A proximal framework for fuzzy subspace clustering

Arthur Guillon, Marie-Jeanne Lesot, Christophe Marsala

A Proximal Framework for Fuzzy Subspace Clustering

Arthur Guillon*, Marie-Jeanne Lesot*, Christophe Marsala*

Sorbonne Université, CNRS, LIP6, Laboratoire d'informatique de Paris 6, F-75005 Paris, France

Abstract

This paper proposes a fuzzy partitioning subspace clustering algorithm that minimizes a variant of the FCM cost function with a weighted Euclidean distance and a non-differentiable penalty term. The form of the cost function suggests to split the optimization problem, taking advantage of the framework of proximal optimization. The expression of the proximal operator for the penalty term is derived and implemented in a new algorithm, PFSCM, which combines proximal descent and alternate optimization. A discussion on the extension of this work to produce sparse estimations of the subspaces is conducted. Experimental results show the relevance of the proposed approach.

Keywords: fuzzy clustering, fuzzy subspace clustering, proximal descent

1. Introduction

Cluster analysis is a central task in unsupervised machine learning that aims at partitioning data into groups with strong internal similarity and external dissimilarity. Depending on the considered framework and context, clusters partition the original data set into groups sharing some common property with varying definitions. Most often, cluster analysis informs the user about the prevailing tendencies of the data set.

Subspace clustering [1] is a generalization of this task which not only searches for a good partition of the data set, but also identifies the subspaces in which these clusters can be formed, that is, the subspaces in which the points of a cluster are similar, excluding the dimensions or features in which they are dissimilar. The identified subspaces are required to be minimal, but sufficient to describe the clusters they contain.

The identifications of the clusters and their subspaces must be simultaneous: indeed, if either the clusters or their subspaces are known beforehand, the problem reduces to finding the subspaces or correct description of the clusters, or to standard clustering, respectively. In addition, as opposed to feature selection,

[*] Corresponding author

different clusters are most of the time discovered in different subspaces: the learned metric or similarity measure is local and may differ for each cluster.

As detailed in Section 2 there exist several families of techniques and algorithms to solve the subspace clustering problem, as well as various representations of the subspaces, depending on the intended application of the subspace clustering. This paper belongs to the partitioning paradigm in a fuzzy setting. It produces clusters identified by a center. It focuses on discovering axes-parallel subspaces, which are thus identified by weights on the original data features.

Subspace clustering searches for appropriate description of the clusters using original features, motivating the identification of sparse descriptions of these subspaces, which use as few features as possible. The study of sparse models in subspace clustering is the topic of active research, as detailed in Section 2.3.

In this work, an original cost function formalising these concepts is presented. It adds, to a FCM cost function with weighted Euclidean distance [2], a penalty term expressing constraints to identify the relevant subspaces. As it is not differentiable, standard optimization techniques such as alternate optimization are not available, and this penalty term requires a new descent procedure which will allow for new constraints on the solutions, such as sparsity. Tools from the proximal descent theory [3] are thus adapted to the subspace clustering problem. The utilisation of such techniques is still relatively new in machine learning and in clustering in particular [4].

This paper introduces a new algorithm, called PFSCM, standing for Proximal Fuzzy Subspace C-Means, which uses this framework to solve the subspace clustering problem through the combination of proximal descent and alternate optimization.

This paper is structured as follows: Section 2 sketches related works and the scientific context of subspace clustering. The proposed cost function is presented and studied in Section 3. In Section 4, the implementation of proximal descent is studied to optimize the proposed function, leading to the update equations from which the PFSCM algorithm is derived. Section 5 discusses variants of the proposed cost function based on the introduction of a sparsity-inducing penalty term. PFSCM is then experimentally validated in Section 6. Conclusion and future works are discussed in Section 7.

2. Related Works

This section first introduces the problem of subspace clustering using standard notations: the data set is denoted $\{x_i, i = 1..n\}$ with $\forall i, x_i \in \mathbb{R}^d$ and the cluster set $\{C_r, r = 1..c\}$. In the following, the related works in fuzzy subspace clustering and sparse methods for subspace clusterings are recalled. These works are presented by order of relevance rather than in chronological order.

2.1. Problem Statement

Subspace clustering [1, 5] is an unsupervised task that aims at identifying groups with strong internal similarity while simultaneously learning this similarity, i.e. the set of features or subspace shared by points of a same group.

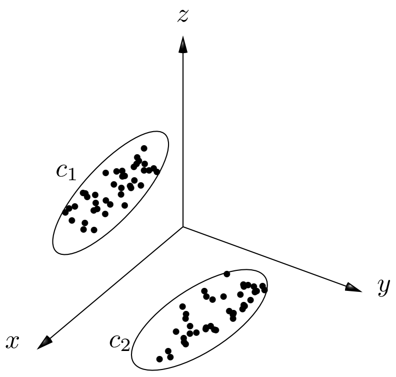

Figure 1: Two clusters, contained in two different planes: c_1 in (x, z) and c_2 in (x, y).

This similarity is local: it represents the subspace in which the cluster points exist. Depending on the field of application, various types of subspaces can be of interest, as hyper-rectangles [1], vector subspaces [5, 6] or hyperplanes of low dimension [7] to name a few.

