy and Brodley [14] found that using separability as feature selection criteria improved external validity, but that using maximum likelihood as selection criteria made external validity worse than the baseline EM algorithm. On the real-world datasets all test conditions demonstrated worse external validation scores than the baseline EM algorithm.

We note several limitations to this work: the work utilised an external validation measure which does not penalise for finding too many clusters, the work failed to improve external validity of partitionings when dimensionality was greater than 2, the statistical internal validation measure did not improve external validation in any tests, and the synthetic datasets chosen were very simple, being both hyperspherical and of low dimensionality.

2.7.2.2 Feature Weighting in Clustering

Modha and Spangler [37] examined feature weighting in k-means clustering on relatively low dimensional datasets.

Specifically a variant on k-means clustering was created, named convex k-means, that assumed all clusters were convex. During clustering a weight vector α was calculated, based on information theoretic criteria with regards to the clusters, and was designed to minimise intra-cluster sparsity and maximise inter-cluster separation.

Because α was calculated in closed form, there were several restrictions placed on it. Namely, for a dataset with dimensionality d, $\sum_{i=1}^{d} \alpha_i = 1$, and $\forall_{i \in \{1,..,d\}} \alpha_i \geq 0$. While the latter means that the algorithm can theoretically remove features from the dataset, in practice no features were ever assigned a weight of 0, and it was never assessed as to whether or not the closed form calculation could ever lead to such a situation.

Modha and Spangler [37] tested their method on a number of datasets with dimensionality less than three, specifying various values for k (i.e. the number of clusters for the convex k-means algorithm). They found that in all cases the weighting improved their internal validation measures, finding better quality clusters for all specified k.

The work by Modha and Spangler [37] is highly novel and with positive results, but also demonstrates some limitations. Firstly, the convex k-means algorithm specifically maintains the original drawbacks present in k-means clustering, namely being highly sensitive to initialisation, requiring a predefined k, and assuming that all clusters are convex (i.e. hyperspherical). Secondly, the convex k-means algorithm, in calculating optimal weightings, does so as a closed form system of equations, which have not been demonstrated as tractable on datasets with high dimensionality (the original work utilised only datasets with dimensionality 2 and 3). Thirdly, the set of considered feature weightings only contains sets of feature weights such that the sum of all weights equals 1, a restriction imposed to ensure that optimal feature weights are calculable in closed form. The effect of this constraint is not examined in the original work. Lastly, the experimental design used in the work did not include any external validation measures. External validation is considered the gold standard within clustering, and is highly important if an algorithm is to be brought into usage outside of experimental contexts.

2.8 Summary

This chapter provided an overview of the theoretic background and specific algorithms related to the current project. In particular the clustering algorithms used in this project were

introduced, some works regarding the use of EC for clustering were examined as tangentially related work, and key works showing the separate use of feature selection and feature weighting in the context of clustering were described.

Where relevant, key limitations of prior work were also identified, and we note that the work in this project extends the existing literature in the following ways:

1. It is the first work to attempt to simultaneously select and weight features in the context of clustering,

2. Prior work on these individual tasks in clustering has used either a poor external validation measure or performed no external validation on results,

3. Where feature selection has been performed in clustering and (limited) external validation has been performed, high dimensionality has damaged results, and

4. Where feature weighting has been performed in prior work it has been a closed form calculation with many assumptions, and has been examined only on low dimensional datasets.

Thus this chapter indicates that this project is novel in several ways in its goal to utilise PSO to perform feature selection weighting simultaneously in a way which improves the validity of found partitions, according to state-of-the-art external validation measures, and interpretability in the context of clustering. Further, we note a lack of substantial work related to the use of EC for feature selection in clustering.

As no work involving the evolution of specific distance functions has been presented, the extension to our main method can be said to extend the literature more broadly.

Chapter 3

Datasets

3.1 Introduction

This chapter details the selection of clustering datasets. In particular the criteria for selecting datasets in this project are stated, and details for generation of datasets satisfying these criteria are explained. Lastly, example datasets are analysed in order to provide intuition for the significance of results in the current project.

3.2 Choice of Dataset

The choice of datasets for this project is based primarily on two criteria. The first criterion is that the chosen datasets should be non-trivial, such that standard clustering algorithms are unable to reliably discern the base truth from the dataset. To facilitate this non-axis aligned, non-hyperspherical datasets of arbitrary orientation are desired. The second criterion is that the datasets chosen should have established use in the wider clustering literature, to improve confidence in results.

While in low dimensions datasets generated using Gaussian distributions with high covariance can create clusters which satisfy our first criterion, the requirement that datasets be non-spherical tends to fail for Gaussian distributions at sufficient dimensionalities [25]. In particular clusters generated in this way tend to be hyperspherical because high variance in any single direction tends to have negligible effect on distance when there are very many dimensions [25].

The current project thus uses datasets generated through a method put forward by Handl and Knowles [25], which uses a genetic algorithm combined with statistical data generation to overcome this problem. Further, datasets generated using this method are widely used in the literature [26, 27, 31], satisfying our second criterion. This method of dataset generation is described below.

3.3 Dataset Generation Method

Handl and Knowles [25] specify a method of dataset generation, which utilises genetic algorithms and statistical data generation to create non-axis aligned, non-hyperspherical datasets of arbitrary orientation.

Specifically, four parameters are considered for each cluster:

1. The origin, which is treated as the first focus,
2. The interfocal distance, generated in $U[1.0, 3.0]$,

3. The orientation of the major axis, chosen uniformly from all possible orientations, and

4. The maximum sum of Euclidean distances between the two foci, generated in $U[1.05, 1.15]$.

The method is then:

1. For each cluster, datapoints are generated at a Gaussian distributed distance from a uniformly random point on the major axis in a uniformly random direction, being rejected if they lie outside the boundary.

2. After all datapoints are generated, with origin set to $0, ..., 0$, a genetic algorithm is used to move the origins such that a cost consisting of deviation of the entire dataset, plus a penalty term for any overlapping clusters, is minimised.

This tends to generate ellipsoid clusters which are non-axis aligned and of arbitrary orientation. Further, while the resulting dataset is arranged compactly, clusters still tend to be separable from other clusters in the dataset.

Some notable characteristics of the datasets selected for the current work can be found in Table 3.1, which also includes the Silhouette and Connectedness scores under perfect partitioning (according to the base truth).

Table 3.1: Characteristics of Ellipsoid Datasets

D	#Clusters	Silhouette	Connectedness	#Instances	Smallest Cluster	Largest Cluster
2	4	0.59	17.56	219	25	78
2	10	0.42	15.98	632	15	122
50	4	0.34	29.84	246	14	90
50	10	0.39	28.90	805	34	124
100	4	0.41	31.51	254	32	93
100	10	0.41	29.80	747	34	103

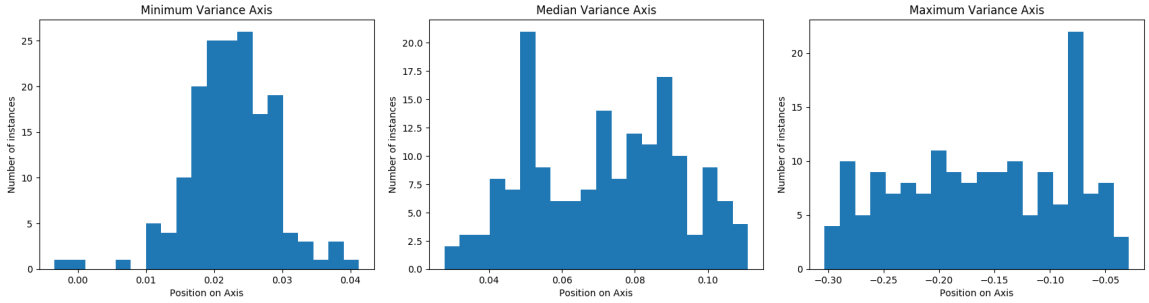
Note: D represents Dimensionality of the dataset.

We note that all datasets contain clusters which, on average, have higher inter-cluster distance than intra-cluster distance with respect to the base truth, as shown by a positive Silhouette value for all datasets. We also note that cluster size varies greatly within datasets, with the largest cluster being several times larger than the smallest cluster in all datasets. Although the Silhouette value is positive for each dataset, these properties indicate that the datasets are not easily clusterable.

3.4 Dataset Analysis

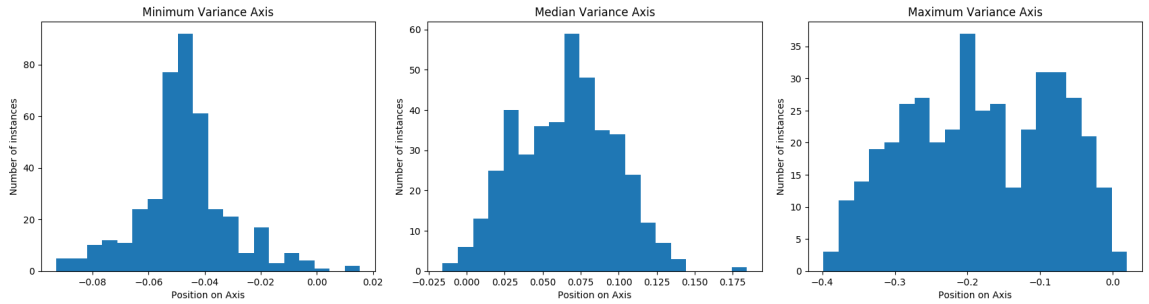
To demonstrate further characteristics of the datasets, we look at the distribution of datapoints within clusters along specific axes.

Namely, from the dataset containing 100 dimensions and four clusters, we select select two clusters for examination. For these two clusters we find the minimum, median, and maximum variance axes, and then plot the distribution of position along these. The histograms for datapoint placement on these axes can be found in Figures 3.1 and 3.2.



(a) Minimum Variance Axis (b) Median Variance Axis (c) Maximum Variance Axis

Figure 3.1: Properties of Axes in First Cluster



(a) Minimum Variance Axis (b) Median Variance Axis (c) Maximum Variance Axis

Figure 3.2: Properties of Axes in Second Cluster

In interpreting these plots we note that, while most axes appear to be Gaussian distributed with a random mean and variance. As the variance increases the rejection criteria in the cluster generation can make some high variance axes more uniformly distributed. Further, where the randomly chosen major axis happens to be strongly aligned with a specific axis that axis is entirely uniformly distributed.

3.5 Summary

This chapter outlined the desired traits of datasets for the current project, identifying that they should be sufficiently complex, such that the baseline algorithms cannot naively find perfect clusters, and widely used in the existing literature.

Suitable datasets proposed by Handl and Knowles [25] were identified and the method of generating the datasets explained before an analysis of clusters in one such dataset was presented.

Chapter 4

Optimisation Criteria

4.1 Introduction

This chapter details and justifies the two novel optimisation criteria that fulfill the first objective of this project. Existing validation measures are examined and considered as opimisation criteria, but are shown to lead to naive solutions. The novel distance based validation measure, Combined Silhouette and Connectedness (CSC) is shown to be more robust to naive solutions, and evidence for choices made regarding its design are presented. The novel statistical validation measure, the Bayesian Clustering Ratio (BCR) is derived, and is unique in its inclusion of a statistical measure of separability.

