α-Clusterable Sets

G.S. Antzoulatos, M.N. Vrahatis

Computational Intelligence Laboratory (CILab)
Department of Mathematics, University of Patras
http://cilab.math.upatras.gr

University of Patras Artificial Intelligence Research Center (UPAIRC)

European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) 2011
5 - 9 September, 2011
Athens, Greece

Outline

What is Clustering?

Clustering is...

the process of identifying sets of *similar* items, called **clusters**.
The *goal of a clustering algorithm* is to produce a set of clusters with high intra-cluster similarity while simultaneously preserving a low inter-cluster similarity

A Categorization of Major Clustering Algorithms

- Hierarchical algorithms
  - Agglomerative (bottom-up)
  - Divisive (top-down)
- Partitioning algorithms
  - Distane - based algorithms such as k-means, fuzzy c-means
  - Density - based algorithms such as DBSCAN and k-windows

Motivation

Open issues...

- Gap between practical and theoretical foundation of clustering
- Lack of a unified definition of *what a cluster is*, which will be independent of
  - the measure of similarity/ dissimilarity
  - the clustering algorithm

as a concequense...

it is difficult to give an explicit answer to the questions like:

- how many clusters exist in a dataset
- whether a clustering solution is meaningful or not

Contribution

Our goal is to

- provide a theoretical framework for clustering by giving a new definition of *what a cluster could be*, based on the density of a dataset
- present an unsupervised clustering algorithm to detect the clusters

Window Density Function – WDF

Let a d-range of size $\alpha \in \mathbb{R}$ and center $z \in \mathbb{R}^d$ be the orthogonal range $[z_1 - \alpha, z_1 + \alpha] \times \cdots \times [z_d - \alpha, z_d + \alpha]$.
Assume further, that the set $S_{\alpha,z}$, with respect to the set X, is defined as:

$$S_{\alpha,z} = \{y \in X : \ z_i - \alpha \leqslant y_i \leqslant z_i + \alpha, \ \forall \ i = 1, 2, \ldots, d\} \ .$$

Then...

Window Density Function

The **Window Density Function (WDF)** for the set X, with respect to a given size $\alpha \in \mathbb{R}$ is defined as:

$$WDF_\alpha(z) = |S_{\alpha,z}| \ .$$

Example plots of WDF

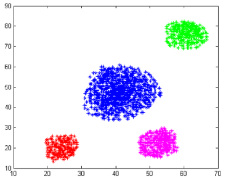


Figure: Dataset of 1600 points forming 4 clusters

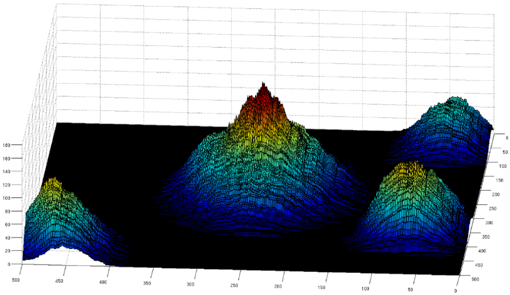


Figure: $\alpha = 0.1$

Example plots of WDF

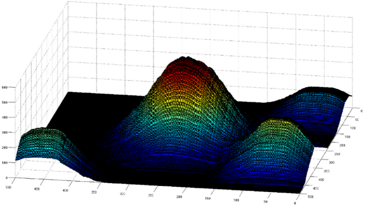


Figure: $\alpha = 0.25$

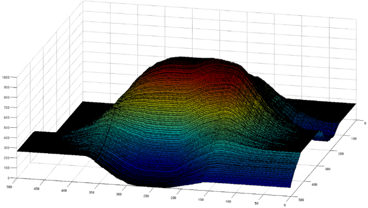


Figure: $\alpha = 0.5$

Basic Notions

α–Clusterable Set

Let the data set $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$. A set of data points $x_m \in X$ is defined as an **α–Clusterable Set**, $C_{\alpha,z}$, if

