Nom et Prénom : PENG Ziqian
L'Enseignant : DECELLE Aurélien

Rapport de Projet non supervisé

Introduction

Ce projet consiste à réaliser une mixture de gaussienne pour traiter des données, par exemple, pour classifier les données de couleur en RGB d'une image en png. Il contient 2 parties, la première est pour vérifier que l'algorithme fonctionne en classifiant correctement les données par couleur et la seconde est des applications de cet algorithme.

Pour ce projet, j'ai réalisé la première partie, les trois premiers et le dernier éléments dans la liste. De plus, je me suis permis de reprendre et *MixGauss* et *TasGauss* de TP et de les adapter au projet. J'ai fait ce projet toute seule mais je restais en contact avec des camarades pour discuter les difficultés et les bugs rencontrés.

Résultat

PartieI *TriParCouleur*

Pour commencer, j'ai décidé de poser 8 centres puisqu'il y a 6 couleurs de mms, une couleur du fond et une couleur pour le reflet. Dans ce cas, j'ai choisi les 6 couleurs inverses de mms et du fond ainsi que le noir pour le reflet car j'ai traité les données d'image inverse.

Après avoir traités par la mixture de gaussienne, j'ai obtenu une image similaire que mms.png mais elle n'a que 4 couleurs de mms (cf Figure1) avec un score de 13.018382373593262.

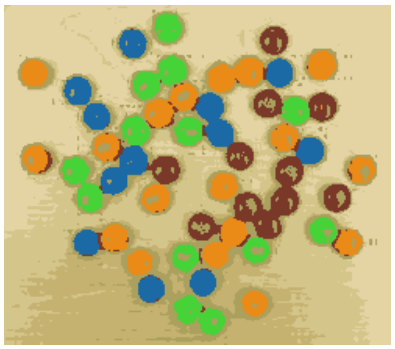

Figure 1

L'application

1. Elément I *CentreAlea10*

Pour cette partie, j'ai initialisé10 fois les 10 centres aléatoirement en utilisant la méthode *initCentre*. J'ai calculé le score pour chaque partition et j'ai remarqué qu'ils sont tous environs 13.1. Pour bien visualiser les résultats, je les ai stocké en image

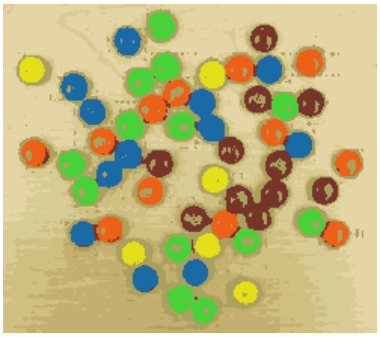

Figure2 AleaCtr5_MaxScore

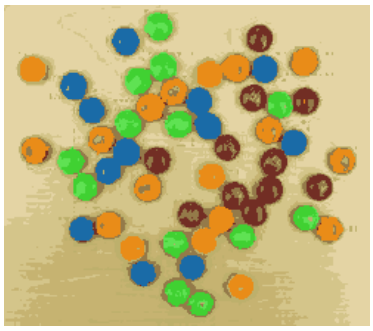

Figure3 AleaCtr7_MinScore

avec *saveImg* et je les ai rangé dans la répertoire *ctrAleaImg*.

En comparant la meilleur et la pire partition, j'ai remarqué que la meilleur partition a une score plus haut et des couleurs plus complets (cf Figure2 et Figure 3).

2. Elément II *CentreMeuilleurNbr*

Cet élément consiste à trouver le meilleur nombre de centres entre 2 et 10, avec 10 conditions initialisées aléatoirement pour chaque nombre, afin d'avoir un résultat plus correct.

Pour chaque k, j'ai stocké le résultat de la partition avec le score le plus haut et sous le nom de *"nbrCtr"+k*.png. Pour garder les informations de chaque partition, les images *"nbrCTR"+k+"CI"* .png ont aussi stocké dans la répertoire *MeilleurKImg*.

Les scores de chaque K pourrait varier entre -3 et 13.6 mais le meilleur score restait entre 11 et 13.6. En comparant les résultats et en traçant les scores par gnuplot, je me suis rendu compte que le meilleur nombre de centres est k =10 (cf Figure4).