4.2 Chapter Goals

Existing distance based validation measures tend to act as poor optimisation criteria under a feature selection and weighting framework, and where statistical validation measures have been used they have not included a notion of separability. This chapter aims to develop two novel validation measures which are suitable as optimisation criteria, while demonstrating the behaviour that justifies the need to create new internal validation measures for this project. This chapter approaches this in the following way:

- Section 3 will examine naive solutions, and demonstrate clearly that existing distance based validation measures are prone to returning naive solutions when used as optimisation criteria, justifying the need for a novel distance based validation measure,
- Section 4 will present the process through which the novel distance based measure, CSC, was constructed, including results supporting choices made, and
- Section 5 will present the statistical foundation for the novel statistical validation measure, the BCR, explaining how it was derived and how a notion of statistical separability is attained.

Thus sections 4 and 5 can be said to satisfy Goal 1 in this project.

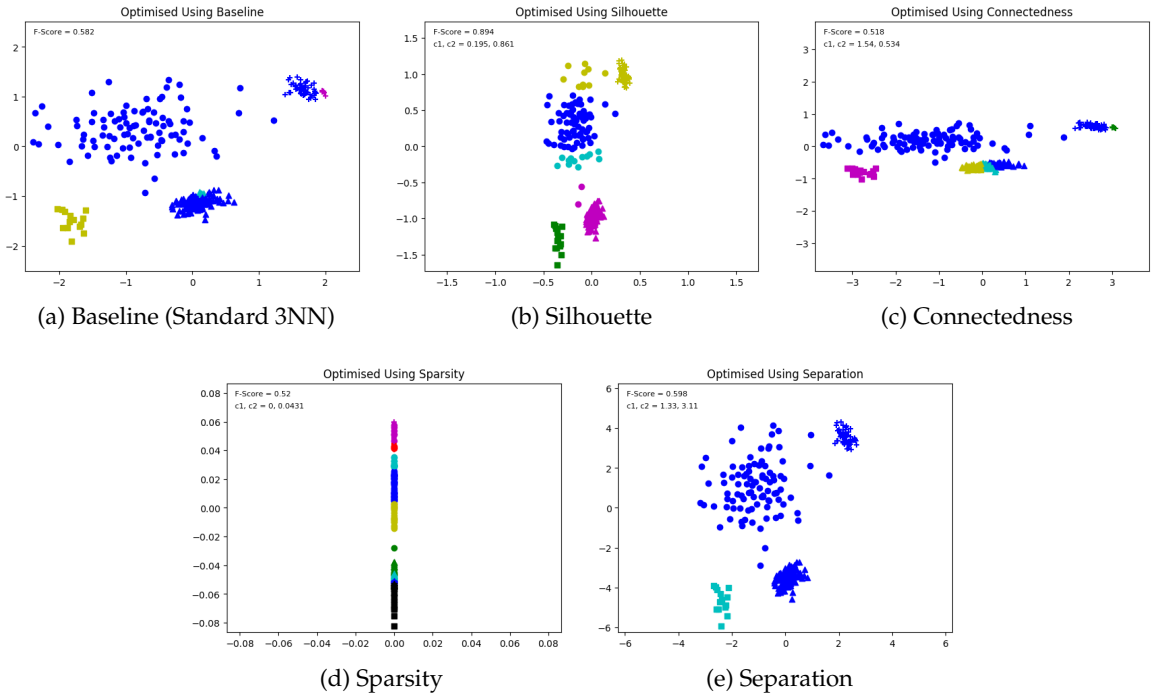
4.3 Naive Solutions

Naive solutions refer to partitionings which satisfy some internal validation measure, but are fundamentally uninteresting. For example, if a clustering algorithm returns only two clusters where the base truth contains many clusters, then this is considered a naive solution.

To examine the occurrence of naive solutions, Particle Swarm Optimisation for Feature Selection and Weighting (PSO-FSW), introduced formally in Chapter 5, is used to select features and weights while optimising according to existing internal validation measures, using the 3NN Clustering algorithm. The preliminary results show that most of the internal validation measures produce naive solutions on datasets with dimensionality as low as 2. The one measure for which this was not the case was the Silhouette measure.

Further investigation demonstrated the the Silhouette measure when used as a sole optimisation criteria also finds naive solutions as dimensionality increases.

Figure 4.1, demonstrates the the partitions that were found when performing feature selection and weighting on a two dimensional dataset with four clusters. In Figure 4.1 colour represents the assigned cluster, shape of datapoints the actual label, and the position of each point is the position after feature selection and weighting.



(a) Baseline (Standard 3NN) (b) Silhouette (c) Connectedness

(d) Sparsity (e) Separation

Figure 4.1: Results of Clustering with different Optimisation Criteria

Here we note some interesting results for sparsity and separation, the only used internal validation measures which do not balance inter-cluster and intra-cluster characteristics. In particular we note that sparsity has returned a set of clusters which have had one feature removed, and the remaining feature scaled close to 0, such that the resulting clusters are arbitrarily dense despite having minimal value as interpretable clusters. Optimising for separability has led to the creation of exactly two clusters such that the distance between these clusters is maximal. It is reasonable to believe that these behaviours would occur regardless of the particular datasets used.

In these tests the Silhouette measure was the only internal validation measure which, when used as optimisation criteria, improved how closely the partitions reflected the base truth. To examine whether or not the Silhouette measure avoided naive solutions in higher

dimensions, trials were run on datasets with dimensionality 50 and 100 using the 3NN-Clustering algorithm.

These results are presented in Table 4.1.

Table 4.1: F-Score and Clusters Found using the Silhouette measure as Optimisation Criterion

Dimensionality	Number Clusters	F-Score	Silhouette	Number of Clusters Found
50	4	0.48	0.67	2
50	10	0.34	0.75	2
100	4	0.56	0.71	2
100	10	0.96	0.41	9

As can be seen in Table 4.1, in higher dimensions the optimisation using the Silhouette measure is also prone to naive solutions, finding two cluster solutions which optimise the Silhouette measure on three of four datasets. This supports the motivation for Goal 1 of the current project.

4.4 Distance Based - Combined Silhouette and Connectedness (CSC)

The process for deriving the CSC validation measure was done in several major stages. First, the four non-EC clustering algorithms detailed in section 2.7.1 were applied to a number of datasets with the same number of clusters and dimensionality as the datasets selected for testing. Secondly, the resulting partitions for each of these trials were assessed using the validation measures outlined in 2.2.1. To see how well each of these measures acted as an indicator for the base truth, particular attention was given to how well each internal validation measure correlated with the F-Score. Finally, after this assessment, the most promising internal validation measures were combined into the new CSC measure.

Lastly, this section details the choice within the CSC measure regarding whether or not, during optimisation, to appraise partitions using the weighted feature subset or the full unweighted feature set.

4.4.1 Evaluating Internal Validation Measures

In order to evaluate the performance of internal cluster validation measures two cases are looked at:

1. The correlation between validation measures when using standard clustering algorithms to partition datasets, with particular focus on how internal validation measures correlate to F-Scores.

2. The resulting clusters when optimising a distance function for the internal validation measures on a low dimensional dataset.

We note that partitionings returned by clustering algorithms are used in the first assessment, rather than the base truth, so that the F-Score is not trivially perfect, i.e. we wish to see how internal validation measures are affected by poor partitioning as well as good partitioning.

4.4.1.1 Correlational Analysis

Each of the four clustering algorithms was run on datasets with the same dimensionality and number of clusters as those outlined in Table 3.1, which were generated using the same method as outlined in Chapter 3. For each algorithm and dataset partitions were assessed using the validation measures outlined in Chapter 2, with these results recorded.

These results were then correlated in order to infer which internal validation measures best reflected the goodness of clusters according to external validation measures, particularly the F-Measure which penalises having more clusters than actually appear in the data. The correlation between validation measures can be seen in Table 4.1, where the correlation between F-Score and internal validation measures are bolded.

Table 4.2: Correlation Matrix of Validation Measures

	F-Score	Purity	Silhouette	Connectedness	Sparsity	Separation
F-Score	1.00	0.93	**0.72**	**0.46**	**-0.41**	**-0.06**
Purity		1.00	0.79	0.52	-0.43	-0.08
Silhouette			1.00	0.27	-0.36	0.31
Connectedness				1.00	-0.63	-0.73
Sparsity					1.00	0.57
Separation						1.00

We note that Silhouette and connectedness have the highest correlation to F-Scores from the internal validation measures, and thus look promising as optimisation criteria to maximise F-Score. Further, Silhouette and connectedness have relatively low correlation to each other, implying that while they are both somewhat indicative of F-Scores they are related to different properties of resulting clusters.

We also note that, as intuition would imply, sparsity is negatively correlated to F-Scores as sparsity is not a desirable feature within clusters. Lastly, we note that separation is effectively uncorrelated to F-Scores, implying that how large the gap is between clusters is not of high importance to cluster quality on this dataset.

4.4.2 A Combined Validation Measure

Using the information presented in Table 4.2, the Silhouette measure and connectedness measures were chosen as the basis for a combined validation measure. This new measure, Combined Silhouette and Connectedness (CSC), seeks to maximise both the Silhouette measure and connectedness, such that values are treated as optimal when clusters are maximally compact relative to the separation between clusters (from the Silhouette measure), but also locally dense (from connectedness).

The combined validation measure is thus a modified product of the silhouette measure and the connectedness measure, modified such that when both of these are negative the product is still negative. This is to say that, given the silhouette measure for a clustering Sil and the connectedness for a clustering $Conn$ the combined validation measure is then:

$$CSC = \begin{cases} Sil * Conn & if\ (Sil > 0) \vee (Conn > 0) \\ -(Sil * Conn) & if\ (Sil < 0) \wedge (Conn < 0) \end{cases}$$

An equivalent formulation, which may be more intuitive, is:

$$CSC = \begin{cases} |Sil * Conn| & if\ (Sil > 0) \wedge (Conn > 0) \\ -|Sil * Conn| & if\ (Sil < 0) \vee (Conn < 0) \end{cases}$$

CSC performs similarly to the standard Silhouette measure on datasets with low dimensionality, producing the result seen in Figure 4.2 on the same dataset. However, as can be seen in Chapter 5, is more robust to naive solutions in high dimensions.

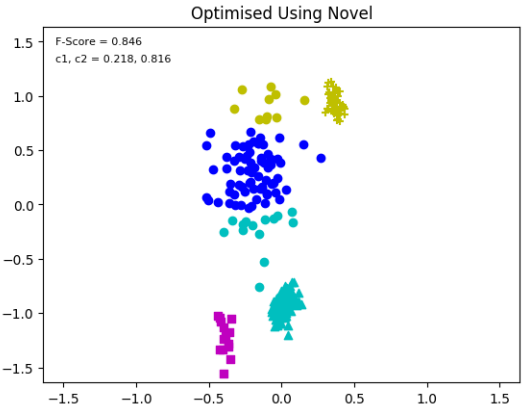


Figure 4.2: Results of Clustering using CSC

4.4.3 Calculating CSC during Optimisation

A novel consideration presents itself in the construction of CSC, that is whether to calculate CSC using the entire, unweighted, feature set, or whether to use the feature selection and weighting found by the algorithm.

The intuition for using the selected and weighted features is this: if the Curse of Dimensionality reduces the meaning of distance measures in high dimensions, then using the weighted feature subset may be necessary to meaningfully analyse partitionings of the dataset. A contrary intuition is that, as naive solutions are problematic, then using the full and unweighted feature set may prevent some naive solutions from occurring.

To test this empirically, four test conditions are considered, corresponding to each component of CSC (that is, the Silhouette measure and connectedness), utilising the full unweighted feature set or using the weighted feature subset during optimisation. For each of these combinations the 3NN Clustering algorithm performed feature selection and weighting on 4 datasets, corresponding to 4 cluster and 10 cluster datasets in each of 50 and 100 dimensions.

