Particle Swarm Optimisation for Feature Selection and Weighting in High-Dimensional Clustering

Damien O'Neill, Andrew Lensen, Bing Xue, and Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
Email: {damien.oneill, andrew.lensen, bing.xue, mengjie.zhang}@ecs.vuw.ac.nz

***Abstract*—Clustering, an important unsupervised learning task, is very challenging on high-dimensional data, since the generated clusters can be significantly less meaningful as the number of features increases. Feature selection and/or feature weighting can address this issue by selecting and weighting only informative features. These techniques have been extensively studied in supervised learning, e.g. classification, but they are very difficult to use with clustering due to the lack of effective similarity/distance and validation measures. This paper utilises the powerful global search ability of particle swarm optimisation (PSO) on continuous problems, to propose a PSO based method for simultaneous feature selection and feature weighting for clustering on high-dimensional data, where a new validation measure is also proposed as the fitness function of the PSO method. Experiments on datasets with varying dimensionalities and different number of known clusters show that the proposed method can successfully improve clustering performance of different types of clustering algorithms over using the baseline of the original feature set.**

I. Introduction

Clustering is one of the most fundamental unsupervised tasks in machine learning and data mining, and can be described as the task of partitioning a dataset into groups, such that objects/examples within the same group are related or similar, and objects of different groups are comparatively unrelated [1]–[3]. However, there is no strict agreement as to what it formally means for examples to be related or similar. This particular difficulty has been summarised aptly by Backer and Jain [1], who stated that "interesting" clusters are often created based on a subjectively chosen measure of similarity.

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 clustering has led to the proposal of many cluster validation measures in the literature [4], [5]. Some examples of such validation measures include: *compactness* [5], which measures of how *close together* the data points within clusters are; and *separation* [5], which measures how *far away* different clusters are from each other. Most commonly used measures are distance based measures, but without careful design, they may lead to trivial or naive solutions, and distance in a high-dimensional space might be meaningless. Because of the inherent subjectivity of what it means for a partitioning of data to be good, and the abundance of existing validation measures, finding a suitable validation measure for a given clustering algorithm on particular datasets is very challenging.

A further complication when performing clustering relates to the dimensionality of data, i.e. having a large number of features. High dimensionality 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], [7]. Thus, for many high-dimensional applications of clustering, dimensionality reduction techniques are required in conjunction with a clustering algorithm in order to address these issues [8]. A key dimensionality reduction method is *feature selection* [9], which attempts to find a subset of the original features, which give improves or similar results with better interpretability than the case when the full feature set is used. However, the search space of feature selection for a dataset with dimensionality d is 2^d. This large search space means that frequently feature selection methods need to utilise some form of heuristic search throughout the space. Many existing methods fail to find optimal or near-optimal solutions, which requires a powerful global search method.

Asides from the large number of features, the relative importance of different features is another key factor, which can be achieved by *feature weighting*, where more important features should be weighted more heavily than less important features. However, there are interactions between features, i.e. features can not be considered individually, which makes both feature selection and feature weighting challenging. Specifically, feature selection needs to have an appropriate optimisation measure together with a powerful global search method, to find a subset of complimentary features. Feature weighting needs to optimise weights (continuous values) of all features simultaneously, which requires a powerful continuous optimisation method. Particle swarm optimisation (PSO) [10] is such a method, and is perhaps the most used evolutionary computation (EC) method in clustering and has shown success in recent years [11], [12]. Furthermore, PSO has also been used for feature selection or feature weighting, mostly in classification [13], but its potential in simultaneous feature selection and weighting in clustering has not been investigated.

