PSO Aided k-Means Clustering: Introducing Connectivity in k-Means

Mihaela Elena Breaban
Faculty of Computer Science
Alexandru Ioan Cuza University
Iasi, Romania
pmihaela@infoiasi.ro

Henri Luchian
Faculty of Computer Science
Alexandru Ioan Cuza University
Iasi, Romania
hluchian@infoiasi.ro

ABSTRACT

Clustering is a fundamental and hence widely studied problem in data analysis. In a multi-objective perspective, this paper combines principles from two different clustering paradigms: the connectivity principle from density-based methods is integrated into the partitional clustering approach. The standard k-Means algorithm is hybridized with Particle Swarm Optimization. The new method (PSO-kMeans) benefits from both a local and a global view on data and alleviates some drawbacks of the k-Means algorithm; thus, it is able to spot types of clusters which are otherwise difficult to obtain (elongated shapes, non-similar volumes). Our experimental results show that PSO-kMeans improves the performance of standard k-Means in all test cases and performs at least comparable to state-of-the-art methods in the worst case. PSO-kMeans is robust to outliers. This comes at a cost: the preprocessing step for finding the nearest neighbors for each data item is required, which increases the initial linear complexity of k-Means to quadratic complexity.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Algorithms, Design, Experimentation

Keywords

multi-criteria clustering, k-Means, PSO, hybridization

1. INTRODUCTION

Clustering is a fundamental exploratory analysis problem akin to discovering natural groupings in data [1]. It is known to be a hard optimization problem mainly in its unsupervised version. The lack of any knowledge, except the data themselves, poses important challenges at different stages in clustering analysis: choosing the right distance function and the relevant features, defining the clustering criteria, validating the solution.

Clustering has a huge search space and a vaguely defined optimum; these characteristics of the problem are addressed in the existing literature by means of various heuristics and experimental studies.

Different paradigms optimize different criteria and thus deliver different partitions in data. Algorithms like K-Means or hierarchical forms (such as Ward's algorithm and average link) account for the global structure/distribution in data. There are also algorithms which exploit only local properties and stipulate that neighboring data items should share the same cluster. This approach to clustering is called the *connectivity principle* and is implemented in the single-link hierarchical algorithm and in density-based methods like DBSCAN.

Each of the algorithms mentioned above has advantages and disadvantages with regard to the computational time and parameters tuning. More importantly, they deliver different solutions. Each algorithm is appropriate to a specific distribution in data. The k-Means algorithm is very effective with regard to the computational time or parameter tuning but is applicable to gaussian clusters of equal volumes. The connectivity principle yields clusters of various shapes but the methods implementing it may suffer from the 'chaining effect' that causes undesirable elongated clusters, or are very sensitive to parameters.

Soft computing techniques were proposed to alleviate some of the drawbacks of the traditional clustering algorithms. Most of them minimize the within-cluster variance, being inspired by the k-Means algorithm.

There are a few strategies designed for clustering that optimize simultaneously several criteria. The use of multi-objective evolutionary algorithms is one of the most important contributions in this regard: the solution is evolved considering several criteria [8].

The current paper proposes a hybridization of k-Means with a Swarm Intelligence technique, aiming at enhancing the performance of the traditional clustering algorithm. Our method is consistently different from existing approaches for clustering based on Swarm Intelligence. Particle Swarm Optimization is used to introduce the connectivity principle into the centroid-based clustering algorithm; the new method thus takes into account both the local and global distribution of data.

Section 2 summarizes the use of Evolutionary Compu-

[1] Where no other explicit statement is made, we imply the partition version of clustering.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

tation techniques for clustering with emphasis on Particle Swarm Optimization (PSO). Section 3 presents the hybridization we propose. The experiments are described in sections 4 and 5, and conclusions are drawn in Section 6.

2. EVOLUTIONARY AND SWARM INTELLIGENCE TECHNIQUES IN CLUSTERING

In general, in Evolutionary Computation (EC), a population of complete solutions evolves during an iterative process, and consequently most approaches to clustering based on EC techniques are **relocation methods**, that improve initially generated partitions. Several encodings were proposed to represent partitions: the **straightforward group-encoding representation** [14], **permutations** [12], **boolean** $k \times n$ **matrices** [2]. One of he most popular encoding is the **string of cluster representatives** introduced in [15]. The criteria optimized in these approaches are generally based on the intra-cluster variance and the between-cluster variance.

Unlike the above-mentioned methods, multi-modal evolutionary algorithms were used to search for cluster centers that lie in dense regions in the feature space [7, 16, 21]. Gaussian functions are used to measure the fit of cluster centroids. A complete clustering solution (partition) is then constructed based on all the individuals in the population.

