# Optimisation proximale pour le subspace clustering flou

Arthur Guillon, Marie-Jeanne Lesot, Christophe Marsala

# Optimisation proximale pour le subspace clustering flou

# Proximal Optimization for Fuzzy Subspace Clustering

Arthur Guillon Marie-Jeanne Lesot Christophe Marsala
Sorbonne Universites, UPMC Univ Paris 06,
CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris, France

{Arthur.Guillon,Marie-Jeanne.Lesot,Christophe.Marsala}@lip6.fr

**Résumé :**

Cet article présente un algorithme de *subspace clustering*, dont la fonction de coût similaire aux $c$-moyennes floues fait apparaître une distance euclidienne pondérée et un terme de pénalité non-différentiable. Cet algorithme s'appuie sur le cadre théorique de l'optimisation par descente proximale qui permet d'établir l'expression d'un terme de mise à jour pour cette fonction de coût. Un nouvel algorithme, nommé PFSCM, est présenté, qui combine descente proximale et optimisation alternée. Les expériences réalisées sur des données artificielles montrent la pertinence de l'approche considérée.

**Mots-clés :**

Clustering, optimisation, sélection d'attributs locale.

**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 penalty term. To this aim it considers the framework of proximal optimization. It establishes the expression of the proximal operator for the considered cost function and derives PFSCM, an algorithm combining proximal descent and alternate optimization. Experiments show the relevance of the proposed approach.

**Keywords :**

Clustering, optimization, local attribute selection.

## 1 Introduction

Le *subspace clustering* [1] est un problème classique de l'apprentissage non-supervisé, qui vise à partitionner un ensemble de données en groupes homogènes mais distincts, tout en identifiant de façon simultanée les sous-espaces qui permettent de représenter ces clusters. Les sous-espaces identifiés doivent être de dimension minimale mais suffisante pour décrire les clusters qu'ils contiennent, et peuvent donc être différents d'un cluster à l'autre.

Comme détaillé dans la section 2, il existe plusieurs familles de techniques pour résoudre le problème du *subspace clustering* qui font également varier la représentation des sous-espaces en fonction de l'utilisation subséquente des clusters. Cet article embrasse le paradigme dit du partitionnement dans un cadre flou, et produit des clusters identifiés par un centre. En outre, les sous-espaces identifiés sont parallèles aux axes et sont décrits par des pondérations des attributs des données. L'article propose une nouvelle fonction de coût, qui se base sur celle des $c$-moyennes floues [3] mais utilise une distance euclidienne pondérée, et fait apparaître un nouveau terme de pénalité exprimant des contraintes sur les sous-espaces identifiés.

Ce dernier terme n'étant pas différentiable, le minimum de cette fonction ne peut être atteint par les techniques d'optimisation usuelles. Cet article présente donc un nouveau schéma d'optimisation alterné pour le *subspace clustering* qui exploite des outils de la théorie proximale [7]. L'utilisation de ces outils reste relativement neuve en apprentissage, et en clustering en particulier [8]. Cet article étudie leur exploitation pour améliorer l'identification des sous-espaces pertinents associés à chaque cluster.

Une implémentation novatrice de ce paradigme est proposée dans le cadre du *subspace clustering* flou. Cet article établit l'expression d'un opérateur proximal permettant l'optimisation de la fonction de coût considérée. Enfin, il introduit l'algorithme PFSCM (pour *Proximal Fuzzy Subspace C-Means*) qui s'appuie sur cette expression analytique pour résoudre le problème du *subspace clustering* en combinant descente

proximale et optimisation alternée.

L'article est structuré comme suit : la section 2 résume l'état de l'art du *subspace clustering*. Une nouvelle fonction de coût est présentée et étudiée en section 3. Dans la section 4, l'implémentation de la descente proximale est étudiée dans le but d'optimiser la fonction de coût proposée, menant à l'équation de mise à jour dont PFSCM est dérivé. Cet algorithme est ensuite validé expérimentalement dans la section 5.