Most of the time, subspaces are defined by projections or linear combination of the original axes. This second point of view presents subspace clustering as a generalization of Principal Component Analysis (PCA) in the sense that, in this case, the PCA is local to each cluster.

In this work, subspaces of interest are defined by weights $(w_{rp})_{p \in \{x,y,z\}}$ of the original features according to their importance in the resulting cluster, corresponding to axes-parallel subspaces. Figure 1 illustrates this choice: points of cluster c_1 all exist on the same plane $y = 0$, which means that the corresponding feature is important to identify the cluster and that it should receive a high weight ($w_{1y} \gg 0$). Points of c_1 are also closer along the z axis than they are along the x axis, and thus $w_{1z} > w_{1x}$.

2.2. Fuzzy Subspace Clustering

Fuzzy subspace clustering generalizes standard subspace clustering by allowing each data point to be assigned to several clusters with various degrees. It usually relies on the introduction, in standard fuzzy clustering algorithms such as the fuzzy c-means algorithm (FCM) [2], of subspaces (not necessarily axes-parallel) and weighted distances. Deng et al. [8] propose a survey on soft subspace clustering models and algorithms.

Keller and Klawonn [9] adapt the FCM cost function by introducing a weight w_{rp} for each cluster C_r and dimension p. Denoting $(u_{ri}) \in [0, 1]$ for $i \in \{1, \ldots, n\}$ and $r \in \{1, \ldots, c\}$ the fuzzy membership degree of x_i to cluster C_r, $c_r \in \mathbb{R}^d$ the center of cluster C_r and $(w_{rp}) \in [0, 1]$ the weight of dimension p for cluster C_r, they study the following cost function:

$$J_{K\&K}(C,U,W) = \sum_{r=1}^{c}\sum_{i=1}^{n} u_{ri}^{m} \sum_{p=1}^{d} w_{rp}^{v}(x_{ip} - c_{rp})^2 \tag{1}$$

where $m, v \in \mathbb{R}$ are fuzzifiers tuned by the user to specify the level of fuzziness of the corresponding parameters and C, U and W are the matrices containing the centers (c_r), the memberships (u_{ri}) and the weights (w_{rp}), respectively. The function is minimized under the three constraints:

- (C1) $\forall i \in \{1, \ldots, n\}, \sum_{r=1}^{c} u_{ri} = 1$
- (C2) $\forall r \in \{1, \ldots, c\}, \sum_{i=1}^{n} u_{ri} > 0;$
- (C3) $\forall r \in \{1, \ldots, c\}, \sum_{p=1}^{d} w_{rp} = 1.$

The first two constraints $(C1)$ and $(C2)$ are similar to the FCM ones and ensure that each cluster has an equal importance in the computation of the solution and that it is not trivially empty. Constraint $(C3)$ on the weights (w_{rp}) is specific to the subspace clustering problem and prevents the trivial solution such that $\forall r, \forall p, w_{rp} = 0$.

The model used in [9] is a simple adaptation of the FCM cost function to the subspace clustering problem. The Gustafson-Kessel model [10] is more complex, based on the Mahalanobis distance, capable of identifying non-axes-parallel subspaces. Borgelt and Kruse [11] improve upon this model to add shape and size regularization for the clusters.

2.3. Sparsity Constraints for Subspace Clustering

The downside of this approach is that each feature typically receives a non-zero weight, even if it plays a small role in the cluster. This not only clutters up the description of the subspaces, but also introduces noise in the cluster analysis of the data, namely the computed centers and memberships. Indeed, as shown by the update equations 3 and 4 in Section 3.2, the weigthing of the dimensions have an influence on the other variables, when they should be identified as irrelevant and not taken into account.

As a good subspace clustering solution consists in minimal yet sufficient descriptions of the clusters and their subspaces, subspace clustering is related to sparsity-inducing techniques. Over the last years, several approaches to subspace clustering heavily relied on sparse solutions to optimization problems in order to identify the correct subspaces. Most of these approaches are related to compressed sensing and are not directly usable in the framework presented in the previous section.

Most notably, Elhamifar and Vidal's Sparse Subspace Clustering (SSC) [12] solves an ℓ_1-based optimization problem for each point x_i, finding a minimal representation of x_i based on its neighbor points. In its simplest setting, this framework makes the assumption that the clusters live in a union of vector subspaces, and thus can be recovered through these local representations (see e.g. [13] for a theoretical analysis). A similarity relation is learnt and used to identify the clusters.

This framework has been extended by various authors. Li and Vidal [14] propose a structured approach to SSC, where the sparse representation and the resulting segmentation are refined iteratively through alternate optimization. Instead of looking for sparse representations, Liu et al. [15, 16] search for low-rank representations (LRR) of the data points, using the nuclear norm [17] as an approximation of the rank function. The underlying idea is that SSC identifies sparse local representations of the data but may have more trouble identifying the global structure of the data in presence of noise, whereas LRR constrains the solution globally.

Closer to the paradigm studied in the present paper, Witten and Tibshirani [18] reformulate the k-means original problem into a maximization problem with a weighted distance. This allows them to add an ℓ_1-based constraint to the problem in order to produce sparse weight vectors and identify the subspaces. Qiu et al. [19] extend this framework to fuzzy clustering and compare it to some usual subspace clustering algorithms. Borgelt [20] proposes alternatives to Keller and Klawonn and Gustafson-Kessel algorithms with sparsity inducing fuzzifier functions for u_{ri} and w_{rp}, the second one leading to the selection of principal features or axes. However, both of these contributions cannot introduce sparsity inducing terms without changing the original subspace clustering model.

