Clustering algorithms

Konstantinos Koutroumbas

Unit 6

- Fuzzy CFO clustering algortihms
- Possibilistic CFO clust. algorithms

koutroum@noa.gr

Fuzzy CFO clustering algorithms

Fuzzy clustering algorithms:

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ be a set of data points.

Each vector $\boldsymbol{x}_i$ **belongs** to all clusters up to a certain degree, $u_{ij}, j = 1, \dots, m$,
Subject to the constraints

- $u_{ij} \in [0,1],\ i = 1, \dots, N, j = 1, \dots, m$
- $\sum_{j=1}^{m} u_{ij} = 1\ ,\ i = 1, \dots, N$
- $0 < \sum_{i=1}^{N} u_{ij} < N,\ j = 1, \dots, m$

Each cluster is **represented** by a representative $\boldsymbol{\theta}_j$ (point repr., hyperplane…).
Let $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m\}$

Define the cost function

$$J_q(U, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j), \qquad (q > 1)$$

When $J_q(U, \Theta)$ is **minimized**?

When large u_{ij}'s are **multiplied** with small $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$'s.

Fuzzy CFO clustering algorithms

Minimizing the cost function

$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}{}^q\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \ldots, N$

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed for **both** of them.

For **fixed $\boldsymbol{\theta}_j$'s:** Define the Lagrangian function

$$\mathcal{L}_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}{}^q\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) - \sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{m} u_{ij} - 1\right)$$

Equating the partial derivative of $\mathcal{L}_q(U,\Theta)$ wrt u_{rs} to 0, it turns out that

$$\frac{\partial \mathcal{L}_q(U,\Theta)}{\partial u_{rs}} = 0 \Leftrightarrow u_{rs} = \frac{1}{\sum_{j=1}^{m}\left(\frac{d(\boldsymbol{x}_r, \boldsymbol{\theta}_s)}{d(\boldsymbol{x}_r, \boldsymbol{\theta}_j)}\right)^{\frac{1}{q-1}}}$$

For **fixed u_{ij}'s:** Solve the following m independent minimization problems

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

Fuzzy CFO clustering algorithms

Generalized Fuzzy Algorithmic Scheme (GFAS)

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \dots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of $u_{ij}'s$*
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j(t) = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^{q}(t-1) d(\boldsymbol{x}_i, \boldsymbol{\theta}_j), j = 1, \dots, m$$

  - End {For-j}
- **Until** a termination criterion is met.

Fuzzy CFO clustering algorithms

Remarks:

- A candidate termination condition is
$$||\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)|| < \varepsilon,$$
where $||\cdot||$ is any vector norm and ε a user-defined constant.

- GFAS may also be initialized from $U(0)$ instead of $\boldsymbol{\theta}_j(0), j = 1, \dots, m$ and start iterations with computing $\boldsymbol{\theta}_j$ first.

- If a point $\boldsymbol{x}_i$ **coincides** with one or more representatives, then it is shared arbitrarily among the clusters whose representatives coincide with $\boldsymbol{x}_i$, s.t. the constraint that the summation of all u_{ij}'s sum to 1.

- The degree of membership of $\boldsymbol{x}_i$ in C_j cluster is related to the grade of membership of $\boldsymbol{x}_i$ in rest $m-1$ clusters.

- If $q = 1$, **no** fuzzy clustering is better than the best hard clustering in terms of $J_q(\Theta, U)$.
- If $q > 1$**, there are** fuzzy clusterings with lower values of $J_q(\Theta, U)$ than the best hard clustering.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The point representatives case

- Point representatives are used in the case of compact clusters.
- Each $\boldsymbol{\theta}_j$ consists of l parameters.
- Any dissimilarity measure $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ between two points can be used.
- Common choices for $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ are

$$d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = (\boldsymbol{x}_i - \boldsymbol{\theta}_j)^T A(\boldsymbol{x}_i - \boldsymbol{\theta}_j),$$

where A is symmetric and positive definite matrix.

It is:

$$\frac{\partial d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = 2A(\boldsymbol{\theta}_j - \boldsymbol{x}_i)$$

In this case the problem

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

is solved as

$$\frac{\partial}{\partial \boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) = 0 \Leftrightarrow 2A \sum_{i=1}^{N} u_{ij}{}^q (\boldsymbol{\theta}_j - \boldsymbol{x}_i) = 0 \Leftrightarrow$$

$$\boldsymbol{\theta}_j = \frac{\sum_{i=1}^{N} u_{ij}{}^q \boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^q}$$

Fuzzy CFO clustering algorithms

GFAS – The point representative with squared Mahalanobis distance

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \dots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of $u_{ij}'s$*
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^{q}(t-1)\boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^{q}(t-1)}, j = 1, \dots, m$$

  - End {For-j}
- **Until** a termination criterion is met.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The point representatives case

Remarks:

- GFAS with the Euclidean distance ($A = I$) is also known as Fuzzy c-Means (FCM) or Fuzzy k-Means algorithm.

- FCM converges to a stationary point of the cost function or it has at least one subsequence that converges to a stationary point. This point may be a local (or global) minimum or a saddle point.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- Here the representatives are quadric surfaces (hyperellipsoids, hyperparaboloids, etc.)
- **First issue:** How to represent them?
- General forms of an equation describing a quadric surface Q:

1. $\boldsymbol{x}^T A\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c = 0$,

where A is an $l \times l$ symmetric matrix, $\boldsymbol{b}$ is an $l \times 1$ vector, c is a scalar and $\boldsymbol{x} = [x_1, \ldots, x_l]^T$.

For various choices of A, $\boldsymbol{b}$ and c we obtain hyperellipses, hyperparabolas and so on.

