Classification non supervisée

1 Objectifs

- Découvrir la classification non supervisée et pratiquer deux algorithmes particuliers

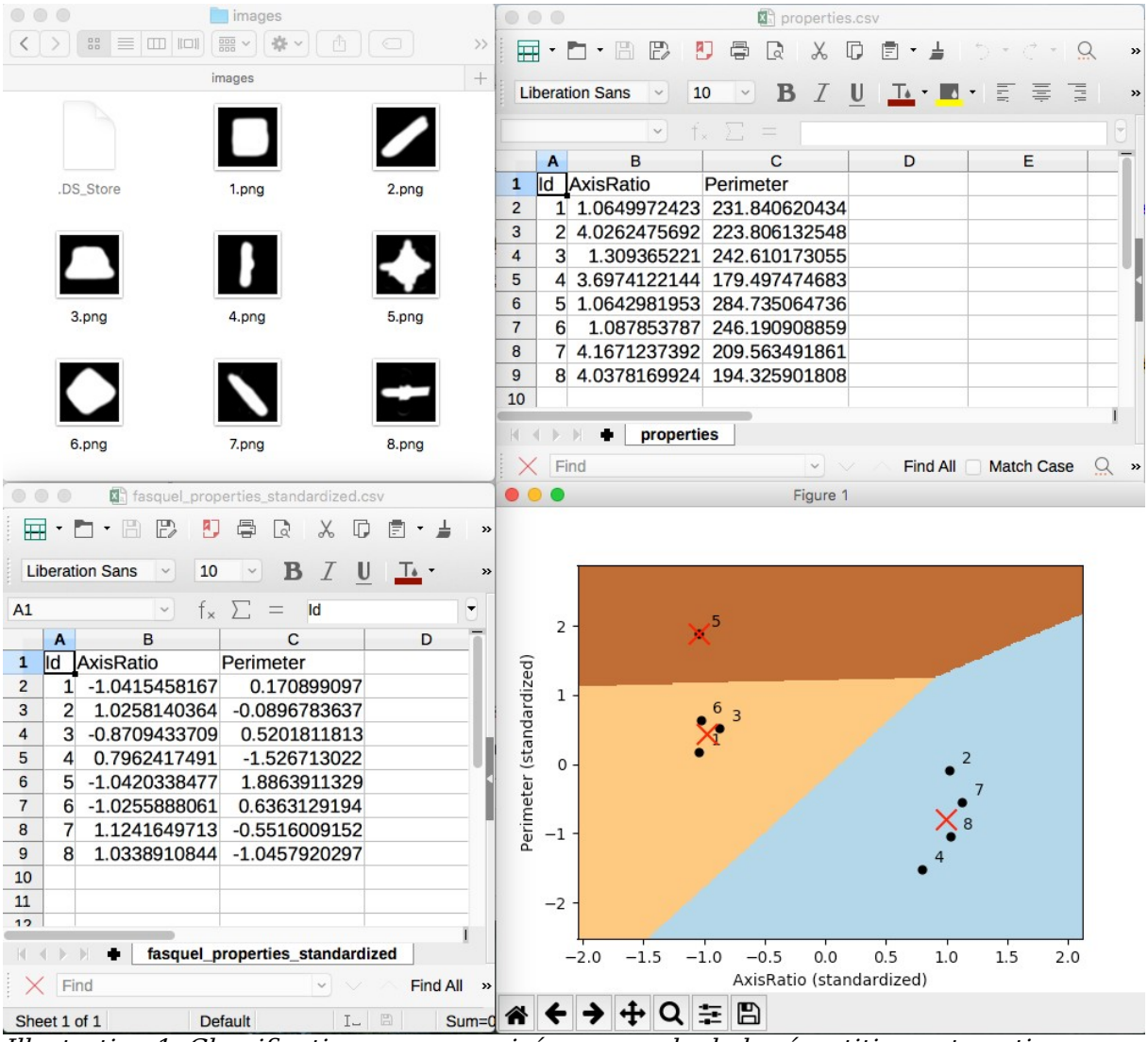

Illustration 1: Classification non supervisée : exemple de la répartition automatique d'images en catégories

2 Données considérées et exemple fourni

La classification non supervisée traite des données d'entraînement dont on ne connaît pas la classe d'appartenance (pas « d'étiquette »). Il s'agit de regrouper automatiquement les données en catégories (partitionnement des données): une catégorie correspond à des données « similaires » entre elles, par rapport aux données des autres catégories. En pratique, cette approche peut-être utilisée par exemple pour :

- La recherche de mots similaires à un mot inconnu: les mots déjà rencontrés ont été regroupés (sans étiquette préalable) en catégories.
- La recherche d'images similaires
- La segmentation d'images: regroupement en pixel similaires !

