Compte-rendu Étude de cas Clustering sur des signatures « en ligne »

NJIFENJOU Ahmed

LIVRABLE I : OBSERVATION DE QUELQUES SIGNATURES

On a choisi aléatoirement 4 individus dont on observe les signatures. Finalement, après observations on a remarqué que les ordonnées (y) avaient déjà été inversée donc en utilisant (-y) pour la représentation, les signatures représentées ne semblaient pas « logique » pour l'œil humain. Ainsi, on obtient les représentations suivantes :

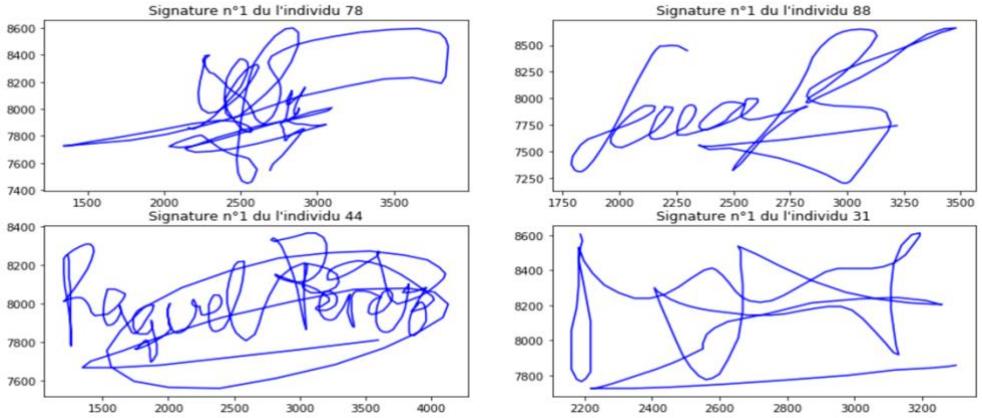


Figure 1: Observation de 4 signatures parmi les 100 de notre base de données

LIVRABLE II : CATEGORISATION DES 100 PERSONNES EN 3 CLUSTERS

On dispose des complexités de 25 signatures par individus sous 3 modèles de mélange de gaussiennes(GMM) différents. Le but étant de catégoriser les individus, la métrique qu'on utilise est la moyenne de ces complexités par individus. Ce choix n'est probablement pas optimal car on perd de l'information en prenant la moyenne notamment la dispersion des valeurs et celles qui sont très fortement éloignées de la moyennes(« outliers »).

Ainsi pour chaque GMM, on commence par calculer la complexité moyenne des signatures par individu. On remarque que plus le nombre de gaussiennes dans le modèle augmente moins les complexités sont élevées, en témoignent l'évolution des valeurs des centres des différents clusters (voir **1er histogrammes**). En effet la complexité ici est calculée avec l'entropie différentielle : $h(x) = -\int_{\mathbb{X}} f(x) \ln f(x)\, dx$
où f(x) sera notre densité de mélange sous la forme $g(x,\theta_{1,\dots,}\ \theta_K)$ avec K valant 4, 8 ou 24. Les θ_i représentant les paramètres des différentes gaussiennes utilisées dans le modèle, elles sont pondérées par des π_k dont la somme fait 1.

Cette notion d'entropie différentielle est une extension de l'entropie de Shannon vue en théorie de l'information, ainsi elle représentent l'incertitude qu'on a sur l'information transmise. Ainsi, il est donc logique que plus le modèle est complexe en terme de nombre de gaussienne, moins cette incertitude est grande d'où la diminution des valeurs de complexités observées.

Synthèse des résultats

Méthode 1: K-moyennes avec distance euclidienne

On utilise la distance euclidienne ici, car l'algorithme K-means est optimisé pour celle-ci.