NOTE: The above representations of Q are equivalent.

2. $\boldsymbol{q}^T\boldsymbol{p} = 0$,

where

$$\boldsymbol{q} = [{x_1}^2, {x_2}^2, \ldots, {x_l}^2, x_1x_2, \ldots, x_{l-1}x_l, x_1, x_2, \ldots, x_l, 1]^T$$

and

$$\boldsymbol{p} = [p_1, p_2, \ldots, p_l, p_{l+1}, \ldots, p_r, p_{r+1}, \ldots, p_s]^T$$

with $r = \frac{l(l+1)}{2}$ and $s = r + l + 1$.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Second issue:** *"Definition of the distance of a point* $\boldsymbol{x}$ *to a quadric surface* Q*"*

Types of distances

– Perpendicular distance:

$$d_p{}^2(\boldsymbol{x}, Q) = min_{\boldsymbol{z}} \|\boldsymbol{x} - \boldsymbol{z}\|^2,$$

subject to the constraint

$$\boldsymbol{z}^T A \boldsymbol{z} + \boldsymbol{b}^T \boldsymbol{z} + c = 0$$

In words, $d_p{}^2(\boldsymbol{x}, Q)$ is the distance between $\boldsymbol{x}$ and the closest to $\boldsymbol{x}$ point that lies in Q.

Prove it for the $l = 2$ case.

– (Squared) Algebraic distance:

$$d_p{}^2(\boldsymbol{x}, Q) = (\boldsymbol{x}^T A \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c)^2 \equiv \boldsymbol{p}^T M \boldsymbol{p}$$

where $M = \boldsymbol{q}\boldsymbol{q}^T$.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

– Radial distance (**only** when Q is a hyperellipsoidal):

For Q hyperellipsoidal, the representative equation can be written as

$$(\boldsymbol{x}-\boldsymbol{c})^T A(\boldsymbol{x}-\boldsymbol{c}) = 1$$

where $\boldsymbol{c}$ is the center of the ellipse and A a **positive definite symmetric** matrix defining major axis, minor axis and orientation.

Then the radial distance is defined as

$$d_r{}^2(\boldsymbol{x}, Q) = \|\boldsymbol{x}-\boldsymbol{z}\|^2$$

subject to the constraints

$$(\boldsymbol{z}-\boldsymbol{c})^T A(\boldsymbol{z}-\boldsymbol{c}) = 1$$

and

$$(\boldsymbol{z}-\boldsymbol{c}) = a(\boldsymbol{x}-\boldsymbol{c}).$$

In words,

–the intersection point $\boldsymbol{z}$ between the line segment $\boldsymbol{x}-\boldsymbol{c}$ and Q is determined

–the $d_r{}^2(\boldsymbol{x}, Q)$ is defined as the squared Euclidean distance between $\boldsymbol{x}$ and $\boldsymbol{z}$.

Fuzzy Clustering – The quadric surfaces representatives case

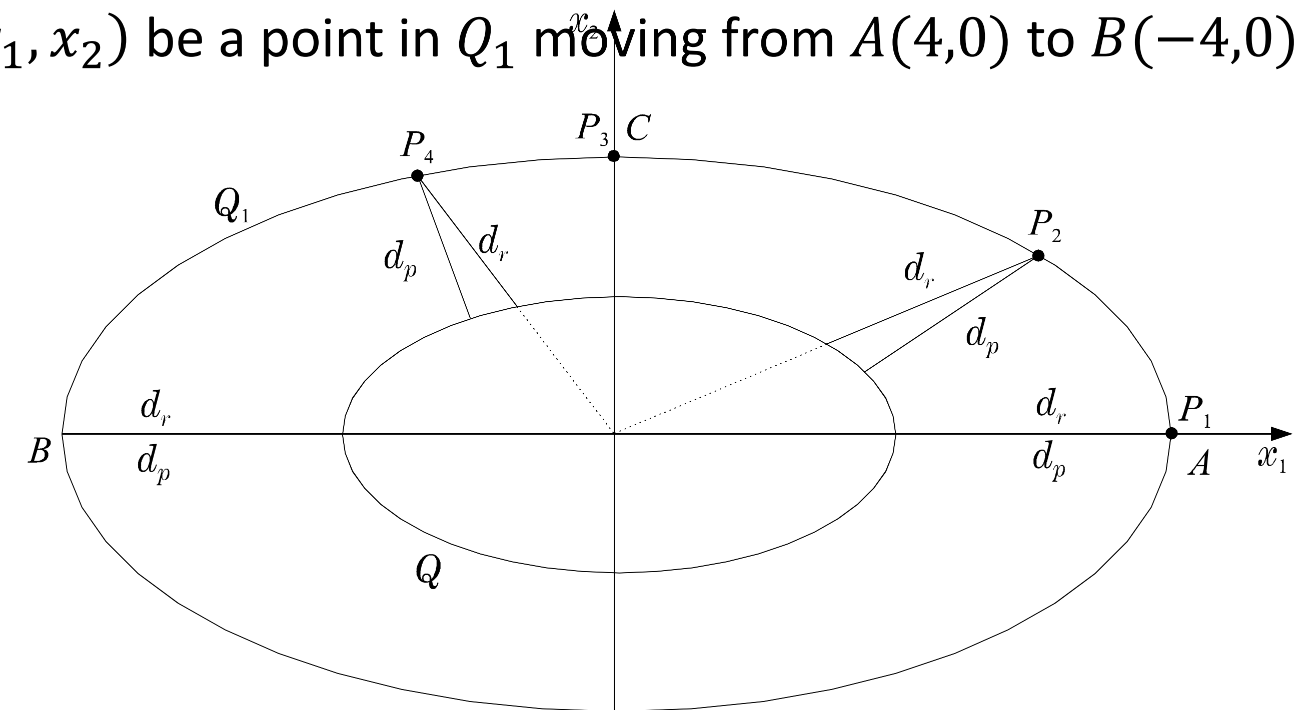
– (Squared) Normalized radial distance (only when Q is a hyperellipsoidal):

$$d_{nr}{}^2(\boldsymbol{x}, Q) = \left(\left((\boldsymbol{x}-\boldsymbol{c})^T A(\boldsymbol{x}-\boldsymbol{c})\right)^{1/2} - 1\right)^2$$

