



**HAL**  
open science

## Optimisation de Fuzzy C-Means (FCM) clustering par la méthode des directions alternées (ADMM)

Benoit Albert, Violaine Antoine, Jonas Koko

### ► To cite this version:

Benoit Albert, Violaine Antoine, Jonas Koko. Optimisation de Fuzzy C-Means (FCM) clustering par la méthode des directions alternées (ADMM). *Extraction et Gestion des Connaissances (EGC)*, 2023, Lyon, France. pp.247-258. hal-03974754

**HAL Id: hal-03974754**

**<https://hal.science/hal-03974754v1>**

Submitted on 24 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Optimisation de Fuzzy C-Means (FCM) clustering par la méthode des directions alternées (ADMM)

Benoit Albert\*, Violaine Antoine\*, Jonas Koko\*

\*LIMOS, Université Clermont Auvergne, France  
{benoit.albert,violaine.antoine,jonas.koko}@uca.fr

**Résumé.** Parmi les méthodes de classification non supervisée, K-Means et ses variantes sont très populaires. Ces méthodes résolvent à chaque itération les conditions d’optimalité du premier ordre. Cependant dans certains cas, la fonction à minimiser n’est pas convexe, comme pour la version Fuzzy C-Mean avec la distance de Mahalanobis (FCM-GK). Dans cette étude, nous appliquons la méthode des directions alternées (ADMM) afin d’assurer une bonne convergence. ADMM est une méthode souvent appliquée à la résolution d’un problème de minimisation convexe séparable avec des contraintes linéaires. ADMM est une méthode de décomposition/coordination avec une étape de coordination assurée par des multiplicateurs de Lagrange. En introduisant avec justesse des variables auxiliaires, cette méthode permet de décomposer le problème en sous problèmes convexes faciles à résoudre tout en gardant la même structure itérative. Les résultats numériques ont démontré la performance significative de la méthode proposée par rapport à la méthode standard surtout pour des données de grandes dimensions.

## 1 Introduction

Le partitionnement de données est un processus d’analyse des données qui consiste à partager les  $n$  objets d’un jeu de données en  $c$  sous-ensembles, dans le but que chaque groupe (sous ensemble) possède des objets similaires et que les groupes soient bien distincts entre eux (Jain et Dubes, 1988). Il permet de détecter des structures cachées dans les jeux de données sans connaissance préalable. Plusieurs approches différentes existent, les méthodes se distinguent par la nature des partitions créées. Elles peuvent être certaines : chaque objet appartient à un unique groupe. *k-means* est la plus célèbre des méthodes formant ce genre de partition. Chaque groupe est représenté par un centroïde (objet moyen), la notion de similarité est définie par la distance entre les objets et les centroïdes. Grâce à sa faible complexité et sa simplicité, cette méthode est très utilisée (Jain, 2010). Les partitions peuvent être floues permettant de modéliser l’incertitude. La variante floue des *k-means* est Fuzzy C-Means (FCM) (Bezdek, 1973 ; Bezdek et Dunn, 1975), chaque objet a un degré d’appartenance à chaque groupe. FCM est encore utilisée dans divers domaines (Anter et al., 2019 ; Yin et Li, 2020 ; Cai et al., 2021). La similarité entre les objets et les centroïdes dans l’algorithme FCM

est calculée par la distance euclidienne formant des groupes sphériques. L'algorithme de Gustafson et Kessel FCM-GK (Gustafson et Kessel, 1979) est une extension de FCM qui ajuste une distance pour chaque cluster. Cela permet de prendre en compte la forme des clusters et de détecter des structures ellipsoïdales. En effet, basé sur la distance de Mahalanobis, l'algorithme adapte des matrices symétriques définies positives interprétées comme les inverses des matrices de covariance floue des clusters. FCM et FCM-GK sont deux problèmes d'optimisation non convexe sous contraintes pour lesquelles la méthode d'optimisation standard est la méthode d'optimisation alternée (AO), une méthode itérative de type Gauss-Seidel.

La méthode des directions alternées, *Alternating Direction Method of Multiplier* (ADMM), est une méthode de décomposition-coordination simple mais puissante. Elle décompose le problème en sous-problèmes, les solutions obtenues localement sont coordonnées (par des multiplicateurs de Lagrange) pour trouver une solution au problème global. Cette méthode a été introduite au milieu des années 1970 par (Gabay et Mercier, 1976), (Glowinski R. et Marocco A., 1975) pour l'approximation numérique de problèmes convexes non lisses issus de la mécanique. Basée sur la formulation Lagrangienne augmentée, cette méthode a été utilisée dans de nombreux domaines en mécanique non linéaire (Glowinski R. et Le Tallec P., 1989; Fortin et Glowinski, 1983; Koko J., 2013, 2011; Glowinski R. et Marocco A., 1975), en restauration d'images (Koko J. et Jehan-Besson S., 2010), en réseaux neuronaux (F. Yu et Zhao, 2019), en optimisation à grande échelle (Eckstein, 1994; Koko J., 2013), etc. Un résumé des applications d'ADMM en apprentissage automatique est disponible dans (Boyd S., Parikh N., Chu E., Peleato B. et Eckstein J., 2011). ADMM standard se concentre sur la minimisation de fonctions séparables (convexes) avec des contraintes de couplages linéaires.

