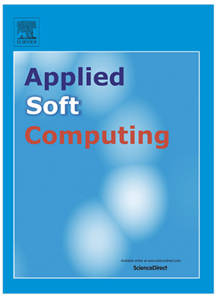
Journal Pre-proof



Balancing the user-driven feature selection and their incidence in the clustering structure formation

Ferdinando Di Martino, Sabrina Senatore

PII: S1568-4946(20)30792-4
DOI: https://doi.org/10.1016/j.asoc.2020.106854
Reference: ASOC 106854

To appear in: *Applied Soft Computing Journal*

Received date : 1 August 2020
Revised date : 8 October 2020
Accepted date : 23 October 2020

Please cite this article as: F. Di Martino and S. Senatore, Balancing the user-driven feature selection and their incidence in the clustering structure formation, *Applied Soft Computing Journal* (2020), doi: https://doi.org/10.1016/j.asoc.2020.106854.

Balancing the user-driven feature selection and their incidence in the clustering structure formation

Ferdinando Di Martino[1], Sabrina Senatore[2]

[1] Università degli Studi di Napoli Federico II, Dipartimento di Architettura,
Via Toledo 402, 80134 Napoli, Italy
email: fdimarti@unina.it
[2] Università degli Studi di Salerno, Dipartimento di Ingegneria dell'Informazione ed Elettrica
e Matematica Applicata,
Via Giovanni Paolo II, 132, 84084 Fisciano (Salerno), Italy
email: ssenatore@unisa.it

Abstract. The feature selection represents a key step in mining high-dimensional data: the significance of features in maintaining the data structure while ignoring the feature redundancy is crucial to improve the final performance of classification methods. At the same time, an accurate understanding of feature domains may need human intervention to balance the importance of structure-based features with those one dictated by human expertise.
To address this issue, this work introduces a human-driven feature selection method for data clustering. The algorithm, called Feature Selection EFCM (FS-EFCM in short), aims at supporting the relevance of some features from the domain of interest, but preserving their incidence in the natural clustering structure. The relevance and incidence of each feature are measure assessed during the FS-EFCM execution, in order to find a balance between the human suggestions about the feature importance and the by feature incidence in the natural cluster-based data distribution. Experimental results and comparisons highlight how the algorithm is robust in the presence of not very significant features, and the classification performance shows the effectiveness of the proposed feature selection method compared with the well-known feature selection algorithms.

Keywords: semi-supervised clustering, FCM, EFCM, Feature selection, Feature Selection EFCM, user-driven feature relevance, feature incidence.

1 Introduction

In high dimensional data classification, feature selection represents a critical aspect to investigate cautiously. Selecting the most informative features while preserving high classification accuracy in representing the original dataset features is a complex task that has been studied extensively in the literature through a wide range of possible formal methods and concrete applications. Data dimensionality is one of the main problems and can hinder the efficiency of the classification process, as well as yield computational burden and compromise model generalization.

Finding relevant features can improve the data quality, and the discrimination capability, as well as discarding irrelevant features, especially redundant features can help the efficacy of the classification process.

Many feature selection methods defined in the literature range from filters based on distinct metrics (e.g., entropy, probability distributions or information theory) to embedded and wrapper methods using different induction algorithms [21, 22]; anyway, no general methodology has been defined for providing intelligent selection modeling. Search techniques such as random search, breadth search, or hybrid are also used to find the proper subset of features. Recent trends converge toward metaheuristic algorithms which are strongly inspired to the collective behavior of organisms: they are valid alternatives to exhaustive search that requires high computational cost and are often time-consuming. However, the metaheuristic algorithms often suffer from local optimum, lack of diversity of the search, and imbalance between the exploration and exploitation ratios of these algorithms [23].

Most of the feature selection approaches seem to exploit intrinsic measures (e.g., the informative content, or empirical errors) that are mainly relied on supervised class information. In clustering algorithms, feature redundancy and relevance are measured with respect to the natural data grouping (clusters instead of classes).

Feature selection models for clustering mainly are supported by heuristic criteria for estimating the resulting cluster quality, such as cluster compactness, scatter separability, and maximum likelihood. Wrapper models for clustering for instance, search for the feature subset that meets these criteria, often exploiting sequential search [26]. Our clustering approach suits a filter model evaluating *feature incidence* w.r.t. how much the feature helps in separating clusters and, at the same time, tries to meet the feature relevance weights provided as additional external expertise.

The nature of data as well as the knowledge about the data domain can hinder peculiarities that are not evident by plain analysis of the domain or by applying traditional statistical measures. Selecting appropriate "relevant" features needs the understanding of the feature domain, and this activity is often intended for human experts, that must know the domain as well as technical details of algorithms, to interpret the classification result and the learning power of the algorithms [21].

The human-driven feature selection can lead the clustering algorithm enhancement in discriminating redundant from relevant features. The expert viewpoint is crucial to understand the application context and role of different features effectively, especially in specialistic reference domains.

Feature selection methods have been employed in many application domains widely, ranging from the bioinformatics [18], medicine [10], to text mining [5] and even in industrial environments where the parameter tuning is essential for the stability analysis of real-time dynamic systems [27]. In particular, the role of feature selection becomes especially evident in the processing of textual information, such as papers, web sites, reviews, twitters, or snippets. The expressive power of natural language accentuates the difficulty to discriminate appropriate features that support the classification methods, accurately. Selecting the highest-ranked terms, according to the most traditional Information Retrieval techniques, does not guarantee to get the most actual relevant features, especially due to the polysemy and synonymy concerns that can affect on the classification result. The semantics behind the words could be very tricky to capture and automatic approaches fail in some circumstances, especially when feeling or metaphors are used. For example, a figurative expression, such as “bad egg” could be literally interpreted by an automatic system that is not trained to recognize the different word senses. To address these issues, many approaches try to mine the latent semantics [24] in the data space, to overcomes the drawback due to polysemy, synonyms, phrases dependencies and metaphors.

The role of humans in supporting the feature selection can be effective, when the NLP techniques cannot solve the intrinsic ambiguities of the natural languages. Depending on data domain, the expert-suggested hints can help the algorithm to discriminate the features that are representative of the domain and improve the classification process performance. In this context, expert systems provide an adequate model for collecting expert level knowledge and guarantee semantic data integration for facing symbolic representation of data [28] and natural language issues.

This paper presents a human-driven feature selection method for data classification, called FS-EFCM. The method acquires a relevance score associated by the experts with each feature: the score plays a discriminating role in the selection of the features that are considered, by the domain experts, crucial to describe the domain of interest.

The FS-EFCM assesses the relevance of the features also in the natural clustering process, considering their individual incidence in the cluster structuring.

Feature relevance and incidence measures are evaluated during the FS-EFCM execution to find a good balance between the human suggestions (in form of feature scores) and the natural cluster-based data distribution revealed by feature incidence. During clustering executing indeed, the two measures “duels” at each iteration raising up the features that are naturally relevant in the clustering, in the light of the expert-based suggestions. The algorithm aims at reaching a trade-off among these two measures, these two forces from a different nature that lead to the identification of the actual relevant features and finally guarantee good clustering performance.

The paper is organized as follows: Section 2 presents the main related work on features extraction and selection, with a focus on the FCM-based approaches; Section 3 introduces the novel FS-EFCM algorithm, after a brief formal background on EFCM. Experiments on UCI Machine Learning datasets are presented in Section 4; performance metrics show the effectiveness of our method in the filtering useless features and classification results. Finally, the main conclusions are presented in the last section.

2 Related Work

Feature selection is a well-known task for discriminating relevant features by removing unnecessary data whereas trying to preserve a good classification accuracy.

In high dimensional data input, it becomes crucial to discover the best subset of informative features that preserving the informativeness of the original dataset features while speeds up learning. Feature selection is a challenging and computationally expensive process: massive dimensionality hides a high level of noise, with irrelevant and redundant features.

Traditional data mining and machine learning methods could not be suitable to control high dimensional feature space. Their performances are often degraded by the inaccurate feature selection, achieved by automatic techniques.

To reduce the negative impact of irrelevant and redundant features, many feature selection algorithms have been developed in literature [25]. Feature selection is often studied as an optimization problem, aimed at finding the (near) optimal subset, but carrying out exhaustive search strategies is quite impractical in this area, especially considering large feature spaces [29].

Depending on the learning method, feature selection is typically divided into two major approaches: *wrapper* which exploits learning methods to evaluate better subsets of feature set; *filter* which relies on main characteristics of the data to evaluate feature subset and are independent of the induction algorithm.

Wrappers have been widely investigated for classification accuracy [30] and optimization techniques to find the best combination of features [31], including evolutionary algorithms or particle swarm optimization.

Stochastic methods and in particular metaheuristic methods are also adopted for selecting the optimal feature set in feature selection: they work better on the local optima problems compared with more conventional optimization algorithms [32]. In particular, PSO algorithms have been widely adopted to select the most important features to accomplish a particular machine learning task.

