Applied Soft Computing Journal 148 (2023) 110917

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy classification with distance-based depth prototypes: High-dimensional unsupervised and/or supervised problems

Itziar Irigoien [a], Susana Ferreiro [b], Basilio Sierra [a], Concepción Arenas [c,*]

[a] Department of Computation and Artificial Intelligence, University of the Basque Country (UPV/EHU), Manuel Lardizabal 1, Donostia, 20018, Gipuzkoa, Spain
[b] Intelligent Information Systems Unit, Iñaki Goenaga kalea, 5, Eibar, 20600, Gipuzkoa, Spain
[c] Statistics Section of the Department of Genetics, Microbiology and Statistics, University of Barcelona (UB), Avinguda Diagonal, 643, Barcelona, 08028, Catalunya, Spain

ARTICLE INFO

Keywords:
Fuzzy classification
Prototypes
Distances
Depth function

ABSTRACT

Supervised and unsupervised classification is crucial in many areas where different types of data sets are common, such as biology, medicine, or industry, among others. A key consideration is that some units are more typical of the group they belong to than others. For this reason, fuzzy classification approaches are necessary. In this paper, a fuzzy supervised classification method, which is based on the construction of prototypes, is proposed. The method obtains the prototypes from an objective function that includes label information and a distance-based depth function. It works with any distance and it can deal with data sets of a wide nature variety. It can further be applied to data sets where the use of Euclidean distance is not suitable and to high-dimensional data (data sets in which the number of features p is larger than the number of observations n, often written as $p >> n$). In addition, the model can also cope with unsupervised classification, thus becoming an interesting alternative to other fuzzy clustering methods. With synthetic data sets along with high-dimensional real biomedical and industrial data sets, we demonstrate the good performance of the supervised and unsupervised fuzzy proposed procedures.

1. Introduction

In a variety of fields and applications, supervised or unsupervised classification is essential. One important question is that not all objects in a group have the same representativeness. Some are more typical than others, and therefore some objects better represent the group they belong to. Perhaps, Fisher's linear discriminant analysis (LDA) is one of the most well-known supervised classification methods, using a crisp type of membership labels. From a crisp point of view, groups are individually exclusive and no ambiguity is allowed. So, all units present a yes/no class membership, and an object belongs to a group if it possesses the necessary and sufficient conditions to determine its membership. However, in real-world data sets, this crisp approach may cause classifiers to be incapable of giving trustworthy rates of correct classification, due to the all-or-nothing concept of group membership. Alternatively, a fuzzy label indicates degrees of membership, which are no longer restricted to just two values (yes=1, no=0), but can be 0, 1, or any value in between. From a fuzzy point of view, it is possible to model real-world problems where some objects are better examples and more characteristic or typical of the group than others. The areas of application of fuzzy approximation are wide, including image classification [1], face recognition [2], genetics [3], medicine [4] or industry. In the latter, fuzzy perspective can be very advantageous to determine when it is necessary to replace components before they can damage a machine, and where there is an ordering among the different conditions of the components.

In the literature, there is an extensive review on unsupervised fuzzy methods highlighting their historical development [5] and another review focusing on the performance of such methods [6]. In brief, the well-known fuzzy c-means, which uses the Euclidean distance, has some shortcomings: it is sensitive to the selection of the initial clustering center point and thus, prone to falling into the problem of an optimal local solution. In [7] there are different approximations to compensate for these limitations. Other works use a fuzzy weighting technique for feature weights [8] or include entropy regularization [9]. The framework introduced in [10] for feature selection in clustering leads to sparse fuzzy c-means algorithms working with high dimensional data [11]. Other approximations for high dimensional data use strategies of parallel computing and ensemble learning [12].

On the other hand, little effort has been made into fuzzy supervised classification. The first fuzzy discriminant method (FDA) computes

[*] Corresponding author.
E-mail address: carenas@ub.edu (C. Arenas).

https://doi.org/10.1016/j.asoc.2023.110917
Received 1 December 2022; Received in revised form 7 September 2023; Accepted 4 October 2023
Available online 10 October 2023

fuzzy within-class and between-class scatter matrices, and the resulting eigenvalue problem is solved [13]. LDA and FDA are linear classifiers, based on the use of the Euclidean distance. That makes them unsuitable for all types of data or more sophisticated situations. Thus, the FDA method has been extended by using kernels to deal with non-linear separable problems [14]. A supervised iterative fuzzy k-means using kernel functions was presented in [15], and kernel-based maximum a posterior classification has been introduced in [16]. A different approach was developed in [17], based on fuzzy regression with point prototypes. More recently, [18] introduced a kernel fuzzy discriminant procedure to facilitate robust classification in the field of image classification. Moreover, different approaches are based on the idea of the nearest prototype [19]. Soft assignments of the data vectors to the prototypes based on a Gaussian mixture approach were introduced in [20] with the soft nearest classification method. In [21] the authors introduced an adaptive prototype-based fuzzy classification approach to address the problem of classification with large data sets when only a few labeled objects can be provided by the user. More recently, Ashtari et al. [22] introduced a different approach, called Supervised Fuzzy Partitioning (SFP). It is derived from k-means and takes advantage of labels and the loss function by incorporating them into the objective function through a surrogate term penalizing the empirical risk. However, the method is only applicable with the Euclidean distance.

This paper, aiming to address this latter issue, and following the ideas of [22], proposes a new supervised classification approach called Fuzzy Classification based on Depth Function (FC-DF).

The major contributions and novelties of this paper are summarized as follows:

1. FC-DF can use any distance. Thus, it can be applied to a large spectrum of data types, where the Euclidean distance is not suitable, but other distances are.
2. FC-DF can be applied to high-dimensional data (data sets in which the number of features p is larger than the number of observations n, often written as $p \gg n$), which therefore helps to overcome the curse of dimensionality.
3. Instead of obtaining centroids as prototypes, FC-DF identifies K observations, selected from the deepest of the fuzzy group, as prototypes. In this way, the prototypes always belong to the sample, which does not always occur with the centroids.
4. The objective function uses the *log-loss* function and includes label information as well as a distance-based depth function to fuzzify the partition.
5. The model can also be adapted to unsupervised classification.
6. The experimental evaluations prove that FC-DF performs significantly better than SFP when the characteristics of the data do not allow the use of the Euclidean distance, but the use of a distance appropriate to the type of data.

The remainder of the paper is structured as follows: a brief description of the background theory is in Section 2. Section 3 describes the proposed algorithm. A brief description of the used synthetic and real data sets is presented in Section 4. Details of the experimental results obtained on data sets, and discussion are in Section 5. Conclusions and future work are given in Section 6.

2. Background

This section introduces the notation and provides a brief description of some concepts, first used in the context of a crisp partition. Then, in the following section, they are extended to the fuzzy case.

Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be n units measured in $\mathbb{R}^p$ and let $\delta(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij}$ be a distance function between observations $i, j = 1, \dots, n$. A fuzzy partition of the n observations into K clusters can be expressed by n membership vectors $\mathbf{u}_i$ in $\mathbb{R}^K$, where each u_{ik} expresses the membership degree of observation i to cluster C_k, with $\sum_{k=1}^K u_{ik} = 1$ and $u_{ik} \geq 0$ for $i = 1, \dots, n$ and $k = 1 \dots, K$. In the particular case with one $u_{ik} = 1$ and the other values equal to 0, the partition is crisp (non-fuzzy). For a crisp partition, the concepts of geometric variability, proximity function, distance between two groups and a distance-based depth function have been previously defined and used in different contexts (for a review see [23] and references therein). In brief, given a crisp partition, with samples $\mathbf{x}_1^1, \dots, \mathbf{x}_{n_1}^1, \dots, \mathbf{x}_1^K, \dots, \mathbf{x}_{n_K}^K$ of sizes $n_1, \dots, n_K$ $(n_1 + \cdots + n_K = n)$ coming from groups $C_1, \dots, C_K$, the geometric variability, $V(C_k)$, of a group C_k is a general measure of dispersion, which reduces to the trace of the covariance matrix if the distance is the Euclidean, and a natural estimator is:

$$V(C_k) = \frac{1}{2n_k^2} \sum_{i,j \in C_k} \delta^2(\mathbf{x}_i^k, \mathbf{x}_j^k), \quad k = 1 \dots, K. \tag{1}$$

Given an observation $\mathbf{x}_0$, the proximity function of $\mathbf{x}_0$ to a group C_k represents the distance (squared) from $\mathbf{x}_0$ to C_k and is estimated by:

$$\phi^2(\mathbf{x}_0, C_k) = \frac{1}{n_k} \sum_{j \in C_k} \delta^2(\mathbf{x}_0, \mathbf{x}_j^k) - V(C_k), \quad k = 1 \dots, K. \tag{2}$$

Given two groups, C_m and C_l with $m \neq l$, the squared distance between them is estimated by:

$$\Delta^2(C_m, C_l) = \frac{1}{n_m n_l} \sum_{i \in C_m, j \in C_l} \delta^2(\mathbf{x}_i^m, \mathbf{x}_j^l) - V(C_m) - V(C_l), \quad m, l = 1 \dots, K. \tag{3}$$

A depth function based on these concepts was introduced in [23] and, for each observation $\mathbf{x}_0$, its depth value concerning group C_k is a value in [0,1], which indicates the depth degree of them with respect to the data cloud and it is estimated by:

$$I(\mathbf{x}_0, C_k) = \left[1 + \frac{\phi^2(\mathbf{x}_0, C_k)}{V(C_k)}\right]^{-1}, \quad k = 1 \dots, K. \tag{4}$$

Note that (4) takes into account both the relation of unit $\mathbf{x}_0$ with respect to the other units in the group and the dispersion of all data. As I is a depth function, it assigns to any observation $\mathbf{x}_0$ a degree of centrality, thus a large value of I, or equivalently a small value of $1/I$, suggests that $\mathbf{x}_0$ is more characteristic or typical of the group.

3. Method

This section details the new proposed approach, called Fuzzy Classification, based on Depth Function (FC-DF). First, based on the distances between observations, we provide fuzzy versions of the previously described concepts and some results when the Euclidean distance is used. Next, we develop the new supervised classification method and its adaptation for fuzzy unsupervised classification. The proposed method FC-DF is summarized in Algorithm 1 and the general process is plotted in Fig. 1.