$d_r{}^2(\boldsymbol{x}, Q) = d_{nr}{}^2(\boldsymbol{x}, Q)\|\boldsymbol{x}-\boldsymbol{z}\|^2$
$\boldsymbol{z}$: intersection of $\boldsymbol{x}-\boldsymbol{c}$ and Q.

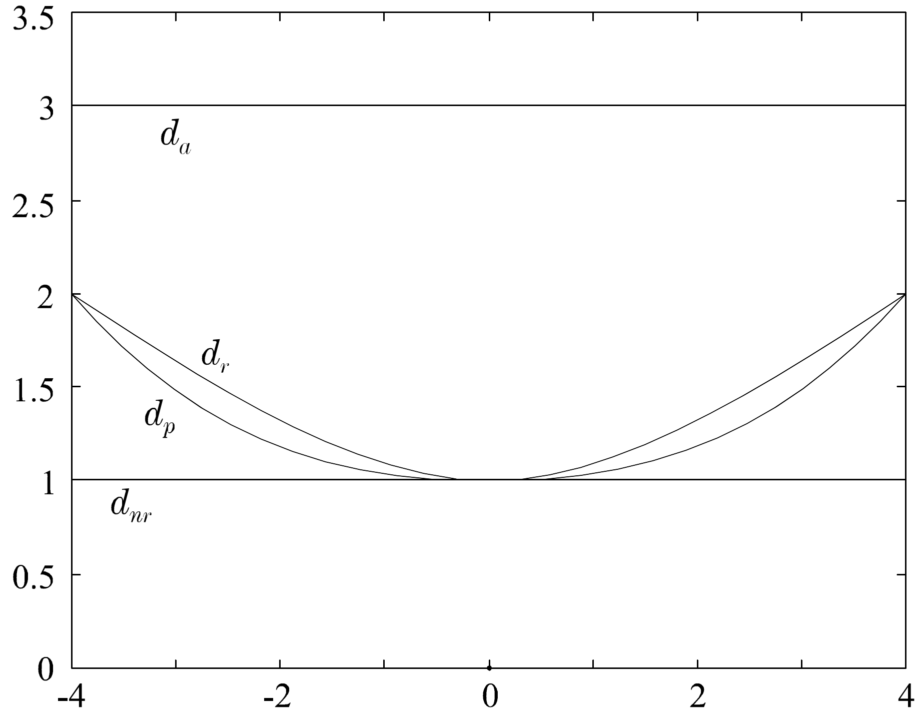
– Example 3:

- Consider two ellipses Q and Q_1, centered at $\boldsymbol{c} = [0, 0]^T$, with $A = diag(\frac{1}{4}, 1)$ and $A_1 = diag\left(\frac{1}{16}, \frac{1}{4}\right)$, respectively.
- Let $P(x_1, x_2)$ be a point in Q_1 moving from $A(4,0)$ to $B(-4,0)$, with $x_2 > 0$



Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case



Remarks:

- d_a and d_{nr} **do not vary** as P moves.
- d_r can be used as an **approximation** of d_p, when Q is a hyperellipsoid.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

Recall that

$$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \text{ s.t. } \sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$$

- The algorithms in this case fall under the umbrella of GFAS.
- They all **share** the same rule for updating the matrix U.
- They **differ** on the choice of the distance between a point and the representative of a quadric surface.

  $\Rightarrow$ they **differ** in the representatives updating part.
- *At each iteration*, the updating of the representatives is carried out by **setting** the gradient of J_q wrt them equal to $\mathbf{0}$ (for **fixed** $\boldsymbol{u_{ij}}$'s) and **solving** (usually using *iterative schemes*) for the involved parameters.

Fuzzy CFO clustering algorithms

Generalized Fuzzy Algorithmic Scheme (GFAS)

- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \dots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of $u'_{ij}s$*
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{d(\boldsymbol{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j), j = 1, \dots, m$$

  - End {For-j}
- **Until** a termination criterion is met.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}{}^q\, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$

Algorithms:

- *Fuzzy C Ellipsoidal Shells* (FCES) Algorithm:

- It **adopts** the radial distance between a vector and the surface representative

- It **recovers** only ellipsoidal clusters.

- *Fuzzy C Quadric Shells* (FCQS) Algorithm:

- It **adopts** the algebraic distance between a vector and the surf. repr. in the form $d_a{}^2(\boldsymbol{x}, Q) = \boldsymbol{p}^T M \boldsymbol{p}$, **imposing** constraints on vector $\boldsymbol{p}$.

- It **recovers** quadric clusters of any kind (ellipsoidal, hyperbolical, paraboloidal, pairs of lines).

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^{q}\, d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \ldots, N$

Algorithms:

- *Modified Fuzzy C Quadric Shells* (MFCQS) Algorithm:

- It **adopts** :
  - the perpendicular distance between a vector and the surface representative for the updating of matrix U
  - The algebraic distance between a vector and the surface representative for the updating of the cluster representatives.
- It **recovers** quadric clusters of any kind (ellipsoidal, hyperbolical, paraboloidal, pairs of lines).

