Acta Univ. Sapientiae Informatica 15, 2 (2023) 404–431

DOI: 10.2478/ausi-2023-0023

A generalized fuzzy-possibilistic c-means clustering algorithm

Mirtill-Boglárka NAGHI
Sapientia Hungarian University of Transylvania, Cluj-Napoca, Romania
Óbuda University, Budapest, Hungary
Doctoral School of Applied Mathematics and Applied Informatics
email: naghi.mirtill@ms.sapientia.ro
ORCID: 0009-0005-8936-7769

Levente KOVÁCS
Óbuda University, Budapest, Hungary
University Research, Innovation and Service Center
email: kovacs@uni-obuda.hu
ORCID:0000-0002-3188-0800

László SZILÁGYI
Computational Intelligence Research Group,
Sapientia Hungarian University of Transylvania, Cluj-Napoca, Romania
Dept. of Electrical Engineering, Târgu Mureş
Óbuda University, Budapest, Hungary
University Research, Innovation and Service Center
email: lalo@ms.sapientia.ro
szilagyi.laszlo@uni-obuda.hu
ORCID: 0000-0001-6722-2642

Abstract. The so-called fuzzy-possibilistic c-means (FPCM) algorithm was introduced as an early mixed-partition method aiming to eliminate

Key words and phrases: fuzzy c-means algorithm, possibilistic c-means algorithm, mixed partition

some adverse effects present in the behavior of the fuzzy c-means (FCM) and the possibilistic c-means (PCM) algorithms. A great advantage of FPCM was the low number of its parameters, as it eliminated the possibilistic penalty terms used by PCM. Unfortunately, FPCM in its original formulation also has a weak point: the strength of the possibilistic term is in inverse proportion with the number of clustered data items, which makes FPCM act like FCM when clustering large sets of data. This paper proposes a modification of the FPCM algorithm by introducing an extra coefficient into the possibilistic term that allows us to control the strength of the possibilistic effect within the mixture model. The modified clustering model will be referred to as generalized FPCM, since a certain value of the extra parameter reduces it to the original FPCM, or in other words, FPCM is a special case of the proposed algorithm. The proposed method is evaluated using noise-free and noisy data as well.

1 Introduction

Data clustering represents one of the first applications of Zadeh's fuzzy logic [24]. The first fuzzy partitioning was defined by Ruspini [15] in 1969, while the first c-means clustering adopting fuzzy partitions is the ISODATA algorithm of Dunn introduced in 1974 [6], which was later generalized by Bezdek [3] and called fuzzy c-means (FCM) algorithm. FCM has been a very popular algorithm over the past decades in a wide range of sciences, in spite of its high sensitivity to noisy data, caused by the probabilistic constraint used by all c-means clustering models defined up to this point.

The necessity to relax the probabilistic constraint led to a series of c-means clustering approaches (e.g. [5, 9]), in which the fuzzy membership functions represented typicality values or the compatibility of data vectors with the clusters. These approaches were able to handle noisy data, to ignore them in establishing the clusters that represent the real, meaningful data vectors. However, they still had limitations: the first one was unable to handle clusters of different sizes (diameter), the second one frequently merged several clusters together.

To avoid this limitation, several mixed partition models of c-means clustering were proposed. In 1997, Pal et al. [11] introduced the fuzzy-possibilistic c-means (FPCM) algorithm, which proposed a mixture partition with a reduced number of parameters. This approach had the limitation of having a behavior strongly influenced by the size of the input dataset. The probabilistic and possibilistic components of the mixed partition were used as a linear combination. This scheme was then reused by the possibilistic-fuzzy c-means

(PFCM) algorithm proposed by Pal et al. [12], while Szilágyi [16]) later presented a different approach that combined the two partition components via multiplication. The most recent mixed partition models proved to be robust as they provide fine partitions both in case of absence and presence of outlier data.

This paper proposes to enhance the services provided by the FPCM algorithm by introducing a generalized formulation. We attempt to eliminate the limitations of PFCM by adding an extra coefficient to the possibilistic term. This parameter denoted by β defines the strength of the possibilistic term in the mixture partition. This modification represents a generalization of the FPCM algorithm because FPCM acts as a special case ($\beta = 1$) of the novel approach, while β can have any positive real value, each leading to a different algorithm.

The proposed clustering model is evaluated using standard datasets taken from the literature, in their original noise-free version, but with some added outliers as well. The evaluation process helps us in formulating recommendations regarding the parameters of the algorithm.

The rest of this paper is structured as follows: Section 2 presents the basic c-means clustering algorithms this work relies on. Section 3 exhibits the details of the proposed clustering model. Section 4 relates on the numerical evaluation of the proposed clustering model in comparison to other c-means clustering algorithms. Section 5 discusses the role of the main parameters and formulates recommendations regarding the use of the proposed method. Section 6 concludes the study.

2 Background works

All c-means clustering algorithms aim at grouping a set of object data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_n\}$ into a fixed number of clusters. Clusters are denoted by Ω_i ($i = 1, \dots, c$), where c is the number of clusters. As precondition, it is supposed that $c < n$. In real-life problems, usually the number of input data vectors exceeds the number of clusters by orders of magnitude. Each cluster Ω_i ($\forall i = 1, \dots, c$) is represented by the cluster prototype $\mathbf{v}_i$, which is a vector of same type as the input data.

All c-means clustering models use a partition matrix. The partition matrix generally describes to what extent data vectors belong to each of the classes. In this study we only investigate clustering algorithms that use fuzzy partitions, meaning that all elements of the partition matrix represent fuzzy membership functions. We use two different notations for the partition matrix: $\mathbf{U} = [u_{ik}] \in \mathcal{M}_{c\times n}$ and $\mathbf{T} = [t_{ik}] \in \mathcal{M}_{c\times n}$. The difference between these two matrices is that u_{ik} values satisfy the probabilistic constraint, meaning that all columns of matrix $\mathbf{U}$ sum up to 1, while the columns of $\mathbf{T}$ contain typicality values, t_{ik} ($i = 1, \dots, c$; $k = 1, \dots, n$) expressing how much vector $\mathbf{x}_k$ is compatible with cluster Ω_i.

2.1 The fuzzy c-means algorithm

The fuzzy c-means clustering algorithm minimizes the following objective function:

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

being subject to the probabilistic constraint

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall k = 1, \dots, n \ , \tag{2}$$

where $d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|$ represents the distance between input vector $\mathbf{x}_k$ and cluster prototype $\mathbf{v}_i$, for any $i = 1, \dots, c$, and $k = 1, \dots, n$. Parameter $m > 1$ is the fuzzy exponent than controls the fuzziness of the algorithm. It is known, that the limit case $m \to 1$ reduces FCM to the k-means algorithm that uses binary logic to describe the partition. On the other side, if $m \to +\infty$, cluster prototypes merge together at the grand mean of the input data vectors. Raising the value of m makes the algorithm more fuzzy.

The optimization formulas of FCM are obtained from the zero gradient conditions of its objective function extended with special terms containing Lagrange multipliers that enforce the probabilistic constraint. The optimization formulas are obtained as:

$$u_{ik} = \frac{d_{ik}^{\frac{-2}{m-1}}}{\sum_{j=1}^{c} d_{jk}^{\frac{-2}{m-1}}} \qquad \begin{array}{l}\forall i = 1, \dots, c \\ \forall k = 1, \dots, n\end{array} \ , \tag{3}$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{n} u_{ik}^{m}} \qquad \forall i = 1, \dots, c \ . \tag{4}$$

The algorithm needs to be initialized with cluster prototypes differing from each other. The optimization is performed by alternately applying the formulas given in Eqs. (3) and (4) until convergence is reached. Convergence is reached when cluster prototypes stabilize. If we need to defuzzify the final partition, we may assign each data vector to the cluster whose prototype is closest, or the one with respect to which the fuzzy membership function has highest value. These two criteria are equivalent:

$$\mathbf{x}_k \to \Omega_i \quad \Leftrightarrow \quad i = \arg\min_j \{d_{jk}, j = 1, \ldots, c\} = \arg\max_j \{u_{jk}, j = 1, \ldots, c\} \tag{5}$$

Besides being a very popular algorithm in all sciences involving numerical data, a major disadvantage of FCM is its sensitivity to outlier data. A single distant outlier can attract cluster prototypes out of the range of the elements that it represents, or in extreme case the outlier may "steal" one of the cluster prototypes, causing poor partitioning of the real meaningful data.