3.1. Fuzzy versions

Given a fuzzy partition in K groups by vector memberships $\mathbf{u}_i \in \mathbb{R}^K$, $i = 1, \dots, n$, the fuzzy versions of the geometric variability, proximity function and distance between two groups can be defined in the following way:

- Fuzzy geometric variability of group C_k $(k = 1, \dots, K)$:

$$V_F(C_k) = \frac{1}{2\left(\sum_j u_{jk}\right)^2} \sum_{i,j} u_{ik} u_{jk} \delta^2(\mathbf{x}_i, \mathbf{x}_j). \tag{5}$$

- Fuzzy proximity function of observation $\mathbf{x}_0$ to C_k $(k = 1, \dots, K)$:

$$\phi_F^2(\mathbf{x}_0, C_k) = \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} \delta^2(\mathbf{x}_0, \mathbf{x}_j) - V_F(C_k). \tag{6}$$

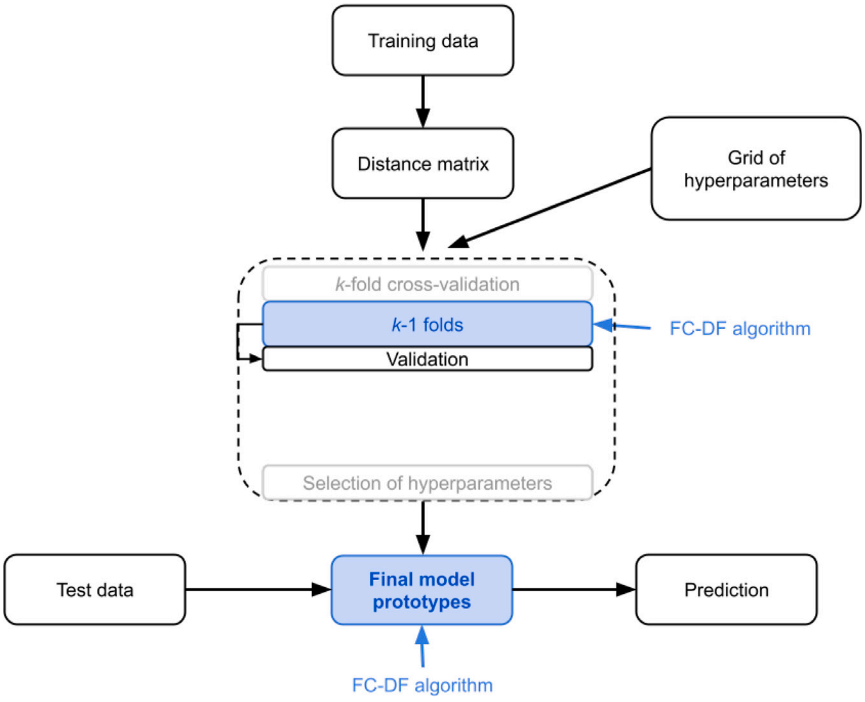

Fig. 1. Workflow of the general process followed to build the classifier.

- Fuzzy squared distance between two groups C_m and C_l ($m \neq l$, m, $l = 1\dots,K$):

$$\Delta^2_F(C_m,C_l) = \frac{1}{\sum_j u_{jm}\sum_j u_{jl}} \sum_{i,j} u_{im}u_{jl}\delta^2(\mathbf{x}_i,\mathbf{x}_j) - V_F(C_m) - V_F(C_l). \quad (7)$$

Once we have defined these concepts, the fuzzy depth function for each observation $\mathbf{x}_0$ concerning group C_k, ($k = 1,\dots,K$) is defined as:

$$I_F(\mathbf{x}_0,C_k) = \left(1 + \frac{\phi^2_F(\mathbf{x}_0,C_k)}{V_F(C_k)}\right)^{-1}. \quad (8)$$

All concepts in Eqs. (5) to (8) can be interpreted as the original concepts, but in a fuzzy context. Therefore, a small value of $I_F^{-1}(\mathbf{x}_0,C_k)$ signals out unit $\mathbf{x}_0$ as a central unit with respect to the fuzzy group C_k.

It is worth noting that all the expressions above are calculated based on the information of distances between pairs of units without the need for the coordinates. Nevertheless, if the coordinates of the observations are available, then the fuzzy center of each group is given by:

$$\mathbf{v}_k = \frac{\sum_{i=1}^n u_{ik}\mathbf{x}_i}{\sum_{i=1}^n u_{ik}}, \quad k = 1,\dots,K. \quad (9)$$

When δ is the Euclidean distance calculated on coordinates $\mathbf{x}_i$ for the units ($i = 1,\dots,n$), the following propositions hold (proofs are in Appendix A, B and C, respectively):

Proposition 1. *The fuzzy geometric variability of cluster C_k is the average squared Euclidean distance of each observation to the fuzzy center:*

$$V_F(C_k) = \frac{1}{\sum_j u_{jk}} \sum_i u_{ik}(\mathbf{x}_i - \mathbf{v}_k)'(\mathbf{x}_i - \mathbf{v}_k). \quad (10)$$

Proposition 2. *The fuzzy proximity function of observation $\mathbf{x}_0$ to C_k is the squared Euclidean distance between the observation and the fuzzy center:*

$$\phi^2_F(\mathbf{x}_0,C_k) = \|\mathbf{x}_0 - \mathbf{v}_k\|^2. \quad (11)$$

Proposition 3. *The fuzzy squared distance between two groups C_m and C_l is the squared Euclidean distance between the corresponding fuzzy centers:*

$$\Delta^2_F(C_m,C_l) = \|\mathbf{v}_m - \mathbf{v}_l\|^2. \quad (12)$$

3.2. Supervised Fuzzy classification

As a supervised approach, each observation $\mathbf{x}_i, i = 1,\dots,n$, in the training data set has its label y_i. We assume that there are M different labels ($y_i \in \{1,\dots,M\}$, $i = 1\dots,n$). Besides, the distances between each pair of observations $\mathbf{x}_i$ and $\mathbf{x}_j$ are also in the training data set $\mathcal{T} = \left\{(\delta_{i,j})^n_{i,j=1}, (y_i)^n_{i=1}\right\}$. Given K initial prototypes $\mathbf{a}_1,\dots,\mathbf{a}_K$ selected at random among the n units and their corresponding label-prototypes $\mathbf{z}_1,\dots,\mathbf{z}_K$, we propose, following the work of [22], a Supervised Fuzzy Partition based on the notion of Depth Function (SFP-DF), which aims to solve the following problem:

$$\min_{u_{ik},\mathbf{a}_k,\mathbf{z}_k} \sum_{k=1}^K\sum_{i=1}^n u_{ik}\delta^2(\mathbf{x}_i,\mathbf{a}_k) + \alpha\sum_{k=1}^K\sum_{i=1}^n u_{ik}l(y_i,\mathbf{z}_k) + \gamma\sum_{k=1}^K\sum_{i=1}^n u_{ik}log(u_{ik}), \quad (13)$$

subject to

$$\sum_{k=1}^K u_{ik} = 1, \quad u_{ik} \geq 0, \quad i = 1,\dots,n,\ k = 1,\dots,K,$$

where $l(y_i,\mathbf{z}_k) = -\sum_{m=1}^M log(z_{mk})\mathbb{1}(y_i = m)$, $\quad m = 1,\dots,M$, $\forall i,k$, and with positive hyperparameters $\gamma > 0$ and $\alpha \geq 0$.

The first term of (13) seeks for the deepest units, and the second term represents the within-cluster variability of the labels. As in [22], the positive hyperparameter α controls the contribution of the labels. When $\alpha = 0$ the label information of the units is not taken into account and it, therefore, offers a non-supervised version (Fuzzy Clustering). Smaller values of γ lead to crisper partitions.

Problem (13) is solved following the block coordinate descent approach presented in [22]. Within this frame, several smaller optimization problems are addressed in each iteration:

Block 1: Optimization of membership vectors. Considering the prototypes and label-prototypes fixed, problem (13) becomes:

$$\min_{u_{ik}} \sum_{k=1}^K\sum_{i=1}^n u_{ik}d_{ik} + \gamma\sum_{k=1}^K\sum_{i=1}^n u_{ik}log(u_{ik}), \quad (14)$$

subject to: $\sum_{k=1}^K u_{ik} = 1, \quad u_{ik} \geq 0, \quad i = 1,\dots,n,\ k = 1,\dots,K,$

and where distances to prototypes, as well as class label information, are considered in $d_{ik} = \delta^2(\mathbf{x}_i, \mathbf{a}_k) - \alpha \sum_{m=1}^{M} log(z_{mk}) \mathbb{1}(y_i = m)$. The solution for (14) is given by:

$$u_{ik} = \frac{\exp(-d_{ik}/\gamma)}{\sum_{l=1}^{K} \exp(-d_{il}/\gamma)}\ , \quad k = 1, \ldots, K,\ i = 1, \ldots, n. \tag{15}$$

Block 2: Optimization of prototypes. Considering the membership vectors and label-prototypes fixed, problem (13) becomes:

$$\min_{\mathbf{a}_1, \ldots, \mathbf{a}_K} \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ik} \delta^2(\mathbf{x}_i, \mathbf{a}_k)\,. \tag{16}$$

Then, (16) requires finding K units among the n units in the training data set that minimize $F(\mathbf{a}_1, \ldots, \mathbf{a}_K) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ik} \delta^2(\mathbf{x}_i, \mathbf{a}_k)$. Even for moderate values of n, it becomes infeasible to find the global optimum of F. Nevertheless, it is sensible to find one prototype $\mathbf{a}_k$ at a time, and work sequentially on each $k = 1, \ldots, K$. That is, for each k, we find the unit $\mathbf{a}_k \in \mathcal{T}$ that minimizes $F_k(\mathbf{a}_k) = \sum_{i=1}^{n} u_{ik} \delta^2(\mathbf{x}_i, \mathbf{a}_k)$.

It is interesting to note that $F_k(\mathbf{a_k}) \propto I_F^{-1}(\mathbf{a}_k, C_k)$, and therefore the prototypes are optimized looking for the deepest unit for each cluster C_k. Moreover, if, instead of prototypes $\mathbf{a}_k$, the fuzzy-centers $\mathbf{v}_k$ are considered as well as the Euclidean distance between units, then $F_k(\mathbf{v}_k) = \sum_i u_{ik} \|\mathbf{x}_i - \mathbf{v}_k\|^2$, linking to classical K-Fuzzy Clustering objective functions.

Block 3: Optimization of label-prototypes. Considering the membership vectors and prototypes fixed, problem (13) becomes:

$$\min_{\mathbf{z}_1, \ldots, \mathbf{z}_K} \left(- \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ik} \sum_{m=1}^{M} log(z_{mk}) \mathbb{1}(y_i = m) \right), \tag{17}$$

having as solution:

$$z_{mk} = \frac{\sum_i u_{ik} \mathbb{1}(y_i = m)}{\sum_{i=1}^{n} u_{ik}}\ , \quad m = 1, \ldots, M,\ k = 1, \ldots, K. \tag{18}$$

Note that each label-prototype $\mathbf{z}_k$ itself is a membership vector in the sense that the sum of its components is 1.

