A family of adaptive fuzzy clustering models

László Szilágyi

Introduction

- There exist evergreen mathematical problems in AI as well
- I was in the kindergarten when we studied the new algorithm of Bezdek, the fuzzy c-means clustering
- I decided that I would try to find the best c-means clustering model
- Goals
  - Create the c-means clustering model that produces fine partitions on any data and any possibles circumstances. The fewer exceptions the better.
- Limitations
  - Stay within the bounds of alternating optimization of a quadratic objective function
  - Using zero gradient conditions and Lagrange multipliers only, no numerical approximations

Where are we now?

- The PFCM algorithm is one of the best existing models
- Uses a mixed partition, linear combination of probabilistic and possibilistic ones
- The possibilistic partition strongly depends on the constant penalty terms established at initialization
- Proposing a modification that would allow us to update the penalty terms in every optimization cycle
- This way our model could adapt the penalty terms to the data and circumstances
- Expecting better performances

$$\mathbb{N}_a = \{1, 2, \ldots, a\}$$

Fuzzy c-means (FCM)

- Bezdek (1981)
- Objective function

$$J_{\mathrm{FCM}} = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m}\|\mathbf{x}_k - \mathbf{v}_i\|^2 = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} d_{ik}^2$$

- Probabilistic constraints

$$\begin{cases} 0 \le u_{ik} \le 1, & \forall i \in \mathbb{N}_c, \forall k \in \mathbb{N}_n \\ \sum_{i=1}^{c} u_{ik} = 1, & \forall k \in \mathbb{N}_n \end{cases}$$

- Alternating optimization
- Partition update formula

$$u_{ik} = \frac{d_{ik}^{-2/(m-1)}}{\sum_{j=1}^{c} d_{jk}^{-2/(m-1)}} \qquad \begin{matrix} \forall i \in \mathbb{N}_c \\ \forall k \in \mathbb{N}_n \end{matrix}$$

- Cluster prototype update formula

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^{m}\mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^{m}} \qquad \forall i \in \mathbb{N}_c$$

FCM evaluation

- Simple, stable, very popular
- Sensitive to noise
  - A true outlier belongs to each class with 1/c probability, which can damage the partition

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^{m}} \qquad \forall i \in \mathbb{N}_c$$

- Reason:
  - The probabilistic constraint is too strong
  - Needs to be relaxed

Fuzzy (c+1)-means

- Davé (1991)
- Objective function

$$d_{0k} = d_0 \qquad \forall k \in \mathbb{N}_n$$

$$J_{\mathrm{F(C+1)M}} = \sum_{k=1}^{n} \left(u_{0k}^{m} \cdot d_0^2 + \sum_{i=1}^{c} u_{ik}^{m} \cdot d_{ik}^2 \right)$$

- Constraints

$$\begin{cases} 0 \le u_{ik} \le 1, & \forall i \in \mathbb{N}_c^0, \forall k \in \mathbb{N}_n \\ \sum_{i=0}^{c} u_{ik} = 1, & \forall k \in \mathbb{N}_n \end{cases}$$

- Alternating optimization
- Partition update formula

$$u_{ik} = \frac{d_{ik}^{-2/(m-1)}}{\sum_{j=0}^{c} d_{jk}^{-2/(m-1)}} \qquad \begin{matrix} \forall i \in \mathbb{N}_c^0 \\ \forall k \in \mathbb{N}_n \end{matrix}$$

- Cluster prototype update formula

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^{m}} \qquad \forall i \in \mathbb{N}_c$$

F(C+1)M evaluation

- Suppresses the noise
- Not very popular as standalone clustering model because it cannot handle clusters of different width

Possibilistic c-means

- Krishnapuram & Keller (1993)
- Objective function

$$J_{\mathrm{PCM}} = \sum_{i=1}^{c} \sum_{k=1}^{n} [t_{ik}^{p} \cdot d_{ik}^{2} + (1 - t_{ik})^{p} \eta_i]$$

- Constraints

$$\begin{cases} 0 \le t_{ik} \le 1, & \forall i \in \mathbb{N}_c, \forall k \in \mathbb{N}_n \\ 0 < \sum_{i=1}^{c} t_{ik} < c, & k \in \mathbb{N}_n \end{cases}$$

- Alternating optimization
- Partition update formula

$$t_{ik} = \left[1 + \left(\frac{d_{ik}^{2}}{\eta_i} \right)^{1/(p-1)} \right]^{-1} \qquad \begin{matrix} \forall i \in \mathbb{N}_c \\ \forall k \in \mathbb{N}_n \end{matrix}$$

- Cluster prototype update formula

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} t_{ik}^{p} \mathbf{x}_k}{\sum_{k=1}^{n} t_{ik}^{p}} \qquad \forall i \in \mathbb{N}_c$$

Recipe for the possibilistic penalty terms

- They are CONSTANT through the optimization of PCM
- Krishnapuram & Keller (1996)
- Run a few iterations of FCM, and use its outcome to set the penalty terms

$$\eta_i = \kappa \cdot \eta_{i0} = \kappa \cdot \frac{\sum_{k=1}^{n} u_{ik}^m d_{ik}^2}{\sum_{k=1}^{n} u_{ik}^m} \qquad \forall i \in \mathbb{N}_c$$

- The constant κ is recommended to be chosen between 1 and 2

PCM evaluation

- It can handle clusters of different size (width or radius)
- Not sensitive to noise
  - Successfully relaxed the probabilistic constraint
- Problem: frequently produces coincident clusters

- Reason:
  - The possibilistic constraint is too week
  - Clusters d do not depend on each other

$$u_{ik} = \frac{d_{ik}^{-2/(m-1)}}{\sum_{j=1}^{c} d_{jk}^{-2/(m-1)}} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

$$t_{ik} = \left[1 + \left(\frac{d_{ik}^2}{\eta_i}\right)^{1/(p-1)}\right]^{-1} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

Mixed clustering models

- Fuzzy-possibilistic c-means (Pal et al 1997) – not scalable
  - We fixed it in (Naghi, Szilágyi et al 2022)
- Possibilistic-fuzzy c-means (Pal et al 2005) – successful algorithm
  - Self-tuning version introduced in (Naghi, Szilágyi et al FUZZ-IEEE 2023), extended version presented here
- Fuzzy-possibilistic product partition c-means (Szilágyi 2011) – totally rejects outliers
  - Self-tuning version introduced in (Naghi, Szilágyi et al SMC 2024), extended version under construction

Possibilistic-fuzzy c-means (PFCM)

- Pal & Pal & Keller & Bezdek (2005)
- Objective function

$$J_{\mathrm{PFCM}} = \sum_{i=1}^{c}\sum_{k=1}^{n}[(au_{ik}^{m} + bt_{ik}^{p})d_{ik}^{2} + (1 - t_{ik})^{p}\eta_i]$$

- Constraints: probabilistic from FCM, possibilistic from PCM
  - a>0, b>0, m>1, p>1, κ =1

$$u_{ik} = \frac{d_{ik}^{-2/(m-1)}}{\sum_{j=0}^{c} d_{jk}^{-2/(m-1)}} \qquad \forall i \in \mathbb{N}_c^0 \quad \forall k \in \mathbb{N}_n$$

- Partition update formula

$$t_{ik} = \left[1 + \left(\frac{bd_{ik}^{2}}{\eta_i}\right)^{1/(p-1)}\right]^{-1} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

- Cluster prototype update formula

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n}(au_{ik}^{m} + bt_{ik}^{p})\mathbf{x}_k}{\sum_{k=1}^{n}(au_{ik}^{m} + bt_{ik}^{p})} \qquad \forall i \in \mathbb{N}_c$$

PFCM evaluation

- It can handle clusters of different size
- FCM and PCM components compensate each other's drawback to a certain extent
- It is still possible to
  - Obtain coincident clusters
  - Make the algorithm crash with a single outlier

- Considered a successful algorithm
- It deserves further development

Let us make it adaptive

- Why is the penalty term treated as constant?
- Minimizing against penalty terms would reduce them to 0
  - Which would reduce the algorithm to FCM

- We need an optimal formula which allows the change of penalty terms η_i

FCMA to handle clusters of different width

- Miyamoto & Kurosawa (2004)
- Objective function
- Probabilistic constraints as in FCM
- Extra constraint

$$J_{\mathrm{FCMA}} = \sum_{i=1}^{c} \sum_{k=1}^{n} \alpha_i^{1-m} u_{ik}^m d_{ik}^2$$

$$\sum_{i=1}^{c} \alpha_i = 1.$$

- Partition update formula

$$u_{ik} = \frac{\alpha_i d_{ik}^{-2/(m-1)}}{\sum_{j=1}^{c} \alpha_j d_{jk}^{-2/(m-1)}} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

- Cluster prototype update formula

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^m} \qquad \forall i \in \mathbb{N}_c$$