Table 4.3: Mean F-Scores for CSC Variants

	Silhouette		
Connectedness		Full Feature Set	Weighted Feature Subset
	Full Feature Set	0.685	0.734
	Weighted Feature Subset	0.698	0.671

As can be seen in Table 4.3, for CSC a mixed approach was found to be most beneficial, using the weighted feature subset for the Silhouette measure, and using the unchanged full feature set for the Connectedness. Thus we can say that CSC, as an optimisation criteria, is seeking feature selection and weighting such that the broader cluster qualities are optimised

under feature selection and weighting, while the local neighbourhood around data instances remains sensible with regards to the original topology of the dataset.

4.5 Statistical Based - Bayes Clustering Ratio

Drawing inspiration from the expectation maximisation algorithm [13] and Bayesian model comparison [22] the novel validation measure Bayes Clustering Ratio (BCR) is proposed.

We assume that data are independently distributed along axis according to a multivariate Gaussian distribution, where each cluster is said to have its own such distribution with means and standard deviations calculated in the usual way from the points associated with the cluster, we define the collection of these distributions as the *Models*. We further adopt the standard method of treating the multivariate probability density function from a model as the likelihood of a point being generated by that distribution, and define the overall probability of a given model associated with a cluster C as $P(Model_C) = \frac{|C|}{|Dataset|}$, or the size of the cluster relative to the size of the dataset.

For each datapoint we consider clustering as making the claim that the $Model \in Models$ associated with its cluster has generated it, a model which we denote $Model_C$, *where* $x \in C$. We thus define the following conditional probability for a datapoint x:

$P(Models|x) = P(Model_C|x) = \frac{P(x|Model_C)P(Model_C)}{P(x)}$ using Bayes theorem.

Clustering, however, is not simply concerned with how well a point sits in its own cluster, but also how well explained that point is by the nearest neighbouring cluster. Thus we introduce a statistical version of separability, denoting how well explained a point is by the statistically 'nearest' cluster. We denote this nearest cluster N as:

$N = argmax_{C' \neq C}\left(P(Model_{C'}|x)\right) = argmax_{C' \neq C}\left(\frac{P(x|Model_{C'})P(Model_{C'})}{P(x)}\right)$

Or in other words, N is the cluster which doesn't contain x but is the most likely cluster given x.

We now need to combine these two measures, and we do so as a likelihood ratio:

$LR(x) = \frac{\frac{P(x|Model_C)P(Model_C)}{P(x)}}{\frac{P(x|Model_N)P(Model_N)}{P(x)}} = \frac{P(x|Model_C)P(Model_C)}{P(x|Model_N)P(Model_N)}$.

The BCR of the entire dataset is then equal to $\prod_{x \in Data} LR(x)$, which we treat as a logarithmic to both avoid rounding error and provide some intuition as to the behaviour of the function, giving us the utilised

$$BCR(Dataset, Clusters) = \sum_{x \in Dataset} log(LR(x)) = \sum_{x \in Dataset} log\left(\frac{P(x|Model_C)P(Model_C)}{P(x|Model_N)P(Model_N)}\right) \tag{4.1}$$

or, equivalently,

$$BCR(Dataset, Clusters) = \sum_{x \in Dataset} \left(log\left(P(x|Model_C)P(Model_C)\right) - log\left(P(x|Model_N)P(Model_N)\right)\right) \tag{4.2}$$

There is however both a computational and theoretic issue here. The computational issue is that while mathematically the denominator in this equation is always greater than zero, computationally in high dimensions this value can easily be rounded to zero, or similarly the logarithm of the value can overflow. The theoretic issue is that if several small clusters are found, each of which has very low standard deviation on at least one axis, then this ratio can be trivially maximised. This is especially true in high dimensions, where the probability of finding points which are arbitrarily close on at least one axis approaches 1.

To address both of these issues, we propose the idea of adding artificial noise to the standard deviations of models, such that for each axis in each model $\sigma := \sigma + c$ for some small constant c. This prevents small clusters of this form from acting as trivial optimal solutions, and also can prevent calculation errors by ensuring that $P(x|Model_N)P(Model_N)$ is not treated as zero. Conceptually, this artificial noise can be thought of as a user specified global uncertainty for each model and each axis, where a higher value for c makes each model more uncertain. Note that adding a constant to all standard deviations reduces the certainty of models with low standard deviations more than robust models which account for a variety of data. In terms of established statistical machine learning methods, the effect is thus similar to adding a uniform prior to the model, and tends to lower peaks in the probability density function while raising troughs [29].

4.6 Summary

This chapter examined the justification for Goal 1 of the current project, and demonstrated that naive solutions do present themselves when using some existing validation measures as optimisation criteria in a feature selection and weighting framework. Further, this chapter detailed the creation of two novel validation measures, satisfying Goal 1, which form the optimisation criterion for the new PSO based feature selection and weighting method, PSO-FSW.

Chapter 5

Particle Swarm Optimisation for Feature Selection and Weighting

5.1 Introduction

This chapter introduces Particle Swarm Optimisation for Feature Selection and Weighting (PSO-FSW), and details both the testing of the algorithm as well as analysing outcomes, contributing towards goals 2 and 3.

PSO-FSW is a novel EC method designed to act as a wrapper method for distance based clustering algorithms. PSO-FSW is designed to automatically perform feature selection while finding optimal weightings for selected features according to some user-defined optimisation criteria. It was tested using the novel validation measures proposed in Chapter 4 for the purposes of this project.

5.2 Chapter Goals

The goals of this chapter are to describe the novel method, PSO-FSW, and demonstrate the results of PSO-FSW when applied to ellipsoid datasets, satisfying goals 2 and 3 respectively. To demonstrate the achievement of these goals clearly, the structure of this section is as follows:

- Section 3 describes how a novel PSO representation is used to perform simultaneous feature selection and weighting in PSO-FSW, and explains some of the characteristics of this novel representation.
- Section 4 formalises how the novel validation measures are utilised as fitness functions used in PSO-FSW.
- Section 5 explains the overall PSO-FSW algorithm, including visualisations of the method and pseudocode.
- Section 6 explains the experimental design, including specific parameter settings.
- Section 7 presents the results of testing.

Thus, sections 3 through 5 will demonstrate that Goal 2 is satisfied, that is that a novel PSO method for feature selection and weighting in clustering has been developed. Sections 6 and 7 will demonstrate that Goal 3 is satisfied, that is that the new method, overall, improves clustering outcomes while using a reduced number of features across a variety of different clustering algorithms.

5.3 A New PSO Representation for Feature Selection and Weighting

As with much prior work utilising PSO for feature selection the dimensionality of each particle in PSO-FSW is equal to the dimensionality of the dataset, with each dimension of the particle corresponding to a dimension in the dataset.

Prior work, however, has mapped each dimension of the particle to a binary value, either through a threshold mapping [56] or probabilistically as in binary PSO [56]. In order to allow both feature selection and weighting, the current algorithm proposes the following interpretation of each dimension in the particle:

$$interpretation_d = \begin{cases} particle_d & \text{, where } particle_d > 0 \\ 0 & \text{, otherwise} \end{cases}$$

This interpretation of the particle is then utilised when calculating pairwise distance between points in the dataset, being utilised in the distance function in the following way, where d is the dimensionality of the dataset, x and y are datapoints, and $c_i = interpretation_i$, $i \in \{1, ..., d\}$:

$$d(x, y) = \sqrt{\sum_{i=1}^{d} (c_i * (x_i - y_i))^2} \tag{5.1}$$

Thus where $particle_i \leq 0$, this interpretation is equivalent to removing the feature indexed at i from the dataset to perform feature selection. Where $particle_i > 0$ this interpretation is equivalent to weighting the feature indexed at i by the value of $particle_i$. Further, as distance functions are symmetric around 0 for each c_i, this method does not remove any unique solutions to the problem. That is, the distance between two points is the same where $c_i = -c_i$, so by setting negative values to 0 we maintain all possible feature weightings. We note as a point of interest that this interpretation is equivalent under certain assumptions to a rectified linear unit [38], as commonly used in artificial neural networks.

5.4 Fitness Function

As PSO-FSW is a wrapper method we are able to evaluate the partitioning that results from a given clustering algorithm using a weighted feature subset corresponding to a particle, and then assign this value to the particle itself.

Specifically, after the interpretation of a particle to the vector c a distance based clustering method is used, with all pairwise distance between datapoints being calculated using the distance function presented in Equation 5. The resulting clusters are then assigned a value according to one of the novel validation measures, namely CSC or BCR. This value is treated as the fitness of the particle used to create the distance function.

5.5 Overall Algorithm

The pseudo-code of the full algorithm with novel particle representation (PSO-FSW), given a user specified base clustering algorithm and evaluation criteria for particles, is presented in Algorithm 1. This algorithm is further visualised in Figure 5.1.

```
1  begin
2  randomly initialise PSO particles and velocities;
3  while termination criteria not met do:
4     for each particle do:
5        create pairwise distance function dist from particle;
6        form clusters using provided clustering algorithm and dist;
7        assign particle value as CSC or BCR of clusters;
8     end for
9     update pbest of particles and gbest;
10    for each particle do:
11       update particle velocity according to equation (2.1)
12       update particle position according to equation (2.2)
13    end for
14 end while
15 create pairwise distance function dist from gbest;
16 form clusters using provided clustering algorithm and dist;
17 calculate the F-Score of clusters;
18 return gbest and the resulting F-Score;
19 end
```

Algorithm 1: Pseudo-code of PSO-FSW

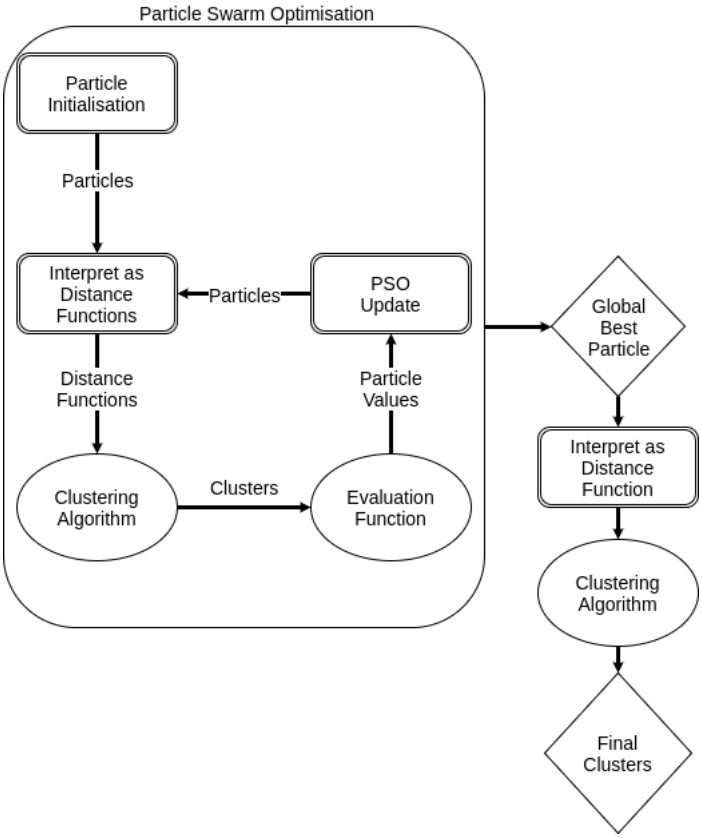


Figure 5.1: Flow Diagram of PSO-FSW

5.6 Experiment Design

The tests evaluated final clusterings from our three conditions (Baseline, PSO-FSW(CSC), and PSO-FSW(BCR)) for each algorithm, using the F-Score. These scores informed whether PSO-FSW was said to have improved over baseline algorithms, and was the basis of statistical significance testing. Further consideration was given to whether or not less features were used to generate the result, an indication of interpretability.