Les illustrations 1 et 2 donnent un exemple de regroupement d'images en catégories, en fonction de leur similarité. La similarité est calculer en fonction de l'allongement (rapport des axes) et du périmètre.

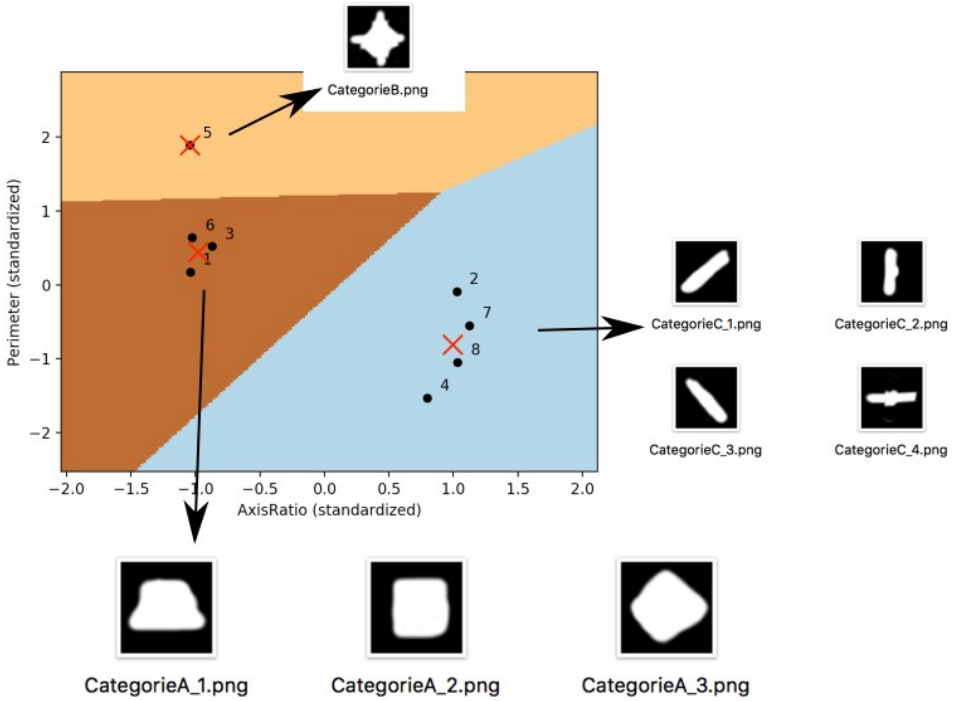

Illustration 2: Images associées aux catégories

Le script [tuto_kmeans] donne les catégories après analyse du tableau de caractéristiques [properties.csv] (la colonne « Id » désigne l'identifiant de l'image). Par la suite, nous allons découvrir deux techniques de classification non-supervisée : les « k-means » (ou k-moyennes) et le « gaussian mixture model » (ou modèle de mélange gaussien), sachant qu'il en existe beaucoup d'autres (voir illustration 3).