3. Proximal Splitting for Fuzzy Subspace Clustering

This section presents a new cost function to model the subspace clustering problem and studies its properties. It is based on the Keller and Klawonn model but moves the constraint (C3) inside the cost function in order to change the optimization dynamic.

3.1. A Model for Fuzzy Subspace Clustering

Using the same notations as in Section 2.2, we introduce the following cost function:

$$J(C,U,W) = \underbrace{\sum_{r=1}^{c}\sum_{i=1}^{n} u_{ri}^{m} \sum_{p=1}^{d} w_{rp}^{2}(x_{ip}-c_{rp})^{2}}_{F(C,U,W)} + \gamma \underbrace{\sum_{r=1}^{c} \left| \sum_{p=1}^{d}(w_{rp}) - 1 \right|}_{G(W)} \qquad (2)$$

which is optimized under the classic FCM constraints (C1) and (C2).

This function is of the form $J(C,U,W) = F(C,U,W) + \gamma G(W)$. As shown in the next section, this form plays an important role in the optimization strategy of J.

Function F is the same as Keller and Klawonn's model, except that the weight fuzzifier v is set to 2 in order to simplify further mathematical analysis. The general case is left to be considered in future works. As it is a special case of Keller and Klawonn's subspace clustering model, function F computes solutions that are close to this model (although the use of a different descent scheme introduces some differences, see Section 6) and thus the function defined in Equation (2) aims at preserving these solutions while using a second function G to enforce some additional constraints.

In this paper, function G adds a penalty to the function which prevents the trivial solution $W = 0$. It can be understood as an inlined version of the constraint (C3), with two differences:

- the hyperparameter $\gamma \in \mathbb{R}$ is not a Lagrange multiplier, but is rather set by the user beforehand. Its purpose is to balance out the two terms, and, if large enough, it ensures that the weights W sum up to 1;
- function G is not differentiable and requires new optimization techniques presented in the next section. The advantage is that, as discussed in Section 5, they will allow for more advanced penalties, e.g. for enforcing new properties such as sparsity.

The cost function J in Equation (2) thus conveys the idea of finding a solution to the subspace clustering problem (as expressed by F) with a relaxed constraint (as expressed by G), inspired by ℓ_1-regularization [21].

3.2. Minimization of the Cost Function

As in Section 2.2, an algorithm solving the subspace clustering problem has to find parameters (C^*, U^*, W^*) minimizing J. As the proposed J is not differentiable, this algorithm cannot be derived from the standard approach of constrained optimization. Instead, the approach we propose is different, and benefits from two observations:

- the second term $\sum_{r=1}^{c} |\sum_{p=1}^{d}(w_{rp}) - 1|$ only involves the variable W and thus the optimization of variables C and U is not impacted by the non-differentiability;
- F is still differentiable in all three parameters and its gradient as a function of W is Lipschitz-continuous: these properties guarantee good performance of well-known optimization algorithms, such as gradient descent.

For the variables C and U, the optimization of F can be done through standard alternate optimization, as for fuzzy c-means: the two update equations for membership degree and cluster centers are derived,

$$u_{ri} = \frac{d_{ri}^{\frac{2}{1-m}}}{\sum_{s=1}^{c} d_{si}^{\frac{2}{1-m}}} \quad \text{where} \quad d_{ri}^2 = \sum_{p=1}^{d} w_{rp}^2 (x_{ip} - c_{rp})^2 \tag{3}$$

$$\text{and } c_{rp} = \frac{\sum_{i=1}^{n} u_{ri}^m \cdot x_{ip}}{\sum_{i=1}^{n} u_{ri}^m} \tag{4}$$

4. Proximal Splitting for Weights Optimization

The function J is not differentiable in W on its domain of definition and thus cannot be optimized by standard alternate optimization. In this section, an alternative optimization technique for the weights W is presented, based on proximal descent [4], and its implementation in a subspace clustering algorithm is then detailed.

4.1. Proximal Descent

Studied as a function of variable W only, the cost function is of the general form $J(W) = F(W)+\gamma G(W)$, where F corresponds to a differentiable model for subspace clustering and G enforces some constraints on the solutions. This form of functions has recently gained interest in the machine learning community, and proximal descent has thus been studied as an alternative to standard, gradient-based optimization techniques [22].

As F is differentiable, usual optimization techniques would suggest to use an iterative algorithm based on its gradient, such as gradient descent. This technique considers an update equation of the general form

$$W^{t+1} = W^t - \eta \cdot \nabla F(W^t)$$

where t is the iteration index and η a descent step size. This simple optimization scheme provides an iterative algorithm in order to minimize any convex function F, starting from any W^0 and iterating until convergence.