- $\exists\ \alpha > 0,\ \alpha \in \mathbb{R}$
- a hyper–rectangle $\mathcal{H}_\alpha$ of size α *and*
- $\exists\ z \in \mathcal{H}_\alpha$, z is a centre of $\mathcal{H}_\alpha$

so that the Window Density Function is *unimodal in* $\mathcal{H}_\alpha$. Formally,

$$C_{\alpha,z} = \Big\{ x_m \mid x_m \in X \ \wedge\ \exists\ z \in \mathcal{H}_\alpha \ :\ \mathrm{WDF}_\alpha(z) \geqslant \mathrm{WDF}_\alpha(y), \forall y \in \mathcal{H}_\alpha \Big\}$$

Example of α–Clusterable Set

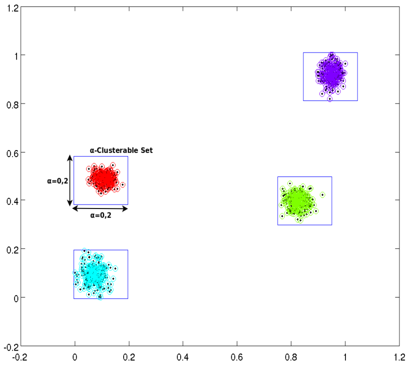


Figure: Dataset of 1000 normal distributed data points forming 4 clusters

Basic Notions

α–Clustering

Given a real value α, an **α–clustering** of a data set X is a partition of X, that is a set of k α–Clusterable Sets of X such that their union is X. Formally, an α–clustering is a set:

$$\mathcal{C} = \left\{ C_{\alpha,z_1}, C_{\alpha,z_2}, \ldots, C_{\alpha,z_k} \right\},$$

where $z_i \in \mathcal{H}_\alpha \subset \mathbb{R}^D$, $i = 1,\ 2,\ \ldots, k$ are the centres of the dense regions C_{α,z_i}

α–Clustering Function

A function $f_\alpha(\mathrm{WDF}_\alpha, X)$ is an **α–clustering function** if for a given window density function, with respect to a real value parameter α, returns a clustering $\mathcal{C}$ of X, such as each cluster of $\mathcal{C}$ is an α–clusterable set of X

α–Clustering Function Properties

Then...

For each dataset X of size $N \geqslant 2$, there is an α–clustering function that satisfies the properties of scale-invariance, richness and consistency

- Scale-Invariance

  in any uniform change in the scale of the domain space of the data, the high-density areas will be maintained and separated by sparse regions of points

- Richness

  there exist a parameter α and points z, such that an α-clustering function f can be constructed, with the property of partitioning the dataset X into α-clusterable sets

- Consistency

  if we shrink the dense areas (α-clusterable sets) and simultaneously expand the sparse areas between the dense areas, then we can get the same clustering solution

α–Clustering Function Properties

Then...

For each dataset X of size $N \geqslant 2$, there is an α–clustering function that satisfies the properties of scale-invariance, richness and consistency

- **Scale-Invariance**

  in any uniform change in the scale of the domain space of the data, the high-density areas will be maintained and separated by sparse regions of points

- **Richness**

  there exist a parameter α and points z, such that an α-clustering function f can be constructed, with the property of partitioning the dataset X into α-clusterable sets

- **Consistency**

  if we shrink the dense areas (α-clusterable sets) and simultaneously expand the sparse areas between the dense areas, then we can get the same clustering solution

α–Clustering Function Properties

Then...

For each dataset X of size $N \geqslant 2$, there is an α–clustering function that satisfies the properties of scale-invariance, richness and consistency

- **Scale-Invariance**

  in any uniform change in the scale of the domain space of the data, the high-density areas will be maintained and separated by sparse regions of points

- **Richness**

  there exist a parameter α and points z, such that an α-clustering function f can be constructed, with the property of partitioning the dataset X into α-clusterable sets

- **Consistency**

  if we shrink the dense areas (α-clusterable sets) and simultaneously expand the sparse areas between the dense areas, then we can get the same clustering solution

α–Clustering Function Properties

Then...