2.2 The possibilistic c-means algorithm

The noise sensitivity of FCM was attributed to the probabilistic constraints of the partition, and thus several solutions emerged that relaxed this too strong limitation. The algorithm called FCM with extra noise class and also referred to as fuzzy $(c+1)$-means defined an extra cluster Ω_0 which has no cluster prototype and is situated at an equal constant distance d_0 from all input vectors x_k $(k = 1, \ldots, n)$. The probabilistic constraint in this case looks like

$$\sum_{i=0}^{c} u_{ik} = 1 \qquad \forall k = 1, \ldots, n \ , \tag{6}$$

but now any noisy data vector x_k receives a high membership towards the noise class and this way it will hardly influence the cluster prototypes. The algorithm needs careful initialization, meaning that initial cluster prototypes should not be set to the noisy data vector. A limitation of this algorithm stands in the fact that similarly to FCM, it cannot handle clusters of different widths (radii).

Theoretically all c-means clustering models that use fuzzy partitions without constraining the fuzzy memberships with probabilistic condition could be called possibilistic c-means. However, the so-called possibilistic c-means algorithm is the one introduced by Krishnapuram and Keller [9]. PCM minimizes the following objective function:

$$J_{\mathrm{PCM}} = \sum_{i=1}^{c} \sum_{k=1}^{n} t_{ik}^{p} d_{ik}^{2} + (1 - t_{ik})^{p} \eta_i \ , \tag{7}$$

subject to the possibilistic constraint

$$0 < \sum_{i=1}^{c} t_{ik} < c \qquad \forall k = 1, \ldots, n \ , \tag{8}$$

which means that all data vectors must belong to at least one cluster to a nonzero extent, and none of the data vectors can be fully compatible with all clusters. Further on, η_i represents the possibilistic penalty term of cluster i ($i = 1, \ldots, c$) which is meant to control the width of the cluster, and $p > 1$ represents the so-called possibilistic exponent.

Similarly to the FCM algorithm, the optimization formulas are extracted from the zero gradient conditions of the objective function, but here there is no need to use Lagrange multipliers. The optimization formulas are obtained as:

$$t_{ik} = \left[1 + \left(\frac{d_{ik}^{2}}{\eta_i} \right)^{\frac{1}{p-1}} \right]^{-1} \qquad \begin{array}{l} \forall i = 1, \ldots, c \\ \forall k = 1, \ldots, n \end{array} \ , \tag{9}$$

and

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} t_{ik}^{p} \mathbf{x}_k}{\sum_{k=1}^{n} t_{ik}^{p}} \qquad \forall i = 1, \ldots, c \ , \tag{10}$$

which are alternately applied until cluster prototypes stabilize. PCM can produce fine partitions even in the presence of outlier data, but unfortunately it frequently merges several or all clusters together. If we need to defuzzify the final partition, each input data vector is assigned to the cluster with which it shows the highest compatibility:

$$\mathbf{x}_k \to \Omega_i \quad \Leftrightarrow \quad i = \arg\max_j \{ t_{jk}, j = 1, \ldots, c \} \tag{11}$$

2.3 Algorithms using mixed partitions

Since none of the two basic approaches of fuzzy logic based c-means clustering proved perfect, several attempts were made to merge the two partition matrices into a mixed partition, and expected them to relax or attenuate each other's limitations. A linear combination of the classical FCM and PCM partitions was proposed by Pal et al. [12], referred to as possibilistic fuzzy c-means algorithm. The other approach called fuzzy possibilistic product partition c-means algorithm was proposed by Szilágyi [16] and later generalized for clusters with special shapes [17, 19]. However, this paper intends to generalize the method called fuzzy-possibilistic c-means (FPCM) algorithm introduced by Pal et al. [11], which uses an alternative definition for the possibilistic partition that is involved into a linear combination with the FCM partition matrix.

FPCM minimizes the following objective function:

$$J_{\mathrm{FPCM}} = \sum_{i=1}^{c}\sum_{k=1}^{n} \left(u_{ik}^{m} + t_{ik}^{p}\right) d_{ik}^{2} \ , \tag{12}$$

constrained by

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall k = 1, \ldots, n \ , \tag{13}$$

and

$$\sum_{k=1}^{n} t_{ik} = 1 \qquad \forall i = 1, \ldots, c \ , \tag{14}$$

where $m > 1$ and $p > 1$ represent the fuzzy and possibilistic exponents, respectively. Both constraints presented in Eqs. (13) and (14) may seem probabilistic at first sight. However, the elements of partition matrix $\mathbf{U}$ sum up to 1 in each column, while in $\mathbf{T}$ they sum up to 1 in each row.

The optimization formulas of FPCM are obtained from the zero gradient conditions of the objective functions, extended with terms that enforce the constraints by the use of Lagrange multipliers. The alternately applied optimization formulas are obtained as:

$$u_{ik} = \frac{d_{ik}^{\frac{-2}{m-1}}}{\sum\limits_{j=1}^{c} d_{jk}^{\frac{-2}{m-1}}} \qquad \text{and} \qquad t_{ik} = \frac{d_{ik}^{\frac{-2}{p-1}}}{\sum\limits_{l=1}^{n} d_{il}^{\frac{-2}{p-1}}} \qquad \begin{array}{l} \forall i = 1, \ldots, c \\ \forall k = 1, \ldots, n \end{array} \ , \tag{15}$$

and

$$\mathbf{v}_i = \frac{\sum\limits_{k=1}^{n} \left(u_{ik}^{m} + t_{ik}^{p}\right) \mathbf{x}_k}{\sum\limits_{k=1}^{n} \left(u_{ik}^{m} + t_{ik}^{p}\right)} \qquad \forall i = 1, \ldots, c \ , \tag{16}$$

which are applied until cluster prototypes stabilize. The defuzzified partition is defined by the maximum value of the combined partition matrix, according to the formula:

$$\mathbf{x}_k \to \Omega_i \quad \Leftrightarrow \quad i = \arg\max_j \{u_{jk}^m + t_{jk}^p, j = 1, \dots, c\} \ . \tag{17}$$

3 Methods

The problem formulation of the original FPCM does not offer equal chances to the probabilistic and possibilistic components to have their effect upon the final partition. This can be explained with the fact that the total sum of fuzzy membership functions in matrix **U** is n, which is the number of data vectors being fed to clustering, while in matrix **T** the total sum is c, the number of clusters. In the very frequent case, when $n >> c$, the possibilistic term hardly can influence the clustering process. This is why we need to introduce a compensating parameter denoted by β, which appears as a multiplying factor to the possibilistic term in the objective function.

The proposed clustering model, which in the following will be referred to as generalized fuzzy-possibilistic c-means algorithm (GFPCM), optimizes the following objective function:

$$J_{\mathrm{GFPCM}} = \sum_{i=1}^{c}\sum_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right) ||\mathbf{x}_k - \mathbf{v}_i||^2 = \sum_{i=1}^{c}\sum_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right) d_{ik}^2 \ , \tag{18}$$

subject to the same constraints as FPCM, presented in Eqs (13) and (14). All notations are the same as in PFCM, with the exception of β, which is a positive valued parameter. The proposed clustering model generalizes the original FPCM because FPCM is a special case of the proposed algorithm, namely the one that uses $\beta = 1$. For any other positive value of β we obtain a different algorithm. Another special case is the one defined by $\beta = 0$, setting that reduces GFPCM to FCM regardless to the value of p. At first sight it would seem logical to set $\beta = n/c$ so that the two components of the partition get the same strength. However, in this study we investigate the behavior of the algorithm in a wide range of β values, up to even the order of 10^6.