Function G is not differentiable, therefore its gradient ∇G is not defined for each W^t. Proximal descent [4] suggests to split the optimization of the two functions F and G, and to enrich gradient descent in the following way:

$$W^{t+1} = \text{prox}_{\frac{\gamma}{L}G}\left(W^t - \frac{1}{L}\nabla F(W^t)\right) \tag{5}$$

$$\text{where } \text{prox}_{\frac{\gamma}{L}G}(W') = \underset{W}{\text{argmin}} \left\{\frac{1}{2}\|W - W'\|^2 + \frac{\gamma}{L}G(W)\right\} \tag{6}$$

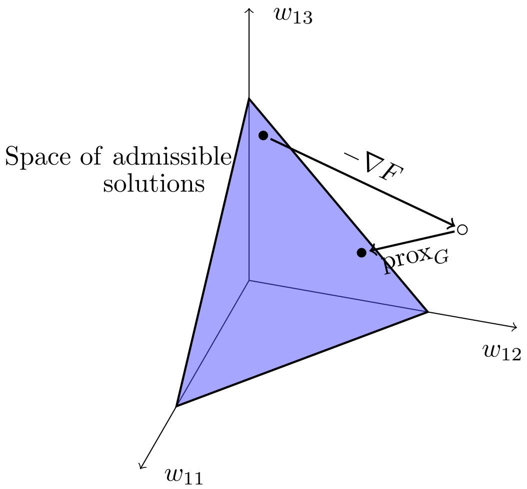

Figure 2: Principle of the two-steps proximal descent.

where $L > 0$ is a descent step size similar to η. That is, in order to solve a global minimization problem, proximal descent solves a minimization problem as defined by Equation (6) at each step of the iteration.

The justification for this additional minimization problem is the following: at each step, starting from the current step of iteration W^t, proximal descent moves according to $-\nabla F$ as would any descent approach, then according to prox_G. Figure 2 illustrates this principle, showing the space of admissible solutions in blue. This justifies the definition of the proximal operator as well: given an intermediate descent step W^t, the solution W' to Equation (6) is sought for in the neighborhood of W^t, hence the first term $\|W - W'\|^2$.

The definition of Equation (6) can seem counter-intuitive, as it proposes to solve an additional minimization problem at each step of the algorithm, in order to minimize the global function J. The key ingredient in order to efficiently solve this problem is the notion of proximal operator: a closed-form expression or algorithm to efficiently solve the optimization problem defined by Equation (6).

4.2. A Proximal Operator for G

We establish in the following theorem a proximal operator for the penalty term $G(W)$. Let K denote the vector $(1,\ 1, \ldots 1) \in \mathbb{R}^{1\times d}$, such that $K \cdot K^\intercal = d$.

Theorem 1. *Let* $G_r(W_r) = |\sum_{p=1}^{d}(w_{rp}) - 1|$ *and* $L \in \mathbb{R}$.

$$\text{prox}_{\frac{\gamma}{L}G_r}(W_r) = W_r + \frac{1}{d}K^\intercal \cdot \left(1 + \text{prox}_{\frac{\gamma d}{L}|\cdot|}(K \cdot W_r - 1) - K \cdot W_r\right) \tag{7}$$

where $\text{prox}_{\lambda|\cdot|}(x) = \text{sign}(x)\max(|x| - \lambda, 0)$.

Moreover, $\text{prox}_{\frac{\gamma}{L}G}(W) = \left(\text{prox}_{\frac{\gamma}{L}G_r}(W_r)\right)_{r=1\ldots c} \in \mathbb{R}^{d\times c}$.

Proof. The proof uses results from [23] and [4]. First, $G_r(W_r) = \phi(K \cdot W_r)$ where $\phi(x) = |x - 1|$. Using the translation and semi-orthogonal linear transform properties [23]:

$$\begin{aligned} \operatorname{prox}_{G_r}(W_r) &= W_r + \frac{1}{d}K^{\intercal} \cdot \left(\operatorname{prox}_{\phi}(K \cdot W_r) - K \cdot W_r \right) \\ &= W_r + \frac{1}{d}K^{\intercal} \cdot \left(1 + \operatorname{prox}_{d|\cdot|}(K \cdot W_r - 1) - K \cdot W_r \right) \end{aligned}$$

Hence the expression of $\operatorname{prox}_{\frac{\gamma}{L}G_r}$ by the postcomposition property [4]. Finally, $\operatorname{prox}_{\frac{\gamma}{L}G}$ is computed using the separable sum property of proximal operators [4]. □

Equation (7) gives the expression of the proximal operator of the G function. This operator can be used to efficiently implement the scheme defined in Equation (5) to find the solution W minimizing function J.

As η for gradient descent, the choice of constant L determines the speed of convergence of the descent. We experimentally observe that setting L to $\operatorname{trace}(H^{-1})$ yields good results, where H is the Hessian matrix of F (as a function of W). As we set $v = 2$ in Equation (2), F is a sum of quadratic functions and its Hessian does not depend on W. It can thus be computed once and for all before running proximal descent.

4.3. A Fuzzy Subspace Algorithm: PFSCM

The previous mathematical results allow us to efficiently implement proximal descent for the identification of the subspaces. Combining the expression of $\operatorname{prox}_{\frac{\gamma}{L}G}$ with the results of Section 3.2, we introduce the PFSCM algorithm for fuzzy subspace clustering, detailed in Algorithm 1 and commented on below. PFSCM is a FCM-style alternate optimization algorithm which substitutes the exact update of the weights W with proximal descent.

