



**HAL**  
open science

# Propriétés émergentes du multi-clustering bayésien non paramétrique: Application aux données images multivues

Reda Khoufache, Mohamed Djallel Dilmi, Hanene Azzag, Etienne Goffinet,  
Mustapha Lebbah

## ► To cite this version:

Reda Khoufache, Mohamed Djallel Dilmi, Hanene Azzag, Etienne Goffinet, Mustapha Lebbah. Propriétés émergentes du multi-clustering bayésien non paramétrique: Application aux données images multivues. SFC'2023 Rencontres de la Société Francophone de Classification, Jul 2023, Strasbourg, France. hal-04457655

**HAL Id: hal-04457655**

**<https://hal.science/hal-04457655>**

Submitted on 14 Feb 2024

**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.

# Propriétés émergentes du *multi-clustering* bayésien non paramétrique: Application aux données images multivues

Reda Khoufache<sup>1,2</sup>, Mohamed Djallel Dilmi<sup>2</sup>, Hanene Azzag<sup>2</sup>, Etienne Goffinet<sup>3</sup>, Mustapha Lebbah<sup>1,2</sup>

<sup>1</sup> DAVID, Université de Versailles, Université Paris-Saclay

<sup>2</sup> LIPN (CNRS UMR 7030), Université Sorbonne Paris Nord

<sup>3</sup> Technology Innovation Institute, TII

## Résumé

*Ce papier a déjà été publié dans [16]. Dans cet article, nous avons développé un nouveau framework qui traite le problème de clustering d'images multivues collectées dans un magasin intelligent. Le framework combine les transformers pré-entraînés avec un nouveau modèle de multi-clustering bayésien non paramétrique. Dans ce travail, nous proposons une approche MCMC afin d'inférer la partition colonne et les partitions lignes. Notre modèle infère de multiples partitions et permet d'estimer automatiquement le nombre de clusters. Notre méthode fournit des blocs homogènes sur un jeu de données d'images multivues. Elle souligne d'une part la puissance des vision transformers pré-entraînés combinés au multi-clustering et d'autre part l'utilité de la modélisation bayésienne non paramétrique qui automatise l'estimation du nombre de clusters. Le code source du framework complet est disponible dans ce lien github : <https://github.com/unsupervise/acomi>*

## Mots-clés

*Modèles bayésiens non paramétriques, sélection de modèles, multi-clustering, clustering d'images multivues, vision transformers*

## Abstract

*This paper has already been published in [16]. In this article, we have developed a new framework to address the multi-view image clustering of images collected from a smart store. The proposed framework combines a pre-trained Vision Transformer with a new bayesian nonparametric multiple clustering model. In this work, we propose an MCMC-based inference approach to learn the column-partition and the row-partitions. This method infers multiple clustering solutions and allows to find the number of clusters automatically. Our method provides interesting results on a multi-view image dataset and emphasizes, on the one hand, the power of pre-trained Vision Transformers combined with the multiple clustering algorithm, on the other hand, the usefulness of the bayesian nonparametric modeling, which automatically performs a model selection. The complete code of our framework is available at <https://github.com/unsupervise/acomi>*

## Keywords

*Bayesian nonparametric, model selection, multiple clustering, multi-view image clustering, vision transformers*

## 1 Introduction

L'intelligence artificielle a un impact considérable sur le secteur du *retail*; les entreprises de grande distribution investissent de plus en plus dans des solutions de vision par ordinateur afin d'améliorer la gestion de leurs magasins et mieux satisfaire les attentes de leurs clients.

La classification de produits dans un magasin autonome est une tâche délicate, ceci est dû à la quantité massive de produits ainsi que les variables qui décrivent chaque produit. Un des enjeux majeurs est l'amélioration de la classification de produits dans un magasin intelligent. Le développement d'algorithmes d'apprentissage passant à l'échelle et qui permet de traiter simultanément les données et les variables est indispensable pour construire des modèles prédictifs. L'apprentissage non supervisé multivues où chaque vue représente une variable fournit des résultats supérieurs à ceux à vue unique, et ce, en raison de l'utilisation des informations complémentaires provenant de différents espaces de représentation.