Sarafis et al. [19] use a **grid-based approach** and a genetic algorithm to search for a partition of the feature space that implicitly provides a partition of the data set. The algorithm evolves rules which build a grid in the feature space. Each individual consists of a set of k clustering rules, each rule corresponding to one cluster. Each rule is encoded in m genes and each gene corresponds to an interval involving one feature. The authors attempt to alleviate certain drawbacks related to the classical minimization of square-error criterion by suggesting a fitness function that takes into consideration cluster asymmetry, density, coverage and homogeneity. The method is able to discover clusters of various shapes, sizes and densities. This comes at a high computational cost due to the form of the fitness function.

A notable contribution the field of Evolutionary Computation made to clustering is the use of multi-objective algorithms, which allow for simultaneous optimization of several criteria. Handl et al. [8] optimize both intra-cluster variance and connectivity. The encoding they use is the **locus-based adjacency representation**. A value j assigned to the ith gene, is interpreted as a link between data items i and j: in the resulting clustering solution they will be in the same cluster. The decoding of this representation requires the identification of all connected components. All data items belonging to the same connected component are then assigned to one cluster. The representation is well-suited for standard crossover operators. Moreover, in conjunction with an objective function based on connectivity, this encoding allows for discovering clusters of various shapes. The method improves substantially over traditional methods like k-Means and hierarchical versions, methods which optimize a single objective out of the two under consideration.

Inspired by dimensionality reduction techniques, Swarm Intelligence algorithms were designed to embed the original data set into a lower-dimensional feature space which preserves the topological relationships among data items. ACO was used to arrange data items within the cells of a two-dimensional grid, a representation well-known from Self Organizing Maps (Kohonen, 1995); a rigorous study on the performance of this approach can be found in [10].

Mostly because of its design for continuous optimization, most approaches to clustering based on PSO use the centroid-based encoding presented in [15] and search for cluster representatives [17]. The performance of these approaches is compared to the standard k-Means algorithm (with the real number of clusters) and is reported to be significantly better - or equal in performance in case k-Means is supplied the best initial configuration. The improved performance is due to the increased exploration capabilities, eliminating one important drawback: strong dependency on initialization. However, other drawbacks may still be present: the result is dependent on the metric used and clusters with similar shapes and volumes tend to be formed. A survey on Swarm Intelligence techniques applied to clustering can be found in [1].

A mapping of the original data set into a two-dimensional Euclidean space is performed using simple PSO rules in [20]; although a metric space is employed, the approach is not aimed at generating an embedding of the original data which faithfully preserves the original pairwise distances among data items (as in Multidimensional Scaling approaches); the focus is on identifying clusters through species separation metaphor. Breaban et al. [3] use a similar technique to find communities in social networks.

The method we propose in the next section is distinct from the existing approaches: inspired by the multi-objective perspective, k-Means is used to enforce clustering from the global distribution in data, whereas PSO is used to enforce the connectivity principle.

3. INTRODUCING CONNECTIVITY IN K-MEANS

K-Means is the most popular clustering algorithm due to its simple implementation, low run-time and space complexity and simple usage since no parameters (except the number of clusters) are involved. However, these advantages come at a cost: due to the local search, the performance is highly dependent on initialization. Moreover, k-Means best fits data sets with spherical clusters of almost-equal volumes.

The first drawback is partially alleviated if smarter initialization schemes are used. The initial centroids should be placed far apart, or a hierarchical clustering method may be used to return an initial partition over a small sample of the data set. The most comfortable (if time-consuming) way to deal with sensitivity to initialization is to run k-Means repeatedly with random initializations and choose the one with the lowest intra-cluster variance.

The second drawback, appropriateness only for particular types of clusters, is present in all clustering algorithms based on representatives: under the Euclidean metric spherical clusters are generated. Even if the centroids-based clustering methods based on Genetic Algorithms or Swarm Algorithms tackle dependency on initialization, they cannot generate clusters of various shapes.

In order to deal with clusters of various shapes, a locality concern may be used: ”neighboring” data items should share the same cluster. We propose a Swarm algorithm called PSO-kMeans which implements this simple connectivity principle and introduces it within k-Means, taking thus into account simultaneously the local and the global distribution in data.

3.1 Basic PSO

Particle Swarm Optimization (PSO) [13] is a meta-heuristic mainly used for numeric optimization. Its use in combinatorial optimization necessitates rather complex adaptations, such as the redefinition of its operators. Initially, PSO was intended to simulate the social behavior of flocks but its authors observed the optimization capability of the agents involved in the simulation.

PSO maintains a population of particles, each one characterized by a position vector x in the search space and a velocity vector v which determines its motion. The velocity vector is computed following the rules:

- each particle tends to keep its current direction (an inertia term);
- each particle is attracted to the best position p it has achieved so far (personal best);
- each particle is attracted to the best particle g in population (the particle having the best fitness value); there are versions of the algorithm in which the best particle g is chosen from a topological neighborhood.