- Update formula

$$\alpha_i = \frac{\left(\sum_{k=1}^{n} u_{ik}^m d_{ik}^2\right)^{1/m}}{\sum_{j=1}^{c} \left(\sum_{k=1}^{n} u_{jk}^m d_{jk}^2\right)^{1/m}} \qquad \forall i \in \mathbb{N}_c$$

Self-tuning PFCM

- Objective function

$$J_{\mathrm{ST-PFCM}} = \sum_{i=1}^{c}\sum_{k=1}^{n} \rho_i^{1-m}(au_{ik}^m + bt_{ik}^p)d_{ik}^2 + (1-t_{ik}^p)\eta$$

- Subject to

$$\begin{cases} u_{ik}, t_{ik} \in [0,1] & \forall i \in \mathbb{N}_c,\ \forall k \in \mathbb{N}_n \\ \sum_{i=1}^{c} u_{ik} = 1 & \forall k \in \mathbb{N}_n \\ 0 < \sum_{i=1}^{c} t_{ik} < c & \forall k \in \mathbb{N}_n \\ \rho_i \in [0,c] & \forall i \in \mathbb{N}_c \\ \sum_{i=1}^{c} \rho_i = c & \forall i \in \mathbb{N}_c \\ m > 1,\ p > 1, a > 0 & b > 0 \end{cases}$$

AO of ST-PFCM

$$u_{ik} = \frac{\rho_i d_{ik}^{-2/(m-1)}}{\sum_{j=1}^{c} \rho_j d_{jk}^{-2/(m-1)}} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

$$t_{ik} = \left[1 + \left(\frac{b d_{ik}^2}{\eta \rho_i^{m-1}} \right)^{1/(p-1)} \right]^{-1} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} (a u_{ik}^m + b t_{ik}^p) \mathbf{x}_k}{\sum_{k=1}^{n} (a u_{ik}^m + b t_{ik}^p)} \qquad \forall i \in \mathbb{N}_c$$

$$\rho_i = c \cdot \frac{\left(\sum_{k=1}^{n} (a u_{ik}^m + b t_{ik}^p) d_{ik}^2 \right)^{1/m}}{\sum_{j=1}^{c} \left(\sum_{k=1}^{n} (a u_{jk}^m + b t_{jk}^p) d_{jk}^2 \right)^{1/m}} \qquad \forall i \in \mathbb{N}_c$$

What makes it self-tuning?

- True penalty term for cluster i

$$\eta_i = \eta \rho_i^{m-1} / b$$

$$t_{ik} = \left[1 + \left(\frac{d_{ik}^2}{\eta_i} \right)^{1/(p-1)} \right]^{-1} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

$$t_{ik} = \left[1 + \left(\frac{b d_{ik}^2}{\eta \rho_i^{m-1}} \right)^{1/(p-1)} \right]^{-1} \qquad \forall i \in \mathbb{N}_c \quad \forall k \in \mathbb{N}_n$$

- ST-PFCM changes the values of ρ_i every cycle
- It updates the width of the clusters

Penalty terms contain a constant η

- It is only a component of the penalty terms
- Recommendation
- Use F(C+1)M, not FCM

$$\eta = \frac{1}{c} \cdot \sum_{i=1}^{c} \eta_{i0} = \frac{1}{c} \cdot \sum_{i=1}^{c} \frac{\sum_{k=1}^{n} u_{ik}^{m} \overline{d}_{ik}^{2}}{\sum_{k=1}^{n} u_{ik}^{m}}$$

$$\overline{d}_{ik} = \begin{cases} d_0 & \text{if} \quad \max\{u_{jk}, j \in \mathbb{N}_c\} < \varepsilon \\ d_{ik} & \text{otherwise} \end{cases}$$

Expectation

- Conditions of success
  - ST-PFCM should work fine wherever PFCM works fine
  - ST-PFCM should work fine also in some cases where PFCM fails

- ST-PFCM is still not expected to be the ideal c-means clustering algorithm

Evaluation criteria

- Main criterion is cluster validity
- There is no cluster validity index for such mixed partitions, so we invent some

- Modified Xie-Beni

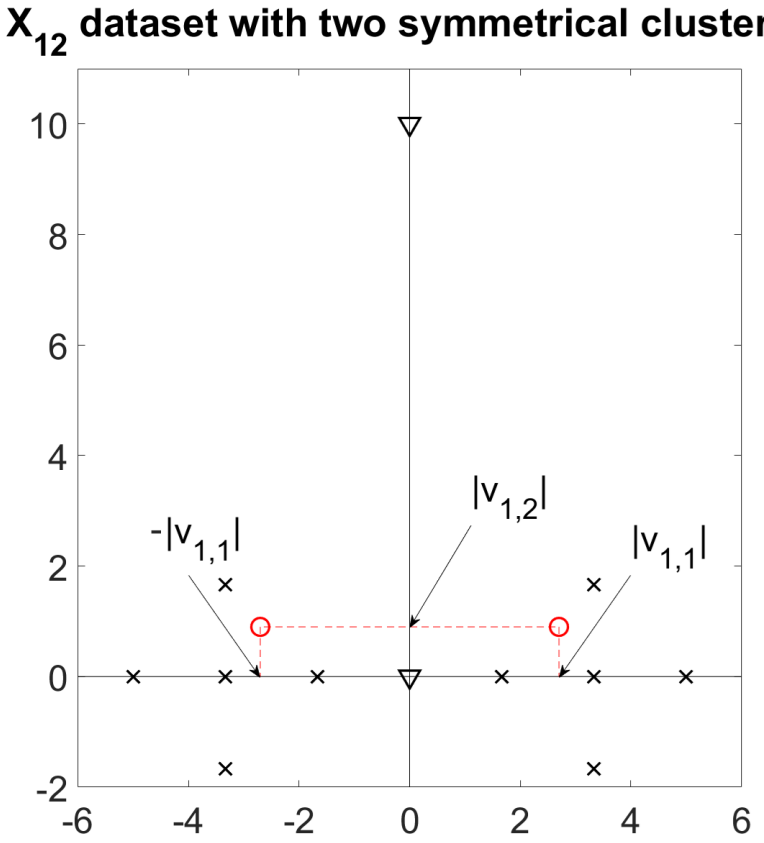
$$MXB = \frac{\sum_{i=1}^{c} \sum_{k=1}^{n} (au_{ik}^{m} + bt_{ik}^{p}) \|\mathbf{x}_k - \mathbf{v}_i\|^2}{n \cdot \min_{i \neq j} \|\mathbf{v}_i - \mathbf{v}_j\|^2}$$

