EE-311—Apprentissage et intelligence artificielle

6. Clustering, k plus proches voisins

Michael Liebling

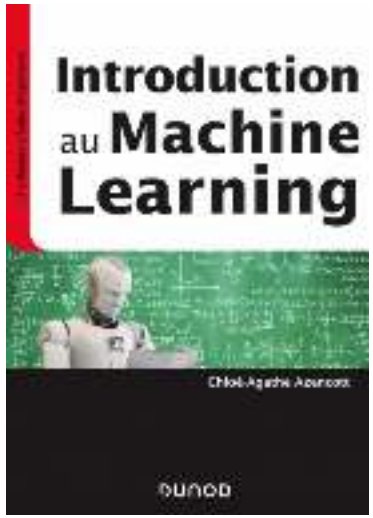
https://moodle.epfl.ch/course/view.php?id=16090

28 mars 2025 (compilé le 25 mars 2025)

idiap RESEARCH INSTITUTE

EPFL

Ouvrage de référence et source



Ces transparents sont basés en grande partie sur le texte de Chloé-Agathe Azencott "Introduction au Machine Learning", Dunod, 2019
ISBN 978-210-080153-4

L'auteure a mis le texte (sans les exercices) à disposition ici : http://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf

Avertissement : Bien que ces transparents partagent la notation mathématique, la structure de l'exposition (en partie), et certains exemples avec le livre, ils ne constituent qu'un complément et non un remplacement ou une source unique pour la couverture des matières du cours. À ce titre, ces transparents ne se substituent pas au texte.

Motivation

Comment étudier des données non étiquetées ?

Si la réduction de dimension nous permet de visualiser les données (en 2D, 3D), les méthodes de *partitionnement de données* **(= clustering)** nous permettent d'aller beaucoup plus loin :

Le clustering permet de séparer les données en sous-groupes homogènes, appelés clusters, qui partagent des caractéristiques communes.

Contenu

- Expliquer l'intérêt d'un algorithme de clustering
- Évaluer le résultat d'un algorithme de clustering
- Décrire les implémentations de
  - clustering hiérarchique
  - clustering par la méthode des k moyennes
  - clustering par densité.

Partitionnement (Clustering)

Définition : on appelle *partitionnement* ou *clustering* un problème d'apprentissage non supervisé pouvant être formalisé comme la recherche d'une partition $\bigcup_{k=1}^{K} \mathscr{C}_k$ des n observations $\{\vec{x}^i\}_{i=1,\dots,n}$. Cette partition doit être pertinente au vu d'un ou plusieurs critères à préciser.

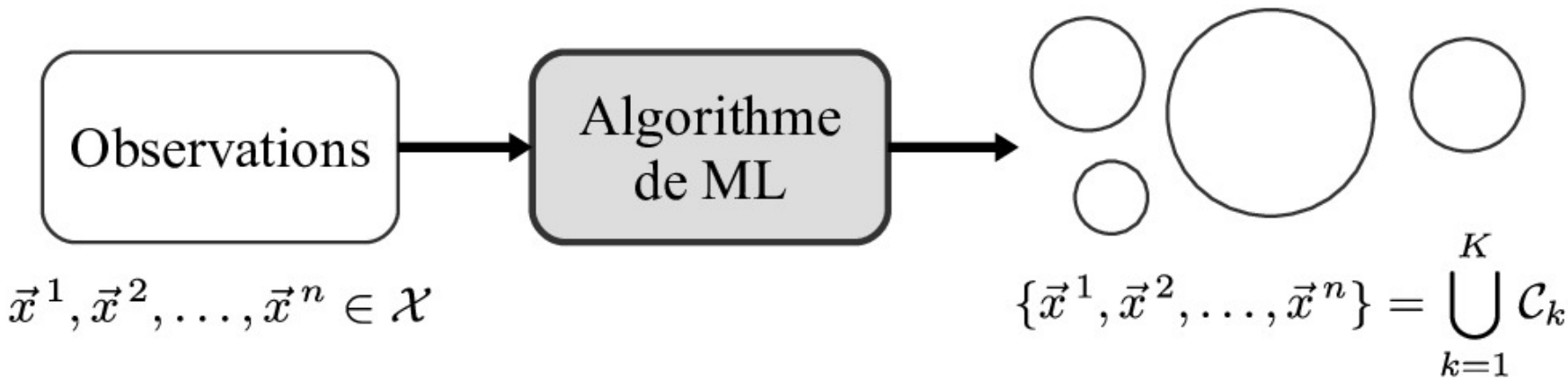


FIGURE 1.3 – Partitionnement des données, ou clustering.

Azencott

Pourquoi partitionner ses données

Analyse exploratoire sur des données non étiquetées par identification :

- de groupes d'utilisateurs qui ont des comportements similaires (segmentation de marché)
- de communautés sur un réseau social
- de motifs récurrents dans des transactions financières
- de pixels appartenant à un même objet dans une image (segmentation d'image)
- de patients dont la maladie s'explique par un même profil génétique

Visualisation :

- représenter un seul exemple représentatif par cluster

Étiquetage rapide lorsque difficile ou coûteux :

- transfert des propriétés (e.g. étiquette) que l'ont sait vraies de l'un des éléments de ce cluster à toutes les observations du même cluster

Exemple : clustering de textes

Problème : assigner un sujet a 600 articles de journal (*sport*, *culture*, *politique*, *santé*, etc.)

Lecture humaine serait fastidieuse et sujette à des erreurs d'inattention.

Solution (moins coûteuse et potentiellement plus efficace) :

1. utiliser un algorithme de clustering pour regrouper automatiquement les documents par sujet (sans que celui-ci soit connu pour autant) sur la base, par exemple, de mots fréquents qu'ils ont en commun
2. recourir à un intervenant humain pour assigner un sujet à chaque cluster en lisant uniquement un ou deux des documents qu'il contient

Notation

Jeu de données non étiqueté

$$\mathscr{D} = \left\{\vec{x}^1, \vec{x}^2, \ldots, \vec{x}^n\right\}$$

de n points d'un espace $\mathscr{X}$ partitionné en K clusters

$$\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_K.$$

Distance sur $\mathscr{X}$: d

Indice du cluster auquel $\vec{x}$ a été assigné : $k(\vec{x})$

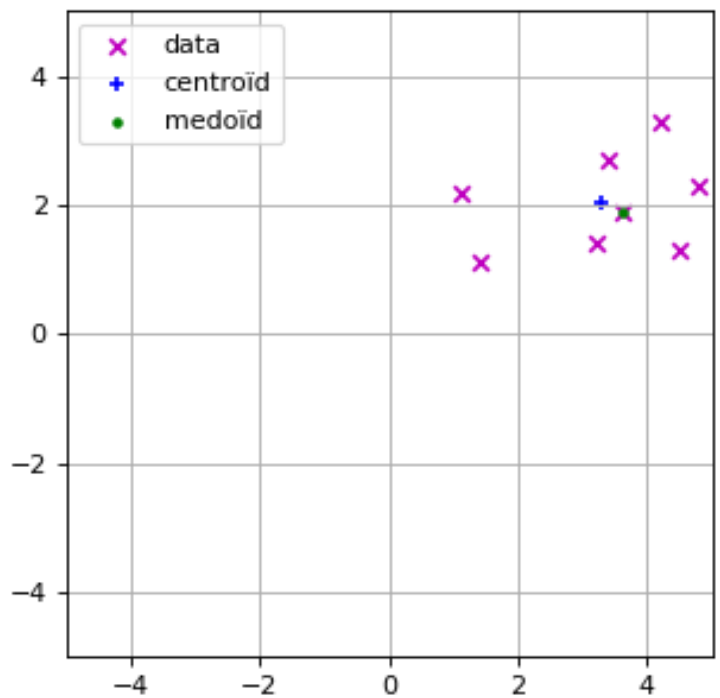
Centroïde et médoïde

Définition 12.1 (Centroïde et médoïde) on appelle *centroïde* du cluster $\mathscr{C}$ le point défini par

$$\vec{\mu}_{\mathscr{C}} = \frac{1}{|\mathscr{C}|} \sum_{\vec{x} \in \mathscr{C}} \vec{x}.$$

Le médoïde est le point du cluster le plus proche du centroïde (il peut ne pas être unique, auquel cas il sera choisi arbitrairement). Il sert de représentant du cluster :

$$\vec{m}_{\mathscr{C}} = \underset{\vec{x} \in \mathscr{C}}{\operatorname{argmin}}\, d(\vec{x}, \vec{\mu}_{\mathscr{C}})$$



Note :

- le centroïde est le barycentre d'un cluster

Homogénéité

Définition 12.2 (Homogénéité) On appelle *homogénéité du cluster* $\mathscr{C}_k$ (*tightness*), la moyenne des distances des observations de ce cluster à son centroïde :

$$T_k = \frac{1}{|\mathscr{C}_k|} \sum_{\vec{x} \in \mathscr{C}_k} d(\vec{x}, \vec{\mu}_k)$$

avec $\vec{\mu}_k$ le centroïde de $\mathscr{C}_k$. *Note :* Plus T_k est petit plus on dit que le cluster est homogène.

L'*homogénéité globale* d'un clustering de D se calcule comme la moyenne des homogénéités des clusters :

$$T = \frac{1}{K} \sum_{k=1}^{K} T_k$$

Exemples : homogénéité

La notion d'homogénéité traduit le fait que des observations proches appartiennent au même cluster.

petit T
homogénéité importante

grand T
peu homogène

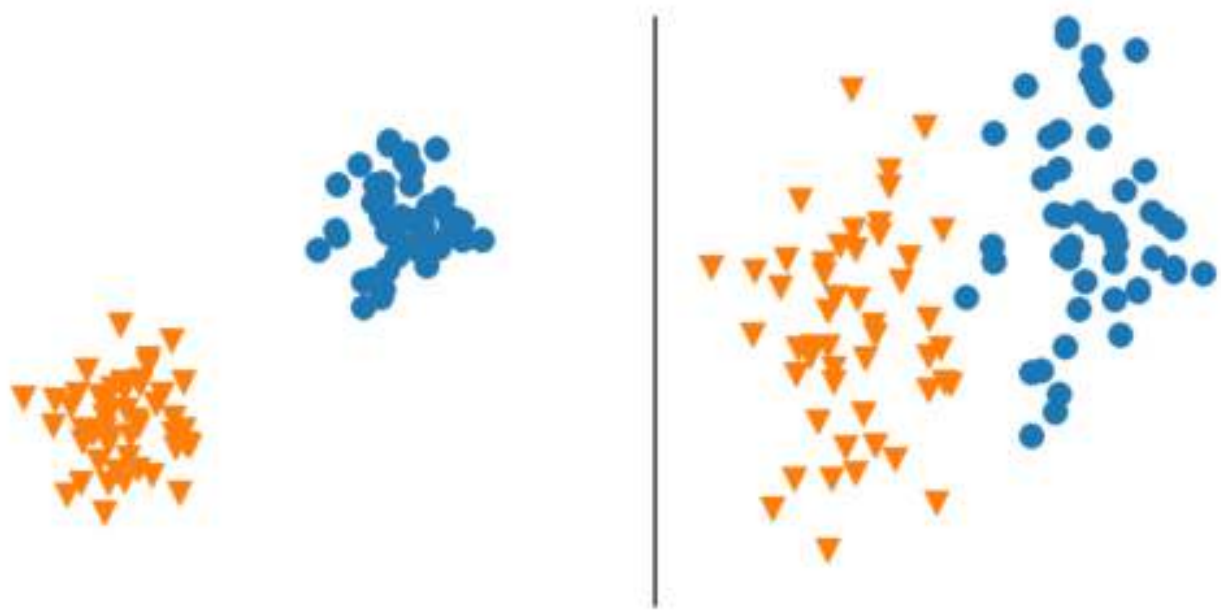


FIGURE 12.1 – Les deux clusters représentés sur le panneau de gauche sont homogènes, resserrés sur eux-mêmes : ils sont composés de points proches les uns des autres. À l'inverse, les deux clusters représentés sur le panneau de droite sont moins homogènes.