5.6.1 Structure of Experiments

For the four clustering algorithms, 30 independent trials were performed for each of the following three cases:

1. Baseline using the relevant algorithm with Euclidean distance on the unchanged feature set
2. PSO-FSW(CSC) generating a final partitioning through PSO-FSW using the novel distance based internal validation measure CSC as an optimisation criteria
3. PSO-FSW(BCR) generating a final partitioning through PSO-FSW using the novel statistical internal validation measure BCR as an optimisation criteria

For each clustering algorithm results were compared between the PSO-FSW trials and the baseline using a paired Wilcoxon test, where the pairing is done by dataset.

5.6.2 Parameter Settings

There are a large number of parameters in different algorithms in the experiments.

PSO

The parameters selected for the PSO algorithm are the ones suggested in [49]. Specifically they are: weight decay, $\omega = 0.73$; weights of best local and global positions, $c_1 = c_2 = 1.5$; maximum velocity, $v_{max} = 1.0$; the initial distribution for velocity and position is uniform in $[-2, 2]$; the number of particles is 30; and the termination criteria is 30 generations, or 5 generations without global best improvement.

DBSCAN

The optimal value for the epsilon parameter was found empirically by performing a linear search over values of epsilon on datasets also generated using the method put forward by Handl and Knowles [25], which hold the same dimensionalities and numbers of clusters, but were not the same datasets as used in final tests. This search found the following optimal values for the Euclidean case by dimensionality: dimensionality 2, $\epsilon = 0.4$; dimensionality 50, $\epsilon = 0.3$; dimensionality 100, $\epsilon = 0.3$. The minimum samples parameter is set to 5 after a similar process. Results from these trials are outlined in Appendix C.

5.6.2.1 KNN-Clustering

The value for K is fixed at 3 based on empirical trials on datasets also generated using the method put forward by Handl and Knowles [25]. Results from these trials are outlined in Appendix C.

5.6.2.2 BCR Artificial Noise

The BCR artificial noise parameter is fixed at 0.1 based on a small number of empirical trials on datasets also generated using the method put forward by Handl and Knowles [25]. Results from these trials are outlined in Appendix D.

5.7 Results and Discussion

Overall, the results of PSO-FSW(CSC) show significant improvement in F-Measure over all respective baselines. PSO-FSW(BCR) had more mixed results; significantly increasing F-Measure scores for all baseline algorithms with the exception of DBSCAN, where it was significantly decreased. Specifically the following F-Measures and corresponding p-values relative to baseline, calculated using an unpaired Wilcoxon test, are found for each algorithm:

Table 5.1: Mean F-Measure and Corresponding P-Values

	Baseline	PSO-FSW(CSC)	P-Value	PSO-FSW(BCR)	P-Value
Affinity Propagation	0.577	**0.639(+)**	2.4e-28	**0.653(+)**	1.9e-20
KNN-Clustering	0.844	**0.946(+)**	1.0e-29	**0.901(+)**	1.5e-14
DBSCAN	0.795	**0.817(+)**	4.5e-08	*0.547(-)*	7.8e-22
Agglomerative	0.539	**0.697(+)**	3.5e-30	**0.693(+)**	1.4e-20

A plot of the aggregate F-Scores can be found in Figure 5.2.

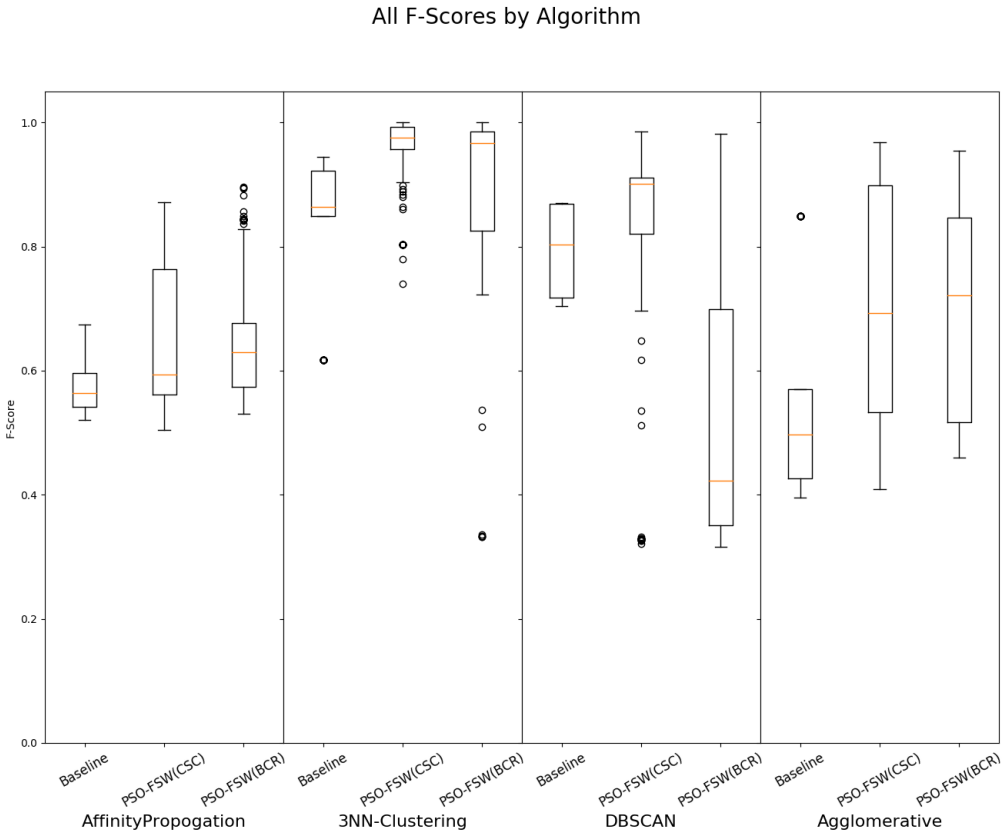


Figure 5.2: F-Scores by Algorithm

This plot, while agreeing with the overall results demonstrated in Table 5.1, shows that finer analysis is needed. In particular, for DBSCAN using PSO-FSW(CSC) and 3NN-Clustering using PSO-FSW(BCR), while the algorithm is overall significantly improved from baseline, it seems to also include several low F-Score outliers.

By separating results by dimensionality we can see that the improvement for some algorithms is relatively consistent across dimensionalities. In particular we note that for the 3NN Clustering algorithm and Affinity Propagation, the improvement in average F-Score can be seen over all dimensionalities for both optimisation criteria, however we note some significant outliers for PSO-FSW(BCR) when used with 3NN-Clustering on the 50 dimensional dataset. These results are presented in Figures 5.3 to 5.5.

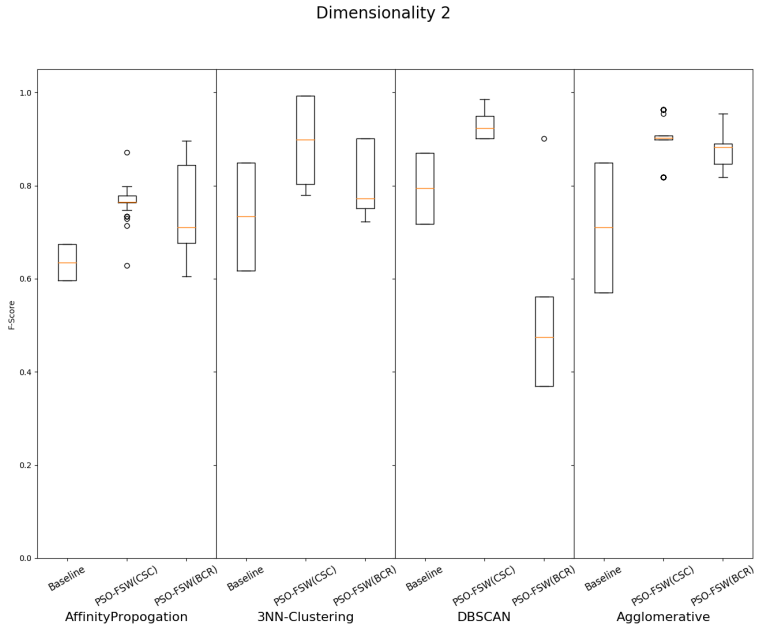


Figure 5.3: F-Scores for Datasets of Dimensionality 2 by Algorithm

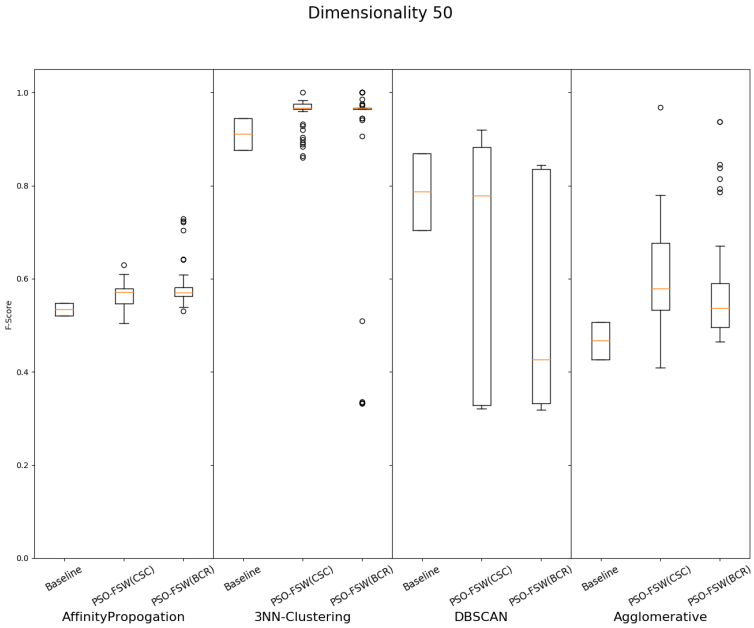


Figure 5.4: F-Scores for Datasets of Dimensionality 50 by Algorithm

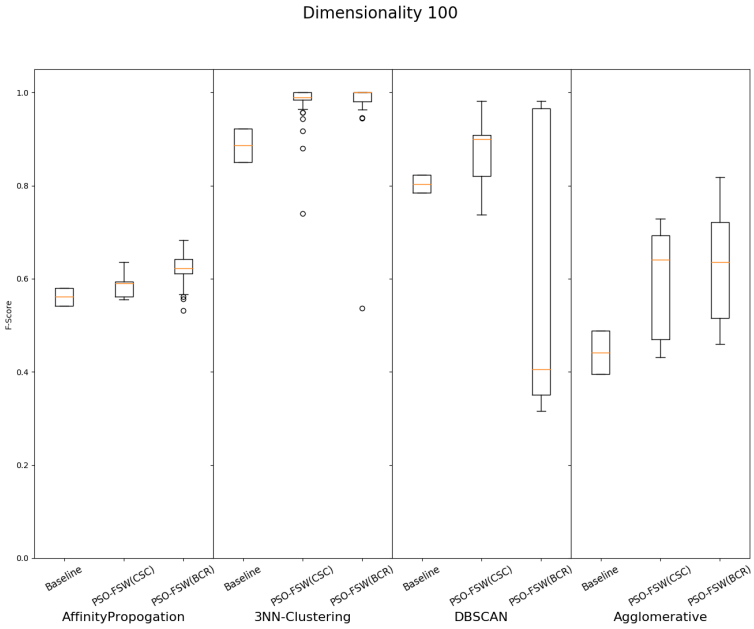


Figure 5.5: F-Scores for Datasets of Dimensionality 100 by Algorithm

It is also important to note the interaction between the number of clusters present in a dataset and the effectiveness of the novel technique. We note that the improvement in F-Score overall seems more pronounced with a smaller number of clusters, however it is still frequently evident when the number of clusters is 10. These results are shown in Figures 5.6 and 5.7.