Dans cette étude, nous étendons l'application d'ADMM à la fonction coût non convexe de FCM-GK. ADMM divise le problème FCM-GK en une séquence de sous-problèmes plus simples et non couplés, grâce à l'introduction appropriée des variables auxiliaires inconnues. Ces sous-problèmes sont, non couplés, plus simples, donc plus faciles à résoudre. La formulation de la solution est proche de celle obtenue par l'optimisation alternée pour les variables originales (centroïdes, matrices d'appartenance reliées à la distance). Le sous-problème des variables auxiliaires conduit à la résolution de petits systèmes linéaires non couplés. Des expériences numériques sur des données de l'UCI machine learning montrent que l'algorithme FCM-ADMM proposé est robuste, insensible à l'initialisation aléatoire et crée généralement un meilleur partitionnement.

Le papier est organisé en 4 parties. La Section 2 présente le modèle GK et l'optimisation par la méthode standard (AO). Puis, nous décrivons dans la Section 3 l'application de la méthode ADMM dans ce contexte. Dans la Section 4, les expériences numériques sont exposées. Enfin la conclusion et les perspectives sont données dans la Section 5.

## 2 Le modèle FCM-GK

### 2.1 Problème d'optimisation

Soit le jeu de données représenté par  $\mathbf{X} = (\mathbf{x}_1 \dots \mathbf{x}_n)$  contenant  $n$  objets  $\mathbf{x}_i \in \mathbb{R}^p$ ,  $p$  est le nombre d'attributs et  $c$  le nombre de classes souhaité,  $2 \leq c < n$ . Le modèle

FCM-GK appliqué à  $\mathbf{X}$  consiste à calculer

— la matrice des degrés d'appartenance  $n \times c$ ,  $\mathbf{U} = (u_{ij})$  telle que,

$$u_{ij} \in [0, 1], \quad \sum_{j=1}^c u_{ij} = 1, \quad \sum_{i=1}^n u_{ij} > 0. \quad (2.1)$$

— les centres de gravité, centroïdes, de chaque groupe  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_c\}$ ,  $\mathbf{v}_j \in \mathbb{R}^p$  ;

— les matrices définies positives,  $\mathbf{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_c\}$ , induisant la norme de chaque groupe,  $\mathbf{S}_j \in \mathbb{R}^{p \times p}$ .

Les variables inconnues  $(\mathbf{U}, \mathbf{V}, \mathbf{S})$  sont déterminées en optimisant le problème suivant

$$\min_{(\mathbf{U}, \mathbf{V}, \mathbf{S})} J(\mathbf{U}, \mathbf{V}, \mathbf{S}) = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \mathbf{d}_{ij}^\top \mathbf{S}_j \mathbf{d}_{ij}, \quad (2.2)$$

avec les contraintes,

$$u_{ij} \geq 0, \quad \forall i, j \in [1, n] \times [1, c] \quad (2.3)$$

$$\sum_{j=1}^c u_{ij} = 1, \quad \forall i \in [1, n] \quad (2.4)$$

$$\sum_{i=1}^n u_{ij} > 0, \quad \forall j \in [1, c] \quad (2.5)$$

$$\det(\mathbf{S}_j) = \rho_j, \quad \forall j \in [1, c] \quad (2.6)$$

tel que la différence entre l'objet  $i$  et le centroïde  $j$  est notée :

$$\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{v}_j. \quad (2.7)$$

Toutes les différences  $\mathbf{d}_{ij}$  sont représentées par la variable  $\mathcal{D}$ .

En plus d'un nombre de classe  $c$ , ce modèle possède un autre hyperparamètre,  $m$  qui contrôle la dureté de la partition. Sans information supplémentaire,  $m$  est habituellement fixé à 2 (Pal et Bezdek, 1995). FCM est un cas particulier de FCM-GK où  $\mathbf{S}_j = \mathbb{I}$  in (2.2). Une solution triviale pour la minimisation est la solution avec toutes les matrices  $\mathbf{S}_j$  nulles. Pour éviter cela, la contrainte Eq. (2.6) est introduite fixant les volumes des ellipsoïdes.

## 2.2 Méthode d'optimisation alternée (AO)

La méthode utilisée par (Gustafson et Kessel, 1979) pour résoudre ce problème sous contrainte est la méthode d'optimisation alternée, *the alternating optimization method* (AO). Elle est également utilisée pour les autres variantes de *k-means*. Partant de  $(\mathbf{U}^0, \mathbf{V}^0, \mathbf{S}^0)$  la méthode minimise successivement  $\mathbf{U}$ ,  $\mathbf{V}$  et  $\mathbf{S}$  :

$$\mathbf{U}^{k+1} = \arg \min_{\mathbf{U} \in \mathcal{U}} J(\mathbf{U}, \mathbf{V}^k, \mathbf{S}^k), \quad (2.8)$$

$$\mathbf{V}^{k+1} = \arg \min_{\mathbf{V}} J(\mathbf{U}^{k+1}, \mathbf{V}, \mathbf{S}^k), \quad (2.9)$$

$$\mathbf{S}^{k+1} = \arg \min_{\mathbf{S} \in \mathcal{S}_1} J(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}, \mathbf{S}). \quad (2.10)$$