The optimization formulas of the GFPCM algorithm are obtained from the zero gradient conditions of the following functional:

$$\mathcal{L}_{\mathrm{GFPCM}} = J_{\mathrm{GFPCM}} + \sum_{k=1}^{n} \lambda_k \left(1 - \sum_{i=1}^{c} u_{ik}\right) + \sum_{i=1}^{c} \tau_i \left(1 - \sum_{k=1}^{n} t_{ik}\right) \ , \tag{19}$$

where λ_k $(k = 1, \ldots, n)$ and τ_i $(i = 1, \ldots, c)$ represent Lagrange multipliers needed to enforce the constraints during optimization. From the partial derivative with respect to u_{ik} $(\forall i = 1, \ldots, c, \forall k = 1, \ldots, n)$ we obtain:

$$\frac{\partial \mathcal{L}_{\text{GFPCM}}}{\partial u_{ik}} = 0 \Longrightarrow m u_{ik}^{m-1} d_{ik}^2 - \lambda_k = 0, \quad , \tag{20}$$

which implies

$$u_{ik} = \left(\frac{\lambda_k d_{ik}^{-2}}{m} \right)^{\frac{1}{m-1}} = \left(\frac{\lambda_k}{m} \right)^{\frac{1}{m-1}} d_{ik}^{\frac{-2}{m-1}} \quad . \tag{21}$$

We know from Eq. (13), that for any $k = 1, \ldots, n$

$$\sum_{j=1}^{c} u_{jk} = 1 \Longrightarrow 1 = \left(\frac{\lambda_k}{m} \right)^{\frac{1}{m-1}} \sum_{j=1}^{c} d_{jk}^{\frac{-2}{m-1}} \quad . \tag{22}$$

Dividing Eqs. (21) and (22) term by term, we obtain

$$u_{ik} = \frac{u_{ik}}{1} = \frac{d_{ik}^{\frac{-2}{m-1}}}{\sum\limits_{j=1}^{c} d_{jk}^{\frac{-2}{m-1}}} \quad , \tag{23}$$

which is exactly the partition update formula known from FCM. Similarly, from the partial derivatives with respect to t_{ik} $(\forall i = 1, \ldots, c, \forall k = 1, \ldots, n)$, we obtain:

$$\frac{\partial \mathcal{L}_{\text{GFPCM}}}{\partial t_{ik}} = 0 \Longrightarrow \beta p t_{ik}^{p-1} d_{ik}^2 - \tau_i = 0 \quad , \tag{24}$$

which implies

$$t_{ik} = \left(\frac{\tau_i d_{ik}^{-2}}{\beta p} \right)^{\frac{1}{p-1}} = \left(\frac{\tau_i}{\beta p} \right)^{\frac{1}{p-1}} d_{ik}^{\frac{-2}{p-1}} \quad . \tag{25}$$

We know from Eq. (14), that for any $i = 1, \ldots, c$

$$\sum_{l=1}^{n} t_{il} = 1 \Longrightarrow 1 = \left(\frac{\tau_i}{\beta p} \right)^{\frac{1}{p-1}} \sum_{l=1}^{n} d_{il}^{\frac{-2}{p-1}} \quad . \tag{26}$$

Dividing Eqs. (25) and (26), we obtain

$$t_{ik} = \frac{t_{ik}}{1} = \frac{d_{ik}^{\frac{-2}{p-1}}}{\sum\limits_{l=1}^{n} d_{il}^{\frac{-2}{p-1}}} \quad , \tag{27}$$

which is exactly the possibilistic component update formula of FPCM.

The partition update formula is obtained from the partial derivatives with respect to cluster prototype vectors $\mathbf{v}_i$ $(i = 1, \ldots, c)$:

$$\frac{\partial \mathcal{L}_{\mathrm{GFPCM}}}{\partial \mathbf{v}_i} = 0 \Longrightarrow \sum_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right)(-2)(\mathbf{x}_k - \mathbf{v}_i) = 0 \ , \tag{28}$$

which implies

$$\mathbf{v}_i \sum_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right) = \sum_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right) \mathbf{x}_k \ , \tag{29}$$

and consequently we obtain the cluster prototype updated as

$$\mathbf{v}_i = \frac{\sum\limits_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right) \mathbf{x}_k}{\sum\limits_{k=1}^{n} \left(u_{ik}^m + \beta t_{ik}^p\right)} \qquad \forall i = 1, \ldots, c \ . \tag{30}$$

Just like in case of any other c-means clustering algorithm, the cluster prototypes are obtained as the weighted average of input data vectors $\mathbf{x}_k$ $(k = 1, \ldots, n)$, where the weights are obtained in the final partition matrices. The defuzzification rule can be formulated as follows:

$$\mathbf{x}_k \to \Omega_i \quad \Leftrightarrow \quad i = \arg\max_j \{u_{jk}^m + \beta t_{jk}^p, j = 1, \ldots, c\} \ . \tag{31}$$

When initializing cluster prototypes, it is recommendable to choose random vectors that are distant from any of the input data vectors $\mathbf{x}_k$ $(k = 1, \ldots, n)$, just as recently suggested in [14]. Let us suppose the contrary, and initialize for example $\mathbf{v}_a = \mathbf{x}_b$ with some valid values of a and b. In this case in the first iteration $t_{ab} = 1$ and $t_{ak} = 0$ for any $k \neq b$. Especially if we use a high value of parameter β, the algorithm will hardly be able to move the cluster prototype $\mathbf{v}_a$ away from $\mathbf{x}_b$.

The GFPCM algorithm can be summarized as follows:

1. Set parameters m, p, and β.
2. Initialize cluster prototypes outside the range of input data vectors.
3. Update the probabilistic term of the partition using Eq. (23).
4. Update the possibilistic term of the partition using Eq. (27).

5. Update the cluster prototypes using Eq. (30).

6. Repeat steps 3-5 until cluster prototypes stabilize.

7. Defuzzify the obtained partition if necessary using Eq. (31).

4 Evaluation

The proposed generalized FPCM method underwent a thorough evaluation process, which aimed to establish the behavior of the algorithm in comparison with its predecessors, mainly the FCM and the original FPCM. We did not expect to find the best clustering model that uses mixed partition. This is why we did not compare the performance of GFPCM with more sophisticated clustering models like PFCM or FPPPCM. So the main goal was to establish under what circumstances GFPCM provides fine partitions and to what extent it can eliminate the sensitivity to outlier data. Details of the evaluation are presented in the following.

4.1 Datasets

Three public datasets are involved in the evaluation process: the IRIS [8], WINE [1], and BreastCancer (Wisconsin) data [21]. The goal was to evaluate the proposed method in clustering problems with more and less dimensions as well. Details of the datasets are given in Table 1. These datasets are involved in clustering in their original format with values normalized in each dimension, and separately with an added outlier. In all cases the added outlier vector is represented as $(\delta, \delta, \dots, \delta)^\top$ in the normalized space, where $\delta > 1$ is a parameter that controls the position of the outlier. By varying the value of δ we can establish to what extent the clustering models can handle an outlier vector in the input data.

4.2 Evaluation criteria

We have chosen the following indicators used in the literature to evaluate the final partitions obtained by the algorithm: purity (abbreviated as PUR) [7], adjusted Rand index (abbreviated as ARI) [13], and normalized mutual information (abbreviated as NMI) [10].

In the context of cluster partition evaluation, purity can be defined as a measure of how well-defined and homogeneous the clusters are. It is a measure of the extent to which each cluster contains instances of only a single class.

Property	IRIS data	WINE data	BreastCancer data
Items (vectors)	150	178	569
Dimensions	4	13	30
Clusters	3	3	2
Cluster sizes	$50, 50, 50$	$59, 71, 48$	$357, 212$
Source	[2, 8]	[1]	[21]

Table 1: Datasets involved in the evaluation process, and their main properties.

The purity of a cluster partition is always in the unit range $[0, 1]$. A purity of 1 indicates that all clusters are well-defined and homogeneous, whereas a purity of 0 indicates that the clusters are completely mixed. Given some clusters M and some set of classes Y, purity is calculated using the following formula:

$$\text{PUR} = \frac{1}{n} \sum_{m \in M} \max_{y \in Y} |m \cap y| \ . \tag{32}$$

In simpler terms, for each cluster, the majority class of the cluster must be found and the number of data points belonging to that majority class must be summed. Finally, the total sum must be divided by the total number of data points (n).

However, this criterion has certain limitations. It does not perform well if the dataset is not balanced, i.e. the number of points belonging to the classes are different. In this case, the purity criterion favors the larger clusters, and as such, some data points from the smaller classes will also be assigned to the larger clusters. Because of this, in unbalanced datasets, a higher purity does not necessarily indicate that the clustering was successful. Therefore, purity may not reflect the true structure of the data in all cases. To alleviate the side effects of solely calculating purity on the cluster partitions of a potentially imbalanced dataset, other criteria must be used, such as the adjusted Rand index (ARI).