Dans le cas multivarié, le *clustering* infère uniquement une partition ligne, tandis que le *multi-clustering* infère une partition en colonne (partition de variables ou vues), et une partition ligne pour chaque vue.

Afin de traiter le problème du *multi-clustering*, une approche probabiliste a été introduite par [10]; ce modèle suppose la présence d'une structure avec de multiples blocs où chaque cellule est une observation univariée continue. Les modèles paramétriques supposent que le vrai nombre de blocs est connu a priori. Cette hypothèse n'est généralement pas satisfaite en pratique. Par conséquent, une étape intermédiaire de sélection de modèle est nécessaire afin d'estimer ce nombre. Cette sélection est généralement réalisée soit avec une recherche exhaustive, soit avec recherche gloutonne. Cependant, le nombre de modèles possibles est très grand même avec un petit nombre de blocs [12]. Ainsi, une recherche exhaustive n'est plus envisageable. Des stratégies heuristiques sont alors considérées afin d'explorer une partie de l'espace des solutions au coût d'hypothèses supplémentaires sur la structure du modèle. Dans [13], les

auteurs ont proposé un modèle de *multi-clustering* paramétrique fonctionnel désigné pour traiter des séries temporelles multivariées ; la sélection de modèle est réalisée en utilisant le critère *ICL (Integrated Classification Likelihood)*.

La modélisation bayésienne non paramétrique permet d’estimer le nombre de composantes durant l’inférence en mettant une distribution a priori sur les paramètres du modèle. Le processus de dirichlet pour les modèles de mélange (*DPMM*) [1] est un exemple de modèle bayésien non paramétrique qui suppose la présence d’un nombre infini de composantes latentes. Deux représentations sont souvent utilisées afin de mieux interpréter le processus de dirichlet (*DP*) : processus du restaurant chinois (*CRP*) [2] et le *Stick-Breaking (SB)* [24]. Ces deux représentations sont liées, l’une peut découler de l’autre [22]. Dans [21], les auteurs ont introduit un modèle de catégorisation croisée, et [14] ont proposé un modèle de *multi-clustering* bayésien non paramétrique. Ces deux travaux partagent la même définition du modèle, qui met d’abord un a priori sur la partition colonne, qui estime automatiquement le nombre de *clusters* colonnes, ensuite met un a priori indépendant sur les proportions de chaque partition ligne. Par ailleurs, dans [14], les auteurs proposent une inférence variationnelle afin d’estimer les paramètres du modèle, tandis que [21] utilise une méthode *MCMC*.

Dans ce travail, nous proposons une nouvelle approche pour le *clustering* d’images multivues basée sur le *clustering* bayésien non paramétrique. Le *clustering* d’images est une tâche complexe car les images sont présentées sous forme de données de grande dimension, ce qui affecte la performance des algorithmes du *clustering*. Des méthodes d’apprentissage de représentation tels que SIFT [19], HOG [7] sont utilisées afin d’en extraire des caractéristiques. Celles-ci sont ensuite utilisées pour discriminer les images. Les approches récentes combinent les méthodes du *clustering* avec les réseaux de neurones profonds (*deep clustering*), ce qui a permis d’atteindre des performances remarquables. Ces méthodes peuvent être divisées en deux catégories : la première catégorie réalise d’abord l’extraction des caractéristiques en utilisant des réseaux de neurones à convolution tels que *deep belief network* [17], et les auto-encodeurs [25]. Ensuite un algorithme de *clustering* est appliqué sur les caractéristiques extraites. La deuxième catégorie prend en considération cette séparation, ainsi des travaux récents tels que [18, 27] proposent des méthodes basées sur un outil unifié qui réalise de façon jointe l’apprentissage profond des représentations et les *clusters* d’images.