The PCA (Principal Component Analysis) model, applied to reduce the dimensionality of the data, operates a linear transformation of the variables projecting the data in a new Cartesian system. However, this approach is not useful in classification methods, because the choice of the features projected with the highest variances is not necessarily related to classification performances.

Variations of the PCA technique such as the LDA (Linear Discriminant Analysis) method [8, 9] find a linear combination of features that maximizes the separation of the classes. The resulting feature combination may be used for a linear classifier or, more commonly, for dimensionality reduction, before applying the next classification. When applied to high dimensional data, the LDA is subject to the well-known curse of dimensionality problem, since the need of increasing the amount of data (for supporting the result) often grows exponentially with the dimensionality. Due to the peaking phenomenon [16], the classifier performance decreases as the number of features increases.

2.1 Fuzzy C-Means-driven feature selection

Some works propose soft computing features reduction algorithms based on Fuzzy C-Means (FCM) algorithm. In FCM clustering, as stated in [17], the nature of the attributes does not affect the cluster centers. However, the cluster centers can be used to choose the attributes that can be used to distinguish between similar/dissimilar points. Let *n* be the number of features and *C* the number of clusters, the cluster center $\mathbf{v}_i = \{v_{i1}, \ldots, v_{in}\}$, i = 1,2,…,C. In [17], the relevant features are those contributing to form clusters, whose centers are very distant from each other. For each feature h = 1, 2, …, n, the authors calculate the minimum of the distances $d_{ikh} = |v_{ih} - v_{kh}|$ between the values assumed by h^th-components of the i^th and of the k^th cluster centers, with i, k = 1,…,C and i≠k. The most relevant features are the features whose value d_{ikh} is the greatest.

In [13], an FCM feature selection method is proposed by adopting the gradient method to minimize the fuzzy objective function by the Kullback-Leibler divergence information measure. The Fisher's Ratio index is used in [14], to select the most informative features on a cancer classification dataset; the authors show that the classification improves performance when the FCM runs on the selected features than the original feature set. Enhanced fuzzy models [37] allow simulating the nonlinear finger dynamics of the human hand for the myoelectric (ME)-based control of a prosthetic hand.

In literature, some hybrid approaches integrate FCM with heuristic optimization algorithms to find the optimal subset of features. A feature selection method, based on FCM and genetic algorithm, is applied in the intrusion selection classification problem [20]. In [14], feature selection is achieved by combining a supervised FCM algorithm and an Ant Colony optimization.

In [36] a classifier for multi-relational data is designed to get a fused feature sets by exploiting correlated information from entity-relationship relations.

An extension of the FCM algorithm called Extended FCM (EFCM) algorithm [7] is applied from social message streams to classify sentiments [6]: the algorithm allows to capture relevant emotions from the text and describe them as a combination of blurred sentiments and moods, reflecting the actual nature of human feelings. Several variations of EFCM have been proposed in the literature: some are applied in practical context [33][34]. In [35] the EFCM algorithm has been used as a hotspot detection method for very large datasets of events.

The EFCM algorithm improves the performance of FCM in terms of robustness to noise and outliers, independence from initialization, and the ability to find the optimal number of clusters by applying a cluster fusion heuristic process.

3 The feature selection algorithm

The proposed FS-EFCM method consists of an iterative process, exploiting the Extended Fuzzy C-Means algorithm [7]. EFCM is a partitive fuzzy clustering algorithm that extends the traditional FCM algorithm [2, 3], overcoming its drawbacks, such as the choice a priori of the number of clusters and the sensibility to the presence of noise and outliers. Moreover, EFCM is more robust to the partition initialization than FCM and does not require to validate the produced clustering partitioning over several random initializations. A merging process between the two most similar clusters during each iteration allows obtaining the optimal number of clusters. A brief overview of EFCM is given in the next section.

3.1 EFCM: some preliminary notions

Let X={**x**$_1$, ..., **x**$_N$}⊂R^n be a set of N data points (the patterns) in the n-dimensional space R^n and **V** = {**v**$_1$,…,**v**$_C$} ⊂ R^n be the set of centers of the C clusters. Let **U** be the C×N partition matrix where u$_{ij}$ is the membership degree of the j*th* pattern **x**$_j$ to the ith cluster **v**$_i$**.** In the EFCM algorithm, the ith cluster prototype is given by a hypersphere with center **v**$_i$ and radius r$_i$. The EFCM clustering algorithm minimizes the following objective function:

$$J_E(\mathbf{U},\mathbf{V},\mathbf{r}) = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m}\left(d_{ij}^{2} - r_{i}^{2}\right) \qquad (1)$$

where **r** = {r$_1$,…,r$_C$} is the set of the radius, m is the fuzzifier parameter and d$_{ij}$ is the distance between the i*th* cluster and the j*th* pattern.
The radius r$_i$ is calculated by considering the covariance of the i*th* cluster;

$$\mathbf{P_i} = \frac{\sum_{j=1}^{N} u_{ij}^{m}(\mathbf{x_j}-\mathbf{v_i})(\mathbf{x_j}-\mathbf{v_i})^{T}}{\sum_{j=1}^{N} u_{ij}^{m}} \qquad (2)$$

where **P**$_i$ is symmetric and positive and can be decomposed in the form:

$$\mathbf{P}_{\mathbf{i}} = \mathbf{Q}_{\mathbf{i}}\boldsymbol{\Lambda}_{\mathbf{i}}\mathbf{Q}_{\mathbf{i}}^{\mathbf{T}} \qquad (1)$$

Q$_i$ is an orthonormal matrix and **Λ**$_i$ = (λ$_{ik}$), k = 1,...,n, is a diagonal matrix. In the EFCM algorithm the radius r$_i$ is calculated as the geometrical mean of the elements λ$_{ik}$ by the formula:

$$r_i = \frac{1}{n}\sqrt{\prod_{k=1}^{n}\lambda_{ik}^{1/n}} = \sqrt{\det\left(P_i\right)^{1/n}} \qquad (3)$$

To avoid divisions by zero in calculus of for u$_{ij}$, for each pattern **x**$_j$ in [7] is calculated a parameter φ_j equal to the number of clusters for which $\delta_{kj} = 0$ for each k = {1,…,C}, giving the following formula for u$_{ij}$:

$$u_{ij} = \begin{cases} \dfrac{1}{\sum_{k=1}^{C}\left(\dfrac{\delta_{ij}}{\delta_{kj}}\right)^{2/(m-1)}} & \text{if} \quad \varphi_j = 0 \\ \begin{cases} 0 & \text{if } \delta_{ij} > 0 \\ \dfrac{1}{\phi_j} & \text{if } \delta_{ij} = 0 \end{cases} & \text{if} \quad \varphi_j > 0 \end{cases} \qquad (2)$$

The centers of the clusters **v**$_i$ i = 1,...,C are calculated as in the traditional fuzzy C-means algorithm, as follows:

$$\mathbf{v}_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} \mathbf{x}_j}{\sum_{j=1}^{N} u_{ij}^{m}} \qquad i = 1,\ldots,C \tag{5}$$

To ensure the separation between cluster centers, in [7] (Kaymak & Setnes, 2002) the mean radius of the i*th* cluster r$_i$ is increased at any iteration by a factor $\frac{\beta^{(t)}}{C^{(t)}}$, where C$^{(t)}$ is the number of clusters detected in the t*th* iteration and β$^{(t)}$ is defined recursively as $\beta^{(0)} = 1, \beta^{(t)} = min\left(C^{(t-1)}, \beta^{(t-1)} + 1\right)$. The optimal number of clusters is obtained merging at any iteration the two most similar clusters if their similarity is under a specified threshold. To measure the similarity between two clusters is calculated the following inclusion index:

$$S_{ik} = \frac{\sum_{j=1}^{N} \min(u_{ij}, u_{kj})}{\min\left(\sum_{j=1}^{N} u_{ij}, \sum_{j=1}^{N} u_{kj}\right)} \tag{6}$$

The similarity cluster matrix **S** is a symmetric matrix. Let **S**$^{(t)}$ be the similarity cluster matrix calculated at the t*th* iteration, η be a fixed merging error, α$^{(t)}$ = 1/(C$^{(t)}$-1) be an adaptive similarity threshold, with C$^{(t)}$ number of clusters in the t*th* iteration, and i* and k* the indices of the two clusters most similar to each other, i.e. $S_{i^*k*}^{(t)} \geq S_{ik}^{(t)}$ for i,k = 1,...,C i≠k. Then, if $\left|S_{i^*k*}^{(t)} - S_{i^*k*}^{(t-1)}\right| < \eta$ and $S_{i^*k*}^{(t)} > \alpha^{(t)}$, the two clusters are merged, and the number of clusters is reduced by one unit. We obtain:

$$\begin{cases} u_{i^*j}^{(t)} = u_{i^*j}^{(t)} + u_{k^*j}^{(t)} & \forall j \in \{1,\ldots,N\} \\ C^{(t)} = C^{(t-1)} - 1 \\ \beta^{(t)} = \beta^{(t-1)} \end{cases} \tag{7}$$

Otherwise, if $S_{i^*k*}^{(t)} > \alpha^{(t)}$the parameter β is increased and we obtain:

$$\begin{cases} \beta^{(t)} = \min\left(C^{(t-1)}, \beta^{(t-1)} + 1\right) \\ C^{(t)} = C^{(t-1)} \end{cases} \tag{8}$$

The EFCM algorithm is described in the following pseudocode.