## 2 État de l'art

Le *subspace clustering* [1] peut être vu comme une combinaison de deux tâches à résoudre simultanément, le clustering et la sélection d'attributs, cette dernière étant locale à chaque cluster. De très nombreuses approches ont été explorées, aussi bien dans les communautés de l'apprentissage que de la fouille de données ou de la vision assistée par ordinateur, voir [11] pour un état de l'art détaillé. Nous présentons ici brièvement les techniques de partitionnement itératives, dans le cadre desquelles cet article s'inscrit.

L'algorithme des $k$-*subspaces* [12] généralise l'approche des $k$-moyennes : il alterne une phase d'affectation des données aux clusters et une ré-estimation des sous-espaces dans lesquels se trouvent ces clusters. Witten & Tibshirani [13] reformulent l'optimisation des $k$-moyennes sous la forme d'un problème de maximisation d'une fonction de coût faisant intervenir une distance pondérée. Une contrainte en norme $\ell_1$ est ajoutée dans le but de produire des vecteurs parcimonieux pour identifier les sous-espaces. Qiu et al. [9] propose une version floue de cet algorithme. Cependant, ces travaux modifient de façon importante la fonction de coût des $k$-moyennes originelles conduisant à un problème de maximisation pour ajouter le terme de régularisation $\ell_1$.

En restant plus proches de la fonction de coût des $k$-moyennes dans un cadre flou, Keller & Klawonn [6] proposent une fonction de coût avec une distance euclidienne pondérée localement. En notant $(x_i)_{i=1}^n \in \mathbb{R}^d$ les données, $d$ la dimension de l'espace de description des données, $x_{ij}$ la $j$-ième composante du vecteur $x_i$, $c$ le nombre de clusters, $u_{ri} \in [0,1]$ le degré d'appartenance de $x_i$ au cluster $C_r$ avec $r \in [\![1, c]\!]$ et $i \in [\![1, n]\!]$, $\mu_r \in \mathbb{R}^d$ le centre du cluster $C_r$ et $w_{rj} \in [0,1]$ le poids de la dimension $j$ pour le cluster $C_r$, cette fonction de coût s'écrit :

$$J_{K\&K}(C, U, W) = \sum_{r=1}^{c} \sum_{i=1}^{n} u_{ri}^m \sum_{j=1}^{d} w_{rj}^v (x_{ij} - \mu_{rj})^2 \tag{1}$$

où $m, v > 1$ sont des coefficients de fuzzification fixés par l'utilisateur (classiquement choisis égaux à 2) et $C$, $U$ et $W$ sont les matrices contenant respectivement les centres $(\mu_r)$, les degrés d'appartenance $(u_{ri})$ et les poids $(w_{rj})$. La fonction est minimisée sous les contraintes suivantes :

- ($\mathrm{C}_1$) $\forall i \in [\![1, n]\!], \sum_{r=1}^{c} u_{ri} = 1$ ;
- ($\mathrm{C}_2$) $\forall r \in [\![1, c]\!], \sum_{i=1}^{n} u_{ri} > 0$ ;
- ($\mathrm{C}_3$) $\forall r \in [\![1, c]\!], \sum_{j=1}^{d} w_{rj} = a \in \mathbb{R}^*$.

($\mathrm{C}_1$) et ($\mathrm{C}_2$) sont similaires aux contraintes des $c$-moyennes floues. La contrainte ($\mathrm{C}_3$) sur les poids $(w_{rj})$, où $a$ est un paramètre défini par l'utilisateur, est spécifique au problème du *subspace clustering* et permet d'éliminer la solution triviale $W = 0$. La minimisation de l'équation (1) sous ces contraintes produit une solution au problème du *subspace clustering* flou, le poids $w_{rj}$ capturant la distance intra-cluster des points de $C_r$ dans la seule dimension $j$. Cette fonction de coût a été généralisée par Borgelt [4] pour favoriser des degrés nuls.

La fonction $J_{K\&K}$ et son lagrangien sont différentiables en chaque paramètre, ce qui permet de la minimiser en restant dans le cadre standard de l'optimisation sous contraintes : l'étude du lagrangien permet d'obtenir trois équations de mise à jour pour les paramètres $C$, $U$ et $W$.