Putting together all the blocks, we have the following algorithm to solve (13) based on the training data set:

Algorithm 1 FC-DF algorithm.

```
Input: Distance matrix and labels $\mathcal{T} = \left\{ (\delta_{i,j})_{i,j=1}^{n}, (y_i)_{i=1}^{n} \right\}$,
       $K$ and hyperparameters $\alpha, \gamma$.
Output: membership vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$,
        prototypes $\mathbf{a}_1, \ldots, \mathbf{a}_K$,
        label-prototypes $\mathbf{z}_1, \ldots, \mathbf{z}_K$.
Initialize: prototypes $\mathbf{a}_1, \ldots, \mathbf{a}_K$,
            label-prototypes $\mathbf{z}_1, \ldots, \mathbf{z}_K$.
repeat
    Update distance between training units and prototypes
        $d_{ik} = \delta^2(\mathbf{x}_i, \mathbf{a}_k) - \alpha \sum_{m=1}^{M} log(z_{mk}) \mathbb{1}(y_i = m)$.
    Update membership vectors $\mathbf{u}_i$ (Block 1).
    Update prototypes as the deepest units (Block 2).
    Update label-prototypes (Block 3).
until Prototypes do not change.
```

3.3. Predicting the class of a new unit

Once the training phase is solved, given a new unit $\mathbf{x}$ and the squared distance from it to the prototypes $\delta^2(\mathbf{x}, \mathbf{a}_1), \ldots, \delta^2(\mathbf{x}, \mathbf{a}_K)$ follow these steps:

1. Compute its membership vector $\mathbf{u} = (u_1, \ldots, u_K)'$, which measures the affinity of $\mathbf{x}$ concerning each prototype, by:

$$\frac{\exp(-\delta^2(\mathbf{x}, \mathbf{a}_k)/\gamma)}{\sum_{l=1}^{K} \exp(-\delta^2(\mathbf{x}, \mathbf{a}_l)/\gamma)}\ , \quad k = 1, \ldots, K. \tag{19}$$

2. Compute weighted memberships relative to classes $1, \ldots, M$, that measure the affinity of $\mathbf{x}$ to the M classes, by:

$$p_m = \sum_{k=1}^{K} u_k z_{mk}\ , \ m = 1, \ldots, M. \tag{20}$$

As $\sum_{m=1}^{M} z_{mk} = 1$ and $\sum_{k=1}^{K} u_k = 1$ by construction, $\sum_{m=1}^{M} p_m = 1$.

3. Finally, predict its class label by

$$\arg \max_{m=1,\ldots,M} \{p_m\}. \tag{21}$$

3.3.1. Notes on initialization and tuning of the hyperparameters

The algorithm needs to initialize prototypes and label prototypes, and it could be interesting to select the prototypes spread along the space [24]. However, in this work, the initialization is carried out randomly. We repeat the initialization several times, considering the one that minimizes the objective function. Then,

1. Initialize K prototypes selecting at random K units ($K \geq M$) among the training data: $\mathbf{a}_1, \ldots, \mathbf{a}_K$.
2. Initialize K label-prototypes $\mathbf{z}_1, \ldots, \mathbf{z}_K$ taking into account labels of the prototypes $y_{\mathbf{a}_k}$, so that $z_{mk} = 1/(1 + (K-1)\epsilon)$ if $y_{\mathbf{a}_k} = m$ and $z_{mk} = \epsilon/(1 + (K-1)\epsilon)$ otherwise, ($\epsilon > 0$).

Besides, hyperparameters γ and α need to be tuned. When the number of hyperparameters to be tuned is large, randomly chosen values might be more efficient [25], but in this case, since there are only two, a grid-search approach is adequate. Therefore, as a general procedure to tune the hyperparameters throughout this work, first positive values for the hyperparameters are set for the grid, and then the most suitable values are selected according to the highest accuracy values. When several hyperparameter combination values reach the highest accuracy, we keep the first combination that reaches the maximum, but other practices could be considered as well. If there is no label information and the algorithm is applied as Fuzzy Clustering ($\alpha = 0$), γ is tuned based on a permutation approach related to the Gap statistics as in [10]. Since FC-DF is distance-based, the B permutations are performed on pairs of units, and the Gap is derived from the objective function $G(\gamma) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ik} \delta^2(\mathbf{x}_i, \mathbf{a}_k) + \gamma \sum_{k=1}^{K} \sum_{i=1}^{n} u_{ik} log(u_{ik})$, and therefore the Gap statistics becomes $Gap(\gamma) = \frac{1}{B} \sum_{b=1}^{B} log(G_b(\gamma)) - log(G(\gamma))$.

4. Data sets

The actual data that support the findings of this study are public except for the oil data set, which is not publicly available due to the policy of the industry but it is available, from the corresponding author, upon reasonable request. The simulated data can be reproduced from the explanation included.

All these public data sets are frequently used in classification approaches, and a summary of them is listed in Table 1. The data sets we use contain continuous data, mixed data (continuous and qualitative), or functional data derived from Near Infrared spectroscopy. We do not focus on huge data sets, instead, we focus on high-dimensional data, unbalanced classes, and different types of data because such data also plays a vital role in real classification problems.

Below is a brief description of each data set and the purpose they have been selected for.

4.1. Iris data set

The well-known Iris data set includes three classes (C_1 = Setosa, C_2 = Versicolor, C_3 = Virginica) with 50 data points in each class and 4 features (sepal length, sepal width, petal length, petal width). The aim of this example is mainly illustrative and shows that the proposed method is competitive when it is compared to other procedures. In addition, it shows that beyond a purely predictive approach, the fuzzy perspective offers the opportunity to obtain an idea of the data set. As the features are continuous, we used the Euclidean distance.

Table 1
Data set summary. p: number of features; n: number of units; M: number of classes, in brackets the number of units in each class; type of the features, and distance used for the FC-DF analysis.

Data set	p	n	M	Type	Distance
Mixture model	2	500	3 (125, 125,250)	continuous	Euclidean
Spiral	2	375	3 (125, 125, 125)	continuous	Euclidean
Iris	4	150	3 (50, 50, 50)	continuous	Euclidean
Alizadeh	2093	64	4 (21, 21, 9, 11)	continuous	correlation
Oil	1751	244	3 (107, 96, 41)	functional	first derivative
Cleveland	13	303	2 (137, 160)	mixed	related

4.2. Synthetic data sets

The two synthetic experiments aim to assess the flexibility of the supervised classification method and its stability in terms of accuracy. Again, we used the Euclidean distance.

4.2.1. Synthetic three-component mixture model

As in [22], we have simulated $n = 500$ random samples from the three-component mixture model with distribution $p(x) = 0.25p(x|y = 1) + 0.25p(x|y = 2) + 0.5p(x|y = 3)$ where:

- $X|y = 1 \sim N(\mu_1, \Sigma_1)$, with $\mu_1 = (0,0)'$ and $\Sigma_1 = diag(15, 0.05)$.
- $X|y = 2 \sim N(\mu_2, I)$, with $\mu_2 = (-12,0)'$ and I is the identity matrix.
- $X|y = 3 \sim \frac{2}{3}N(\mu_{31}, 4I) + \frac{1}{3}N(\mu_{32}, I)$ with $\mu_{31} = (0,8)'$ and $\mu_{32} = (0,-4)'$.

4.2.2. Spiral data set

Spiral data was generated with library KODAMA in R and $n = 375$ samples from 3 classes uniformly distributed were drawn.

4.3. Real data sets

With the following three data sets, we illustrate the good performance of both the unsupervised and the supervised classification new procedures. The aim is to show how the method works with high-dimensional data (data sets in which the number of features p is larger than the number of observations n, $p \gg n$) and with distances different from the Euclidean.

4.3.1. Alizadeh data set

To illustrate the FC-DF fuzzy clustering, we consider the gene expression data from adult lymphoid malignancies [26], widely used to illustrate cluster methodology [27,28]. We compared our results with those obtained using the well-known Fuzzy Clustering based on distances Fanny [29]. This data set used microarray to characterize gene expression patterns of the three most prevalent adult lymphoid malignancies: diffuse large B-cell lymphoma (DLBCL) separated into two groups DLBCL1 and DLBCL2, follicular lymphoma (FL) and chronic lymphocytic leukemia (CLL). The data set includes 64 patients (21 in DLBCL1 and DLBCL2; 9 in FL and 11 in CLL) and the expression level of 2,093 genes. In this case, the correlation distance was used as usual for this type of data.

4.3.2. Oil data set

We use this real example to show the usefulness of the supervised proposal with high dimensional data in which the Euclidean distance is not appropriate and another type of distance must be used. We compared our results with those obtained using SFP.

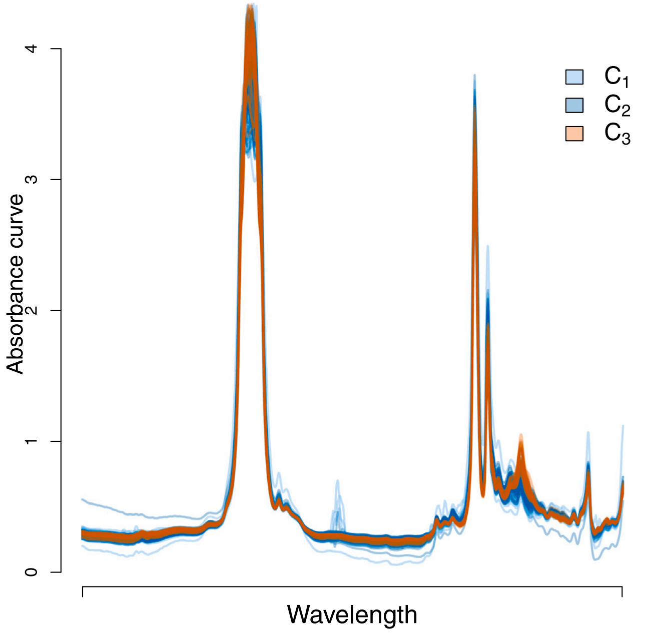

Fig. 2. Oil data set. Each oil, shown as a curve, is represented by 1,751 measurements of the absorbance at different wavelengths. Classification of oils is according to their alkaline reserve condition: good condition (C_1), fairly good condition (C_2), and warning condition (C_3).