Goals: This research aims to address issues that arise in clustering high-dimensional datasets by developing a new effective measure and then a simultaneous feature selection and weighting approach using PSO, with the expectation of improving the interpretability and quality of resulting clusters and providing insight as to how important different features

000-0-000-0000-0/18/$00.00 ©2018 IEEE

are to the dataset as a whole. The selected and weighted features will be used with 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. The specific goals are to:

1) Create a novel optimisation criterion as the fitness function to guide the search of PSO,
2) Develop a novel PSO method for simultaneous feature selection and weighting in clustering, and
3) Investigate the generality of the novel algorithm and optimisation criterion by applying them to several different types of clustering algorithms over several datasets, and to test whether they can achieve better clustering performance while using fewer number of features.

II. Background

A. Particle Swarm Optimisation

Particle Swarm Optimisation [10] (PSO) is swarm based EC method, where each candidate solution in the swarm is denoted as a particle and represented by a vector. A random d-dimensional particle $x_i \in \mathbb{R}^d$ has a 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 $gbest$. Each iteration of the PSO algorithm performs the following updates, given user specified inertia weight ω, acceleration coefficient for $pbest$ and $gbest$, denoted as c_1 and c_2, respectively, and a function r which returns a uniform random value in $[0, 1]$:

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

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

PSO searches for the optimal solution(s) by iteratively updating the position, velocity, $pbest$ and $gbest$ until a predefined termination criterion has been met, then $gbest$ is returned as the final solution.

B. 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 high-dimensional spaces has been formalised by Beyer et al. [6] which is the aspect of the Curse of Dimensionality which this research 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 $argmax_y(dist(x, y))$ and a point z which satisfies $argmin_z(dist(x, z))$, the $lim_{d\to\infty} \frac{dist(x,y)-dist(x,z)}{dist(x,z)} \to 0$. This is to say, while all distances tend towards infinity as dimensionality tends to infinity, the maximum and minimum distances between all points tend towards equidistance. Strong evidence of this behaviour was found at a dimensionality of only $d = 10$ for many synthetic and real-world datasets.

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 given sufficient dimensionality. Research has supported this implication, showing that this aspect of the Curse of Dimensionality significantly impacts the effectiveness of clustering algorithms [14].

C. Clustering Algorithms

Fahad et al. [2] propose a taxonomy for clustering algorithms consisting of the following five overall types: partitioning based, hierarchical based, density based, grid based, and model based. We will consider several typical methods to investigate the generality of the proposed method (i.e. Goal 3). Grid based and model based methods are statistical clustering techniques [2], and thus are not applicable to this research, since a key motivation is addressing issues relating to distance on high-dimensional datasets. Therefore, we *propose the use of four algorithms*, one from each type: the partitioning based Affinity Propagation [15], the hierarchical based Complete linkage agglomerative clustering [16], density based Spatial Clustering of Applications with Noise (DBSCAN) [17], and the graph based KNN-neighbour clustering for additional comparisons [18].

Affinity Propagation [15] is a medoid based clustering algorithm and finds clusters based on similarity between datapoints, rather than distance, but the convention of using negative squared Euclidean distance as similarity is well established, making this a suitable algorithm for this research.

Given a similarity matrix S, the general process is:

1) Two square matrices R and A are initialised to arrays of zeros, 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 datapoints,
2) Until a convergence criterion is 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 datapoints 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. Affinity Propagation has improved clustering outcomes relative to k-means on several complicated datasets [15].

DBSCAN [17] is a popular density based 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 the 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 clustering methods [16] perform the following steps for a given number of clusters, $K \in \mathbb{N}$:

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.

KNN-Clustering [18] is a graph based clustering algorithm which performs the following steps given a user specified $K \in \mathbb{N}$ [18]:

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 value of 3 for the parameter K 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 research.

D. Feature Selection and Feature Weighting in Clustering

There have been some work on feature selection in clustering, which are not reviewed in detail due to the page limit; interested readers are referred to [19]. One such work by Dy and Brodley [20] utilises 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 cannot improve external validation in any tests, and the synthetic datasets chosen are very simple, being both hyperspherical and of low dimensionality. EC based methods have been used for feature selection in clustering, such as PSO is used to estimate the number of clusters, select features, and perform clustering in [21], and a memetic algorithm with niching is used for simultaneous feature selection and clustering in [8]. Existing work has shown the potential benefit of using EC for feature selection in clustering, but compared with the extensive research in supervised learning, e.g. classification, the potential of EC for feature selection in clustering is still yet to be investigated.