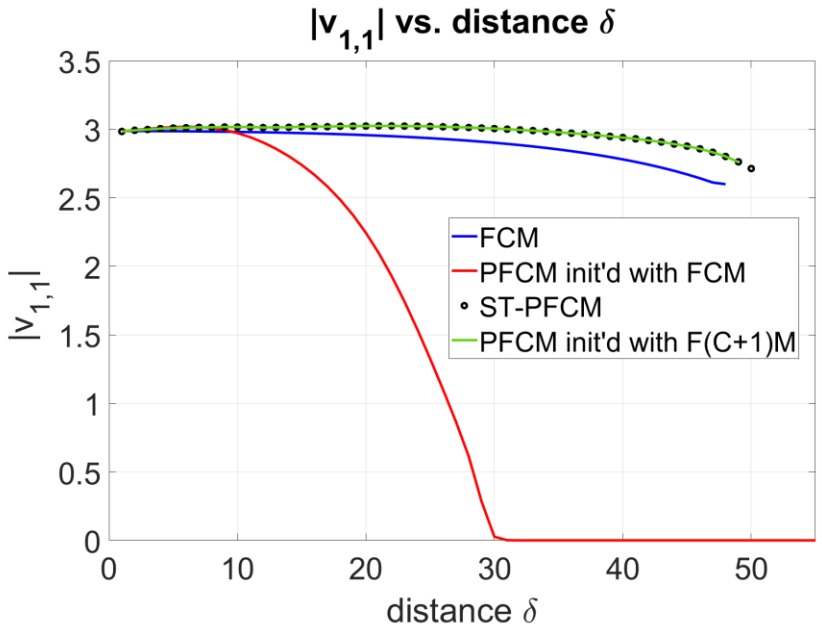
- Modified Fukuyama-Sugeno

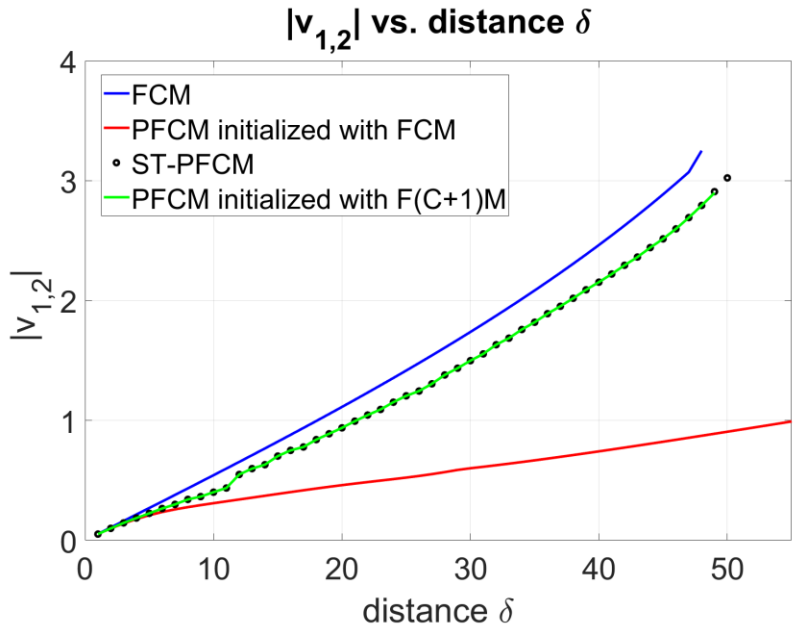
$$MFS = \sum_{i=1}^{c} \sum_{k=1}^{n} (au_{ik}^{m} + bt_{ik}^{p})(\|\mathbf{x}_k - \mathbf{v}_i\|^2 - \|\mathbf{v}_i - \overline{\mathbf{v}}\|^2),$$

- Valid partitions are represented by low values
- MFS can be negative

Initialization problem demonstrated





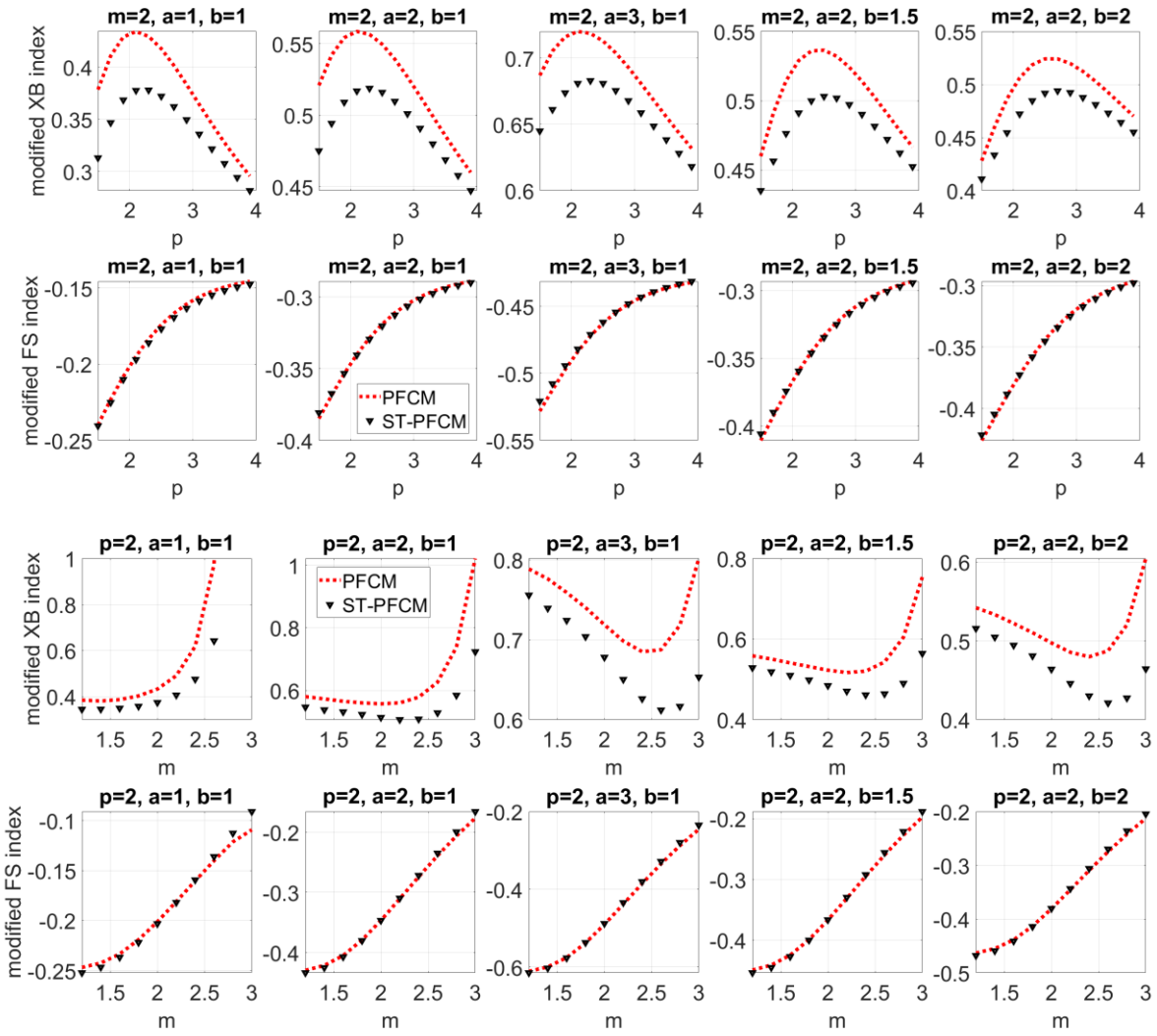


Data

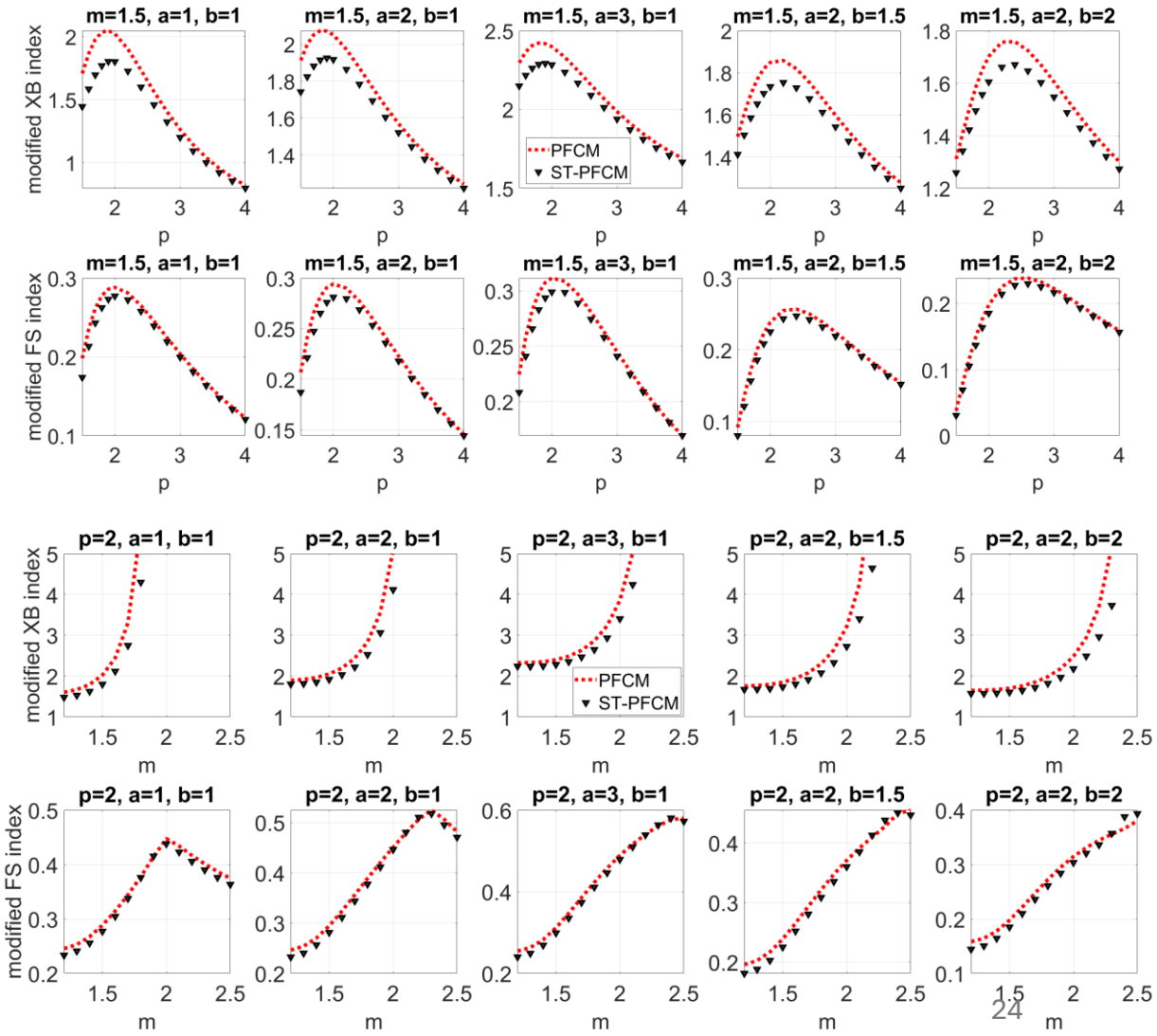
- Test datasets in Table below
- All data vectors normalized, sclaed to [0,1] interval in all dimensions
- Scenario 1: no added noise
- Scenario 2: single outlier at (δ,δ. δ,… δ), with δ>1
- Scenario 3: increasing number of random data points with coordinates in [-1,1] interval in all dimensions