The velocity vector is computed as a weighted sum of three terms corresponding to the rules above. Two random multipliers r_1, r_2 are used to gain stochastic exploration capability while w, c_1, c_2 are weights usually empirically determined. The formulae used to update each individual in the population at iteration t are:

$$v_i^t = w\cdot v_i^{t-1} + c_1\cdot r_1\cdot(p_i^{t-1} - x_i^{t-1}) + c_2\cdot r_2\cdot(g_i^{t-1} - x_i^{t-1}) \quad (1a)$$

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

3.2 Hybridization

The connectivity principle was introduced in different forms in clustering algorithms. In density-based approaches it works towards putting in the same cluster neighboring data items that form dense regions. Handl et al. [9] formulate and maximize explicitly by means of GAs, a connectivity objective defined as the number of neighboring data items which reside in the same cluster.

We propose another approach for implementing the connectivity principle, mainly inspired from metric learning. In supervised metric learning, a metric is learned so that the distance between similar data items is as small as possible under the new metric. In our unsupervised context, similarity is defined with respect to the Euclidean distance: neighboring data items in the Euclidean space are considered to be similar. Consequently, in PSO-kMeans the distance between neighboring data items is shortened by modifying their representation in the fraim of the PSO paradigm.

Generally, in solving optimization problems with PSO, the position vectors x correspond to complete candidate solutions and the particles p and g which dictate the particles' motion are chosen from the population with regard to a fitness/objective function from the population.

In our approach, particles are used to link data items to clusters. Each data item is assigned to a particle in the swarm. The feature space defined by the data set provides the environment for the swarm of particles. The position vector x of each particle is initialized with the feature vector of the corresponding data item. The original PSO rules that dictate the motion of each particle are used to change the representation of the corresponding data item. No objective function is explicitly formulated, but through an appropriate definition of the vectors p and g in equation 1a the connectivity is maximized.

Each particle updates its position to match its nearest neighbors. With this aim, each particle x_i should move iteratively towards each of its neighbors. In order to reduce the run time, a centroid over the neighbors is computed and the particle moves towards it. This centroid plays the role of "p_i" in formula 1a. Its use accounts for local distribution in data.

To take into account the global distribution in data, "g_i" is defined to be the centroid closest to particle i in the partition returned by k-Means.

Our clustering algorithm is presented in pseudocode 1.

Algorithm 1 PSO-kMeans

```
Require: The set of data items $D = \{x_1, x_2, ..., x_n\}$, the number of clusters $k$.
Ensure: a hard partition $C = \{C_1, C_2, ..., C_k\}$, $\bigcup_{i=1}^{k} C_i = D$ and $C_i \bigcap C_j = \emptyset \forall i, j = \overline{1..n}$.

  // preprocessing step:
  for all data item $x_i$ do
    $NN_i \leftarrow$ the $n_s$ nearest neighbors for $x_i$
  end for

  // initialization phase:
  apply k-Means until convergence and store:
  $C \leftarrow \{C_1, C_2, ..., C_k\}$, the hard k-Means partition;
  $c_j \leftarrow \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i$ the centroid of cluster $j$, $\forall j = \overline{1..k}$;
  $d_i \leftarrow dist(x_i, c_j)$, $\forall i = \overline{1..n}$, where $c_j$ is the centroid of cluster $C_j \in C$ and $x_i \in C_j$, $dist$ is the Euclidean distance;
  $\sigma^2 \leftarrow \frac{1}{n}\sum_{i=1}^{n} d_i^2$ (approximates the variance within clusters)

  //the PSO-kMeans iterations:
  while $C$ has not changed for $itr$ iterations do
    //run one PSO iteration:
    for $i \leftarrow 1$ to $n$ do
      $p_i \leftarrow \frac{1}{|NN_i|}\sum_{x_j \in NN_i} x_j$
      $g_i \leftarrow c_j$ s.t. $x_i \in C_j$
      update $x_i$ applying formulae 2
    end for

    //run one k-Means iteration:
    for $i \leftarrow 1$ to $n$ do
      reassign $x_i$ to $C_j$, where $C_j = argmin_{c_l, l=1..k}\{dist(x_i, c_l)\}$
    end for
    for $j \leftarrow 1$ to $k$ do
      $c_j \leftarrow \frac{1}{|C_j|}\sum_{x_i \in C_j} x_i$
    end for
  end while
```

A pre-processing step is required to find the n_s nearest neighbors of each particle. This set of neighbors is computed only once, at the beginning, and is not modified throughout the run. In this way, subsequent changes of the positions of the data items, resulting from changes in the representation of the data items to be clustered, preserve much of the initial topology.

The batch version of the k-Means algorithm is run until convergence. The centroids retrieved with k-Means serve further as g_i in the first iteration of PSO.