## 3 Fonction de coût proposée

Dans cette section, une nouvelle fonction de coût est proposée pour modéliser le problème du *subspace clustering*. Celle-ci fait intervenir un terme de régularisation non-différentiable portant sur les poids $(w_{rj})$, qui intègre directement la contrainte sous la forme d'une pénalisation.

### 3.1 Introduction d'un terme de pénalisation

En réutilisant les notations de la section 2, nous proposons la fonction de coût suivante :

$$J(C,U,W) = \sum_{r=1}^{c}\sum_{i=1}^{n} u_{ri}^{m} \sum_{j=1}^{d} w_{rj}^{2}(x_{ij}-\mu_{rj})^2 + \gamma \sum_{r=1}^{c} \Big| \sum_{j=1}^{d} (w_{rj}) - \alpha \Big| \qquad (2)$$

sous les contraintes ($C_1$) et ($C_2$), typiques des $c$-moyennes floues. Le premier terme est le même que celui de la fonction de coût $J_{K\&K}$, à l'exception de l'exposant $v$, qui est fixé à 2 pour simplifier l'analyse. Ce terme correspond à la fonction de coût des $c$-moyennes floues avec un facteur de pondération propre à chaque dimension pour le calcul de la distance euclidienne.

Le second terme introduit un coût supplémentaire qui empêche la somme des poids de chaque cluster $C_r$ de s'éloigner d'un paramètre $\alpha \in \mathbb{R}$ fourni par l'utilisateur et qui joue le même rôle que le paramètre $a$ de la contrainte ($C_3$). Pour $\alpha \neq 0$, ce paramètre élimine la solution triviale $W = 0$. Le paramètre $\gamma \in \mathbb{R}$ sert à équilibrer les deux termes : il suffit qu'il soit assez grand pour que la contrainte soit effective et que les solutions triviales soient éliminées. Ce terme peut être vu comme une version de la contrainte ($C_3$) insérée dans la fonction de coût. Cette contrainte n'a donc plus de raison d'être optimisée selon la technique habituelle du lagrangien, mais permet de s'intéresser à de nouvelles techniques d'optimisation.

La fonction de coût $J$ modélise donc le problème du *subspace clustering* avec une contrainte relâchée, inspirée par la régularisation $\ell_1$ [10].

### 3.2 Minimisation de la fonction proposée

La fonction $J$ peut être décomposée comme $J(C,U,W) = F(C,U,W) + \gamma G(W)$, où

$$F(C,U,W) = \sum_{r=1}^{c}\sum_{i=1}^{n} u_{ri}^{m} \sum_{j=1}^{d} w_{rj}^{2}(x_{ij}-\mu_{rj})^2$$

$$G(W) = \sum_{r=1}^{c} \Big| \sum_{j=1}^{d} (w_{rj}) - \alpha \Big| \qquad (3)$$

La fonction $J$ possède des propriétés intéressantes, qui justifient l'utilisation de la technique présentée dans la section suivante. Tout d'abord, $J$ est une somme de deux fonctions convexes en $W$ et est donc convexe. De plus, $F$ est différentiable en ses trois paramètres et lipschitzienne en $W$ pour $C$ et $U$ fixés, ce qui encourage l'utilisation de techniques standard telles que la descente de gradient.

Pour $W$ fixé, minimiser $J$ sous les contraintes ($C_1$) et ($C_2$) est équivalent à minimiser $F$. Comme pour les $c$-moyennes floues, ceci peut être fait par optimisation alternée qui conduit aux équations de mise à jour classiques où la distance est remplacée par sa variante pondérée :

$$u_{ri} = \frac{d_{ri}^{\frac{2}{1-m}}}{\sum_{s=1}^{c} d_{si}^{\frac{2}{1-m}}} \qquad (4)$$

$$\text{où } d_{ri}^2 = \sum_{j=1}^{d} w_{rj}^2 (x_{ij}-\mu_{rj})^2$$

$$\mu_{rj} = \frac{\sum_{i=1}^{n} u_{ri}^{m} \cdot x_{ij}}{\sum_{i=1}^{n} u_{ri}^{m}} \quad (5)$$

Ces deux équations sont utilisées dans l'algorithme PFSCM décrit dans la section suivante pour mettre à jour les termes $u_{ri}$ et $\mu_r$ de façon à déterminer le minimum de J.