Property	IRIS	WINE	BC	SEEDS
Items	150	178	569	210
Dimensions	4	13	30	7
Clusters	3	3	2	3
Cluster sizes	50, 50, 50	59, 71, 48	357, 212	70, 70, 70

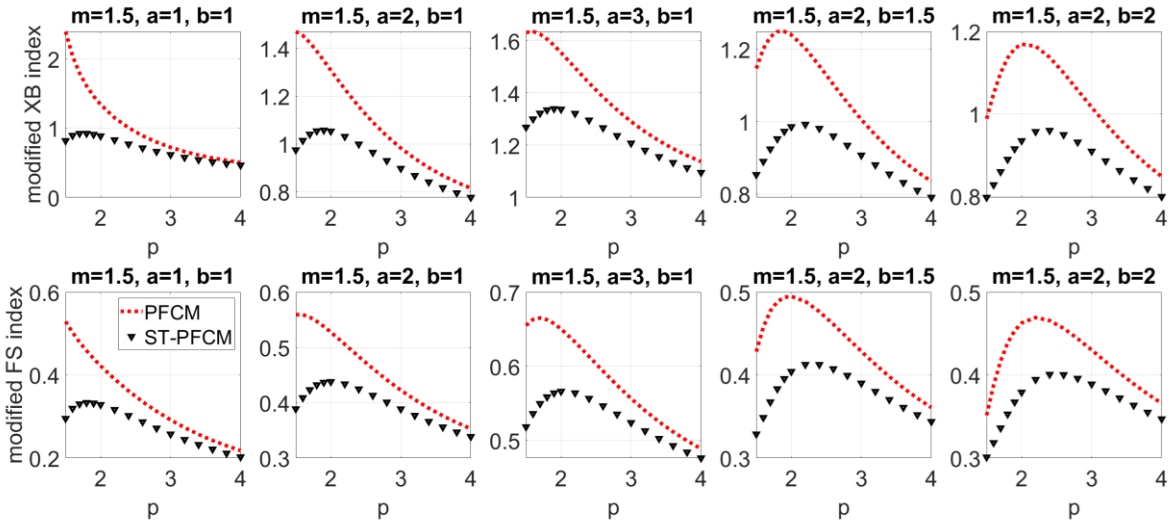
IRIS data

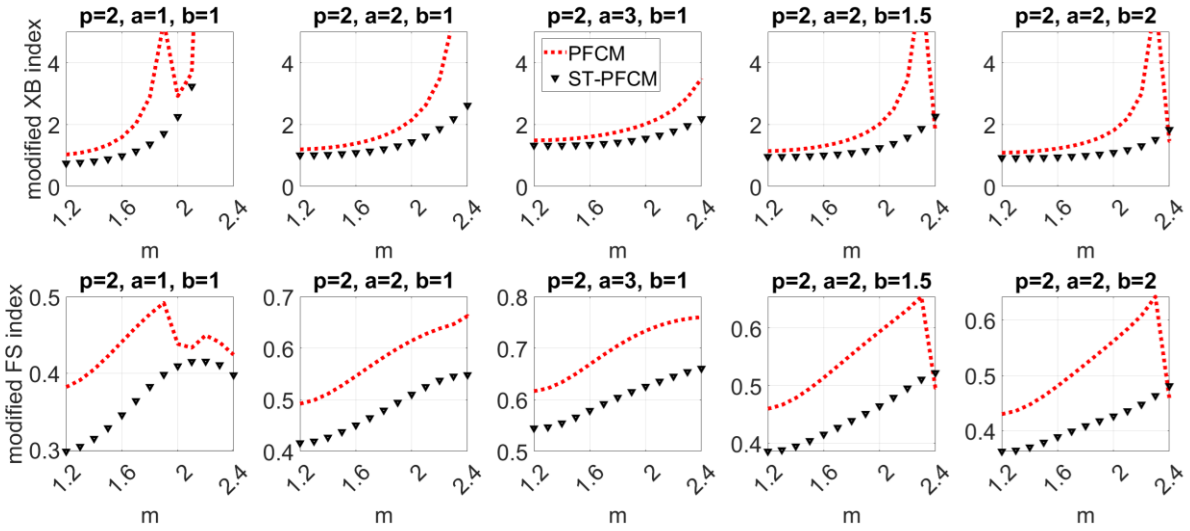


WINE data

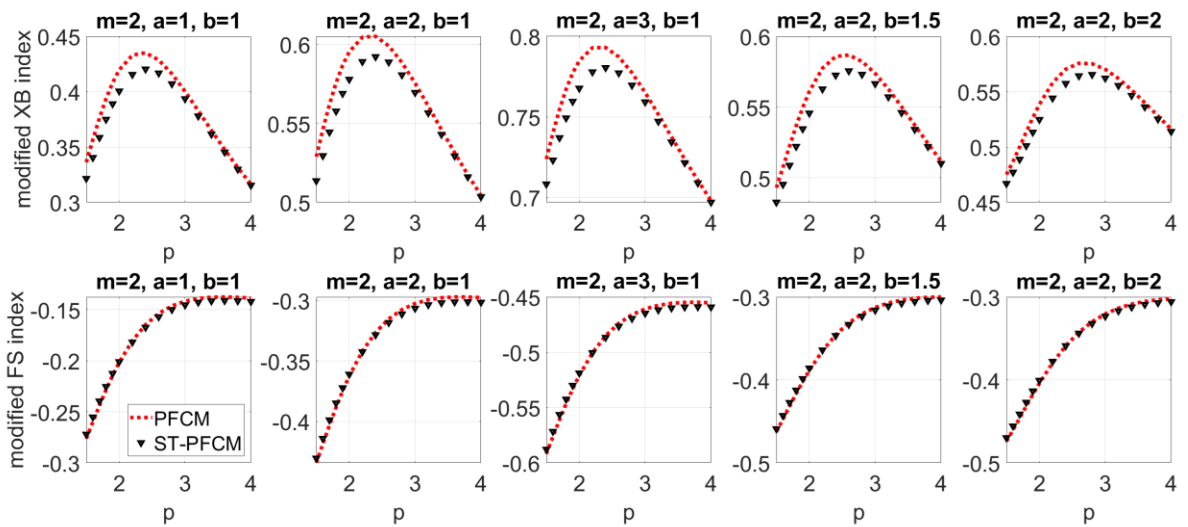


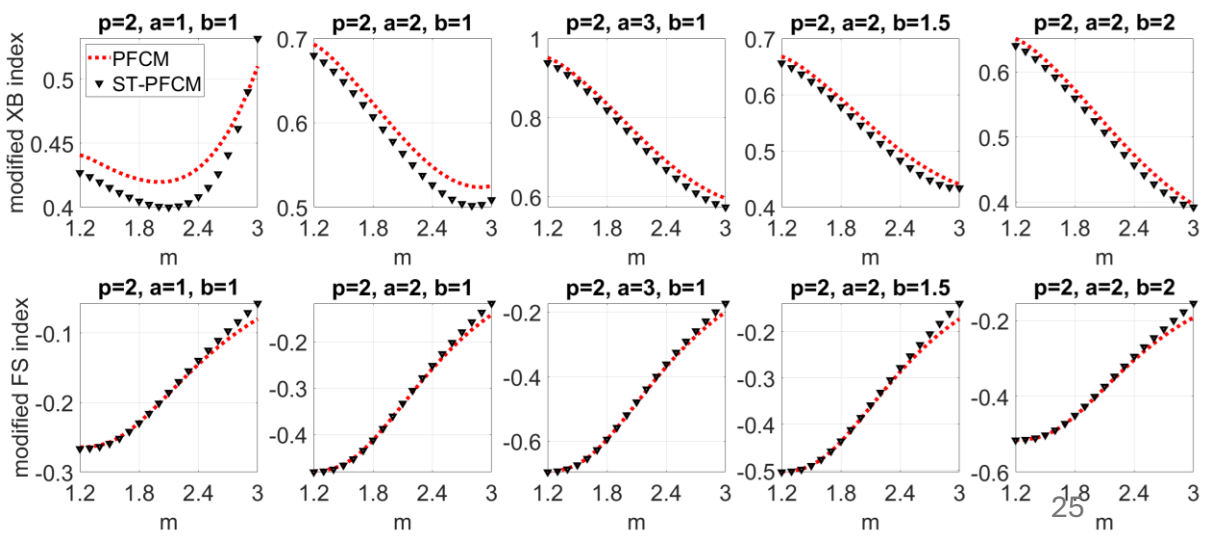
BC data



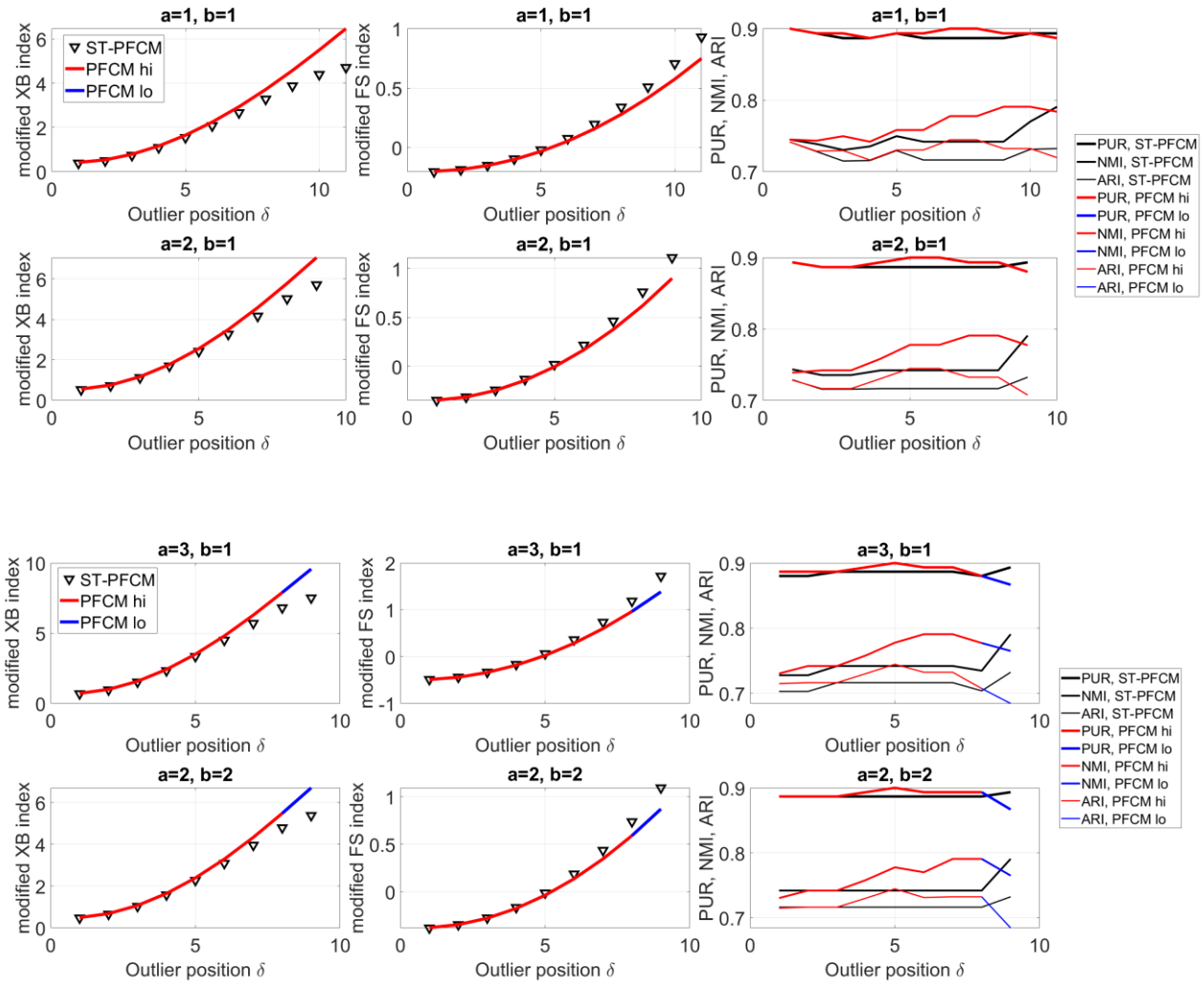