Then, an iterative process begins that alternates iterations of PSO and k-Means until a stable partition is reached. The PSO iteration consists of recomputing p_i and applying formulae 2 which modify each data item x_i in the data set. The particle p_i is updated using the set of neighboring data items computed in the pre-processing step; because all particles/data items are subject to the PSO updating rules, the configuration of the neighborhood changes implicitly.

The k-Means iteration reconstructs the partition and reassigns the modified data items to the previously found centroids. The centroids g_i are updated.

The initial velocity is set to 0 for all particles. The random multipliers in formula 1a of the basic PSO are not needed. The weights for the inertia term and for the p_i term are set to 1. Preliminary experiments showed that the inertia term has an important influence on the speed of convergence: the number of iterations in PSO-kMeans reduces to almost a half for some data sets in its presence, compared to the case when it is not used at all.

If a unit weight is assigned as well to the third term in equation 1a, the impact of our hybridization is much reduced: almost all particles will end up in the centroids identified with k-Means on the original representation of the data set. Generally in k-Means, for any given cluster the data items situated closer to the centroid are more likely to belong to the corresponding cluster than the data items situated farther (fuzzy k-Means originates from this principle). This is why we apply the third update rule in equation 1a (the move towards the cluster centroid) on only 10% of the data items situated closest to the centroid. For clusters obeying the normal distribution, 10% of the data items are supposed to lie within a distance of 0.125 standard deviations from the centroid. To reduce the computational cost, we adopt the hypotheses of normal distribution and apply the third updating rule on the data items satisfying the property above. Obviously, this does not imply that our method could not be used on other types of distributions. The average within-cluster variance σ^2 is computed in the k-Means iteration when data items are assigned to clusters. A vector of length n (the number of data items) is used to store at this step the distances between each data item i and its closest centroid g_i. Using the average over all clusters of the within-cluster variance instead of the exact values for each cluster, brings some advantages. Well-initialized cluster centroids will "consume" most of this rule compared to wrongly placed centroids. The particles on the boundary of the clusters are attracted to their neighbors situated closer to the centroids and migrate together to the center of the cluster, leading to more stable clusters.

$$v_i^t = v_i^{t-1} + (p_i - x_i^{t-1}) + w(i) \cdot (g_i - x_i^{t-1}) \tag{2a}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{2b}$$

$$w(i) = \begin{cases} 1, dist(x_i^{t-1}, g_i) < 0.125 \cdot \sigma \\ 0, otherwise \end{cases} \tag{2c}$$

4. EXPERIMENTS

We conducted experiments with PSO-kMeans on both synthetic and real-world data sets.

In a first scenario the number of clusters is part of the input; this setting corresponds to *supervised clustering.* In a second scenario some internal clustering validation indexes are used to decide on the optimal number of clusters - scenario denoted as *unsupervised clustering.*

4.1 Data Sets

In order to test the technique we propose, some complex data sets made available by Julia Handl [2] are used:

- a standard cluster model using multivariate normal distributions. Different combinations *number of attributes / number of clusters* are considered. In low dimensions (2 features), the clusters generated are frequently elongated and of arbitrary orientation but in high dimensions (10 features) they tend to become spherical. The clusters have various volumes/densities.

- data sets of high dimensionality consisting of ellipsoidal clusters with the major axis of arbitrary orientation.

For each combination *number of attributes / number of clusters* 10 different problem instances were generated and referred to as the group of problems (#*attributes*)d-(#*clusters*)c. A total of 90 synthetic data sets are thus used in this part of experiments.

The real-world data sets used are Iris, Soybean, and Breast Cancer from UCI Repository [3].

4.2 Experimental Setup

In order to test the performance of PSO-kMeans in the supervised context of clustering, 50 runs of the algorithm were performed for each dataset. Random initialization was used at each run, each cluster centroid being initialized with a randomly chosen data item from the dataset.

A comparison between PSO-kMeans and the batch version of the standard k-Means is presented. The Adjusted Rand Index [11] was computed for the partition derived with k-Means in the initialization phase of PSO-kMeans and then for the partition obtained after running PSO-kMeans. Both the number of iterations required for standard k-Means to reach convergence and the additional number of iterations performed in PSO-kMeans until convergence was reached are reported.

If one would be willing to substitute the standard k-Means with PSO-kMeans, an important concern arises with regard to the optimum number of clusters. Because PSO-kMeans modifies the representation of data items and consequently the distances between them, there exists the risk of breaking one initial cluster into smaller dense clusters; this would mislead the unsupervised clustering analysis. Therefore, experimental analysis is required for the unsupervised scenario.

In the unsupervised context (the number of clusters is not known in advance), the optimal partition can be obtained as follows: k-Means is run iteratively with k in a large range of values and the resulted partitions having different numbers of clusters are evaluated using an unsupervised clustering

[2] http://dbkgroup.org/handl/generators/
[3] http://archive.ics.uci.edu/ml/

criterion. The winning partition and consequently the optimum number of clusters is considered to be the one with the best score.We adopt this scenario to study PSO-kMeans in the unsupervised context: an unsupervised clustering criterion is computed on the modified representation of the data items for the partition provided by the algorithm.