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The hyperplane surfaces representatives case

- Here the representatives are hyperplanes (lines in the 2-D space, planes in the 3-D space etc.)
- **First issue:** How to represent them?

1. Via the equation of a hyperplane H:

$$H\colon \boldsymbol{\theta}^T \boldsymbol{x} + \theta_0 = 0,$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_l]^{\boldsymbol{T}}$, $\boldsymbol{x} = [x_1, x_2, \dots, x_l]^{\boldsymbol{T}}$.

2. Via a center $\boldsymbol{c}_j$ and a covariance matrix Σ_j, that is, $\boldsymbol{\theta}_j = (\boldsymbol{c}_j, \Sigma_j)$.

NOTE: Another choice for **representing** such clusters is by using line segments. (only for the *2-D case*).

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The hyperplane surfaces representatives case

- **Second issue**: *"Definition of the distance of a point $\boldsymbol{x}$ to a cluster"*

Types of distances

– Distance of a point from a hyperplane:

$$d(\boldsymbol{x}, H) = \frac{|\boldsymbol{\theta}^T \boldsymbol{x} + \theta_0|}{||\boldsymbol{\theta}||}$$

– GK distance:

$$d_{GK}{}^2(\boldsymbol{x}, \boldsymbol{\theta}_j) = \left|\Sigma_j\right|^{1/l} \left(\boldsymbol{x} - \boldsymbol{c}_j\right)^T \Sigma_j{}^{-1} \left(\boldsymbol{x} - \boldsymbol{c}_j\right)$$

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The hyperplane surfaces representatives case

- **Third issue:** Choice of algorithm.

Recall that

$$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^{q}\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \text{ s.t. } \sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$$

- The algorithms in this case fall under the umbrella of GFAS.
- They all **share** the same rule for updating the matrix U.
- They **differ** on the choice of the distance between a point and the representative of a plane cluster.

  $\Rightarrow$ they **differ** in the representatives updating part.
- *At each iteration*, the updating of the representatives is carried out by **setting** the gradient of J_q wrt them equal to 0 (for **fixed** $\boldsymbol{u_{ij}}$**'s**) and **solving** (usually using *iterative schemes*) for the involved parameters.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$$J_q(U, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^{q}\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \text{ s.t. } \sum_{j=1}^{m} u_{ij} = 1, i = 1, \dots, N$$

Algorithms:

- *Fuzzy C varieties* (FCV) Algorithm:

- It **adopts** the classical distance between a point and a hyperplane.

- *Disadvantages*:
    - It tends to **recover** very long clusters and, thus, collinear distinct clusters may be **merged** to a single one.
    - If, at a certain iteration, a hyperplane representative crosses two distinct clusters, there is no way to recover from this situation.

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- **Third issue:** Choice of algorithm.

$J_q(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ s.t. $\sum_{j=1}^{m} u_{ij} = 1, i = 1, \ldots, N$

Algorithms:

- *Gustafson-Kessel* (GK) algorithm:

- It **adopts** the GK distance between a point and a cluster.
- The parameter updating takes place via the following two equations

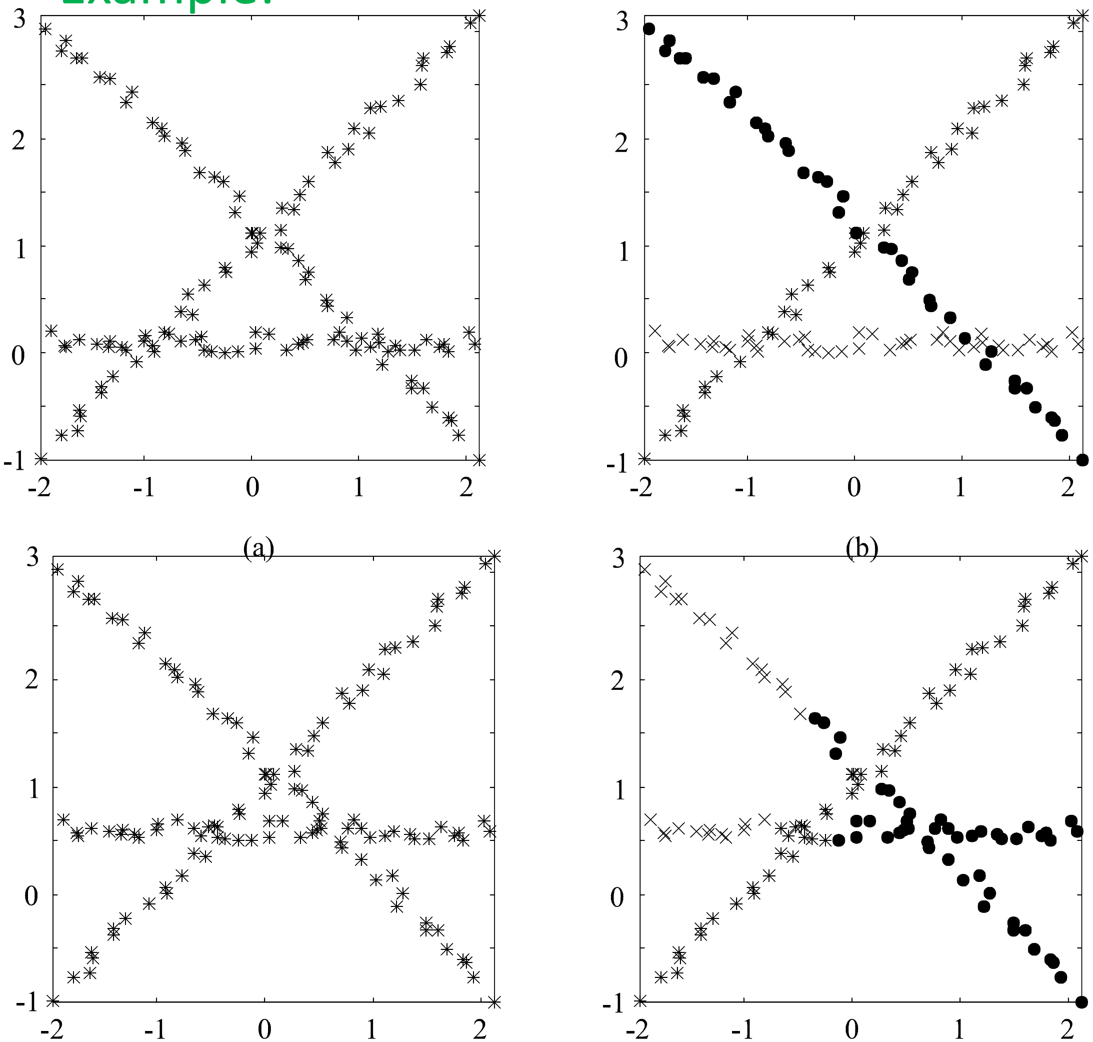
$$\boldsymbol{c}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^q(t-1)\,\boldsymbol{x}_i}{\sum_{i=1}^{N} u_{ij}{}^q(t-1)}$$