## 2 Contexte

### 2.1 Vision transformers

Les *transformers* sont des réseaux de neurones qui utilisent des mécanismes d’attention [26], leurs permettant de se focaliser sur certaines régions de l’entrée, et ainsi d’extraire des caractéristiques intrinsèques et significatives. Les *transformers* ont d’abord été introduits pour le traitement naturel du langage (*NLP*) [26], ces modèles ont eu un énorme suc-

ès avec leur capacité de représentation remarquable.

Récemment, les *transformers* ont été adaptés afin de traiter des tâches de vision par ordinateur. Le modèle *ViT (Vision Transformer)* [9] applique une architecture du *transformer* sur des séquences de petits patches d’images afin de les classifier. Le modèle a atteint des performances exceptionnelles sur de nombreux jeux de données tel que *Imagenet* [8]. Une étude dédiée à l’apprentissage par transfert de ces modèles a été réalisée dans [20]. En effet, l’apprentissage par transfert [3] est un outil important qui permet de gagner en temps et en ressources nécessaires au réentraînement complet de ces réseaux de neurones. Ainsi, au lieu de réentraîner le *ViT* entièrement, nous considérons un réseau déjà pré-entraîné sur *ImageNet* et l’utilisons pour l’extraction des caractéristiques.

## 2.2 Multi-clustering

### 2.2.1 Définition du modèle

Notons  $X \in \mathbb{R}^{n \times p \times d}$  l’espace latent obtenu après une certaine transformation du jeu de données d’images multivues, où  $n$  est le nombre d’observations,  $p$  est le nombre de variables  $d$  est la dimension de l’espace de représentation. Soit  $H$  le nombre de *clusters* de variables,  $v$  la partition de variables,  $Z$  une matrice indicatrice  $n \times H$ , des partitions lignes. Le modèle est défini comme suit :

$$\begin{aligned} x_{i,j} &| \{v_j = h, z_i^h = k, \theta_k^h\} \sim \mathcal{N}(\theta_k^h), \\ \theta_k^h &\sim G_0, v_j \sim \text{Mult}(\eta), z_i^h \sim \text{Mult}(\pi_h), \\ \eta_j(\mathbf{r}) &= r_j \prod_{j'=1}^{j-1} (1 - r_{j'}), r_j \stackrel{\text{i.i.d.}}{\sim} \text{Beta}(1, \gamma), \\ \pi_j^h(\mathbf{t}^h) &= t_j^h \prod_{j'=1}^{j-1} (1 - t_{j'}^h), t_j^h \stackrel{\text{i.i.d.}}{\sim} \text{Beta}(1, \alpha_h), \\ \gamma &\sim \text{Gamma}(a_\gamma, b_\gamma), \alpha_h \sim \text{Gamma}(a_\alpha, b_\alpha). \end{aligned}$$

Où  $x_{i,j}$  est l’élément de la ligne  $i$  et la colonne  $j$ ,  $v_j$ , la variable d’appartenance de la colonne  $j$ ,  $z_i^h$  la variable d’appartenance de la ligne  $i$  dans le cluster de variables (vue)  $h$ . Les proportions de la partition de variables  $\eta$  et des partitions lignes  $\pi_h$  suivent le processus *Stick-Breaking* [24]. Chaque paramètre de concentration suit une loi a priori Gamma, ce qui permet de les adapter à chaque bloc qui peut contenir un nombre différent de composantes. La densité associée à chaque bloc est une gaussienne multivariée de dimension  $d$  et de paramètre  $\theta_k^h = (\mu_k^h, \Sigma_k^h)$ .  $G_0$  est la loi à priori conjuguée Normale-Wishart-Inverse (*NIW*), l’ensemble des hyperparamètres est noté  $\chi$ .

Le modèle défini dans ce papier est une extension au cas bayésien non paramétrique du *multi-clustering* paramétrique introduit dans [13]. Il est important de noter que dans ce travail, les densités composantes associées à chaque bloc modélisent les cellules et non pas les lignes. En d’autres termes, les cellules appartenant au même bloc suivent indépendamment la même distribution multivariée.