La fonction $G$ est convexe mais non différentiable en $W$, ce qui ne permet pas d'établir une équation de mise à jour pour l'optimisation des poids $W$ et justifie l'utilisation de la technique de la descente proximale présentée dans la section suivante.

# 4 Optimisation des poids par descente proximale

Cette section présente la méthode d'optimisation des poids $W$ de la fonction $J$ donnée dans l'équation (2), qui exploite le cadre de la descente proximale [8] rappelé dans la section 4.1 ci-dessous. Elle décrit ensuite l'algorithme complet permettant d'optimiser la fonction $J$ globalement selon ses trois paramètres. Dans cette section, on se focalise sur la matrice des poids $W$, en maintenant $C$ et $U$ fixés. On écrit donc $J(W)$ au lieu de $J(C, U, W)$ pour plus de simplicité.

## 4.1 Descente proximale

La fonction de coût est de la forme $J(W) = F(W) + \gamma G(W)$, où $F$ et $G$ sont convexes, mais seule $F$ est différentiable. Cette forme de problèmes de minimisation a rencontré un intérêt croissant ces dernières années (par exemple quand la fonction $G$ est un terme de régularisation), et la descente proximale a été étudiée comme alternative aux techniques d'optimisation classiques [2].

Parmi celles-ci, la descente de gradient cherche le minimum de $J$ par itérations de l'équation de mise à jour $W^{t+1} = W^t - \eta \cdot \nabla J(W^t)$, avec $t$ l'indice d'itération, $\eta$ le pas de descente et $\nabla J$ le gradient de $J$. La technique de la descente proximale enrichit ce schéma comme suit :

$$W^{t+1} = \mathrm{prox}_{\frac{\gamma}{L}G}\left(W^t - \frac{1}{L}\nabla F(W^t)\right) \quad (6)$$

$$\text{où } \mathrm{prox}_{\frac{\gamma}{L}G}(W) = \underset{W'}{\mathrm{argmin}} \left\{\frac{1}{2}\|W - W'\|^2 + \frac{\gamma}{L}G(W')\right\} \quad (7)$$

$L > 0$ est un pas de descente, similaire à $\eta$. L'équation (6) s'interprète de la façon suivante : pour chercher le minimum de $J$, on fait d'abord un pas vers le minimum de $F$ puis l'opérateur proposé dans l'équation (7) prend le point $W'$ le plus proche de $W$ qui respecte les contraintes portées par $G$.

Cette méthode fait donc apparaître dans l'équation (7) un problème de minimisation local, qui doit être résolu à chaque étape de l'itération, de façon à résoudre le problème global. La clef de l'efficacité de cette technique réside dans l'existence d'une solution analytique pour ce problème local, appelé opérateur proximal de $G$.

## 4.2 Un opérateur proximal pour $G$

Nous établissons dans le théorème suivant l'expression de l'opérateur proximal pour le terme de pénalité $G(W)$ défini dans l'équation (3). On note $K$ le vecteur $(1,\ 1, \ldots 1) \in \mathbb{R}^{1\times d}$, tel que $K \cdot K^{\intercal} = d$.

**Théorème 1** *Soit* $G_r(W_r) = |\sum_{j=1}^{d}(w_{rj}) - \alpha|$ *et* $L \in \mathbb{R}^*$*, alors :*

$$\mathrm{prox}_{\frac{\gamma}{L}G_r}(W_r) = W_r + \frac{1}{d}K^{\intercal} \cdot \big(\alpha + \mathrm{prox}_{\frac{\gamma d}{L}|\cdot|}(K \cdot W_r - \alpha) - K \cdot W_r\big) \quad (8)$$

*où* $\mathrm{prox}_{\lambda|\cdot|}(x) = \mathrm{sign}(x)\max(|x| - \lambda, 0)$.