$$\Sigma_j(t) = \frac{\sum_{i=1}^{N} u_{ij}{}^q(t-1)(\boldsymbol{x}_i - \boldsymbol{c}_j(t))(\boldsymbol{x}_i - \boldsymbol{c}_j(t))^T}{\sum_{i=1}^{N} u_{ij}{}^q(t-1)}$$

Fuzzy CFO clustering algorithms

Fuzzy Clustering – The quadric surfaces representatives case

- *Gustafson-Kessel* (GK) algorithm (cont.):
- Example:



(a) (b)

(a) (b)

Comments:

In the **first case**, the clusters are well discriminated and the GK-algorithm **recovers** them correctly.

In the **second case**, the clusters are not well discriminated and the GK-algorithm **fails** to recover them correctly.

Possibilistic CFO clustering algorithms

Possibilistic clustering algorithms:

Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ be a set of data points.

For each vector $\boldsymbol{x}_i$ its degree of compatibility with all clusters, $u_{ij}, j = 1, \dots, m$, is considered.

The constraints on u_{ij}'s are

- $u_{ij} \in [0,1],\ i = 1, \dots, N, j = 1, \dots, m$
- $0 < \sum_{i=1}^{N} u_{ij} < N,\ j = 1, \dots, m$

Each cluster is **represented** by a representative $\boldsymbol{\theta}_j$ (point repr., hyperplane…).

Let $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m\}$

Define the cost function

$$J_q(U, \Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} {u_{ij}}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

When $J_q(U, \Theta)$ is **minimized**?

When all u_{ij}'s are (very close to) zero.

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

A possible scenario:

Minimize the cost function

$$J_q(U,\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{m} {u_{ij}}^q \, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \sum_{j=1}^{m} \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q$$

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed $\boldsymbol{\theta}_j$'s:** Equating the partial derivative of $J_q(U,\Theta)$ wrt u_{ij} to 0 we obtain

$$\frac{\partial J_q(U,\Theta)}{\partial u_{ij}} = 0 \Leftrightarrow u_{ij} = \frac{1}{1 + \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\eta_j}\right)^{\frac{1}{q-1}}}$$

Notes: (a) u_{ij} **depends exclusively** on $\boldsymbol{\theta}_j$.

(b) It is $u_{ij} \in [0,1]$

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

A possible scenario:

Minimize the cost function

$$J_q(U,\Theta)=\sum_{i=1}^{N}\sum_{j=1}^{m}{u_{ij}}^{q}\,d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)+\sum_{j=1}^{m}\eta_j\sum_{i=1}^{N}(1-u_{ij})^{q}$$

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed u_{ij}'s:** Solve the following m independent minimization problems

$$\boldsymbol{\theta}_j=argmin_{\boldsymbol{\theta}_j}\sum_{i=1}^{N}{u_{ij}}^{q}d(\boldsymbol{x}_i,\boldsymbol{\theta}_j)$$

Possibilistic CFO clustering algorithms

<u>*Generalized Possibilistic Algorithmic Scheme (GPAS1)*</u>

- **Fix** η_j**'s,** $j = 1, \ldots, m$.
- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \ldots, m$.
- $t = 0$
- **Repeat**
  - For $i = 1$ to N *% Determination of $u'_{ij}s$*
    - o For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{1 + \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{\eta_j}\right)^{\frac{1}{q-1}}}$$

    - o End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m *% Parameter updating*
    - o Set

$$\boldsymbol{\theta}_j(t) = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}{}^q(t-1) d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right), j = 1, \ldots, m$$

  - End {For-j}
- **Until** a termination criterion is met.

Possibilistic CFO clustering algorithms

Remarks:

- A candidate termination condition is

$$||\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)|| < \varepsilon,$$

  where $||\cdot||$ is any vector norm and ε a user-defined constant.

- GFAS may also be initialized from $U(0)$ instead of $\boldsymbol{\theta}_j(0),\ j = 1, \dots, m$ and start iterations with computing $\boldsymbol{\theta}_j$ first.

- Based on GPAS, a possibilistic algorithm can be derived, for each fuzzy clustering algorithm derived previously.

- High values of q:
  - In possibilistic clustering **cause** almost equal contributions of all vectors to all clusters
  - In fuzzy clustering cause increased sharing of the vectors among all clusters.