### 2.2.2 Inférence

L'inférence est réalisée en utilisant une variante de l'algorithme Gibbs et alterne entre trois étapes principales : en partant d'une partition initiale, nous commençons par mettre à jour la partition des variables sachant la partition des lignes. Ceci permet d'estimer les différentes vues. Ensuite, pour chaque vue, nous mettons à jour la partition des lignes sachant la partition de variables. Finalement, nous mettons à jour les paramètres de concentration. Dans ce papier, nous détaillons uniquement l'inférence de la partition de variables, les deux autres étapes peuvent être retrouvées dans [16].

**Inférence de la partition de variables :** La partition des variables  $v$  est mise à jour sachant les partitions lignes  $Z$ . Chaque variable d'appartenance  $(v_j)_p$  est échantillonnée selon  $p(v_j | \mathbf{v}_{-j}, Z, \mathbf{x}_{:,j}, \chi) \propto$  :

$$\begin{cases} \frac{p_h}{p-1+\gamma} p(\mathbf{x}_{:,j} | \mathbf{z}^h, \chi), & \text{cluster existant } h, \\ \frac{\gamma}{p-1+\gamma} p(\mathbf{x}_{:,j} | \chi), & \text{nouveau cluster,} \end{cases} \quad (1)$$

Avec  $p_h$  le cardinal du *cluster* en colonne  $h$  et  $\mathbf{v}_{-j} = \{v_i : i \neq j\}$ . Dans l'équation 1, en vertu du choix approprié de la loi a priori conjuguée  $G_0$ , la distribution prédictive jointe a priori  $p(\mathbf{x}_{:,j} | \mathbf{z}^h, \chi)$  se réduit à un produit de densités de Student multivariées [12]. Dans l'équation 2,  $p(\mathbf{x}_{:,j} | \chi)$  est estimé en appliquant un *DPM* par variable avant l'inférence du *multi-clustering*, ce qui fournit également la partition ligne optimale  $\hat{\mathbf{z}}^j$  associée à la variable  $j$ .

### 2.2.3 Implémentation

Nous initialisons l'algorithme comme suit :  $\mu_0$  est le vecteur nul de dimension  $d$ . D'après [23], la matrice de précision  $\Psi_0$  obtenue avec l'estimateur du maximum de vraisemblance est une bonne initialisation,  $\kappa_0$  et  $\nu_0$  sont initialisés avec leur plus faible valeur afin qu'ils soient le moins informatifs possible. La partition initiale est la partition à un seul *cluster*.

### 2.2.4 Inférer la partition finale

L'algorithme de Gibbs produit une chaîne de Markov de partitions échantillonnées à partir de la distribution a posteriori approximée. Ces partitions échantillonnées doivent être agrégées à partir d'un certain nombre d'itérations. Dans notre cas, nous gardons la dernière partition échantillonnée.

### 2.2.5 La complexité algorithmique

Avant l'inférence principale, un *DPM* est lancé sur chaque variable  $\mathbf{x}_{:,j}$  pour estimer  $p(\mathbf{x}_{:,j} | \chi)$ . Chaque *DPM* a une complexité de  $O(Mnd^2)$ . Comme l'inférence est appliquée sur chaque variable, la complexité est  $O(Mnpd^2)$ . Durant l'inférence principale, la mise à jour des *clusters* en colonne a une complexité de  $O(p\bar{H}(nd^2 + \bar{K}d^3))$  avec  $\bar{H}$  et  $\bar{K}$  les nombres maximaux des *clusters* en colonne et *clusters* lignes respectivement. La mise à jour des *clusters* lignes a une complexité de  $O(npd^2 + (n+p)\bar{K}d^3)$ . Finalement, le processus complet d'inférence a une complexité de  $O(M(npd^2 + (n+p)\bar{K}d^3))$ .

## 3 Le framework proposé

### 1. Préparation du jeu de données

Le jeu de données d'images est réarrangé sous forme d'un tableau bi-dimensionnel où les lignes représentent les observations, les colonnes sont les différentes variables et chaque cellule est une image qui décrit l'observation correspondante.