Algorithm: *EFCM*

```
Set m, ε, η, the initial number of clusters $C^{(0)}$
β←1, S* ←0, S*^prev←1
Initialize randomly the partition matrix U and the centers $\mathbf{v}_i$
Repeat
```

```
    For  i = 1 to C        // calculate centers and radius of clusters
        Calculate the center of the $i^{th}$ cluster $\mathbf{v}_i$ by (4)
        Calculate the radius of the $i^{th}$ cluster $r_i$ by (12)
        $r_i \leftarrow r_i \cdot \beta / C$    //enlarge the radius of the $i^{th}$ cluster
    For i = 1 to C        // calculate new partition matrix
        For j = 1 to N
            Calculate the membership degree component $u_{ij}$ by (14)
    For i = 1 to C-1       //Find the two most similar clusters
        For k = i+1 to C
            Calculate $S_{ik}$ by (15)
            If $S_{ik} > S^*$
                $S^* \leftarrow S_{ik}$
    If $|S^*-S^{*prev}| < \eta$
        $\alpha=1/(C-1)$
        If $S^* > \alpha$          //merge the two most similar clusters
            For j= i+1 to N
                $u_{ij} \leftarrow u_{ij} + u_{kj}$
            Remove the kth row from U
            $C \leftarrow C-1$
        Else
            $\beta \leftarrow \min(C, \beta+1)$
  Until $\left|\boldsymbol{U}^{(t)} - \boldsymbol{U}^{(t-1)}\right| > \varepsilon$
Return the partition matrix and the volume prototypes of the final C Clusters
```

3.2 FS-EFCM algorithm: our proposal

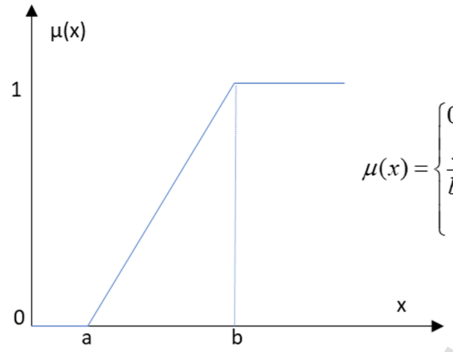
The proposed framework accomplishes the data partitioning by a semi-supervised strategy that achieves a trade-off between the expert-based evaluation of how the selected features are relevant in the current domain of interest and the incidence degree of those features in the cluster formation generated using the clustering algorithm.

The FS-EFCM exploits the known performance benefits of EFCM and introduces two fuzzy indices related to the feature description: the Feature Relevance (FR), representing the relevance degree assessed by human experts, and the Feature Incidence (FI), describing how the feature influences on the cluster formation.

The first stage of the proposed algorithm consists of an initial expert-based relevance evaluation of the features. The experts assign a score to each feature, translated in a form digest to be interpreted as a membership degree to the FR fuzzy set.

During the FS-EFCM execution, the algorithm calculates the degree of incidence of the features in the cluster, describing the membership degree to the FI fuzzy set. At each iteration, the features whose FR and FI fuzzified values are greater than or equal to a predefined threshold, are selected.

The FR and FI values are fuzzified according to the two fuzzy indices, defined by sigma fuzzy sets on a universe of a discourse given by an interval of the real line. The sigma fuzzy set, also called L-function fuzzy set, is a semi-trapezoidal fuzzy number constructed as in Figure 1 with two values a and b on the universe of the discourse, with a < b. The membership degree is 0 when x ≤ a and 1 if x ≥ b, as shown in the figure.



$$\mu(x) = \begin{cases} 0 & x < a \\ \dfrac{x - a}{b - a} & a \le x \le b \\ 1 & x > b \end{cases}$$

Figure 1. Example of a sigma fuzzy set

The use of sigma-type fuzzy sets allows modeling the two fuzzy indices easily; they are built only using the two parameters a and b. Additional details are provided in the next section.

3.3 The fuzzy indices for the feature assessment

As stated, the FS-EFCM is an extension of the EFCM algorithm, that initially works on the input whole feature set.

At each iteration, the process selects some features that, depending on the two fuzzy indices assessment, contribute to support on the one hand, the expert-defined feature importance and on the other hand, the clustering structure, that reveals how much a feature affects the detected cluster. More specifically, the two fuzzy indices are described as follows.

- *Feature relevance* (FR): it evaluates the importance of the feature in the domain where it is defined; it is often an external (subjective) assessment. For example, in the Sentiment Analysis domain, a feature could be associated with a natural language word, with a strong meaning in the emotional context; for this reason, the corresponding feature could assume a high relevance value.
  An FR value s_h is assigned by humans (in general, the domain experts) to a specific feature h_{th}, in a range of values, from a minimum m_{FR} to a maximum value M_{FR}. These boundary values are user-defined, depending on how they intend to distribute the range of relevance, according to the domain of interest. A fuzzification process is achieved by constructing a sigma fuzzy set in which

the membership degree of each feature to the FR fuzzy set is evaluated on the range [m_{FR}, M_{FR}].

- *Feature incidence* (FI): it represents the incidence of the feature on the specification of the clusters (clusters incidence), i.e., how the feature affects the cluster structure. This index is defined as a sigma fuzzy set, whose membership degree of a feature is determined by the EFCM algorithm, once the final clusters are detected. Precisely, at each algorithm execution, the FI membership degree of a feature is calculated by considering the feature value assumed in the cluster center coordinates, opportunely normalized on the domain of the feature.
  More formally, let $C^{(t)}$ be the number of clusters detected in the tth iteration and $\mathbf{v}^{(t)}_i = \{v^{(t)}_{i1},\dots,v^{(t)}_{in}\}$, with $i = 1,2,\dots,C^{(t)}$ be the center of the ith cluster, and let $\min_h = \text{mean}_h - \text{std}_h$ and $\max_h = \text{mean}_h + \text{std}_h$ be two boundary values obtained calculating the mean mean_h and the standard deviation std_h of the values of the hth component in the data. Then, the hth component of the ith cluster are normalized as follows:

$$v'^{(t)}_{ih} = \begin{cases} \dfrac{v^{(t)}_{ih} - \min_{ih}}{\max_{ih} - \min_{ih}} & if\ \max_{ih} \neq \min_{ih} \qquad i = 1,\dots, C^{(t)} \\ 0 & otherwise \end{cases} \tag{9}$$

  Finally, the weight of the hth feature is given by:

$$w^{(t)}_h = \max_{\substack{i=1,2,\dots,C^{(t)} \\ k=1,2,\dots,C^{(t)} \\ i \neq k}} \left(\left| v'^{(t)}_{ih} - v'^{(t)}_{kh} \right| \right) \tag{10}$$

  $w^{(t)}_h$ assumes values between 0 and 1; the higher the $w^{(t)}_h$ value, the more the feature affects the specification of the clusters. A value of $w^{(t)}_h$ close to zero indicates that the centers of the clusters have hth component values very similar to each other, that is, the clusters are not affected by the hth feature.

The two sigma fuzzy sets FR and FI are shown in Fig. 2, where *s* is the input variable describing the relevance score assigned by the experts to a specific feature and *w* is the feature weight calculated with respect to the cluster centers. The two sigma fuzzy sets FR and FI are defined assigning, the parameters a_{FR} and b_{FR} and a_{FI} and b_{FI}, respectively.

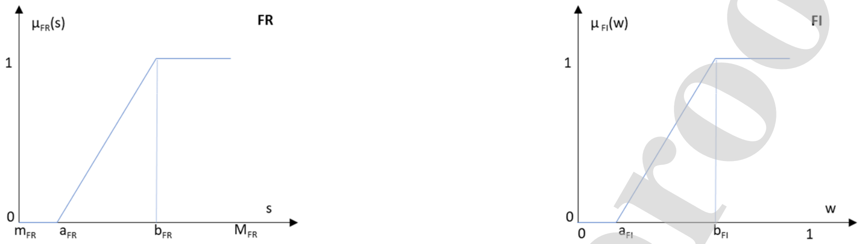


Figure 2. The sigma fuzzy sets FR and FI associated with the feature relevance and the feature incidence

3.2 The feature selection algorithm

The feature selection is described in Figure 3. It finds a balance between the two indices FR and FI. The expert associates a relevance score with each feature; these values allow building the FR index and are used in the algorithm to weight the features, taking into account the FI index as well.

The algorithm implements an iterative process: in each iteration, the EFCM optimization function helps to evaluate the incidence of each feature in the clustering structure. The feature values assumed in the cluster center coordinates allow indeed building the FI index, according to Equations (9)-(10).