For each dataset X of size $N \geqslant 2$, there is an α–clustering function that satisfies the properties of scale-invariance, richness and consistency

- Scale-Invariance

  in any uniform change in the scale of the domain space of the data, the high-density areas will be maintained and separated by sparse regions of points

- Richness

  there exist a parameter α and points z, such that an α-clustering function f can be constructed, with the property of partitioning the dataset X into α-clusterable sets

- Consistency

  if we shrink the dense areas (α-clusterable sets) and simultaneously expand the sparse areas between the dense areas, then we can get the same clustering solution

Proposed Algorithm

Main goal is...

to identify the dense regions of points. These regions constitute the α–Clusterable Sets that enclose the real clusters of the dataset

Benefits of the algorithm...

- unsupervised clustering algorithm, in the sense that it doesn't require a predefined number of clusters to detect the α–clusterable sets in X
- iteratively defines the correct number of clusters
- simple to implement, since it exploits the PSO algorithm to explore the space for a global optimum of WDF

Steps of the algorithm "PSO α–Cl"

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm "PSO α–Cl"

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm “PSO α–Cl”

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm "PSO α–Cl"

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm "PSO α–Cl"

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm "PSO α–Cl"

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm "PSO α–Cl"

Repeat

1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

Steps of the algorithm "PSO α–Cl"

Repeat

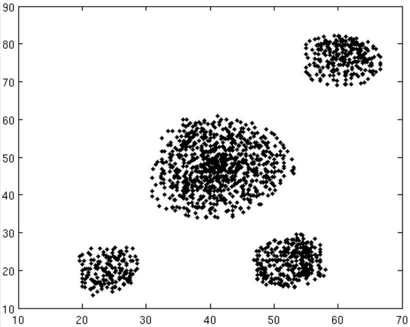
1. **Create** a data structure that holds all unclustered points
2. **Perform** the PSO algorithm returning the centre z of an α–Clusterable set
3. **Construct** the window w of size α around the centre z
4. **Mark** the points that lie in the window w as clustered
5. **Remove** the clustered points from the dataset

Until no left unclustered points

Mark the points that lie in overlapping windows as members of the same cluster and **merge** these windows to form the clusters.