In the unsupervised context, we run PSO-kMeans successively with the number of clusters ranging between 2 and 30. Because the performance of PSO-kMeans is dependent on initialization (as in standard k-Means), for each number of clusters, 10 runs of the algorithm with random initializations were performed; from the 10 partitions resulted, the one with the lowest intra-cluster variance was selected. Eventually, 29 partitions with different numbers of clusters were obtained. From these 29 partitions, the best one was extracted according to unsupervised clustering criteria. For each problem instance the above steps of analysis were repeated 10 times and averages were computed. We report the results obtained under Silhouette Width(SW) [18] and a recently proposed criterion CritC [4] for which experimental studies showed to outperform the well-known Davies-Bouldin Index [6].

There remains to be discussed the neighborhood size - the only parameter of the algorithm not discussed yet To eliminate the need of fine-tuning under costly experimental studies, we base the choice of the value of this parameter on the working assumptions that the size of the smallest cluster in the partition is at least 10% of the average size of all clusters and, moreover, that each cluster contains at least 10 data items (a study on data not obeying these assumptions is ongoing). The size of the neighborhood is therefore computed for each data set automatically to be $n_s = 10\% \cdot (n/k)$. If n_s is less than 10, then the neighborhood size is set to 10.

The partitions returned by the clustering algorithms under test are evaluated against the optimal clustering using the Adjusted Rand Index (ARI).

4.3 Results

Figure 1 illustrates the comparative performance of the standard k-Means and PSO-kMeans. For each problem instance 50 runs of the algorithms were performed. For the synthetic data sets the box plots present the results over all 10 problem instances of each class of problems, summarizing a total of 500 values of the Adjusted Rand Index.

One can observe that the performance of PSO-kMeans is still dependent on the initialization but it improves in most cases the results delivered by standard k-Means. The gain in performance is more obvious in the case of ellipsoidal clusters (data sets 50d-*c).

Table 1 presents the number of iterations required by PSO-kMeans to improve the solution returned by the standard k-Means. Generally, the stronger the difference in performance is between k-Means and PSO-kMeans, the higher the number of iterations required to converge for PSO-kMeans is.

Table 2 presents the results obtained in the unsupervised scenario. In this scenario we address the risk of a bad initialization by running the standard k-Means, in the initialization phase of PSO-kMeans, for 10 times with random initializations. The solution with the lowest intra-cluster variance is chosen and constitutes the basis for further PSO-kMeans iterations. The table presents averages over 10 complete runs of the algorithm (each run benefiting from 10 different

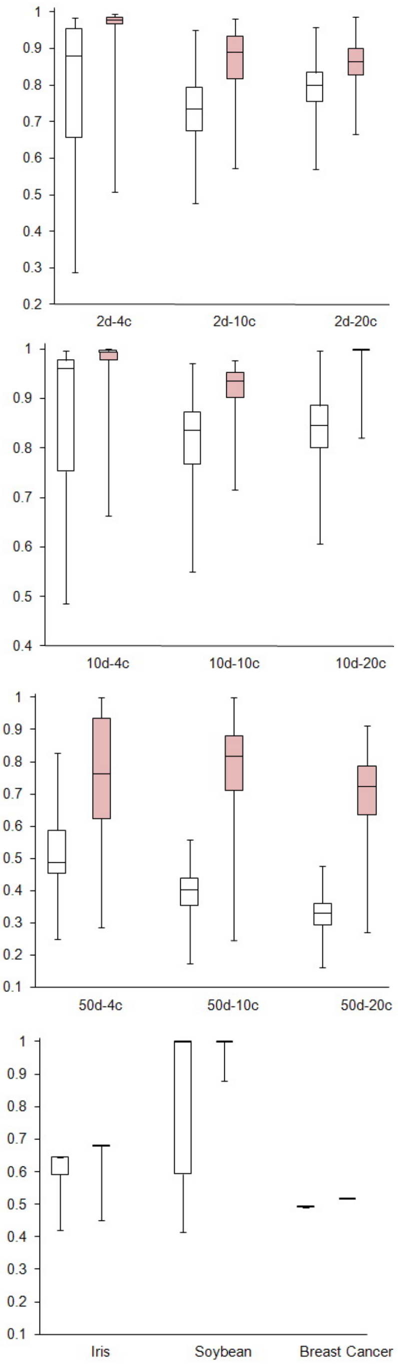

Figure 1: Comparative results for supervised clustering: the first box plot in each group corresponds to the standard k-Means and the second box plot in each group corresponds to PSO-kMeans. Each box plot from the groups *d-*c corresponds to 10 problem instances × 50 runs of the algorithm with random initializations (a total of 500 values of the Adjusted Rand Index). For real-world data sets, the box plots present the values over 50 runs of the algorithms with random initializations.