The algorithm stops when a condition of stability is reached, i.e., when the difference of the FI membership degrees of a feature between two consecutive iterations is below a prefixed threshold.

Otherwise, i.e., if the stability condition does not hold, a further analysis, accomplished on the two indices allows the determination of features candidate to be removed. The process is re-iterated considering the remaining features.

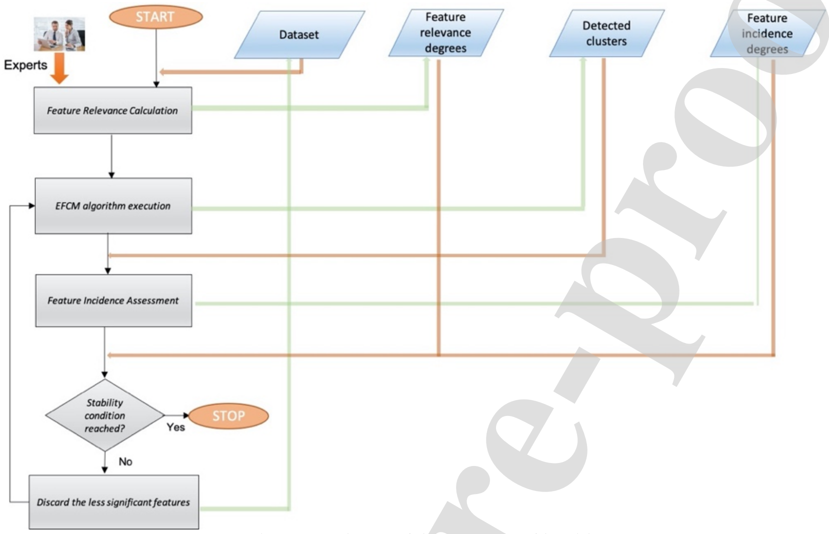

START
Experts
Dataset
Feature relevance degrees
Detected clusters
Feature incidence degrees
Feature Relevance Calculation
EFCM algorithm execution
Feature Incidence Assessment
Stability condition reached?
Yes
STOP
No
Discard the less significant features

Figure 3. Schema of the FS-EFCM algorithm

The main steps of the FS-EFCM algorithm are detailed below:

1. *Feature relevance Calculation*: the team of experts assigns a relevance score to each feature s_h with h = 1,…, H; the FR fuzzy set is defined as shown in Figure 2, where $\mu_{FR}(s_h)$ is the membership degree to FR of the h*th* feature.
2. *EFCM algorithm execution:* the EFCM algorithm is executed, taking into account the weighted feature set; initially all the input features are considered for the EFCM execution.
3. *Feature incidences assessment in the cluster formation*: once the cluster centers are normalized, as in Equation (9), the incidence of each feature in clustering structure is calculated by Equation (10). The incidence value is then used to calculate the membership degrees to the FI.
4. *Test on Stability Condition reached:* the algorithm stops when the absolute maximum difference between the FI values of a feature in two successive iterations is lower than a prefixed threshold θ. Formally, the stability condition is given by:

$$\max_{h=1,\ldots,H}\left(\left|\mu_{FI}(w_h^{(t)})-\mu_{FI}(w_h^{(t-1)})\right|\right)\le\theta \qquad (11)$$

   If the condition is satisfied, the current feature is selected to be still part of the feature set, else, the algorithm continue to the step 5.
5. *Discard less significant features*: the remaining features could be candidate to be removed by the feature set, since their contribution might be not significant in the clustering process. The selection of features candidate to be removed from the feature set is achieved by defining a new measure of significance μ_h of the *h*th feature for the clustering structure, applying the t-norm operator ● to the two fuzzy sets FR and FI:

$$\mu_h = \mu_{FR}(s_h) \bullet \mu_{FI}(w_h) \qquad h = 1,\ldots, H \tag{12}$$

If μ_h is less than a prefixed threshold δ, then the h^{th} feature is considered not significant for the cluster detection process and it is removed from the current feature set. Finally, the process returns to step 2, considering just the filtered features in the next iteration.

Below is shown the pseudocode describing the FS-EFCM algorithm.

Algorithm: FS-EFCM

Input: FS={f_1,..f_H} feature set; N x H matrix MM

```
Set $a_{FR}$, $b_{FR}$, $a_{FI}$, $b_{FI}$, $\delta$, $\theta$
Create the FR fuzzy set based on the score values assigned by the experts
Repeat
    H := number of features    //initially all the features are considered
    Execute EFCM
        For h := 1 to H
            Calculate $w_h$, by using Eq.(10)
            Calculate $\mu_{FI}(w_h)$
            Calculate $\mu_h$ by (12)
            If $\mu_h < \delta$
                Remove the $h^{th}$ feature $f_h$ from FS
                H=H-1
Until the condition of stability is not reached (by using Eq. (11))
Return the partition matrix and the volume prototypes of the final C Clusters
```

3. 4 Setting parameters

The role of the parameters is crucial to guarantee the good performance of the algorithm. The parameters a_{FR}, b_{FR}, a_{FI}, and b_{FI} used to define the two sigma fuzzy sets FR and FI, along with the thresholds δ and θ need to be set.

A proper parameter setting of the FR range boundaries is to fix a_{FR} and b_{FR} respectively equals to the bound values m_{FR} and M_{FR} of the FR range defined by the experts. Let us notice that the value m_{FR} fixed by the experts represents the lower bound of the feature relevance interval, i.e., it is the absence (or zero) of relevance, whereas M_{FR} is the highest relevance value which van be assigned to a feature.

Then, the mapping to the fuzzy set will assign $\mu_{FR}(m_{FR}) = 0$ and $\mu_{FR}(M_{FR}) = 1$.

Regarding the incidence value *w* of the feature in a cluster, it is important to consider the value range that each feature can assume in the cluster center coordinates, to avoid flattening the peculiar values of a feature in the prototypes characterization, by considering a common value range for all the features. For this reason, the range interval taken into account is $[\mu_f - \sigma_f, \mu_f + \sigma_f]$, where μ_f and σ_f represent the mean and the standard deviation of the values assumed by the feature *f* among the cluster centers.

Bearing this in mind, it makes sense to fix the parameters a_{FI} and b_{FI} of the fuzzy set FI, respectively to the lower and upper bound of this defined range.

Another parameter value to fix is θ which determines the convergence of the algorithm: the stability is reached when Equation (11) is satisfied. Empirically, it has been verified that $\theta = 0.01$ is an acceptable value, this value is assumed also by the threshold parameter value ε used for the EFCM convergence evaluation.

The setting of the parameter δ needs more investigation, as δ plays a crucial role in FS-EFCM: in each iteration, it controls the significance of the feature, determining if the feature must be removed from the set. Thus, a preliminary step has been defined by executing the EFCM algorithm for the first time. This step indeed is aimed at discovering the potential removable features, which have been considered relevant by the experts.

Let us assume that n_R is the number of features considered relevant by the experts and n_D is the number of these features candidate to be discarded after the EFCM execution. If n_D is greater than 50% of n_R (i.e., many features relevant for the expert are likely to be discarded), then the initial value of the parameter δ is re-set to a lower value.

Algorithm *SelectDeltaThreshold* describes how the parameter δ is defined. The pseudo-code shows how n_R is calculated: it is the number of features considered relevant for the experts, i.e., those ones with membership degree $\mu_{FR} \geq 0.7$ (lines 6-8).

The relevant features candidate to be removed are stored in the array sig[] of n_D size: lines 14-16 indeed describe how the array is filled; then, it is ascendingly sorted in line 17. Line 13 verifies the condition of discarding a feature (like in *FS-EFCM Algorithm*) and, at the same time, checks if the feature is relevant ($\mu_{FR} \geq 0.7$). The feature significance value is added to the array when both conditions hold. Then, the array is ordered ascendingly, ranked by feature significance values. The (non-zero) value in the intermediate position of the array will be the new value of the threshold δ. If the value in that position is zero, the next non-zero value in the array will be set to the value δ.

Let us notice that the parameter value δ must be lower than the initial value of $\mu_{FR}(s_h)$, the membership degree related to the hth feature, calculated for the FR fuzzy set. This way, line 14 is satisfied; a refinement of δ is then calculated in line 21.

The described strategy ensures that more than half of the features deemed relevant by the experts are not initially removed by FS-EFCM.