SEEDS data

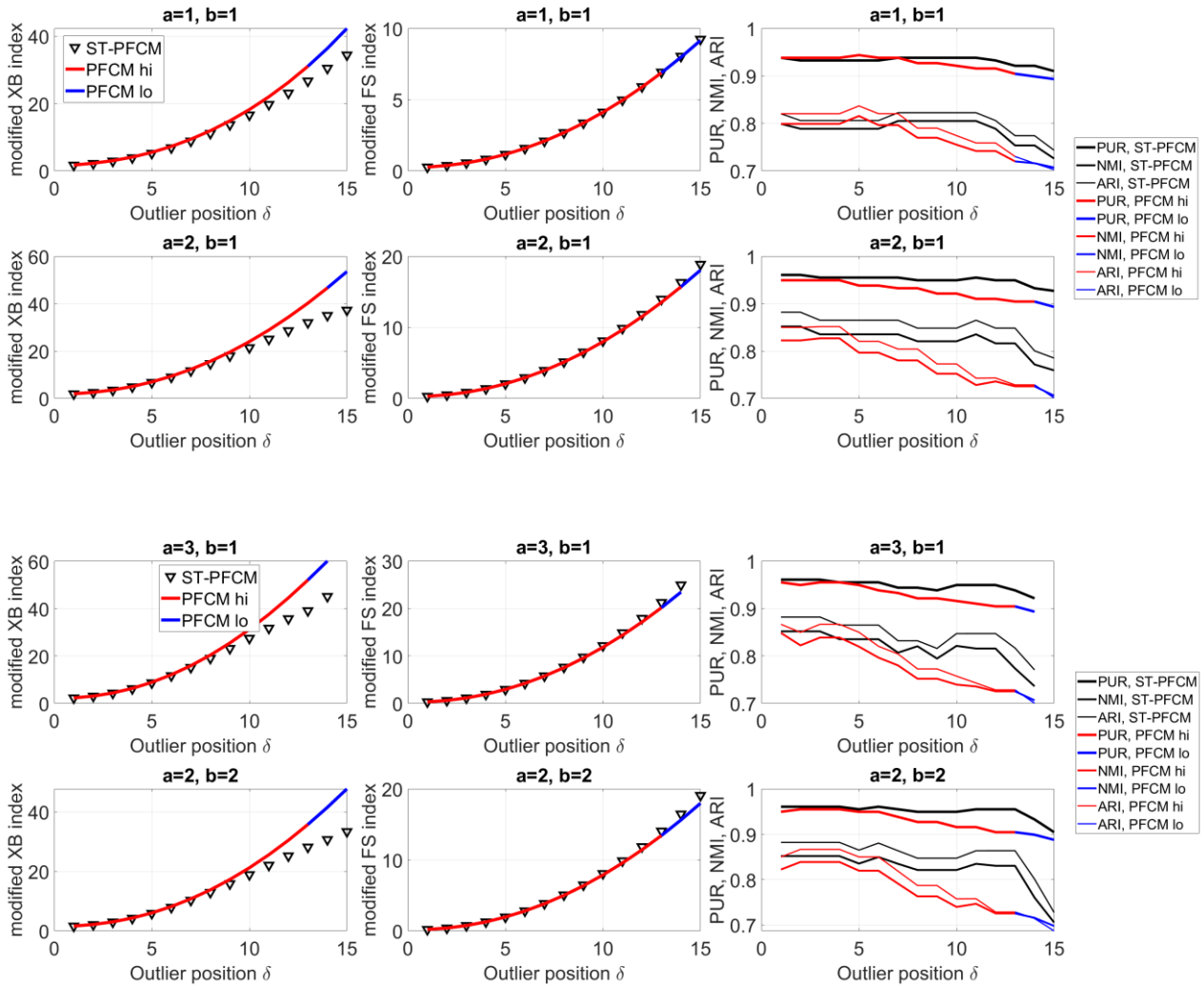




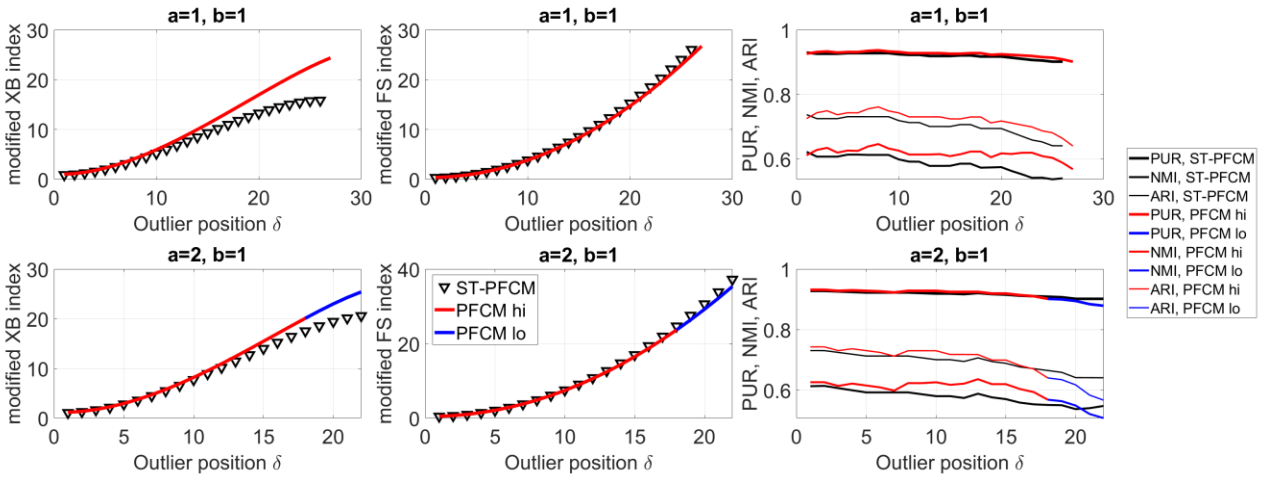
IRIS data

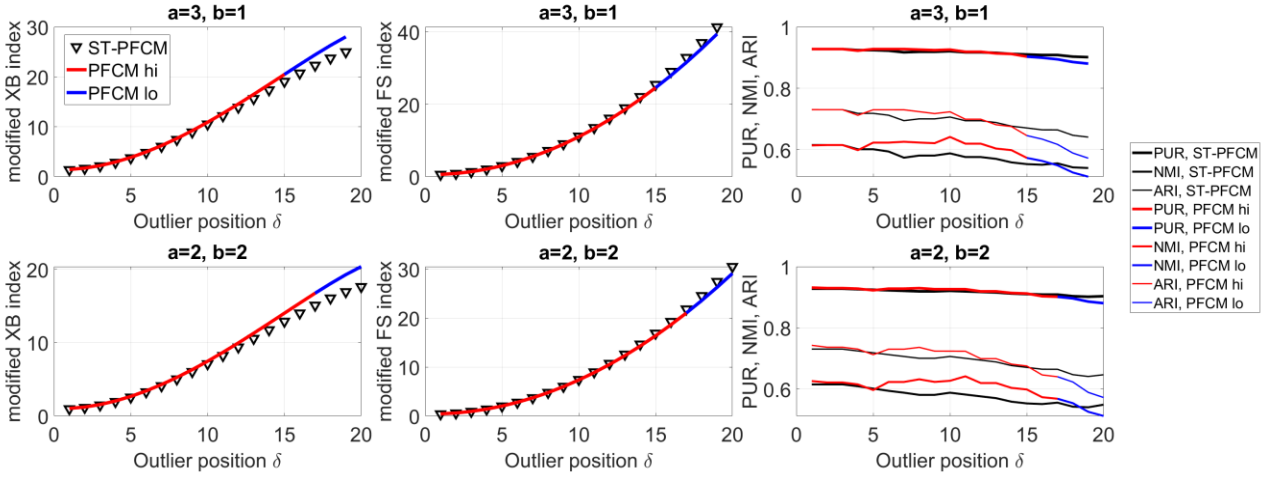


WINE data

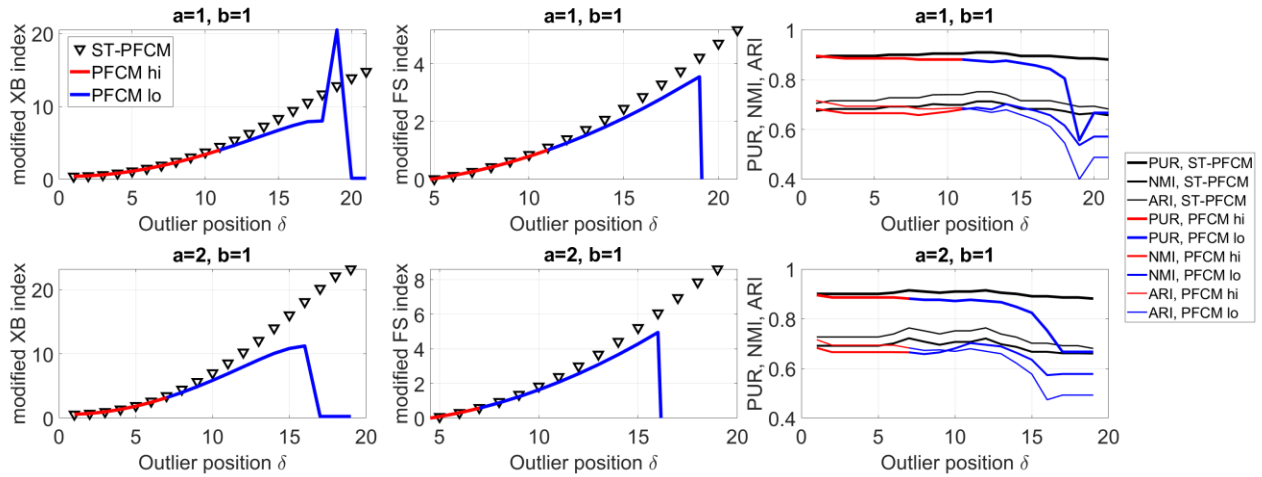


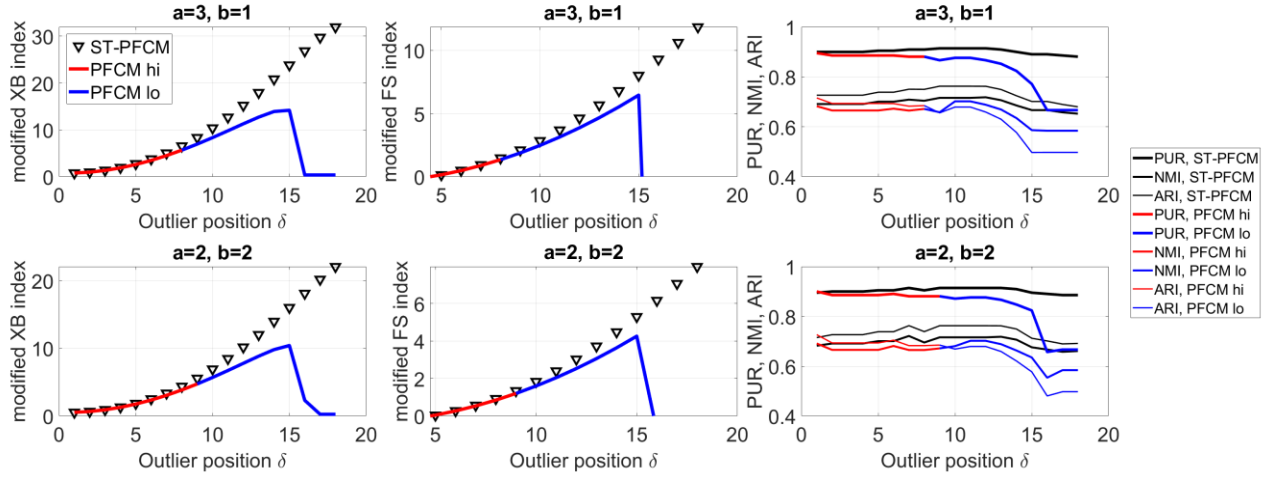
BC data





SEEDS data



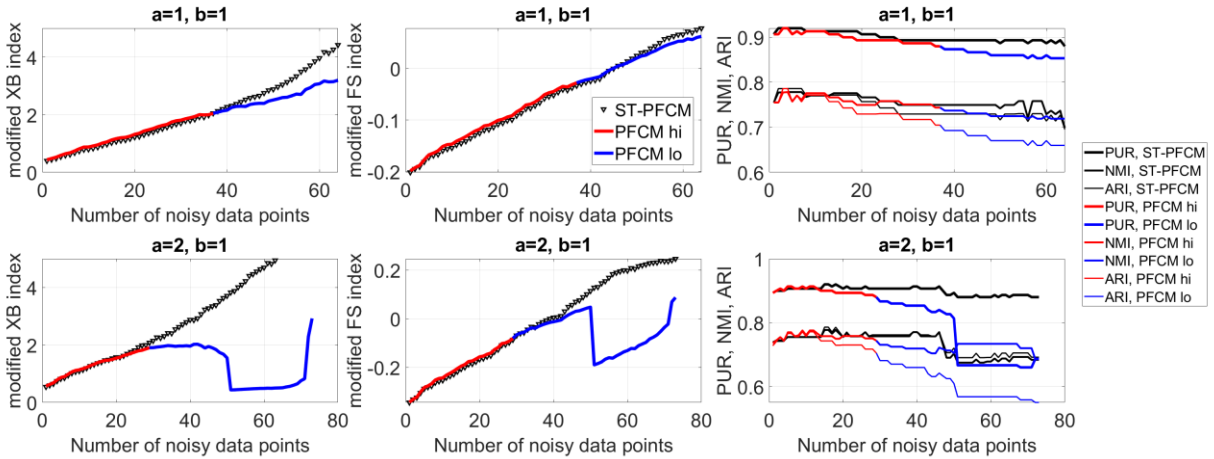


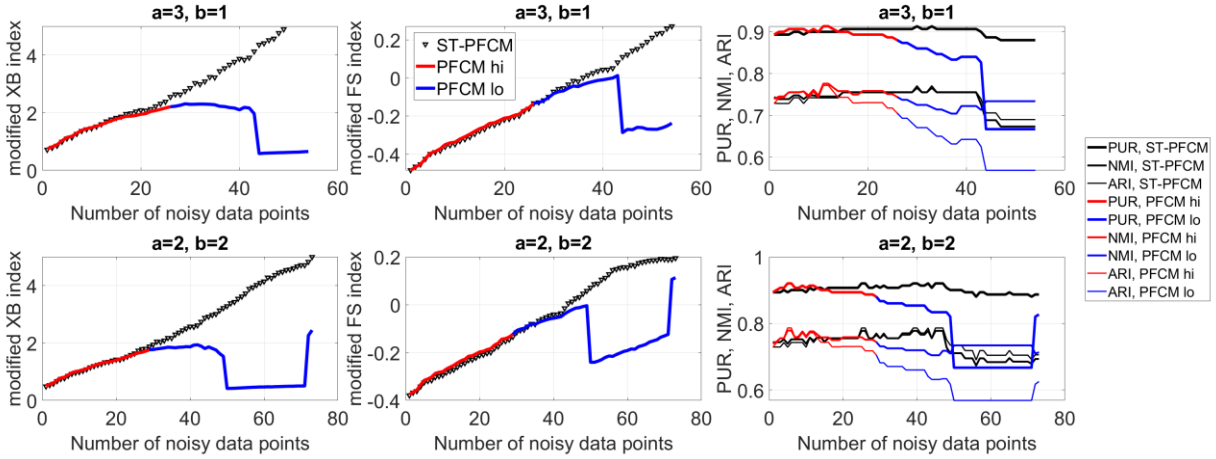