### 2. Extraction des caractéristiques et réduction de dimension

Dans la première étape, nous proposons d'utiliser un *Vision Transformers* pré-entraîné comme extracteur de caractéristiques. Nous traitons chaque image avec le *ViT* et gardons la dernière couche avant la prédiction des classes comme vecteur de représentation latent. Nous appliquons ensuite une analyse en composantes principales (*PCA*) [15] afin de réduire la dimension de l'espace de représentation en raison du fléau de la dimension et de la complexité algorithmique. L'espace latent obtenu est un *tensor* de dimension  $n \times p \times d_r$ , où  $d_r$  est le nombre de composantes conservées après l'application de la *PCA*.

### 3. Multi clustering

Le jeu de données obtenu après la transformation est ensuite passé en entrée de l'algorithme du *multi-clustering* détaillé dans la section 2.2, ce qui nous fournit de multiples partitions. Le *framework* complet est illustré dans la figure 1.

## 4 Expériences

### 4.1 Jeu de données

Aloi<sup>1</sup> [11] est un jeu de données d'images multi-vues contenant 1000 objets, chacun est décrit par 108 images différentes, celles-ci sont obtenues en variant trois critères : l'angle de prise de l'image, l'intensité lumineuse et la position de la caméra. Ainsi, 72 variables notées  $\{r_0, r_5, \dots, r_{355}\}$  représentent des prises du même objet pivoté par incrément de  $5^\circ$  couvrant les  $360^\circ$ ; 12 variables notées  $\{i_{110}, i_{120}, \dots, i_{250}\}$  représentent des prises avec différentes intensités lumineuses et 24 variables  $\{l1_c1, \dots, l8_c3\}$  qui représentent des prises de 3 différentes caméras avec 8 positions différentes de la source lumineuse. Le jeu de données contient des annotations des caractéristiques de ces produits tels que la matière (*Material*) (vêtement, bois, fruit, etc). Nous avons choisi ce jeu de données, car celui-ci contient des objets similaires à ceux qu'on peut retrouver dans un magasin, de plus, comme dans un magasin intelligent, ces objets sont capturés sous différents angles et sous différentes conditions lumineuses (intensité et angles). En notre connaissance, le jeu de données Aloi, est le seul jeu de données images multi-vues public satisfaisant ces caractéristiques.

1. <https://aloi.science.uva.nl/>

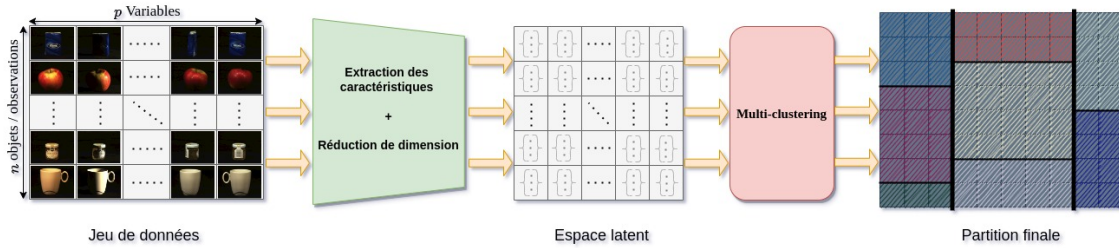


FIGURE 1 – Le framework complet

Méthode	Vues	Scores		
		<i>RI</i>	<i>Purity</i>	<i>NMI</i>
<i>GMM</i>	-	0.890	0.299	0.367
<i>LBM</i>	-	0.892	0.298	0.367
Le notre	Vue 2	<b>0.899</b>	0.669	0.592
	Vue 1	<b>0.899</b>	0.682	0.596
	Vue 3	<b>0.899</b>	<b>0.685</b>	<b>0.599</b>
	Vue 10	0.898	0.579	0.560

TABLE 1 – Résultats des métriques *RI*, *Purity* et *NMI*.