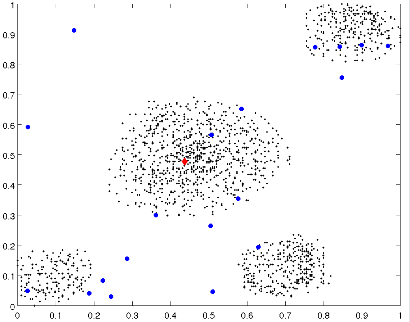
Step 1 of the algorithm

Step 1: Create a data structure that holds all unclustered points of the dataset

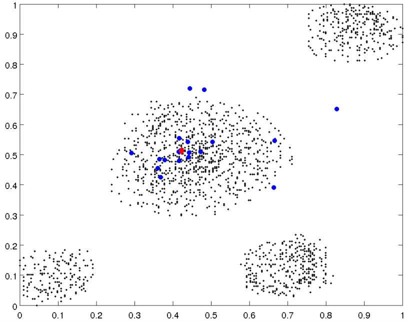


Step 2: Perform the PSO algorithm to return the centre z of an α–Clusterable Set

Step 2: 1^{st} Epoch

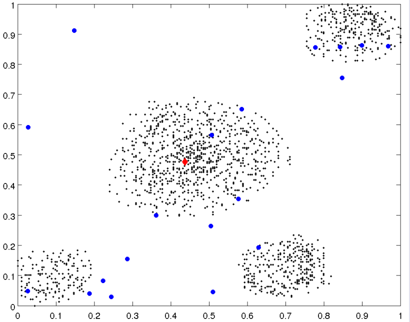


Step 2: 25^{th} Epoch

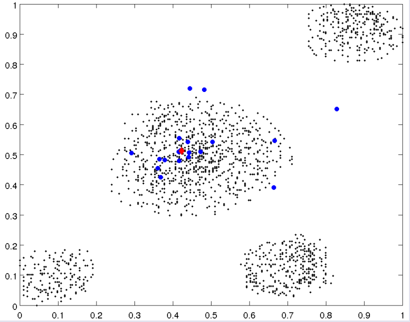


Step 2: Perform the PSO algorithm to return the centre z of an α–Clusterable Set

Step 2: 1^{st} Epoch

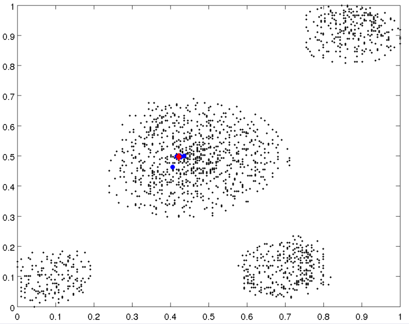


Step 2: 25^{th} Epoch

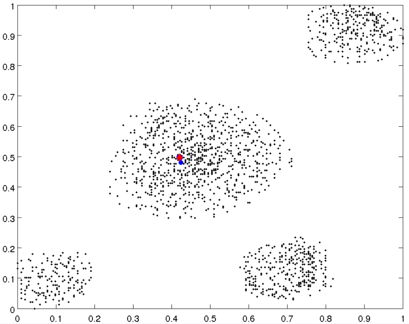


Step 2: Perform the PSO algorithm to return the centre z of an α–Clusterable Set

Step 2: 80^{th} Epoch

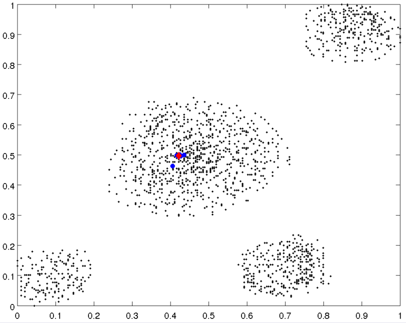


Step 2: 100^{th} Epoch

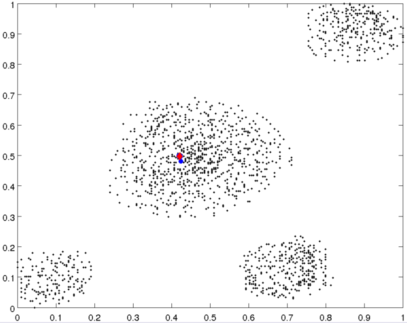


Step 2: Perform the PSO algorithm to return the centre z of an α–Clusterable Set

Step 2: 80^{th} Epoch

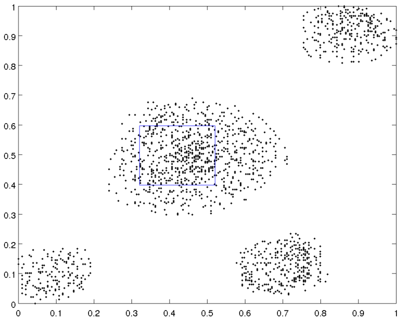


Step 2: 100^{th} Epoch

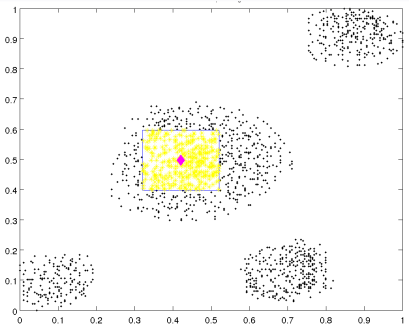


Steps 3 and 4

Step 3: Construct the window w of size α around the centre z

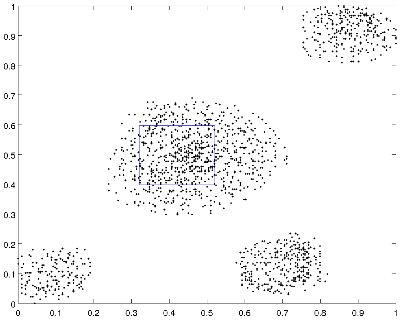


Step 4: Mark the points that lie in the window w as clustered