Table 1: The number of iterations for standard k-Means and the number of additional iterations performed by PSO-kMeans, computed as averages over 50 runs of the algorithm.

Problem	k-Means	PSO-kMeans
2d-4c	18	17
2d-10c	28	71
2d-20c	22	24
10d-4c	19	7
10d-10c	28	39
10d-20c	18	54
50d-4c	19	37
50d-10c	25	80
50d-20c	23	65
Iris	13	9
Soybean	11	2
BCancer	13	3

initialization).

The first 4 columns present the results when the best partition is identified in a supervised manner: the optimal number of clusters is identified by maximizing the value of the Adjusted Rand Index (which is an external validation criterion). It illustrates once again the significant gain in performance if PSO-kMeans is used. It is worth noticing that, compared against standard k- Means, PSO-kMeans provided partitions with slightly higher numbers of clusters.

Even when the partition is chosen using unsupervised criteria, PSO-kMeans still wins the competition with standard k-Means. The Wilcoxon Signed-Rank non-parametric test was applied for all pairs of ARI scores corresponding to (k-Means, PSO-kMeans) under the same criterion. Where differences are significant (at the level 1%) the winner is marked in bold.

Experimental results suggest that PSO-kMeans improves the performance over standard k-Means on all test cases; it achieves this by integrating the connectivity principle in the standard algorithm (neighboring data items should reside in the same cluster). Moreover, the standard k-Means algorithm is sensitive to outliers: the cluster centroids are biased if isolated data items exist because generally the mean is not a stable statistic and extreme values affects it. PSO-kMeans is more robust to outliers which are attracted towards dense regions and do not bias the position of the cluster centroids.

The complexity of a PSO-kMeans iteration is still linear in the number of data items. However, the pre-processing time complexity is $O(n^2)$ as it is necessary to compute the pairwise distances between data items.

As the experiments show, our algorithm does not require parameter tuning in order to increase its performance when dealing with different datasets.

5. COMPARATIVE STUDY

The experimental section suggests the superiority of our method over its main ingredient, standard k-Means. However, we would like to place our method in the wide context of clustering; a comparative analysis with other state-of-the-art clustering methods is in order.

In conducting experiments for such a comparison, we used problem instances presenting various challenges for clustering algorithms: elongated clusters (2 a), data with noise(2 b), spherical clusters of different volumes (2 c) and overlapped clusters (2 d). Table 5 presents the results obtained by various algorithms.

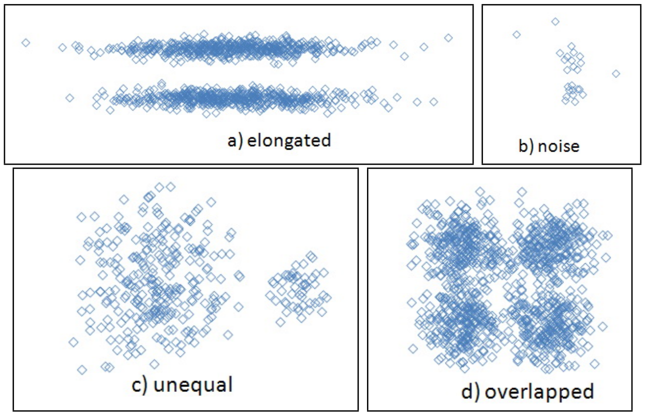

Figure 2: Data sets presenting various clustering challenges

5.1 Centroids-based Methods

The standard k-Means and an existing hybridization of k-Means with PSO [5] are used to study the behavior of centroids-based clustering methods relative to PSO-kMeans which integrates the connectivity principle into the representative-based approaches for clustering.

Figure 3 presents the best partitions obtained with standard k-Means in multiple runs. These partitions are identical to those obtained by the PSO algorithm presented in ([5]) which performs a global search in the space of possible initializations for k-Means.

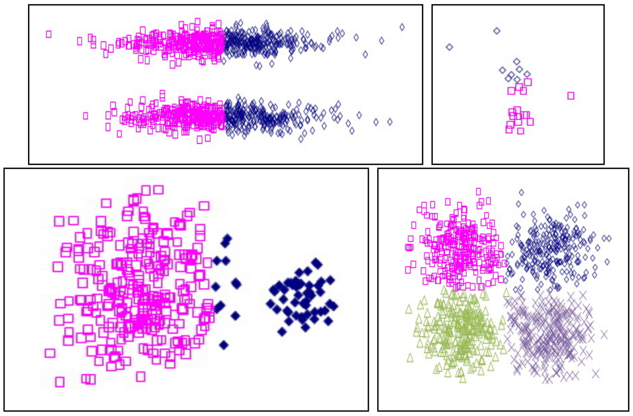

Figure 3: Results obtained by standard k-Means

The main drawback of the centroids-based methods is well illustrated: if clusters of elongated shape or of various volumes are involved, these methods fail to provide the optimal partition.