The adjusted Rand index is another widely used clustering evaluation criterion. It assesses the similarity of clustering outcomes. ARI is a suitable evaluation criterion for datasets with imbalanced cluster sizes. It takes unexpected cluster assignments into consideration, producing a result that is robust when faced with clusters with significantly different sizes.

The ARI value of a cluster partition is always in the range $[-1, 1]$. However, ARI values are mostly expected to be in the $[0, 1]$ range. An ARI value of 1

indicates a perfect match between the two measured cluster partitions. Otherwise, an ARI value of 0 indicates the baseline with respect to randomness. Negative ARI values represent a result that is worse than random clustering. As such, ARI by itself can also be used to compare two distinctly parameterized clustering results, making sure that any improvements in the clustering similarity are due to the better selection of parameters, rather than random fluctuations.

ARI can be computed with the help of a contingency table that encodes the pairwise relationship between two partitions. Let M and Y denote the two partitions such as $M = \{M_1, M_2, \dots, M_r\}$ and $Y = \{Y_1, Y_2, \dots, Y_s\}$. The contingency table, more precisely, the $r \times s$ table counts the pairs that are assigned to the same or different clusters in M and Y, i.e. each cell (n_{ij}) represents the number of data points that belong to both clustering partitions (for the intersection of M_i and Y_j would yield $n_{ij} = |M_i \cap Y_j|$). Naturally, the elements on the diagonal represent the number of data points that are assigned to the same cluster in both partitions. The other elements represent the remaining ones that are assigned to different clusters. Then the table is extended by one row and column that sum all the values in their respective row or column. Precomputing these sums enables easier computation.

Taking everything into consideration, the contingency table has the following structure:

	Y_1	Y_2	$\cdots$	Y_s	$\sum$
M_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
M_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
M_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
$\sum$	b_1	b_2	$\cdots$	b_s	

Then ARI is calculated as follows:

$$\mathrm{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}\ , \tag{33}$$

where n_{ij}, a_i, b_j are values taken from the contingency table.

Normalized mutual information is an evaluation criterion deeply rooted in information theory. It assumes that the more information is mutual between the two clustering outcomes the more valid the overall result is. It is commonly used because of its capability to assess partitions even in scenarios where the number of clusters varies. The values range from 0 to 1, where a higher NMI value indicates better agreement between the clustering assignments and the true class labels. A value of 0 indicates no mutual information, whereas a value of 1 indicates a perfect correlation. Altering the order or values of the class or cluster labels through permutation does not impact the NMI value. Let M and Y denote the clustering assignments, then NMI can be calculated as follows:

$$\mathrm{NMI}(Y, M) = \frac{2 \cdot \mathrm{I}(Y; M)}{H(Y) + H(M)} \ , \tag{34}$$

where $\mathrm{I}(Y; M)$ is the mutual information between Y and M, $H(Y)$ is the entropy of Y and similarly, $H(M)$ is the entropy of M.

There is another common formulation of the normalized mutual information which is more computational heavy than the aforementioned one:

$$\mathrm{NMI}(Y, M) = \frac{\mathrm{I}(Y; M)}{\sqrt{H(Y) \cdot H(M)}} \ . \tag{35}$$

Both formulations compute the same results and are valid representations of NMI. Furthermore, NMI is symmetric in the sense that Y and M is interchangeable, i.e. yielding the same result when switched.

These measures can assess the similarity between two clustering partitions, according an overall overview of the efficiency of clustering methods.

4.3 Tests using the IRIS dataset

Clustering algorithms are reported to work fine enough on IRIS data if the number of correct decisions reaches 133 out of 150, which corresponds to PUR=0.8867. When using the IRIS dataset without the addition of noise, we are interested in establishing those cases where GFPCM produces this outcome or better than that. It is also known about IRIS data, that the FCM algorithm produces clusters of better and better purity if the fuzzy exponent m rises, culminating at PUR=0.9333 (140 correct decisions out of 150), even though this pure partition is of low validity according to any cluster validity index (CVI) from the literature [23, 18].

Figure 1 exhibits the benchmarks of the GFPCM algorithm achieved on the IRIS dataset in case of no added outliers. The evolution of the benchmarks are all plotted against the fuzzy exponent m, and the behavior of the algorithm is investigated within a wide range of m. FPCM produces a high-purity partition on IRIS, which is influenced even by a very weak possibilistic term ($\beta = 1$). The

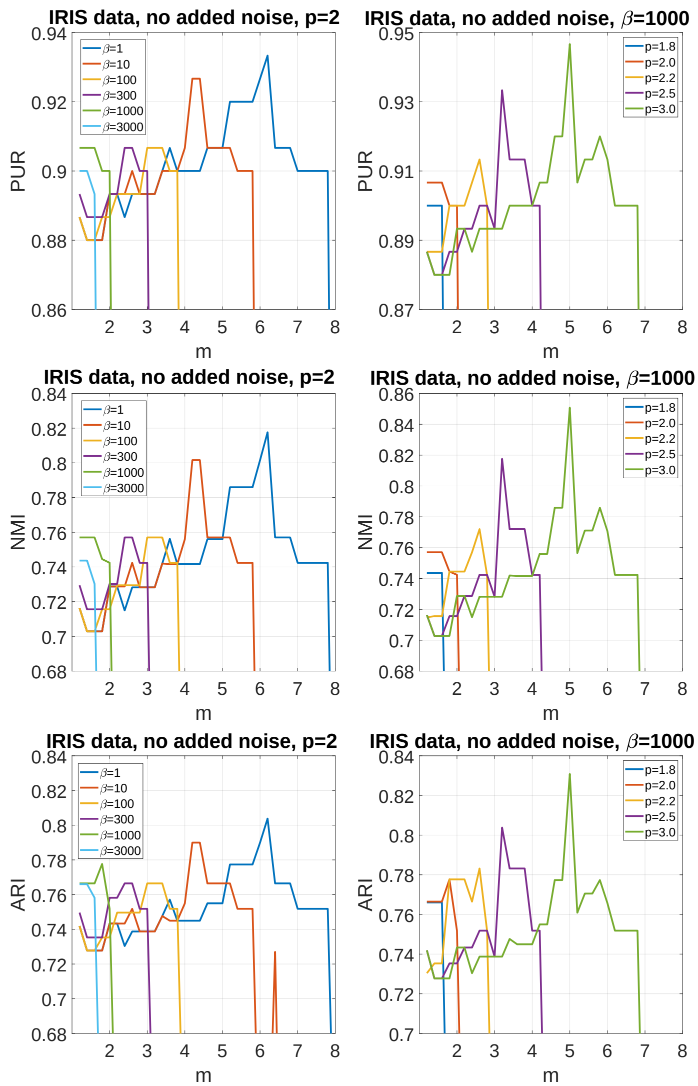

Figure 1: GFPCM benchmarks obtained on the IRIS dataset in case of no added noise. Graph representations in the left column show PUR, NMI and ARI values, respectively, all plotted against fuzzy exponent m, obtained with various values of trade-off parameter β, while possibilistic exponent was fixed at $p = 2$. Graphs in the right column represent PUR, NMI and ARI values plotted against m, obtained at fixed trade-off $\beta = 1000$, and selected values of exponent p.