The lubricant must be considered as a component of the machine, and the condition of the oil is fundamental to ensure its correct operation and to prevent potential damages. Given the degradation process that the oils suffer, it becomes appealing to consider the fuzzy approach to predict the condition of the oils, which is usually assessed by measuring the Base Number describing the level of reserve alkaline. In a different context, [30] also uses the infrared spectrum to classify alkanes in hydrocarbons. Fig. 2 shows 244 spectrometric curves of oils obtained from infrared spectroscopy and converted to absorbance curves that represent the amount of incident light absorbed by the oil sample. These oils were classified based on their alkaline reserve condition (107 are in good condition, 96 are in fairly good condition, and 41 are in warning). The detection of an oil in the warning condition is essential so that it can be replaced before engine damage occurs.

Here, we carried out two approaches. One, considering each observation i as a $p = 1,751$ dimensional vector $\mathbf{x}_i \in \mathbb{R}^p$, which is the classical approach cases × variables to represent the data and we applied the SFP method, as it is a fuzzy method that showed good performance on high-dimensional data [22]. Furthermore, as SFP calculates weights for the variables for each prototype, it could highlight interesting regions of wavelength to discriminate classes. The other approach was our proposed method. FC-DF allows the perspective that each oil is represented as a functional observation to be integrated, i.e., as an observation of a random curve. Particularly, we calculated the semi-metric distance based on the first derivative of their absorbance curves [31]. Under this approach, first B-splines of the curves are fitted, and then, given that the B-splines are smooth curves, the derivative of them is considered to compute the classical L^2-norm distance. For both approaches, we split the data randomly into train (90%) and test (10%) sets. To tune the hyperparameters, we proceed with a grid search along with a 10-fold cross-validation approach within the train set. Values in the grid for SFP were obtained following [22].

4.3.3. Cleveland data set

This heart disease data set [32] is composed of mixed data. There is quantitative data (age, pressure, cholesterol, ST depression induced by exercise relative to rest and maximum heart rate level) as well as binary (exercise induced angina, fasting blood sugar > 120 mg/dl) and qualitative (gender, resting electrocardiographic results, chest pain, the slope of the peak exercise ST segment, major vessels colored by fluoroscopy and thal). The data set contains the measures of these

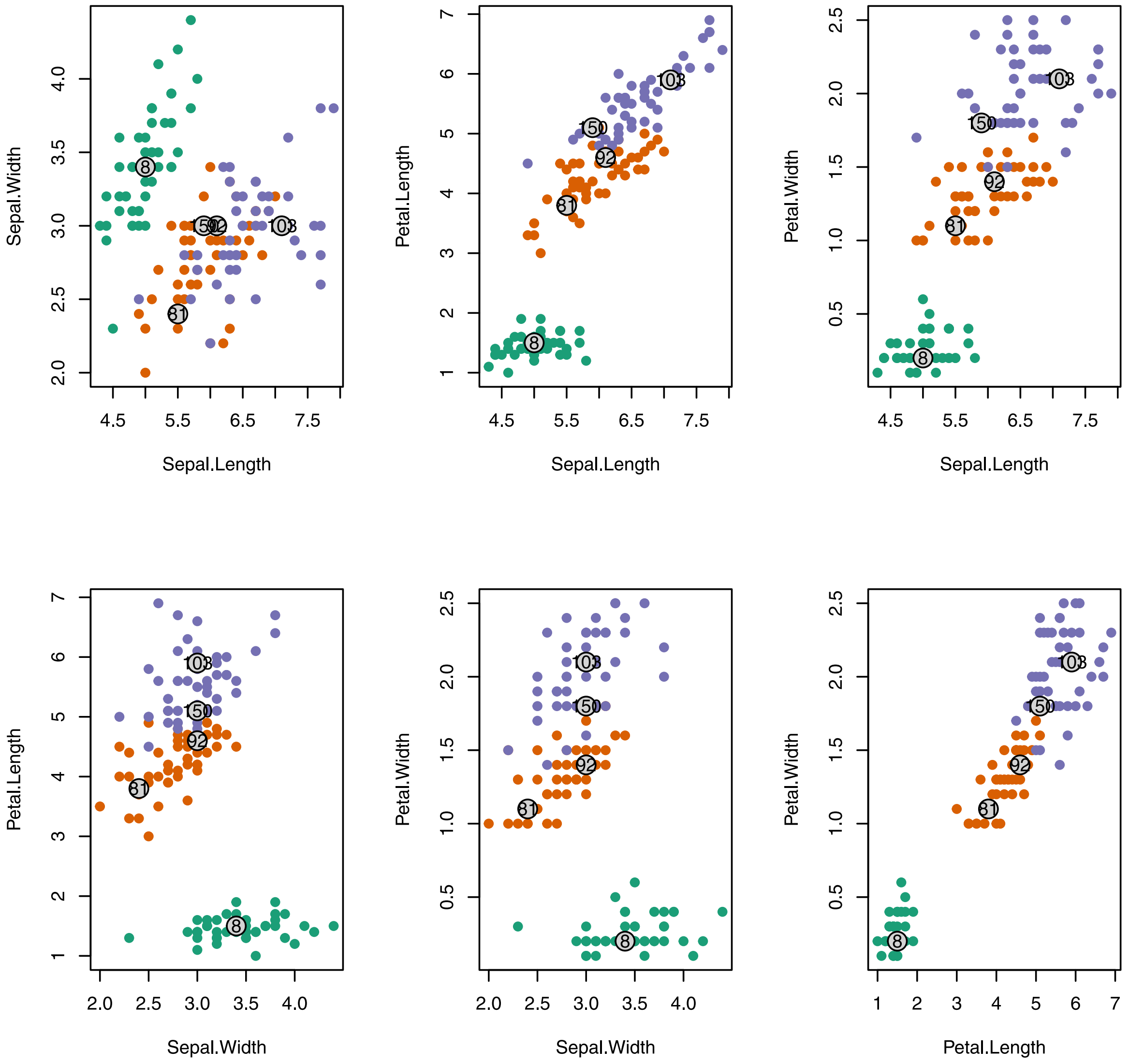

Fig. 3. Iris data set. Dispersion plots according to the four measured variables. Large circles indicate the $K = 5$ prototypes obtained with FC-DF. Setosa, Versicolor, and Virginica groups in green, orange, and purple, respectively.

variables on 303 units, however, 6 which present missing values, were removed. The aim is to predict the diagnosis of heart disease (less chance *vs* more chance). Besides, this data set presents the opportunity to show how a method that can use any distance allows the results to be improved. Because we work with mixed variables, as before, we carried out two approaches.

On the one hand, we codified the qualitative variables with indicator variables (i.e., one-hot-encoded) so that they can be considered as quantitative. Then the SFP method was applied. The other approach was our supervised proposed method. Since there is a mixture of different types of variables, we first separated the quantitative and the qualitative variables to pool them together later on as follows. First, the Euclidean distance was considered for quantitative (D_1) and the Gower's distance for qualitative (D_2) data. Then, we considered the related distance [33] obtained from distances D_1 and D_2, to get the distance that integrates both types of variables. Note that, as a general approach, the Gower distance may be appropriate for a combination of different types of variables. However, we have not used it because problems arise when the estimated range of a quantitative feature in the training split is shorter than that observed in the test split.

For both approaches, we split the data randomly in train (66.7%) and test (33.3%) sets. The quantitative variables were scaled according to their standard deviation in the training set. Again, to tune the hyperparameters, we proceed with a grid search along with a 10-fold cross-validation approach within the training set, and values in the grid for SFP follow [22].

For all the data sets, using a grid search approach, we determined the appropriate hyperparameters. To this end, training was done on 9 out of 10 folds of the training set, and the remaining fold was left as the validation set. The best values of the hyperparameters were chosen according to the results obtained in the validation set. In each situation, the initialization follows Section 3.3.1 and, to reduce randomness due to the haphazard initialization of prototypes, we repeated the analysis five times for each combination of hyperparameters. Then, we considered the best one.

All the experimentation was performed on a personal computer (Intel i5-7500, 4 cores, 3.50 GHz, 15 Gi of RAM and 2.0 Gi of swap space) running Ubuntu 20.04.6LTS. The code was implemented in R and it is available in repository https://github.com/rsait/FC-DF.

5. Results and discussion

This section analyzes the results and discusses the behavior of the proposed methodology.

Table 2
Iris data set. Number of prototypes, correct classification rates, and number of incorrect classified units obtained with different approaches.

Method	# Prototypes	Correct (%)	Incorrect classified units
Classical DA [17]	–	98.0	3
FDA [14]	–	89.3	16
KFDA [14]	–	93.3	10
Fuzzy DA [17]	–	89.3	16
Chang [34]	14	100	0
Dasarathy [35]	15	100	0
Bezdek [24]	11	100	0
SFP [22]	5	100/98.7[a]	0/2[a]
FC-DF	5	100/98.0[a]	0/3[a]

[a] Results obtained making predictions recalculating the membership vectors.

Table 3
Iris data set. Columns 2-4: non-fuzzy geometric variability, $V(C_m)$, calculated based on the crisp partition C_1, C_2, C_3 and fuzzy geometric variability, $V_F(C_m)$, based on weighted memberships relative to the 3 classes. Columns 5-9: fuzzy geometric variability based on the memberships relative to the 5 obtained prototypes.

	C_1	C_2	C_3	Prototypes				
				$\mathbf{a}_1 = 8$	$\mathbf{a}_2 = 81$	$\mathbf{a}_3 = 92$	$\mathbf{a}_4 = 103$	$\mathbf{a}_5 = 150$
$V(C_m)$	0.303	0.612	0.871	–	–	–	–	–
$V_F(C_m)$	0.303	0.624	0.809	0.303	0.319	0.310	0.563	0.286

5.1. Iris data set

Table 2 presents the correct classification obtained with FC-DF and different procedures, which consider the whole data set as a train and test set, reported in the literature. Following a grid-search approach to tune the hyperparameters ($\gamma, \alpha \in \{2^l \mid l = -4, -3.5, \dots, 0.5, 1\}$ and $K \in \{3, 4, \dots, 15\}$), the best accuracy rate for FC-DF is obtained with $\gamma = 0.0884$, $\alpha = 0.354$ and $K = 5$ prototypes. One of the prototypes belongs to the Setosa class, two to Versicolor, and the other two to Virginica (see Fig. 3).

Both SPF and FC-DF achieved a 100% accuracy when the predictions were based on the membership vectors $\mathbf{u}_i$ obtained at the end of the optimization procedure, being i a unit in the training set. Since the training and test sets are the same, strictly speaking, it is not necessary to recalculate the membership vectors following the prediction step (Section 3.3). However, we proceeded with this prediction step to have more robust results. For other procedures in Table 2, the information about the recalculation of the membership vectors is unknown. It is crucial to highlight, that although SFP gets a slightly higher accuracy, classifying correctly one more instance than FC-DF, the latter achieves practically the same accuracy with the same number of prototypes and using one less hyperparameter.