## 4.2 Extraction des caractéristiques

Chaque image a été encodée en utilisant le vit\_b16<sup>2</sup> pour obtenir une représentation vectorielle de dimension  $d = 768$ . L'analyse en composantes principales avait permis la réduction de la dimension à  $d_r = 53$  conservant variance expliquée à 50%.

## 4.3 Évaluation du multi-clustering

### 4.3.1 Modèles concurrents

Afin d'évaluer l'approche proposée dans ce papier, le multi-clustering a été comparé aux deux méthodes paramétriques de l'état de l'art : le Modèle de Mélange Gaussien (*GMM*) [4] et le Modèle aux Blocs Latents (*LBM*) [6].

### 4.3.2 Paramétrisation

Le multi-clustering non-paramétrique possède 4 paramètres de concentration ( $a_\gamma, b_\gamma$ ) et ( $a_\alpha, b_\alpha$ ). L'expérience réalisée dans ce travail, fixe ces paramètres à (10, 1) et (10, 5) respectivement, le nombre d'itérations utilisé est  $M = 20$ . *GMM* et *LBM* sont deux modèles paramétriques; *GMM* possède un paramètre  $k$  (nombre de *clusters* lignes). Une recherche exhaustive en optimisant le critère *ICL* [5] a été réalisée sur les éléments de la liste {60, 76, 100, 200} pour trouver le  $k$  optimal. *LBM* quant à lui possède deux paramètres  $k$  et  $l$  (nombre de *clusters* lignes et colonnes). Une recherche similaire dans les listes {60, 76, 100, 200} resp. {10, 14, 18, 20, 30, 33} a été réalisée pour estimer les  $k$  et  $l$  optimaux. L'étape d'extraction des caractéristiques et la réduction de dimension est communes aux trois méthodes.

### 4.3.3 Critères d'évaluation

La qualité des partitions lignes inférées (l'unique partition ligne du *GMM* et *LBM* et les multiples partitions inférées

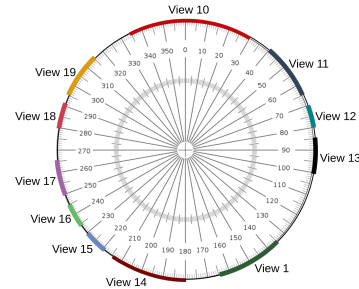


FIGURE 2 – Positions spatiales des différentes vues

Vues	Clusters de variables	Interprétation
Vue 1	r_135, ..., r_165	Agrégation de légères rotations
Vue 2	i110, i120, i130, i150, i160	Température faible
Vue 3	i3_c1, i3_c2, i3_c3	Projecteur frontal et caméra 3
Vue 4	i1_c1, i2_c1, i6_c1	Projecteur à droite et caméra frontale
Vue 5	i1_c2, i2_c2, i6_c2	Projecteur à droite et caméra 2
Vue 6	i1_c3, i2_c3, i6_c3	Projecteur à droite et caméra 3
Vue 7	i4_c1, i5_c1, i7_c1	Projecteur à gauche caméra frontale
Vue 8	i4_c2, i5_c2, i7_c2	Projecteur à gauche et caméra 2
Vue 10	i_140, i_170, i_180, i_190, i_210, i_230, i_250, i8_c1, i8_c2, i8_c3, r_0, ..., r_30, r_335, ..., r_355	Capture frontale
Vue 14	r_180, ..., r_215	Capture prise de derrière

TABLE 2 – Les dix-neuf (33 - 14) *clusters* en colonnes obtenus par le multi-clustering

par le multi-clustering) ont été comparées en utilisant la variable *Material* comme annotation externes (labels) : trois métriques avaient été évaluées sur les résultats des trois méthodes sur la base de cette annotation : l'indice de Rand (*RI*), la pureté (*purity*), et l'information mutuelle normalisée (*NMI*).