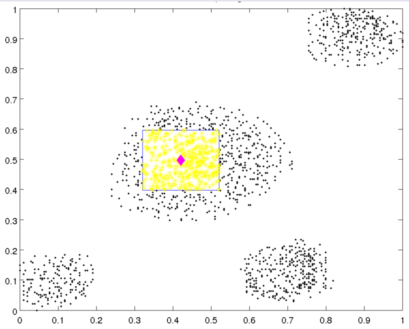


Steps 3 and 4

Step 3: Construct the window w of size α around the centre z

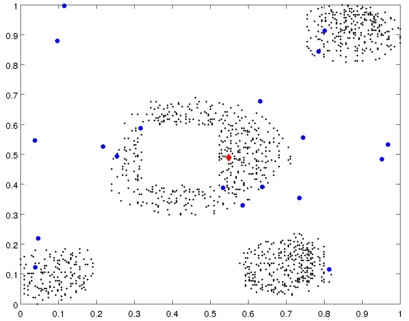


Step 4: Mark the points that lie in the window w as clustered

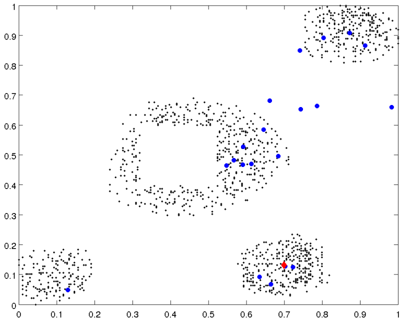


Repeat the previous Steps 1-4 for the remaining dataset

Step 2: 1^{st} Epoch

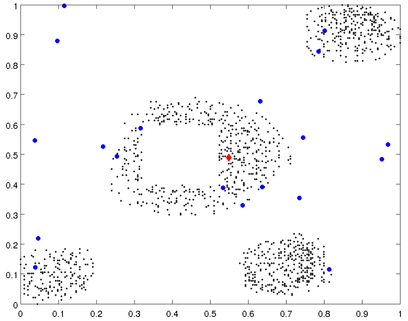


Step 2: 25^{th} Epoch

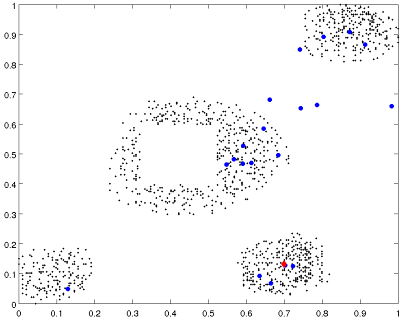


Repeat the previous Steps 1-4 for the remaining dataset

Step 2: 1^{st} Epoch

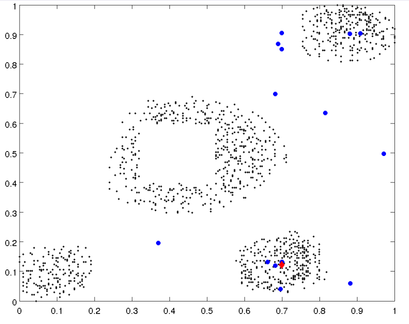


Step 2: 25^{th} Epoch

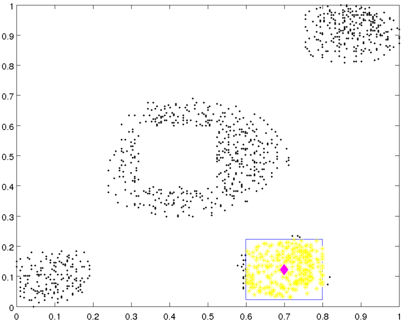


Steps 2, 3 and 4

Step 2: 100^{th} Epoch

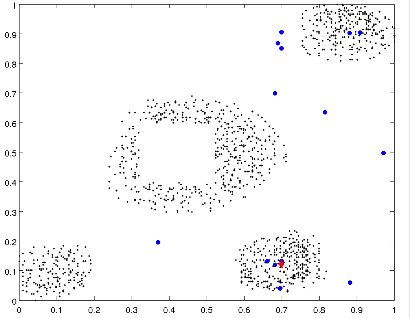


Steps 3 and 4

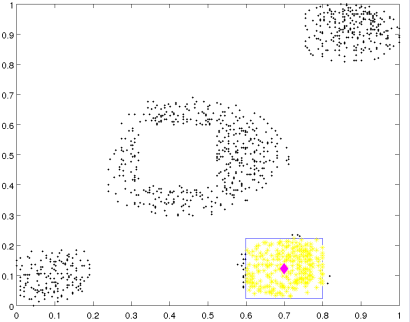


Steps 2, 3 and 4

Step 2: 100^{th} Epoch

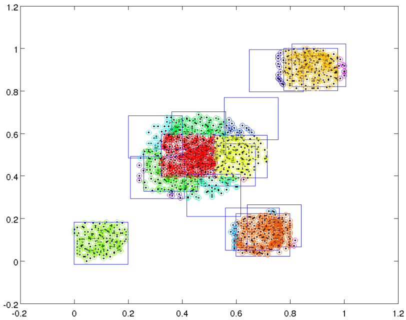


Steps 3 and 4

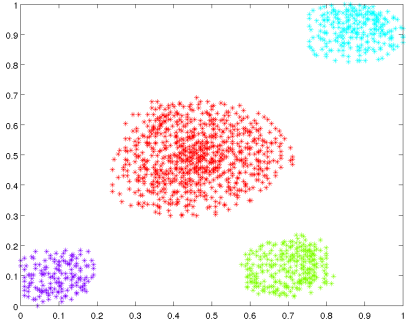


After 16^{th} iterations, the algorithm attains to detect the α–Clusterable Sets

Final α–Clusterable Sets

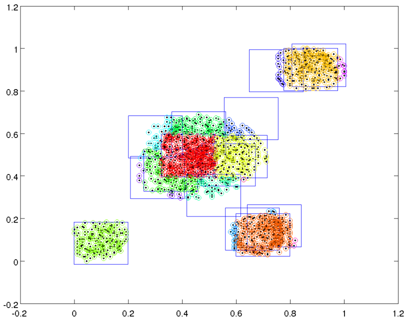


Merge the α–Clusterable Sets to form the final clusters

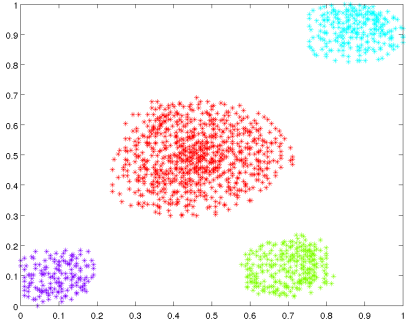