The obtained label prototypes $(\mathbf{z}_1, \dots, \mathbf{z}_5)$ show that each prototype is clearly from only one class. For instance, $\mathbf{z}_1 = (1, 0, 0)'$ indicates that prototype $\mathbf{a}_1$ (unit 8) is clearly from class C_1 (Setosa). The same happens for the rest of the prototypes: prototypes $\mathbf{a}_2$ (unit 81) and $\mathbf{a}_3$ (unit 92) are related to class C_2 (Versicolor) and prototypes $\mathbf{a}_4$ (unit 103) and $\mathbf{a}_5$ (unit 150) to class C_3 (Virginica).

Note that, in all, we have 3 partitions: two fuzzy partitions, one in $K = 5$ classes given by membership vectors $\mathbf{u}_i$ and the other in $M = 3$ classes given by the weighted membership $\mathbf{p}_i$; and the third one, the crisp partition determined by labels in classes C_1, C_2 and C_3. The membership vectors can be used to compute the fuzzy geometric variabilities. Particularly, it can be seen that there are no differences between the classical geometric variability and the fuzzy version for class C_1 ($V(C_1) = V_F(C_1) = 0.303$). Besides, the fuzzy geometric variability related to prototype $\mathbf{a}_1 = 8$ of class C_1 remains the same, proving that the prototype captures the variability around it well. For classes C_2 and C_3 small differences between classical and fuzzy versions appear, showing there is some fuzziness (see Table 3). Concerning fuzzy geometric variabilities, it can be seen that prototypes $\mathbf{a}_2 = 81$ and $\mathbf{a}_3 = 92$ account for the variability of C_2 in two halves. However, for class C_3, the class with the greater variability, the variability around prototype $\mathbf{a}_4 = 103$ is bigger (0.563) than the variability around $\mathbf{a}_5 = 150$ (0.286).

Furthermore, the entropies related to the weighted membership vectors, $\mathbf{p}_i$, $i = 1, 2, 3$, relative to classes help to measure the fuzziness of the units. In Fig. 4 the horizontal lines, from top to bottom, correspond to the values of the entropy for four different distributions: uniform distribution $\mathbf{f}_1 = (1/3, 1/3, 1/3)$ with entropy $H(\mathbf{f}_1) = 1.58$; distribution $\mathbf{f}_2 = (0, 1/2, 1/2)$ with $H(\mathbf{f}_2) = 1$; distributions $\mathbf{f}_3 = (0, 1/4, 3/4)$ and $\mathbf{f}_4 = (0, 3/4, 1/4)$ both with $H(\mathbf{f}_3) = H(\mathbf{f}_4) = 0.81$. It can be seen that any unit has a fuzziness close to a uniform distribution $\mathbf{f}_1$, five units are in between the entropy values of distributions $\mathbf{f}_2$ and $\mathbf{f}_3$, and some with lower entropy than $H(\mathbf{f}_3)$ but never null. All units in C_1 have null entropy and, therefore, their classification is not doubtful.

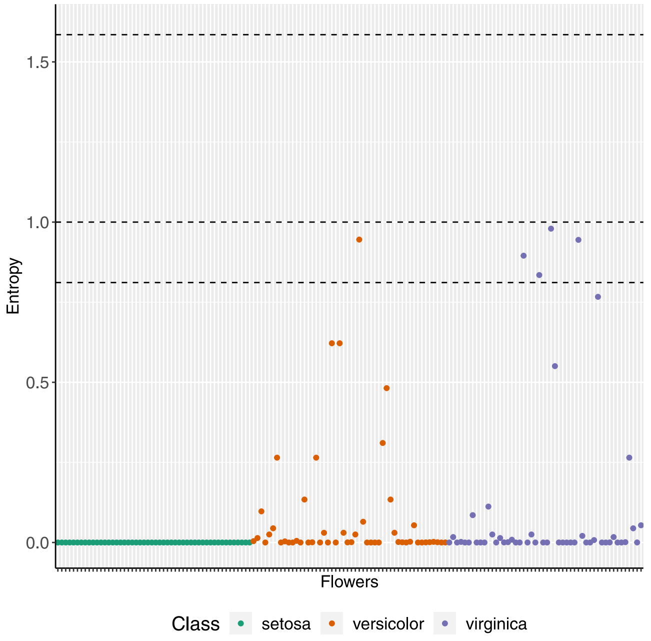

Fig. 4. Iris data set. For each flower, the entropy of its weighted membership vector is shown. The horizontal lines correspond to the entropy for different distributions.

5.2. Synthetic three-component mixture model

Fig. 5 shows the precision obtained for the different values of the hyperparameters, whose adjustment was achieved by grid search with $\gamma, \alpha \in \{2^l \;:\; l = -1, -0.5, \dots, 3.5, 4\}$. It can be observed that the range for γ values was wide enough (Fig. 5, right), as well as the risk of choosing a too-large value for α (Fig. 5, left). The hyperparameter α is related to the labels and considering a value which is too big could lead to overfitting. Concerning the number of prototypes, with only 3 prototypes just approximately 80% of accuracy is reached since only one prototype falls into class 2, so the units of the two subgroups that comprise this class are not well characterized. However, with 4 prototypes, one belongs to class 1, one to class 3, and two to class 2 (one in each subgroup), an accuracy equal to 97% is achieved in both the training group and in the test set. When considering 5 or more prototypes, these accuracy values are improving, since the dispersion of the data is covered better. Concerning the number of prototypes as a general trend, it can be seen that $K = 3$ might not be enough and that having more prototypes raises the classification accuracy.

For each number of prototypes K and to get a tuned FC-DF with appropriate values of hyperparameters, we considered values related to maximum accuracy. Then, the performance offered by the tuned FC-DF

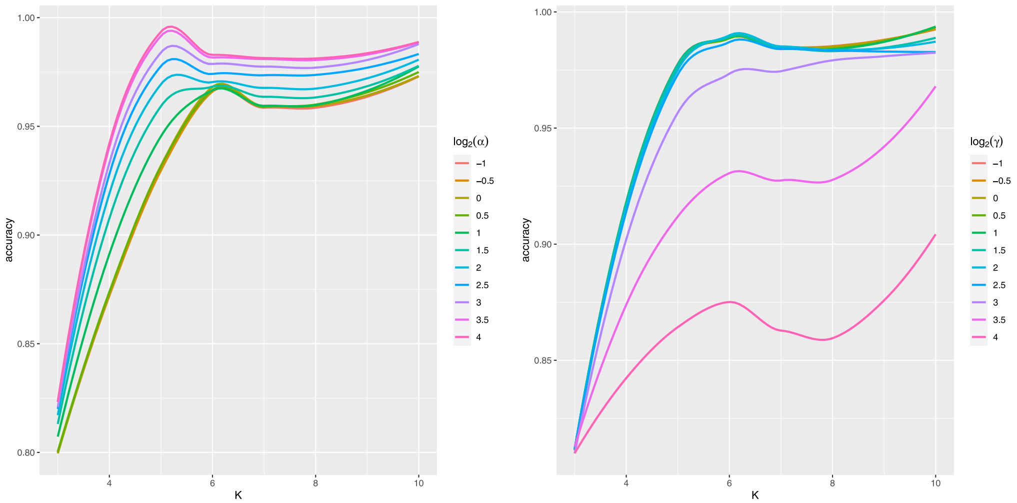

Fig. 5. Synthetic three component data set. Each colored line is the *loess* smoothing curve of the 10-fold cross-validation accuracy against the number of prototypes obtained using FC-DF and for each value of the hyperparameters. Left: accuracy versus K given α. Right: accuracy versus K given γ.

Table 4
Synthetic three component data set. For each K, values of γ and α with best accuracy values using the training data. Column 5, the accuracy obtained on the test set using these hyperparameters.

K	γ	α	Accuracy	
			Evaluation	Test
3	4.00	0.50	0.81	0.82
4	1.00	4.00	0.97	0.97
5	0.50	2.83	0.99	0.97
6	2.00	0.50	0.99	0.98
7	2.83	0.50	0.99	0.98
8	1.00	0.50	0.99	0.98
9	0.50	8.00	0.99	0.99
10	1.41	0.50	0.99	0.99

Table 5
Synthetic three component data set. After repeating the algorithm 5 times, running times (seconds) with respect to the sample size in the training set.

	Sample size				
	500	1000	5000	10000	15000
Running times (s)	1.9	8.2	213.5	760.2	1729.1

Table 6
Spiral data set. For each K, values of γ and α with best accuracy values obtained using the training data (columns 1 to 4 and 6 to 9). In columns 5 and 10, accuracies obtained on the test set with the selected hyperparameter values.

K	γ	α	Evaluation	Test	K	γ	α	Evaluation	Test
3	2	4	0.54	0.56	12	2	4	0.91	0.85
4	4	0.25	0.54	0.55	13	2	4	0.92	0.88
5	1	0.25	0.57	0.56	14	2	4	0.92	0.92
6	4	4	0.60	0.62	15	0.5	4	0.93	0.94
7	0.5	2	0.64	0.61	16	0.25	4	0.95	0.96
8	1	4	0.68	0.75	17	2	4	0.96	0.94
9	2	4	0.79	0.84	18	0.25	1	0.96	0.92
10	2	4	0.82	0.85	19	1	1	0.97	0.95
11	1	4	0.86	0.84	20	1	1	0.97	0.94

was assessed on the test set, containing 500 new samples drawn from the same model. The obtained results can be found in Table 4, where for each K the values of hyperparameters γ and α with best accuracy values obtained in the validation approach using the training data are shown in the first four columns. The accuracy values obtained on the test set with the selected hyperparameter values are in the last column. We can observe that the 10-fold cross-validation approach carried out within the training set leads to stable values of the hyperparameters, as the accuracy in the test set is very similar to the accuracy obtained in the training set. Fig. 6 illustrates the solutions with $K = 4$ and $K = 10$ prototypes. Note that the method adequately selects the prototypes, considering more prototypes where the spread of the units is wider. These results demonstrate the correct performance of the FC-DF procedure and its ability to predict the class of new units.

Furthermore, as it is a distance-based method, it is necessary to mention the difficulties that arise with its scalability. Precisely, difficulties related to the computation time and storage of the distance matrix. The method needs to compute the distances between every pair of units in a set of n units, making running times and memory needs grow quadratic $O(n^2)$. We generated samples of different sizes ($n = 500, 1000, 5000, 10000, 15000$) and computed the algorithm sequentially. The algorithm is repeated 5 times as mentioned at the beginning of the paragraph. The observed running time can be seen in Table 5. Fig. 7 shows the running times (s) and storage need (Mb) according to the number of samples in the training set.