Algorithm: *SelectDeltaThreshold*

```
1.    Set $a_{FR}$, $b_{FR}$, $a_{FI}$, $b_{FI}$, $\delta$, $\theta$
2.    Create the FR fuzzy set based on the score values assigned by the experts
3.    H := number of features
4.    $n_R$:= 0    // Initialize the number of relevant features
5.    $n_D$:=0    // Initialize the number of relevant features potentially removable
6.    For h := 1 to H    // Calculate the number of relevant features
7.        If $\mu_{FR}(s_h) \geq 0.7$
8.            $n_R$:= $n_R$ +1
9.    Execute EFCM
10.   For h := 1 to H
11.       Calculate $w_h$ by using (10)
12.       Calculate $\mu_{FI}(w_h)$
```

```
13.       Calculate $\mu_h$ by using (12)
14.       If $\mu_h < \delta$ and $\mu_{FR}(s_h) \geq 0.7$
15.           $n_D := n_D + 1$
16.           $sig[n_D] := \mu_{FR}(s_h) \bullet \mu_{FI}(w_h)$
17.   Sort sig[] //the array sig[] is ascendingly sorted
18.   If $n_D / n_R > 0.5$
19.     $med := n_D / 2$   // the integer, intermediate value between 1 and $n_D$
20.     If $sig[med] > 0$
21.       $\delta := sig[med]$
22.     Else
23.       $\delta :=$ the first not null value in sig[]
24.   Return $\delta$
```

A further choice concerns the t-norm operator to be used to determine the combination of the two indices, as defined in step 5 of the algorithm, defined in Section 3.2. Among the many families of t-norms defined in literature, the most widely used in application problems are the triangular norms, shown as follows:

$$\begin{array}{lll} - \ \textit{minimum (Gödel)}\ \text{t-norm} & x \bullet y = \min(x, y) & \\ - \ \textit{product (Goguen) t-norm} & x \bullet y = x \cdot y & (13) \\ - \ \textit{Lukasiewicz t-norm} & x \bullet y = \max(x+y-1, 0) & \end{array}$$

Depending on the selected t-norm, different fuzzy intersections are generated; in particular, the minimum t-norm is the most used in fuzzy controls, whereas the product t-norm produces a more drastic intersection than the minimum t-norm.

The FS-EFCM algorithm has been carried out on several UCI Machine Learning datasets, by considering for each experiment, the three t-norms, namely minimum, product, and Lukasiewicz t-norms, respectively. As stated, the *SelectDeltaThreshold* algorithm has been initially executed to calculate the parameter δ. The FSEFCM algorithm performance has been evaluated in terms of accuracy, precision, and recall measures.

4. Test results

The FS-EFCM algorithm has been evaluated on several well-known UCI Machine Learning datasets. The effectiveness of our algorithm has been evaluated by adding some dummy features to the datasets. The first experiments reported in this section are mainly achieved on the Breast Cancer dataset, by considering three different parameter configurations. The objective of these tests is to verify the effectiveness of our method in selecting only the relevant features, analyzing how FS-EFCM manages to optimally make the trade-off between the feature relevance and its incidence in the cluster formation. To this purpose, dummy features have been added to the original features to analyze the effectiveness of FS-EFCM.

Additional experiments for a comparative analysis with other methods have been accomplished as well. Our tests are executed on an Intel Core^{TM} I7 2.90 GHz processor. The EFCM[1] parameters setting is as follows: m = 2, ε = 0.01 and η = 0.01. The stability condition threshold θ is set to 0.01 and the threshold δ is initially set to 0.1.

4.1 Breast Cancer dataset: Test Case A

The Breast Cancer dataset is composed of 699 data samples, and 9 features, in addition to the attributes that classifies the data samples (2 for benign, 4 for malignant). An initial data analysis reveals 16 data points with missing attribute values, which have been removed from the dataset.

To evaluate the applicability of the proposed method, several experiments have been carried out by considering some additional "dummy" features. Their relative relevance scores are assumed to have been assigned by the experts.

In a first experiment (*Test Case A*), two dummy features, namely, Fea_1 and Fea_2 have been added to the Breast Cancer dataset. Values assumed by these features are randomly generated for each sample: Fea_1 assumes a random value in the range [0, 1], whereas Fea_2 gets values in [0, 10].

According to the *SelectDeltaThreshold* algorithm, as an initial configuration, the threshold δ is equal to 0.1.

Then, applying the algorithm steps, let us assume that all the features have the maximum relevance score i.e., $\mu_{FR}(s_h) = 1$ with h = 1,…, H. The number of the features is given by the number of actual and the dummy features (n_R = 11, lines 6-9 of Algorithm *SelectDeltaThreshold*). After the EFCM execution (with the number of clusters equal to 2), the resulting cluster centers values are shown in Table 1.

Table 1: Breast Cancer - Test Case A - Cluster centers after running EFCM for the first time

Feature \ Cluster center	v_1	v_2
Clump Thickness	7.14	3.08
Uniformity of Cell Size	6.90	1.38
Uniformity of Cell shape	6.84	1.49
Marginal Adhesion	5.89	1.40
Single Epithelial Cell Size	5.46	2.14
Bare Nuclei	7.93	1.45
Bland Chromatin	6.19	2.15
Normal Nucleoli	6.19	1.31
Mitoses	2.58	1.11
Fea_1	0.50	0.49
Fea_2	4.92	4.99

[1] https://drive.google.com/drive/folders/1IlUTz47IK7zM411NWR__4xaiWnFG70U3?usp=sharing

The next step of the algorithm requires the cluster incidence calculation of each feature, after the fuzzification of the features to calculate $\mu_{FI}(w_h)$ (lines 11-12) and the evaluation of the significance of the feature through the calculation $\mu_{FR}(s_h)$ ● $\mu_{FI}(w_h)$ (line 15). Table 2 shows the feature significance values (line 15), resulting by using the three introduced t-norms: minimum, product and Lukasiewicz, (see Eq. 13). Let us notice that the feature significance evaluated using the three t-norms is the same for all the original features, whereas assumes values lower than the fixed threshold $\delta = 0.1$ for the dummy features Fea_1 and Fea_2. This emphasizes that these latter are the potential candidates to be removed, as their significance is very low.
Since n_D, the number of relevant features potentially removable is 2 ($n_D = 2$) and it is less than half of the relevant features (line 17 of Algorithm *SelectDeltaThreshold*), no action is required to modify the threshold value δ.

Table 2: Breast cancer - Test Case A - Feature significance and feature candidate to be removed

Feature	*Feature relevance*	*Cluster incidence*	*Feature significance t-norm min*	*Feature significance t-norm product*	*Feature significance t-norm Lukasiewicz*	*Selected*
Clump Thickness	1	0.72	0.72	0.72	0.72	YES
Uniformity of Cell Size	1	0.90	0.90	0.90	0.90	YES
Uniformity of Cell shape	1	0.90	0.90	0.90	0.90	YES
Marginal Adhesion	1	0.78	0.78	0.78	0.78	YES
Single Epithelial Cell Size	1	0.75	0.75	0.75	0.75	YES
Bare Nuclei	1	0.89	0.89	0.89	0.89	YES
Bland Chromatin	1	0.82	0.82	0.82	0.82	YES
Normal Nucleoli	1	0.80	0.80	0.80	0.80	YES
Mitoses	1	0.42	0.42	0.42	0.42	YES
Fea_1	1	0.02	0.02	0.02	0.02	NO
Fea_2	1	0.01	0.01	0.01	0.01	NO

The two features Fea_1 and Fea_2 have an almost null cluster incidence value; this is consistent with the fact that the features, assuming random values in a fixed range, do not affect the cluster formation. This result highlights the effectiveness of FS-EFCM in discarding features that play no role in the clustering formation, even they assume non-negligible relevance values given by the expert.

Once FS-EFCM removed the two intruded features, the EFCM has re-launched on the remaining features. The FS-EFCM will stop, as the condition of stability will be hold. Table 3 shows the performance results of the algorithm on the breast cancer dataset, c with all the 11 features. As shown in the table, after the second cycle, the stability condition value is less than the prefixed threshold θ (see (11), or line 12 of Algorithm

FS-EFCM) and the algorithm ends. Table 3 shows also the accuracy, precision, recall, and F1-score evaluated on all the 11 features and then, once applied *SelectDeltaThreshold* algorithm, on the remaining (original) 9 features.
Let us notice that the classification performance of the algorithm increases in the second cycle, once removed the two added features.

Table 3: Breast cancer - Test Case A - Performance measures after any cycle

Measure \ Cycle	1	2
Number of features	11	9
Stability condition	---	0.007
Accuracy	96.90%	97.14%
Precision	96.81%	97.06%
Recall	96.39%	96.63%
F-score	96.64%	96.84%

4.2 Breast Cancer dataset: Test Case B

The second experiment (Test Case B) has been carried out on the same dataset, with the same parameter configuration. The difference consists on the possible random range values assumed by the dummy feature Fea_2, which can vary depending on the class label. It assumes values in the range [1, 5], when the class label associated with the relative sample is 2, whereas the values are in the interval [6,10] if the class label is 4. Let us assume that all the original features have the maximum relevance score, i.e., equal to 1, while the feature relevance for Fea_1 and Fea_2 is 0.5. In this case, all the original features are considered relevant to the user, except the two dummy features, which have a feature relevance less than 0.7. We apply the *SelectDeltaThreshold* algorithm setting initially the threshold δ to 0.1. Table 4 shows the cluster center values after running EFCM (line 9).