Possibilistic CFO clustering algorithms

Three observations

- Decomposition of $J(\Theta, U)$:

Since for each vector $\boldsymbol{x}_i$, u_{ij}'s, $j = 1, \ldots, m$ are independent from each other, $J(\Theta, U)$ can be written as

$$J(\Theta, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^{q}\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \sum_{j=1}^{m} \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q$$

$$= \sum_{j=1}^{m} \left[\sum_{i=1}^{N} u_{ij}{}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q \right] \equiv \sum_{j=1}^{m} J_j$$

where

$$J_j = \sum_{i=1}^{N} u_{ij}{}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \eta_j \sum_{i=1}^{N} (1 - u_{ij})^q$$

Each J_j is **associated** with a different cluster and *minimization of $J(\Theta, U)$ with respect to u_{ij}'s can be carried out separately for each J_j*.

Possibilistic CFO clustering algorithms

<u>*Three observations*</u>

- About η_j's:

–They **determine** the relative significance of the two terms in $J(\Theta, U)$.

–They are **related** to the "variance" of the points of C_j's, $j = 1, \dots, m$, around their centers.

–Two scenarios for the estimation of η_j's, for the point representatives case, are the following:

o **Run** the related FCM algorithm and after its convergence estimate η_j's as

$$\eta_j = \frac{\sum_{i=1}^{N} u_{ij}{}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\sum_{i=1}^{N} u_{ij}{}^q} \quad \text{or} \quad \eta_j = \frac{\sum_{u_{ij}>a} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\sum_{u_{ij}>a} 1}$$

○ **Set** $\eta_j = \eta = \frac{\beta}{q\sqrt{m}}$, where $\beta = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \overline{\boldsymbol{x}}\|^2$ and $\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$

Possibilistic CFO clustering algorithms

Three observations

- The mode-seeking property

Unlike Hard and fuzzy clustering algorithms which are partition algorithms (they terminate with the predetermined number of clusters no matter how many physical clusters are naturally formed in X), GPAS is a mode-seeking algorithm (it searches for dense regions of vectors in X).

Advantage: The number of clusters need not be a priori known.
If the number of clusters in GPAS, m, is greater than the true number of clusters k in X, some representatives will coincide with others. If $m < k$, ***some*** (and not all) of the clusters will be identified.

Disadvantage: Need for estimating η_j.

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

Another possible scenario:

Minimize the cost function

$$J(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \sum_{j=1}^{m} \eta_j \sum_{i=1}^{N} (u_{ij} \ln u_{ij} - u_{ij})$$

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed $\boldsymbol{\theta}_j$'s:** Equating the partial derivative of $J(U,\Theta)$ wrt u_{ij} to 0 we obtain

$$\frac{\partial J_q(U,\Theta)}{\partial u_{ij}} = 0 \Leftrightarrow u_{ij} = exp\left(-\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)}{\eta_j}\right)$$

Notes: (a) u_{ij} **depends exclusively** on $\boldsymbol{\theta}_j$.

(b) It is $u_{ij} \in [0,1]$

Possibilistic CFO clustering algorithms

How to avoid the trivial zero u_{ij}'s solution?

Add a suitable term that discourages the zero solution.

A possible scenario:

Minimize the cost function

$$J(U,\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}\, d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) + \sum_{j=1}^{m} \eta_j \sum_{i=1}^{N} (u_{ij} \ln u_{ij} - u_{ij})$$

where η_j's are suitably defined constants (one for each cluster), **associated** with the variance of the clusters.

Since $\boldsymbol{\theta}_j$'s, u_{ij}'s are continuous valued, tools from analysis may be employed.

For **fixed u_{ij}'s:** Solve the following m independent minimization problems

$$\boldsymbol{\theta}_j = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

Possibilistic CFO clustering algorithms

Generalized Possibilistic Algorithmic Scheme (GPAS2)

- **Fix** η_j**'s,** $j = 1, \ldots, m$.
- **Choose** $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$, $j = 1, \ldots, m$.
- $t = 0$
- **Repeat**

  – For $i = 1$ to N *% Determination of $u_{ij}'s$*

  o For $j = 1$ to m

$$u_{ij}(t) = exp\left(-\frac{d(\boldsymbol{x}_i, \boldsymbol{\theta}_j(t))}{\eta_j}\right)$$

  o End {For-j}

  – End {For-i}

  $-t = t + 1$

  –For $j = 1$ to m *% Parameter updating*

  o Set

$$\boldsymbol{\theta}_j(t) = argmin_{\boldsymbol{\theta}_j} \sum_{i=1}^{N} u_{ij}(t-1) d\left(\boldsymbol{x}_i, \boldsymbol{\theta}_j\right), j = 1, \ldots, m$$

  – End {For-j}

- **Until** a termination criterion is met.

CFO clustering algorithms: A unified view

Data

$$X = \{\boldsymbol{x}_j \in R^l, j = 1, \ldots, N\}$$

Recall the general CFO framework

Basic parameters - notation

✓ $\Theta = \{\boldsymbol{\theta}_j, j = 1, \ldots, m\}$ ($\boldsymbol{\theta}_j$ is the representative of cluster C_j).

- **Proximity** between $\boldsymbol{x}_i$ and C_j: $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$

Basic parameters – notation (cont.)

In the probabilistic case u_{ij} stands for $P(j|\boldsymbol{x}_i)$

Recall the general CFO framework

✓ $$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N1} & u_{N2} & \cdots & u_{Nm} \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_N \end{bmatrix}$$

- $u_{ij} \in [0,1]$ quantifies the "relation" between $\boldsymbol{x}_i$ and C_j.
- "Large" ("small") u_{ij} values indicate close (loose) relation between $\boldsymbol{x}_i$ and C_j.