*De plus,*

$$\mathrm{prox}_{\frac{\gamma}{L}G}(W) = \Big(\mathrm{prox}_{\frac{\gamma}{L}G_r}(W_r)\Big)_{r=1\ldots c} \in \mathbb{R}^{d\times c}$$

La preuve, non détaillée ici, repose sur l'utilisation des propriétés et expressions des opérateurs proximaux établis par [5] et [8].

L'équation (8) donne l'expression de l'opérateur proximal de la fonction $G$, qui peut être utilisé pour implémenter de façon efficace le schéma donné dans l'équation (6) de façon à optimiser le paramètre $W$ de la fonction $J$.

Comme pour la descente de gradient classique, le choix du pas de descente $L$ est critique pour assurer la convergence vers un minimum. Nous observons empiriquement que la valeur $L = \mathrm{Tr}(H^{-1})$ fournit de bons résultats, où $H$ est la matrice hessienne de $F$ (en tant que fonction de $W$) et Tr l'opérateur de trace. La fonction $F$ étant simple, $H$ est une matrice diagonale qui ne fait pas intervenir $W$.

### 4.3 Un algorithme de subspace clustering flou : PFSCM

En utilisant les résultats précédents, nous proposons l'algorithme PFSCM, décrit dans l'algorithme 1 : PFSCM combine l'optimisation alternée des $c$-moyennes floues pour les paramètres différentiables, et la descente proximale pour l'optimisation des poids.

L'initialisation est souvent une étape critique des algorithmes de type $k$-moyennes. Dans cet article, les centres sont tirés de façon aléatoire et les poids des dimensions sont initialement répartis de façon uniforme pour chaque cluster. À l'instar de la plupart des algorithmes de partitionnement, le nombre $c$ de clusters à identifier doit être fourni par l'utilisateur, de même que les constantes $\gamma$ et $\alpha$.

L'algorithme itère ensuite la mise à jour des trois paramètres $U$, $C$ et $W$ à la façon de l'optimisation alternée de l'algorithme des $c$-moyennes floues. Cette itération fait intervenir deux boucles : les paramètres internes $C$ et $U$ sont optimisés indépendamment de $W$, qui nécessite la procédure d'optimisation particulière de la section précédente. Les paramètres $C$ et $U$ sont optimisés une dernière fois à la fin de l'algorithme, de façon à garantir que le résultat prend en compte les poids finals.

**Données :** $X$ : données
**Paramètres :** $c$, $\gamma$, $\alpha$ : réels ;
**Variables :** $C$, U, W : matrices ;
W$_{last}$ : matrice
**Initialisation :** $\forall r, W_r \leftarrow (1,\ 1, \ldots\ 1)$ ;
$C \leftarrow c$ centres aléatoires
**Résultat :** $C$, U, W$_{last}$
**repeat**
  **repeat**
    Mise à jour U d'après équation (4) ;
    Mise à jour C d'après équation (5)
  **until** *convergence(C, U)* ;
  **repeat**
    Mise à jour W d'après équation (6)
  **until** *convergence(W)* ;
  W$_{last}$ ⟵ W
**until** *convergence(W$_{last}$)* ;
Mise à jour U et C.

**Algorithme 1 :** L'algorithme PFSCM

Les critères de convergence sont définis comme la distance entre la valeur courante et la valeur précédente des paramètres optimisés. En particulier, la convergence pour le couple $(C, U)$ est définie comme
$\|C_t - C_{t+1}\|_2 < \varepsilon \vee \|U_t - U_{t+1}\|_2 < \varepsilon$.

PFSCM renvoie trois matrices, $U$, $C$ et $W$. De façon à exploiter le résultat de l'algorithme, l'utilisateur peut devoir identifier la dimension de chaque cluster, c'est-à-dire le nombre de dimensions pertinentes pour le décrire. Ces dimensions sont celles dans lesquelles les données d'un même cluster sont les plus similaires. Dans ce but, nous proposons une étape de post-traitement de la matrice $W$ à l'aide d'un paramètre additionnel $cut > 0$ dont le but est d'éliminer de façon simple les dimensions non-pertinentes : pour un cluster $C_r$, une dimension $j$ est considérée comme pertinente si $w_{rj} > cut$.