- Tableau 1/ Clusters de type N°1 : complexité moyenne faible et variance élevée

Caractéristique\NG	4	8	24
Pourcentage de personnes appartenant au cluster	18,00%	8,00%	7,00%
Complexité moyenne des signatures du cluster	27.811652	22.687009	11.655480
Variance	0.568700	0.973108	2.817467

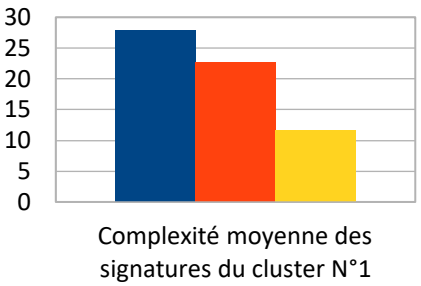
- Tableau 2/ Clusters de type N°2 : complexité moyenne intermédiaire et variance intermédiaire

Caractéristique\NG	4	8	24
Pourcentage de personnes appartenant au cluster	40,00%	52,00%	44,00%
Complexité moyenne des signatures du cluster	29.763118	27.880915	23.016404
Variance	0.184842	0.541212	2.592242

- Tableau 3/ Clusters de type N°3 : complexité moyenne élevée et variance faible

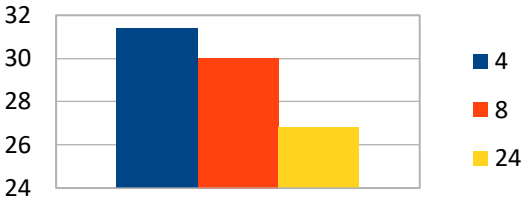
Caractéristique\NG	4	8	24
Pourcentage de personnes appartenant au cluster	42,00%	40,00%	49,00%
Complexité moyenne des signatures du cluster	31.386733	30.010458	26.796095
Variance	0.296468	0.373008	1.005286

À l'aide ces tableaux, on obtient les histogrammes suivant :



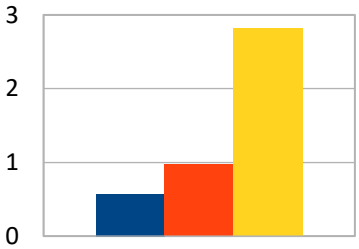
Complexité moyenne des signatures du cluster N°1

Complexité moyenne des signatures du cluster N°2

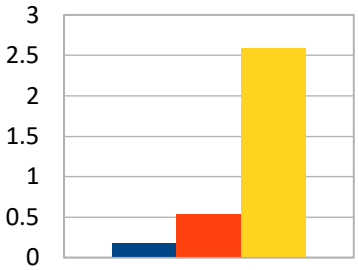


Complexité moyenne des signatures du cluster N°3

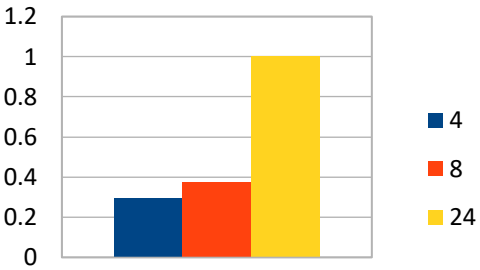
Comme noté en début de cette partie, les valeurs des centres des clusters (complexité moyenne du clusters sur les graphes ci-dessus) diminuent avec le nombre de gaussienne du fait de l'entropie différentielle. Ainsi comparer les trois modèles sur ce paramètre serait difficilement pertinent.