Table 4: Breast Cancer – Test Case B - Cluster centers after the first running of EFCM

Feature \ Cluster center	$\boldsymbol{v}_1$	$\boldsymbol{v}_2$
Clump Thickness	7.20	3.03
Uniformity of Cell Size	6.91	1.34
Uniformity of Cell shape	6.86	1.45
Marginal Adhesion	5.88	1.37
Single Epithelial Cell Size	5.46	2.12
Bare Nuclei	7.97	1.39
Bland Chromatin	6.20	2.12
Normal Nucleoli	6.19	1.28
Mitoses	2.59	1.10

Fea_1	0.50	0.49
Fea_2	7.85	2.74

Table 5 instead shows the feature relevance and cluster incidence values obtained for each feature, evaluated by applying the three t-norms (lines 10-15). Let us notice that the only feature candidate to be removed is Fea_1; also, in this case, the number of relevant features, potentially removable is 0. Moreover, since the feature relevance of Fea_1 is less than 0.7, no further change of the threshold δ is required.

Let us notice that Fea_2 instead has non-negligible values for the cluster incidence as well as the t-norm-based significance, even though it is not very important for experts, which assigned a lower relevance value.

Table 5: Breast Cancer - Test Case B - Feature significances and features removed

Feature	*Feature relevance*	*Cluster incidence*	*Feature significance t-norm min*	*Feature significance t-norm product*	*Feature significance t-norm Lukasiewicz*	*Selected*
Clump Thickness	1	0.74	0.74	0.74	0.74	YES
Uniformity of Cell Size	1	0.91	0.91	0.91	0.91	YES
Uniformity of Cell shape	1	0.90	0.90	0.90	0.90	YES
Marginal Adhesion	1	0.79	0.79	0.79	0.79	YES
Single Epithelial Cell Size	1	0.75	0.75	0.75	0.75	YES
Bare Nuclei	1	0.90	0.90	0.90	0.90	YES
Bland Chromatin	1	0.83	0.83	0.83	0.83	YES
Normal Nucleoli	1	0.81	0.81	0.81	0.81	YES
Mitoses	1	0.43	0.43	0.43	0.43	YES
Fea_1	0.5	0.02	0.02	0.01	0.00	NO
Fea_2	0.5	0.86	0.50	0.43	0.36	YES

Let us also remark that the feature significance values calculated for the features Fea_1 and Fea_2 depend on the type of t-norm used. The t-norm *min* returns the highest value, viz., the less reduced feature significance value. Let us observe that FS-EFCM keeps working correctly. In fact, even if both the two dummy features have the same relevance value, only Fea_1 is removed, as Fea_2 (whose values were defined according to the class label) assumes a non-neglectable incidence value in the clustering generation.

Once removed Fea_1, the FS-EFCM algorithm is carried out on the remaining features. Table 6 reports the performance results after the two EFCM executions; it is evident that classification improves after the second execution, for all the metrics analyzed. This fact evidences that although the dummy feature Fea_2 is not useful for cluster structuring, it does not affect the clustering process negatively: it acts reinforcing the classification, because its values are by construction, strictly correlated to the class labeling.

Table 6: Breast cancer - Test Case B - Performance measures after any cycle

Measure \ Cycle	1	2
Number of features	11	10
Stability condition	---	0.008
Accuracy	97.58%	97.63%
Precision	97.44%	97.53%
Recall	96.91%	97.15%
F-score	97.06%	97.26%

Finally, let us observe that the overall performances of this experiment improve slightly, compared to those obtained in Test Case A, just because of the construction of the Fea_2 values, strictly correlated to the designated classes.

4.3 Breast Cancer dataset: Test Case C

The third experiment (Test Case C) consists of extending the breast cancer dataset with 6 dummy features, Fea_1, …, Feat_6. Let us assume random values in the range [0-10] with a feature relevance equal to 1 for all the introduced dummy features, whereas the relevance values for the original features will be 0.5. This setting forces the dummy features are supposed to be more relevant than the original ones.
Likewise, as the previous experiments, the *SelectDeltaThreshold* algorithm has been carried out with δ=0.01. After EFCM execution (line 9), the detected two clusters and the relative cluster center values are shown in Table 7.

Table 7: Breast Cancer - Test Case C - Cluster centers after running EFCM for the first time

Feature \ Cluster center	v_1	v_2
Clump Thickness	3.19	6.59
Uniformity of Cell Size	1.57	6.01
Uniformity of Cell shape	1.67	6.00
Marginal Adhesion	1.55	5.15
Single Epithelial Cell Size	2.26	4.95
Bare Nuclei	1.67	6.94
Bland Chromatin	2.28	5.54
Normal Nucleoli	1.48	5.38
Mitoses	1.17	2.36
Fea_1	5.11	4.84

Fea_2	4.79	4.59
Fea_3	5.18	4.88
Fea_4	4.88	5.00
Fea_5	5.16	5.05
Fea_6	4.81	4.81

Table 8 synthesizes the feature relevance and cluster incidence values obtained for each feature, as well as the features significance evaluated with the three t-norms.

Table 8: Breast Cancer - Test Case C - Feature significances and features removed

Feature	*Feature relevance*	*Cluster incidence*	*Feature significance t-norm min*	*Feature significance t-norm product*	*Feature significance t-norm Lukasiewicz*	*Selected*
Clump Thickness	0.5	0.60	0.50	0.30	0.10	YES
Uniformity of Cell Size	0.5	0.72	0.50	0.36	0.22	YES
Uniformity of Cell shape	0.5	0.72	0.50	0.36	0.22	YES
Marginal Adhesion	0.5	0.63	0.50	0.31	0.13	YES
Single Epithelial Cell Size	0.5	0.60	0.50	0.30	0.10	YES
Bare Nuclei	0.5	0.72	0.50	0.36	0.22	YES
Bland Chromatin	0.5	0.66	0.50	0.33	0.16	YES
Normal Nucleoli	0.5	0.64	0.50	0.32	0.14	YES
Mitoses	0.5	0.34	0.34	0.17	0.00	YES
Fea_1	1	0.04	0.04	0.04	0.04	NO
Fea_2	1	0.04	0.04	0.04	0.04	NO
Fea_3	1	0.05	0.05	0.05	0.05	NO
Fea_4	1	0.03	0.03	0.03	0.03	NO
Fea_5	1	0.02	0.02	0.02	0.02	NO
Fea_6	1	0.00	0.00	0.00	0.00	NO

It is interesting to notice that all the six added features are not selected, even though they are all considered relevant by the expert.

At this point, according to the *SelectDeltaThreshold* algorithm, since the condition $n_D/n_R > 0.5$ (line 17) is verified, i.e., the number of feature candidate to be removed is greater than the half of the relevant features, a new value 0.03 is set for the threshold δ. Fixing δ = 0.03 allows us to discard just two features Fea_5 and Fea_6 and save all the remaining dummy and original features, whose significance value are higher than δ.

The *SelectDeltaThreshold* algorithm is re-launched. In this execution, also the features Fea_2 and Fea_4 are removed since their feature significance values are below the threshold 0.03, as shown in Table 9.

Table 9: Breast Cancer - Test Case C - Feature significances and features removed in the second cycle

Feature	*Feature relevance*	*Cluster incidence*	*Feature significance t-norm min*	*Feature significance t-norm product*	*Feature significance t-norm Lukasiewicz*	*Selected*
Clump Thickness	0.5	0.67	0.50	0.33	0.33	YES
Uniformity of Cell Size	0.5	0.81	0.50	0.41	0.41	YES
Uniformity of Cell shape	0.5	0.81	0.50	0.41	0.41	YES
Marginal Adhesion	0.5	0.71	0.50	0.35	0.35	YES
Single Epithelial Cell Size	0.5	0.67	0.50	0.34	0.34	YES
Bare Nuclei	0.5	0.81	0.50	0.40	0.40	YES
Bland Chromatin	0.5	0.74	0.50	0.37	0.37	YES
Normal Nucleoli	0.5	0.72	0.50	0.36	0.36	YES
Mitoses	0.5	0.38	0.38	0.19	0.19	YES
Fea_1	1	0.03	0.03	0.03	0.03	YES
Fea_2	1	0.02	0.02	0.02	0.02	NO
Fea_3	1	0.04	0.04	0.04	0.04	YES
Fea_4	1	0.01	0.01	0.01	0.01	NO

After the third execution, also the features Fea_1 and Fea_3 are removed, as shown in Table 10. It is evident that the algorithm is able to “recognize” insignificant features, even though they have been selected as relevant.