Azencott

Séparabilité

Définition 12.3 (Séparabilité) On appelle *séparabilité* des clusters $\mathscr{C}_k$ et $\mathscr{C}_\ell$ la distance entre leurs centroïdes :

$$S_{k\ell} = d(\vec{\mu}_k, \vec{\mu}_\ell).$$

La *séparabilité globale* d'un clustering de $\mathscr{D}$ se calcule comme la moyenne des séparabilités des clusters deux à deux :

$$S = \frac{2}{K(K-1)} \sum_{k=1}^{K} \sum_{\ell=k+1}^{K} S_{k\ell}$$

Exemples : séparabilité

La séparabilité quantifie à quel point les clusters sont distants les uns des autres.

High Separability S
grande séparabilité

Low Separability S
faible séparabilité

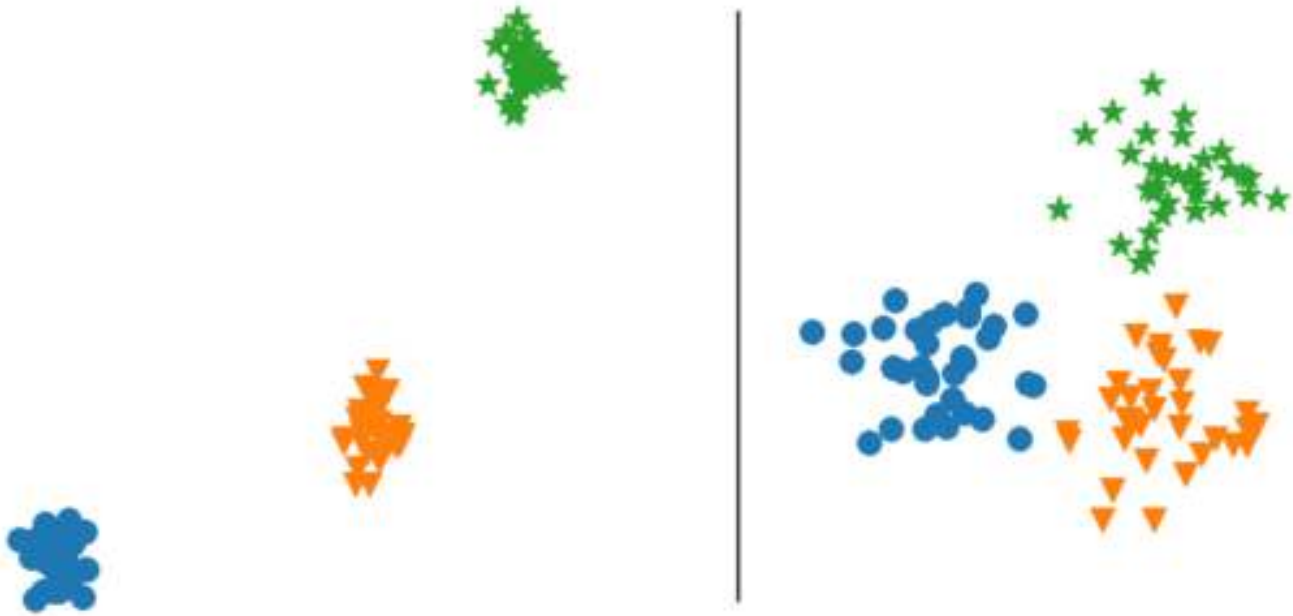


FIGURE 12.2 – Les trois clusters représentés sur le panneau de gauche sont bien séparés, contrairement à ceux représentés sur le panneau de droite qui sont proches les uns des autres.

Azencott

Indice de Davies-Bouldin

Un bon clustering présente

- des critères de séparabilité élevés (S grand)
- bonne homogénéité (T petit)

Y'a-t-il moyen de combiner les deux notions ?

Définition 12.4 (Indice de Davies-Bouldin) On appelle *indice de Davies-Bouldin* du cluster $\mathscr{C}_k$ la valeur

$$\underbrace{0}_{\text{le pire cluster (proche de } \mathscr{C}_k \text{ et peu homogène) est néanmoins très loin et très homogène}} \leq D_k = \max_{\ell \neq k} \frac{T_k + T_\ell}{S_{k\ell}} < \underbrace{\infty}_{\text{il y a un cluster voisin très proche et très peu homogène}}$$

L'*indice de Davies-Bouldin* global d'un clustering de $\mathscr{D}$ se calcule comme la moyenne des indices de Davies-Bouldin des clusters :

$$\underbrace{0}_{\text{clustering idéal}} \leq D = \frac{1}{K} \sum_{k=1}^{K} D_k < \underbrace{\infty}_{\text{pire clustering}}$$

Coefficient de silhouette

Définition 12.5 (Coefficient de silhouette) On appelle *coefficient de silhouette* de l'observation $\vec{x} \in \mathscr{D}$ la valeur

$$s(\vec{x}) = \frac{b(\vec{x}) - a(\vec{x})}{\max(a(\vec{x}), b(\vec{x}))}$$

où $a(\vec{x})$ est la distance moyenne de $\vec{x}$ à tous les autres éléments du cluster auquel il appartient et $b(\vec{x})$ est la plus petite valeur que pourrait prendre $a(\vec{x})$ si $\vec{x}$ appartenait à un autre cluster :

$$a(\vec{x}) = \frac{1}{|\mathscr{C}_{k(\vec{x})}| - 1} \sum_{\vec{u} \in \mathscr{C}_{k(\vec{x})}, \vec{u} \neq \vec{x}} d(\vec{u}, \vec{x})$$

$$b(\vec{x}) = \min_{\ell \neq k(\vec{x})} \frac{1}{|\mathscr{C}_\ell|} \sum_{\vec{u} \in \mathscr{C}_\ell} d(\vec{u}, \vec{x}).$$

Le *coefficient de silhouette global* du clustering est son coefficient de silhouette moyen :

$$s = \frac{1}{n} \sum_{i=1}^{n} s\left(\vec{x}^i\right).$$

Coefficients de silhouette (suite)

Le coefficient de silhouette de $\vec{x}$ est d'autant plus proche de 1 que son assignation au cluster $\mathscr{C}_{k(\vec{x})}$ est satisfaisante.

Si le coefficient est proche de -1, l'assignation au cluster est erronée.

Si le coefficient est 0, l'assignation à l'un ou l'autre cluster est indifférente, il n'y aurait pas de changement significatif si $\vec{x}$ était assigné à un autre cluster.

Stabilité des clusters

Clustering stable : le partitionnement ne change pas si

- on supprime quelques éléments
- on perturbe quelques éléments
- on initialise l'algorithme de partitionnement de manière différente

Ce critère peut être utilisé pour choisir les hyperparamètres de l'algorithme : si on obtient des clusters très différents pour différentes initialisations de l'algorithme de partitionnement, cela peut indiquer que les hyperparamètres sont mal choisis.

Connaissances expert

Si on dispose d'un jeu de données partiellement étiqueté par des classes auquel on applique une algorithme de clustering : les indices des clusters ne correspondent pas forcément à ceux des classes (première classe, seconde classe, etc.) $\Rightarrow$ une évaluation de la qualité de clustering ne devra donc pas pénaliser l'assignation à une partition d'indice différent que celui de la classe ; seul les groupements sont à prendre en compte.

Connaissances expert

Définition 12.6 (Indice de Rand) On appelle *indice de Rand* la proportion de paires d'observations qui sont soit :

- de la même classe et dans le même cluster
- de classe différente et dans deux clusters différents :

$$\mathrm{RI} = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{\ell=i+1}^{n} \delta\left(k\left(\vec{x}^i\right) = k\left(\vec{x}^\ell\right)\right) \delta\left(y^i = y^\ell\right) + \delta\left(k\left(\vec{x}^i\right) \neq k\left(\vec{x}^\ell\right)\right) \delta\left(y^i \neq y^\ell\right)$$

Connaissances expert : ontologies, cohérence d'un clustering par analyse d'enrichissement

ontologie : une classification d'objets en catégories décrites par un vocabulaire commun et organisées de manière hiérarchique (par exemple, des gènes en bioinformatique)

Analyse d'enrichissement : évaluer la cohérence d'un clustering en vérifiant si, à l'intérieur d'un cluster, le nombre d'objets d'une catégorie de l'ontologie s'écarte de ce à quoi on pourrait s'attendre si leur distribution était aléatoire (distribution hypergéométrique). Pour un cluster $\mathscr{C}_k$, une catégorie $\mathscr{G}$ et un seuil $t \in \mathbb{N}$ on calcule :

$$\mathbb{P}\left[|\mathscr{G} \cap \mathscr{C}_k| \geq t\right] = 1 - \sum_{s=0}^{t-1} \frac{\binom{|\mathscr{G}|}{s}\binom{n-|\mathscr{G}|}{|\mathscr{C}_k|-s}}{\binom{n}{|\mathscr{C}_k|}}$$

avec $\frac{\binom{|\mathscr{G}|}{s}\binom{n-|\mathscr{G}|}{|\mathscr{C}_k|-s}}{\binom{n}{|\mathscr{C}_k|}}$ la probabilité que, lorsqu'on tire $|\mathscr{C}_k|$ éléments parmi n, s d'entre eux appartiennent à $\mathscr{G}$

Algorithmes de clustering

- Les algorithmes de clustering cherchent à optimiser les critères d'homogénéité et de séparabilité.

- Pas d'approche exacte, seulement des manières approchées.