### 4.3.4 Résultats

Le multi-clustering non-paramétrique avait inféré (33) *cluster* en colonne (Vues) : seuls (19) d'entre eux constituent réellement des agglomérations des colonnes originales, les (14) autres *clusters* sont des singletons. Ceci est peut-être dû à une surestimation du nombre de *clusters* en colonne. Le tableau 2 présente certaines vues ainsi que les variables qui y sont affectées ainsi que les interprétations proposées.

Les résultats des évaluations métriques sont présentées dans le tableau 1 et montrent que les solutions obtenues avec le multi-clustering enregistrent de meilleurs scores comparées aux approches classiques (*GMM* et *LBM*). Ces dernières

2. <https://github.com/faustomorales/vit-keras>



FIGURE 3 – Un exemple de partition inférée par le *multi-clustering*. Elle représente la vue 2 et les partitions lignes, de haut en bas (23, 0, 65) respectivement.

présentent des indices de Rand (RI) comparables à celui du *multi-clustering* mais les scores de la pureté et de l'information mutuelle normalisée sont beaucoup plus faibles. A noter que seules les vues (2,1,3 et 10) sont présentées dans le tableau pour *multi-clustering* pour alléger la lecture de ce dernier, les autres vues présentent des métriques similaires. En plus des bons scores enregistrés par les *clusters* lignes du *multi-clustering*, les *clusters* colonnes (vues) préservent la topologie spatiale des prises comme l'illustrent le tableau 2 et la figure 2.

En effet, la figure 2 montre les positions spatiales des différentes vues. elles sont ordonnées et forment des partitions compactes d'un point de vue spatial. La figure 3 montre un exemple de partition inférée par le *multi-clustering*. Elle représente la vue 2, et les quelques partitions lignes correspondantes, respectivement, (23, 0, 65, 162) de haut en bas. Les résultats montrent que les *clusters* lignes regroupent des objets similaires (même nature, formes,...). Ces résultats sont également vérifiés pour les (33) autres vues.

Contrairement au *multi-clustering*, le modèle à bloc latent (*LBM*) a échoué dans la tâche du regroupement des prises similaires dans le même *cluster*. La figure 4 montre un exemple de bloc latent regroupant dans le même *cluster* colonne (0) différentes prises telles que ( $r_{85}$ ,  $r_{235}$ , et  $r_{275}$ ) qui restent différentes et éloignées du point de vue spatial. De plus, il regroupe dans la même partition ligne des objets différents d'un point de vue humain (formes différentes,...); pour certains objets affectés à ce bloc latent, les affectations par *multi-clustering* conservent une cohérence (voir exemple sur la vue (3) de la figure 5).

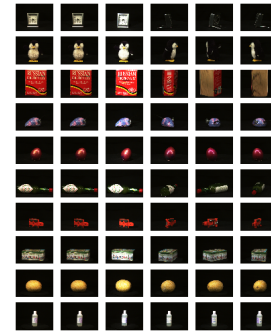


FIGURE 4 – Bloc latent du *cluster* en colonne (0) et *cluster* ligne (75) inféré par le *LBM*

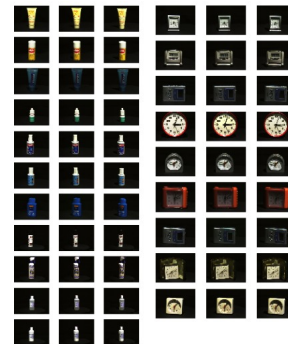


FIGURE 5 – *Cluster* ligne (3) (droite) et (18) (gauche) de la vue (3) inférés par le *multi-clustering*.

## 5 Conclusion et perspectives

Dans ce papier, nous avons proposé un *framework* pour inférer de multiples solutions de *clustering* sur un jeu de données d'images multivues. Notre *framework* combine un *vision transformers* pré-entraîné avec un algorithme du *multi-clustering*. Le problème majeur des algorithmes de *multi-clustering* paramétriques est l'étape de sélection de modèle lorsque le nombre de modèles possibles est grand. Nous avons proposé une approche bayésienne non paramétrique, qui permet d'estimer la structure du modèle durant l'inférence. Notre approche a fournit des blocs homogènes et cohérents. De plus, elle a surpassé les méthodes traditionnelles telles que le *GMM* et *LBM*.