5.3. Spiral synthetic data

We proceeded in a similar way to tune the hyperparameters and we generated 375 new samples to create a test set. The results obtained for the behavior of the hyperparameters are very similar to those commented for the previous data set, the accuracy being higher as K gets bigger and obtaining the stability of the accuracy in the train and test sets. FC-DF can deal very well with the linear non-separable problem (Table 6), with an accuracy greater than 90%, and it has the ability to place the prototypes in such a way that the entire spiral shape presented by the data is covered (Fig. 8). This experiment demonstrates an accurate performance of FC-DF in front of non-linear situations.

5.4. Alizadeh data set

We applied FC-DF as Fuzzy Clustering, with $\alpha = 0$. Hyperparameter γ is tuned based on a permutation approach related to the Gap statistics as in [10]. Within this approach, $B = 100$ times permutation was performed in pairs of units. The number of clusters K were selected based on the Fuzzy Silhouette [36]. This index suggested $K = 3$ clusters for Fanny (membership exponent $r = 1.2$) while FC-DF indicates $K = 2$ clusters ($\gamma = 2^{-9}$), although the silhouette values were low (0.318 and 0.185, respectively). Both methods tend to classify CLL and FL classes together in one cluster. This was expected, as the gene expression profiles of DLBCL are very different from those of CLL and FL [26]. Moreover, FC-DF classified DLBCL1 and DLBCL2 together, while the

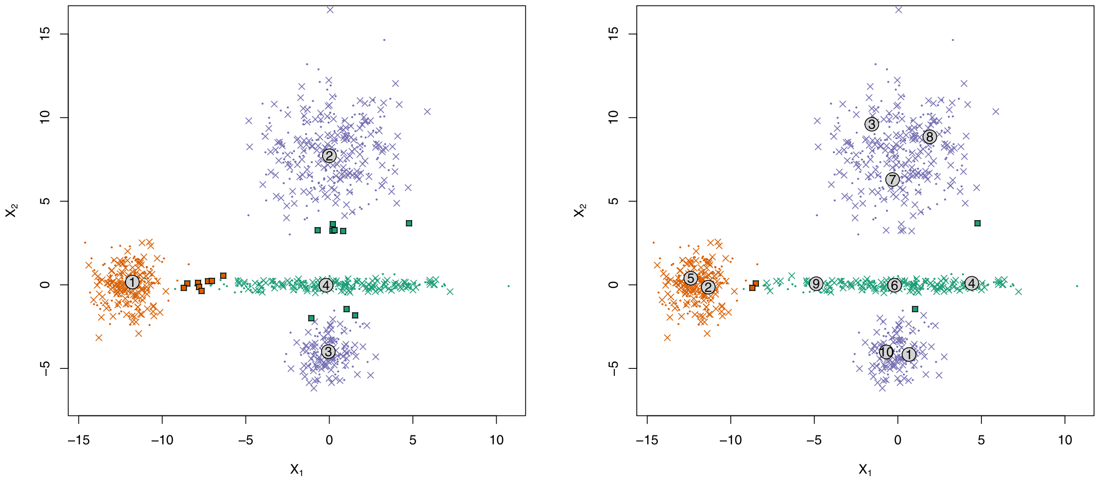

Fig. 6. Synthetic three component data set. Each color indicates a class. Training data set in small circles. Test data set in crosses, when the predicted label is correct; otherwise squares. Prototypes are indicated by large circles. Left: $K = 4$ prototypes. Right: $K = 10$ prototypes.

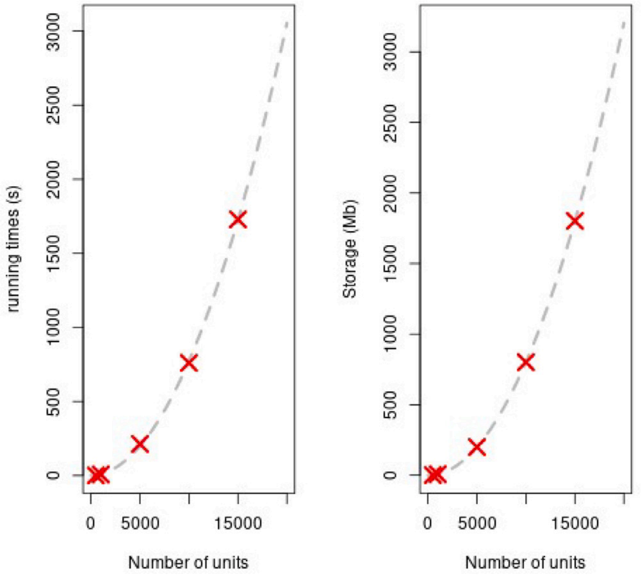

Fig. 7. Running times (s) and Storage need (Mb) according to the number of samples in the training set.

Table 7
Alizadeh data set. For both FC-DF and Fanny, confusion matrices with the number of clusters pointed by the Silhouette index (left), and considering four groups of patients (right).

Real	FC-DF		Fanny			FC-DF				Fanny			
	C_1	C_2	C_1	C_2	C_3	C_1	C_2	C_3	C_4	C_1	C_2	C_3	C_4
CLL	11	0	11	0	0	11	0	0	0	11	0	0	0
DLBCL1	1	20	1	13	7	0	2	14	5	0	2	11	8
DLBCL2	2	19	0	11	10	1	0	10	10	0	1	10	10
FL	9	0	9	0	0	0	9	0	0	0	9	0	0

Fanny tried to separate them, but did not achieve a good partition (Table 7, left side).

As we know that there are four groups of patients, we also compared the solutions obtained for $K = 4$ with $\gamma = 2^{-1}$ for FC-DF and membership exponent $r = 1.2$ for Fanny (Table 7, right side). Both methods correctly split the CLL and FL classes. However, neither method achieves a good separation of the DLBCL1 and DLBCL2 patients, this might be due to the considerable molecular heterogeneity within the DLBCL group [26]. Nevertheless, FC-DF obtains a better separation between them. Besides the confusion matrix, the fuzzy squared distance also points in that direction. Identifying the clusters C_3 and C_4 with their closest real class DLBCL1 and DLBCL2, respectively, the fuzzy squared distance with FC-DF is $\Delta_F^2(C_3, C_4) = 0.0064$; Surprisingly the fuzzy squared distance for Fanny is $\Delta_F^2(C_3, C_4) = 0$, and therefore the hypothetical fuzzy center is the same for clusters C_3 and C_4. Membership vectors of the obtained prototypes by FC-DF also suggest that there is a principal separation between clusters $C_1 \cup C_2$ and $C_3 \cup C_4$, the first two related to CLL and FL and the other two to the different forms of DLBCL, as expected (Table 8.)

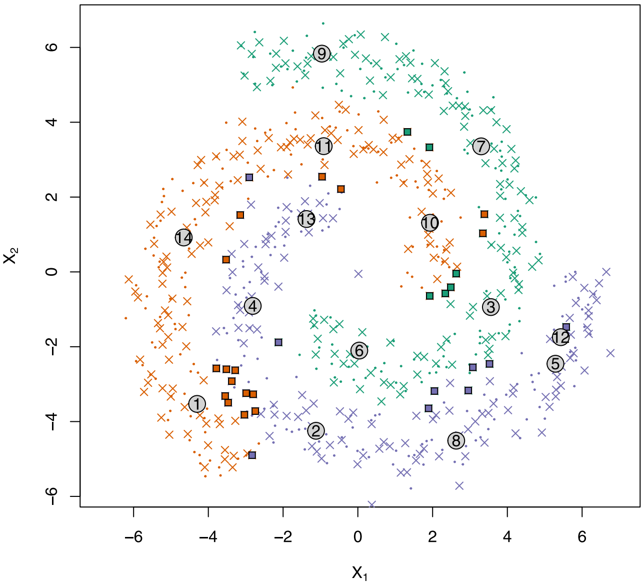

Fig. 8. Spiral data set. Each color indicates a class. Training data set in small circles. Test data set in crosses, when the predicted label is correct; otherwise squares. $K = 14$ prototypes (indicated by large circles).

5.5. *Real oil data set*

After a grid search ($\gamma \in \{2^{-10}, 2^{-9}, \dots, 2^{-2}\}$ and $\alpha \in \{2^{-2}, 2^{-1.5}, \dots, 2^{2}\}$) and according to the best accuracy value obtained on the validation set, the best solutions for SFP and FC-DF are obtained with $K = 10$ ($\gamma = 1.37$, $\alpha = 1.27$ and $\lambda = 1.11$) and $K = 3$ ($\gamma = 2^{-9}$ and $\alpha = 2^{-2}$) prototypes, respectively.

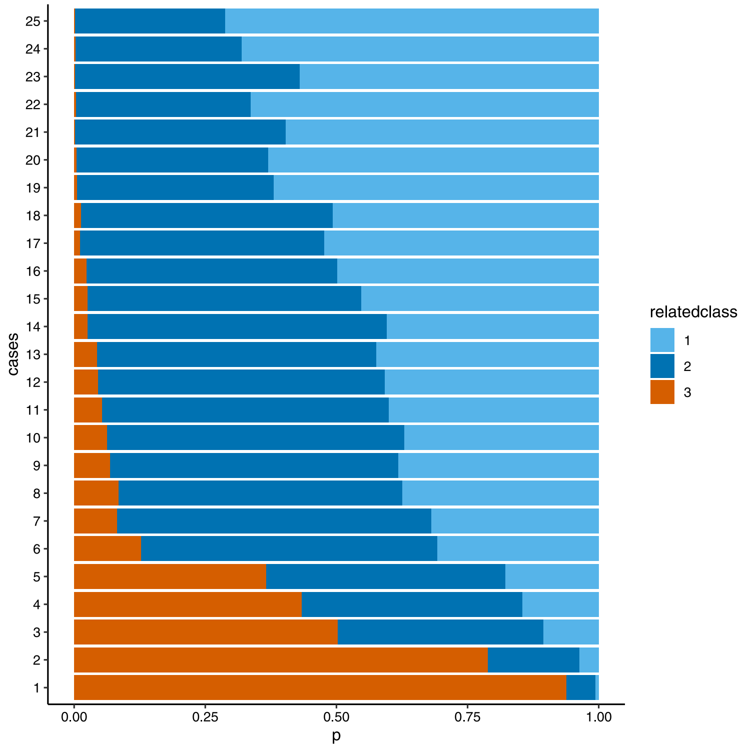

Fig. 9. Oil data set. Weighted membership vectors **p** for oils in the test set, the colors indicate the three classes (class 1: good condition; class 2: fairly good condition; class 3: warning condition).