- Trois principales familles d'algorithmes de clustering :

  - clustering hiérarchique,
  - clustering par centroïdes,
  - clustering par densité.

Clustering hiérarchique

Le *clustering hiérarchique* forme des clusters séparés par récurrence : il s'agit de partitionner les données pour toutes les échelles possibles de taille de partition, dans une hiérarchie à plusieurs niveaux.

Dendrogramme Le résultat d'un clustering hiérarchique peut se visualiser sous la forme d'un dendrogramme. Il s'agit d'un arbre dont les n feuilles correspondent chacune à une observation. Chaque nœud de l'arbre correspond à un cluster :

- la racine est un cluster contenant toutes les observations
- chaque feuille est un cluster contenant une observation
- les clusters ayant le même parent sont agglomérés en un seul cluster au niveau au-dessus
- un cluster est subdivisé en ses enfants au niveau au dessus

Exemple de dendrogramme

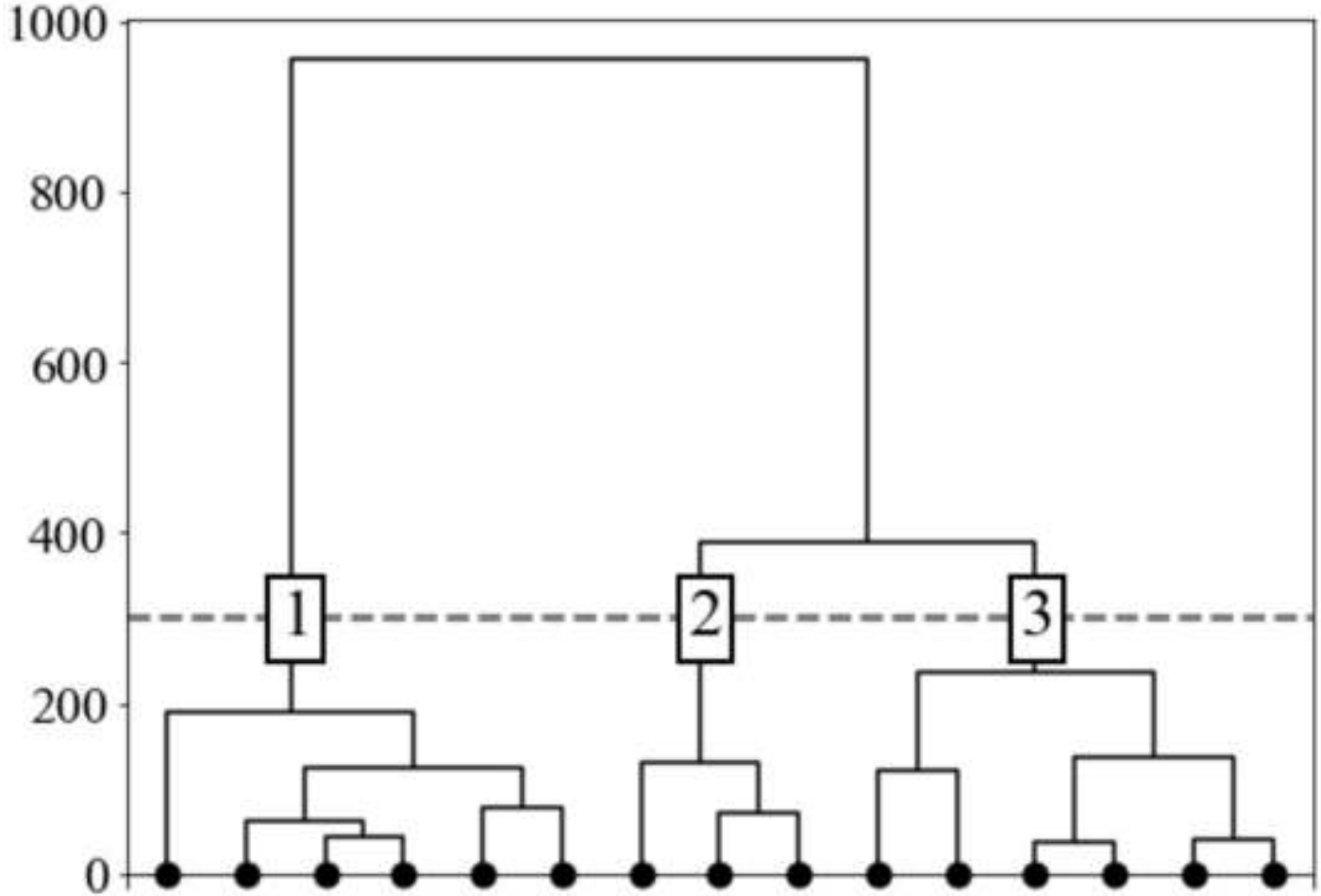


FIGURE 12.3 – Un exemple de dendrogramme. En coupant au niveau de la ligne en pointillés, on obtient 3 clusters. Chaque feuille de l'arbre (sur l'axe des abscisses) correspond à une observation.

Azencott

Types de méthodes de construction des clustering hiérarchique

Un clustering hiérarchique peut se construire par :

- *clustering agglomératif* (bottom-up clustering), commence par les feuilles avec chaque observation qui forme un cluster de taille 1. À chaque itération, on trouve et agglomère les deux clusters les plus proches jusqu'à ne plus avoir qu'un unique cluster contenant les n observations.
- *clustering divisif*, ou top-down clustering : initialise en considérant un seul cluster contenant toutes les observations. À chaque itération, on sépare un cluster en deux, jusqu'à ce que chaque cluster ne contienne plus qu'une seule observation.

Focus sur clustering agglomératif.
Question clé : comment trouver déterminer les deux clusters les plus proches à agglomérer ?
Réponse : définir une distance entre clusters, i.e. une fonction de lien (linkage). Se base sur une distance d sur $\mathcal{X}$.

Fonctions de lien qui visent à garantir la séparabilité des clusters

Définition 12.7 (Lien simple) On appelle *lien simple* ou *single linkage*, la distance entre deux clusters définie par :

$$d_{\text{simple}}\left(\mathscr{C}_k, \mathscr{C}_\ell\right) = \min_{(\vec{u},\vec{v})\in\mathscr{C}_k\times\mathscr{C}_\ell} d\left(\vec{u}, \vec{v}\right)$$

$\Rightarrow$ minimiser le lien simple revient à examiner la distance la plus proche entre éléments de toutes les paires de clusters et à agglomérer les deux clusters pour laquelle cette distance est la plus petite (minimisation d'une distance min)

Définition 12.8 (Lien complet) On appelle *lien complet* ou *complete linkage*, la distance entre deux clusters définie par :

$$d_{\text{complet}}\left(\mathscr{C}_k, \mathscr{C}_\ell\right) = \max_{(\vec{u},\vec{v})\in\mathscr{C}_k\times\mathscr{C}_\ell} d\left(\vec{u}, \vec{v}\right)$$

$\Rightarrow$ minimiser le lien complet agglomère les deux clusters qui ont tous leurs éléments proches (minimisation d'une distance max)

Fonctions de lien qui visent à garantir la séparabilité des clusters (suite)

Définition 12.9 (Lien moyen) On appelle *lien moyen* ou *average linkage* ou *Unweighted paired group method with arithmetic mean (UPGMA)*, la distance entre deux clusters définie par :

$$d_{\text{moyen}}\left(\mathscr{C}_k, \mathscr{C}_\ell\right) = \frac{1}{|\mathscr{C}_k|}\frac{1}{|\mathscr{C}_\ell|}\sum_{\vec{u}\in\mathscr{C}_k}\sum_{\vec{v}\in\mathscr{C}_\ell} d\left(\vec{u}, \vec{v}\right)$$

$\Rightarrow$ minimiser le lien moyen revient à agglomérer les deux clusters dont les éléments sont les plus proches en moyenne

Définition 12.10 (Lien centroïdal) On appelle *lien centroïdal* ou *centroid linkage*, la distance entre deux clusters définie par :

$$d_{\text{centroid}}\left(\mathscr{C}_k, \mathscr{C}_\ell\right) = d\left(\frac{1}{|\mathscr{C}_k|}\sum_{\vec{u}\in\mathscr{C}_k}\vec{u}, \frac{1}{|\mathscr{C}_\ell|}\sum_{\vec{v}\in\mathscr{C}_\ell}\vec{v}\right) = d\left(\vec{\mu}_k, \vec{\mu}_\ell\right)$$

$\Rightarrow$ minimiser le lien centroïdal agglomère les deux clusters qui ont leur deux éléments centroïdes le plus proches

Agglomération de cluster visant une bonne homogénéité

Favoriser l'homogénéité : on cherchera à identifier les deux clusters dont l'agglomération fournit le cluster qui minimise

$$T_k = \frac{1}{|\mathscr{C}_k|} \sum_{\vec{x} \in \mathscr{C}_k} d(\vec{x}, \vec{\mu}_k)$$

En particulier, dans le cas de la distance euclidienne, on a

$$T_k = \frac{1}{|\mathscr{C}_k|} \sum_{\vec{x} \in \mathscr{C}_k} \|\vec{x} - \vec{\mu}_k\|_2 \quad \rightarrow \text{Attention : distance (et pas } \|\cdot\|_2^2 \text{)}$$

Agglomération de cluster visant une bonne homogénéité (suite) : clustering de Ward

Définition 12.11 (Inertie) On appelle *variance intra-cluster* ou *inertie du cluster* $\mathscr{C}$ la valeur

$$\mathrm{Var}_{\mathrm{in}}(\mathscr{C}) = \frac{1}{|\mathscr{C}|} \sum_{\vec{x} \in \mathscr{C}} \|\vec{x} - \vec{\mu}\|_2^2$$

L'*inertie globale* d'un clustering de $\mathscr{D}$ est alors donnée par la somme des inerties des clusters :

$$V = \sum_{k=1}^{K} \frac{1}{|\mathscr{C}|} \sum_{\vec{x} \in \mathscr{C}_k} \|\vec{x} - \vec{\mu}_k\|_2^2$$

$\Rightarrow$ agglomérer les deux classes de sorte à minimiser la variance intra-cluster.