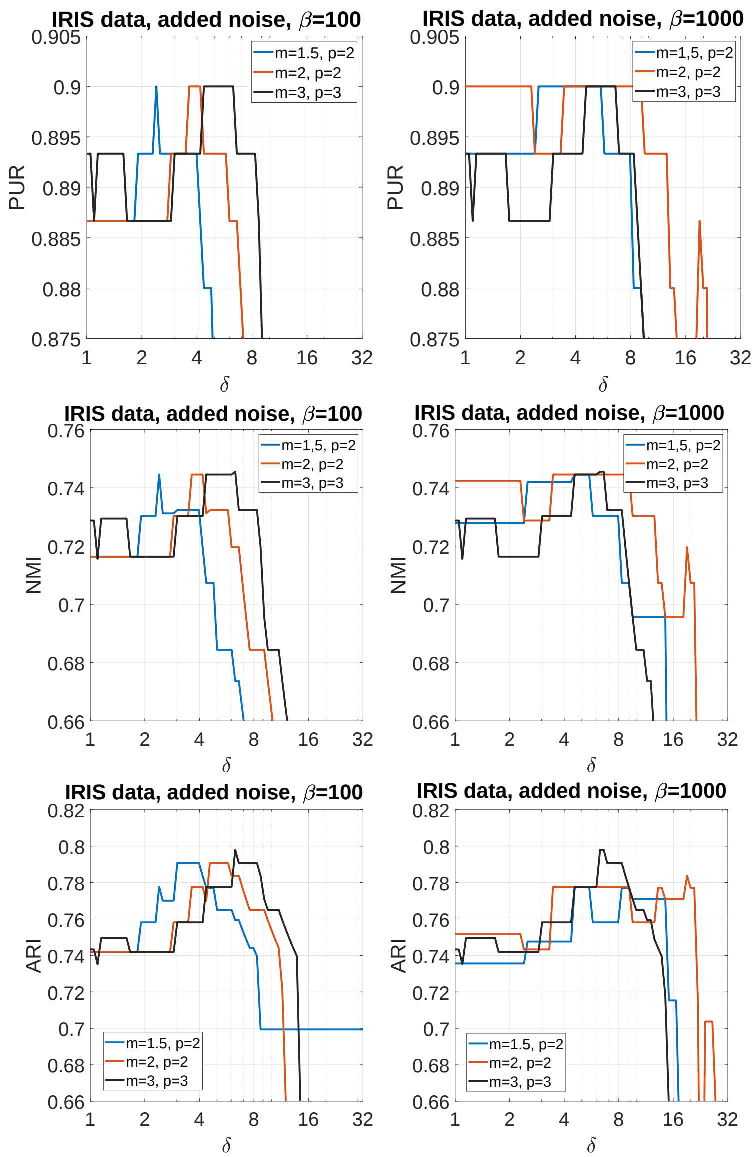

Figure 2: GFPCM benchmarks obtained on the IRIS dataset in case of an outlier added at $(\delta, \delta, \delta, \delta)^T$. A comparison of the cases with trade-of parameter $\beta = 100$ and $\beta = 1000$ is presented, where the former works quite the same as FCM ($\beta = 0$) and FPCM ($\beta = 1$).

higher the value of β, the more restricted becomes the domain of acceptable partitions. However, let us clarify that this phenomenon is not a problem, because the recommended range of the exponent m hardly exceeds the value of 3 [20]. What we can see from the results is that it is not recommendable to use a very strong possibilistic component. This criterion restricts us to set the possibilistic parameters $p \geq 2$, and $\beta < 1000$. On the other hand, it is also visible that a high value of p (e.g. $p = 3$ is already high) weakens the effect of the possibilistic term within GFPCM.

Another thing that deserves to be remarked here: so far we did not see any reported case where any c-means algorithm provided 142 correct decision on the IRIS dataset. Figure 1 shows us such an example: the GFPCM algorithm used at $m = 5$, $p = 3$, and trade-off set to $\beta = 1000$ produced this outstanding PUR benchmark. This experience convinced us that a weak possibilistic term added to the objective function of the FCM algorithm can cause significant alterations in its behavior, even if it is not visible in every scenario.

Figure 2 presents the benchmarks of GFPCM, obtained on the IRIS dataset, with an added outlier whose position depends on parameter δ. The goal is to establish how far the outlier needs to stand to ruin the final partition. Conversely, we may ask what settings are needed for the GFPCM to assure a fine partition even in case of very distant outlier? Figure 2 relates on the examples of $\beta = 100$ (left column) and $\beta = 1000$ (right column). We experienced no intensive change in the behavior of GFPCM while varying the trade-off parameter between 0 and 100. However, as the possibilistic terms is becoming stronger while raising β further, the algorithm demonstrates an enhanced capability to accommodate outliers that are increasingly distant. The limit value of δ still tolerated by GFPCM in various circumstances is studied in Section 4.6.

4.4 Tests using the WINE dataset

WINE dataset contains vectors in a normalized 13-dimensional space, which are organized in unequal groups. The FCM algorithm can produce its partitioning with PUR ≈ 0.95 at reasonably low values of the exponent ($m \leq 2$), but this benchmark strongly drops if we increase the exponent. Figure 3 exhibits the behavior of the GFPCM algorithm when applied to the WINE dataset under various circumstances.

The original FPCM algorithm ($\beta = 1$) seems to perform the same as FCM ($\beta = 0$). However, if we increase further the value of β, GFPCM tends to extend the domain where it can provide acceptable partitions to higher value

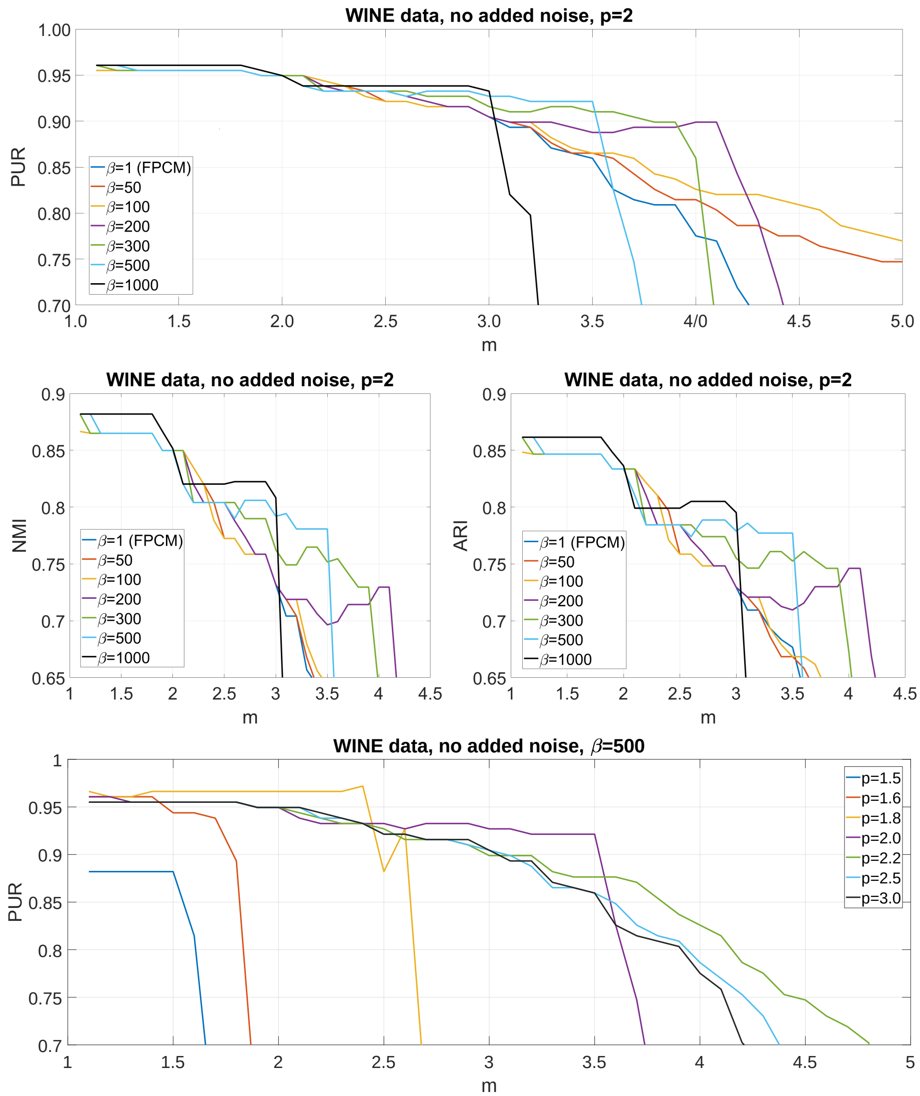

Figure 3: GFPCM benchmarks obtained on the WINE dataset in case of no added noise. Different curves relate on cases with various trade-off values of β at fixed $p = 2$, or at various values of p at fixed $\beta = 500$. Curves indicate up to which value of the fuzzy exponent m we have a stable solution in each scenario.