Table 8
Alizadeh data set. Membership vectors related to clusters obtained for $K = 4$ with FC-DF.

Prototypes	p_1	p_2	p_3	p_4
$\mathbf{a}_1$ (case 6)	0.671	0.259	0.031	0.039
$\mathbf{a}_2$ (case 58)	0.254	0.659	0.045	0.042
$\mathbf{a}_3$ (case 29)	0.033	0.049	0.711	0.207
$\mathbf{a}_4$ (case 26)	0.041	0.045	0.206	0.708

Table 9
Oil data set. For each K, best accuracy values obtained using the train data. With the corresponding parameters, accuracy obtained on the test set for both SFP and FC-DF procedures. In bold, best accuracy values on the validation set.

K	SFP		FC-DF	
	Validation	Test	Validation	Test
3	0.48	0.44	**0.71**	0.80
4	0.46	0.20	0.65	0.60
5	0.47	0.52	0.71	0.68
6	0.46	0.52	0.69	0.68
7	0.47	0.44	0.69	0.56
8	0.47	0.44	0.68	0.60
9	0.47	0.56	0.67	0.68
10	**0.49**	0.44	0.66	0.72

The obtained results (Table 9) clearly show the benefits of taking into account the characteristics of the particular data and the use of a distance according to the data type. Besides, the solution given by SFP was not able to predict any oil in the warning class (C_3). Given that SFP calculates weights for the features and for each prototype, it could point out interesting regions of wavelength to discriminate classes. Nevertheless, the obtained weights were uniformly distributed and did not point to any wavelength of interest to help discriminate the conditions of the oils. Concerning FC-DF, all oils in the warning class were well classified and the information given by the weighted membership vectors is particularly interesting (Fig. 9). For instance, it is worth looking for oils with a membership degree bigger than 0.15 (half of what is expected under the uniform distribution) for class warning (class 3), in order to raise an alarm before any damage occurs. If so, although oils 6, 7, and 8 are in fairly good condition (class 2), they are signaled out to move to warning condition (class 3). This might be relevant information for the managers of such engines.

5.6. Cleveland data set

After a grid search ($\gamma \in \{2^{-4}, 2^{-3.5}, \dots, 2^{1}\}$ and $\alpha \in \{2^{-6}, 2^{-5.5}, \dots, 2^{1}\}$) and according to the best accuracy value obtained on the validation set, the best solutions for SFP and FC-DF were obtained (Table 10) with $K = 9$ ($\gamma = 1.68$, $\alpha = 1.04$ and $\lambda = 1.37$) and $K = 3$ ($\gamma = 2^{-2}$ and $\alpha = 2^{-2}$) prototypes, respectively. The results, again, show the benefits of using an adequate distance according to the type of data. FC-DF is more accurate in classifying and, moreover, it is more robust in general terms since the differences in the assessment between the validation split and test split are clearly smaller for FC-DF.

Table 10
Cleveland data set. For each K, best accuracy values obtained using the train data. With the corresponding parameters, accuracy obtained on the test set for both SFP and FC-DF procedures. In bold, the best accuracy values on the validation set.

K	SFP		FC-DF	
	Validation	Test	Validation	Test
2	0.712	0.535	0.813	0.828
3	0.742	0.556	**0.854**	0.838
4	0.763	0.697	0.848	0.788
5	0.783	0.576	0.854	0.859
6	0.783	0.717	0.854	0.859
7	0.783	0.848	0.854	0.818
8	0.808	0.596	0.854	0.828
9	**0.813**	0.768	0.854	0.818
10	0.798	0.657	0.854	0.859

6. Conclusions and future work

FC-DF is a useful and competitive method since it has a number of advantages over other methods. For instance, it is independent of the type of data as it is only necessary to select a suitable distance for each specific data set. For this reason, the procedure becomes more general than those that are only valid with continuous features and need to define centroids. Another strength is that, as FC-DF works with the distances between units, the number of variables is not relevant. Thus, it works without any type of restriction with high dimensional data sets. The use of an appropriate distance can prevent the curse of dimensionality that may arise with high dimensional data. The obtained results with different types of real data, where the number of units n is smaller than the number of features p, show how the method when using an appropriate distance overcomes such a difficulty. On the basis of the above, the overall performance of FC-DF is significantly improved than that of SFP. While distance-based methods offer a good alternative to deal with different types of data, they also have an intrinsic computational issue related, precisely, to the distance matrix they are based on. Similar to kernel methods, the computation and storage of the distance matrices are the bottleneck to scaling them up easily. As execution times and required memory grow quadratically, the user must take this limitation into account, evaluating the feasibility of the method when applying it to their own data sets with a large number of units. However, the method does not present any limitation if the dimension is large. In that sense, the potential user should check only that the number of units at hand is not huge to prevent computational limits. Although this can be considered a limitation of the proposed method, there are many real situations where the number of samples is not necessarily huge. For example, in Biomedicine, the number of cases is relatively small on many occasions (especially for rare diseases) but the variables (e.g., genes) are the ones that increase dramatically. Although we live in a time of great data consumption, it should not be forgotten that not all studies can have this enormous amount of data available and, therefore, the development of an adequate methodology for more modest data sets should not be neglected. In any case, future work should aim to find ways to deal with large distance matrices. As R is an interpreted language, memory management is not as efficient as in other compiled languages. Thus, as a first step, it could be necessary to implement the algorithm in a compiled language, such as C. In addition, the option of making the prediction and classification based on partially observed distances should be explored. The method has hyperparameters that need to be tuned. To this end, a grid search is appropriate. So, when the approach is supervised, the selection of the hyperparameter is made according to an adequate metric (accuracy rate, for instance) reached on a k-fold cross-validation setting. The metric to measure the goodness of the hyperparameters should be considered depending on the particular characteristics of the data set. When the approach is non-supervised there is a lack of an external validation variable and, as an internal validation approach, a permutation approach related to the Gap statistics leads to good results. Throughout this work, we derived the Gap statistic based on the main objective function. We think that other means could be explored to derive the Gap statistics, for instance, based on the concept of fuzzy squared distance. Nevertheless, this is beyond the scope of this work. Regarding hyperparameters, it would be very challenging to find ways to offer default values for the hyperparameters that could give reasonably good results. The behavior of FC-DF in simulated and real data was evaluated, verifying its usefulness and competitiveness with other methods. For all these reasons, FC-DF can be a very useful tool for researchers from different current and top leading fields of knowledge.

CRediT authorship contribution statement

Itziar Irigoien: Conceptualization, Methodology, Writing – original draft, Software, Supervision, Validation, Writing – review & editing. **Susana Ferreiro:** Oil data writing – review. **Basilio Sierra:** Supervision, Validation, Writing – review & editing. **Concepción Arenas:** Conceptualization, Methodology, Writing – original draft, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Basilio Sierra reports financial support was provided by Spanish Ministerio deEconomia yCompetitividad.

Data availability

Data are public except the oil data set which is not publicly available due to our industry's policy but is available from the corresponding author upon reasonable request.

Acknowledgments

We express our sincere thanks to the Editor and five reviewers, whose comments and suggestions led to substantial improvements in the organization and clarity of the paper. We also want to thank J.M. Otzeta for his valuable help building the docker container image.

This research was partially supported: II by the Spanish 'Ministerio de Economia y Competitividad' (PID2019-106942RB-C31). CA by grant 2021SGR01421 (GRBIO) from the Departament de Economia i Coneixement de la Generalitat de Catalunya, Spain. II, CA and BS by the Spanish 'Ministerio de Economia y Competitividad' (PID2021-122402OB-C21).

Appendix A. Proof of Proposition 1

Taking into account that δ is the Euclidean distance between coordinates $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^p$ $(i, j = 1, \dots, n)$ and that $\mathbf{v}_k = \sum_i u_{ik}\mathbf{x}_i / \sum_i u_{ik}$ is the fuzzy center of group C_k $(k = 1, \dots, K)$:

$$
\begin{aligned}
\sum_{i=1}^{n} u_{ik} \|\mathbf{x}_i - \mathbf{v}_k\|^2 &= \sum_{i=1}^{n} u_{ik} \|\mathbf{x}_i - \frac{1}{\sum_j u_{jk}} \sum_{j=1}^{n} u_{jk}\mathbf{x}_j\|^2 \\
&= \sum_{i=1}^{n} u_{ik} \| \frac{1}{\sum_j u_{jk}} \sum_{j=1}^{n} u_{jk}(\mathbf{x}_i - \mathbf{x}_j)\|^2 \\
&= \frac{1}{\left(\sum_j u_{jk}\right)^2} \sum_{i=1}^{n} u_{ik} \sum_{d=1}^{p} \left(\sum_{j=1}^{n} u_{jk}(x_{id} - x_{jd}) \right)^2 \\
&= \frac{1}{\left(\sum_j u_{jk}\right)^2} \sum_{d=1}^{p} \sum_{i} u_{ik} \Bigg(\sum_{j} u_{jk}^2 (x_{id} - x_{jd})^2 \\
&\quad + 2 \sum_{1 \le j < l \le n} u_{jk} u_{lk} (x_{id} - x_{jd})(x_{id} - x_{ld}) \Bigg) \\
&= \frac{1}{\left(\sum_j u_{jk}\right)^2} \sum_{d=1}^{p} \Bigg(\sum_{i} \sum_{j} u_{ik} u_{jk}^2 (x_{id} - x_{jd})^2 \\
&\quad + \sum_{1 \le j < l \le n} u_{ik} u_{jk} u_{lk} \left[(x_{id} - x_{jd})^2 \right. \\
&\quad + \left. (x_{id} - x_{ld})^2 + (x_{jd} - x_{ld}^2) \right] \Bigg) \\
&= \frac{1}{\left(\sum_j u_{jk}\right)^2} \sum_{d=1}^{p} \Bigg(\sum_{1 \le i < j \le n} (u_{ik} u_{jk}^2 + u_{jk} u_{ik}^2)(x_{id} - x_{jd})^2 \\
&\qquad\qquad + \sum_{1 \le i < j < l \le n} u_{ik} u_{jk} u_{lk} \left[(x_{id} - x_{jd})^2 \right. \\
&\quad + \left. (x_{id} - x_{ld})^2 + (x_{jd} - x_{ld}^2) \right] \Bigg)
\end{aligned}
$$

$$= \frac{1}{\left(\sum_j u_{jk}\right)^2} \sum_{d=1}^{p} \left(\sum_{1 \le i < j \le n} u_{ik} u_{jk} \left(\sum_l u_{lk} \right) (x_{id} - x_{jd})^2 \right)$$

$$= \frac{1}{\sum_j u_{jk}} \sum_{1 \le i < j \le n} u_{ik} u_{jk} (\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j)$$

$$= \frac{1}{\sum_j u_{jk}} \sum_{1 \le i < j \le n} u_{ik} u_{jk} \delta^2(\mathbf{x}_i, \mathbf{x}_j) = V_F(C_k).$$