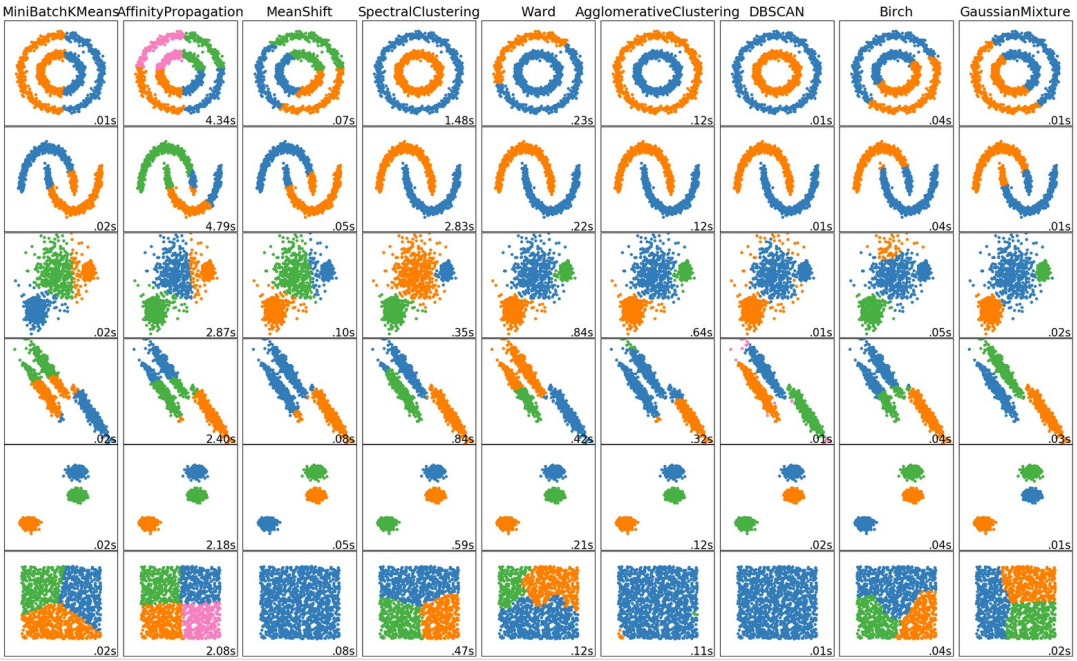

Illustration 3 Il existe beaucoup d'autres techniques en fonction de la structure des nuages. On pourra consulter [http://scikit-learn.org/stable/modules/clustering.html] et les exemples fournis.

3 K-means

A partir d'un ensemble de points, l'algorithme des « k-means » (ou k-moyennes) détermine, pour un nombre de classes k fixé, une répartition des points qui minimise un critère E appelé *inertie* ou variance *intra-classe :*

$$\bar{x}_1, \ldots, \bar{x}_k = argmin_{\bar{x}_1, \ldots, \bar{x}_k} E = \sum_{i=1}^{K} \sum_{x_j \in C_i} d(x_j, \bar{x}_i)$$

où x_j sont les points associés au cluster C_i (centre $\bar{x}_i$). On retient le regroupement (clustering) minimisant E, en testant différentes initialisations des centres (conduisant aux centres $\bar{x}_1, \ldots, \bar{x}_k$). d(.,.) représente la distance entre deux points (e.g. distance euclidienne). Les deux paramètres initiaux sont :

- Le nombre k de classes attendues
- Le nombre de tirages aléatoires de la position initiale des centres (choisis parmi les données)

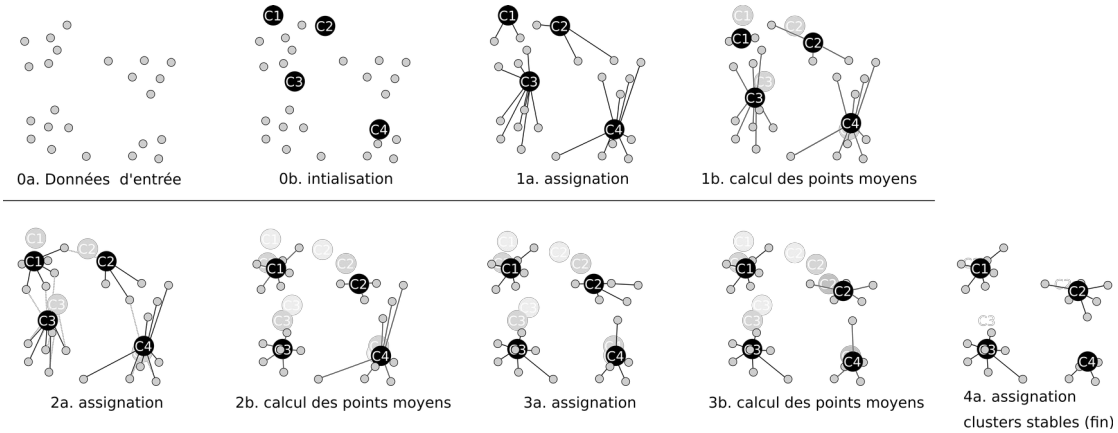

Illustration 4: K-Means: Exemple formation de 4 regroupements pour une initialisation donnée. [Source : wikipédia]

Soit le tableau suivant, associé à différents objets (A à E), chacun étant caractérisé pour une valeur.

id	Descripteur
A	1
B	7
C	3
D	6
E	2
F	5
G	8

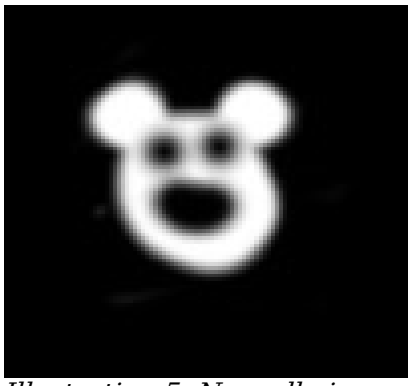

Illustration 5: Nouvelle image

Question : Que vaut E pour une partition C1=(A,C,E,F) et C2=(B,D,G) ?