In case outliers are present in data and no pre-processing step was used to eliminate them, the centroids were biased and erroneous partitions were delivered.

For datasets containing overlapped spherical clusters, the centroids-based methods outperform other strategies.

5.2 Hierarchical Methods

Table 2: Results for unsupervised clustering. For each data set and each algorithm, the ARI and the number of clusters are reported for three partitions: the partition with the highest Adjusted Rand Index (ARI) score, the best partition under Silhouette Width (SW) and the best partition under criterion CritC.

Problem	Best ARI				SW				CritC			
	k-Means		PSO-kMeans		k-Means		PSO-kMeans		k-Means		PSO-kMeans	
	ARI	k	ARI	k	ARI	k	ARI	k	ARI	k	ARI	k
2d-4c	0.92	4.02	**0.98**	4.60	0.87	3.70	**0.95**	5.88	0.85	4.01	**0.94**	5.34
2d-10c	0.83	10.93	**0.94**	10.11	0.78	11.24	**0.91**	12.34	0.79	10.39	**0.90**	11.40
2d-20c	0.91	19.63	0.93	21.24	0.87	16.71	**0.90**	20.70	0.89	17.45	0.90	20.10
10d-4c	0.97	3.99	**0.99**	4.56	0.90	3.59	**0.99**	4.50	0.93	3.5	**0.95**	6.06
10d-10c	0.92	9.21	**0.97**	11.32	0.91	9.03	**0.94**	9.72	0.89	8.36	**0.95**	12.44
10d-20c	0.97	20.23	**0.99**	21.76	0.94	18.05	**0.99**	21.7	0.96	20.44	**0.99**	21.40

Table 3: The ARI computed for the datasets presented in Figure 2: our method(PSO-kMeans), standard k-Means, the clustering method proposed in[5](PSO), 4 hierarchical algorithms and a density-based method.

Problem	PSO-kMeans	k-Means	PSO	Single Link	Average Link	Complete Link	Ward	DBSCAN
elongated	1	0.00	0.00	1	0.00	0.01	1	1
noise	1	0.80	0.93	1	1	1	1	1
unequal	1	0.84	0.86	1	1	0.10	1	1
overlapped	0.90	0.90	0.90	0.00	0.00	0	0.00	0.00

Usually, clusters having different shapes are not a major challenge for hierarchical methods; however, the outcome is highly dependent on the metric used for measuring (di)-similarity between clusters.

For the dataset with elongated clusters, Single link and Ward's method identify correctly the two clusters. Average link and complete link deliver erroneous results as shown in Figure 4.

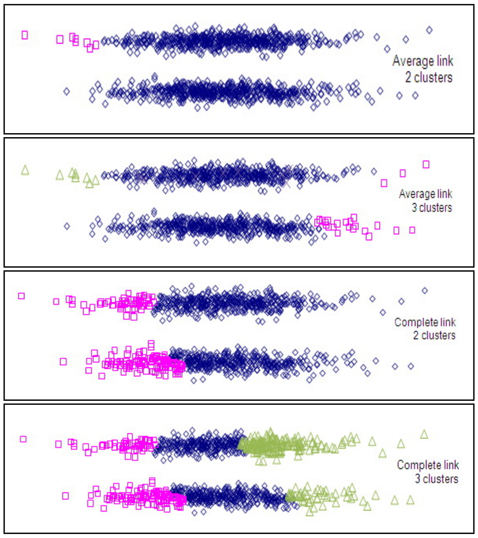

Figure 4: Results for hierarchical algorithms on elongated data

In case of the dataset with clusters of various volumes, all hierarchical methods performed well, except for the Complete link variant.

All hierarchical algorithms identified the noise in case of the data set in figure 2 a (noise).

In case of overlapped clusters, no hierarchical method was able to identify the clusters.

5.3 Density-based Methods

After fine-tuning efforts, DBSCAN identified the clusters in datasets 2a,b,c. It failed to identify the overlapped clusters.

5.4 Discussion

Because our method takes into account the local structure in data implementing the connectivity principle, it is able to identify clusters of different volumes and shapes. Therefore, it identified correctly the clusters for the datasets in figure 2 a and b,having a performance as good as that of Single link, Ward's method and DBSCAN and outperforming all other tested techniques. Experiments showed that PSO-kMeans can outperform its basic ingredient, the standard k-Means algorithm, but also other state-of-the-art algorithms. Tests were performed using a centroid method proposed in [5] which is also based on PSO. As explained in section 2, the existing methods based on PSO or Genetic Algorithms behave like an upper bound for the standard k-Means: they deliver (in the best case) the partition retrieved by a k-Means algorithm if the latter is supplied with the best initialization.