Choix du nombre de clusters et complexité algorithmique

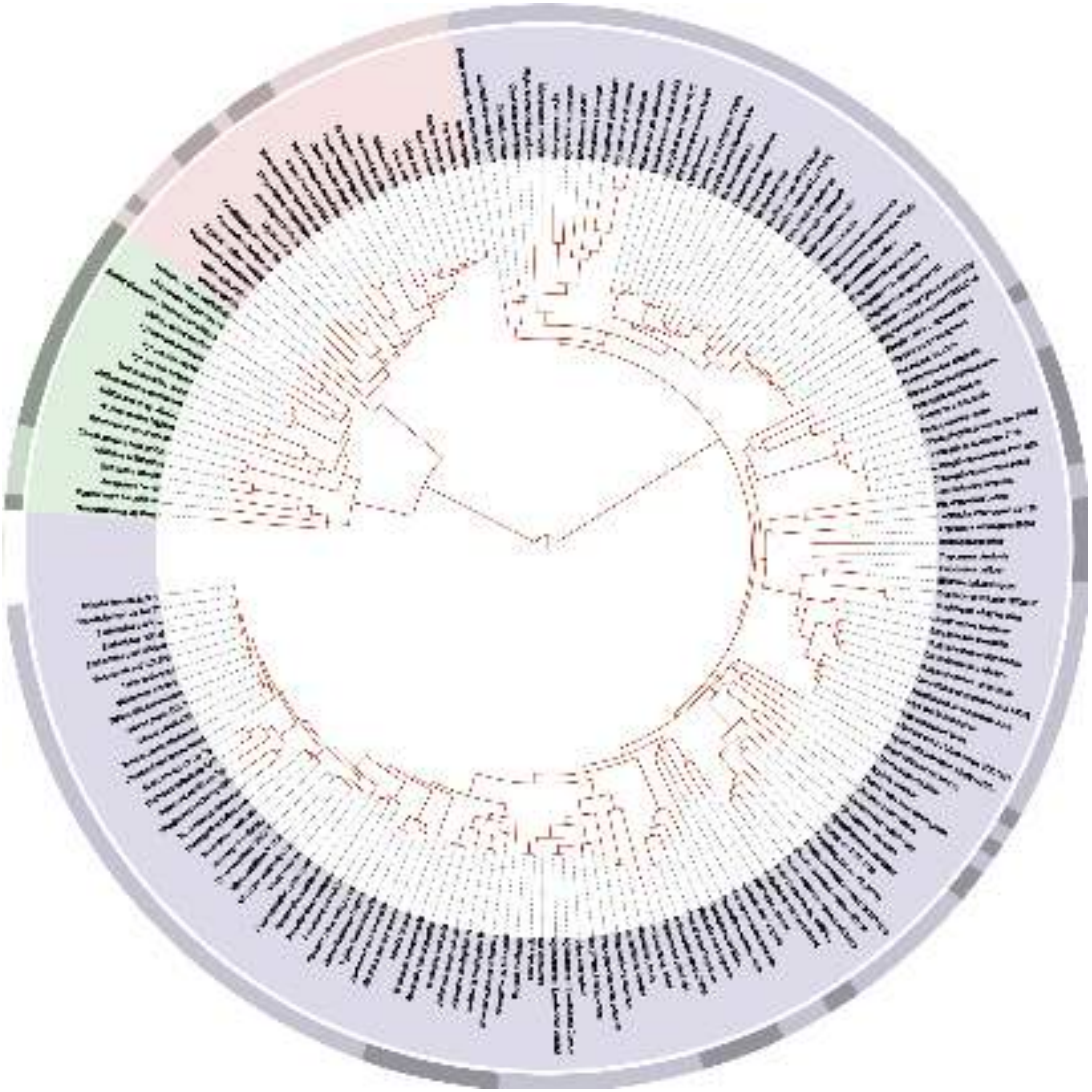
Clustering hiérarchique ne requiert pas de définir un nombre de classes par avance : toutes les possibilités peuvent être explorée (mais la décision est tout de même requise à un moment).

Organisation des clusters en dendrogramme (longueur d'une branche proportionnelle à la distance)

Complexité algorithmique du clustering hiérarchique est élevée : à chaque itération pour décider quels clusters regrouper, il faut calculer les distances deux à deux entre toutes les paires d'observations du jeu de données, pour une complexité en $\mathcal{O}(pn^2)$. Alternative : stocker ces distances en mémoire pour pouvoir les réutiliser (complexité quadratique en le nombre d'observations).

Le clustering hiérarchique est donc plus adapté aux jeux de données contenant peu d'échantillons.

David Hillis's 2008 plot of the tree of life, based on completely sequenced genomes



rose : eukaryotes (animaux, plantes, fungi)
bleu : bactéries
vert : archaea.

https://upload.wikimedia.org/wikipedia/commons/1/11/Tree_of_life_SVG.svg
Ivica Letunic : Iletunic. Retraced by Mariana Ruiz Villarreal : LadyofHats

Méthode des k-moyennes

Alternative à l'exploration de toutes les partitions possibles à des échelles différentes : se fixer un nombre K de clusters

La méthode des *k-moyennes* (k-means), Hugo Steinhaus (1957), cherche trouver l'affectation des observations à K clusters qui minimise la variance intra-cluster globale (favorisation des clusters homogènes) **pour un nombre de clusters K fixé** :

$$\underset{\mathscr{C}_1,\mathscr{C}_2,\ldots,\mathscr{C}_K}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{\vec{x}\in\mathscr{C}_k} \|\vec{x} - \vec{\mu}_k\|_2^2$$

Résolution exacte n'est pas possible
$\Rightarrow$ on utilise une heuristique : algorithme de Lloyd.

Méthode des k-moyennes : Algorithme de Lloyd (Stuart Lloyd, 1982)

Définition 12.12 (Algorithme de Lloyd) Etant données n observations dans $\mathbb{R}^p$ et un nombre K de clusters, l'*algorithme de Lloyd* procède de la manière suivante :

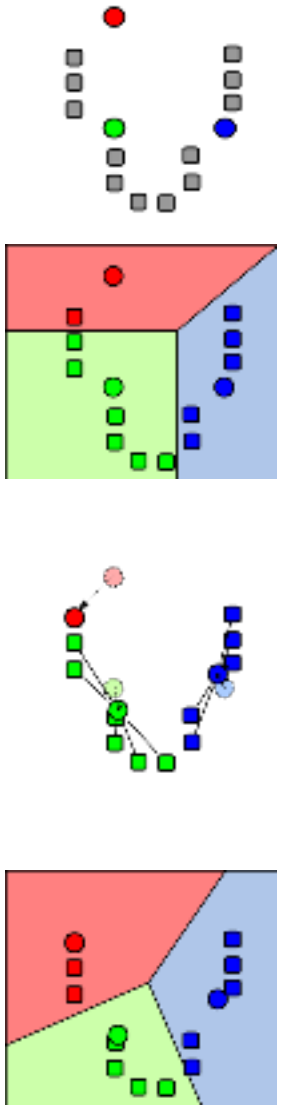
1. choisir K observations $\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_K$ parmi les n observations (centroïdes initiaux)
2. Affecter chaque observations $\vec{x}^i \in \mathscr{D}$ au centroïde dont elle est le plus proche

$$k\left(\vec{x}^i\right) = \underset{k=1,\ldots,K}{\operatorname{argmin}} \left\|\vec{x}^i - \vec{\mu}_k\right\|_2$$

3. Recalculer les centroïdes de chaque cluster :

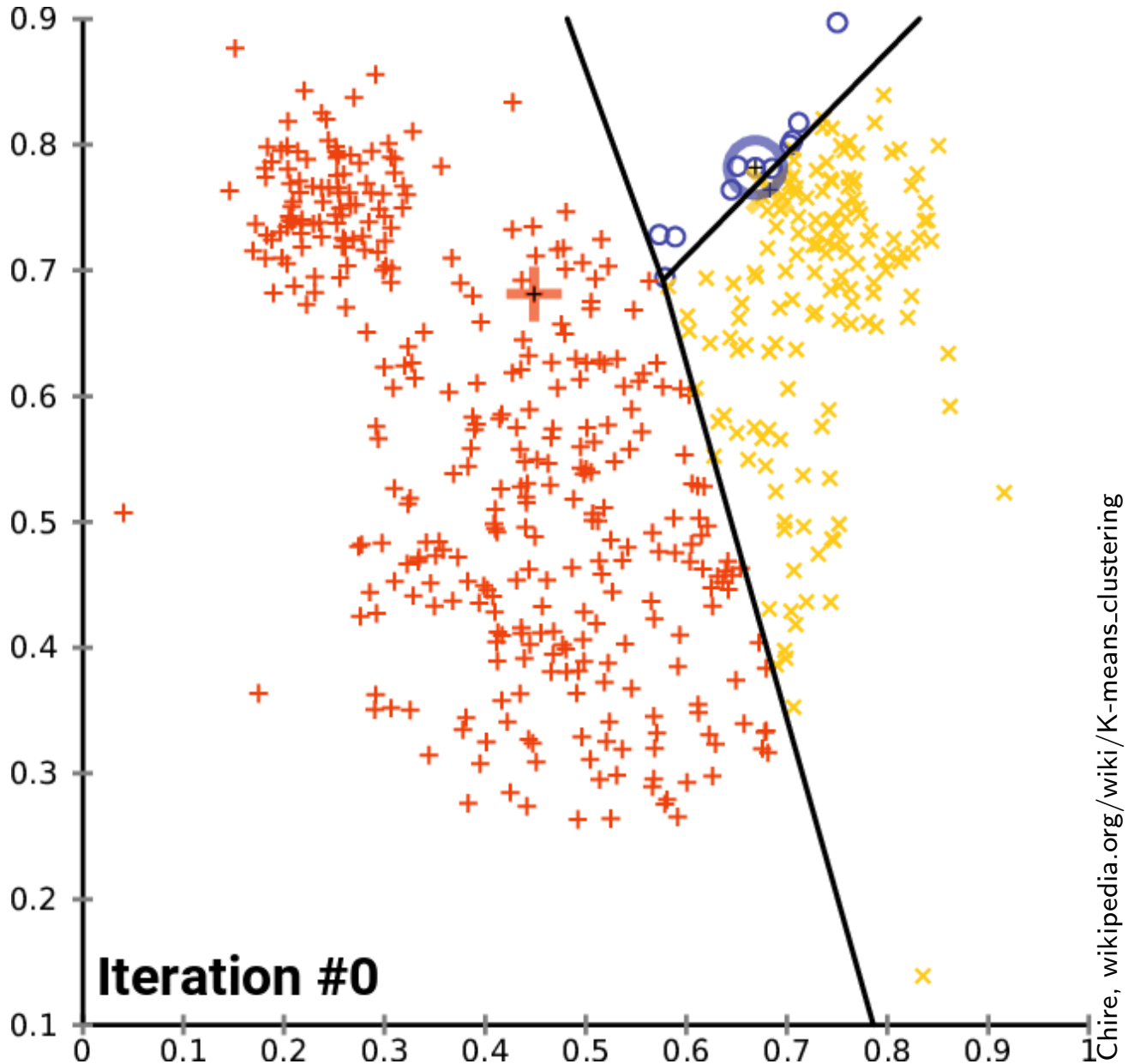
$$\vec{\mu}_k = \frac{1}{|\mathscr{C}_k|} \sum_{\vec{x}^i \in \mathscr{C}_k} \vec{x}^i$$