Initialization is a typical issue of k-means-like algorithms. In this paper, initial centers are randomly chosen and each cluster receives uniform weights for all dimensions. As C and U are optimized before W, this is equivalent to running the FCM algorithm and initializing the subspace algorithm with the result, as is often done for subspace clustering algorithms such as Keller and Klawonn [9] or Gustafson-Kessel [10] algorithms.

As most partitioning algorithms, the number c of clusters to identify must be set by the user, as well as the constant $\gamma > 0$. PFSCM is not sensitive to precise values of γ and just needs a value "large enough" (e.g. 10^4) in order to impose $\forall r, \sum_{p=1}^{d} w_{rp} = 1$.

The algorithm then iterates the update of U, C and W, similarly to alternate optimization in k-means-like algorithms. It consists of two alternate inner loops: the regular parameters C and U are optimized separately from W, which requires the special optimization procedure described in the previous subsection. Variables C and U are optimized one last time at the end of the algorithm, in order to guarantee that the result takes the final computed weights into account.

```
Data: $X$: data matrix
Parameters: $c$,$\gamma$: numbers;
Variables: C, U, W: arrays;
           W$_{last}$: array
Initialization: $W_r \leftarrow (1,\ 1, \dots\ 1)$ for each $C_r$;
                C $\leftarrow$ random centers
Output: C, U, W
repeat
    repeat
        Update U according to Equation (3);
        Update C according to Equation (4)
    until convergence(C, U);
    repeat
        Update W according to Equation (6)
    until convergence(W);
    W$_{last}$ $\longleftarrow$ W
until convergence(W$_{last}$);
Update U and C one last time.
```

Algorithm 1: Proximal fuzzy subspace clustering PFSCM algorithm

The convergence criteria are defined as the distance between the current and the previous values of the parameters being optimized. In particular, convergence for (C, U) is defined as $\|C^t - C^{t+1}\|_2 < \epsilon \wedge \|U^t - U^{t+1}\|_2 < \epsilon$.

PFSCM outputs U, C and W. In order to exploit the result of the algorithm, it may be of interest to extract the dimension associated to each cluster. To that aim we propose to post-process the matrix W using a threshold parameter *cut* to cut out the irrelevant dimensions in a simple fashion: a dimension p for a cluster C_r is considered relevant if $w_{rp} > cut$.

However, instead of using a threshold parameter, an improvement of the algorithm would be to identify the irrelevant dimensions during the descent in order to get rid of them and not have them influence the identification of the centers. This motivates the research of sparsity-inducing models, which are discussed in the next section.

5. Towards a Sparsity-Inducing Penalty Function

This section discusses extensions of PFSCM aiming at producing sparse solutions: as can be experimentally observed (see Section 6), PFSCM leads to solutions where the weights can be small but different from 0. In order to produce sparse estimations of the subspaces, the previous function J must be changed. In this section, various solutions are discussed to define variants for G, listing some of the difficulties or the similarities with other approaches of the literature.

Indeed, for function G defined in Equation (2) and for large values of γ, $\text{prox}_{\gamma G}$ acts as a projection towards the simplex $\Delta = \left\{ W_r \in \mathbb{R}^d \middle| \sum_{p=1}^{d} w_{rp} = 1 \right\}$ for each W_r. The framework presented in this paper would encourage looking for a function which would favor sparse projections on Δ.

ℓ_1-norm regularization. The most common sparsity-inducing norm, the ℓ_1 norm, is useless in the current setting: indeed, after projection using prox_G, the solutions of interest all verify $\forall r, \|W_r\|_1 = 1$, and thus this norm does not discriminate any solution under this constraint. A better penalty-inducing function H must be found, which still needs to be "proximable". Moreover, even if G and H are proximable, this is not necessarily the case of their sum $G + H$ [4].

Negentropy based regularization. An alternative to ℓ_1 and other non-differentiable norms may come from negentropy-based regularization, already used in conjunction of standard FCM clustering to regularize the number of clusters (see e.g. [24]). The study conducted in Section 6 has not yet be extended to negentropy-regularized subspace algorithms. However, Jing et al. [25] use a very similar technique for a slightly different subspace clustering model, in a different context: as they consider sparse data, they use entropy (not negentropy) in order not to promote sparsity, but to promote subspaces of high dimensionality.

Projection-based regularization. Closer to the spirit of the present paper, another direction of work is to reformulate G as an actual projection on Δ and then to introduce a sparsity-promoting penalty. First define G_1 to be the following penalty function:

$$G_1(W_r) = \begin{cases} 0 \text{ if } \sum_{p=1}^{d} w_{rp} = 1 \\ \infty \text{ otherwise} \end{cases}$$

Function G_1 is convex and thus has a proximal operator:

$$\begin{aligned} \text{prox}_{G_1}(W_r) &= \underset{W'_r \in \mathbb{R}^d}{\text{argmin}} \left\{ G_1(W'_r) + \frac{1}{2} \|W_r - W'_r\|^2 \right\} && (8) \\ &= \underset{W'_r \in \Delta}{\text{argmin}} \left\{ \frac{1}{2} \|W_r - W'_r\|^2 \right\} && (9) \\ &= \text{proj}_\Delta(W_r) && (10) \end{aligned}$$

that is, the proximal operator of G_1 is the projection on Δ.