Our method was still able to properly identify the clusters when outliers were present in data (the dataset in figure 2 c). This behavior is due to the change in representation, which causes outlier data items to be attracted towards denser regions. Again, the performance of PSO-kMeans was comparable to hierarchical methods and density based methods and better than that of centroids-based methods.

For the data set with overlapped clusters, the performance of our method was comparable to that of the standard k-Means and significantly superior to that of hierarchical methods and density-based methods. In contrast with the other test-cases, the dataset in figure 2 d illustrates the positive effect of the centroid approach incorporated in our method.

Due to the global view over the data in addition to the connectivity principle, the experiments show that our method is able to outperform state-of-the-art clustering methods or behaves equally-well in the worst case.

6. CONCLUDING REMARKS AND FUTURE WORK

We propose a hybridization of the standard k-Means algorithm with a technique from Swarm Intelligence, with the aim of enhancing the performance of the traditional clustering method. The new algorithm modifies the representation of the data items in order to implement the connectivity principle for clustering. The changes in representation lead to changes in the distribution of distances between data items. Therefore, the new algorithm can be easily tuned to perform semi-supervised clustering. The additional information available in the form of similarity/dissimilarity pairwise constraints should be easily incorporated in the PSO iterations to simulate metric learning along with the clustering process; this idea will be studied in our future work.

7. REFERENCES

[1] A. Abraham, S. Das, and S. Roy. Swarm intelligence algorithms for data clustering. *Soft Computing for Knowledge Discovery and Data Mining, Springer Verlag*, pages 279–313, 2007.

[2] J. C. Bezdek, S. Boggavarapu, L. O. Hall, and A. Bensaid. Genetic algorithm guided clustering. In *International Conference on Evolutionary Computation*, pages 34–39, 1994.

[3] M. Breaban, L. Alboaie, and H. Luchian. Guiding users within trust networks using swarm algorithms. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, CEC'09, pages 1770–1777, Piscataway, NJ, USA, 2009. IEEE Press.

[4] M. Breaban and H. Luchian. A unifying criterion for unsupervised clustering and feature selection. *Pattern Recognition*, In Press, Corrected Proof:–, 2010.

[5] X. Cui, T. E. Potok, and P. Palathingal. Document clustering using particle swarm optimization. In *IEEE Swarm Intelligence Symposium, The Westin*, 2005.

[6] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[7] D. Dumitrescu and K. Simon. Evolutionary prototype selection. In *Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics Ű ICTAMI*, pages 183–190, 2003.

[8] J. Handl and J. Knowles. Improving the scalability of multiobjective clustering. In *Proceedings of the Congress on Evolutionary Computation*, 2005.

[9] J. Handl and J. Knowles. Improving the scalability of multiobjective clustering. In *Proceedings of the Congress on Evolutionary Computation*, pages 2372–2379. IEEE Press, 2005.

[10] J. Handl, J. Knowles, and M. Dorigo. Ant-based clustering and topographic mapping. *Artificial Life*, 12, 2005.

[11] A. Hubert. Comparing partitions. *Journal of Classification*, 2:193–198, 1985.

[12] D. R. Jones and M. A. Beltramo. Solving partitioning problems with genetic algorithms. In *4th International Conference on Genetic Algorithms*, pages 442–45O, 1991.

[13] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of the 1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[14] R. Krovi. Genetic algorithms for. clustering: A preliminary investigation. In *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, pages 540–544. IEEE Computer Society Press, 1991.

[15] S. Luchian, H. Luchian, and M. Petriuc. Evolutionary automated classification. In *Proceedings of 1st Congress on Evolutionary Computation*, pages 585–588, 1994.

[16] O. Nasraoui, E. Leon, and R. Krishnapuram. Unsupervised niche clustering: Discovering an unknown number of clusters in noisy data sets. In A. Ghosh and L. Jain, editors, *Evolutionary Computation in Data Mining*, volume 163 of *Studies in Fuzziness and Soft Computing*, pages 157–188. Springer Berlin / Heidelberg, 2005.

[17] T. Niknam and B. Amiri. An efficient hybrid approach based on pso, aco and k-means for cluster analysis. *Appl. Soft Comput.*, 10:183–197, January 2010.

[18] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(1):53–65, 1987.

[19] I. Sarafis, A. M. S. Zalzala, and P. W. Trinder. A genetic rule-based data clustering toolkit. In *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02*, CEC '02, pages 1238–1243, Washington, DC, USA, 2002. IEEE Computer Society.

[20] C. Veenhuis and M. Koeppen. Data swarm clustering. *Swarm Intelligence in Data Mining, Springer Berlin / Heidelberg*, pages 221–241, 2006.

[21] D. Zaharie. Density based clustering with crowding differential evolution. *Symbolic and Numeric Algorithms for Scientific Computing, International Symposium on*, pages 343–350, 2005.