Compared with feature selection, there has not much work done in feature weighting in clustering. One example is that Modha and Spangler [22] examined feature weighting in k-means clustering based on information theory criteria with regards to the clusters, but the method was tested on a number of datasets with dimensionality less than three. The results show that in all cases, the weighting improved the internal validation measures, finding better quality clusters for all specified k. However, it is unclear whether such conclusions can be carried to higher dimensions.

Existing work on feature selection in clustering has shown the benefit of feature selection or feature weighting, but there is limited work on using EC, especially PSO, to achieve these tasks. Therefore, it is still an open issue to use PSO for feature selection and weighting in clustering.

III. The Proposed Algorithm

Our proposed algorithm is denoted Particle Swarm Optimisation for Feature Selection and Weighting (PSO-FSW).

A. PSO Representation

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/feature in the dataset. In order to allow both feature selection and feature weighting, the following interpretation of each dimension in the particle is proposed:

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

This interpretation of the particle is then utilised when calculating the pairwise distance between datapoints in the dataset. The formed distance function is shown by Equation (3), where d is the dimensionality of the dataset, x and y are datapoints, and $c_i = interpretation_i,\ i \in \{1, ..., d\}$:

$$dist(x, y) = \sqrt{\sum_{i=1}^{d} (c_i * (x_i - y_i))^2} \tag{3}$$

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 [23], as commonly used in artificial neural networks.

B. Fitness Function

In PSO-FSW, each particle is firstly decoded to the vector c, then a distance based clustering method is used to perform clustering, with all pairwise distance between datapoints being calculated using the distance function shown by Equation (3). The performance of the resulting cluster partition is used as the fitness value of each particle. This also means that PSO-FSW is a wrapper feature selection and weighting method, which can wrap any distance based clustering method for evaluating the fitness of each particle.

A new validation measure, forming a new fitness function, is proposed to evaluate the performance of the created partition. Two well-known measures, Silhouette [24] and connectedness, [5] are utilised to form the new fitness function:

- The Silhouette [24] of a given datapoint is calculated as $sil(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$, 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 the datapoint i and all datapoints within that cluster, returning the minimum of these values. The Silhouette for a set of clusters is considered to be the mean of Silhouettes of all datapoints, reflecting on average how close datapoints 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.
- The connectedness [5] of a given datapoint i in cluster C is defined by

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

  with its n closest neighbours and a specified maximum absolute distance m that provides a practical limit on these values such that no distance can dominate the overall sum. The connectedness assigns a high positive value to i if its closest neighbours are from the same cluster, and a high negative value if its closest neighbours are from different clusters.

  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 $[-50, 50]$.

The fitness function is shown in Equation (4), named the Combined Silhouette and connectedness (CSC), which is a modified product of the Silhouette measure and the connectedness measure, such that when both of them 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:

$$CSC = \begin{cases} |Sil * Conn| & if\ (Sil > 0) \wedge (Conn > 0) \\ -|Sil * Conn| & if\ (Sil < 0) \vee (Conn < 0) \end{cases} \tag{4}$$

Algorithm 1: Pseudo-code of PSO-FSW

```
1  begin
2      randomly initialise particles and velocities in the swarm;
3      while the termination criterion is not met do
4          for each particle do
5              decode particle to a pairwise distance function dist, i.e. Equation (3);
6              produce clusters using the predefined clustering algorithm and dist;
7              evaluate the fitness value of particle using Equation (4);
8          end
9          for each particle do
10             update pbest and gbest;
11             update velocity according to Equation (1);
12             update position according to Equation (2);
13         end
14     end
15     create a pairwise distance function dist from gbest ;
16     form clusters using the predefined clustering algorithm and dist;
17     calculate the F-Score of clusters
18     return gbest and the resulting F-Score;
19 end
```

Equation (4) seeks to maximise both the Silhouette and the connectedness measures. The Silhouette measure is calculated using the weighted feature subset, so that the broader cluster qualities are optimised. Connectedness, however, uses the full unweighted feature set, so that the local neighbourhood around data instances remains sensible with regards to the original topology of the dataset. The design or maximisation of Equation (4) is to ensure 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).