Variance N°1

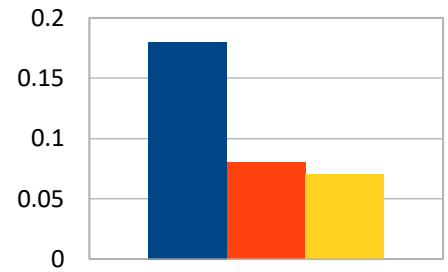


Variance N°2

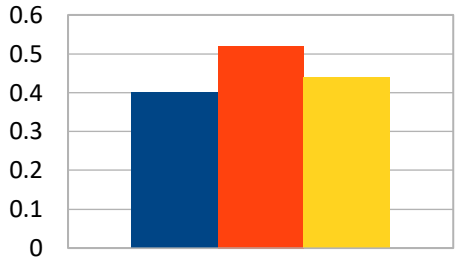


Variance N°3

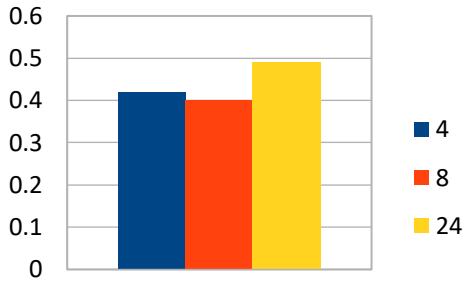
Par contre on note que pour chacun des types de clusters, plus on augmente la complexité du modèle plus la variance des données dans le cluster est grande(voir ci-dessus). Ceci veut dire que les données en entrée sont plutôt dispersées. Alors l'augmentation de la complexité du modèle(en terme du nombre de gaussienne) permet de bien séparer les signatures (du moins les individus vu qu'ici on travaille sur les moyennes par individus).



Pourcentage de personnes appartenant au cluster N°1



Pourcentage de personnes appartenant au cluster N°2

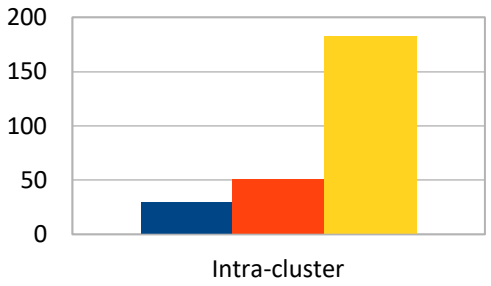


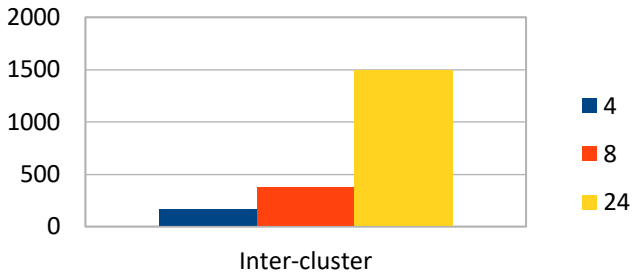
Pourcentage de personnes appartenant au cluster N°3

Au vu de ces graphiques, il est difficile de faire un lien direct entre le nombre de gaussiennes utilisées et la populations dans les différents clusters. En effet, pour les cluster a moyenne faible et variance élevé(Type N°1) on a une décroissance en fonction du nombre de gaussienne tandis que pour les autres le lien n'est pas évident. Il faudrait peut-être envisager un échantillon encore plus grand sur lequel on ferait une classification pour différents nombres de cluster afin de possiblement voir apparaitre une relation.
Toutefois il est important de noter qu'on retrouve le moins de personne dans les clusters à faible moyenne car cela correspond aux signatures de faible complexité(donc de faible entropie différentielle) i.e. les signatures sur lesquelles l' « incertitude » pour reprendre les termes de théorie de l'information est faible. Elle sont logiquement en sous-nombre car moins « désordonnée ». La signature ayant pour vocation l'identification unique, elle doit être difficilement reproductible, une entropie trop faible représente une signature pauvre en sécurité. Donc par nature les signatures définies par les gens sont généralement complexes(« désordonnées ») d'où le pourcentage élevé dans les autres clusters. Pour diminuer encore plus cette valeur on pourrait envisager des GMM à plus de gaussiennes cependant on prendrait le risque en entrainant le modèle à fortement dépendre des données d'entrainement(sur apprentissage).

Valeurs des inerties :

Inertie\NG	4	8	24
Intra-cluster	30.0819461117	50.8482055034	183.039971206
Inter-cluster	168.331969997	378.175326179	1491.83207165





Ces inerties sont calculées à l'aide des formules données dans le cours et la distance utilisée est la distance euclidienne qui en une dimension est juste la valeur absolu de l'écart entre les deux éléments (équivalent donc à la distance de Manhattan dans ce cas précis).