Table 10: Breast Cancer - Test Case C - Feature significances and features removed in the third cycle

Feature	*Feature relevance*	*Cluster incidence*	*Feature significance t-norm min*	*Feature significance t-norm product*	*Feature significance t-norm Lukasiewicz*	*Selected*
Clump Thickness	0.5	0.71	0.50	0.35	0.35	YES
Uniformity of Cell Size	0.5	0.87	0.50	0.44	0.44	YES
Uniformity of Cell shape	0.5	0.87	0.50	0.44	0.44	YES
Marginal Adhesion	0.5	0.76	0.50	0.38	0.38	YES
Single Epithelial Cell Size	0.5	0.73	0.50	0.36	0.36	YES
Bare Nuclei	0.5	0.87	0.50	0.43	0.43	YES
Bland Chromatin	0.5	0.80	0.50	0.40	0.40	YES
Normal Nucleoli	0.5	0.77	0.50	0.39	0.39	YES

Mitoses	0.5	0.41	0.41	0.21	0.21	YES
Fea_1	1	0.00	0.00	0.00	0.00	NO
Fea_3	1	0.01	0.01	0.01	0.01	NO

The performance of the FS-EFCM over its four iterations (before reaching the stability condition) are reported in Table 11. The convergence is reached at the end of the fourth cycle.

Table 11: Breast cancer - Test Case C - Performance measures after each cycle

Measure \ Cycle	1	2	3	4
Number of features	15	13	11	9
Stability condition	---	0.037	0.019	0.008
Accuracy	95.72%	95.83%	96.15%	97.14%
Precision	95.65%	95.72%	96.12%	97.07%
Recall	95.18%	95.26%	95.66%	96.62%
F-score	95.41%	95.49%	95.88%	96.84%

Let us observe that the classification performances for the last execution of the *SelectDeltaThreshold* algorithm are similar to the ones calculated for *Test Case A*, where just two additional dummy features were considered. These results reveal that the FS-EFCM algorithm is robust with respect to the feature relevance values assigned by the humans. In *Test Case C* indeed, although the relevance assigned by humans is higher for the dummy features, compared with the original ones (1 versus 0.5), the algorithm can discriminate the proper features accurately, with classification performances very similar to those one obtained for *Test Case A*.

4.4 Synoptic comparative analysis

In order to assess the effectiveness of the proposed algorithm, additional performance evaluation has been accomplished, comparing our method with well-known feature selection algorithms such as Principal Concept Analysis (PCA) and Linear Discriminant Analysis (LDA), used in combination with equally well known classification algorithms such as Support Vector Machine (SVM) [1,19], and K-Nearest Neighbours (KNN) [4, 11, 12].

For a complete analysis of the proposed classification method, the hold-out technique has been applied. In all experiments, the dataset was randomly partitioned in two subsets: the former containing 80% of the data points used as a training set and the former containing 20% of the data points used as a test set.

Let us notice that the result of the FS-EFCM execution is the data partitioning in clusters; no class label is given; thus, each data point is assigned to the cluster to whose center distance it is closest.

The KNN algorithms are executed several times varying the number of neighbors K and the Euclidean distance is used. Similarly, the SVM algorithms is executed many times, by varying the values of the C penalty parameter and the type of kernel function (linear, polynomial, RBF, sigmoidal).

Comparative analyses on several UCI Machine Learning training sets are shown in Table 12: for each dataset, the final performance is reported for different combination of selection algorithms and classification ones. The best performance results obtained by varying KNN and SVM parameter configurations are shown in Table 12. Let us observe that our method is quite stable to the influence of not very significant features, and it seems that the classification performances are promising, compared with the other approaches.

Table 12: Performance comparison on UCI machine learning datasets – training sets

Dataset	*Index*	*PCA +* KNN	*LDA +* KNN	*PCA +* SVM	*LDA +* SVM	*FS-EFCM*
BREAST CANCER	Accuracy	93.26%	78.29%	93.45%	78.51%	97.14%
	Precision	93.13%	78.03%	93.32%	78.30%	97.06%
	Recall	92.88%	77.74%	93.07%	78.19%	96.63%
	F-score	93.00%	77.88%	93.19%	78.24%	96.84%
IRIS	Accuracy	83.41%	63.29%	83.88%	64.05%	91.14%
	Precision	91.02%	52.03%	91.76%	53.76%	91.45%
	Recall	84.19%	62.98%	85.03%	63.74%	92.29%
	F-score	87.47%	56.98%	88.27%	58.33%	93.38%
GLASS	Accuracy	80.31%	53.31%	80.11%	67.23%	91.14%
	Precision	91.17%	67.88%	90.02%	66.41%	95.39%
	Recall	80.38%	54.56%	80.13%	67.29%	91.45%
	F-score	85.44%	60.50%	84.78%	66.85%	93.38%
VEHICLE	Accuracy	51.58%	47.31%	51.33%	46.87%	59.52%
	Precision	51.22%	46.88%	51.05%	46.26%	59.71%
	Recall	51.37%	47.23%	51.18%	46.49%	59.56%
	F-score	51.29%	47.05%	51.11%	46.37%	59.63%
WINE	Accuracy	80.31%	64.31%	80.11%	68.23%	92.74%
	Precision	90.47%	72.88%	90.02%	73.41%	90.39%
	Recall	80.38%	64.56%	80.13%	64.29%	90.45%
	F-score	85.13%	68.47%	84.79%	68.55%	90.42%
WISCONSIN	Accuracy	92.31%	75.52%	92.56%	76.23%	94.67%
	Precision	92.47%	75..81%	92.60%	76.41%	94.68%
	Recall	92.08%	75.69%	92.13%	77.29%	94.52%
	F-score	92.27%	75.75%	92.36%	76.85%	94.60%

Table 13: Performance comparison on UCI machine learning datasets -test sets

Dataset	*Index*	*PCA +* KNN	*LDA +* KNN	*PCA +* SVM	*LDA +* SVM	*FS-EFCM*
BREAST CANCER	Accuracy	93.04%	75.86%	93.23%	77.44%	96.98%
	Precision	93.21%	77.48%	92.17%	77.91%	96.85%
	Recall	92.57%	77.12%	92.36%	77.79%	96.61%
	F-score	92.89%	77.30%	92.26%	77.85%	96.73%
IRIS	Accuracy	83.08%	63.29%	82.95%	62.63%	91.36%
	Precision	90.75%	52.03%	88.70%	54.58%	91.28%
	Recall	85.31%	62.98%	84.31%	60.71%	92.10%
	F-score	87.95%	56.98%	86.45%	57.48%	91.69%
GLASS	Accuracy	84.29%	52.57%	80.59%	65.72%	85.63%
	Precision	93.04%	64.49%	89.34%	65.65%	95.21%
	Recall	83.10%	52.53%	80.06%	66.11%	91.47%
	F-score	87.79%	57.90%	84.45%	65.88%	93.30%
VEHICLE	Accuracy	52.13%	48.44%	50.21%	46.31%	58.74%
	Precision	52.05%	46.02%	50.59%	45.74%	59.08%
	Recall	52.38%	46.50%	50.76%	46.10%	58.72%
	F-score	52.21%	46.26%	50.67%	45.92%	58.90%
WINE	Accuracy	79.67%	63.41%	79.39%	65.12%	92.16%
	Precision	91.34%	71.37%	88.82%	70.20%	90.43%
	Recall	81.59%	65.29%	79.65%	65.95%	90.58%
	F-score	86.19%	68.19%	83.99%	68.01%	90.50%
WISCONSIN	Accuracy	92.56%	74.87%	91.28%	75.50%	94.52%
	Precision	92.64%	75.33%	92.04%	75.69%	94.56%
	Recall	92.33%	75.41%	91.43%	76.15%	94.43%
	F-score	92.48%	75.37%	91.73%	75.92%	94.49%

Table 13 shows the performance results obtained on the test sets. The results highlight the absence of overfitting for all the methods used in the comparison tests as the performances obtained for the test sets are comparable with those obtained for the corresponding training sets, shown in Table 12, with irrelevant differences.
Let us observe that all the considered metrics show better performances on our methods compared with the results returned by PCA+KNN, LDA+KNN, PCA+SVM and LDA+SVM for all classification datasets used.

5 Conclusion

The paper presents a novel clustering algorithm that takes into account the feature relevance proposed by human experts. The algorithm accomplishes an initial feature selection based on the expert suggestions that "duels" with the intrinsic data distribution among cluster structure. The clustering algorithm, called Feature Selection EFCM (FS-EFCM in short), aims at promoting the importance of some features in the reference domain of interest and, at the same time, preserving their incidence in the natural clustering formation. The benefits introduced by the FS-EFCM algorithm are manifold:

- the algorithm exploits the EFCM, an extension of FCM algorithm, presenting interesting peculiarity, such as the stability to the random initialization, compared to FCM (as stated in Section 3);
- the algorithm design takes into account the human suggestions for the selection of relevant features, viz., the features that, from the expert viewpoint, are crucial to describe the domain of interest;
- at the same time, the algorithm filters irrelevant and noise information: its execution indeed evidences robustness to the dummy features, and once reached the optimization function stability, it is not affected by insignificant features.

Experimental results confirm the effectiveness of the proposed algorithm, showing its benefits into discriminating not very significant features and show promising classification performance compared with known classification approaches.