Application de la méthode ADMM pour le modèle FCM-GK

Avec les deux ensembles correspondant aux contraintes (2.3)-(2.5) et (2.6) :

$$\mathcal{U} = \left\{ u_{ij} \geq 0, \sum_{j=1}^c u_{ij} = 1, \sum_{i=1}^n u_{ij} > 0 \right\},$$

$$\mathcal{S}_1 = \{ \mathbf{S}, \quad p \times p \text{ matrice symétrique définie positive, } \det(\mathbf{S}) = 1 \}.$$

En utilisant les conditions d'optimalité du premier ordre, les solutions de (2.8)-(2.10) sont calculées,  $\forall i, j \in [1, n] \times [1, c]$  :

$$u_{ij}^{k+1} = \left[ \sum_{\ell=1}^c \frac{(\mathbf{d}_{ij}^k)^\top \mathbf{S}_j^k \mathbf{d}_{ij}^k}{(\mathbf{d}_{i\ell}^k)^\top \mathbf{S}_\ell^k \mathbf{d}_{i\ell}^k} \right]^{-1}, \quad (2.11)$$

$$\mathbf{v}_j^{k+1} = \frac{\sum_{i=1}^n u_{ij}^{k+1} \mathbf{x}_i}{\sum_{i=1}^n u_{ij}^{k+1}}, \quad (2.12)$$

$$\Sigma_j^{k+1} = \sum_{i=1}^n u_{ij}^{k+1} \mathbf{d}_{ij}^{k+1} (\mathbf{d}_{ij}^{k+1})^\top, \quad (2.13)$$

$$\mathbf{S}_j^{k+1} = \det(\Sigma_j)^{\frac{1}{p}} (\Sigma_j^{k+1})^{-1}. \quad (2.14)$$

L'algorithme FCM-GK est présenté (algorithme 1). Il s'arrête lorsque la partition est stabilisée autrement dit que l'erreur absolue entre deux matrices  $\mathbf{U}$  (degrés d'appartenance) successives plus petite que  $10^{-3}$ . Remarquons que pour  $t$  itérations, sa complexité est en  $O(tnc^2p)$

---

#### Algorithme 1 FCM-GK

---

```

1 : Entrée :  $c$ 
2 :  $err = 0, k = 0,$ 
3 :  $\mathbf{U}^0$  initialisation aléatoire ou via FCM.
4 : tant que  $err > 10^{-3}$  faire
5 :    $k \leftarrow k + 1$ 
6 :    $\mathbf{V}^k$  selon (2.12)
7 :    $\mathbf{S}^k$  selon (2.14)
8 :    $\mathbf{U}^k$  selon (2.11)
9 :    $err \leftarrow \|\mathbf{U}^k - \mathbf{U}^{k-1}\|$ 
10 : fin tant que
11 : Sortie :  $\mathbf{U}^k, \mathbf{V}^k, \mathbf{S}^k$ 

```

---

### 3 Méthode ADMM

La principale idée de la méthode ADMM, *Alternating Direction Methods of Multipliers*, (Glowinski R. et Marocco A., 1975 ; Gabay et Mercier, 1976 ; Fortin et Glowinski, 1983), est d'utiliser un processus de décomposition/coordination dont la coordination est réalisée par les multiplicateurs de Lagrange.

### 3.1 Formulation du Lagrangien augmenté

ADMM ne minimise pas seulement la fonction objectif mais le Lagrangien augmenté associé au problème. Avant de formuler ce dernier, il est nécessaire d'introduire des variables auxiliaires dans le problème original afin d'obtenir un problème d'optimisation par bloc sous contraintes. Tout d'abord, nous écrivons les fonctions caractéristiques des contraintes originales pour les introduire dans la fonction à minimiser.

$$I_{\mathcal{U}}(\mathbf{U}) = \begin{cases} 0 & \text{si } \mathbf{U} \in \mathcal{U} \\ +\infty & \text{sinon,} \end{cases} \quad (3.1)$$

$$I_{\mathcal{S}_1}(\mathbf{S}_j) = \begin{cases} 0 & \text{si } \mathbf{S}_j \in \mathcal{S}_1 \\ +\infty & \text{sinon.} \end{cases} \quad (3.2)$$

et  $I_{\mathcal{S}_1}(\mathbf{S}) = \sum_j I_{\mathcal{S}_1}(\mathbf{S}_j)$ . En plus des variables auxiliaires  $\mathcal{D}$  (2.7), nous introduisons les variables  $\mathcal{P}$

$$\mathbf{p}_{ij} = u_{ij} \mathbf{d}_{ij} = u_{ij} (\mathbf{x}_i - \mathbf{v}_j).$$

Ainsi, nous reformulons la fonction coût en (2.2) qui devient :

$$J(\mathbf{U}, \mathbf{V}, \mathbf{S}, \mathcal{D}, \mathcal{P}) = \sum_{i=1}^n \sum_{j=1}^c \mathbf{p}_{ij}^\top \mathbf{S}_j \mathbf{p}_{ij}. \quad (3.3)$$