Unlike the function G in Equation (2), this projection does not depend on the value of γ: G_1 strictly enforces the constraint, but still does not discriminate between the points of Δ. The goal now becomes to assign a price to the solutions on Δ, in order to favor sparse ones. However, this direction is left for future works.

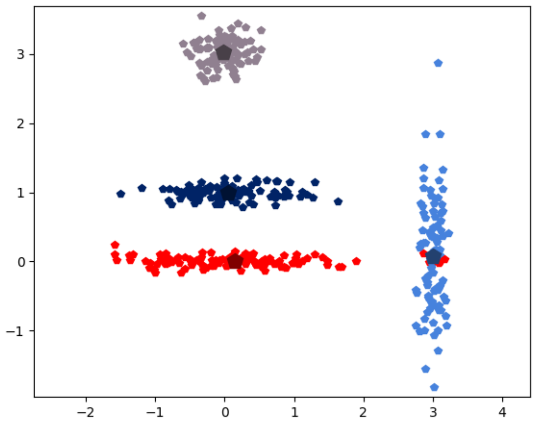

Figure 3: Clustering example in two dimensions

	Gray cluster		Dark cluster		Red cluster		Blue cluster	
	w_1	w_2	w_1	w_2	w_1	w_2	w_1	w_2
Weights	0.528	0.472	0.063	0.937	0.027	0.973	0.964	0.036

Table 1: Computed weights for the example given in Figure 3. Column w_1 (resp. w_2) denotes the weight associated to the x-axis (resp. y-axis).

6. Experimental Study

The proposed PFSCM algorithm has been tested on artificial data in order to study its ability to correctly identify centers of non-circular clusters, as well as the dimensions that are relevant to describe the clusters. The results show the effectiveness of PFSCM in detecting the clusters and their subspaces. Moreover, PFSCM is compared to Keller and Klawonn's algorithm [9] and shows to provide a better estimation of the dimensionality of the subspaces.

6.1. Illustrative Example

This subsection presents illustrative experiments of subspace clustering using the PFSCM algorithm, similar to the visual examples given in [9]. The first example in $d = 2$ dimensions is represented in Figure 3: four clusters are generated, one of them (the top gray one in Figure 3) being circular while the others have a very low variance in one dimension. PFSCM is run with $c = 4$, $m = 2$, and $\gamma = 1000$.

In Figure 3 the points are colored according to the cluster C_r for which u_{ri} is maximum and Table 1 presents the weights computed for each dimension and cluster. It can be observed that PFSCM correctly identifies the desired clusters and their dimensions: the two weights (w_1, w_2) found for the circular cluster are

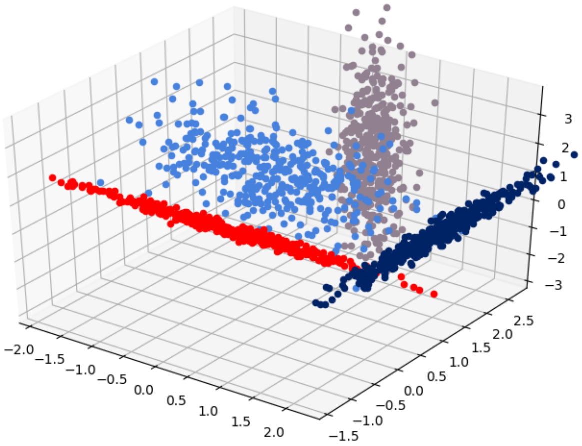

Figure 4: Clustering example in three dimensions

similar, whereas the horizontal (respectively vertical) clusters verify $w_2 \gg w_1$ (respectively $w_1 \gg w_2$).

It is worth noting that, for this specific instance, some points close to the blue cluster are assigned to one of the horizontal clusters, as it minimizes the cost function. This kind of inliers is frequent in subspace clustering problems, and naturally leads to the use of fuzzy membership values (u_{ri}). It was also studied for its own sake by Guillon et al. [26], which suggest the use of a Laplacian-based regularization term to prevent it.

Another example is given in Figure 4. Again, 4 clusters are depicted and colored according to the cluster maximizing u_{ri}. These clusters live in various subspaces of various dimensionalities, but are still identified by PFSCM.

6.2. Artificial Data Set with Ellipsoidal Clusters

The previous example is generalized to higher dimensions and several algorithms of fuzzy subspace clustering are compared: PFSCM, Keller and Klawonn (KK), Gustafson-Kessel (GK) and WLFC algorithm from [26].

6.2.1. Experimental Protocol

Considered Data. In order to validate PFSCM, the previous experiment is generalized to higher dimensions, more precisely to artificial data of dimension $d \in \{5, 7, 9, 11, 13, 15\}$. For each experiment, $k = 4$ centers are randomly generated in the hypercube $[-3, 3]^d$. Then, d_r dimensions $j_1, \dots j_{d_r}$ are randomly picked, with d_r randomly chosen between 1 and $d - 3$. Dimensions $j_1, \cdots, j_{d_r}$ are thereafter called the "relevant dimensions" for cluster C_r. For each cluster,