Limit value of δ, PFCM vs. ST-PFCM

Table 3: Comparison PFCM vs ST-PFCM in case of a single outlier: the limit value of parameter δ tolerated by the algorithms while providing acceptable partition

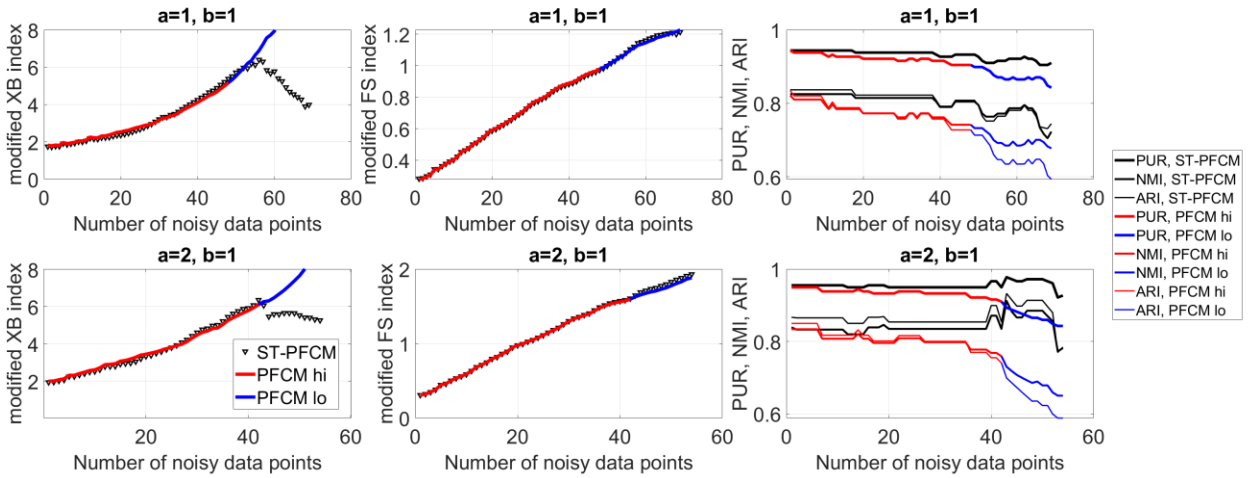
Trade-off		IRIS data				WINE data				BC data				SEEDS data			
		$m = 2.0$				$m = 1.5$				$m = 1.5$				$m = 2.0$			
a	b	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$
1	1	11:11	11:11	11:11	10:11	14:16	13:15	14:16	13:16	25:24	22:26	25:25	21:22	12:21	12:21	11:21	10:19
2	1	9:9	9:9	9:9	8:9	14:15	14:15	13:14	13:14	18:20	18:22	17:21	15:21	9:19	7:19	9:18	9:18
3	1	8:9	8:9	8:9	7:9	13:14	13:14	12:13	12:13	15:19	15:19	15:19	13:19	8:18	8:18	6:18	8:17
2	2	8:9	8:9	8:9	8:9	14:15	13:15	13:14	13:14	14:18	17:20	17:20	15:21	9:18	9:18	9:18	9:17