Pour alléger les écritures nous notons :  $\mathbf{U} = (\mathbf{U}, \mathbf{V}, \mathbf{S})$  l'ensemble des variables du problème et  $\mathbf{D} = (\mathcal{D}, \mathcal{P})$  l'ensemble des variables auxiliaires. Le problème de minimisation sous contrainte devient (2.2)-(2.10)

$$\min J(\mathbf{U}, \mathbf{D}) + I_{\mathcal{U}}(\mathbf{U}) + I_{\mathcal{S}}(\mathbf{S}), \quad (3.4)$$

$$\text{sous contraintes} \quad (3.5)$$

$$\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{v}_j, \quad (3.6)$$

$$\mathbf{p}_{ij} = u_{ij} \mathbf{d}_{ij}. \quad (3.7)$$

Les contraintes de couplages sont définies de manière à garantir l'équivalence (en terme de solution) avec le problème d'origine (2.2)-(2.10), tout en permettant une optimisation indépendante selon les variables.

Avec (3.4)-(3.7), la fonction du Lagrangien augmenté est :

$$\begin{aligned} \mathcal{L}_r(\mathbf{U}, \mathbf{D}, \mathbf{Y}) &= J(\mathbf{U}, \mathbf{D}) + I_{\mathcal{U}}(\mathbf{U}) + I_{\mathcal{S}_1}(\mathbf{S}) \\ &+ \sum_{i,j} [\mathbf{y}_{ij}^\top (\mathbf{d}_{ij} - \mathbf{x}_i + \mathbf{v}_j) + \mathbf{z}_{ij}^\top (\mathbf{p}_{ij} - u_{ij} \mathbf{d}_{ij})] \\ &+ \frac{r}{2} \sum_{i,j} [\| \mathbf{d}_{ij} - \mathbf{x}_i + \mathbf{v}_j \|^2 + \| \mathbf{p}_{ij} - u_{ij} \mathbf{d}_{ij} \|^2] \end{aligned} \quad (3.8)$$

où  $r > 0$  est le terme de pénalité,  $\| \cdot \|$  est la norme euclidienne,  $\mathbf{y}_{ij}$  et  $\mathbf{z}_{ij}$  sont les multiplicateurs de Lagrange associés aux contraintes sur les variables auxiliaires (3.6) et (3.7), représentés par  $\mathbf{Y} = (\mathbf{Y}, \mathbf{Z})$ .

### 3.2 Application d'ADMM

Nous appliquons la méthode ADMM au Lagrangien augmenté (3.8) par l'algorithme itératif suivant. Commençons avec  $\mathbf{D}^0 : (\mathcal{D}^0, \mathcal{P}^0)$  et  $\mathbf{Y}^0 : (\mathcal{Y}^0, \mathcal{Z}^0)$ , on calcule successivement  $\mathbf{U}^k : (\mathcal{U}^k, \mathcal{V}^k, \mathcal{S}^k)$ ,  $\mathbf{D}^k : (\mathcal{D}^k, \mathcal{P}^k)$  et  $\mathbf{Y}^k : (\mathcal{Y}^k, \mathcal{Z}^k)$  comme suite.

$$\mathbf{U}^{k+1} = \arg \min_{\mathbf{U}} \mathcal{L}_r(\mathbf{U}, \mathbf{D}^k, \mathbf{Y}^k), \quad (3.9)$$

$$\mathbf{D}^{k+1} = \arg \min_{\mathbf{D}} \mathcal{L}_r(\mathbf{U}^{k+1}, \mathbf{D}, \mathbf{Y}^k), \quad (3.10)$$

$$\mathbf{y}_{ij}^{k+1} = \mathbf{y}_{ij}^k + r(\mathbf{d}_{ij}^{k+1} - \mathbf{x}_i + \mathbf{v}_j^{k+1}), \quad (3.11)$$

$$\mathbf{z}_{ij}^{k+1} = \mathbf{z}_{ij}^k + r(\mathbf{p}_{ij}^{k+1} - u_{ij}^{k+1} \mathbf{d}_{ij}^{k+1}). \quad (3.12)$$

Il faut remarquer que les itérations de la méthode ADMM (3.9)-(3.12) admettent des mises à jour exactes si la fonction est bi-convexe, c'est-à-dire convexe selon  $\mathbf{U}$  pour  $\mathbf{D}$  fixé et réciproquement et si les contraintes sont bi-affines, c'est-à-dire affines en  $\mathbf{U}$  pour  $\mathbf{D}$  fixé et réciproquement (Boyd S., Parikh N., Chu E., Peleato B. et Eckstein J., 2011). Dans (3.4),  $I_{S_1}(\mathcal{S})$  est non convexe à cause de la contrainte  $\det(\mathcal{S}_j) = 1$ . Pour assurer la convergence de la méthode, il suffit de fixer un nombre  $it_a$  de répétitions des blocs de relaxation (3.9)-(3.10) avant la mise à jour des multiplicateurs (Glowinski R. et Le Tallec P., 1989; Koko J., 2013). Il est conseillé  $it_a = 5$ .