Table 2: Average and standard deviation of δ

d	WLFC	K&K	GK	PFSCM
5	0.58 ± 0.09	1.18 ± 0.77	0.59 ± 0.10	0.60 ± 0.11
7	0.72 ± 0.12	1.49 ± 1.03	0.75 ± 1.67	0.73 ± 0.13
9	0.86 ± 0.14	1.98 ± 1.58	3.36 ± 5.09	0.84 ± 0.13
11	0.99 ± 0.13	2.02 ± 1.36	9.78 ± 10.4	0.98 ± 0.12
13	1.12 ± 0.11	1.77 ± 0.91	19.4 ± 13.0	1.11 ± 0.10

Table 3: Average and standard deviation of ϕ

d	WLFC	K&K	GK	PFSCM
5	1.83 ± 1.63	1.59 ± 1.26	11.7 ± 16.7	1.32 ± 1.12
7	1.93 ± 2.09	1.74 ± 1.83	13.8 ± 18.7	1.73 ± 1.43
9	2.20 ± 2.64	2.14 ± 2.54	15.2 ± 16.9	2.37 ± 2.10
11	2.06 ± 2.42	1.97 ± 2.01	NA	1.93 ± 2.04
13	2.99 ± 3.82	2.99 ± 4.00	NA	3.12 ± 2.19

100 points are generated according to a Gaussian distribution, with variance $v_j < 0.1$ for dimensions $j_1, \ldots, j_{d_r}$ and $v_j \in [0.5, 0.9]$ for other dimensions. The generated points in cluster r in dimension j thus follow $X_r \sim \mathcal{N}(c_r, v_j)$.

Algorithm Parameters. All algorithms are initialized with FCM centers and use $m = v = 2$ and $c = 4$. PFSCM is ran with $\gamma = 1000$ and $c = 4$. All algorithms were implementend using the same convergence criterion, with $\epsilon = 10^{-4}$.

The parameter *cut* is set to $\frac{1}{2d}$, which is a simple rule of thumb to identify the dimensions selected as relevant by the algorithms in each considered dimension d.

Quality Criteria. All algorithms are evaluated on three metrics in order to qualify their results and their ability to discover the desired clusters and subspaces, and their dimensions.

First, let $\delta = \sum_{r=1}^{4} \|\hat{c}_r - c_r\|_2$ be the sum of the Euclidean distances between the generated centers and the computed ones (c_r): this metric is a standard quality criterion for evaluating the produced clusters. A low value means that the computed centers are close to the original ones.

We also consider θ defined as the percentage of clusters for which all relevant dimensions are correctly identified by the algorithm: the relevant dimensions are correctly identified if $w_{rj} > cut \Leftrightarrow j \in \{j_1, \cdots j_{d_r}\}$.

Finally, for the clusters for which the relevant dimensions have been correctly identified, let the weight ratio be $\phi = \frac{\omega_1}{\omega_{j_{d_r}}}$ where ω_1 is the largest computed weight and $\omega_{j_{d_r}}$ the smallest computed weight for the relevant dimensions. This metric measures the distortion of the cluster between the relevant dimensions, as estimated by the algorithms.

Table 4: Average and standard deviation of θ

d	WLFC	K&K	GK	PFSCM
5	52%	51%	29%	63%
7	53%	51%	17%	68%
9	48%	46%	12%	74%
11	48%	50%	3%	78%
13	58%	56%	1%	85%

6.2.2. Obtained Results

Cluster identification and stability. The results of the experiment are gathered from [26] and reported in three tables. Table 2 and Table 3 give the means and standard deviations for δ and ϕ. Table 4 gives the criterion θ, computed over 100 runs of each algorithm. The low mean distances δ in Table 2 show that WLFC and PFSCM succeed in identifying the centers of the generated clusters, while K&K is not as efficient. The GK algorithm performs well in low dimensions, but significantly worse when d increases. Moreover, WLFC and PFSCM appear more stable than K&K and GK, judging by the lower standard deviation.

Dimensionality. The values of θ in Table 4 show that WLFC and K&K succeed in estimating the dimensionality of roughly half of the generated clusters whereas PFSCM performs better when d increases. These results may be related to the choice of the parameter *cut* used to defuzzify W. This table also shows that GK has troubles identifying the relevant dimensions of the generated clusters; as θ gets very low in higher dimensions, the values of ϕ for these dimensions is not relevant. A possible explanation of these poor results is that GK is not constrained to search for axes-parallel subspaces and converges towards another solution, as there are not enough points to guide the algorithm. This would also explain the results of the algorithm in Table 2. In Table 3, WLFC, K&K and PFSCM show similar results, bearing in mind the very large standard deviations, PFSCM seeming slightly more stable.

Computation time. In terms of computational time, K&K is significantly faster than all other algorithms. WLFC appears to be globally twice as slow as K&K on average regardless of the dimension. GK is two to five times slower than K&K, probably because of the use of expensive matrix operations. Finally, PFSCM is much slower than the other algorithms: in higher dimensions it is more than ten times slower than K&K. This may be due to a naive implementation of the inner loop and to the choice of the convergence criterion.

Axes-parallel Gustafson-Kessel algorithm. For the aforementioned reasons, the general GK algorithm is not suited to this experiment. As presented in [20], this algorithm can be modified in order to identify only axes-parallel soft subspaces.