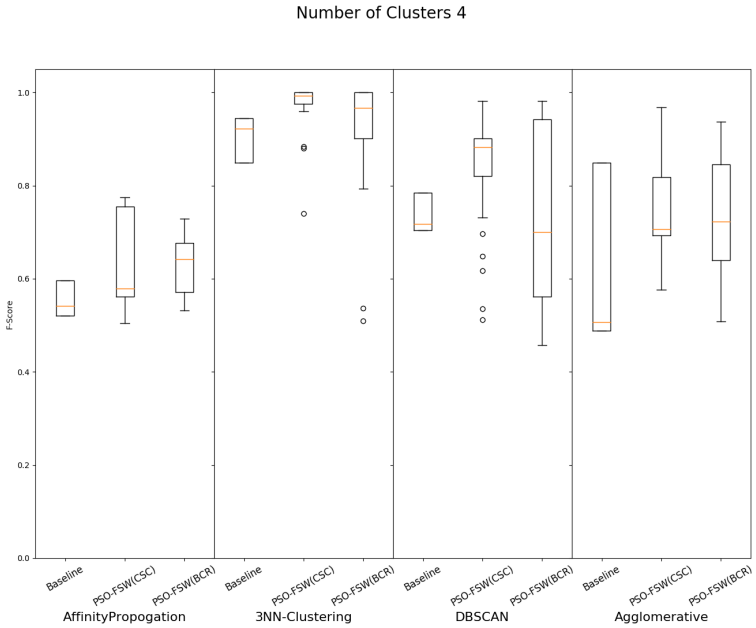


Figure 5.6: F-Scores for Datasets with 4 Clusters by Algorithm

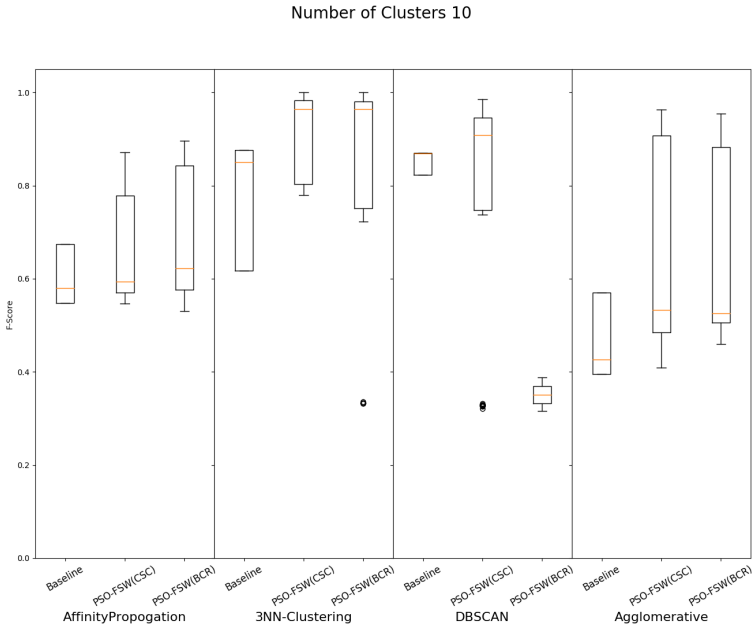


Figure 5.7: F-Scores for Datasets with 10 Clusters by Algorithm

We further note that in all cases where dimensionality is greater than 2 the novel approach always reduces the number of features used, making the clusters more interpretable. Further, and interestingly, occasionally on two dimensional datasets PSO-FSW(BCR) will select only one of the two features. This is particularly interesting when considering that no trials for PSO-FSW(BCR) when using Affinity Propagation or Agglomerative clustering fall

below the worst case performance of the baseline algorithm on these datasets. The mean ratio of features used are presented in Table 5.3.

Table 5.2: Mean Percent of Features Used

(a) PSO-FSW(CSC)

	Dimensionality 2	Dimensionality 50	Dimensionality 100
Affinity Propagation	1.0	0.527	0.498
KNN-Clustering	1.0	0.500	0.488
DBSCAN	1.0	0.485	0.515
Agglomerative	1.0	0.448	0.523

(b) PSO-FSW(BCR)

	Dimensionality 2	Dimensionality 50	Dimensionality 100
Affinity Propagation	0.875	0.49	0.487
KNN-Clustering	1.0	0.483	0.497
DBSCAN	0.833	0.503	0.503
Agglomerative	0.833	0.512	0.509

5.8 Further Analysis

While PSO-FSW(CSC) and PSO-FSW(BCR) improved outcomes relative to baseline algorithms in most cases, each method had instances of returning poor clustering solutions. This analysis is primarily focused on explaining why these behaviours occurred, as well as showing the specific characteristics of some final partitionings.

5.8.1 PSO-FSW(CSC)

While PSO-FSW(CSC) improved overall results relative to all baselines, we note that for DBSCAN, PSO-FSW(CSC) tended to return solutions which were worse than baseline for the the 50 dimensional 10 cluster dataset.

The overall results for this test condition are presented in Table 5.3.

Table 5.3: Results of PSO-FSW(CSC) when used with DBSCAN

Algorithm	d	Clusters	μ_{base}	μ_{test}	σ_{test}	p-value	$\#clus_{base}$	$\#clus_{test}$
DBSCAN	2	4	0.72	**0.90**	0.0	2.9e-11	7.0	3.0
	2	10	0.87	**0.95**	0.015	2.9e-11	11.0	10.19
	50	4	0.70	**0.82**	0.13	9.2e-06	5.0	4.38
	50	10	0.87	*0.47*	0.27	1.1e-06	14.0	7.0
	100	4	0.78	**0.87**	0.066	2.9e-11	6.0	5.38
	100	10	0.82	**0.89**	0.050	8.1e-09	15.0	10.94

In investigating this result, it is important to determine whether the PSO algorithm is unable to find good optima, or whether good optima are found corresponding to poor solutions.

From the individual solutions for this dataset we thus choose a solution with a high F-Score, and one with a low F-Score, and compare the CSC scores for these resulting solutions.

Table 5.4: Comparison of Individual Solutions

	F-Score	CSC	BCR	#Clusters	$\frac{\#selected\ features}{\#features}$
Good Solution	0.919	9.60	11835.62	12	0.50
Poor Solution	0.327	11.30	19187.77	5	0.46

From these results it can be seen that PSO is finding a better optima for at least some trials where F-Score is significantly lower than baseline. This indicates that the poor solutions in this test case represent something approaching a trivial solution, rather than insufficient optimisation.

We further note that both of the lowest base-truth Silhouette results occur on the 50 dimensional data, but that this behaviour not seen frequently for the 50 dimensional dataset with 4 clusters. Thus it appears that this result is an interaction between a higher number of clusters, and a dataset for which the base-truth Silhouette is low relative to the 100 dimensional case.

5.8.2 PSO-FSW(BCR)

PSO-FSW(BCR) contained two specific cases where F-Score was lowered relative to baseline on the higher dimensional datasets. These cases are where PSO-FSW(BCR) was used with DBSCAN, and for 3NN-Clustering on the 50 dimensional 10 cluster dataset. This subsection considers each of these cases.

5.8.2.1 PSO-FSW(BCR) and DBSCAN

It is important to note that there is a single test condition in which PSO-FSW significantly lowered F-Score relative to baseline, namely where PSO-FSW(BCR) was used with DBSCAN. DBSCAN is unique in that it can return unassigned points, which are deemed outliers. If treated as a cluster, this group of outliers have very high variance and a centre roughly in the middle of other clusters. Thus, this significantly reduces all scores in BCR by providing a second cluster which is a viable explanation for all other points in the dataset, damaging our notion of statistical separability, thus making BCR inappropriate for use with DBSCAN.

Cursory tests were done with regards to preventing this behaviour. Specifically, not considering these unassigned outliers as a cluster when calculating the BCR for resulting clusters from DBSCAN was considered, but this was found not to improve results, and to lead to many datapoints being left unassigned. That is, when unassigned points are not included in the calculation for BCR, PSO-FSW(BCR) would find a weighted subset of features such that a large number of points were classified as outliers, while finding naive solutions of two clusters. This effect was more evident on higher dimensional datasets. These results can be found in Appendix E.

5.8.2.2 PSO-FSW(BCR) and 3NN-Clustering

An outlier with regards to lowering F-Scores was also seen when PSO-FCW(BCR) was used 3NN-Clustering on the 50 dimensional dataset with 10 clusters. In particular, the PSO-FSW(BCR) frequently returned two cluster solutions in this case tended to correspond to

high BCR values, indicating that BCR is not robust to naive solutions on this dataset. An illustrative example of this behaviour can be found in Table 5.5.

Table 5.5: Comparison of PSO-FSW(BCR) 3NN-Clustering Solutions

	F-Score	CSC	BCR	#Clusters	$\frac{\#selected\ features}{\#features}$
Good Solution	1.0	10.62	19829.73	10	0.54
Poor Solution	0.334	0.011	55713.51	2	0.52

Thus, rather than a failure of PSO to find a sufficiently good solution, these outliers occurred because of a situation in which there is a disjoint between the base truth and which solutions BCR can evaluate highly.

We also note that, with 3NN-Clustering, where the mean F-Scores were highest for the baseline algorithm as well as both PSO-FSW tests, PSO-FSW(BCR) provides higher F-Scores as dimensionality increases, achieving nearly perfect F-Scores by the time dimensionality is 100. In considering why this may be it's important to note that there are two key assumptions behind the BCR validation measure. Firstly that dimensions are independently distributed, and secondly that all clusters are Gaussian in each dimension. We note in Chapter 3 that the ellipsoid datasets are designed such that this first assumption should not hold in general, and that the second assumption can also potentially be invalid for some clusters in some dimensions.

To investigate the independence axes within datasets we look at the covariance matrix for each base truth cluster from the 50 and 100 dimensional datasets looking at two values: the mean variance of diagonal entries in the covariance matrices, and the mean absolute value of non-diagonal entries in the covariance matrices. Both of these values should be zero where a cluster in a dataset is perfectly hyperspherical and independent along axes. In particular the mean absolute value of non-diagonal entries in the covariance matrices being above zero indicates that independence relations do not hold for that cluster.

Table 5.3 shows these values for the least hyperspherical base truth cluster in each dataset.

Table 5.6: Interpretation of Covariance Matrices for Selected Base Truth Cluster by Dataset

D	#Clusters	$\sigma_{cov(i,i)},\ i \in \{1,...,D\}$	$\mu_{\lvert cov(i,j)\rvert},\ i,j \in \{1,...,D\},\ i \neq j$
50	4	0.011	0.0059
50	10	0.022	0.0089
100	4	0.0091	0.0044
100	10	0.0077	0.0043

We note that the least hyperspherical cluster in the 50 dimensional 10 cluster dataset contains at least twice the standard deviation in the variance of each dimension when compared with other datasets, and also has the highest mean absolute value for non-diagonal entries in the covariance matrix. Thus, the independence assumption made for the BCR validation measure appear to hold less for at least one base truth cluster in the dataset for which it found naive solutions frequently. We note that this violation of specific statistical assumptions is the likely explanation for the failure of BCR to act as a good optimisation criteria for this dataset.