On remarque que l'inertie, qu'elle soit intra ou inter-cluster, augmente avec le nombre de gaussiennes utilisées. On voit même que pour le modèle GMM24, elles atteignent des valeurs extrêmement élevées. On retrouve encore le fait que plus le modèle est complexe plus les données vont être séparées et donc les inerties montent. Étant donné que pour avoir un bon clustering on doit minimiser l'inertie intra-cluster et maximiser l'inertie inter-cluster, cette augmentation nous arrange si on veut privilégier la disparité des classes entre elles à leur homogénéité. Si on veut privilégier l'homogénéité alors un modèle à faible nombre de gaussienne est préférable. Ainsi, il faut trouver le compromis acceptable afin de minimiser l'inertie intra-cluster et de maximiser l'inertie inter-cluster en fonction du nombre de gaussiennes utilisées.

Méthode 2: Ascension hiérarchique

- Tableau 4/ Clusters de type N°1 : complexité moyenne faible et variance élevée

Caractéristique\NG	4	8	24
Pourcentage de personnes appartenant au cluster	20,00%	8,00%	7,00%
Complexité moyenne des signatures du	27.917292	22.687009	11.655480

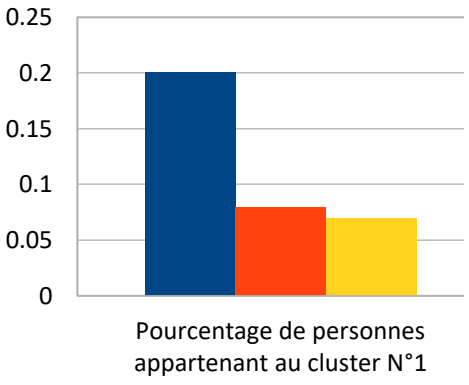
cluster			
Variance	0.612272	0.973108	2.817467

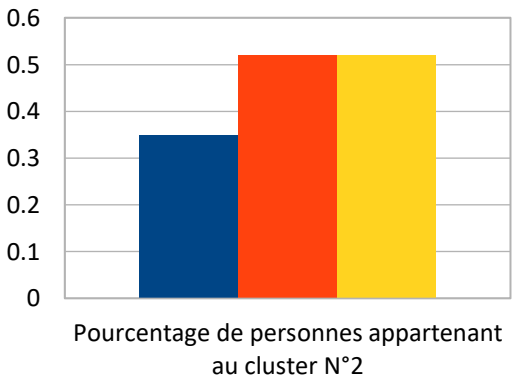
- Tableau 5/ Clusters de type N°3 : complexité moyenne et variance intermédiaire

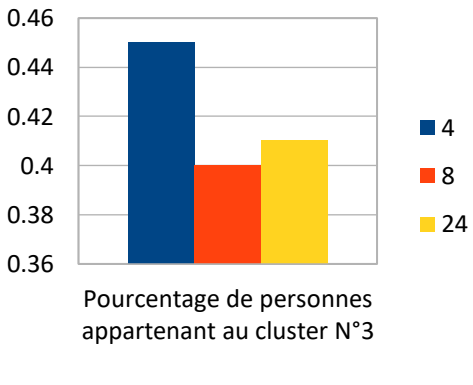
Caractéristique\NG	**4**	**8**	**24**
Pourcentage de personnes appartenant au cluster	35,00%	52,00%	52,00%
Complexité moyenne des signatures du cluster	29.755524	27.880915	23.369515
Variance	0.124673	0.541212	2.891756