C. Overall Algorithm

The pseudo-code of PSO-FSW is described in Algorithm 1. The evolutionary process finishes in Line 14. Line 15 to 17 show the final evaluation of the best evolved solution, i.e. the final *gbest*, after the evolutionary process stops. The clusters generated by the predefined clustering algorithm using the selected and weighted feature subset by *gbest* is evaluated using the F-Score external validation measure.

TABLE I
CHARACTERISTICS OF DATASETS.

D	K	Sil.	Conn.	n	Min. Cluster Size	Max. Cluster Size
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.

IV. EXPERIMENT DESIGN

A number of experiments have been conducted to evaluate the performance of PSO-FSW against the baseline of using all features. The four clustering methods described in Section II-C, i.e. Affinity Propagation, DBSCAN, Agglomerative, and KNN-Clustering, are each used as the wrapped clustering algorithm with PSO-FSW. The popular F-Score is used to evaluate the resulted clusters after the evolutionary process. The unpaired Wilcoxon test is used to evaluate the statistical significance of the differences in performance.

A. Dataset Selection

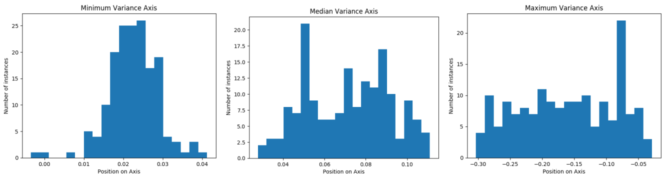
The choice of datasets for this research is based primarily on two criteria. The first criterion is that the chosen datasets should be non-trivial, such that standard clustering algorithms are unable to reliably discern the base truth from the dataset. To facilitate this non-axis aligned, non-hyperspherical 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 research thus uses ellipsoid datasets generated through a method put forward by Handl and Knowles [25], which uses a genetic algorithm (GA) combined with statistical data generation to overcome this problem. Further, datasets generated using this method are widely used and cited in the literature [26], [27], satisfying our second criterion.

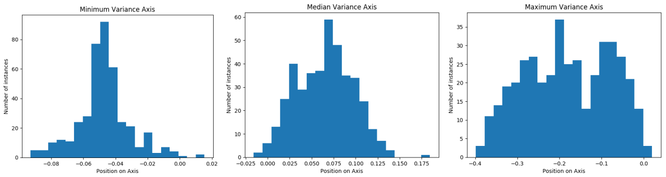
1) Dataset Characteristics: The data generation method in [25] tends to generate ellipsoid clusters which are non-axis aligned and of arbitrary orientation. Further, while the resulting dataset is arranged compactly, clusters still tend to be separable from other clusters in the dataset. Some notable characteristics of the datasets selected for the current work can be found in Table I, which also includes the Silhouette and connectedness scores under perfect partitioning (according to the base truth).

It is found 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 easy to cluster correctly.



(a) (b) (c)

Fig. 1. Properties of Axes in First Cluster.



(a) (b) (c)

Fig. 2. Properties of Axes in Second Cluster.

2) Dataset Analysis: To demonstrate further characteristics of the datasets, the distribution of datapoints within clusters are analysed along different axes. Namely, from the dataset containing 100 dimensions and 4 clusters, 2 clusters are used for examination. For these two clusters the minimum, median, and maximum variance, and the distribution of datapoints along them are plotted in Figs 1 and 2.

These plots show 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.