We also note that, for the 100 dimensional datasets, this independence assumption seems to hold more strongly. Thus, the clusters within datasets becoming more independent with respect to individual axes can explain why PSO-FSW(BCR) tended to perform better as dimensionality increased.

5.9 Chapter Summary

This chapter presented the novel method PSO-FSW, and tested it over a variety of datasets using four different baselines, optimising according to each of the novel validation measures introduced in Chapter 4. Results indicated that PSO-FSW achieved significantly improved results over all baselines when optimising for the distance based validation measure CSC while improving interpretability. Further, in three of the four test conditions PSO-FSW also improved results when optimising for the novel statistical validation measure BCR. Where BCR was found to significantly lower results an explanation was presented and examined.

Further explanations for the somewhat counter-intuitive result that PSO-FSW(BCR) was a better method at high dimensionality when using the 3NN Clustering algorithm were presented and examined, finding that at higher dimensions the assumptions behind the BCR validation measure may hold more frequently.

Chapter 6

Extending PSO-FSW

6.1 Introduction

This investigates an extension to PSO-FSW, which utilises PSO to select features, find weightings for selected features, and also finding exponents with which to calculate pairwise distance. This extension is named Particle Swarm Optimisation for Feature Selection, Weighting, and Exponents, denoted PSO-FSWE.

6.2 Chapter Goals

The goal of this chapter is to provide a preliminary investigation into an extension to PSO-FSW, seeing if the results demonstrated by PSO-FSW can be further improved by also searching for unique exponents in the distance function presented in Equation 5. We note that, as PSO-FSW is highly novel already, the concept of this extension is unapproached in the literature. Thus, this extension represents exploratory work supplementary to our main research goals.

6.3 Augmenting PSO-FSW

PSO-FSW created a particle of length d where d was the dimensionality of the dataset in order to perform feature selection and weighting. The extension, PSO-FSWE, will utilise a particle of length $2d$, where the values $1, .., d$ of the particle are used for feature selection and weighting, and the values $(d+1), ..., 2d$ are used to represent unique exponents for each dimension.

The range $[1, 3]$ was chosen for exponents, and they were left real-valued. This range is chosen such the average exponent matches that of PSO-FSW, which utilised Euclidean distance, and such that no single feature should be able to dominate the distance function.

Thus we use the following interpretation of particles:

$$interpretation_i = \begin{cases} particle_i & \text{, where } particle_i > 0 \\ 0 & \text{, otherwise} \end{cases}, \ i \in \{1, ..., d\}$$

$$interpretation_i = \frac{particle_i + 4}{2}, \ i \in \{d+1, ..., 2d\}$$

We note that this interpretation allows us to leave our PSO initalisation uniformly in $[-2, 2]$, as with PSO-FSW, while achieving our desired exponent mapping.

We then denote $c_i = interpretation_i$, $i \in \{1, ..., d\}$, $p_i = interpretation_{i+d}$, $i \in \{1, ..., d\}$, and $\mu_p = mean(\bar{p})$, *where* $\bar{p} = \bigcup\{p_i : c_i > 0\}$, utilising them in the pairwise distance function below:

$$d(x, y) = \sqrt[\mu_p]{\sum_{i=1}^{d} |c_i * (x_i - y_i)|^{p_i}} \tag{6.1}$$

We note importantly that PSO-FSWE strictly extends PSO-FSW, as every possible particle in PSO-FSW has an equivalent particle in PSO-FSWP where $\forall_{i \in \{1,..,d\}}(p_i = 2)$.

6.4 Parameters and Experimental Design

The parameters and datasets used are all exactly as in Chapter 5, with the exception that particles now have length $2d$, where d is the dimensionality of the dataset. The experimental design differs from that in the previous chapter in that each algorithm was tested for only four independent trials per dataset, meaning that each algorithm has a total of 24 tests associated with it.

6.5 Results and Discussion

Overall, the results of PSO-FSWE improvement in F-Measure relative to baseline in only four of 8 test conditions. Namely, the F-Measure significantly improved overall for the Affinity Propagation and Agglomerative clustering algorithms. The F-Measure was significantly decreased for the case where PSO-FSWE(BCR) was used with DBSCAN, and no significant effect was found for other test conditions.

Thus, PSO-FSWE can be said to show less improvement in F-Measure than the simpler PSO-FSW.

Aggregate results for PSO-FSWE can be seen in Table 6.1, with p-values calculated using an unpaired Wilcoxon test.

Table 6.1: Mean F-Measure and Corresponding P-Values

	Baseline	PSO-FSWE(CSC)	P-Value	PSO-FSWE(BCR)	P-Value
Affinity Propagation	0.577	**0.67(+)**	5.6e-05	**0.633(+)**	0.0133
KNN-Clustering	0.844	0.820(=)	0.345	0.723(=)	0.0990
DBSCAN	0.795	0.771(=)	0.290	*0.483(-)*	7.5e-07
Agglomerative	0.539	**0.701(+)**	0.00061	**0.682(+)**	0.0392

A plot of the aggregate F-Scores can be found in Figure 6.1.

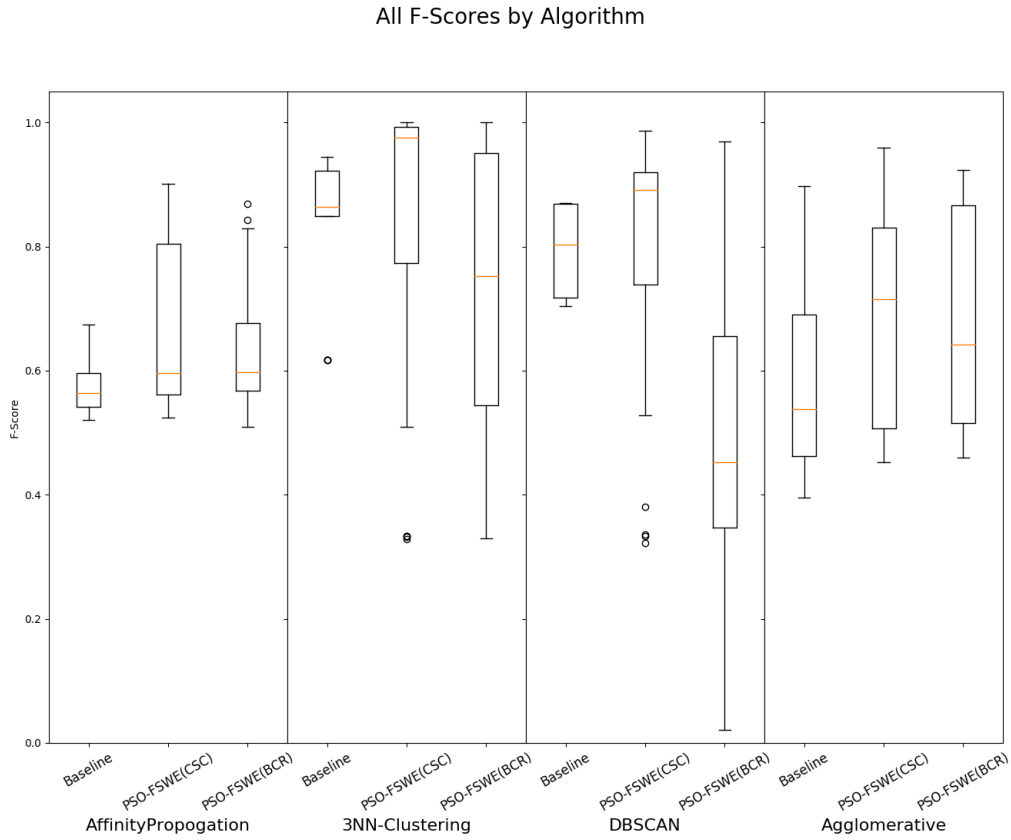


Figure 6.1: F-Scores by Algorithm

This plot shows that poor solutions appear more frequently when using PSO-FSWE relative to PSO-FSW, with results from KNN-Clustering and DBSCAN both indicating a number of poor solutions.

As with PSO-FSW, the mean number of features used for datasets with dimensionality above two is approximately half the full feature set. These results are shown in Table 6.2.

Table 6.2: Mean Percent of Features Used

(a) PSO-FSWE(CSC)

	Dimensionality 2	Dimensionality 50	Dimensionality 100
Affinity Propagation	1.0	0.455	0.456
KNN-Clustering	1.0	0.430	0.484
DBSCAN	1.0	0.463	0.495
Agglomerative	1.0	0.498	0.476

(b) PSO-FSWE(BCR)

	Dimensionality 2	Dimensionality 50	Dimensionality 100
Affinity Propagation	0.750	0.500	0.476
KNN-Clustering	1.0	0.435	0.518
DBSCAN	0.688	0.528	0.456
Agglomerative	1.0	0.55	0.505

6.6 Further Analysis

While PSO-FSWE improved outcomes when used with two of the four baseline algorithms, a number of poor clustering solutions were returned. This analysis is primarily focussed on providing specific examples from the test results for analysis and examining why poor clustering solutions were returned for some algorithms.

6.6.1 Specific Test Outcomes

We note, as with PSO-FSW, that naive solutions can occur primarily due to two possible things: the PSO algorithm being unable to find a good optima, or good optima not corresponding to cluster partitions which align with the base truth.

To examine this we observe specific test outcomes from the same dataset, one which has a high F-Measure, the other which has a low F-Measure, and we observe the difference in fitness of each solution according to our optimisation criterion.

As can be seen in Table 6.3, the CSC measure is higher for the solution with lower F-Score, indicating that poor solutions are indeed naive solutions, rather than a case of PSO being unable to find good optima in the larger search space.

Table 6.3: Comparison of PSO-FSWE(CSC) Outcomes for 3NN-Clustering

	D	#Clusters	F-Score	CSC	BCR	#Clusters Found	$\frac{\#selected\ features}{\#features}$
Good Solution	100	4	0.822	15.08	9133.85	4	0.47
Poor Solution	100	4	0.509	15.65	14874.82	2	0.43

We observe similar behaviour for the same dataset when using PSO-FSWE(BCR), demonstrated in Table 6.4. Further, in this instance, it seems as if the solution with perfect F-Score corresponds to a particularly poor optima, providing strong evidence that the results seen are due to naive solutions occurring.

Table 6.4: Comparison of PSO-FSWE(BCR) Outcomesfor 3NN-Clustering

	D	#Clusters	F-Score	CSC	BCR	#Clusters Found	$\frac{\#selected\ features}{\#features}$
Good Solution	100	4	1.0	12.30	10064.55	4	0.47
Poor Solution	100	4	0.520	4.57	22620.18	2	0.43

Thus, close analysis of specific results demonstrates that the novel validation measures lead to naive solutions when used as optimisation criteria with the PSO-FSWE method, and that this is responsible for the results seen.

6.6.2 Affinity Propagation and Agglomerative Clustering

The two algorithms which showed significant improvement when using the PSO-FSWE method were Affinity Propagation and Agglomerative clustering. We present simple observations regarding why these algorithms did not return naive solutions under the extended novel framework.