Une de nos perspectives consiste à étudier la possibilité d'unifier notre *framework* afin de joindre l'apprentissage profond des représentations à l'aide des *transformers* et l'étape du *multi-clustering*.

## Références

- [1] Charles E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6) :1152–1174, 1974.
- [2] Charles E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6) :1152 – 1174, 1974.

- [3] Timothy Baldwin and J. Ford. Transfer of training : A review and directions for future research. *Personnel Psychology*, 41 :63–105, 03 1988.
- [4] Jeffrey D. Banfield and Adrian E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3) :803–821, 1993.
- [5] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7) :719–725, jul 2000.
- [6] Charles Bouveyron, Laurent Bozzi, Julien Jacques, and François-Xavier Jollois. The Functional Latent Block Model for the Co-Clustering of Electricity Consumption Curves. *Journal of the Royal Statistical Society : Series C Applied Statistics*, December 2017.
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet : A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words : Transformers for image recognition at scale, 2020.
- [10] Giuliano Galimberti and Gabriele Soffritti. Model-based methods to identify multiple cluster structures in a data set. *Computational Statistics & Data Analysis*, 52(1) :520–536, 2007.
- [11] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders. The amsterdam library of object images. *International Journal of Computer Vision*, 61(1) :103–112, 2005.
- [12] Etienne Goffinet. *Multi-Block Clustering and Analytical Visualization of Massive Time Series from Autonomous Vehicle Simulation*. Theses, Université Paris 13 Sorbonne Paris Nord, December 2021.
- [13] Etienne Goffinet, Anthony Coutant, Mustapha Lebbah, Hanane Azzag, and Loïc Giraldi. CONDITIONAL LATENT BLOCK MODEL : A MULTIVARIATE TIME SERIES CLUSTERING APPROACH FOR AUTONOMOUS DRIVING VALIDATION. working paper or preprint, January 2022.
- [14] Yue Guan, Jennifer Dy, Donglin Niu, and Zoubin Ghahramani. Variational inference for nonparametric multiple clustering. 01 2010.
- [15] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24 :498–520, 1933.
- [16] Reda Khoufache, Mohamed Djallel Dilmi, Hanane Azzag, Etienne Goffinet, and Mustapha Lebbah. Emerging properties from bayesian non-parametric for multiple clustering : Application for multi-view image dataset. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 31–38, 2022.
- [17] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. page 77, 01 2009.
- [18] Hongfu Liu, Ming Shao, Sheng Li, and Yun Fu. Infinite ensemble for image clustering. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, page 1745–1754, New York, NY, USA, 2016. Association for Computing Machinery.
- [19] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.
- [20] Durvesh Malpure, Onkar Litake, and Rajesh Ingle. Investigating transfer learning capabilities of vision transformers and cnns by fine-tuning a single trainable block, 2021.
- [21] Vikash Mansinghka, Eric Jonas, Cap Petschulat, Beau Cronin, Patrick Shafto, and Joshua Tenenbaum. Cross-categorization : A method for discovering multiple overlapping clusterings. 01 2009.
- [22] Jeffrey W. Miller. An elementary derivation of the chinese restaurant process from sethuraman’s stick-breaking process, 2018.
- [23] Noémi Schuurman, Raoul Grasman, and Ellen Hamaker. A comparison of inverse-wishart prior specifications for covariance matrices in multilevel autoregressive models. *Multivariate behavioral research*, 51 :1–22, 03 2016.
- [24] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4(2) :639–650, 1994.
- [25] F. Tian, B. Gao, Q. Cui, Enhong Chen, and T.-Y. Liu. Learning deep representations for graph clustering. *Proceedings of the National Conference on Artificial Intelligence*, 2 :1293–1299, 01 2014.
- [26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [27] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters, 2016.