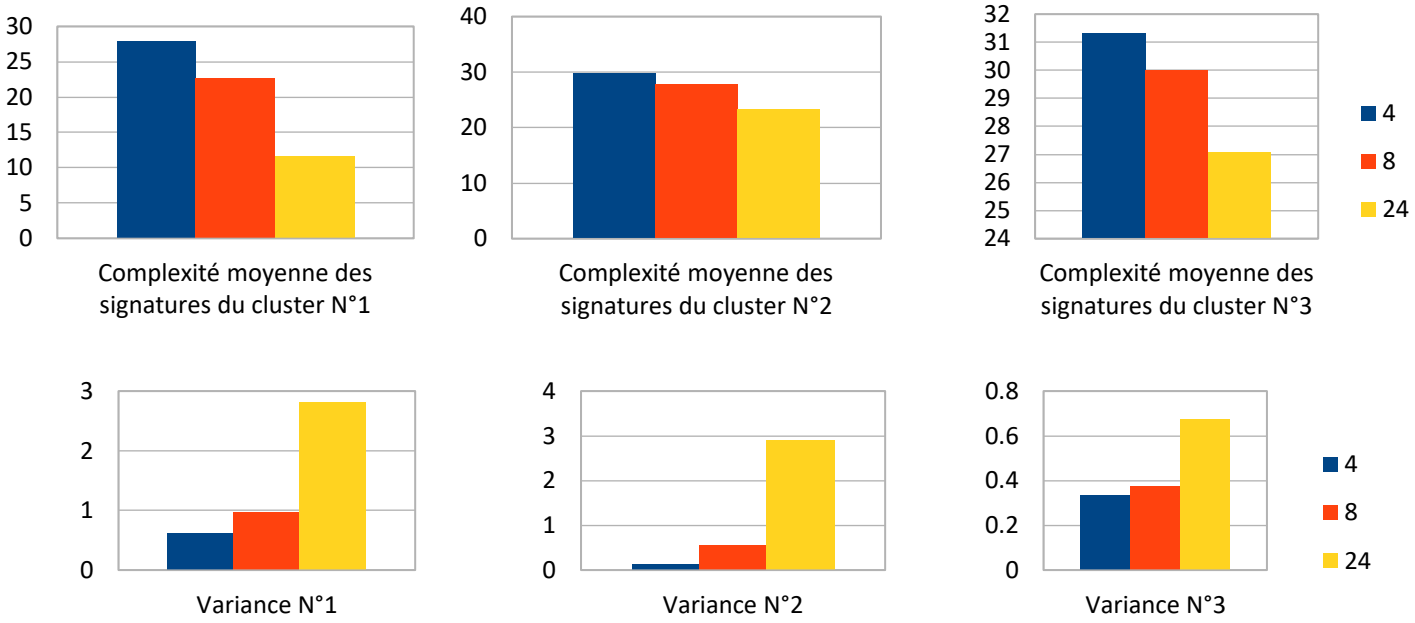
- Tableau 6/ Clusters de type N°3 : complexité moyenne élevée et variance faible

Caractéristique\NG	**4**	**8**	**24**
Pourcentage de personnes appartenant au cluster	45,00%	40,00%	41,00%
Complexité moyenne des signatures du cluster	31.324179	30.010458	27.085749
Variance	0.331859	0.373008	0.671648





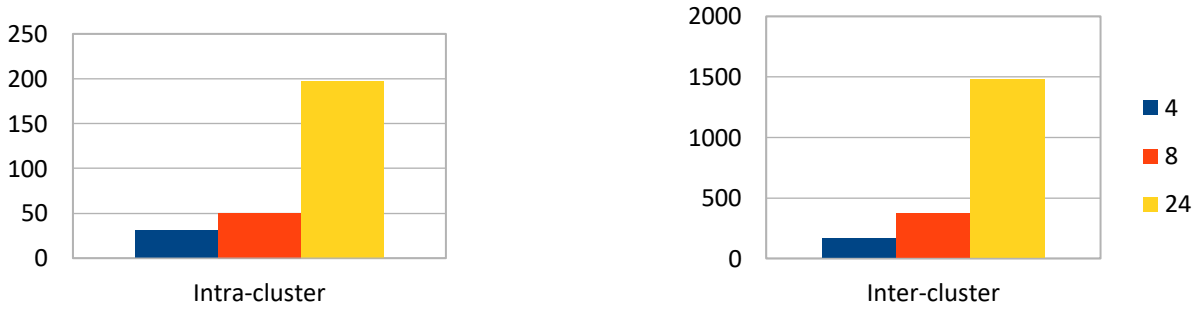




On voit qu'au niveau des différentes catégories, les constats restent les mêmes que ceux mis en évidence avec le K-means. Ayant pareillement imposé 3 clusters à l'algorithme et le jeu de données étant petit, il est logique que les clustering se comportent globalement de la même manière.