B. External Validation Measure

The F-Score [28] of a cluster is defined as the square root of the product of precision and recall. These are defined as the standard way, as shown in Equations (5), (6), and (7). 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 when two datapoints share a label and are also in the same cluster, a FP is when two datapoints are within the same cluster but do not share the same label, and a FN is when two datapoints share a label but are not in the same cluster [29].

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \tag{5}$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \tag{6}$$

$$\text{F} = \sqrt{Precision \times Recall} \tag{7}$$

C. Structure of Experiments

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

1) Baseline using the relevant algorithm with Euclidean distance on the original full feature set; and
2) PSO-FSW generating a final partitioning through PSO-FSW using the novel distance based internal validation measure CSC as the optimisation criterion.

For each clustering algorithm results were compared between the PSO-FSW trial and the baseline using an unpaired Wilcoxon test.

D. Parameter Settings

There are a number of parameters in the proposed PSO-FSW and the clustering methods in the experiments, which are listed below. No parameters needed to be set for Agglomerative and Affinity Propagation.

PSO-FSW: The parameters selected for the PSO algorithm are the ones suggested in [30]. 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 swarm size is 20; and the termination criterion is 30 iterations, or if there is no improvement in $gbest$ for 5 iterations.

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.

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].

V. Results and Discussions

The mean F-score of the experiments and the baseline (using all features) are shown in Table II. The p-value of the Wilcoxon test performed between the baseline and experiment result is also provided. Table IV shows the mean ratio of features used by PSO-FW on each clustering algorithm and number of dimensions. Fig. 3 shows the F-Scores achieved by the proposed PSO-FSW method with the clustering methods, with all features, different dimensionality, and different number of clusters. Results for further investigation of having different clustering methods within PSO-FSW are shown in Tables V and VI.

A. Overall Results

Overall, the results of PSO-FSW show significant improvement in F-Score and Silhouette over all respective baselines, as shown in Fig. 3(a), Table II, and Table III. The performance will be further discussed in terms of the number of features selected, the dimensionality, and the number of clusters. The Silhouette values shown above are calculated using the full-unweighted feature set, thus the novel algorithm was not explicitly optimising for this value.

TABLE II
Mean F-Score by Algorithm.

	Baseline	PSO-FSW	P-Value
Affinity Prop.	0.577	**0.638**(+)	1.62×10^{-28}
KNN-Clustering	0.844	**0.930**(+)	1.90×10^{-23}
DBSCAN	0.795	**0.816**(+)	6.42×10^{-08}
Agglomerative	0.539	**0.697**(+)	2.67×10^{-30}

TABLE III
Mean Silhouette by Algorithm.

	Baseline	PSO-FSW	P-Value
Affinity Prop.	0.365	**0.381**(+)	1.66×10^{-16}
KNN-Clustering	0.300	**0.413**(+)	2.30×10^{-31}
DBSCAN	0.268	**0.374**(+)	1.77×10^{-28}
Agglomerative	0.418	**0.443**(+)	1.98×10^{5}

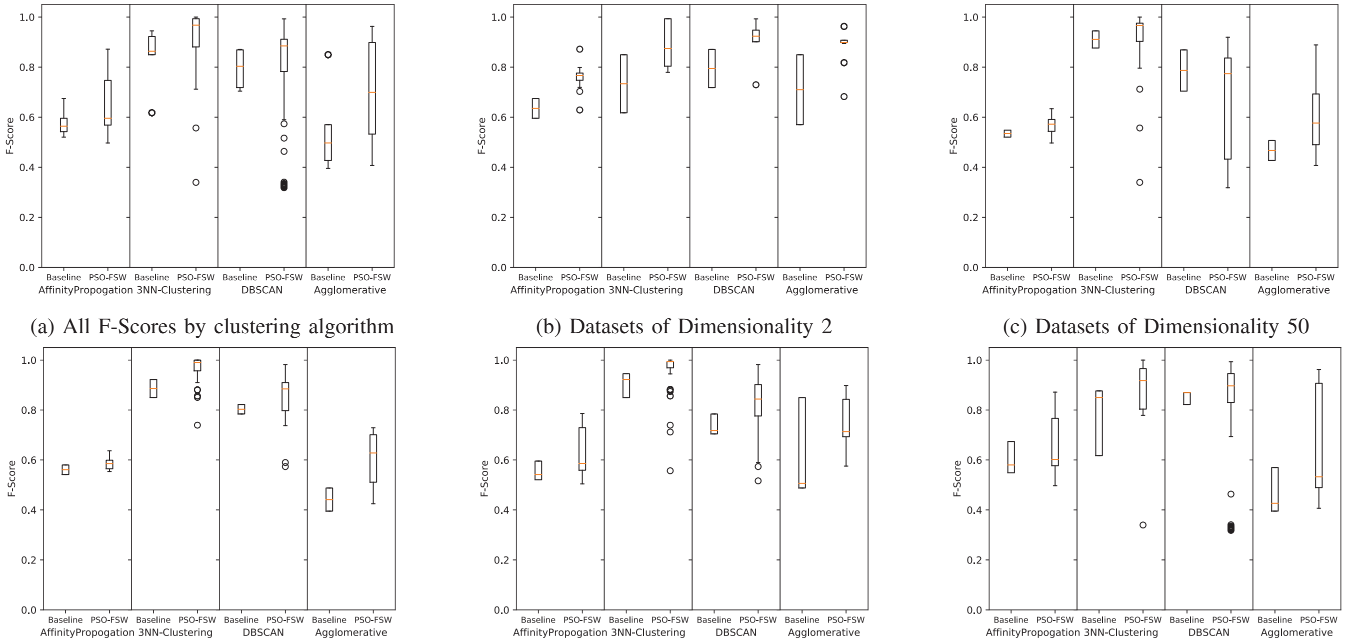
Note: Affinity Propagation had undefined baseline Silhouette due to singleton clusters on all two dimensional datasets. Thus these datasets were not included for Affinity Propagation in this table.

Fig. 3(a) shows that PSO-FSW improved the performance of all the four clustering methods using all features. This can be confirmed by the significantly higher F-Score values as shown in Table II (where "+" means PSO-FSW is significantly better than the baseline) especially for KNN-Clustering and Agglomerative. There are also outliers with low F-Score, especially when using DBSCAN with the weighted feature subset from PSO-FSW, although overall it is significantly improved from the baseline. This will be investigated in the future. By looking at Table IV, which shows the mean ratio of features selected and weighted by PSO-FSW, it can be seen that the number of features have been significantly reduced, making the clusters more interpretable. Especially for high-dimensional datasets with dimensionality of 50 and 100, where on average approximately half of the original features are selected and weighted, but still leads to better performance.

Figs. 3(b-d) show the results on datasets with different dimensionalities, i.e. 2, 50 and 100. It can be observed that the improvement is relatively consistent across dimensionalities, with the exception of DBSCAN on the 50 dimensional datasets. It can be found that PSO-FSW appears to lead to some low F-Score outliers when used with 3NN-Clustering

TABLE IV
Mean Ratio of Features Used.

	$D = 2$	$D = 50$	$D = 100$
Affinity Propagation	0.983	0.508	0.495
KNN-Clustering	1.0	0.511	0.487
DBSCAN	1.0	0.524	0.524
Agglomerative	1.0	0.490	0.492



(a) All F-Scores by clustering algorithm

(b) Datasets of Dimensionality 2

(c) Datasets of Dimensionality 50

(d) Datasets of Dimensionality 100

(e) Datasets with 4 Clusters

(f) Datasets with 10 Clusters

Fig. 3. F-Scores with all features, different dimensionality, and different number of clusters.

and DBSCAN, where the dimensionality of datasets is greater than 2.