After 16^{th} iterations, the algorithm attains to detect the α–Clusterable Sets

Final α–Clusterable Sets



Merge the α–Clusterable Sets to form the final clusters



Goals of the Experiments

Three-fold objectives

- Investigate the effect of the parameter α
- Compare the performance of the proposed algorithm
- Investigate the scalability of the algorithm

Validity measures

1. Entropy: $H_i = -\sum_{j=1}^{m} P(x \in L_j | x \in C_i) \log P(x \in L_j | x \in C_i)$
   higher homogeneity means that entropy's values $\to 0$
2. Purity: $r = \frac{1}{n} \sum_{i=1}^{k} \alpha_i$
   α_i represents the number of patterns of the class to which the majority of points in cluster i belongs to it

Entropy and Purity vs Window Size α

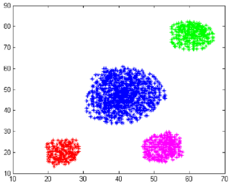


Figure: $2D$ Dataset of 1600 points ($Dset_1$)

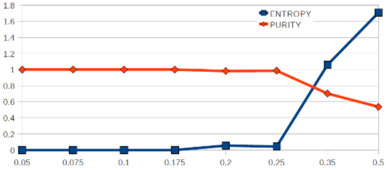


Figure: Entropy and Purity vs α

Conclusion

- The clustering quality is better for small values of the parameter α
- Values of α greater than 0.25 lead to the creation of clusters which contain data from different groups

Entropy and Purity vs Window Size α

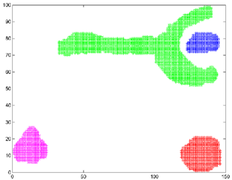


Figure: $2D$ Dataset of 2761 points ($Dset_2$)