$\Rightarrow$ u_{ij} varies **inversely proportional** wrt $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

- $\boldsymbol{u}_i$: vector containing the u_{ij}'s of $\boldsymbol{x}_i$ with all clusters.

(*) Unless otherwise stated, the case where **cluster representatives** are used is considered.

CFO clustering algorithms: A unified view

Aim:

✓ To place the representatives into dense in data regions (physical clusters).

Recall the general CFO framework

How this is achieved:

✓ Via the minimization of the following type of cost function (wrt Θ, U)

$$J(\Theta, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}{}^{q} d(\boldsymbol{x}_i, \boldsymbol{\theta}_j) \ (q \geq 1)$$

s.t. some **constraints** on U, $C(U)$.

For the probabilistic case $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ is embedded in the log-likelihood of suitably defined exponential distributions

Intuition:

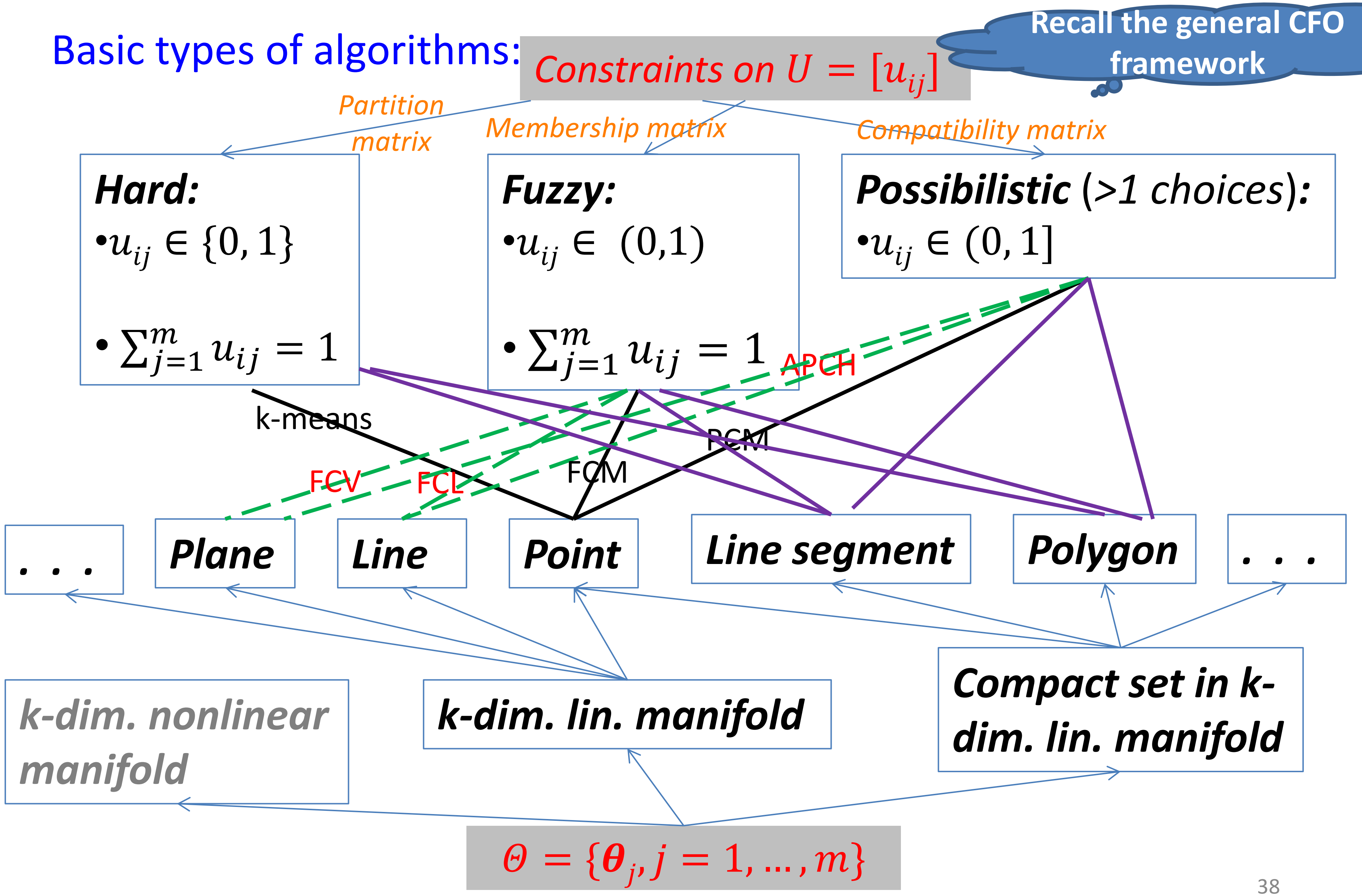
✓ For **fixed** $\boldsymbol{\theta}_j$'s, $J(\Theta, U)$ is a weighted sum of **fixed** distances $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

⇒ **Minimization** of $J(\Theta, U)$ wrt u_{ij} instructs for large weights (u_{ij}) for small distances $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$.

✓ For **fixed** u_{ij}'s, **minimization** of $J(\Theta, U)$ wrt $\boldsymbol{\theta}_j$'s leads $\boldsymbol{\theta}_j$'s closer to their most relative data points.

CFO clustering algorithms: A unified view

Basic types of algorithms:



CFO clustering algorithms: A unified view

"Array of CFO algorithms"

algorithm

Recall the general CFO framework

$C(U)$

$\boldsymbol{\theta}_j$

	Hard Constr.	Fuzzy Constr.	Possi Constr.	
Point				
Line				
Hyperplane				
Hyperellipsoid				
. . .				