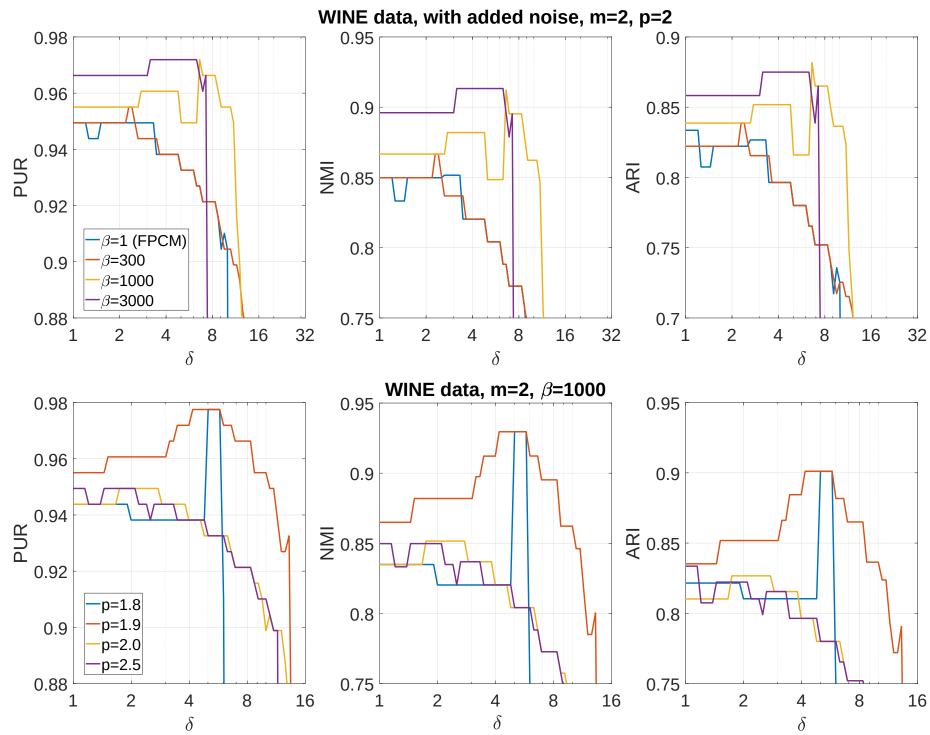

Figure 4: GFPCM benchmarks obtained on the WINE dataset, in case of an outlier added at $(\delta, \delta, \ldots, \delta)^T$. The behavior of the GFPCM algorithm can be observed for scenarios of various trade-off values β at fixed $p = 2$ (upper row), and at various possibilistic exponent values p at fixed trade-off $\beta = 1000$. In all these tests, fuzzy exponent was set to $m = 2$.

of fuzzy exponent m. However, this effect saturates around $\beta = 100$. Above this value we can see that GFPCM provides finer partition than FCM or FPCM up to a certain limit of m, beyond which there is an abrupt drop in partition quality. The limit value of m seems to be in inverse proportion with trade-off value β. On the other hand, if we fix the trade-off parameter at a reasonably high value (e.g., in our case, $\beta = 500$), we can study the effect of various possibilistic exponents p upon the behavior of the algorithm. The possibilistic effect becomes stronger as p approaches 1. Experiments showed that reducing p far below 2 damages the clustering outcome. The best partitions in this case were obtained at $m \in [2.0, 2.5]$ and $p = 1.8$.

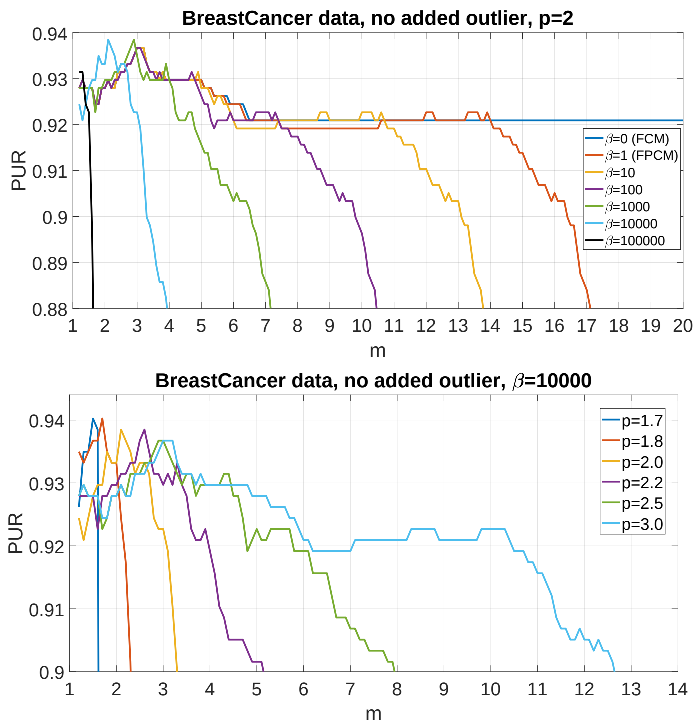

Figure 5: Purity benchmarks obtained on the BreastCancer dataset, with no added outliers. PUR is plotted against fuzzy exponent m, at fixed $p = 2$ and various trade-off values β (upper panel), and at fixed trade-off parameter $\beta = 10000$ and various possibilistic exponents p.

Figure 4 exhibits the behavior of the proposed clustering model in various scenarios, when applied to the WINE dataset with an added outlier whose position is controlled by the parameter δ. What we can observe is that there are settings which can extend the limit value of δ up to which we obtain a fine partitioning (e.g., $p = 1.9$ and $\beta = 1000$), while there are other settings which improve the purity of the obtained partition compared to FCM or FPCM if the outlier is very distant (e.g., PUR > 0.97 at $m = p = 2$ and $\beta = 3000$). Many of the phenomena are similar to the ones observed in IRIS data tests, but the best choice of β strongly depends on the data.

4.5 Tests using the BreastCancer dataset

The BreastCancer dataset presents vector data in a multi-dimensional setting. Each dimension was normalized before feeding the data to the clustering algorithms. Having only two clusters, we found it unnecessary to report NMI and ARI benchmarks, PUR contains all relevant information on the obtained partitions.

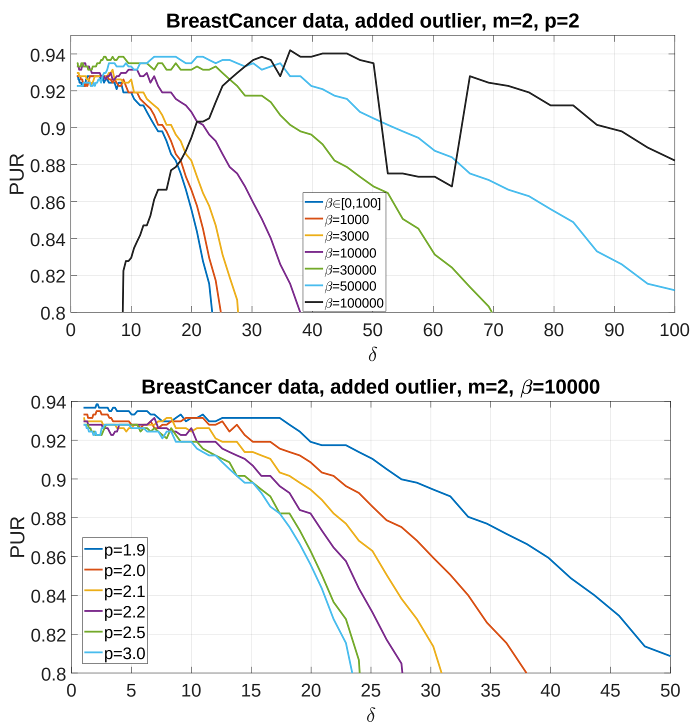

Figure 6: Purity benchmarks obtained on the BreastCancer dataset, with outlier added at $(\delta, \delta, \ldots, \delta)^T$ according to parameter $\delta > 1$. Graphs indicate how the position of the outlier influences the final partition produced by GFPCM, for various scenarios regarding parameters p and β. In all cases, fuzzy exponent was fixed at $m = 2$.

Figure 5 exhibits the clustering outcome of GFPCM at various settings, when applied to the BreastCancer dataset with no added outlier. The result of FCM obtained at $\beta = 0$ presents acceptable quality at any reasonable value of fuzzy exponent m, while the slightly modified version FPCM ($\beta = 1$) already sets up a limit value for m above which we do not obtain fine partitioning. High PUR values are achieved at fuzzy exponents $m < 4$, especially when using trade-off parameter value $\beta \in [1000, 10000]$. If we investigate the effect of different possibilistic exponents upon the clustering outcome, the most convincing benchmarks are obtained at not much lower and not much higher than $p = 2$. Again, we need to mention that the reasonable and most frequented range of m is using values below 3.