Inertie\NG	4	8	24
Intra-cluster	31.542610617	50.8482055034	197.631168946
Inter-cluster	166.871305492	378.175326179	1477.24087391

Ainsi, au-delà de l'interprétation des classes, les deux méthodes donnent des résultats assez proches dans le détails. Sachant que à chaque fois que l'on fait tourner l'algorithme K-means il y a une nouvelle initialisation qui conditionne la classification, on pourrait se demander si la légère différence n'est pas due à celle-ci. Les inerties montrent bien que les deux méthodes sont proches, avec un léger avantage à la méthode K-means avec



distance euclidienne (où l'initialisation est un facteur non négligeable).

Nombre optimal de cluster existants :

On utilise pour ce faire le coefficient de silhouette qui pour chaque point mesure la différence entre la distance moyenne avec les points du même groupe que lui et la distance moyenne avec le points des autres groupes voisins. Et le coefficient de silhouette global est la moyenne des coefficients de silhouette pour tous les points. On obtient le graphe suivant :

Étant donné que plus la silhouette est grande mieux le clustering est (on a plus de points bien classés) on en déduit que le nombre optimal de clusters est **2.** On aurait pu l'observer aussi, en faisant la méthode CAH sans imposer le nombre de clusters au départ.

LIVRABLE III : CATEGORISATION DES 2500 SIGNATURES MODEL GMM24

On choisit d'utiliser la méthode K-means.

Les clusters obtenues sont caractérisés comme suit :

GMM24/2500 signatures	% de signatures	Moyenne	Variance
Cluster 1 : moyenne et variance intermédiaire	33,48%	22.310420	3.083082
Cluster 2 : moyenne faible, variance élevée	6,96%	11.505539	8.469476
Cluster 3 : moyenne forte, variance faible	59,56%	26.532697	1.585257

Pour tester si les signatures d'une même personne appartiennent toutes à la même catégorie, on a choisi de faire la **somme des labels affectés aux signatures d'un même individus**. En effet, on ne sait pas à l'avance à quel cluster devrait appartenir l'ensemble des signatures d'un individus mais on sait que si elles ont toutes été classées dans le même cluster, cette somme vaudra soit 0, soit 25, soit 50. Toutes sommes différentes de ces 3 valeurs signale donc un individu dont les signatures n'appartiennent pas au même cluster.
Ceci n'est pas une méthode optimale car il est vrai qu'on peut obtenir par exemple un total de 25 pour une suite de label non tous à 1 même si cela est très peu probable. En effet, au vu des labels associés à chaque individu, le nombre de label différents est souvent très faible.

Ainsi on observe que pour certains individus toutes les signatures ne sont pas associées à un unique cluster. Il s'agit des individus dont la liste suit :

```
[1, 3, 4, 6, 7, 9, 14, 15, 17, 19, 20, 21, 22, 23, 27, 29, 30, 34, 35, 37, 38, 40,
42, 47, 49, 50, 51, 56, 57, 58, 59, 61, 62, 63, 65, 66, 67, 72, 73, 74, 76, 78, 79,
 80, 85, 86, 88, 90, 99]
```

Quand on regarde les complexités associées à ces individus et qu'on compare leurs valeurs aux centres des différents clusters, on observe qu'il y en a qui sont entre deux centres de clusters. Soit presqu'au milieu du segment qui les sépare, soit plus proche d'un cluster auquel n'appartient pas la majorité des signatures de

l'individu.