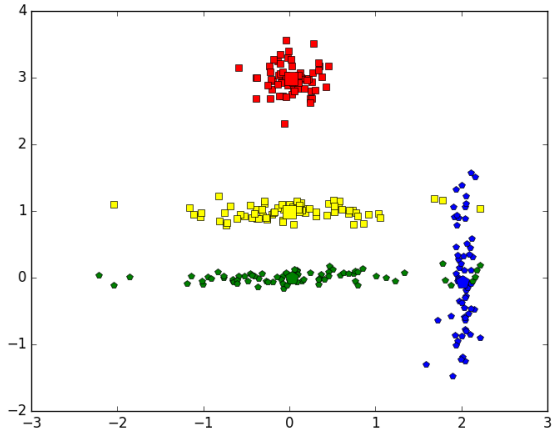

Figure 1 – Exemple de clustering en deux dimensions obtenu par PFSCM.

# 5 Étude expérimentale

Afin d'évaluer la pertinence de l'algorithme PFSCM, nous l'appliquons à des donnés artificielles constituées de clusters ellipsoïdaux. Nous évaluons la capacité de PFSCM à identifier à la fois les centres attendus ainsi que les dimensions pertinentes pour les décrire. PFSCM est comparé à l'algorithme de Keller et Klawonn (K&K) [6], et se montre plus efficace dans l'identification de l'importance relative des différentes dimensions.

## 5.1 Exemple illustratif

Cette section présente un exemple pour $d = 2$ dimensions, similaire à l'exemple donné par Keller & Klawonn [6] et représenté graphiquement sur la Figure 1 : quatre clusters sont générés, l'un d'eux (en rouge sur la Figure 1) est circulaire, les autres ont une variance très faible dans une des deux dimensions. PFSCM est appliqué avec les paramètres $c = 4$, $m = 2$ et $\alpha = 1$ de façon analogue à K&K. Le paramètre $\gamma$ nécessitant seulement d'être assez grand, on prend ici $\gamma = 1000$. Des expérimentations sur une large plage de valeurs ont mis en évidence la robustesse de l'algorithme par rapport à ce paramètre.

Dans la Figure 1 les points sont colorés en fonction du cluster $C_r$ pour lequel $u_{ri}$ est maximum, et le Tableau 1 donne les poids calculés pour chaque dimension et chaque cluster. On observe que PFSCM identifie correctement les clusters cherchés et leurs dimensions : les poids $(w_1, w_2)$ du cluster circulaire sont similaires, alors que les clusters horizontaux (respectivement verticaux) vérifient $w_2 \gg w_1$ (respectivement $w_1 \gg w_2$).

## 5.2 Protocole expérimental

**Données considérées.** Pour évaluer PFSCM, l'expérience précédente est généralisée en dimension supérieure pour $d \in \{5, 7, 9, 11, 13, 15\}$. Pour chaque expérience, $k = 4$ centres $c_1, \cdots, c_4$ sont générés aléatoirement dans l'hypercube $[-3{,}3]^d$ avec une distance euclidienne minimum de $0{,}3$ entre les centres. Ensuite, $d_r$ dimensions $j_1, \ldots j_{d_r}$ sont tirées aléatoirement, avec $d_r$ pris entre 1 et $d - 3$. Les dimensions $j_1, \cdots, j_{d_r}$ sont par la suite appelées dimensions pertinentes pour le cluster $C_r$.

Pour chaque cluster, 100 points sont générés selon une distribution gaussienne, avec une variance $v < 0,1$ pour les dimensions $j_1, \cdots, j_{d_r}$ et $v \in [0{,}5, 0{,}9]$ pour les autres dimensions.

**Paramètres des algorithmes.** L'algorithme K&K est initialisé avec des centres pré-calculés par l'algorithme des $c$-moyennes floues et utilise les paramètres $m = v = 2$, $a = 1$ and $c = 4$. PFSCM utilise les paramètres $m = 2$, $\alpha = 1$, $\gamma = 1000$ et $c = 4$. Les deux algorithmes utilisent le même critère de convergence, $\varepsilon = 10^{-4}$.