Figure 6 presents how the parameter settings affect the clustering result in case of an added outlier, for various scenarios and outlier positions. When both exponents are fixed at $m = p = 2$, raising the trade-off parameter value extends the tolerance range of the outlier up to a certain extent. At $\beta < 100$ hardly any difference is visible between the behavior of GFPCM and FCM. Through changing the trade-off value up to $\beta = 50000$, GFPCM tends to tolerate the presence of an outlier at increasingly distant positions. However, at $\beta = 100000$ or higher, the algorithm no more produces fine partitions. If we fix $m = 2$ and $\beta = 10000$, and vary the possibilistic exponent value, we obtain similar phenomena to other datasets. GFPCM works best in the proximity of $p = 2$ or slightly below that, where it can provide partitions of better purity than FCM or FPCM. Larger values of p bring the performance of the algorithm close to FCM, which does not come as a surprise as with these settings we are weakening the possibilistic term in the objective function.

4.6 The limits of outlier tolerance

In case of all three datasets, we attempted to identify the maximum distance of the outlier defined by parameter δ, which is tolerated by the GFPCM algorithm without damaging the partition quality. Let us denote the limit value of δ by δ_{max}, and investigate how this value depends on the chosen dataset and the settings of the other three parameters m, p, and β. Further on, we denote by δ_{FCM} the maximum value tolerated by the FCM algorithm under the same circumstances (same m, but $\beta = 0$ and p irrelevant) where δ_{max} was established. The final partition was called acceptable if the PUR benchmark exceeded 0.88, 0.93, and 0.9 for the IRIS, WINE and BreastCancer datasets, respectively. These thresholds were established empirically.

A detailed summary of the obtained δ_{max} values is exhibited in Figure 7 and Table 2. Figure 7 shows us how the tolerated limit distance varies with trade-off parameter β when using various datasets and various settings for the fuzzy exponent, while the possibilistic exponent is fixed at $p = 2$. This is the main result of this study, as FPCM and GFPCM was meant to be an extension of FCM to improve the way it handles outliers. A general thing that we can see in all these graphs is that the behavior of the GFPCM algorithm hardly changes below $\beta < 100$. Consequently, and not at all surprisingly, FCM ($\beta = 0$) and FPCM ($\beta = 1$) hardly manifest any visible difference.

However, if we raise the value of the trade-off parameter to a reasonable level, we may obtain a considerable extension of the tolerated noise range. When using the algorithm at low value of the fuzzy exponent, (e.g., $m = 1.5$), the ratio $\delta_{max}/\delta_{\mathrm{FCM}}$ can be as high as 10. For higher values of the fuzzy exponent, the extension is approximately twofold. As an exception, in case of WINE dataset we do not achieve any improvement at $m > 2$.

Further on, we also need to remark that the best performance by GFPCM on various datasets is achieved at different values of the trade-off parameter β. This did not come as a surprise either, since β needs to compensate the disequilibrium caused by the difference between $\sum_i \sum_k u_{ik}^m$ and $\sum_i \sum_k t_{ik}^p$ within the objective function given in Eq. (18).

Table 2 presents a matrix of δ_{max} and their corresponding δ_{FCM} values, obtained at various settings of the two exponents m and p, and indicating the optimal β_{opt} trade-off value with which they were achieved. In this table, cases labeled as "not improving" mean that GFPCM does not bring any favorable change compared to FCM or FPCM, while "unstable" means that under those circumstances none of the FCM, FPCM or GFPCM can produce fine clustering outcome. This table suggests that using a possibilistic exponent in the proximity of $p = 2$ can significantly extend the tolerated distance of the outlier, with the condition that the necessary trade-off value is properly approximated.

5 Discussion

The main goal of this study was to eliminate some limitations of the FPCM algorithm, the behavior of which in its initial formulation strongly depended on the difference between the number input data vectors n and the number of clusters c. Whenever $n >> c$, the presence of the possibilistic part in the mixed partition is hardly observable. In our consideration, FPCM deserved an improvement because of the way it defined the possibilistic part of the mixture. It did not follow the conventional way indicated by the PCM algorithm [9] using the possibilistic penalty terms η_i $(i = 1, \ldots, c)$. Instead of that, the typicality values represented by fuzzy membership functions t_{ik} $(i = 1, \ldots, c;$

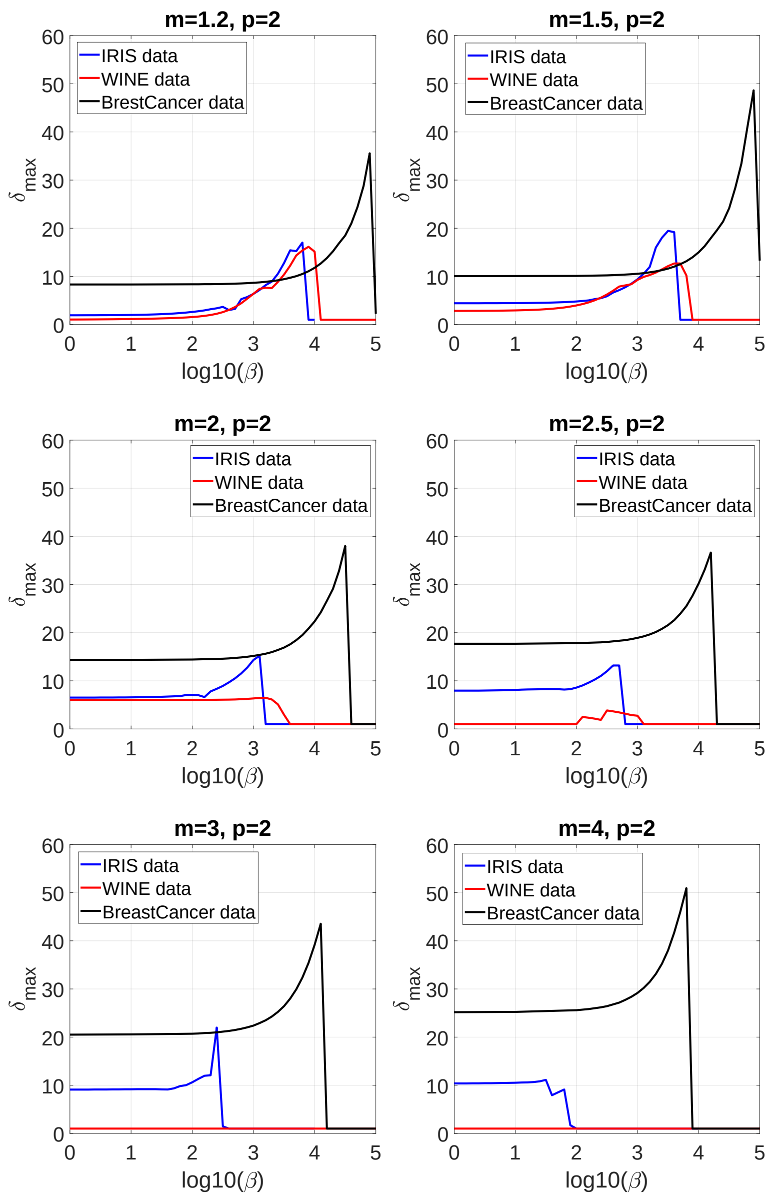

Figure 7: The evolution of the limit distance δ_{max} of the outlier plotted against the trade-off parameter β represented on a logarithmic scale, in case of various datasets and parameter settings.