4. Répéter les opérations 2–3 jusqu'à convergence (lorsque les affectations ne changent plus)

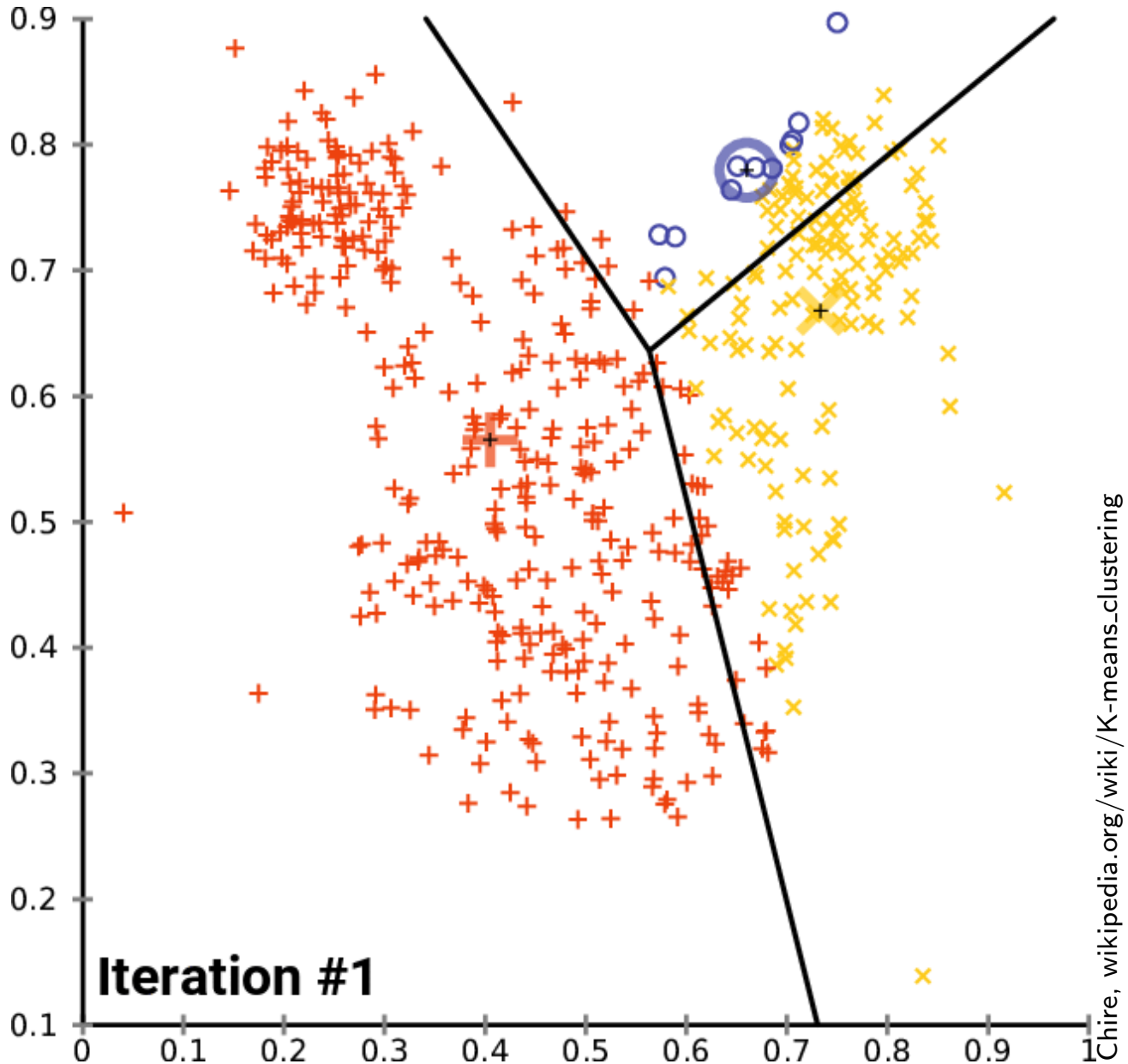


Weston.pace https ://en.wikipedia.org/wiki/K-means_clustering

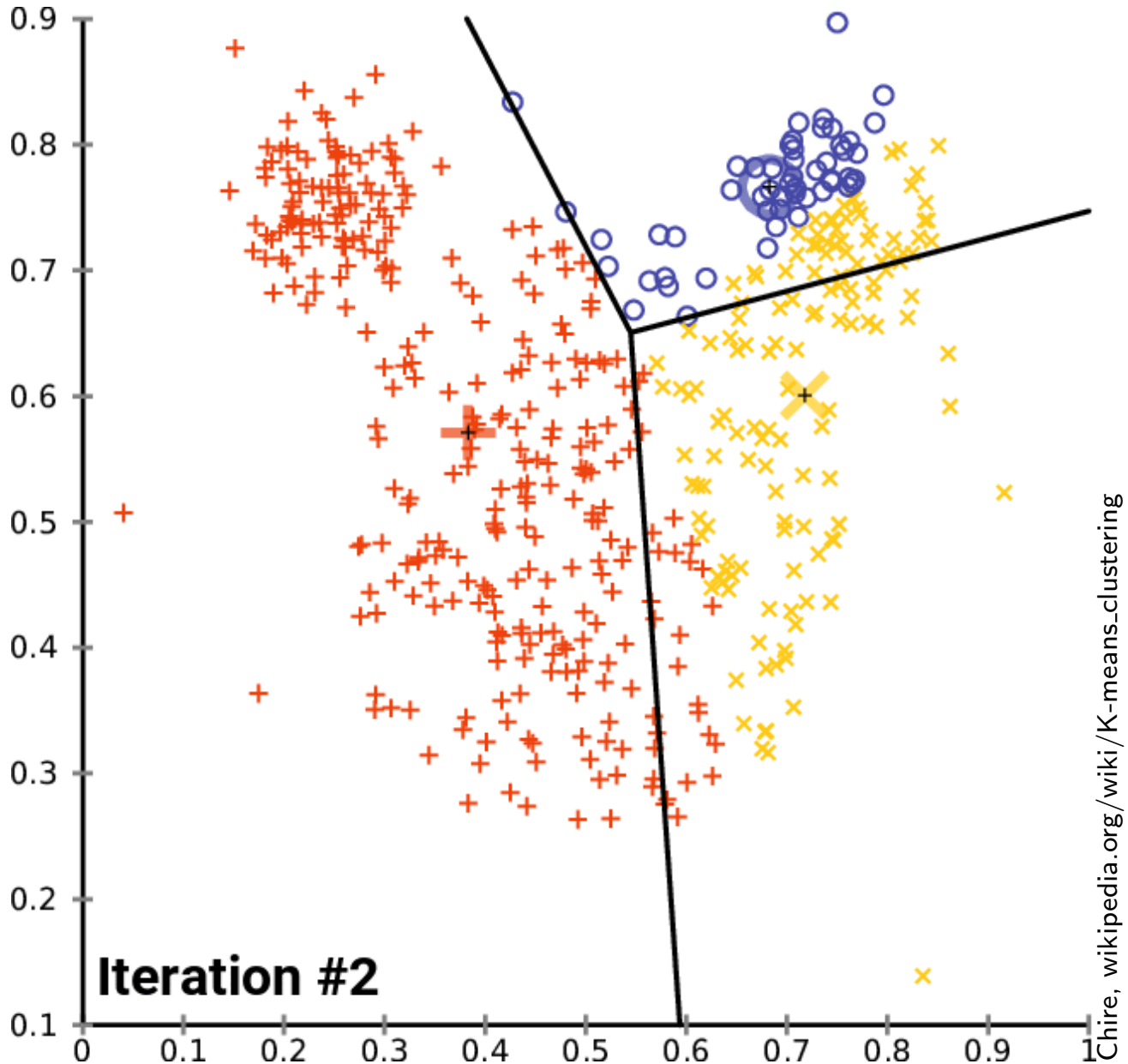
Exemple K-means (itération 00)



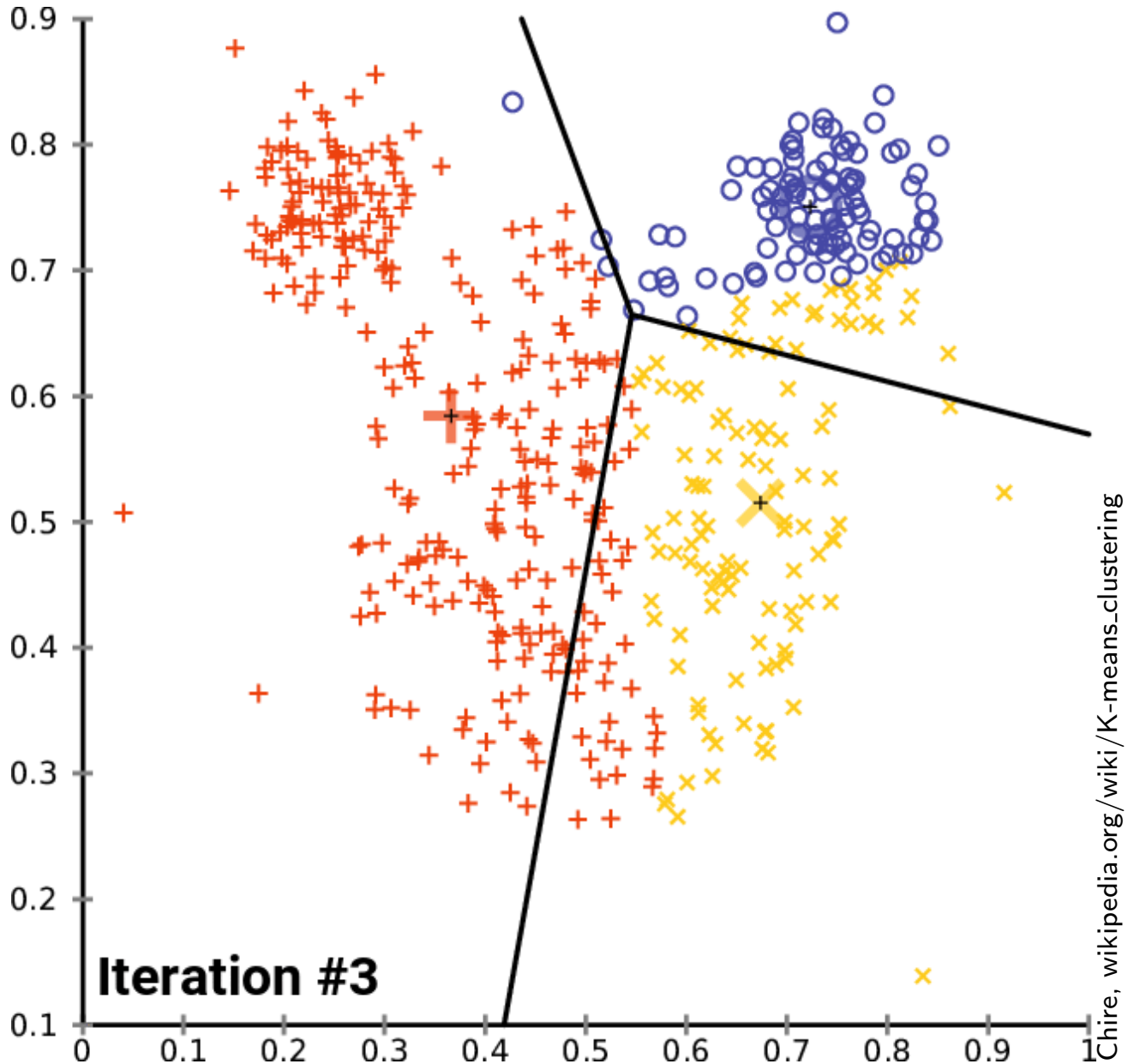
Exemple K-means (itération 01)