Le paramètre *cut* est défini comme $\frac{1}{2d}$, qui offre une évaluation des dimensions estimées comme pertinentes par les algorithmes en fonction de la dimension $d$.

**Critère de qualité.** Trois critères sont considérés pour évaluer la qualité des clusters retrouvés, ainsi que les dimensions des

Tableau 1 – Poids calculés par PFSCM pour les clusters de la Figure 1.

| Cluster rouge | | Cluster jaune | | Cluster vert | | Cluster bleu | |
|---|---|---|---|---|---|---|---|
| $w_1$ | $w_2$ | $w_1$ | $w_2$ | $w_1$ | $w_2$ | $w_1$ | $w_2$ |
| 0.528 | 0.472 | 0.063 | 0.937 | 0.027 | 0.973 | 0.964 | 0.036 |

sous-espaces identifiés.

Soit $\delta = \sum_{r=1}^{4} \|c_r - \mu_r\|_2$ la somme des distances euclidiennes entre les centres générés et obtenus $(\mu_r)$ : elle constitue un critère de qualité standard pour évaluer les clusters produits. Une valeur faible signifie que les centres calculés sont proches des centres originaux.

Nous considérons aussi $\theta$, défini comme le pourcentage de clusters pour lesquels toutes les dimensions retrouvées sont correctement identifiées par les algorithmes : les dimensions pertinentes sont correctement identifiées si $w_{rj} > cut \Leftrightarrow j \in \{j_1, \cdots j_{d_r}\}$.

Enfin, pour les clusters dont les dimensions pertinentes ont été correctement identifiées, soit $\phi = \frac{\omega_1}{\omega_{j_{d_r}}}$ le ratio des poids, où $\omega_1$ est le plus grand et $\omega_{j_{d_r}}$ est le plus petit des poids calculés pour les dimensions pertinentes. Ce critère évalue la distortion du cluster obtenu : un $\phi$ faible témoigne d'un meilleur respect des proportions relatives du cluster et, donc, une meilleure identification de son sous-espace.

### 5.3 Résultats expérimentaux

Les résultats sont présentés dans le Tableau 2 sous la forme de la moyenne et de l'écart-type $\sigma$ des deux critères $\delta$ et $\phi$ et du pourcentage moyen pour $\theta$, calculés pour $100$ exécutions de chaque algorithme. Parmi ces exécutions, et pour les deux algorithmes, on rejette les résultats obtenus fournissant des centres trop éloignés des centres attendus. Ceux-ci découlent d'une initialisation non pertinente et ne se produisent que dans moins de 2% des tests.

Les valeurs de $\delta$ indiquent que PFSCM identifie correctement les clusters générés et produit des résultats stables dans chaque dimension, comme le montre le faible écart-type. De plus, les valeurs de $\theta$ montrent que PFSCM retrouve les dimensions pertinentes pour décrire les sous-espaces de chaque cluster. Enfin, le ratio des poids $\phi$ est relativement stable quand le nombre de dimensions augmente.

L'algorithme K&K identifie aussi les centres et les différences avec PFSCM ne sont pas significatives sur ce plan. En revanche, l'algorithme semble sous-évaluer le nombre de dimensions pertinentes pour décrire les sous-espaces. Cette caractéristique est déjà évoquée par Keller & Klawonn [6] : alors que la dimension principale est presque toujours correctement identifiée, K&K produit un poids plus faible pour les autres dimensions pertinentes, ce qui est également montré par la moyenne plus élevée de $\phi$. Cette caractéristique peut être modulée en changeant la valeur de $v$, mais cela affecte alors les poids de toutes les dimensions, y compris celui de la dimension principale.

Ainsi, PFSCM identifie des clusters similaires à ceux de K&K mais produit une meilleure estimation des dimensions des sous-espaces. Il est aussi plus régulier quand la dimension $d$ augmente.