Affinity Propagation, when used with these ellipsoid datasets, has a tendency to return many more clusters than the base truth, frequently finding more than twice the number of acutal clusters present in the dataset (this behaviour can be seen clearly for all tests in Appendix A and B). Where naive solutions are found by finding significantly fewer clusters than are present in the base truth, it appears that Affinity Propagation is unable to reasonably find these naive solutions. Thus, it appears as if Affinity Propagation tends not to return naive solutions using these validation measures under this framework simply because the naive solutions conflict with how Affinity Propagation tends to partition these datasets.

We note that agglomerative clustering is unique among the algorithms used in this work in that it is the only algorithm for which the number of clusters to be found is predefined. Thus, where naive solutions correspond to partitionings which have far fewer clusters than the base truth, agglomerative clustering is unable to return these naive solutions, being restricted to solutions which contain only the correct number of clusters.

6.7 Chapter Summary

This chapter demonstrated an extension to PSO-FSW named PSO-FSWE, which also finds exponents for use in distance functions which are generated for a given dataset. Test results were mixed for PSO-FSWE, with improvements only seen where PSO-FSWE was used with Affinity Propagation and agglomerative clustering.

These behaviours were examined and it was found that PSO-FSWE is significantly more prone to returning naive solutions than our primary method PSO-FSW.

Chapter 7

Conclusions and Future Work

The work presented in this project aimed to develop a novel method for simultaneous feature selection and weighting in clustering. Additionally, the project aimed to develop internal cluster validation measures which were suitable as fitness functions in order to achieve this goal. Two new such validation measures were proposed, each indicating benefit as a fitness function for feature selection and weighting. Of the two created validation measures, the Bayes Clustering Ratio represented the first clustering specific statistical validation measure, however it was found to only be appropriate for use on 3 of the 4 tested clustering algorithms.

In creating the novel method for simultaneous feature selection and weighting in clustering, PSO was utilised. This approach, PSO-FSW, overall, led to clusters which were much more aligned with the base truth than baseline algorithms. Further, these improvements were made in a way that made clusters more interpretable, selecting approximately 50% of features to perform clustering.

An extension to this method, named PSO-FSWE, was also investigated, however it demonstrated less benefit PSO-FSW. In particular, PSO-FSWE was found to be prone to discovering optima that represented solutions that appeared less representative of the base truth than those found by baseline algorithms.

Overall, each of the goals specified in this project was met.

7.1 Major Conclusions

Several major conclusions can be drawn from this project:

1. Many existing internal validation measures are not robust to naive solutions when used as fitness functions in a feature selection and weighting framework. In particular, it was shown that many internal validation measures will lead to two cluster partitions when used as fitness functions, regardless of the base truth number of clusters.

2. The two novel validation measures presented have shown that there are internal validation measures which are relatively robust to naive solutions. In particular it has been shown under a feature selection and weighting framework the novel CSC measure avoids naive solutions when paired with each of the 4 tested distance-based clustering algorithms. The novel statistical measure, the BCR, was found to be robust for the tested distance-based algorithms which did not allow for outliers.

3. It was shown that a cluster analysis specific statistical internal validation measure provides benefit. BCR, which accounts for a statistical notion of cluster separability, was found to lead to a better mean F-Score when used as a fitness function relative to

CSC on some algorithms and datasets, particularly when the datasets were 100 dimensional.

4. Simultaneous feature selection and weighting can be used in clustering to improve the performance of several distance based clustering algorithms on ellipsoid datasets. In particular, using PSO to perform simultaneous feature selection and weighting as a wrapper method was shown to improve how closely generated cluster partitions reflect the base truth in many cases, while significantly reducing the number of features used.

5. It was also shown that an extension to the proposed PSO-FSW method tended to generate more naive solutions. In particular, PSO-FSWE was introduced, which also found exponents for use in a pairwise distance function. PSO-FSWE was found to underperform relative to PSO-FSW in most test cases, due to returning naive solutions more frequently. Despite the underwhelming performance, this result represented the first work in creating distance functions which are specific to a given dataset in clustering, and provides opportunity for future work.

7.2 Future Work

There are several extensions to the current project which could be investigated. PSO-FSW was demonstrated to largely outperform baseline algorithms on the used metric. However, the current project has not compared PSO-FSW directly to existing algorithms for feature selection in clustering. In particular, future work should compare PSO-FSW to a binary PSO feature selection algorithm directly, in order to ascertain whether or not the novel approach holds benefit over existing interpretable dimensionality reduction techniques in clustering.

Prior work [55] has also demonstrated the benefit of careful initialisation to PSO algorithms for feature selection. In particular, it is possible to emulate a sequential forward search, which has demonstrated an ability to lower the number of features selected by stochastic feature selection algorithms. The application of this technique to PSO-FSW might encourage the selection of smaller feature subsets, and thus even more interpretable cluster partitions.

This work also utilised only generated ellipsoid datasets. Future work should investigate the application of PSO-FSW to real-world datasets on which the base truth is known. This would establish whether or not PSO-FSW is a method that specifically improves the performance of clustering algorithms on ellipsoid datasets, or whether the effect is more general.

The creation of the clustering specific statistical validation measure, the BCR, also offers opportunity for extension. In particular, a thorough analysis of the effect and importance of the artificial noise parameter in the BCR could yield benefit. Further, BCR was shown to be sensitive to assumptions of independence in this project. A further investigation of how the BCR is affected by different qualities of datasets is essential before it can be presented as a robust validation measure.

Bibliography

[1] Charu C. Aggarwal and Chandan K. Reddy, editors. *Data Clustering: Algorithms and Applications*. CRC Press, 2014.

[2] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[3] Eric Backer and Anil K Jain. A clustering performance measure based on fuzzy set decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):66–75, 1981.

[4] Horace B Barlow. Unsupervised learning. *Neural computation*, 1(3):295–311, 1989.

[5] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[6] Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? In C. Beeri and P. Bruneman, editors, *Proceeding of the 7th International Conference on Database Theory*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer, 1999.

[7] Christopher M Bishop. Latent variable models. In *Learning in graphical models*, pages 371–403. Springer, 1998.

[8] Daniel Bratton and James Kennedy. Defining a standard for particle swarm optimization. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE*, pages 120–127. IEEE, 2007.

[9] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

[10] Scott Chen and Ponani Gopalakrishnan. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. darpa broadcast news transcription and understanding workshop*, volume 8, pages 127–132. Virginia, USA, 1998.

[11] Li-Yeh Chuang, Hsueh-Wei Chang, Chung-Jui Tu, and Cheng-Hong Yang. Improved binary PSO for feature selection using gene expression data. *Computational Biology and Chemistry*, 32(1):29–38, 2008.

[12] Daniel Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

[13] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[14] Jennifer G Dy and Carla E Brodley. Feature subset selection and order identification for unsupervised learning. In *ICML*, pages 247–254, 2000.

[15] Jennifer G Dy, Carla E Brodley, Avi Kak, C Shyu, and Lynn S Broderick. The customized-queries approach to CBIR using EM. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 400–406. IEEE, 1999.

[16] Ahmed AA Esmin, Rodrigo A Coelho, and Stan Matwin. A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artificial Intelligence Review*, 44(1):23–45, 2015.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[18] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y Zomaya, Sebti Foufou, and Abdelaziz Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3):267–279, 2014.

[19] Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.

[20] Chris Fraley and Adrian E Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The computer journal*, 41(8):578–588, 1998.

[21] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[22] Steven N Goodman. Toward evidence-based medical statistics. 2: The Bayes factor. *Annals of internal medicine*, 130(12):1005–1013, 1999.

[23] Vishal Gupta, Gurpreet S Lehal, et al. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, 1(1):60–76, 2009.

[24] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[25] Julia Handl and Joshua Knowles. Cluster generators for large high-dimensional data sets with large numbers of clusters. *Dimension*, 2:20, 2005.

[26] Julia Handl and Joshua Knowles. Improvements to the scalability of multiobjective clustering. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 3, pages 2372–2379. IEEE, 2005.

[27] Julia Handl and Joshua Knowles. An evolutionary approach to multiobjective clustering. *IEEE transactions on Evolutionary Computation*, 11(1):56–76, 2007.

[28] Eduardo Raul Hruschka, Ricardo JGB Campello, Alex A Freitas, et al. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155, 2009.

[29] Edwin T Jaynes. Prior probabilities. *IEEE Transactions on systems science and cybernetics*, 4(3):227–241, 1968.

[30] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[31] Sebastian Klie, Zoran Nikoloski, and Joachim Selbig. Biological cluster evaluation for gene function prediction. *Journal of Computational Biology*, 21(6):428–445, 2014.

[32] Mario Köppen. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, pages 4–8, 2000.

[33] Andrew Lensen, Bing Xue, and Mengjie Zhang. Particle swarm optimisation representations for simultaneous clustering and feature selection. In *SSCI*, pages 1–8. IEEE, 2016.

[34] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502, 2005.

[35] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE, 2010.

[36] Thomas Marill and D Green. On the effectiveness of receptors in recognition systems. *IEEE transactions on Information Theory*, 9(1):11–17, 1963.

[37] Dharmendra S Modha and W Scott Spangler. Feature weighting in k-means clustering. *Machine learning*, 52(3):217–237, 2003.

[38] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[39] Kourosh Neshatian, Mengjie Zhang, and Peter Andreae. A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Transactions on Evolutionary Computation*, 16(5):645–661, 2012.

[40] Karl Pearson. On lines and planes of closest fit to points in space. *Philos. Mag*, 2,:559–572, 1901.

[41] Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.

[42] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[43] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[44] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[45] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[46] Weiguo Sheng, Xiaohui Liu, and Mike Fairhurst. A niching memetic algorithm for simultaneous clustering and feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 20(7):868–879, 2008.

[47] Michael Steinbach, Levent Ertoz, and Vipin Kumar. Challenges of Clustering High Dimensional Data. In L. T. Wille, editor, *New Vistas in Statistical Physics – Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag, 2003.

[48] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[49] Frans Van Den Bergh. *An analysis of particle swarm optimizers*. PhD thesis, University of Pretoria, 2007.

[50] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[51] Xiangyang Wang, Jie Yang, Xiaolong Teng, Weijun Xia, and Richard Jensen. Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters*, 28(4):459–471, 2007.

[52] A Wayne Whitney. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers*, 100(9):1100–1103, 1971.

[53] Hui Xiong and Zhongmou Li. Clustering Validation Measures., 2013.

[54] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[55] Bing Xue, Mengjie Zhang, and Will N Browne. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 18:261–276, 2014.

[56] Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2016.

Appendix A

Full Results for PSO-FSW

This section of the appendix displays the detailed results for each dataset and test condition for both PSO-FSW(CSC) and PSO-FSW(BCR). Each row in each table is a summation of 30 independent trials.