#### 3.2.1 Solution du sous problème (3.9) en $\mathbf{U}$

Supposons les variables auxiliaires  $\mathbf{D}$  et les multiplicateurs  $\mathbf{Y}^k$  fixés, le problème (3.9) du Lagrangien augmenté (3.8) est découpé selon chaque variable de  $\mathbf{U}$ , à optimiser séparément

$$\begin{aligned} \mathcal{V}^{k+1} = \arg \min_{\mathcal{V}} \sum_{i=1}^n \sum_{j=1}^c (\mathbf{y}_{ij}^k)^\top (\mathbf{d}_{ij}^k - \mathbf{x}_i + \mathbf{v}_j) \\ + \frac{r}{2} \|\mathbf{d}_{ij}^k - \mathbf{x}_i + \mathbf{v}_j\|^2, \end{aligned} \quad (3.13)$$

$$\begin{aligned} \mathcal{U}^{k+1} = \arg \min_{\mathcal{U}} I_{\mathcal{U}}(\mathcal{U}) + \sum_{i=1}^n \sum_{j=1}^c (\mathbf{y}_{ij}^k)^\top (\mathbf{p}_{ij}^k - u_{ij} \mathbf{d}_{ij}^k) \\ + \frac{r}{2} \|\mathbf{p}_{ij}^k - u_{ij} \mathbf{d}_{ij}^k\|^2, \end{aligned} \quad (3.14)$$

$$\mathcal{S}^{k+1} = \arg \min_{\mathcal{S}} \sum_{i=1}^n \sum_{j=1}^c (\mathbf{p}_{ij}^k)^\top \mathcal{S}_j \mathbf{p}_{ij}^k + I_{S_1}(\mathcal{S}). \quad (3.15)$$

Les sous problèmes (3.13)-(3.15) sont résolus en prenant les conditions d'optimalité du premier ordre, comme pour la méthode AO. De ce fait, les formulations obtenues sont

assez proches

$$\mathbf{v}_j^{k+1} = \frac{1}{n} \sum_{i=1}^n \left( \mathbf{x}_i - \mathbf{d}_{ij}^k - \frac{1}{r} \mathbf{y}_{ij}^k \right), \quad (3.16)$$

$$u_{ij}^{k+1} = \frac{1}{r^2 \alpha_i^k \|\mathbf{d}_{ij}^k\|^2} C_{ij}^k, \quad (3.17)$$

$$\mathbf{S}_j^{k+1} = \det(\boldsymbol{\Sigma}_j^k)^{1/p} (\boldsymbol{\Sigma}_j^k)^{-1}, \quad (3.18)$$

avec,

$$\begin{aligned} C_{ij}^k &= \left[ r \alpha_i^k (\mathbf{d}_{ij}^k)^\top \tilde{\mathbf{z}}_{ij}^k + 1 - \sum_{\ell=1}^c \frac{(\mathbf{d}_{i\ell}^k)^\top \tilde{\mathbf{z}}_{i\ell}^k}{\|\mathbf{d}_{i\ell}^k\|^2} \right], \\ \tilde{\mathbf{z}}_{ij}^k &= \mathbf{z}_{ij}^k + r \mathbf{p}_{ij}^k, \\ \alpha_i^k &= \frac{1}{r} \sum_{j=1}^c \frac{1}{\|\mathbf{d}_{ij}^k\|^2}, \\ \boldsymbol{\Sigma}_j^k &= \sum_{i=1}^n \mathbf{p}_{ij}^k (\mathbf{p}_{ij}^k)^\top. \end{aligned}$$

### 3.2.2 Solution du sous problème (3.10) en $\mathbf{D}$

Le sous problème en  $\mathbf{D}$  :  $(\mathcal{D}, \mathcal{P})$  est un problème d'optimisation sans contrainte. Comme  $\mathbf{D} \mapsto F(\mathbf{D}) = \mathcal{L}_r(\mathbf{U}^{k+1}, \mathbf{D}, \mathbf{Y}^k)$  est quadratique, l'unique solution est obtenue en résolvant l'équation du gradient  $\nabla F(\mathbf{D}) = 0$ . Un calcul simple permet d'obtenir le système linéaire suivant en  $(\mathbf{d}_{ij}, \mathbf{p}_{ij})$

$$r(1 + (u_{ij}^{k+1})^2) \mathbf{d}_{ij} - r u_{ij}^{k+1} \mathbf{p}_{ij} = u_{ij}^{k+1} \mathbf{z}_{ij}^k - \mathbf{y}_{ij}^k + r(\mathbf{x}_i - \mathbf{v}_j^{k+1}) \quad (3.19)$$

$$-r u_{ij}^{k+1} \mathbf{d}_{ij} + (2\mathbf{S}_j^{k+1} + r\mathbb{I}) \mathbf{p}_{ij} = -\mathbf{z}_{ij}^k \quad (3.20)$$

Il s'ensuit qu'à chaque itération, on résout  $nc$  systèmes linéaires de taille  $2p$

$$\mathbf{A}_{ij}^k \begin{bmatrix} \mathbf{d}_{ij}, \\ \mathbf{p}_{ij} \end{bmatrix} = \mathbf{b}_{ij}^k \quad (3.21)$$

$$\mathbf{A}_{ij}^k = \begin{bmatrix} r(1 + (u_{ij}^{k+1})^2)\mathbb{I} & -r u_{ij}^{k+1} \mathbb{I} \\ -r u_{ij}^{k+1} \mathbb{I} & 2\mathbf{S}_j^{k+1} + r\mathbb{I} \end{bmatrix},$$

$$\mathbf{b}_{ij}^k = \begin{bmatrix} u_{ij}^{k+1} \mathbf{z}_{ij}^k - \mathbf{y}_{ij}^k + r(\mathbf{x}_i - \mathbf{v}_j^{k+1}) \\ -\mathbf{z}_{ij}^k \end{bmatrix}.$$



### 3.2.3 Algorithme

Pour résumer la méthode ADMM, nous présentons l'algorithme 2. La condition d'arrêt est désormais l'erreur relative sur toutes les variables primales et duales inférieure à  $10^{-3}$ . Pour  $t$  itérations, la complexité de notre méthode est la même en  $O(tnc^2p)$ . Nous devrions initialiser ADMM par  $\mathbf{D}$  aléatoire, mais il est plus simple de commencer avec  $\mathbf{U}$  aléatoire et de construire toutes les autres variables, via les formules (2.12)-(2.14) et (3.6)-(3.7). Les multiplicateurs de Lagrange sont initialisés en résolvant la condition d'optimalité du première ordre (3.4)-(3.7) (dériver le lagrangien selon les variables  $\mathbf{D}, \mathbf{P}$ ) :  $\mathbf{z}_{ij}^0 = 2\mathbf{S}_j^0\mathbf{p}_{ij}^0, \mathbf{y}_{ij}^0 = u_{ij}^0\mathbf{z}_{ij}^0, \forall i, j$ .

---

#### Algorithme 2 ADMM

---

```

1 : Entrée :  $c, r$ 
2 :  $err = 1, k = 0$ ,
3 : Initialisation aléatoire ou via ADMM(euclidien).
4 : tant que  $err > 10^{-3}$  faire
5 :    $k \leftarrow k + 1$ 
6 :   pour  $k_2 = 1$  jusqu'à  $it_a = 5$  faire
7 :      $\mathcal{V}^k, \mathcal{S}^k$  et  $\mathbf{U}^k$  respectivement selon (3.16), (3.18) et (3.17)
8 :      $\mathbf{D}^k, \mathbf{P}^k$  selon la résolution du système (3.21)
9 :   fin pour
10 :   $\mathcal{Y}^k, \mathcal{Z}^k$  respectivement selon (3.11) et (3.12)
11 :   $err \leftarrow \|(\mathbf{U}, \mathbf{D})^k - (\mathbf{U}, \mathbf{D})^{k-1}\| / \|(\mathbf{U}, \mathbf{D})^k\|$ 
12 : fin tant que
13 : Sortie :  $\mathbf{U}^k, \mathcal{V}^k, \mathcal{S}^k$ 

```

---

## 4 Expériences numériques

Dans cette section, nous avons étudié les performances de notre méthode ADMM pour le problème FCM avec la distance de Mahalanobis. Nous avons utilisé Matlab (R2021). Le terme de pénalité  $r$  influence la performance d'ADMM. Afin d'affiner au mieux ce paramétrage, nous avons normalisé toutes les données entre  $[-1, 1]$ , i.e.,  $-1 \leq x_{il} \leq 1, \forall i, l \in [1, n] \times [1, p]$ . Le choix de la valeur est très important, si  $r$  est trop petit, la convergence n'est pas assurée. Le problème doit être au moins fermé, il faut que la pénalité assure la coordination des variables. Pour ce faire, nous choisissons de donner par défaut une importance égale entre la partie augmentée du Lagrangien et la fonction objectif. Les données étant entre -1 et 1, la distance maximale est de  $2\sqrt{2} (< 4)$  sur les  $p$  attributs,  $n$  objets et  $c$  classes, d'où  $r_d = 4cnp$ . Pour trouver la valeur optimale  $r^*$ , nous testons plusieurs valeurs et conservons celle qui converge le plus rapidement (itérations).

Dans notre étude, nous avons fixé  $m = 2$  et  $\rho_j = 1, \forall j \in [1, c]$ . Nous comparons les 3 algorithmes suivants :

- **FCM-GK**, la méthode originale avec l'optimisation alternée sur le modèle GK.

- **ADMM** $_{r^*}$ , ADMM appliquée au Lagrangien augmenté (3.8), avec la valeur de pénalité optimale  $r^*$ .
- **ADMM** $_{r_d}$ , le même algorithme avec la valeur par défaut  $r_d$ .

Afin d'évaluer ces différentes méthodes, nous allons utiliser un critère d'évaluation externe qui mesure la ressemblance entre deux partitions. Ici, les données sont étiquetées, nous pouvons les comparer avec celles obtenues par les méthodes. Il est cependant nécessaire de transformer les partitions floues en partitions dures en affectant à chaque objet la classe ayant la plus forte appartenance.

Nous avons utilisé l'Adjusted Rand Index (ARI) introduit par (Hubert et Arabie, 1985). Soient deux partitions  $\pi_1$  et  $\pi_2$ , on note  $a$  le nombre de paires d'objets qui sont dans le même groupe dans  $\pi_1$  et  $\pi_2$ ,  $b$  le nombre de paires d'objets qui sont dans des groupes différents dans  $\pi_1$  et  $\pi_2$ ,  $c$  le nombre de paires qui sont dans le même groupe dans  $\pi_1$  mais pas dans  $\pi_2$  et  $d$  le nombre de paires qui sont dans le même groupe dans  $\pi_2$  mais pas dans  $\pi_1$ .

$$ARI(\pi_1, \pi_2) = \frac{2(ab - cd)}{(a + d)(d + b) + (a + c)(c + b)}$$

Nous avons utilisé 11 jeux de données. Les 5 premiers sélectionnés sont des données réelles issues de la librairie UCI<sup>1</sup> : IRIS, WINE, SEEDS, WDBC et DRYBEAN. Nous avons référencé dans le tableau 1 leurs caractéristiques, c'est-à-dire le nombre de classes  $c$ , d'objets  $n$  et d'attributs  $p$ , ainsi que le paramètre de pénalité optimale  $r^*$  et par défaut  $r_d$ . Nous avons aussi utilisé 6 jeux de données synthétiques<sup>2</sup>. Deux jeux issus

	IRIS	WINE	SEEDS	WDBC	DRYBEAN
$c$	3	3	3	2	7
$n$	150	178	210	569	13611
$p$	4	13	7	30	16
$r^*$	30	40	1500	1700	1.710 <sup>5</sup>
$r_d$	7200	27768	17640	136560	6.097.728

TAB. 1 – *Caractéristiques des jeux de données UCI.*

de 3 familles A sets, DIM sets et S sets. Leurs caractéristiques sont présentées dans le tableau 2.

Pour une insensibilité des résultats à l'initialisation, nous avons d'abord exécuté ADMM avec la distance euclidien ( $ADMM_{eu}$ ) avec  $r = 2, 5$  et fixant un nombre d'itérations maximal à 50 en partant avec  $U^0$  aléatoire.

Le tableau 3 permet de constater que les méthodes ADMM sont globalement plus performantes que la méthode FCM-GK. Sauf pour DRYBEAN, où FCM-GK est meilleur. Il semble que le nombre bien plus important d'individus par classe et le ratio nombre de classes et nombre d'individus expliquent ce comportement.

Le tableau 4, correspondant aux résultats pour les données synthétiques, confirme cette caractéristique : plus le nombre de classes est important (A1, A3) moins le score ARI

1. <https://archive.ics.uci.edu/ml/datasets.php>

2. <https://cs.joensuu.fi/sipu/datasets/>

Application de la méthode ADMM pour le modèle FCM-GK

	A1	A3	DIM32	DIM64	S1	S3
$c$	20	50	16	16	15	15
$n$	3000	7500	1024	1024	5000	5000
$p$	2	2	32	64	2	2
$r^*$	10	10	500	500	20	20
$r_d$	$4, 8 \cdot 10^4$	$3 \cdot 10^6$	$2^{21}$	$2^{22}$	$6 \cdot 10^5$	$6 \cdot 10^5$

TAB. 2 – *Caractéristiques des jeux synthétiques.*

	IRIS	WINE	SEEDS	WDBC	DRYBEAN
FCM-GK	0.74	0.34	0.72	0.41	<b>0.70</b>
ADMM $_{r^*}$	<b>0.78</b>	0.81	0.71	<b>0.74</b>	0.32
ADMM $_{r_d}$	0.72	<b>0.90</b>	0.71	<b>0.74</b>	0.32

TAB. 3 – *Score ARI (UCI).*

sera élevé. En revanche, plus le nombre de dimensions (DIM32, DIM64) est importante, meilleur est le score.

	A1	A3	DIM32	DIM64	S1	S3
FCM-GK	<b>0.90</b>	<b>0.93</b>	0.44	0.18	<b>0.97</b>	<b>0.66</b>
ADMM $_{r^*}$	0.23	0.16	<b>0.57</b>	<b>0.68</b>	0.33	0.24
ADMM $_{r_d}$	0.20	0.16	<b>0.57</b>	<b>0.68</b>	0.33	0.26

TAB. 4 – *Score ARI (Données synthétiques).*

Le tableau 5 recense le temps d'exécution des algorithmes lorsqu'il est significatif (supérieur à une seconde), mesuré pour 10 initialisations aléatoires différentes (ADMM $_{eu}$ ). Nous observons que les méthodes ADMM sont souvent plus rapides. Le temps d'exécution est intrinsèquement lié au nombre d'itérations ( $k$ ) qui varie selon les jeux de données généralement entre 2 et 4 pour ADMM $_{r_d}$ , 4 et 10 ADMM $_{r^*}$  et 50 et 200 pour FCM-GK.

## 5 Conclusion

Nous avons proposé une application de la méthode ADMM pour le modèle de clustering FCM avec la distance de Mahalanobis. L'intérêt de cette méthode est de diviser le problème en une séquence de sous-problèmes plus simples, faciles à résoudre. La convergence vers le même minimum, supposé global, est assurée.