## 6 Conclusion et travaux futurs

Cet article introduit une nouvelle approche pour le problème du *subspace clustering* flou. Une fonction de coût originale est proposée, qui fait intervenir une somme de termes non-différentiables. L'utilisation de techniques d'optimisation avancées est proposée pour remplacer certaines des équations de mise à jour classiques de l'algorithme des $c$-moyennes floues. Une étude expérimentale sur des données artificielles valide la pertinence de l'approche proposée.

Tableau 2 – Comparaison entre PFSCM et K&K [6] en fonction de la dimension des données

| | $d$ | $\delta$ | | $\phi$ | | $\theta$ |
|---|---|---|---|---|---|---|
| | | Moy. | $\sigma$ | Moy. | $\sigma$ | % |
| PFSCM | 5 | 0.90 | 0.67 | 2.51 | 1.41 | 76 |
| | 7 | 0.98 | 0.81 | 3.08 | 1.72 | 79 |
| | 9 | 0.90 | 0.50 | 3.96 | 2.09 | 80 |
| | 11 | 0.88 | 0.33 | 4.35 | 2.01 | 83 |
| | 13 | 0.97 | 0.40 | 4.78 | 1.99 | 83 |
| | 15 | 0.90 | 0.10 | 5.22 | 1.84 | 91 |
| K&K | 5 | 1.27 | 1.03 | 2.61 | 1.78 | 43 |
| | 7 | 1.55 | 1.38 | 3.12 | 2.29 | 39 |
| | 9 | 1.39 | 1.18 | 4.05 | 3.01 | 31 |
| | 11 | 1.26 | 0.90 | 4.50 | 3.48 | 28 |
| | 13 | 1.42 | 1.29 | 4.68 | 3.60 | 25 |
| | 15 | 1.21 | 1.06 | 8.05 | 3.27 | 10 |

Les perspectives de ce travail incluent la généralisation à d'autres contraintes ou d'autres problèmes d'apprentissage : une fonction différentiable correspondant à la spécification d'un problème, et une ou plusieurs fonctions de pénalité, qui expriment des contraintes sur la forme de la solution. L'introduction de termes de régularisation pour d'autres paramètres que $W$ est également envisagée. Une comparaison avec d'autres schémas d'optimisation est aussi à considérer.

**Remerciements :**

Les auteurs remercient Nikhil Pal pour son aide sur ce travail. Ces travaux entrent dans le cadre du projet « REQUEST », financé par le programme « Investissement d'Avenir » (appel « Big Data - Cloud Computing »).

# Références

[1] Agrawal, R., Gehrke, J., Gunopulos, D. & Raghavan, P. 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, p. 94–105.

[2] Bach, F., Jenatton, R., Mairal, J., Obozinski, G. et al. Convex optimization with sparsity-inducing norms. In *Optimization for Machine Learning* (2012), p. 19–53.

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

[4] Borgelt, C. Fuzzy subspace clustering. In *Advances in Data Analysis, Data Handling and Business Intelligence*. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, 2010, p. 93–103.

[5] Combettes, P. L. & Pesquet, J.-C. Proximal Splitting Methods in Signal Processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Sous la dir. de Bauschke, H. H., Burachik, S. R., Combettes, L. P., Elser, V., Luke, R. D. & Wolkowicz, H. New York, NY : Springer New York, 2011, p. 185–212.

[6] Keller, A. & Klawonn, F. Fuzzy clustering with weighting of data variables. In *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 8.06 (2000), p. 735–746.

[7] Moreau, J.-J. Fonctions convexes duales et points proximaux dans un espace hilbertien. In *CR Acad. Sci. Paris Sér. A Math* 255 (1962), p. 2897–2899.

[8] Parikh, N. & Boyd, S. Proximal Algorithms. In *Foundations and Trends in Optimization* 1.3 (2014), p. 127–239.

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

[10] Tibshirani, R. Regression shrinkage and selection via the lasso. In *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), p. 267–288.

[11] Vidal, R. A tutorial on subspace clustering. In *IEEE Signal Processing Magazine* 28.2 (2010), p. 52–68.

[12] Wang, D., Ding, C. & Li, T. K-subspace clustering. In *Machine learning and knowledge discovery in databases*. Springer, 2009, p. 506–521.

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