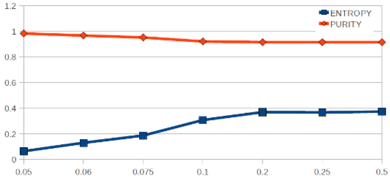


Figure: Entropy and Purity vs α

Conclusion

- The clustering quality is better for small values of the parameter α
- Values of α greater than 0.075 lead to the creation of clusters which contain data from different groups

Entropy and Purity vs Window Size α

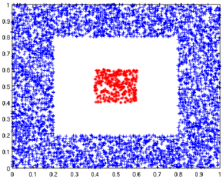


Figure: $2D$ Dataset of 5000 points ($Dset_3$)

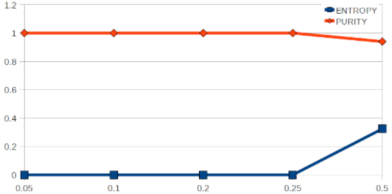


Figure: Entropy and Purity vs α

Conclusion

- The clustering quality is better for small values of the parameter α
- Values of α greater than 0.25 lead to the creation of clusters which contain data from different groups

Performance of PSO α-Cl algorithm *VS* other clustering algorithms

1. $Dset_1$: 1600 points, 2D, 4 spherical cluster
2. $Dset_2$: 2761 points, 2D, 4 arbitrary shape clusters
3. $Dset_4$: 15000 points, 3D, 6 clusters, randomly gen. by normal distribution with unary convex matrix
4. $Dset_5$: 15000 points, 5D, 8 clusters, randomly gen. by normal distribution with random parameters

	$Dset_1$	**$Dset_2$**	**$Dset_4$**	**$Dset_5$**
IUC	entropy purity	entropy purity	entropy purity	entropy purity
DEUC	entropy purity	entropy purity	entropy purity	entropy purity
k-means	entropy purity	entropy purity	entropy purity	entropy purity
k-windows	entropy purity	entropy purity	entropy purity	entropy purity
DBSCAN	entropy purity	entropy purity	entropy purity	entropy purity

Conclusion

Our algorithm exhibits better or similar performance versus other clustering algorithms in a majority of the experiments

Scalability of the “PSO α-Cl” algorithm

- 15.000 points generated by normal distribution
- 8 clusters with different cardinalities
- Dimensionality: $\{3, 5, 10\}$

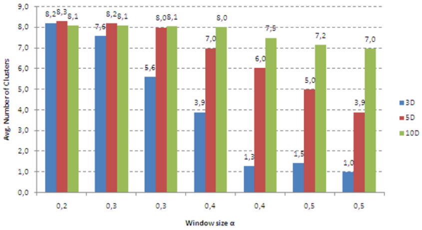


Figure: Average number of detected Clusters vs window size α

Conclusion

For small values of the parameter α, the proposed algorithm can detect the underlying clustering structure of the dataset

Scalability of the "PSO α-CI" algorithm

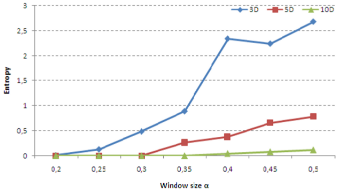


Figure: Entropy versus window size

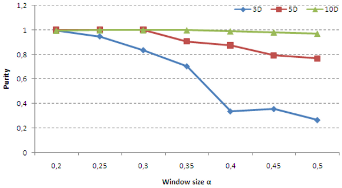


Figure: Purity versus window size

Conclusion

Good scalability properties since *Entropy* $\rightarrow 0$ and *Purity* $\rightarrow 1$ when the dimensionality of the datasets increased

Concluding Remarks - Future Work

In this first study...

- we proposed a theoretical framework for clustering, introducing a new notion, called α-Clusterable Set
- we proved that there exist an α-clustering function that satisfies the properties of scale-invariance, richness and consistency
- the proposed unsupervised clustering algorithm based on this framework, exhibits better or similar performance comparing with other clustering algorithms

More research have to be done...

- to enhance the theoretical framework
- to develop a self-adaptive algorithm, so as to evolve the value of parameter α during the clustering process

Thank you for your attention...

Any Questions ?

email: antzoulatos@upatras.gr