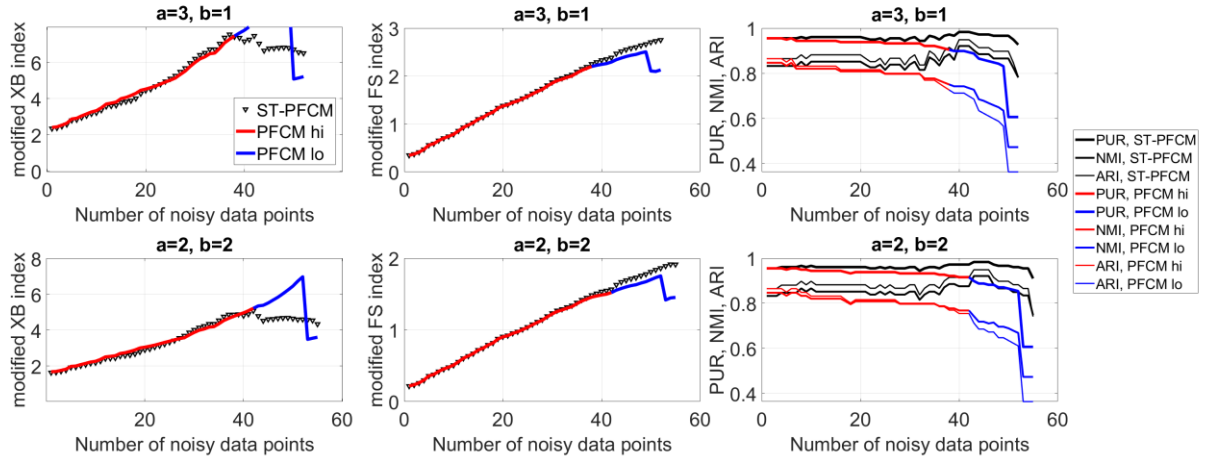
IRIS data



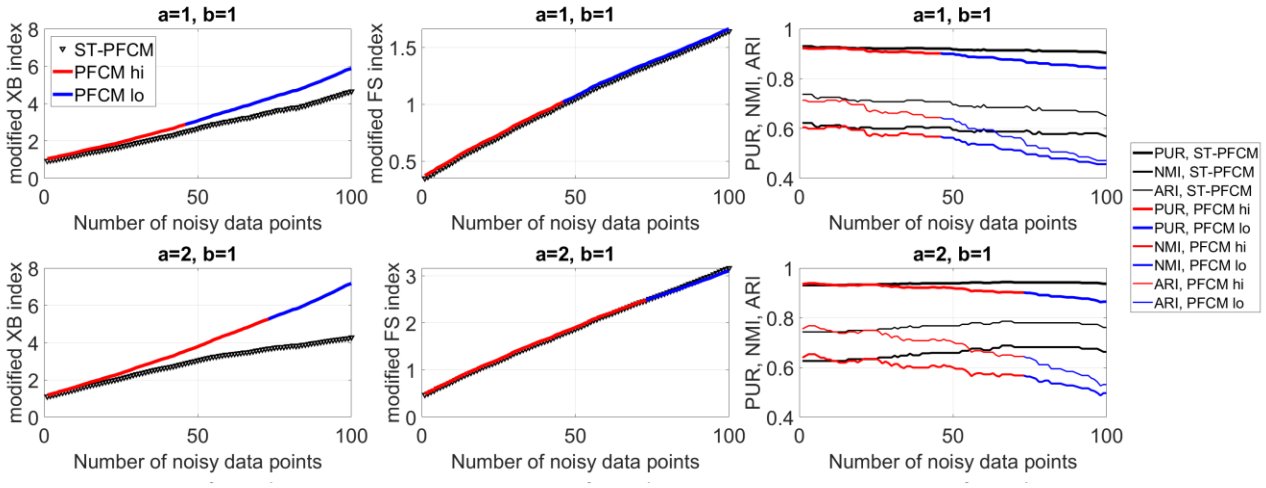


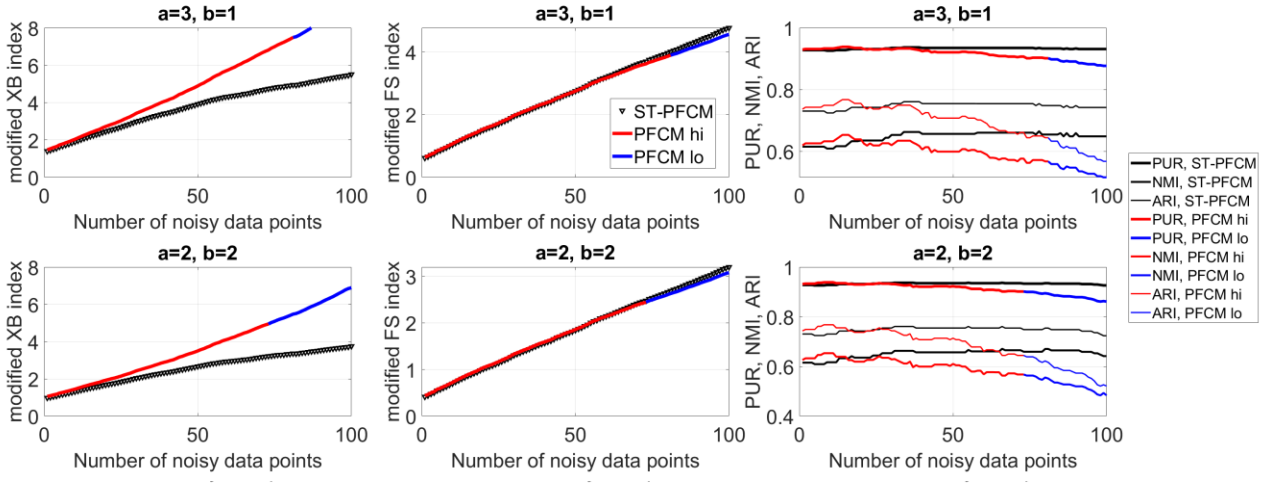
WINE data



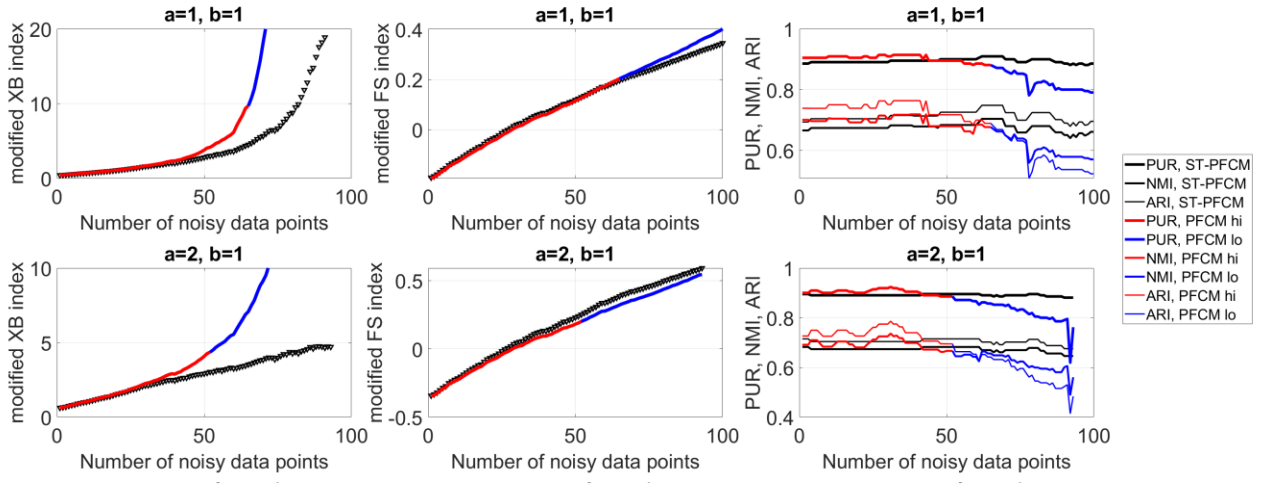


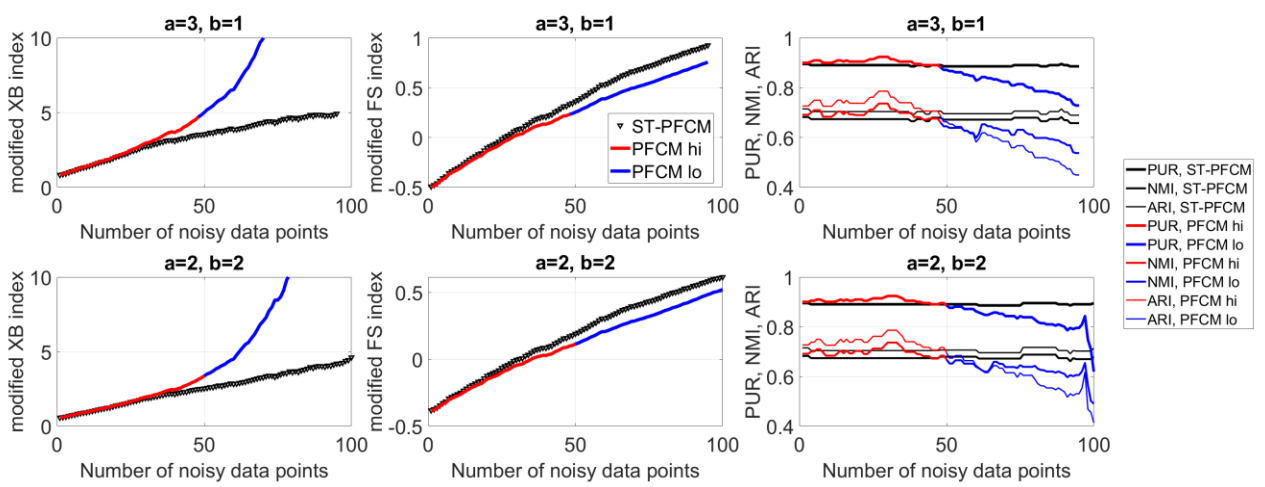
BC data





SEEDS data





Noisy data points tolerated, PFCM vs. ST-PFCM

Table 4: Comparison PFCM vs ST-PFCM in case of noisy data: the limit number of noisy vectors tolerated by the algorithms while providing acceptable partition

Trade-off		IRIS data				WINE data				BC data				SEEDS data			
		$m = 2.0$				$m = 1.5$				$m = 1.5$				$m = 2.0$			
a	b	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$	$p = 1.5$	$p = 2$	$p = 2.5$	$p = 3$
1	1	43:67	33:60	29:76	26:51	65:65	47:67	41:55	38:59	21:53	62:116	75:200	75:200	62:47	47:112	55:92	52:83
2	1	29:47	28:73	25:56	25:50	49:59	47:54	37:50	35:49	44:200	73:200	80:200	73:200	52:93	47:122	46:92	45:89
3	1	27:43	25:46	25:52	23:45	42:55	37:52	35:50	35:49	63:200	81:200	85:200	79:200	49:50	43:48	43:95	43:89
2	2	27:48	27:54	25:56	25:53	43:55	42:54	39:52	36:50	75:97	85:200	89:200	83:200	55:45	47:48	46:106	45:89

Remarks

- If we are not sure that our data is noise-free, F(c+1)M should be used for initialization
- In most of the noise-free cases, PFCM and ST-PFCM perform equally well
- In few noise-free cases and in more noisy cases, ST-PFCM works better
- In a noisy environment, ST-PFCM is likely to perform better than PFCM
- It is hard to define cases where PFCM is definitely better than ST-PFCM

- Further needs: e.g. complexity analysis

Conclusions

- We proposed an alternative initialization scheme to avoid the mistaken start
- We have created a self-tuning version of the PFCM algorithm
- Adaptively changes the possibilistic penalty terms
- It has the chance to create finer partitions than PFCM
- In several test cases it really performed better than PFCM

Acknowledgements