Par exemple pour l'individu n°99 on a la séquence de complexité suivante :

```
[ 24.75389247,  22.32664027,  23.54342033,  24.94904457,
         21.87812839,  22.23607104,  24.25955269,  23.67429876,
         25.49102723,  22.78721392,  22.77633756,  25.78513889,
         22.56886427,  22.71995824,  21.93302674,  21.74916508,
         25.58509436,  21.76216959,  22.39389515,  21.51083071,
         20.21357896,  20.97371801,  22.4602686 ,  20.61011107,  20.7058419 ]
```

On voit que la majorité, des complexités sont proches du centre du cluster 1 à savoir 22.310420 donc on devrait en principe avoir une somme de label des signatures de cet individu égale à 0 contre 10 obtenu. Les 5 valeurs surlignées ci-dessus, représentent probablement celles qui ont été mal classées et c'est logique, quand on voit qu'elles sont plutôt proche du centre du cluster 3 à savoir 26.532697 ; la complexité minimale prise dans ce cluster doit y être inférieur. Enfin, on a bien 5 valeurs d'où 5 labels valant 2 et une somme égale à 10. Le raisonnement est ainsi pareil pour les autres individus dont certaines signatures sont « mal » classées. Ces données seraient donc les données aberrantes à retirer à ces individus pour obtenir un clustering optimal.

On voit donc qu'en utilisant la moyenne pour classifier les individus on perd l'information sur les valeurs aberrantes qui existent chez certains individus car ces données sont « aplaties » par la moyenne. Mais on peut aussi se poser la question de savoir si finalement la moyenne n'est pas suffisante car avec celle-ci l'individu serait dans la bonne catégorie (les valeurs « non-aberrantes » étant en large surnombre). De plus en détectant les valeurs aberrantes on aura tendance à les supprimer pour retomber sur une classification des individus identique à celle donnée en prenant la moyenne. Ainsi, en fonction de la problématique traitée on pourra donc choisir la moyenne ou pas.

LIVRABLE IV : APPRENTISSAGE ET GENERALISATION AVEC LE MODEL GMM24

Ici on utilise une fois de plus la méthode K-means pour classer les signatures.

Avant de commencer l'apprentissage, on mélange toute les signatures aléatoires, pour vraiment supprimer l'appartenance à un individu et aussi la notion d'ordre qui peut avoir une influence sur l'exécution de l'algorithme K-means : `GMM240=shuffle(GMM24)`.

On sépare ensuite notre jeu de signature en un 2 partie égale : le training_set prend les 1250 premières signatures, et le testing_set prend le reste.
On commence cependant par faire un clustering sur l'ensemble des signatures mélangées `GMM240` pour pouvoir par la suite comparer les prédictions effectuées sur les données de tests avec les « vraies » valeurs déterminées ici.

Apprentissage : On effectue un K-means sur les données d'entrainement et on obtient les clusters suivant :

GMM24/1250 signatures	% de signatures	Moyenne	Variance
Cluster 1 : moyenne et variance intermédiaire	62,8%	26.453971	1.840675
Cluster 2 : moyenne faible, variance élevée	7,04%	10.880787	7.861505

Cluster 3 : moyenne forte, variance faible	30,16%	21.887489	3.459674

Prédiction : On utilise la méthode des k-plus proche voisins pour prédire les clusters dans lesquels seront assignées les signatures de l'échantillons de test i.e. on va calculer la distance qui sépare leurs complexités aux centres de clusters et les associer au cluster pour lequel on aura eu la distance minimale.
Une fois la prédiction réalisée on comparer les labels prédits et les vrais labels obtenus avant l'apprentissage et on recense le nombre de mauvais prédictions.

On obtient ainsi un taux d'erreur de `3.76%` sur 1250 prédiction, ce qui est un taux d'erreur acceptable. On arrive plus ou moins à généraliser le modèle à des données inconnues. Cependant on pourrait essayer d'améliorer ce taux d'erreurs en faisant varier le nombre de clusters par exemple et garder celui qui garantit le taux d'erreurs le plus faible.

Annexes Python :

Vous trouverez à cette adresse notre notebook avec tous les codes :

https://github.com/ahmednjifenjou/Nonsupervised-signature-classification