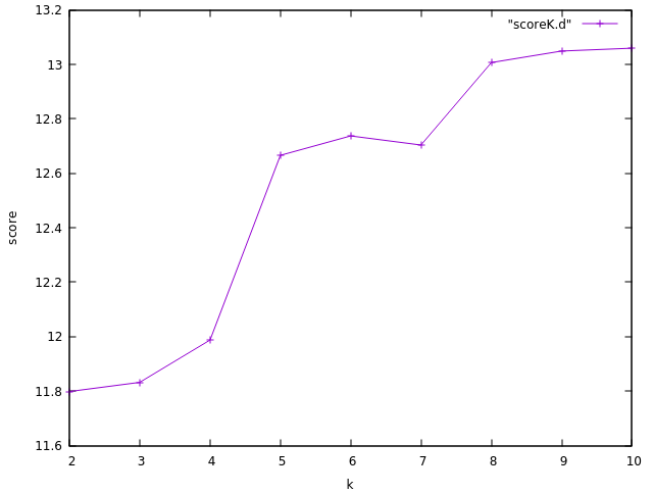

Figure 4 score pour k entre 2 et 10

Cependant, l'algorithme produirait peut-être des images tout bleu avec des scores négatifs si j'exécute plusieurs fois le main dans cette classe. Dans ce cas de mon côté, il faut redémarrer IntelliJ pour le rendre normale.

3. Elément III *TriPourImg2*

J'ai choisi l'image2 *imgQ3.png* pour cette partie car elle a 9 couleurs différentes et distinguées. J'ai décidé de poser 10 centres en considérant le résultat de l'élément 2 . En

essayant aussi pour k = 9 et pour k = 11, j'ai remarqué que k = 10 reste le meilleur.

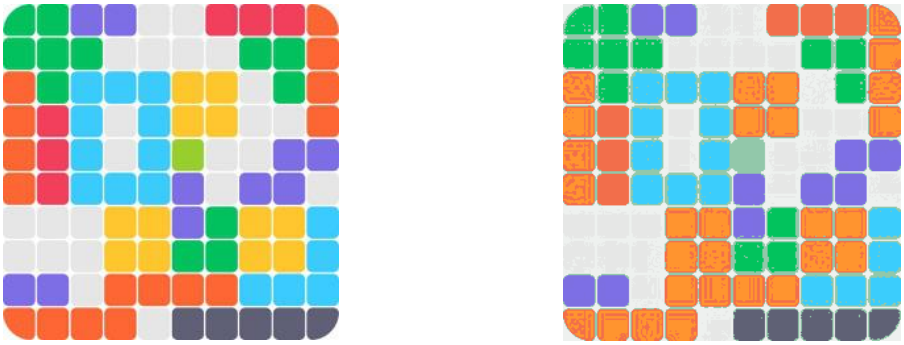

Figure 5 à gauche l'image originale imgQ3.png , à droite, le résultat avec 10 centres

4. Elément V *MillePts1D*

Pour cette partie j'ai repris la classe *TasGauss* de TP et Je l'ai adapté pour la mixture de gaussienne en 2D. La mixture a bien appris la densité car elle a bien trouvé que densite = 500 pour chaque cluster. De plus, pour le centre et la variance, la mixture a bien fonctionné pour classifier les données autour de -2 avec une variance de 0.2, mais pour l'autre centre, le résultat est autour de 2 et la variance est autour de 0.38.

Cependant, ce résultat est cohérent avec les données de l'histogramme.

Par ailleurs le score de cette partie était environ 6.36 et il a varié en fonction de différents centres initialisés. Enfin, j'ai fait le fichier de l'histogramme avec 12 colonne par la méthode *figureFile* dans *MillePts1D*

Conclusion

Grâce à ce projet, j'ai bien compris les fonctionnalités et les applications de la mixture de gaussienne, surtout pour le traitement des images. De plus, je me suis rendu compte qu'il faut mieux d'avoir 10 centres pour traiter les données de couleur en RGB. Pourtant, ma mixture de gaussienne