We thus ran a specific experiment in order to compare PFSCM to axes-parallel GK, which shows that both algorithms perform similarily in terms of the δ metric. However, as GK algorithm describes the subspaces in terms of the variance of their dimension, it provides no indication as to if a dimension is actually relevant or not in the description of a cluster. This feature also seems to preclude its use for the main goal of this work, which is the identification of sparse subspaces.

7. Conclusion and Future Works

In this paper, a new approach to solve the fuzzy subspace clustering problem with a cost function involving a non-differentiable term has been introduced. The cost function is expressed as the sum of two functions, a model and a penalty or regularizer term. Advanced optimization techniques are explored, which replace the standard update equations of fuzzy c-means-like algorithms.

Experiments on synthetic data show the relevance of the proposed approach, that appears to correctly identify all the relevant dimensions and not more, whereas Keller and Klawonn's algorithm tends to underestimate the number of relevant dimensions. This provides more information about the importance of each dimension for the subspaces and clusters.

Future works will aim at generalizing this approach around the same key ideas based on the discussion initiated in the paper: a differentiable function matching the specification of the problem and one or several penalty functions, expressing constraints on the shape of the solution. The introduction of regularization terms for parameters other than W will also be studied. Finally, more efficient descent schemes will be considered, in order to speed up the descent.

Acknowledgments

The authors would like to thank Nikhil R. Pal for his valuable help and comments regarding this work.

This work was supported by the French "Investissement d'avenir" program (Big-data – Cloud Computing) and was carried out as part of the REQUEST project.

References

[1] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, in: Proc. of the 1998 ACM SIGMOD Int. Conf. on Management of Data, ACM, 1998, pp. 94–105. doi:10.1145/276304.276314.

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

[3] J.-J. Moreau, Fonctions convexes duales et points proximaux dans un espace hilbertien, Compte-rendu de l'Académie des Sciences de Paris Sér. A Math 255 (1962) 2897–2899.

[4] N. Parikh, S. Boyd, Proximal algorithms, Foundations and trends in optimization 1 (2013) 123–231.

[5] R. Vidal, A tutorial on subspace clustering, IEEE Signal Processing Magazine 28 (2010) 52–68.

[6] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, J. S. Park, Fast algorithms for projected clustering, in: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD '99, ACM, New York, NY, USA, 1999, pp. 61–72. URL: http://doi.acm.org/10.1145/304182.304188. doi:10.1145/304182.304188.

[7] D. Wang, C. Ding, T. Li, K-subspace clustering, in: Machine learning and knowledge discovery in databases, Springer, 2009, pp. 506–521.

[8] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, S. Wang, A survey on soft subspace clustering, Information Sciences 348 (2016) 84–106.

[9] A. Keller, F. Klawonn, Fuzzy clustering with weighting of data variables, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 8 (2000) 735–746.

[10] D. E. Gustafson, W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: IEEE Conference on Decision and Control, volume 17, IEEE, 1978, pp. 761–766. URL: http://dx.doi.org/10.1109/cdc.1978.268028.

[11] C. Borgelt, R. Kruse, Shape and size regularization in expectation maximization and fuzzy clustering, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2004, pp. 52–62.

[12] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, 2009, pp. 2790–2797. URL: http://dx.doi.org/10.1109/CVPRW.2009.5206547. doi:10.1109/CVPRW.2009.5206547.

[13] V. M. Patel, H. Van Nguyen, R. Vidal, Latent space sparse subspace clustering, in: Proc. of the IEEE Int. Conf. on Computer Vision, 3-6 December 2013, Sydney, Australia, 2013, pp. 225–232.

[14] C.-G. Li, R. Vidal, Structured sparse subspace clustering: A unified optimization framework, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 277–286.

[15] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, in: Procs. of the 27th international conference on machine learning (ICML10), 21-24 June 2010, Haifa, Israel, 2010, pp. 663–670.

[16] G. Liu, S. Yan, Latent low-rank representation for subspace segmentation and feature extraction, in: 2011 IEEE International Conference on Computer Vision (ICCV), 6-13 November 2011, Barcelona, Spain, IEEE, 2011, pp. 1615–1622.

[17] C. D. Meyer, Matrix analysis and applied linear algebra, volume 2, SIAM, 2000.

[18] D. M. Witten, R. Tibshirani, A framework for feature selection in clustering, Journal of the American Statistical Association 105 (2010) 713–726.

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

[20] C. Borgelt, Fuzzy subspace clustering, in: Advances in Data Analysis, Data Handling and Business Intelligence, Studies in Classification, Data Analysis, and Knowledge Organization, Springer, 2010, pp. 93–103.

[21] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1996) 267–288.

[22] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, Optimization with sparsity-inducing penalties, Foundations and Trends in Machine Learning 4 (2012) 1–106.

[23] P. L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in: Fixed-point algorithms for inverse problems in science and engineering, Springer, 2011, pp. 185–212.

[24] H. Sahbi, N. Boujemaa, Validity of fuzzy clustering using entropy regularization, in: IEEE International Conference on Fuzzy Systems, IEEE, 2005, pp. 177–182.

[25] L. Jing, M. K. Ng, J. Z. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Transactions on knowledge and data engineering 19 (2007).

[26] A. Guillon, M.-J. Lesot, C. Marsala, Laplacian regularization for fuzzy subspace clustering, in: IEEE International Conference on Fuzzy Systems, IEEE, 2017, pp. 1–6.