Question : Que vaut E pour une partition C1=(A,C,E) et C2=(F,B,D,G) ?

Question : Quelle est la meilleur partition ? Vérifier votre résultat utilisant [sklearn.cluster.KMeans].

Question : En reprenant l'exemple considéré (avec les images) et en adaptant le script fourni, si l'on se limite à 2 catégories, quelles sont-elles ?

Question : En adaptant le script fourni, si l'on se limite à 4 catégories, quelles sont-elles ?

Question : Appliquer les « k-means » sans normalisation (3 classes): quelles sont les catégories ?

Question : Soit la nouvelle image associée à l'illustration 4. Son allongement a été mesuré et vaut **1.14**. Son périmètre vaut **297.90**. En utilisant le script, déterminer à quelles images cette nouvelle image est associée. Pour cela on utilisera la méthode [**predict**]. Attention : **cette nouvelle image ne sera pas utilisée pour recréer un nouveau regroupement → elle ne sera pas intégrée au données associées au fichier [properties.csv] !!!**

4 Gaussian Mixture Model

Cette approche consiste à modéliser chaque cluster par une densité de probabilité (loi normale) : chaque point est attaché à un cluster en fonction de la probabilité d'appartenance à celui-ci.

On fait l'hypothèse que l'on peut modéliser la distribution des données comme une somme pondérée de K gaussiennes. Pour une donnée [x] (vecteur ou scalaire), on suppose que la probabilité sera :

$p(x)=\sum_{i=1}^{K} \pi_i p_{\mu_i,\sigma_i}(x)$, où K est le nombre de classes, μ_i, σ_i sont la moyenne (vecteur de taille D) et la matrice de variance-covariance (matrice de taille DxD). D est la dimension de l'espace (2 dans notre exemple). Le terme π_i est la proportion de la classe i dans le mélange.

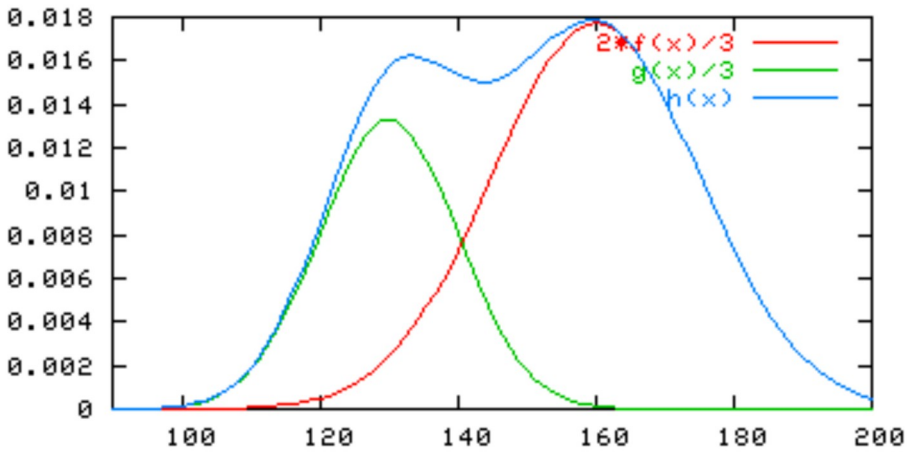

*Illustration 6: Exemple de mélange: h(x)=2/3*f(x)+1/3*g(x)*

Travail à faire : Ecrivez le script permettant de regrouper les images de manière non supervisée en utilisant la mixture de gaussiennes (classe [sklearn.mixture.GaussianMixture]).

Question : Quels sont les regroupements (hypothèse 3 groupes) ?

Question : En consultant les attributs, quelle est la proportion de chaque classe dans le mélange ?

Question : En consultant les attributs (voir documentation en ligne), quelles sont les propriétés (moyenne, matrice de variance-covariance) de la loi normale estimée et associée au groupe de l'image 7 ?