CFO scheme

There are **several** unexplored areas (groups of algorithms) in this array.

CFO clustering algorithms: A unified view

Recall the general CFO framework

General cost function opt. (**CFO**) scheme:

✓ *Initialize* $\Theta = \Theta(0)$

✓ $t = 0$

✓ **Repeat**

- $U(t) = argmin_U \ J(\Theta(t), U)$, s.t. $C(U(t))$
- $t = t + 1$
- $\Theta(t) = argmin_\Theta \ J(\Theta, U(t-1))$

✓ **Until convergence**

CFO clustering algorithms: A unified view

“Array of CFO algorithms”

Recall the general CFO framework

$C(U)$

$\boldsymbol{\theta}_j$

	Hard Constr.	Fuzzy Constr.	Possib. Constr.	. . .
Point				
Line				
Hyperplane				
Hyperellipsoid				
. . .				

Hard CFO scheme

Fuzzy CFO scheme

Possib. CFO scheme

CFO scheme

CFO clustering algorithms: A unified view

"Array of CFO algorithms"

Recall the general CFO framework

$C(U)$

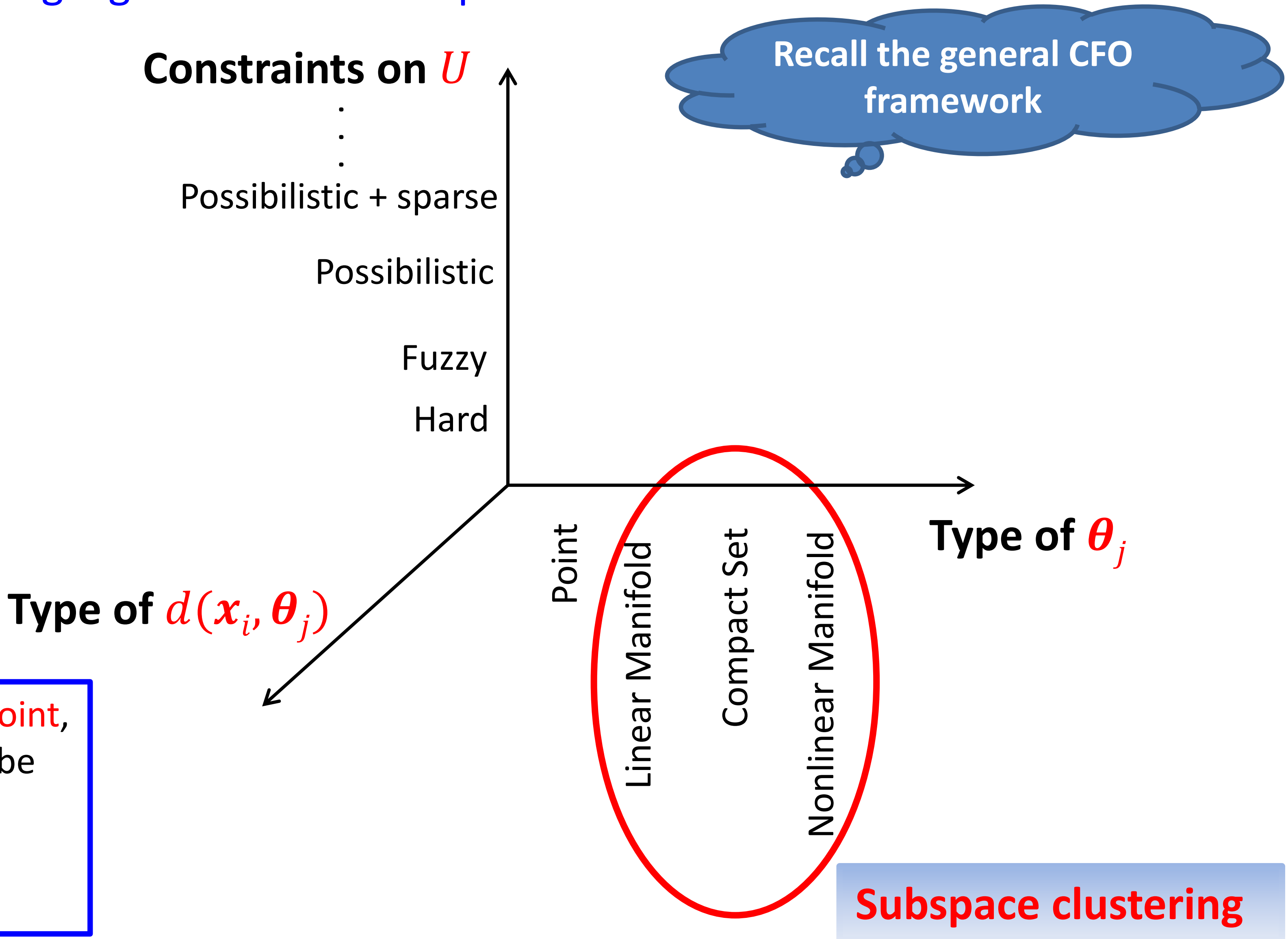
$\boldsymbol{\theta}_j$

	Hard Constr.	Fuzzy Constr.	Possib. Constr.	. . .
Point	c-means scheme			
Line	c-lines scheme			
Hyperplane	c-hyperplanes scheme			
Hyperellipsoid	c-hyperellipsoids scheme			
. . .				

CFO scheme

CFO clustering algorithms: A unified view

CFO clustering algorithms: A loose presentation



E.g.:If $\boldsymbol{\theta}_j$ is a point, $d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$ may be
- Sq. Euclidean
- l_p norm
- Mahalanobis