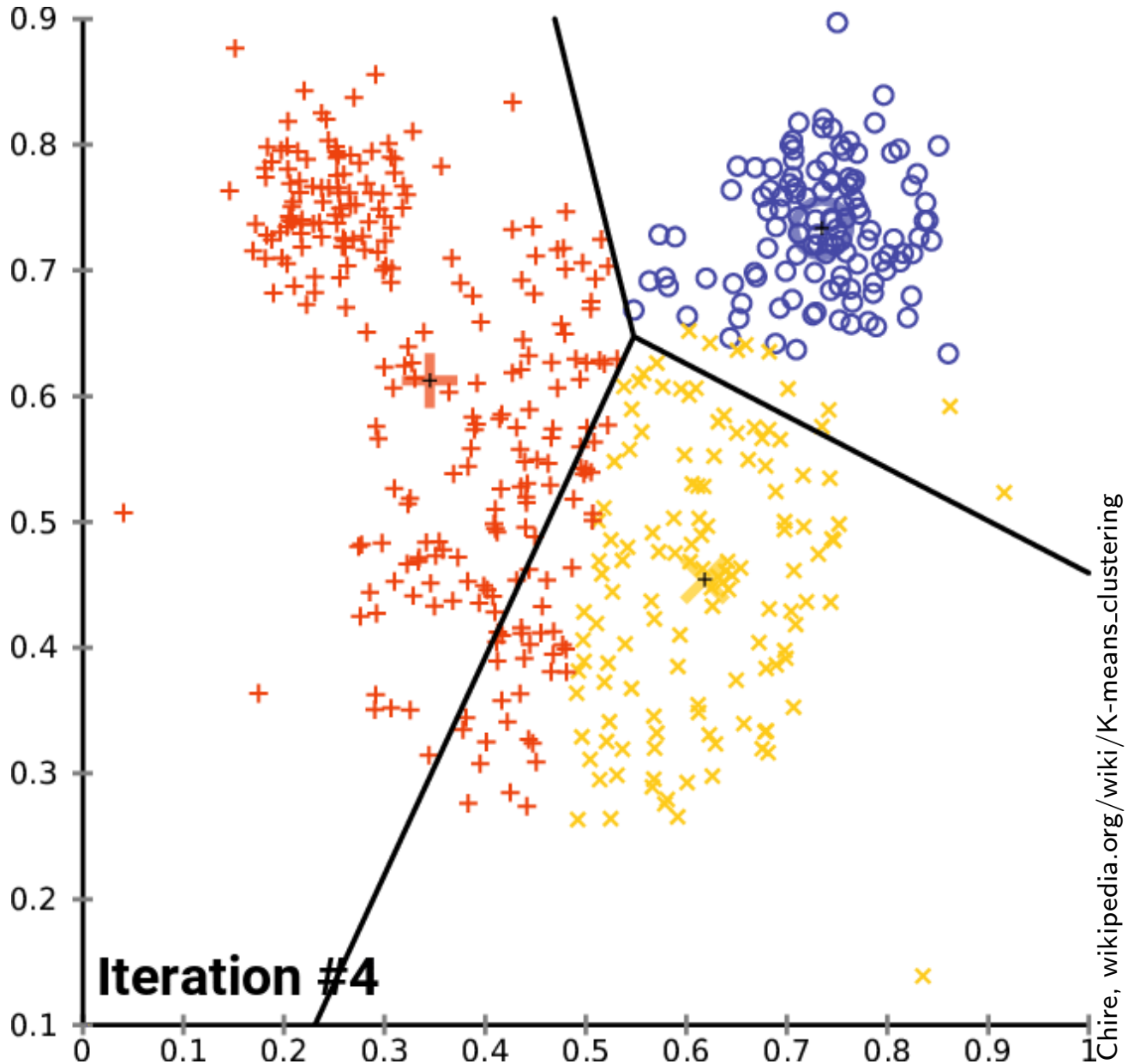
Exemple K-means (itération 02)



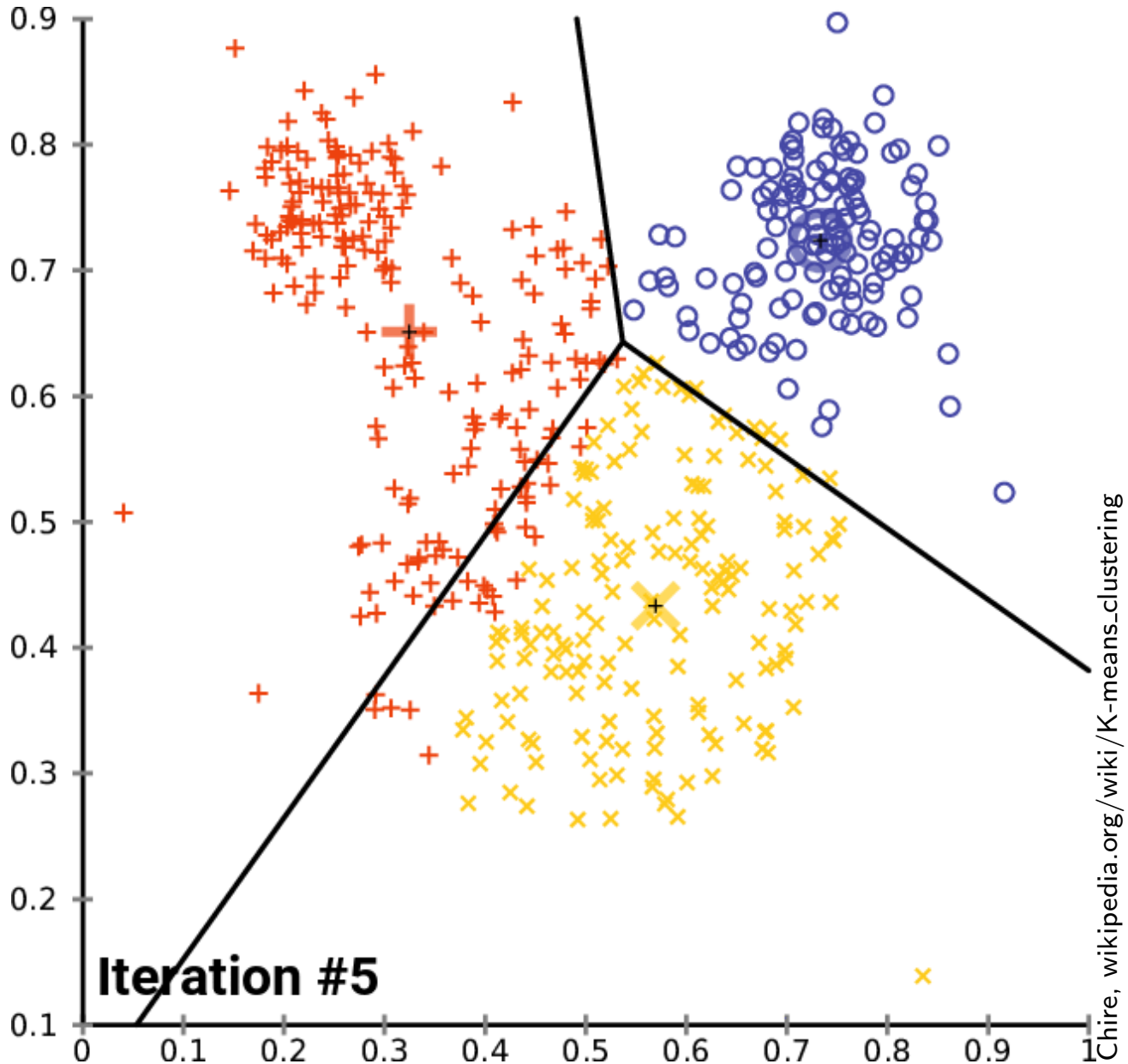
Exemple K-means (itération 03)



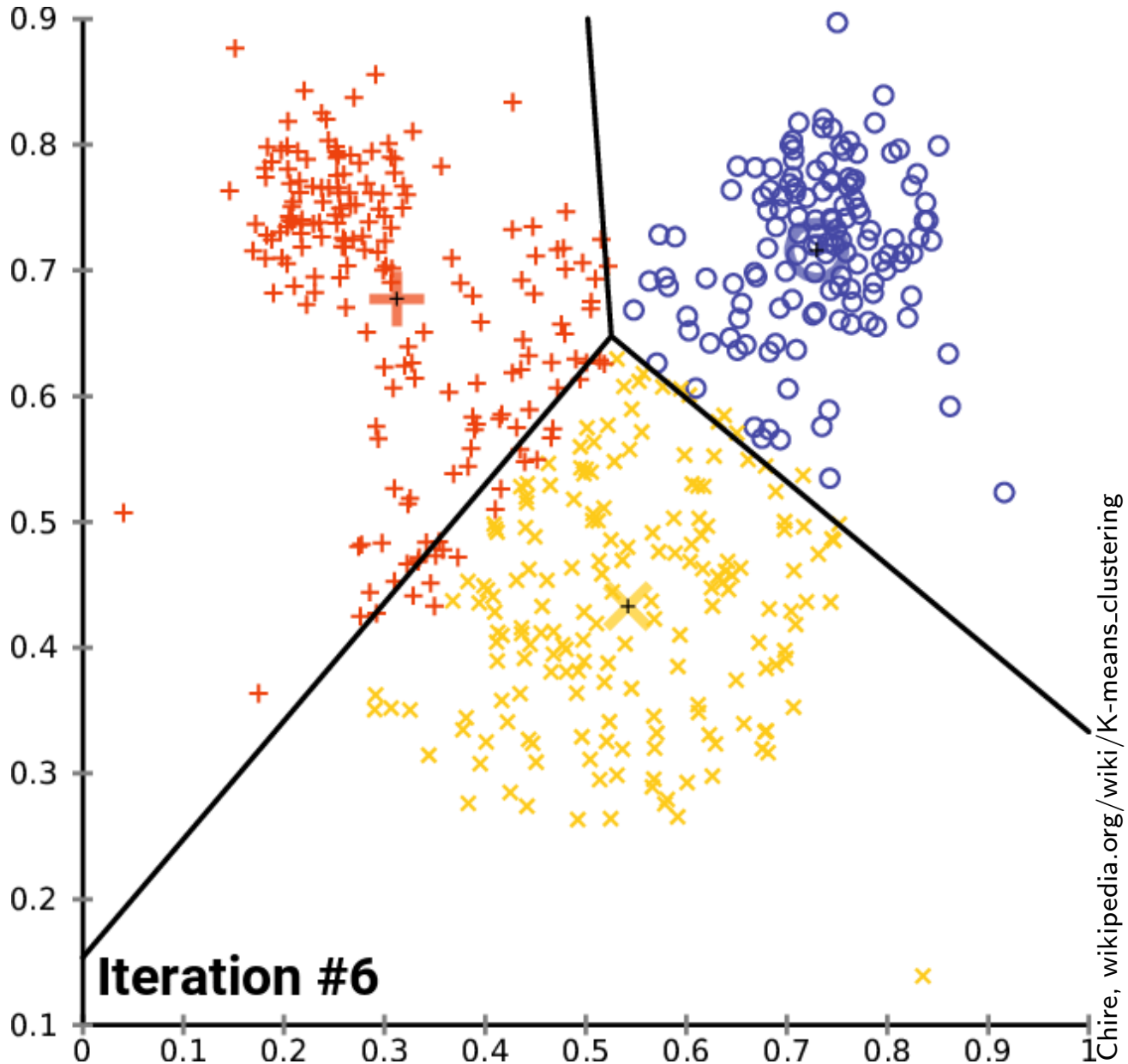
Exemple K-means (itération 04)



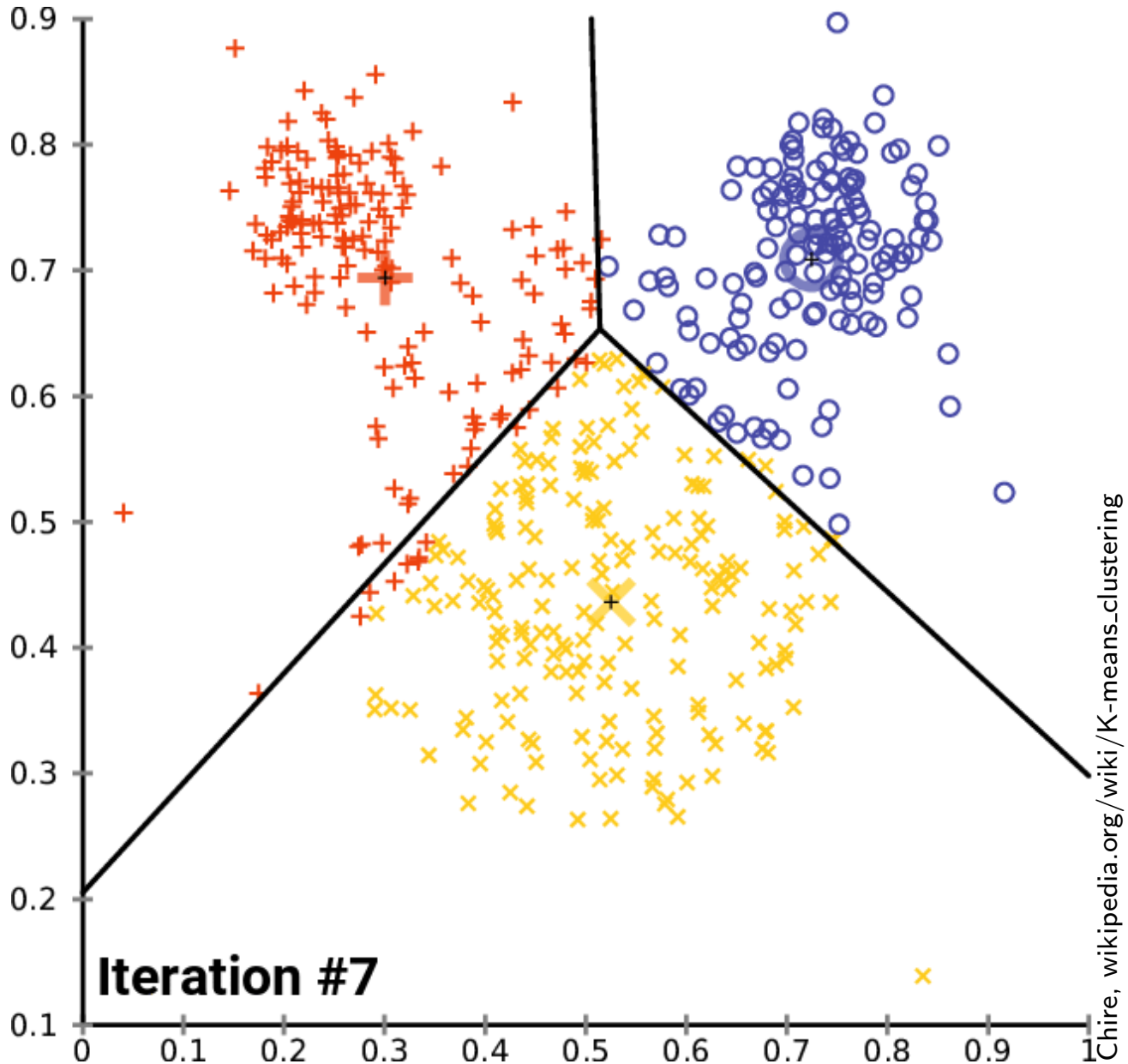
Exemple K-means (itération 05)



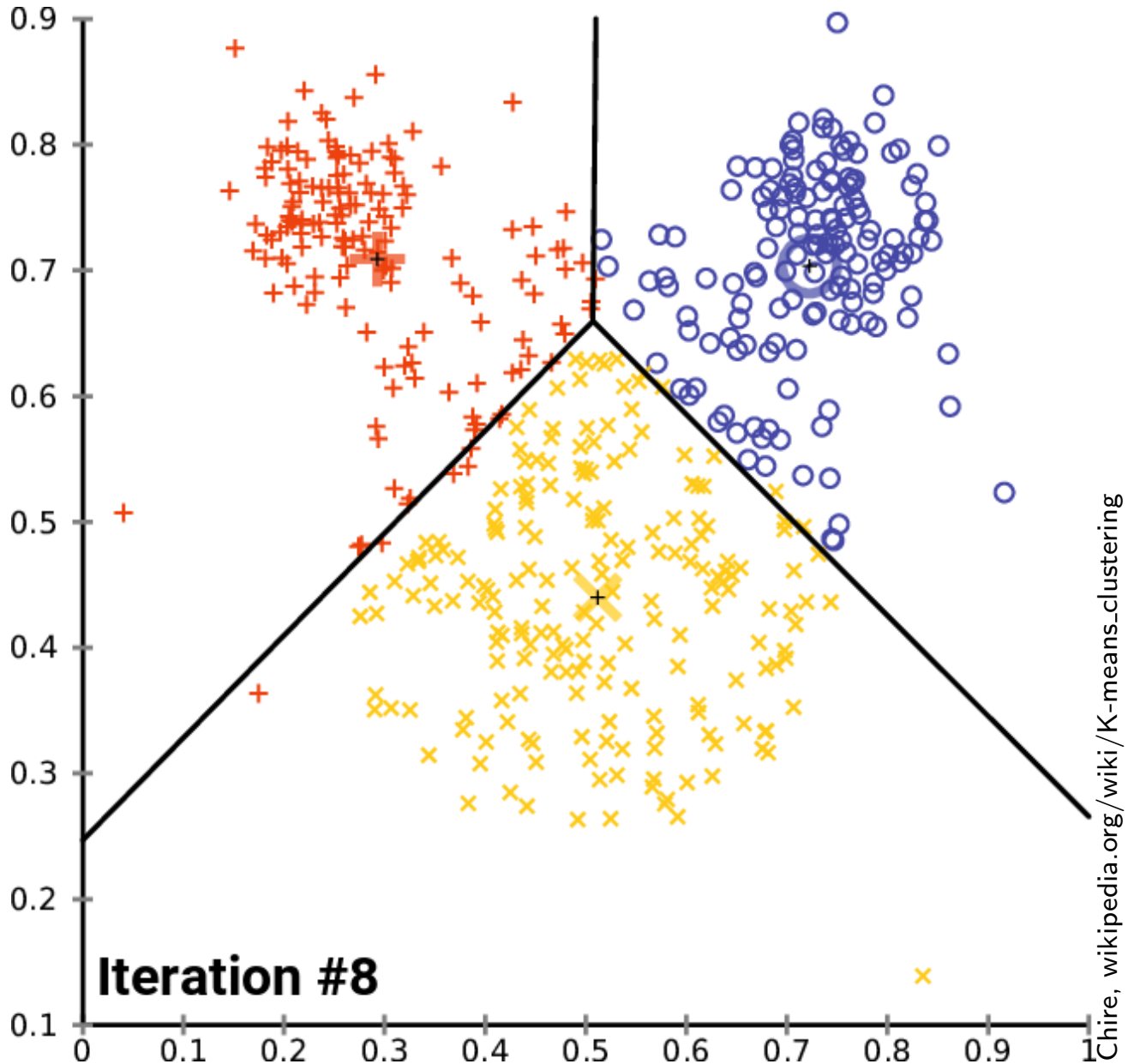
Exemple K-means (itération 06)