As a future development, the effectiveness FS-EFCM algorithm will be assessed on high-dimensional and massive datasets where a solid feature selection method is required as well as specific strategies for the treatment of large amounts of data.

References

[1] M. Awad, R. Khanna, Support Vector Machines for Classification. In: Efficient Learning Machines. A press, Berkeley, CA (2015), pp. 39-66., DOI: 10.1007/978-1-4302-5990-9_3

[2] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA (1981).

[3] J. C. Bezdek, R. Ehrlich, W. Full, The Fuzzy C-Means clustering algorithm, Computers & Geosciences 10 (2-3), (1984), 191-203.

[4] B. V. Dasarthy, Nearest Neighbor Classification Techniques. IEEE Press, Hoboken (NJ), (1990), 447 pp., ISBN: 978-0818689307.

[5] X. Deng, Y Li, J. Weng, J. Zhang, J., Feature selection for text classification: A review. Multimed Tools Appl 78, 3797–3816 (2019). https://doi.org/10.1007/s11042-018-6083-5

[6] F. Di Martino, S. Senatore, S. Sessa, A lightweight clustering–based approach to discover different emotional shades from social message streams, International Journal of Intelligent Systems, (2019), vol. 0, pp. 1-19, DOI: 10.1002/int.22105.

[7] U. Kaymak, M. Setnes, Fuzzy clustering with volume prototype and adaptive cluster merging, IEEE Transactions on Fuzzy Systems 10, no. 6, (2002), 705-712.

[8] A. M., Martinez, A. C, PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence. 23 (2), (2001), 228–233. doi:10.1109/34.908974.

[9] G. J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience, (2004), ISBN 978-0-471-69115-0. MR 1190469.

[10] S. Mohapatra, D. Patra, D., S. Satpathy, An ensemble classifier system for early diagnosis of acute lymphoblastic leukemia in blood microscopic images. Neural Comput & Applic 24, 1887–1904 (2014). https://doi.org/10.1007/s00521-013-1438-3

[11] A. Mucherino, P. J. Papajorgji, P. M. Pardalos, k-Nearest Neighbor Classification. In: Data Mining in Agriculture. Springer Optimization and Its Applications, Springer, New York, NY , Springer Optimization and Its Applications book series (SOIA), vol. 34, (2009), DOI: 10.1007/978-0-387-88615-2_4.

[12] M. N. Murty, V. S: Devi, Nearest Neighbour Based Classifiers. In: Pattern Recognition. Undergraduate Topics in Computer Science, vol 0. Springer, London, (2011)., pp. 48-85, DOI: 10.1007/978-0-85729-495-1_3.

[13] T. M. Nguyen, Q. M. J. Wu, Online Feature Selection Based on Fuzzy Clustering and Its Applications, IEEE Transactions on Fuzzy Systems, 24 (6), (2016), 1294 – 1306.

[14] A. A. Rachman, Z. Rustam, Cancer classification using Fuzzy C-Means with feature selection, 2016 IEEE 12th International Conference on Mathematics, Statistics, and Their Applications (ICMSA), Banda Aceh, Indonesia, 4-6 October 2016, (2016), DOI: 10.1109/ICMSA.2016.7954302.

[15] M. Shokouhifar, F. Farokhi, Feature Selection using Supervised Fuzzy C-Means algorithm with Ant Colony Optimization, 2010 IEEE 3rd International Conference on Machine Vision (ICMV 2010), 28-30 December 2010, Hong Kong, 2010, pp. 441-446, DOI: 10.13140/2.1.2210.8167

[16] C. Sima, E. R. Dougherty, The peaking phenomenon in the presence of feature-selection, Pattern Recognition Letters, 29 (11), (2008), 1667-1674.

[17] K. G. Srinivasa, A Singh, A O Thomas, K. R. Venugopal, L. M. Patnaik, Generic Feature Extraction for Classification using Fuzzy C - Means Clustering. ICISIP 2005 3rd International Conference on Intelligent Sensing and Information Processing, 14-17 december 2005, Bangalore, India, IEEE publisher, (2005), pp. 33-38, DOI: 10.1109/ICISIP.2005.1619447.

[18] M. Tsagris, V. Lagani, I. Tsamardinos, Feature selection for high-dimensional temporal data. BMC Bioinformatics 19, 17 (2018). https://doi.org/10.1186/s12859-018-2023-7

[19] J. Weston, C. Watkins. Support Vector Machines for Multi-Class Pattern Recognition. In ESANN 1999: Proceedings of the 7th European Symposium on Artificial Neural Networks, Bruges, Belgium, 21–23 April 1999, (1999), pp. 219–224.

[20] J. Wu, Unsupervised Intrusion Feature Selection based on Genetic Algorithm and FCM, In: Zhu R., Ma Y. (eds) Information Engineering and Applications. Lecture Notes in Electrical Engineering, vol 154. Springer, London, (2012), pp. 1005-1012, DOI: 10.1007/978-1-4471-2386-6_131.

[21] V. Bolón-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: A review and future trends, Information Fusion, Volume 52, 2019, Pages 1-12, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2018.11.008.

[22] Y. Wang, L. Feng, Hybrid feature selection using component co-occurrence based feature relevance measurement, Expert Systems with Applications, Volume 102, 2018, Pages 83-99, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2018.01.041.

[23] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. de Albuquerque, S. Mirjalili, A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection, Expert Systems with Applications, Volume 139, 2020, 112824, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2019.112824.

[24] P. Della Rocca, S. Senatore, V. Loia, A semantic-grained perspective of latent knowledge modeling, Information Fusion, Volume 36, 2017, Pages 52-67, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2016.11.003.

[25] Chandrashekar G., Sahin F., A survey on feature selection methods, Computers & Electrical Engineering,Volume 40, Issue 1,2014, Pages 16-28, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2013.11.024

[26] Y. Kim, W. Street, and F. Menczer, "Feature Selection for Unsupervised Learning via Evolutionary Search," Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 365-369, 2000.

[27] B. Powalka, J. S. Dhupia, A. Galip Ulsoy, R. Katz, "Identification of machining force model parameters from acceleration measurements" International Journal of Manufacturing Research (IJMR), Vol. 3, No. 3, 2008

[28] C. Pozna, R.-E. Precup, "Applications of Signatures to Expert Systems Modelling" Acta Polytechnica Hungarica Vol. 11, No. 2, 2014.

[29] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3u (2003), 1157–1182.

[30] Huan Liu and Lei Yu, "Toward integrating feature selection algorithms for classification and clustering," in IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pp. 491-502, April 2005, doi: 10.1109/TKDE.2005.66.

[31] Amin Zarshenas and Kenji Suzuki. 2016. Binary coordinate ascent. Know.-Based Syst. 110, C (October 2016), 191–201. DOI: 10.1016/j.knosys.2016.07.026

[32] D. Asir, S. Appavu, E. Jebamalar, Literature review on feature selection methods for high-dimensional data. International Journal of Computer Applications, 136 (1), 9–17, 2016.

[33] F. Di Martino, S. Sessa. The extended fuzzy C-means algorithm for hotspots in spatio-temporal GIS, Expert Systems with Applications, 2011, 38 (9), Issue 9 11829-11836.

[34] F. Di Martino, S Sessa, U. E. S. Barillari, M. R: Barillari. Spatio-temporal hotspots and application on a disease analysis case via GIS. Soft Comput 18, 2377-2384 (2014).

[35] F. Di Martino, S. Sessa. Extended Fuzzy C-Means Hotspot Detection Method for Large and Very Large Event Dataset, Information Sciences, 2018, 441, 198-2015.

[36] Zall, Raziyeh and Mohammad Reza Kangavari. “On the Construction of Multi-Relational Classifier Based on Canonical Correlation Analysis.” International journal of artificial intelligence 17 (2019): 23-43.

[37] R. Precup, T. Teban, A. Albu, A. Borlea, I. A. Zamfirache and E. M. Petriu, "Evolving Fuzzy Models for Prosthetic Hand Myoelectric-Based Control," in IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 7, pp. 4625-4636, July 2020, doi: 10.1109/TIM.2020.2983531.

HIGHLIGHTS

- A clustering algorithm called Feature Selection EFCM (FS-EFCM) has been defined.
- It acquires human score describing the feature relevance.
- It finds a balancing between the feature relevance and their incidence in the cluster formation.
- The core algorithm EFCM is stable to the random initialization of the cluster centres.
- Comparative experiments on known UCI datasets show the effectiveness of the approach.

Credit author statement

Author Contributions: Conceptualization, F.D.M. and S. S.; methodology, F.D.M. and S. S.; software, F.D.M. and S. S.; validation, F.D.M. and S. S.; formal analysis, F.D.M. and S. S.; investigation, F.D.M. and S. S.; resources, F.D.M. and S. S.; data curation, F.D.M. and S. S.; writing—original draft preparation, F.D.M. and S. S.; writing—review and editing, F.D.M. and S. S.; visualization, F.D.M. and S. S.; supervision, F.D.M. and S. S.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: