Apprentissage Non Supervisé : Co-clustering Robuste dans un Contexte de Données Contaminées

Nafissa BENALI & Lucas BIECHY
Professeur : Christine KERIBIN

Papier de référence
Co-clustering contaminated data : a robust model-based approach
Fibbi et al. (2023)

Institut Mathématique d'Orsay(IMO)
Paris-Saclay University

1er décembre 2023

Résumé

A complete reevaluation of exploratory and pre-processing techniques is underway in response to the substantial growth in dataset dimensions. One widely employed methodology in this context is co-clustering, particularly grounded in probabilistic models. The escalating dimensions also increases the likelihood of encountering outliers within datasets, necessitating the development of novel models tailored to address this emerging challenge. This brings us to Fibbi et al. (2023)'s paper, which we scrutinize within the course named Unsupervised Learning. Following a comprehensive exposition on co-clustering, we introduce an innovative model characterized by a unique likelihood function. Subsequently, we illustrate how, via an iterative algorithm, it becomes viable to estimate both this likelihood function and the co-clusters. This novel algorithm exhibits monotonic likelihood growth, ensuring convergence under specific conditions while maintaining a computationally feasible complexity, ultimately enhancing model robustness. Despite our inability to reproduce this algorithm, we meticulously examine the results of its application to two disparate datasets. We engage in a comprehensive discussion of the algorithm's limitations and elaborate on the initiatives we pursued to push these mathematical frontiers further. Finally, we will conclude with insights into potential improvements for this algorithm.

Keywords : Co-clustering · Robustness · Trimming · LBM · CEM algorithm

Table des matières

1 Introduction

La croissance exponentielle des dimensions des jeux de données impose aujourd'hui une réévaluation approfondie des techniques exploratoires et de pré-traitement qui leur sont appliquées. Parmi ces approches, le co-clustering se démarque en regroupant simultanément les lignes et les colonnes d'un tableau de données, visant ainsi à réduire sa complexité. Cette méthode, largement utilisée dans des domaines aussi variés que la biologie, les systèmes de recommandation et l'analyse de texte, suscite un intérêt croissant.

Du fait de cette expansion, la probabilité d'apparition de valeurs aberrantes (*outliers*) augmente. Malheureusement, peu d'algorithmes de co-clustering sont conçus pour prendre en compte ce phénomène. Des tentatives ont été faites, comme celle de Cuesta-Albertos et al. (1997), qui ont cherché à développer un algorithme tronquant de manière déterministe ces valeurs considérées comme contaminées. Cette année, Fibbi et al. (2023) s'inspirent de cette approche pour élaborer un nouvel algorithme reposant sur une modélisation probabiliste.

Dans le cadre du cours d'*Apprentissage Non Supervisé* du Master 2 Mathématiques et Intelligence Artificielle de l'université Paris-Saclay, nous entreprenons l'étude de cet article. Nous commencerons par rappeler la méthode classique de co-clustering basée sur des modèles probabilistes ainsi que ses limites. Ensuite, nous introduirons les améliorations apportées par l'article, mettant en lumière les nouvelles propriétés qui en découlent. Enfin, nous appliquerons ce nouvel algorithme à deux jeux de données identiques à ceux présentés dans l'article afin de vérifier son intérêt et tester ses limites.

2 Co-clustering

Le principe du co-clustering est une extension directe à celui du clustering et est particulièrement utilisé dans le cas d'un nombre important de variables. En plus de partitionner les individus d'une matrice d'observations $X = (x_{ij})$ à n lignes et d colonnes, un clustering supplémentaire sur les colonnes est effectué. L'objectif est de liéer des variables latentes à de blocs d'observations homogènes mais dissemblables entre eux. Il existe de nombreuses méthodes de co-clustering, qu'elles soient déterministes (basées sur la reconstruction, Govaert (1995)) ou basées sur une modélisation probabiliste (*model-based*, Govaert and Nadif (2003)). L'article adopte celle des blocs latents (*latent block models*, LBMs).

2.1 Latent Block Models

2.1.1 Définition

Plusieurs hypothèses sont fait sur ce modèle :

- **$\mathbf{H_1}$: produit cartésien** Les blocs sont le résultat du produit cartésien d'une partition (latente) des lignes en g clusters-ligne par une partition (latente) des colonnes en m clusters-colonne, excluant les cas de chevauchement ou de non alignement de blocs.

  On introduit ainsi $Z = (z_{ik})_{i=1,\dots,n,k=1,\dots,g}$, matrice de classification des lignes où $z_{ik} = 1$ si la ligne i appartient au bloc-ligne k et $z_{ik} = 0$ sinon ; $W = (w_{jl})_{j=1,\dots,d,l=1,\dots,g}$, matrice de classification des colonnes où $w_{jl} = 1$ si la colonne j appartient au bloc-colonne l et $w_{jl} = 0$ sinon ;

- **$\mathbf{H_2}$: indépendance des labels** Les variables latentes (z_{ik}) et (w_{jl}) sont indépendantes. Les labels-lignes z_i sont i.i.d. de loi multinomiale :

$$z_i \sim \mathcal{M}(1, \pi = (\pi_1, ..., \pi_g))$$

avec $\pi_k = \mathbb{P}(z_{ik} = 1) = \mathbb{P}(z_i = k)$ pour $k = 1, .., g$ et $i = 1, .., n$.
Les labels-colonnes (w_j) sont i.i.d. de loi multinomiale :

$$w_j \sim \mathcal{M}(1, \rho = (\rho_1, ..., \rho_m))$$

avec $\rho_l = \mathbb{P}(w_{jl} = 1) = \mathbb{P}(w_j = l)$ pour $l = 1, ..., m$ et $j = 1, ..., d$;

- **$\mathbf{H_3}$: indépendance conditionnelle** Les observations (x_{ij}), conditionnellement aux labels $((z_{ik}),(w_{jl}))$, sont des variables aléatoires indépendantes dont la loi appartient à une famille paramétrique $\mathcal{F}$, i.e la densité conditionnelle de x_{ij} dépend uniquement de son bloc d'appartenance (k, l) :

$$x_{ij}|z_{ik} = 1, w_{jl} = 1 \sim f(x_{ij}|z_i w_j) = f(x_{ij}|\lambda_{k,l})$$

avec $\lambda_{k,l}$ sont des paramètres spécifiques au bloc (k, l) et f choisie suivant le type de données manipulées ;

Sous ces hypothèses, la vraisemblance s'écrit

$$L(\theta = (\pi, \rho, \lambda)|X) = \sum_{(z,w)\in\mathcal{Z}\times\mathcal{W}} \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} f(x_{ij}|\lambda_{kl})^{z_{ik}w_{jl}} \tag{1}$$

où $\mathcal{Z} \times \mathcal{W}$ représente l'ensemble de toutes les partitions croisées de $n \times d$ cellules en $g \times m$ blocs.

2.1.2 Estimation

Au vu de la complexité de l'équation, il est privilégié d'estimer le maximum de vraisemblance numériquement avec, par exemple, l'algorithme Expectation-Maximisation (EM Dempster et al. (1977)). Malheureusement, l'expression (1) ne peut se factoriser à cause de la dépendance complexe qu'induit la double structure $((z_{ik}), (w_{jl}))$ sur les observations (x_{ij}), son logarithme nécessite la somme de $g^n m^d$ termes, ce qui n'est pas réalisable en temps raisonnable. C'est pourquoi, plusieurs aménagements de cette algorithme existent. Comme dans le papier, nous allons nous concentrer sur celui du Block Classification Expectation-Maximisation (Block CEM, Govaert and Nadif (2003, 2005)), en notant $\delta_x(y)$ la fonction de Kronecker de y au point x. Il a pour particularité de maximiser l'équation (2) au lieu de la (1).

$$L_C(\theta = (\pi, \rho, \lambda), Z, W|X) = \prod_{i,k} \pi_k^{z_{ik}} \prod_{j,l} \rho_l^{w_{jl}} \prod_{i,j,k,l} f(x_{ij}|\lambda_{kl})^{z_{ik}w_{jl}} \tag{2}$$

Algorithme 1 : Block CEM

initialisation : $\hat{Z}, \hat{W}, \hat{\theta}$
tant que *n'a pas convergé* **faire**
  $\forall i, k$ calculer $s_{i,k} \propto \mathbb{P}\left(Z_{i,k} = 1 | X, \hat{W}, \hat{\theta}\right)$ ▷ étape CE sur les lignes
  $\forall i, k^*(i) \leftarrow \underset{k}{argmax} \;\; s_{i,k}, \hat{z}_{ik} = \delta_{k^*(i)}(k)$

  $\forall j, l$ calculer $t_{j,l} \propto \mathbb{P}\left(W_{j,l} = 1 | X, \hat{Z}, \hat{\theta}\right)$ ▷ étape CE sur les colonnes
  $\forall j, l^*(j) \leftarrow \underset{l}{argmax} \;\; t_{j,l}, \hat{w}_{jl} = \delta_{l^*(j)}(l)$

  mettre à jour $\hat{\theta}$ ▷ étape M
retourner $\hat{Z}, \hat{W}, \hat{\theta}$;

2.2 Principal problème : la robustesse du modèle

En générale, les modèles co-clustering ne sont pas robustes, i.e. leur capacité à ne pas être perturbé par une modification dans une petite partie des données, y compris par l'introduction de données aberrantes, est faible. Par exemple, Farcomeni (2009) met en évidence cela avec le point de rupture de la matrice centroïde M (définit comme la matrice dont les éléments sont les moyennes des co-cluster), donnée par :

$$\varepsilon_{nd}^{cell}(M, X) = \frac{1}{nd} \min \left\{ o : \sup_{X_0} ||M(X) - M(X_0)|| = \infty \right\} \tag{3}$$

où o est le nombre de données aberrantes introduit dans les données X, formant alors la matrice X_0.

Dans le cas de LBMs Gaussien avec données continues, nous montrons d'après (A.1) que $\varepsilon_{nd}^{cell}(M, X) = \frac{1}{nd}$. Il est possible d'étendre ce résultat à la plupart des LBMs. De ce fait, dans le cas d'un grand nombre de données et donc d'une existence presque sûre de données aberrantes (qu'on va appeler contaminées dans la suite), un problème de robustesse du modèle se pose.

3 Amélioration

3.1 Modèle

Pour intégrer la modélisation de la contamination dans notre problème de LBM, nous nous appuyons sur une généralisation de la formule tirée de García-Escudero et al. (2008). Cette approche avait été conçue pour effectuer un regroupement avec une méthode de rognage, visant à résoudre les problèmes liés aux données contaminées.

Le concept du rognage des données à chaque étape consiste à ne pas considérer les données ayant une faible probabilité d'appartenir à la classe qui leur a été assignée. Cela permet d'écarter les données qui correspondent le moins à la distribution que suivent les données. Dans notre cas, réalisant un co-regroupement, nous introduisons deux seuils d'élagage, un pour les clusters de lignes et l'autre pour les clusters de colonnes. La formule de modélisation est présentée dans l'équation (4), exprimant la vraisemblance du modèle.

$$\mathcal{L}(\theta|X) = \left[\sum_{(Z,W)\in\mathcal{Z}\times\mathcal{W}} \prod_{i\in S,k} \pi_k^{z_{ik}} \prod_{j\in T,l} \rho_l^{w_{jl}} \prod_{i\in S,j\in T,k,l} f(x_{ij}|\lambda_{kl})^{z_{ik}w_{jl}} \right] \left[\prod_{(i,j)\notin S\times T} g_{ij}(x_{ij}) \right], \quad (4)$$

où S et T désignent les ensembles des individus et variables non contaminés. La principale différence avec la formule usuelle réside dans l'ajout de la fonction de densité (inconnue) g_{ij} pour les éléments contaminés.

L'objectif demeure de maximiser la vraisemblance nouvellement obtenue selon S, T et les paramètres du modèle :

$$\max_{S,T} \max_{\theta\in\Theta} \mathcal{L}(\theta|X), \quad (5)$$

où $S \subseteq \{1, ..., n\}$, $T \subseteq \{1, ..., d\}$, et $\text{Card}(S) = n - \lceil \alpha_1 n \rceil$, et $\text{Card}(T) = d - \lceil \alpha_2 d \rceil$. Remarquons que lorsque $\alpha_1 = \alpha_2 = 0$, l'expression retrouve la forme du LBM classique. Néanmoins, ce problème est intractable, et par conséquent, nous envisageons une méthode qui permet, au moins empiriquement, de trouver une solution.

3.2 Estimation

Dans cette section, nous présentons une version de l'algorithme EM qui permet de réaliser un co-clustering de nos données en intégrant l'étape de rognage précédemment introduite. Nous cherchons à maximiser la vraisemblance tronquée, comme présenté dans l'équation (6).

$$\max_{S,T} \max_{Z,W,\theta} \prod_{i\in S,k} \pi_k^{z_{ik}} \prod_{j\in T,l} \rho_l^{w_{jl}} \prod_{i\in S,j\in T,k,l} f(x_{ij}|\lambda_{kl})^{z_{ik}w_{jl}} \quad (6)$$

Le nouveau modèle utilise une version modifiée de l'algorithme BCEM, incorporant une étape d'élagage. Comme l'algorithme reste inchangé, à l'exception de cette étape, nous nous concentrerons uniquement sur sa description.

Pour clarifier, focalisons-nous sur l'élagage des lignes. On estime la probabilité qu'une observation i appartienne à la classe k avec l'équation :

$$s_{ik} = \pi_k \prod_{j,l} f(x_{ij}|\lambda_{kl})^{w_{jl}}.$$

Cette quantité est proportionnelle à $P(Z_{ik} = 1|X, W, \theta)$ (à une constante de normalisation près). Ainsi, pour chaque observation, nous déterminons $k^*(i)$ comme étant l'indice maximisant s_{ik}, avec $s_{i,k^*(i)}$ comme valeur associée. Chaque observation est provisoirement assignée à la classe $k^*(i)$.

Dans l'étape de rognage des lignes, un seuil α est introduit comme pourcentage de données potentiellement contaminées. Il permet de déterminer un quantile $\lceil n\alpha \rceil$. Les valeurs dont la probabilité d'appartenance à la classe provisoire est inférieure au $\lceil n\alpha \rceil$-quantile sont exclues du calcul des nouveaux paramètres du modèle. Les autres valeurs conservent leur assignation provisoire de classe.

Cette procédure est également appliquée aux colonnes, comme synthétisé dans l'algorithme 2.

Algorithme 2 : Block CEM rogné

initialisation : $\hat{Z}$, $\hat{W}$, $\hat{\theta}$;
tant que *n'a pas convergé* **faire**
  $\forall$ i,k calculer $\log \hat{s_{ik}}$; ▷ étape CE et rognage des lignes
  $\forall$ i,$k^*(i) \leftarrow arg\max_k \log \hat{s_{i,k}}$, $\hat{s^*_{i,k}} \leftarrow \hat{s_{i,k^*(i)}}$;
  pour *i = 1,...,n* **faire**
    si $\hat{s^*_i} > \hat{s^*_{\lceil \alpha_1 n \rceil}}$ **alors**
      $\hat{z_{i,k}} = \delta_{k^*(i)}(k)$;
    sinon
      $\hat{z_{i,k}} = 0 \; \forall k$; ▷ exclut les lignes qualifiées d'outliers
  $\forall$ j,l calculer $\log \hat{t_{j,l}}$; ▷ étape CE et rognage des colomnes
  $\forall$ j, $l^*(j) \leftarrow arg\max_l \hat{t_{j,l}}$, $\hat{t^*_j} \leftarrow \hat{t_{j,l^*}}(j)$;
  pour *j = 1,...,d* **faire**
    si $\hat{t^*_j} > \hat{t^*_{\lceil \alpha_2 d \rceil}}$ **alors**
      $\hat{w}_{j,l} = \delta_{l^*(j)}(l)$
    sinon
      $\hat{w}_{j,l} = 0 \; \forall l$ ▷ exclut les colomnes qualifiées d'outliers
  mettre à jour $\hat{\theta}$;
retourner $\hat{Z}, \hat{W}, \hat{\theta}$;

3.3 Propriétés

3.3.1 Convergence

Plusieurs propriétés découlent de l'algorithme 2 (éléments de preuves en annexes (B.1.1)).

Propriété 1. *Chaque itération de l'algorithme 2 ne diminue pas la vraisemblance de la classification du modèle.*

Corollaire 1.1. *S'il existe une borne supérieur de* $\log L_C$ *(2), alors l'algorithme 2 converge en un temps fini.*

Propriété 2. *Si le bloc de densité* f *est discret, alors* L_C *est borné supérieurement uniformément à* X *pour chaque* θ *et donc par le corollaire (1.1) l'algorithme 2 converge en un temps fini.*

Propriété 3. *Si le bloc de densité* f *est normale et que l'on considère que*

$$\frac{\sigma_{max}}{\sigma_{min}} \leq c$$

où $c > 0$, $\sigma_{max} = \max_{k,l} \sigma_{k,l}$ *et* $\sigma_{\min} = \min_{k,l} \sigma_{k,l}$ *et aussi que*

$$Card(\sup \mathbb{P}_{nd}) > gm + \lceil n\alpha_1 \rceil + \lceil d\alpha_2 \rceil - \lceil n\alpha_1 \rceil \lceil d\alpha_2 \rceil$$

(i.e. la mesure empirique de $\mathbb{P}_{nd}$ *ne se concentre pas sur gm points ou moins), alors* L_C *a une borne supérieur uniformément à* X *pour chaque* θ*, et donc par le corollaire (1.1) l'algorithme 2 converge en un temps fini.*

3.3.2 Existence

Lorsqu'on restreint le domaine, on peut garantir l'existence d'une solution au problème étudié.

Propriété 4. *Si l'on se trouve dans l'un des cas suivants :*

(i) La distribution de f suit une loi de Poisson.

(ii) La distribution de f suit une loi normale et les conditions (1.1) sont satisfaites.

Alors, il existe θ^* solution de (6).

3.3.3 Robustesse

En ce qui concerne la robustesse de la méthode proposée, il convient de noter que les mêmes résultats sur le point de rupture de l'échantillon fini que dans Farcomeni (2009) s'appliquent dans notre cas. Il est alors possible de montrer que :

$$\varepsilon_{nd}^{cell}(M, X) \leq \frac{\lceil n\alpha_1 \rceil + \lceil d\alpha_2 \rceil + 1}{nd}$$

Dans le cas de données bien séparées, il semble à travers des études empiriques que l'inégalité tend à devenir une égalité. En ce sens, l'algorithme 2 est robuste et permet le co-clustering de données dont certaines sont contaminées.

3.3.4 Complexité

Dans cette section, nous effectuons une analyse de la complexité afin d'estimer le coût de l'algorithme.

Le calcul des probabilités a posteriori des classes nécessite $O(ndgm)$ opérations, tandis que l'attribution de chaque ligne à sa classe correspondante en utilisant le principe MAP requiert $O(ng)$ opérations. On réalise un élagage avec un coût de $O(n)$ pour trouver la statistique d'ordre α_1n-ième de s^*. La classification et l'élagage des colonnes ont une complexité moyenne de $O(ndgm)$, avec, au pire des cas, une complexité de $O(ndgm + m^2)$.

La formulation précise de l'étape M dépend de la distribution de bloc choisie, mais le coût asymptotique demeure $O(ndgm)$.

En résumé, la complexité temporelle totale d'une itération est $O(ndgm)$, ou au pire $O(ndgm + n^2 + d^2)$, bien que ce soit peu probable. Si n et d augmentent linéairement avec l'une des autres variables, le coût asymptotique est réduit à $O(ndmg)$.

4 Applications

Aucune implémentation de l'article n'est disponible en accès libre sur internet. Par conséquent, nous avons entrepris la reproduction de l'algorithme, en particulier dans le cas gaussien, que vous pouvez consulter ici : https://github.com/Biechy/Critique-Article-Fibbi-2023. Cependant, malgré nos efforts, il semble que cette implémentation ne soit pas fonctionnelle pour des raisons qui nous échappent.

Dans cette section, nous détaillerons la méthodologie employée pour reproduire l'étude présentée dans l'article. Par la suite, nous exposerons les résultats obtenus par Fibbi et al. (2023). Enfin, nous aborderons une idée d'amélioration que nous avons tenté d'implémenter.

4.1 Méthodologie

4.1.1 Jeux de données

Le premier jeu de données se penche sur le métabolisme du radiostrontium chez les poissons, un élément qui, s'il est radioactif, peut influencer la formation des cellules sanguines chez de nombreux poissons. Au fil des dernières décennies, il a été clairement démontré que plusieurs produits de fission présentent des risques potentiels pour la santé publique. Les données "Poissons d'Amiard" proviennent de Lebart (1969). Vingt-quatre mulets sont répartis dans trois aquariums radio-contaminés de manière identique, correspondant à différentes durées de contact avec le polluant radioactif : le premier contient les poissons numérotés de 1 à 8 ; le second contient les poissons numérotés de 9 à 17 ; le troisième contient les poissons numérotés de 18 à 24 (le poisson 17 est décédé en cours d'expérience). Les variables synthétisent des données de radioactivité (yeux, branchies, etc.) et des caractéristiques de taille (poids, longueur, etc.).

Le deuxième jeu de données concerne la macroéconomie. Il comprend sept indicateurs macroéconomiques relatifs à huit pays dont sept du G7, à savoir la France (FRA), l'Allemagne (GER), le Royaume-Uni (GBR), l'Italie (ITA), les États-Unis (USA), le Japon (JPA) et le Canada (CAN), avec l'ajout de l'Espagne (SPA). Les sept variables incluent le produit intérieur brut (PIB), l'inflation (INF), le ratio déficit budgétaire-PIB (DEF), le ratio dette publique-PIB (DEB), le taux d'intérêt à long terme (INT), le ratio balance commerciale-PIB (TRB) et le taux de chômage (UNE). Bien que de taille modeste, ce jeu de données du G7 constitue un exemple intéressant. Contrairement à l'exemple des données sur les poissons d'Amiard, il démontre comment différentes hypothèses de modélisation peuvent conduire à l'identification de valeurs aberrantes différentes. Cette diversité souligne l'importance des choix méthodologiques dans l'analyse des données macroéconomiques et met en lumière la nécessité d'une approche critique lors de l'identification des valeurs aberrantes dans ce contexte spécifique.

4.1.2 Critères d'évaluation

Il existe plusieurs moyens d'évaluer la qualité d'un co-clustering. Le choix de l'article est de s'intéresser à trois critères différents.

Pour évaluer la qualité de la classification, nous avons adopté l'indice Co-clustering Adjusted Rand (CARI), un indice proposé par Robert et al. (2021) qui varie entre -1 et 1 et étend l'ARI (Adjusted Rand Index) au co-clustering. Étant donné deux partitions de co-clustering (Z, W) et (Z', W') de tailles (g, m) et (g', m'), la définition de CARI est donnée par la formule suivante :

$$CARI(Z,W),(Z',W') = \frac{\sum_{rs} A_{rs} - \sum_r B_r \sum_s C_s / \binom{np}{2}}{\frac{1}{2}\left(\sum_r B_r + \sum_s C_s\right) - \left(\sum_r B_r \sum_s C_s\right) / \binom{np}{2}} \tag{7}$$

où

$$A_{rs} = \binom{n_{rs}^{ZWZ'W'}}{2}, B_r = \binom{\sum_s n_{rs}^{ZWZ'W'}}{2}, C_s = \binom{\sum_r n_{rs}^{ZWZ'W'}}{2},$$

avec $r \in [gm]$, $s \in [g'm']$, et $n_{rs}^{ZWZ'W'}$ désignant le nombre d'éléments de la matrice de données X appartenant simultanément au bloc r (associé à une paire d'indices de partition (k,l) spécifiée par (Z,W)) et au bloc s (identifié par une paire (k',l') spécifiée par (Z',W')).

Dans le cas continu, en plus du CARI, nous pouvons calculé la Somme des Carrés des Erreurs (SSE), donnée par

$$SSE = \sum_{i,j,k,l} z_{ik} w_{jl} (x_{ij} - \hat{\mu}_{kl})^2 \tag{8}$$

où i, j, k, l sont des indices correspondant aux éléments de la matrice de données, z_{ik} et w_{jl} sont des indicateurs de co-clustering, x_{ij} est l'élément de la matrice de données, et $\hat{\mu}_{kl}$ est la moyenne estimée pour le bloc de co-clustering spécifié par k et l.

À ces deux mesures de qualité d'une partition, on peut ajouter celui du calcul de la log-vraisemblance donné par l'expression (2).

4.2 Résultats

Nous présentons individuellement chaque jeux de données, détaillant les paramètres testés, les représentations de co-regroupement, ainsi que les performances évaluées selon divers critères. Les jeux de données sont normalisés pour le co-regroupement.

4.2.1 Jeu de Données Poissons d'Amiard

Le jeu de données provient d'une étude de Govaert et Nadif, qui ont opté pour un co-regroupement avec $g = 5$ et $m = 3$. Une Analyse en Composantes Principales (ACP) révèle un groupe de quatre variables comportant un comportement distinct, écarté comme des valeurs aberrantes par une ACP robuste. Cela motive l'utilisation d'une méthode robuste pour évaluer la nature potentiellement contaminée de ces variables.

L'algorithme est appliqué avec $\alpha_2 = \frac{4}{d}$, et les tailles de groupes sont fixées à $(g, m) = (5, 2)$. De plus, 500 initialisations différentes sont lancées, et la solution avec la plus grande log-vraisemblance est retenue.

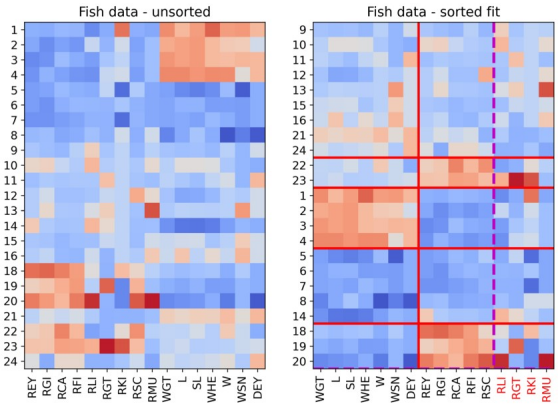

FIGURE 1 – Matrice de données avant et après le tri selon le modèle ajusté. Les couleurs plus chaudes correspondent à des valeurs plus élevées.

En partant du sommet du graphique à droite de la Figure 1, on observe un vaste regroupement de lignes avec des valeurs proches de zéro. En descendant, le deuxième et le cinquième regroupement

de lignes se caractérisent tous deux par des valeurs faibles et élevées pour les variables liées à la taille et aux niveaux de radioactivité, respectivement. Les deux regroupements de lignes diffèrent en ce que le cinquième présente des valeurs sensiblement plus basses dans le bloc "froid". Le troisième regroupement de lignes, quant à lui, est caractérisé par le schéma opposé. Enfin, le quatrième groupe de lignes présente des valeurs nettement basses dans les deux blocs. Les co-regroupements que nous avons identifiés sont similaires, mais pas identiques, à ceux de Govaert et Nadif, paraissant légèrement plus naturels que ces derniers.

Pour cet ensemble de données, nous pouvons observer que le rognage a rendu possible l'identification et l'exclusion des colonnes anormales qui ne contribuaient pas au co-regroupement, améliorant ainsi le résultat final.

4.2.2 Jeu de Données de Macro-économie

Le jeu de données de macro-économie provient de Farcomeni (2009), qui utilise une version de l'algorithme de double k-means avec une étape d'élagage à seuils.

Les paramètres utilisés dans cet article sont tels que $(g, m) = (3, 2)$, $\alpha_1 = \frac{1}{8}$, et $\alpha_2 = 0$. On souhaite seulement écarter les données aberrantes parmi les individus. Dans l'article de Farcomeni, les poids sont égaux à $(1/d)$, et les variances à une même constante arbitraire.

Avec ces paramètres, et malgré des initialisation différentes, l'algorithme de double k-means rogné écarte systématiquement l'Italie. L'algorithme TBCEM quand à lui présente deux résultats différents. Selon si l'algorithme est contraint ou non (en terme de poids) il va écarter l'Italie ou l'Espagne respectivement.

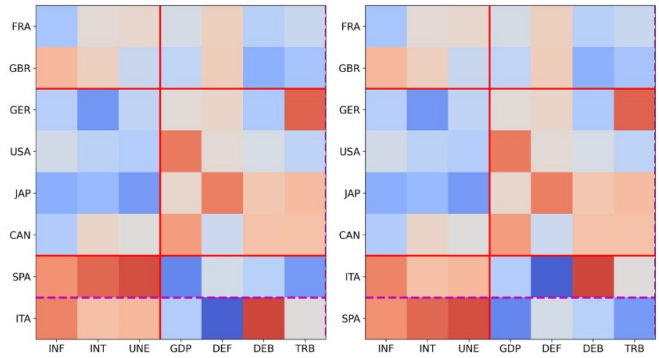

FIGURE 2 – Double k-means rogné (à gauche) et TBCEM (à droite). Les deux produisent une partition pratiquement identique, à l'exception de la dernière ligne qui exclut l'Espagne ou l'Italie.

Ce qu'on peut conclure de cette simulation est le fait que l'algorithme de TBCEM a plus de flexibilité et offre une plus grande variété de solutions. Cela met en avant l'importance du modèle considéré par l'algorithme. Cela montre aussi qu'il ne faut pas se fier uniquement à la valeur la plus grande de log-vraisemblance en étape finale pour affirmer de la meilleure qualité de co-regroupement.

4.3 Extensions

Malgré des résultats non satisfaisants de notre implémentation, nous avons observé dans le cas gaussien que le bon fonctionnement de l'algorithme dépend considérablement de l'initialisation des matrices μ et σ. Il était donc évident que le choix aléatoire de ces deux matrices pourrait entraîner

l'absence de données dans un cluster ligne ou un cluster colonnes. Cette situation résulterait en une normalisation des μ_{kl} et σ_{kl} par une division par zéro, entraînant ainsi un blocage complet de l'algorithme.

Afin de remédier à cette problématique, nous nous sommes inspirés de la définition du point de rupture de l'échantillon (3), en particulier de la matrice centroïde M. Initiants des co-clusters de manière arbitraire, nous avons postulé que la matrice $M(X)$ serait une meilleure approximation de μ par rapport à une initialisation aléatoire. Par extension, nous avons défini $V(X)$ comme la matrice contenant les variances des co-clusters, en supposant qu'elle représente une meilleure approximation de σ. Toutefois, cette initialisation seule ne suffit pas à résoudre entièrement le problème de clusters vide.

Ainsi, nous avons réitéré un nombre arbitraire de fois le processus en conservant le même co-clusters, mais en appliquant un mélange aléatoire des lignes puis des colonnes à la matrice X, créant ainsi la matrice X'. Cette étape fournit une nouvelle initialisation pour $M(X')$ et $V(X')$, et par conséquent, de nouvelles valeurs pour les éléments des matrices μ et σ. Cela dans l'espoir d'obtenir au moins une initialisation optimale permettant d'atteindre le maximum global de la vraisemblance, cette approche est implémentée au moyen de la fonction *auto_ reply* de la classe *TrimmedBlockCEM*.

5 Discussion

Cet article suit la voie de l'amélioration de la robustesse des modèles probabilistes de co-clustering, ce qui n'avait pas été pleinement exploré auparavant. Il répond convenablement aux nouvelles problématiques soulevées par l'augmentation actuelle des dimensions des jeux de données.

Cependant, les démonstrations présentées dans l'article manquent parfois de formalisme. Ceci s'explique probablement par la nature classique des propriétés déduites, lesquelles sont communes dans le contexte des extensions de l'algorithme E-M.

Aussi, nous avons trouvé que ce document manquait de rigueur. Cela découle probablement du principe sous-jacent de l'article qui consiste à étendre un algorithme de co-clustering déjà existant au cas des données contaminées. Par conséquent, les auteurs se réfèrent principalement à d'autres ouvrages et ne décrivent donc pas complètement le processus permettant sa mise en place. De plus, bien que l'accent soit mis sur les applications aux données, l'indisponibilité du code source a limité notre capacité à confirmer concrètement leurs résultats.

Informations supplémentaires

L'écriture du rapport a été traité de la manière suivante : Lucas Biéchy a rédigé l'intégralité de la partie 2 ainsi que les propriétés présentées 3.3.1 et 3.3.3. Nafissa Benali aura rédigé les parties 3.1 et 3.2 ainsi que les propriétés présentées en 3.3.2 et 3.3.4. L'intégralité des parties restantes ont été réalisées conjointement.

6 Conclusion

L'étude menée par Fibbi et al. (2023) représente une avancée significative dans le domaine des techniques de co-clustering basées sur des modèles probabilistes, offrant ainsi une robustesse accrue, conformément aux critères de robustesse définis par Farcomeni (2009). Cette amélioration se traduit par une capacité renforcée à traiter des jeux de données incluant des valeurs aberrantes.

Néanmoins, l'importante dépendance à l'initialisation constitue un défi non négligeable pour le bon fonctionnement de l'algorithme. Une piste de recherche prometteuse consisterait à explorer des

méthodes visant à atténuer cette dépendance (comme nous avons initié ou comme Keribin et al. (2015) dans le cadre de variables catégorielles), avec pour objectif de rendre l'algorithme plus accessible et applicable dans un éventail plus large de contextes.

En résumé, bien que cette étude ouvre des perspectives encourageantes en matière de co-clustering probabiliste, la question de la dépendance à l'initialisation demeure un point critique à adresser pour concrétiser pleinement le potentiel de cet algorithme. Des efforts supplémentaires dans la recherche de solutions à ce défi spécifique pourraient ainsi contribuer de manière significative à la pertinence et à l'efficacité générale de cette approche.

Références

Coretto, P. and Hennig, C. (2017). Consistency, breakdown robustness, and algorithms for robust improper maximum likelihood clustering. *Journal of Machine Learning Research*, 18(142) :1–39.

Cuesta-Albertos, J. A., Gordaliza, A., and Matrán, C. (1997). Trimmed k-means : an attempt to robustify quantizers. *The Annals of Statistics*, 25(2) :553–576.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society : series B (methodological)*, 39(1) :1–22.

Farcomeni, A. (2009). Robust double clustering : a method based on alternating concentration steps. *Journal of classification*, 26 :77–101.

Fibbi, E., Perrotta, D., Torti, F., Van Aelst, S., and Verdonck, T. (2023). Co-clustering contaminated data : a robust model-based approach. *Advances in Data Analysis and Classification*, pages 1–41.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2008). A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3).

Govaert, G. (1995). Simultaneous clustering of rows and columns. *Control and Cybernetics*, 24 :437–458.

Govaert, G. and Nadif, M. (2003). Clustering with block mixture models. *Pattern Recognition*, 36(2) :463–473.

Govaert, G. and Nadif, M. (2005). An em algorithm for the block mixture model. *IEEE Transactions on Pattern Analysis and machine intelligence*, 27(4) :643–647.

Keribin, C., Brault, V., Celeux, G., and Govaert, G. (2015). Estimation and selection for the latent block model on categorical data. *Statistics and Computing*, 25(6) :1201–1216.

Lebart, L. (1969). Introduction à l'analyse des données. *Consommation/Revue de socio-économie*, 3 :57–96.

Robert, V., Vasseur, Y., and Brault, V. (2021). Comparing high-dimensional partitions with the Co-clustering Adjusted Rand Index. *Journal of Classification*, 38 :158–186.

Annexes

A Co-clustering

A.1 Robustesse CEM : Cas Gaussien

Étant donné les matrices de partition Z et W, les estimations des moyennes des densités de blocs peuvent être obtenues analytiquement en résolvant les équations du maximum de vraisemblance correspondantes et sont données par :

$$M_{kl}(X) = \frac{\sum_{i,j} z_{ik} x_{ij} w_{jl}}{\sum_i z_{ik} \sum_j w_{jl}} \tag{9}$$

En prenant la matrice perturbée $X_0 := (\delta_{i=i',j=j'} x + \delta_{i\neq i',j\neq j'} x_{ij})_{i,j}$ avec $x \in \mathbb{R}$ et $z_{i'k'} = w_{j'l'} = 1$, on a alors :

$$(M(X) - M(X_0))_{k,l} = \frac{x_{i'j'} - x}{\sum_i z_{ik'} \sum_j w_{jl'}} \delta_{k=k',l=l'}$$

et ainsi

$$\sup_{X_0} ||M(X) - M(X_0)|| \geq \sup_{X_0} ||M(X) - M(X_0)\mathbf{e}_{l'}|| = \lim_{x\to\infty} \left| \frac{x_{i'j'} - x}{\sum_i z_{ik'} \sum_j w_{jl'}} \right| = \infty$$

or la matrice n'est perturbé qu'au point (i', j') donc $\varepsilon^{cell}_{nd}(M, X) = \frac{1}{nd}$, ce qui fait sens car (9) est simplement la moyenne arithmétique des éléments de X appartenant au bloc (k,l).

B Amélioration

B.1 Éléments de preuves

B.1.1 Convergence

Propriété 1

Une itération de l'algorithme se compose de trois étapes d'optimisation alternées : deux étapes de Classification-Expectation (sur les lignes et sur les colonnes) et une étape de Maximisation. Considérez l'étape EC sur les lignes. Puisque nous maximisons le $\log L_C$ par rapport à Z, on peut considérer le terme $\sum_{j,l} w_{jl} \log \rho_l$ comme constante $c = c(W, \rho)$ et on a alors :

$$\log L_C = \sum_{i,k} z_{ik} \left(\log \pi_k + \sum_{j,l} w_{jl} \log f(x_{ij}|\lambda_{kl}) \right) + c = \sum_{i,k} z_{ik} s_{ik} + c \tag{10}$$

Pour tout i, on a $k^*(i) = \text{argmax}_k\ s_{ik}$ et pour l'indice i correspondant au plus grand des $n - \lceil n\alpha_1 \rceil s_{ik}$ on pose $z_{ik} = \delta_{k^*(i)}(k)$. Les lignes $\lceil n\alpha_1 \rceil$ restantes ne sont affectées à aucune partition de ligne, c'est-à-dire que si i' est l'une de ces lignes, nous définissons $z_{i'k} = 0$ pour tout k. Par conséquent, à chaque étape, dans l'équation (10), les z_{ik} sont choisis de manière à maximiser la

contribution des s_{ik} sous les contraintes de classification et de troncature. Le même argument s'applique, par symétrie, à l'étape EC sur les colonnes.

Enfin, par définition de l'étape M, les paramètres θ sont estimés précisément comme l'argument max du log L_C compte tenu des partitions. Par conséquent, cette étape ne peut pas non plus diminuer la fonction objectif.

Corollaire 1.1

Notons ℓ^t la valeur du log L_C à l'itération t. Par la proposition (1), la séquence de réels $(\ell^t)_{t>0}$ est monotone, donc la limitation est une condition nécessaire et suffisante pour sa convergence. La dernière partie de l'affirmation découle du fait que le nombre de partitions possibles est fini.

Propriété 2

Nous pouvons borner uniformément en x la fonction de masse de probabilité, pour toutes valeurs de paramètres, simplement en notant que $f(x_{ij}|\lambda_{kl})$ est une probabilité et donc bornée entre 0 et 1. La majoration de f implique automatiquement que la majoration de la fonction objectif (2).

Propriété 3

Dans le cas gaussien, f n'explose que lorsque $\sigma \to 0$ et $x = \mu$. Or nos hypothèses empêchent que cela se produit et empêche une divergence de vraisemblance. Pour des raisons de simplicité de notation, nous supprimons les indices doubles et écrivons σ_k pour indiquer la k-ième composante de la version vectorisée de la matrice des variances de bloc. Supposons que $\sigma \to 0$ pour certains k. La première hypothèse implique que $\sigma \to 0$ pour tout k. Or, lorsque $\sigma \to 0$, $f(\mu|\mu, \sigma)$ diverge comme $\frac{1}{\sigma}$, alors que si $x = \mu$ on dit que $f(x|\mu, \sigma) \in o(\sigma^q)$, pour tout $q > 0$ (voir par exemple Coretto and Hennig (2017)). Grâce à la seconde hypothèse, on est assuré qu'il existe un x qui ne coïncide avec aucune des moyennes μ_k. Par conséquent (2) contient au moins un facteur qui disparaît plus rapidement que n'importe quel polynôme, et donc toute la vraisemblance converge vers 0.

Propriété 4 Tout d'abord, considérons la log-vraisemblance L_c à la distribution empirique :

$$L_c(\theta) = \sum_{ik} z_{ik} \log(\pi_k) + \sum_{jl} w_{jl} \log(\rho_l) + \sum_{ijkl} z_{ik} w_{jl} \log f(x_{ij}|\lambda_{kl}) \quad (12)$$

où $z_{ik} = z_k(x_i, \theta)$ et de même pour w_{jl} (il suffit de redéfinir les étiquettes en fonction de θ) qui effectuent l'assignation optimale selon le principe MAP, comme vu par exemple dans la sous-section 3.2. Il apparaît que

$$M := \sup_\theta L_c > -\infty \quad (13)$$

Par exemple, dans le cas normal, prenons θ tel que $\pi_k = 1/g$ pour tous les k, $\rho_l = 1/m$ pour tous les l, $\mu_{kl} = 0$ et $\sigma_{kl} = 1$ pour tous les k, l. Il est facile de voir que la même chose est vraie dans le cas de Poisson. Nous savons déjà de la Proposition 2 que $M < +\infty$ également.

Montrons maintenant que toute séquence maximisante $\{\theta_t\}_{t=0}^{\infty}$ dans l'espace des paramètres (satisfaisant les contraintes (C1) et (C2) de la Propriété 2 dans le cas normal) doit être contenue dans un ensemble compact.

Considérons $\{\theta_t\}$ tel que $\exists k$ tel que $\sigma_k \to +\infty$. Par la contrainte (C1) de la Propriété 2, il en résulte que $\sigma_k \to +\infty$ pour tous les k. Il est alors clair que toutes les densités de bloc $f(x|\mu_k, \sigma_k)$ tendent vers zéro pour n'importe quel x et μ_k, donc $L_c \to -\infty$ tous π, ρ et μ.

Dans la démonstration de la Proposition 2, nous avons déjà montré que si au contraire $\sigma_k \to 0$ pour un certain k, alors (C1) et (C2) garantissent que, de nouveau, $L_c \to -\infty$.

Considérons maintenant le cas où $|\mu_k| \to +\infty$. On voit facilement que cela implique $f(x|\mu_k, \sigma_k) \to 0$, quelles que soient les valeurs de x et les autres paramètres. Par conséquent, dans ce cas également, $L_c \to -\infty$. Le cas où un nombre arbitraire de moyennes diverge n'est pas différent.

Enfin, il reste à montrer que $L_c \to -\infty$ même lorsque l'on prend des limites sur μ et σ simultanément. Nous avons deux cas à considérer, à savoir : $|\mu_k| \to +\infty$ et $\sigma_k \to +\infty$, ou $|\mu_k| \to +\infty$ et $\sigma_k \to 0$. Remarquez que si certains σ_k divergent ou tendent vers zéro, alors, en raison de (C1), toutes les variances de blocs divergent/tendent vers zéro au même taux, donc il n'est pas important de savoir à quel indice k nous nous référons. Si plus d'une moyenne diverge, cela n'impacte pas les arguments qui suivent. Dans le premier cas, nous avons deux sous-cas, en fonction du taux de divergence relatif entre μ et σ.

a) $\mu_k/\sigma_k \to c \in \mathbb{R}$: cela implique $f(x|\mu_k, \sigma_k) \to 0$ ($f(x|\mu_k, \sigma_k)$ est $o(1/\sigma_k)$), donc $L_c \to -\infty$.
b) $|\mu_k|/\sigma_k \to +\infty$: dans ce cas, $f(x|\mu_k, \sigma_k) \to 0$ exponentiellement et donc $L_c \to -\infty$.

Lorsque $|\mu_k| \to +\infty$ et $\sigma_k \to 0$, $|\mu_k|/\sigma_k \to +\infty$ et donc nous tombons dans le cas b) ci-dessus.