Appendix B. Proof of Proposition 2

$$\begin{aligned}
\phi_F^2(\mathbf{x}_0, C_k) &= \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} \delta^2(\mathbf{x}_0, \mathbf{x}_j) - V_F(C_k) \\
&= \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} (\mathbf{x}_j - \mathbf{x}_0)'(\mathbf{x}_j - \mathbf{x}_0) \\
&\quad - \frac{1}{2\left(\sum_j u_{jk}\right)^2} \sum_{i,j} u_{ik} u_{jk} (\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j) \\
&= \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} (\mathbf{x}_j - \mathbf{v}_k + \mathbf{v}_k - \mathbf{x}_0)'(\mathbf{x}_j - \mathbf{v}_k + \mathbf{v}_k - \mathbf{x}_0) \\
&\quad - \frac{1}{\sum_j u_{jk}} \sum_i u_{jk} (\mathbf{x}_j - \mathbf{v}_k)'(\mathbf{x}_j - \mathbf{v}_k) \\
&= \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} (\mathbf{v}_k - \mathbf{x}_0)'(\mathbf{v}_k - \mathbf{x}_0) = \|\mathbf{x}_0 - \mathbf{v}_k\|^2.
\end{aligned}$$

$\phi_F^2(\mathbf{x}_0, C_k) = \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} \delta^2(\mathbf{x}_0, \mathbf{x}_j) - V(C_k) = \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} (\mathbf{x}_j - \mathbf{x}_0)'(\mathbf{x}_j -$
$\mathbf{x}_0) - \frac{1}{2\left(\sum_j u_{jk}\right)} \sum_{i,j} u_{ik} u_{jk} (\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j) = \frac{1}{\sum_j u_{jk}} \sum_j u_{jk} (\mathbf{x}_j - \mathbf{a}_k +$
$\mathbf{a}_k - \mathbf{x}_0)'(\mathbf{x}_j - \mathbf{a}_k + \mathbf{a}_k - \mathbf{x}_0) - \frac{1}{2\left(\sum_j u_{jk}\right)^2} \sum_{i,j} u_{ik} u_{jk} (\mathbf{x}_i - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{x}_j) =$
$\frac{1}{\sum_j u_{jk}} \sum_j u_{jk} (\mathbf{x}_j - \mathbf{a}_k + \mathbf{a}_k - \mathbf{x}_0)'(\mathbf{x}_j - \mathbf{a}_k + \mathbf{a}_k - \mathbf{x}_0) - \frac{1}{\left(\sum_j u_{jk}\right)} \sum_i u_{ik} (\mathbf{x}_j -$
$\mathbf{a}_k)'(\mathbf{x}_j - \mathbf{a}_k)$

Appendix C. Proof of Proposition 3

$$\begin{aligned}
\Delta_F^2(C_k, C_l) &= \frac{1}{\sum_j u_{jk} \sum_j u_{jl}} \sum_{i,j} u_{ik} u_{jl} \delta^2(\mathbf{x}_i, \mathbf{x}_j) - V_F(C_k) - V_F(C_l) \\
&= \frac{1}{\sum_j u_{jk} \sum_j u_{jl}} \sum_{i,j} u_{ik} u_{il} (\mathbf{x}_i - \mathbf{v}_k + \mathbf{v}_k - \mathbf{v}_l + \mathbf{v}_l - \mathbf{x}_j)'(\mathbf{x}_i - \mathbf{v}_k \\
&\quad + \mathbf{v}_k - \mathbf{v}_l + \mathbf{v}_l - \mathbf{x}_j) - V_F(C_k) - V_F(C_l) \\
&= \frac{1}{\sum_j u_{jk} \sum_j u_{jl}} \sum_{i,j} u_{ik} u_{jl} \|\mathbf{v}_k - \mathbf{v}_l\|^2 \\
&= \frac{1}{\sum_j u_{jk} \sum_j u_{jl}} \sum_i u_{ik} \sum_j u_{jl} \|\mathbf{v}_k - \mathbf{v}_l\|^2 = \|\mathbf{v}_k - \mathbf{v}_l\|^2.
\end{aligned}$$

References

[1] A. Orcan, D. Rafael, R. Pavel, K. Ondrej, Nakagami-fuzzy imaging framework for precise lesion segmentation in MRI, Pattern Recognit. 128 (2022) 108675.

[2] G. Zhang, Face recognition based on fuzzy linear discriminant analysis, IERI Proc. 2 (2012) 873–879.

[3] T. Villmann, F.-M. Schleif, B. Hammer, Prototype-based fuzzy classification with local relevance for proteomics, Neurocomputing 69 (2006) 2425–2428.

[4] G.T. Reddy, M. Reddy, K. Lakshmanna, D.S. Rajput, R. Kaluri, G. Srivastava, Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis, Evol. Intell. 13 (2020) 185–196.

[5] E.H. Ruspini, J.C. Bezdek, J.M. Keller, Fuzzy clustering: A historical perspective, IEEE Comput. Intell. Mag. 14 (2019) 45–55.

[6] A. Gosain, S. Dahiya, Performance analysis of various fuzzy clustering algorithms: a review, Proc. Comput. Sci. 79 (2016) 100–111.

[7] H. Bulut, A. Onan, S. Korukoğlu, An improved ant-based algorithm based on heaps merging and fuzzy c-means for clustering cancer gene expression data, Sādhanā 45 (2020) 1–17.

[8] E.Y. Chan, W.K. Ching, M.K. Ng, J.Z. Huang, An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recognit. 37 (2004) 943–952.

[9] L. Jing, M.K. Ng, J.Z. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Trans. Knowl. Data Eng. 19 (2007) 1026–1041.

[10] D.M. Witten, R. Tibshirani, A framework for feature selection in clustering, J. Amer. Statist. Assoc. 105 (2010) 713–726.

[11] X. Qiu, Y. Qiu, G. Feng, P. Li, A sparse fuzzy c-means algorithm based on sparse clustering framework, Neurocomputing 157 (2015) 290–295.

[12] M. Yashuang, L. Xiaodong, W. Lidong, J. Zhoue, A parallel fuzzy rule-base based decision tree in the framework of map-reduce, Pattern Recognit. 103 (2020) 107326.

[13] Z.-P. Chen, J.-H. Jiang, Y. Li, Y.-Z. Liang, R.-Q. Yu, Fuzzy linear discriminant analysis for chemical data sets, Chemometr. Intell. Lab. Syst. 45 (1999) 295–302.

[14] X.-H. Wu, J.-J. Zhou, Fuzzy discriminant analysis with kernel methods, Pattern Recognit. 39 (2006) 2236–2239.

[15] C. Cifarelli, L. Nieddu, O. Seref, P.M. Pardalos, K-TRACE: A kernel k-means procedure for classification, Comput. Oper. Res. 34 (2007) 3154–3161.

[16] Z. Xu, K. Huang, J. Zhu, I. King, M.R. Lyu, A novel kernel-based maximum a posteriori classification method, Neural Netw. 22 (2009) 977–987.

[17] H.F. Pop, C. Sârbu, A new fuzzy discriminant analysis method, Commun. Math. Comput. Chem. 69 (2013) 391–412.

[18] Y. Wenzhu, S. Quansen, S. Huaijiang, L. Yanmeng, Semi-supervised learning framework based on statistical analysis for image set classification, Pattern Recognit. 103 (2020) 107500.

[19] E. McDermott, S. Katagiri, Prototype-based minimum classification error/generalized probabilistic descent training for various speech units, Comput. Speech Lang. 8 (1994) 351–368.

[20] S. Seo, M. Bode, K. Obermayer, Soft nearest prototype classification, IEEE Trans. Neural Netw. 14 (2003) 390–398.

[21] N. Cebron, M.R. Berthold, Adaptive prototype-based fuzzy classification, Fuzzy Sets Syst. 159 (2008) 2806–2818.

[22] P. Ashtari, F.N. Haredasht, H. Beigy, Supervised fuzzy partitioning, Pattern Recognit. 97 (2020) 107013.

[23] I. Irigoien, F. Mestres, C. Arenas, The depth problem: identifying the most representative units in a data group, IEEE/ACM Trans. Comput. Biol. Bioinform. 10 (2012) 161–172.

[24] J.C. Bezdek, T.R. Reichherzer, G.S. Lim, Y. Attikiouzel, Multiple prototype classifier design, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 28 (1998) 67–79.

[25] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[26] A.A. Alizadeh, M.B. Eisen, R.E. Davis, et al., Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature 403 (2000) 503–511.

[27] G. Gan, N. Michael Kwok-Po, Subspace clustering with automatic feature grouping, Pattern Recognit. 48 (2015) 3703–3713.

[28] Y. Zhiwen, D. Wang, Y. Jane, W. Hau-San, W. Si, Z. Jun, H. Guoqiang, Progressive subspace ensemble learning, Pattern Recognit. 60 (2016) 692–705.

[29] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction To Cluster Analysis, second ed., John Wiley & Sons, 2005.

[30] L. Wang, Y. Cheng, D. Lamb, R. Dharmarajan, S. Chadalavada, R. Naidu, Application of infrared spectrum for rapid classification of dominant petroleum hydrocarbon fractions for contaminated site assessment, Spectrochim. Acta Part A Mol. Biomol. Spectrosc. 207 (2019) 183–188.

[31] F. Ferraty, P. Vieu, Nonparametric Functional Data Analysis: Theory and Practice, Springer, 2006.

[32] D. Dua, C. Graff, UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2017, URL http://archive.ics.uci.edu/ml.

[33] I. Irigoien, C. Arenas, Diagnosis using clinical/pathological and molecular information, Statist. Methods Med. Res. 725 (2016) 2878–2894.

[34] C.-L. Chang, Finding prototypes for nearest neighbour classifiers, IEEE Trans. Comput. 23 (1974) 1179–1184.

[35] B. Dasarathy, Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design, IEEE Trans. Syst. Man Cybern. 24 (1994) 511–517.

[36] R.J. Campello, E.R. Hruschka, A fuzzy extension of the silhouette width criterion for cluster analysis, Fuzzy Sets Syst. 157 (2006) 2858–2875.