Figs. 3(e-f) shows the results on datasets with different number of clusters, i.e. 4 and 10. It is also important to note the interaction between the number of clusters present in a dataset and the effectiveness of the novel technique. It can be seen that the improvement in F-Score overall seems more pronounced with a smaller number of clusters. However the improvement is still evident when the number of clusters is 10.

B. Further Analysis

PSO-FSW improved the results over the baseline of using the original feature set in most cases, but there were still outliers with poor clustering solutions. The analysis in this section is primarily focused on explaining why these behaviours occurred, as well as showing the specific characteristics of some final partitionings. DBSCAN is used here as an example since outliers are observed when DBSCAN was used with PSO-FSW for the dataset with 50 dimensions and 10 clusters.

The overall results for this analysis are presented in Table V, where D indicates the dimensionality, K_{real} represents the actual number of clusters used when generating the dataset, μ_{base} and μ_{test} shows the average value of F-Scores when using the baseline of all features and the feature subset produces by PSO-FSW, σ_{test} shows the standard deviation, p-value is generated from the Wilcoxon test, and K_{base} and K_{test} denote the number of clusters generated by DBSCAN when using the baseline of all features and the feature subset produces by PSO-FSW.

TABLE V
RESULTS OF PSO-FSW WHEN USED WITH DBSCAN.

D	K_{real}	μ_{base}	μ_{test}	σ_{test}	p-value	K_{base}	K_{test}
2	4	0.72	**0.89(+)**	0.042	2.9e-11	7.0	3.0
2	10	0.87	**0.95(+)**	0.014	2.9e-11	11.0	10.13
50	4	0.70	**0.77(+)**	0.097	0.004	5.0	4.5
50	10	0.87	*0.57(-)*	0.25	1.0e-07	14.0	8.6
100	4	0.78	0.83(=)	0.11	0.076	6.0	5.7
100	10	0.82	**0.89(+)**	0.037	8.1e-09	15.0	11.13

Table V shows that PSO-FSW significantly improved the performance in most cases, indicated by "+". There is only one worse case indicated by "-", where the standard deviation σ_{test} is very big, suggesting there are poor outliers generated. The results also show that the number of clusters generated by PSO-FSW, K_{test}, is closer to the actual number of clusters K_{real} than that of using the baseline of all features, K_{base}.

It is also important to determine whether the PSO algorithm is unable to find good optima, or whether good optima are found corresponding to poor solutions. Therefore, we further investigate individual solutions to confirm this point. Two solutions, one with a high F-Score and one with a low F-Score on the dataset with 50 dimensions and 10 clusters are used as examples, to compare the fitness values, i.e. the CSC scores for these resulting solutions. The results are presented in Table VI.

The results in Table VI show that PSO-FSW is finding a better solution but achieving a worse F-Score. This indicates that the poor solutions represent something approaching a trivial solution, rather than insufficient optimisation. It is further

TABLE VI
COMPARISON OF INDIVIDUAL SOLUTIONS

	F-Score	CSC	K	Selected Features (%)
Good Solution	0.919	9.60	12	50%
Poor Solution	0.327	11.30	5	46%

noted that both of the lowest base-truth Silhouette results occur on the 50 dimensional dataset with 10 clusters, but that this behaviour not seen frequently for the 50 dimensional dataset with 4 clusters. It seems this result is an interaction between a higher number of cluster and a dataset for which the base-truth Silhouette is low relative to the 100 dimensional cases. It's clear that the novel algorithm, which is designed to address problems on high-dimensional data, is not failing to improve results when paired with DBSCAN as dimensionality increases. The evidence for this is that on the 100 dimensional dataset on which it actually does better.