Les résultats obtenus sur plusieurs jeux de données (réels ou synthétiques) montrent de bonnes performances, en termes de ratios d'échantillons bien classés, lorsque le nombre de classes n'est pas trop grand ou lorsque le nombre de dimensions est significativement

	A3	DIM32	DIM64	DRYBEAN	S3
FCM-GK	129±40	14±4	30±7	69±3	14±12
ADMM <sub>r,*</sub>	<b>8.3</b> ±0.1	13.2±0.1	49±0.2	72±3	2.5±0.1
ADMM <sub>r,d</sub>	8.5±0.1	<b>8.1</b> ±0.1	<b>33</b> ±0.1	<b>63</b> ±2	<b>1.6</b> ±0.1

TAB. 5 – Temps CPU en seconde (moyenne ± écart type).

plus élevé. Dans le cas contraire, le score ARI est moins bon que la méthode FCM-GK, notamment en deux dimensions. Pour simplifier l'utilisation de notre méthode, nous avons proposé une valeur par défaut pour le terme de pénalité (hyper paramètre), dont la convergence est assurée et proche de celle de la valeur optimale.

Les résultats sont très encourageants. Pour les confirmer, nous désirons appliquer notre méthode à un jeu de données provenant de la biologie, où de nombreux objets à classifier possèdent un grand nombre d'attributs. Pour faciliter l'utilisation de notre méthode, une formulation avec une pénalité adaptative est envisagée pour remplacer l'étude du  $r$  optimal. Enfin notre étude ouvre la possibilité d'appliquer la méthode ADMM à d'autres méthodes de clustering, ayant une fonction objective non convexe et en particulier celles utilisant actuellement l'optimisation alternée.

## Références

- Anter, A. M., A. E. Hassenian, et D. Oliva (2019). An improved fast fuzzy c-means using crow search optimization algorithm for crop identification in agricultural. *Expert Systems with Applications* 118, 340–354.
- Bezdek, J. et J. Dunn (1975). Optimal fuzzy partitions : A heuristic for estimating the parameters in a mixture of normal distributions. *IEEE Transactions on Computers* 100(8), 835–838.
- Bezdek, J. C. (1973). *Fuzzy Mathematics in pattern classification*. Cornell University.
- Boyd S., Parikh N., Chu E., Peleato B. et Eckstein J. (2011). Distributed optimization and statistical learning via alternating direction method of multipliers. *Foundation and Trends in Machine Learning* 3, 1–122.
- Cai, W., B. Zhai, Y. Liu, R. Liu, et X. Ning (2021). Quadratic polynomial guided fuzzy c-means and dual attention mechanism for medical image segmentation. *Displays* 70, 102106.
- Eckstein, J. (1994). Parallel alternating direction multiplier decomposition of convex-programs. *Journal of Optimization Theory and Applications* (1), 39–62.
- F. Yu, X. C. et L. Zhao (2019). ADMM for efficient deep learning with global convergence. In *KDD 19 : Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, pp. 111–119.
- Fortin, M. et R. Glowinski (1983). *Augmented Lagrangian Methods : Applications to the Numerical Solution of Boundary-Value Problems*. Amsterdam : North-Holland.

- Gabay, D. et B. Mercier (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Computers and Mathematics with Applications* 2, 17–40.
- Glowinski R. et Le Tallec P. (1989). *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*. Philadelphia : SIAM.
- Glowinski R. et Marocco A. (1975). Sur l'approximation par éléments finis d'ordre un, et la résolution par pénalisation–dualité, d'une classe de problèmes de Dirichlet non linéaires. *RAIRO-AN* 9(2), 41–76.
- Gustafson, D. et W. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. In *1978 IEEE conference on decision and control including the 17th symposium on adaptive processes*, pp. 761–766. IEEE.
- Hubert, L. et P. Arabie (1985). Comparing partitions. *Journal of classification* 2(1), 193–218.
- Jain, A. K. (2010). Data clustering : 50 years beyond k-means. *Pattern recognition letters* 31(8), 651–666.
- Jain, A. K. et R. C. Dubes (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.
- Koko J. (2011). Uzawa block relaxation for the unilateral contact problem. *Journal Computational Applied Mathematics* 235, 2343–2356.
- Koko J. (2013). Parallel Uzawa method for large-scale minimization of partially separable functions. *Journal Optimization Theory Applications* 158, 172–187.
- Koko J. et Jehan-Besson S. (2010). An augmented lagrangian method for  $TV_g+L^1$ -norm minimization. *Journal Mathematica Imaging Vision* 38, 182–196.
- Pal, N. R. et J. C. Bezdek (1995). On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy systems* 3(3), 370–379.
- Yin, S. et H. Li (2020). Hot region selection based on selective search and modified fuzzy c-means in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13, 5862–5871.

## Summary

Among the clustering methods, K-Means and variants are very popular. These methods solve at each iteration the first order optimality conditions. However, in some cases, the function to be minimized is not convex, as for the Fuzzy C-Means version with Mahalanobis distance (FCM-GK). In this study, we apply the Alternating Directions Method of Multiplier (ADMM) to ensure a good convergence. ADMM is often applied to solve a separable convex minimization problem with linear constraints. ADMM is a decomposition/coordination method with a coordination step provided by Lagrange multipliers. By appropriately introducing auxiliary variables, this method allows the problem to be decomposed into easily solvable convex subproblems while keeping the same iterative structure. Numerical results have demonstrated the significant performance of the proposed method compared to the standard method especially for high dimensional data.