Table A.1: Full PSO-FSW(CSC) Results

Algorithm	D	Clusters	μ_{base}	μ_{test}	σ_{test}	p-value	$\#clus_{base}$	$\#clus_{test}$
AffinityProp	2	4	0.60	**0.75**	0.034	2.9e-11	34.0	5.75
	2	10	0.67	**0.78**	0.030	2.9e-11	37.0	14.81
	50	4	0.52	**0.56**	0.038	9.2e-06	11.0	10.19
	50	10	0.55	0.58	0.021	0.41	29.0	25.81
	100	4	0.54	**0.58**	0.22	2.9e-11	12.0	10.31
	100	10	0.58	**0.59**	0.020	1.1e-06	28.0	25.58
Agglomerative	2	4	0.85	0.86	0.040	0.38	4.0	4.0
	2	10	0.57	**0.93**	0.028	2.9e-11	10.0	10.0
	50	4	0.51	**0.70**	0.16	2.9e-11	4.0	4.0
	50	10	0.43	**0.51**	0.049	5.3e-10	10.0	10.0
	100	4	0.49	**0.70**	0.043	2.9e-11	4.0	4.0
	100	10	0.40	**0.48**	0.032	2.9e-11	10.0	10.0
DBSCAN	2	4	0.72	**0.90**	0.0	2.9e-11	7.0	3.0
	2	10	0.87	**0.95**	0.015	2.9e-11	11.0	10.19
	50	4	0.70	**0.82**	0.13	9.2e-06	5.0	4.38
	50	10	0.87	*0.47*	0.27	1.1e-06	14.0	7.0
	100	4	0.78	**0.87**	0.066	2.9e-11	6.0	5.38
	100	10	0.82	**0.89**	0.050	8.1e-09	15.0	10.94
KNN-Clus	2	4	0.85	**0.99**	2.2e-16	2.9e-11	4.0	4.0
	2	10	0.62	**0.80**	1.1e-16	2.9e-11	17.0	18.0
	50	4	0.94	**0.97**	0.025	5.3e-10	6.0	4.08
	50	10	0.88	**0.95**	0.043	8.1e-09	14.0	9.67
	100	4	0.92	**0.99**	0.071	8.1e-09	6.0	4.2
	100	10	0.85	**0.98**	0.023	2.9e-11	19.0	11.07

Table A.2: Full PSO-FSW(BCR) Results

Algorithm	D	Clusters	μ_{base}	μ_{test}	σ_{test}	p-value	$\#clus_{base}$	$\#clus_{test}$
AffinityProp	2	4	0.60	**0.67**	0.017	2.9e-11	34.0	6.67
	2	10	0.67	**0.84**	0.028	2.9e-11	37.0	10.50
	50	4	0.52	**0.59**	0.053	2.9e-11	11.0	8.83
	50	10	0.55	**0.57**	0.020	8.1e-09	29.0	24.25
	100	4	0.54	**0.63**	0.028	5.3e-10	12.0	8.92
	100	10	0.58	**0.61**	0.023	8.1e-09	28.0	24.00
Agglomerative	2	4	0.85	*0.84*	0.020	1.02e-07	4.0	4.0
	2	10	0.57	**0.91**	0.035	2.9e-11	10.0	10.0
	50	4	0.51	**0.63**	0.12	2.9e-11	4.0	4.0
	50	10	0.43	**0.52**	0.042	2.9e-11	10.0	10.0
	100	4	0.49	**0.74**	0.034	2.9e-11	4.0	4.0
	100	10	0.40	**0.51**	0.023	2.9e-11	10.0	10.0
DBSCAN	2	4	0.72	*0.57*	0.061	5.3e-10	7.0	3.0
	2	10	0.87	*0.38*	0.0092	2.9e-11	11.0	2.25
	50	4	0.70	0.76	0.10	0.18	5.0	3.17
	50	10	0.87	*0.33*	0.0043	2.9e-11	14.0	3.42
	100	4	0.78	**0.90**	0.14	9.2e-06	6.0	4.41
	100	10	0.82	*0.34*	0.013	2.9e-11	15.0	3.0
KNN-Clus	2	4	0.85	**0.87**	0.039	0.027	4.0	4.67
	2	10	0.62	**0.75**	0.0051	2.9e-11	17.0	10.17
	50	4	0.94	**0.95**	0.082	1.0e-07	6.0	3.83
	50	10	0.88	*0.86*	0.24	9.2e-06	14.0	7.41
	100	4	0.92	**0.98**	0.083	5.32e-10	6.0	3.83
	100	10	0.85	**0.99**	0.012	2.9e-11	19.0	10.08

Appendix B

Full Results for PSO-FSWE

This section of the appendix displays the detailed results for each dataset and test condition for both PSO-FSWE(CSC) and PSO-FSWE(BCR). Each row in each table is a summation of 4 independent trials. Because of the small number of trials per dataset caution is recommended in drawing conclusions from these tables.

Table B.1: Full PSO-FSWE(CSC) Results

Algorithm	D	Clusters	μ_{base}	μ_{test}	σ_{test}	p-value	$\#clus_{base}$	$\#clus_{test}$
AffinityProp	2	4	0.60	**0.89**	0.0076	0.021	34.0	3.0
	2	10	0.67	**0.82**	0.040	0.021	37.0	11.5
	50	4	0.52	**0.57**	0.047	0.021	11.0	10.0
	50	10	0.55	0.56	0.021	0.25	29.0	27.8
	100	4	0.54	**0.57**	0.020	0.021	12.0	10.8
	100	10	0.58	**0.61**	0.022	0.021	28.0	24.8
Agglomerative	2	4	0.85	0.85	0.056	0.25	4.0	4.0
	2	10	0.57	**0.92**	0.035	0.021	10.0	10.0
	50	4	0.51	**0.75**	0.070	0.021	4.0	4.0
	50	10	0.43	0.48	0.024	0.25	10.0	10.0
	100	4	0.49	0.71	0.045	0.083	4.0	4.0
	100	10	0.40	**0.49**	0.016	0.021	10.0	10.0
DBSCAN	2	4	0.72	**0.90**	0.0	0.021	7.0	3.0
	2	10	0.87	**0.96**	0.017	0.021	11.0	10.3
	50	4	0.70	0.74	0.14	0.25	5.0	4.8
	50	10	0.87	*0.34*	0.022	0.021	14.0	5.3
	100	4	0.78	0.77	0.041	1.0	6.0	6.3
	100	10	0.82	**0.91**	0.020	0.021	15.0	11.3
KNN-Clus	2	4	0.85	**0.99**	0.0047	0.021	4.0	4.3
	2	10	0.62	**0.79**	0.020	0.021	17.0	19.5
	50	4	0.94	**0.97**	0.0034	0.021	6.0	4.0
	50	10	0.88	**0.33**	0.0018	0.021	14.0	2.0
	100	4	0.92	0.83	0.20	1.0	6.0	3.5
	100	10	0.85	**1.0**	0.0038	0.021	19.0	10.5

Table B.2: Full PSO-FSWE(BCR) Results

Algorithm	D	Clusters	μ_{base}	μ_{test}	σ_{test}	p-value	$\#clus_{base}$	$\#clus_{test}$
AffinityProp	2	4	0.60	**0.67**	0.012	0.021	34.0	6.5
	2	10	0.67	**0.82**	0.056	0.021	37.0	12.3
	50	4	0.52	**0.59**	0.033	0.021	11.0	9.3
	50	10	0.55	**0.58**	0.032	0.021	29.0	25.0
	100	4	0.54	0.56	0.035	0.25	12.0	10.0
	100	10	0.58	0.59	0.014	0.25	28.0	28.0
Agglomerative	2	4	0.85	0.85	0.019	0.56	4.0	4.0
	2	10	0.57	**0.91**	0.014	0.021	10.0	10.0
	50	4	0.51	0.55	0.029	0.25	4.0	4.0
	50	10	0.43	**0.51**	0.026	0.021	10.0	10.0
	100	4	0.49	0.76	0.054	0.83	4.0	4.0
	100	10	0.40	**0.50**	0.024	0.021	10.0	10.0
DBSCAN	2	4	0.72	*0.56*	0.0	0.021	7.0	3.0
	2	10	0.87	*0.38*	0.0081	0.021	11.0	2.3
	50	4	0.70	*0.63*	0.064	0.021	5.0	4.8
	50	10	0.87	*0.18*	0.15	0.021	14.0	3.0
	100	4	0.78	0.81	0.14	1.0	6.0	5.0
	100	10	0.82	*0.34*	0.016	0.021	15.0	3.0
KNN-Clus	2	4	0.85	0.84	0.033	0.25	4.0	4.8
	2	10	0.62	**0.73**	0.028	0.021	17.0	11.5
	50	4	0.94	*0.64*	0.17	0.021	6.0	2.5
	50	10	0.88	*0.49*	0.27	0.25	14.0	4.0
	100	4	0.92	*0.65*	0.20	0.25	6.0	2.5
	100	10	0.85	**0.98**	0.013	0.021	19.0	10.0

Appendix C

Baseline Algorithm Parameter Selection

This section of the appendix displays the results of algorithms utilising different parameters on datasets with the same dimensionality and clusters as those used in testing. Considering that all used baseline clustering algorithms are largely deterministic (with the exception of tie-breaking in DBSCAN), each of these entries uses a single trial. Where datasets are used they are datasets with the same characteristics as those used for testing.

Table C.1: Selecting K in KNN-Clustering

Value of K	Mean F-Score Over All Datasets
2	0.74
3	**0.79**
4	0.71

Table C.2: Selecting DBSCAN Parameters

(a) Mean F-Score for Datasets with Dimensionality 2

		ϵ					
		0.1	0.2	0.3	0.4	0.5	0.6
Minimum Samples	4	0.58	0.64	0.73	0.76	0.73	0.69
	5	0.58	0.69	0.75	**0.78**	0.72	0.68
	6	0.57	0.72	0.75	0.76	0.71	0.66

(b) Mean F-Score for Datasets with Dimensionality 50

		ϵ					
		0.1	0.2	0.3	0.4	0.5	0.6
Minimum Samples	4	0.18	0.25	0.38	0.47	0.46	0.45
	5	0.16	0.29	**0.49**	0.46	0.46	0.45
	6	0.15	0.30	0.47	0.46	0.45	0.45

(c) Mean F-Score for Datasets with Dimensionality 100

		ϵ					
		0.1	0.2	0.3	0.4	0.5	0.6
Minimum Samples	4	0.19	0.23	0.37	0.45	0.45	0.44
	5	0.16	0.33	**0.48**	0.46	0.45	0.43
	6	0.15	0.35	0.45	0.46	0.45	0.43

Appendix D

Artificial Noise Selection

This section of the appendix displays the results on various datasets of utilising 3NN-Clustering with PSO-FSW(BCR) using a small sample of possible values for the artificial noise parameter. The parameter was selected to maximise overall F-Score on these datasets, which share charactaristics with the datasets used for testing. The best F-Score for each dataset is bolded.

Table D.1: Selection of the Artificial Noise Parameter for BCR

Artificial Noise	D	#Clusters	F-Score	BCR
0.0	2	4	0.83	6567.89
	2	10	**0.75**	11838.57
	50	4	0.80	inf
	50	10	0.33	inf
	100	4	0.97	inf
	100	10	0.89	inf
0.1	2	4	**0.83**	6282.32
	2	10	0.72	9955.01
	50	4	**0.86**	7601.55
	50	10	**0.72**	24745.29
	100	4	**1.0**	10064.55
	100	10	**0.99**	19947.82
0.2	2	4	0.79	4889.78
	2	10	0.75	9264.97
	50	4	0.54	6736.25
	50	10	0.47	18224.70
	100	4	0.96	2013.96
	100	10	0.96	3646.89

Note: 'inf' denotes an overflow in calculating the BCR, and is trivially treated as greater than all numbers.

Appendix E

DBSCAN and PSO-FSW(BCR)

This section of the appendix displays the results where the BCR excluded datapoints considered outliers by DBSCAN during calculations.

Table E.1: Results of PSO-FSW(BCR) with DBSCAN, Outliers not used in BCR Calculation

D	Clusters	F-Score	#Assigned Datapoints	#Outliers	BCR	#Clusters
2	4	0.56	213	6	6599.07	2
2	10	0.38	628	4	40677.80	2
50	4	0.52	194	52	11988.75	2
50	10	0.35	651	154	64778.91	2
100	4	0.53	194	60	43873.21	2
100	10	0.34	627	120	181577.65	2