VI. CONCLUSIONS

This paper aimed to propose a new PSO based method to achieve simultaneous feature selection and feature weighting for high-dimensional clustering problems. This goal has been successfully achieved by developing a new particle representation encoding information about selecting and weighting features, and a new validation measure forming the fitness function. The datasets were carefully chosen based on initial analysis to ensure the difficulty and to have different dimensions and different clusters. Four different clustering algorithms were chosen to represent different types of clustering approaches in the experiments and comparisons. The experimental results show that with the features selected and weighted by PSO-FSW, all the four different clustering algorithms achieved better performance in terms of F-Score on datasets with different dimensionalities and different number of (base-truth) clusters.

Besides the largely improved performance, there are also outliers with poor F-Score. Further analysis revealed that there exist solutions leading to trivial clusters, which is a key challenge in feature selection for clustering, as we discussed in Section I. In the future, we will investigate how to effectively avoid or eliminate such situations, but avoiding trivial clusters is very challenging, which requires appropriate design of similarity or distance measures for clustering and the validation measure in the fitness function.

REFERENCES

[1] E. Backer and A. K. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 66–75, 1981.

[2] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE transactions on emerging topics in computing*, vol. 2, no. 3, pp. 267–279, 2014.

[3] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[4] H. Xiong and Z. Li, "Clustering Validation Measures." 2013.

[5] C. C. Aggarwal and C. K. Reddy, Eds., *Data Clustering: Algorithms and Applications*. CRC Press, 2014.

[6] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When Is "Nearest Neighbor" Meaningful?" in *Proceeding of the 7th International Conference on Database Theory*, ser. Lecture Notes in Computer Science, vol. 1540. Springer, 1999, pp. 217–235.

[7] M. Köppen, "The curse of dimensionality," in *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, 2000, pp. 4–8.

[8] W. Sheng, X. Liu, and M. Fairhurst, "A niching memetic algorithm for simultaneous clustering and feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 7, pp. 868–879, 2008.

[9] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation (CEC)*, 1998, pp. 69–73.

[11] A. A. Esmin, R. A. Coelho, and S. Matwin, "A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 23–45, 2015.

[12] E. R. Hruschka, R. J. Campello, A. A. Freitas *et al.*, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 2, pp. 133–155, 2009.

[13] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[14] M. Steinbach, L. Ertoz, and V. Kumar, "Challenges of Clustering High Dimensional Data," in *New Vistas in Statistical Physics – Applications in Econophysics, Bioinformatics, and Pattern Recognition*, L. T. Wille, Ed. Springer-Verlag, 2003.

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

[16] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 1977.

[17] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[18] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[19] M. Dash and H. Liu, "Feature selection for clustering," in *Knowledge Discovery and Data Mining. Current Issues and New Applications*, T. Terano, H. Liu, and A. L. P. Chen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 110–121.

[20] J. G. Dy and C. E. Brodley, "Feature subset selection and order identification for unsupervised learning," in *ICML*, 2000, pp. 247–254.

[21] A. Lensen, B. Xue, and M. Zhang, *Using Particle Swarm Optimisation and the Silhouette Metric to Estimate the Number of Clusters, Select Features, and Perform Clustering*. Springer International Publishing, 2017, vol. 10199, pp. 538–554.

[22] D. S. Modha and W. S. Spangler, "Feature weighting in k-means clustering," *Machine learning*, vol. 52, no. 3, pp. 217–237, 2003.

[23] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

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

[25] J. Handl and J. Knowles, "Cluster generators for large high-dimensional data sets with large numbers of clusters," *Dimension*, vol. 2, p. 20, 2005.

[26] ——, "Improvements to the scalability of multiobjective clustering," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 3. IEEE, 2005, pp. 2372–2379.

[27] S. Klie, Z. Nikoloski, and J. Selbig, "Biological cluster evaluation for gene function prediction," *Journal of Computational Biology*, vol. 21, no. 6, pp. 428–445, 2014.

[28] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[29] A. Lensen, B. Xue, and M. Zhang, "Particle swarm optimisation representations for simultaneous clustering and feature selection," in *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016, Athens, Greece, December 6-9, 2016*, 2016, pp. 1–8.

[30] F. Van Den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, University of Pretoria, 2007.