Params		IRIS data			WINE data			BreastCancer data		
m	p	δ_{FCM}	δ_{max}	β_{opt}	δ_{FCM}	δ_{max}	β_{opt}	δ_{FCM}	δ_{max}	β_{opt}
1.2	1.8	1.91	16.14	1995	1.02	14.42	1995	8.32	24.10	15849
1.5	1.8	4.41	18.66	1000	2.82	12.85	1585	10.05	38.46	19953
2	1.8	6.50	22.91	501	6.04	9.02	501	14.35	37.76	10000
2.5	1.8	7.94	22.39	200	1.00	4.08	63	17.70	35.73	5012
3	1.8	9.10	16.14	100		unstable		20.51	38.37	3162
4	1.8	10.35	12.16	16		unstable		25.18	45.39	1585
1.2	2	1.91	17.02	6310	1.02	16.14	7943	8.32	35.56	79433
1.5	2	4.41	19.45	3162	2.82	12.68	3981	10.05	48.64	79433
2	2	6.50	15.21	1259	6.04	6.47	1585	14.35	38.02	31623
2.5	2	7.94	13.18	398	1.00	3.85	316	17.70	36.64	15849
3	2	9.10	21.98	251		unstable		20.51	43.55	12589
4	2	10.35	11.12	32		unstable		25.18	50.93	6310
1.2	2.2	1.91	14.03	10000	1.02	12.59	19953	8.32	18.84	100000
1.5	2.2	4.41	20.51	10000	2.82	10.16	12589	10.05	24.27	100000
2	2.2	6.50	17.26	1995	6.04	6.34	3981	14.35	37.07	100000
2.5	2.2	7.94	22.8	1585	1.00	3.48	1585	17.70	40.83	63096
3	2.2	9.10	22.28	794		unstable		20.51	42.46	39811
4	2.2	10.35	19.41	158		unstable		25.18	49.43	19953
1.2	2.5	1.91	7.13	10000	1.02	8.87	100000	8.32	10.00	100000
1.5	2.5	4.41	13.12	10000	2.82	8.83	79433	10.05	12.39	100000
2	2.5	6.50	16.14	10000	6.04	6.30	15849	14.35	18.03	100000
2.5	2.5	7.94	23.17	7943		unstable		17.70	23.17	100000
3	2.5	9.10	22.49	3981		unstable		20.51	28.97	100000
4	2.5	10.35	12.50	631		unstable		25.18	43.25	100000
1.2	3	1.91	2.40	10000	1.02	3.85	100000	8.32	not improving	
1.5	3	4.41	4.76	10000	2.82	6.98	100000	10.05	not improving	
2	3	6.50	7.06	10000	6.04	6.28	100000	14.35	not improving	
2.5	3	7.94	8.75	10000		unstable		17.70	not improving	
3	3	9.10	9.82	10000		unstable		20.51	not improving	
4	3	10.35	15.92	10000		unstable		25.18	not improving	

Table 2: The limit position of the outlier (δ_{max}) in case of various values of exponents m and p, and the value of trade-off parameter β_{opt} with which it is achieved.

$k = 1, \ldots, n$) were constrained probabilistically such a way, that they sum up to 1 with respect to each cluster.

The proposed modification in the objective function of FPCM, namely the introduction of trade-off parameter β enabled us to raise the strength of the possibilistic part of the mixed partition. The proposed clustering model (GF-PCM) can be considered a generalization of FPCM, since FPCM is equivalent with the special case defined by $\beta = 1$, and FCM is obtained if $\beta = 0$ – the value of p is irrelevant in this case. Any other positive values of the trade-off parameter β lead to different partition mixtures, and consequently to different clustering algorithms.

The proposed clustering model uses three parameters, one more than FPCM. These are the fuzzy exponent m, the possibilistic exponent p, and the trade-off parameter β. To set the appropriate value of m, we may use the same criteria as we would use for FCM. For the general case, without knowing the properties of the input data, it is recommendable to keep m in the proximity of 1. There are several papers discussing the choice of this parameter, e.g., [4, 20, 22]. The experimental part of this study provided us enough evidence that the possibilistic exponent p should be chosen in the interval $p \in [1.8, 2.0]$. Lower values than that did not lead to convincing results in any of the circumstances. Higher values make the possibilistic part too weak, making the compensatory effect of GFPCM negligible.

For the trade-off parameter β, the ideal value seems to be proportional with $(n/c)^2$, but this remark needs further investigation. From the shape of the curves exhibited in Figure 7 we can easily realize that a careful prediction is needed for the choice of β, to place it below β_{opt}, but not very much below it. To provide a reliable approximation formula, a deeper investigation is needed, using several more datasets and experiments with multiple outliers as well. This is going to be the topic of a future study.

One of the relevant limitations of this study is the fact that we only tested the effect of a single outlier vector. Handling multiple outliers would have meant a lot more test cases, whose evaluation details hardly fit within the frame of such a study.

6 Conclusion

In this paper we proposed a generalization of the so-called fuzzy-possibilistic c-means algorithm, which in its original formulation had a strong limitation in the strength of the possibilistic part of the mixed partition. With the introduction of a trade-off parameter we were able to amplify the phenomenon caused by the possibilistic extension of the fuzzy c-means objective function. The proposed clustering method was evaluated using three public datasets that contain real-life data. The proposed clustering model is capable to better handle datasets containing outlier data than its predecessors, namely the fuzzy c-means and the fuzzy-possibilistic c-means clustering algorithms.

Acknowledgements

This study was supported in part by the Collegium Talentum 2023 Programme of Hungary and the Consolidator Researcher Program of Óbuda University.

References

[1] S. Aeberhard, M. Floina, Wine, *UCI Machine Learning Repository* (1991) ⇒ 414, 415

[2] R. Andersen, Irises of the Gaspe Peninsula, *Bull. Amer, Iris Soc.* **59** (1935) 2–5. ⇒ 415

[3] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum, New York (1981) ⇒ 405

[4] H. Choe, J.B. Jordan, On the optimal choice of parameters in a fuzzy c-means algorithm, *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, 1992, pp. 349–354. ⇒ 429

[5] R.N. Davé, Characterization and detection of noise in clustering. *Pattern Recognition Letters*, **12**, 11 (1991) 657–664. ⇒ 405

[6] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters, *J. Cybern.* **3**, 3 (1974) 32–57. ⇒ 405

[7] B. Everitt, *Cluster analysis*, Chichester, West Sussex, U.K. (2011) ⇒ 414

[8] R.A. Fisher, Iris, *UCI Machine Learning Repository* (1988) ⇒ 414, 415

[9] R. Krishnapuram, J.M. Keller, A possibilistic approach to clustering, *IEEE Transactions on Fuzzy Systems* **1**, 2 (1993) 98–110. ⇒ 405, 408, 426

[10] T. O. Kvålseth, On normalized mutual information: Measure derivations and properties, *Entropy* **19**, 11 (2017) 613–114. ⇒ 414

[11] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A mixed c-means clustering model, *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, 1997, pp. 11–21. ⇒ 405, 410

[12] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, *IEEE Transactions on Fuzzy Systems* **13**, 4 (2005) 517–530. ⇒ 406, 410

[13] W. Rand, Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association* **66**, 336 (1971) 846–850. ⇒ 414

[14] T.A. Runkler, A Convergence Study of the Possibilistic One Means Algorithm, *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, 2023, art. no. 10309756, pp. 1–6. ⇒413

[15] E,H. Ruspini, A new approach to clustering *Information and Control* **15**, 1 (1969) 22–32. ⇒405

[16] L. Szilágyi, Fuzzy-Possibilistic Product Partition: a novel robust approach to c-means clustering, *Lect. Notes Comput Sci.* **6820** (2011) 150–161. ⇒406, 410

[17] L. Szilágyi, Robust spherical shell clustering using fuzzy-possibilistic product partition *Int. J. Intell. Syst.* **28**, 6 (2013) 524–539. ⇒410

[18] L. Szilágyi, S. M. Szilágyi, Generalization rules for the suppressed fuzzy c-means clustering algorithm, *Neurocomput.* **139** (2014) 298–309. ⇒417

[19] L. Szilágyi, Zs.R. Varga, S.M. Szilágyi, Application of the fuzzy-possibilistic product partition in elliptic shell clustering, *Lect. Notes Comput Sci.* **8825** (2014) 158–169. ⇒410

[20] V. Torra, On the selection of m for Fuzzy c-Means, *Conference of the International Fuzzy Systems Association*, 2015, pp. 1571–1577. ⇒420, 429

[21] W. Wolberg, O. Mangasarian, N. Street, W. Street, Breast Cancer Wisconsin (Diagnostic), *UCI Machine Learning Repository* (1995) ⇒414, 415

[22] K. L. Wu, Analysis of parameter selections for fuzzy c-means, *Pattern Recognition* **45**, 1 (2012) 407–415. ⇒429

[23] X. L. Xie, G. A. Beni, A validity measure for fuzzy clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, 8 (1991) 841–847. ⇒417

[24] L. Zadeh, Fuzzy sets, *Information and Control* **8**, (1965) 338–353. ⇒405

Received: 15 November 2023 • Revised: December 6, 2023