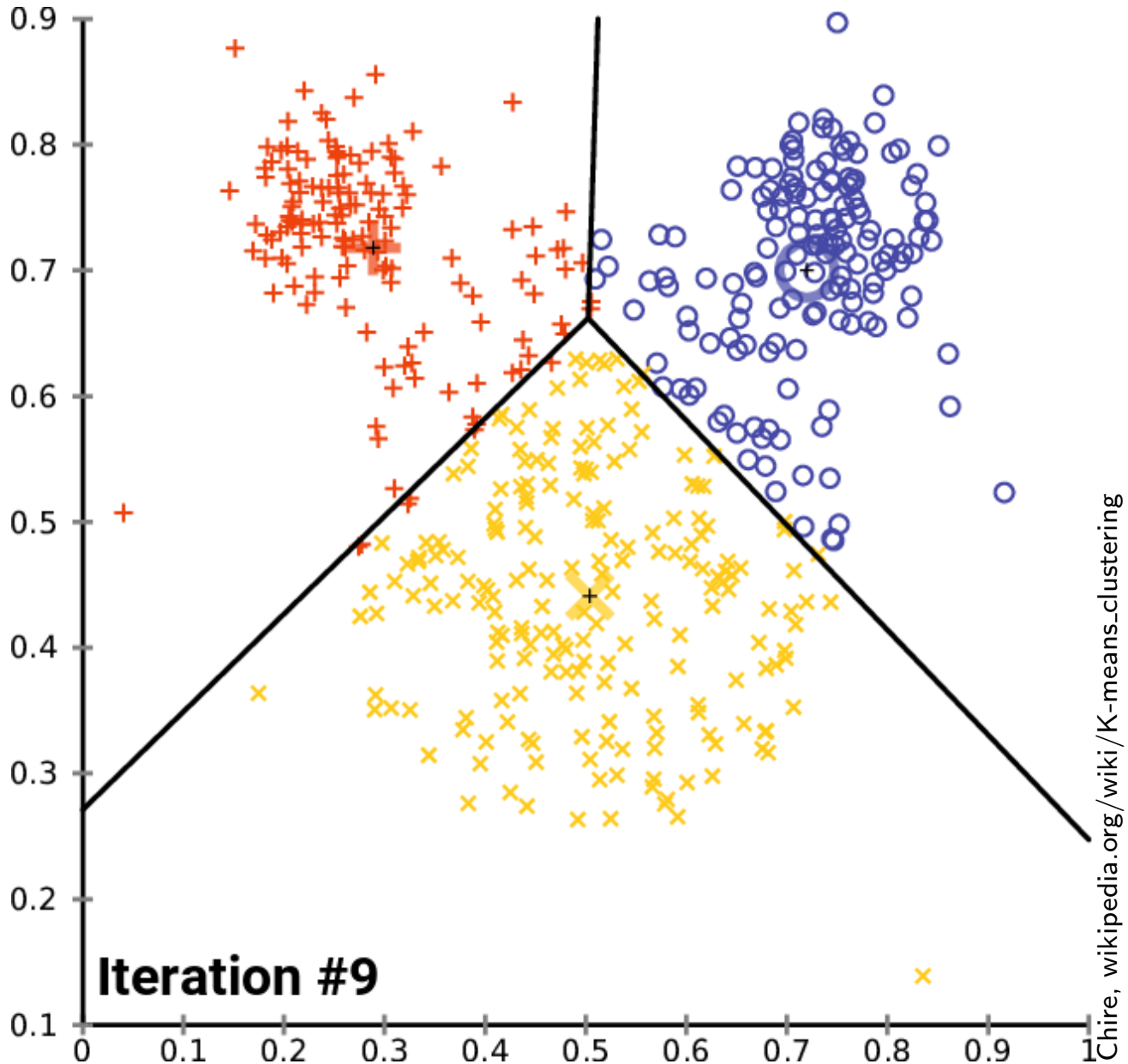
Exemple K-means (itération 07)



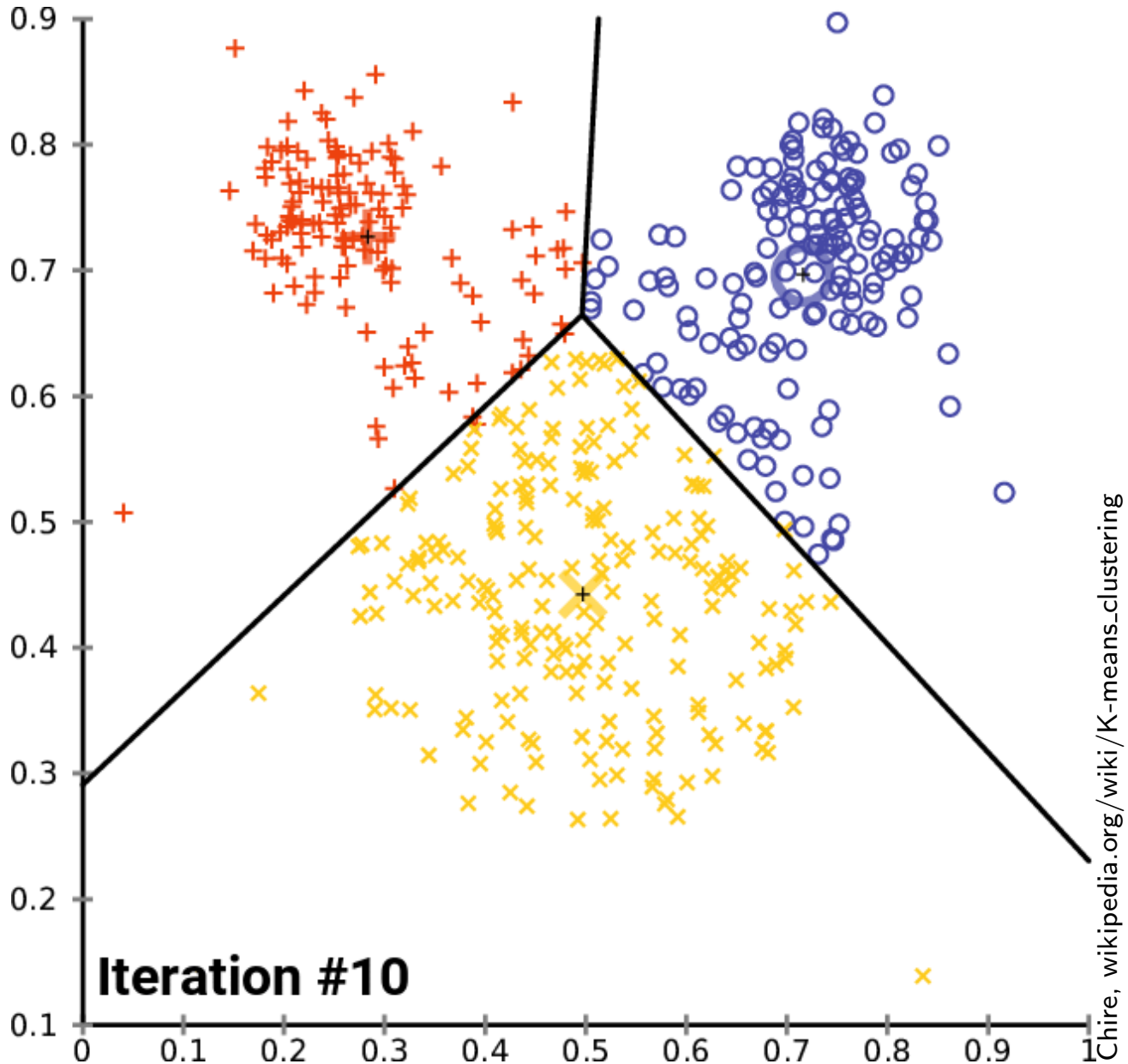
Exemple K-means (itération 08)



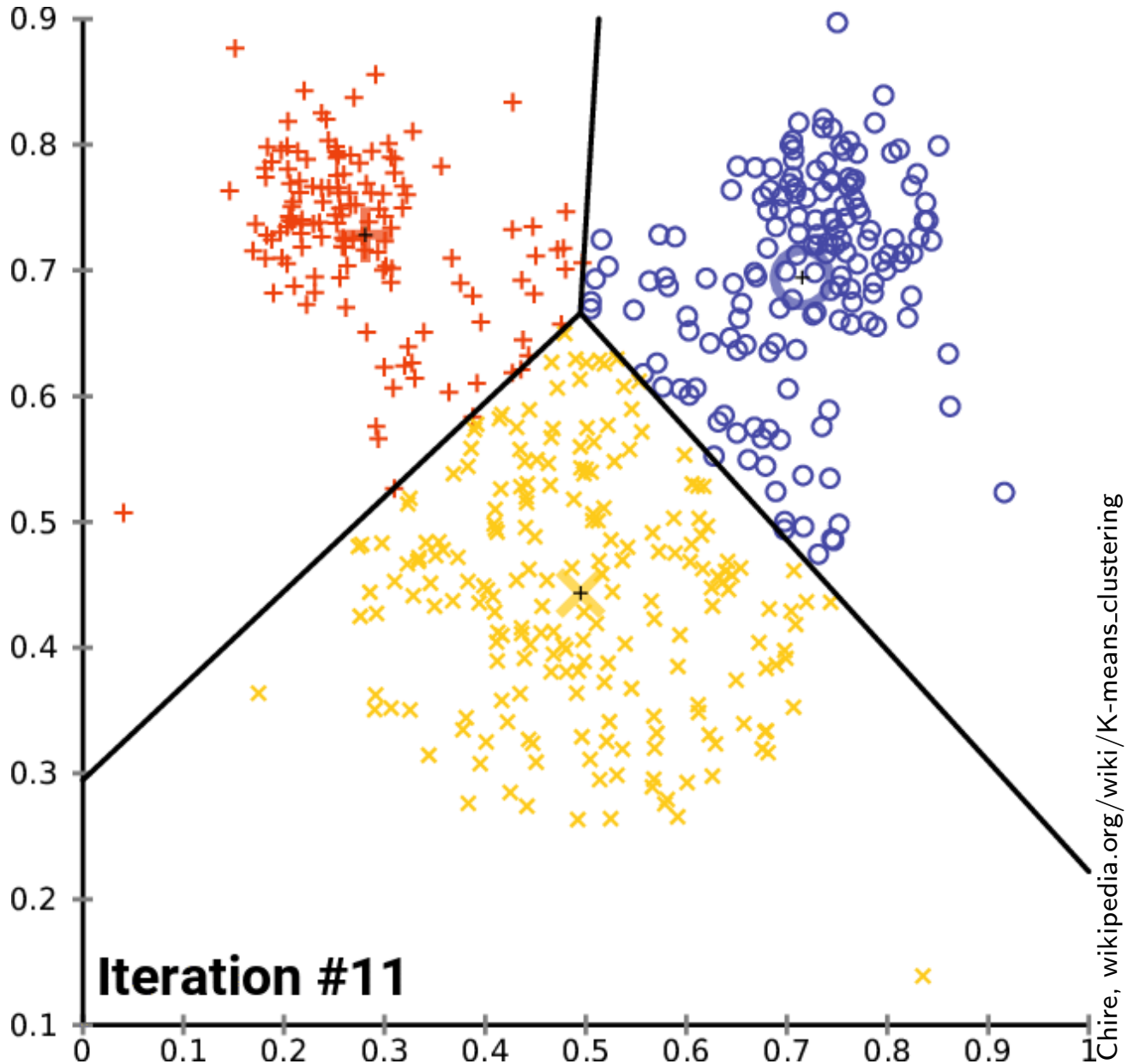
Exemple K-means (itération 09)



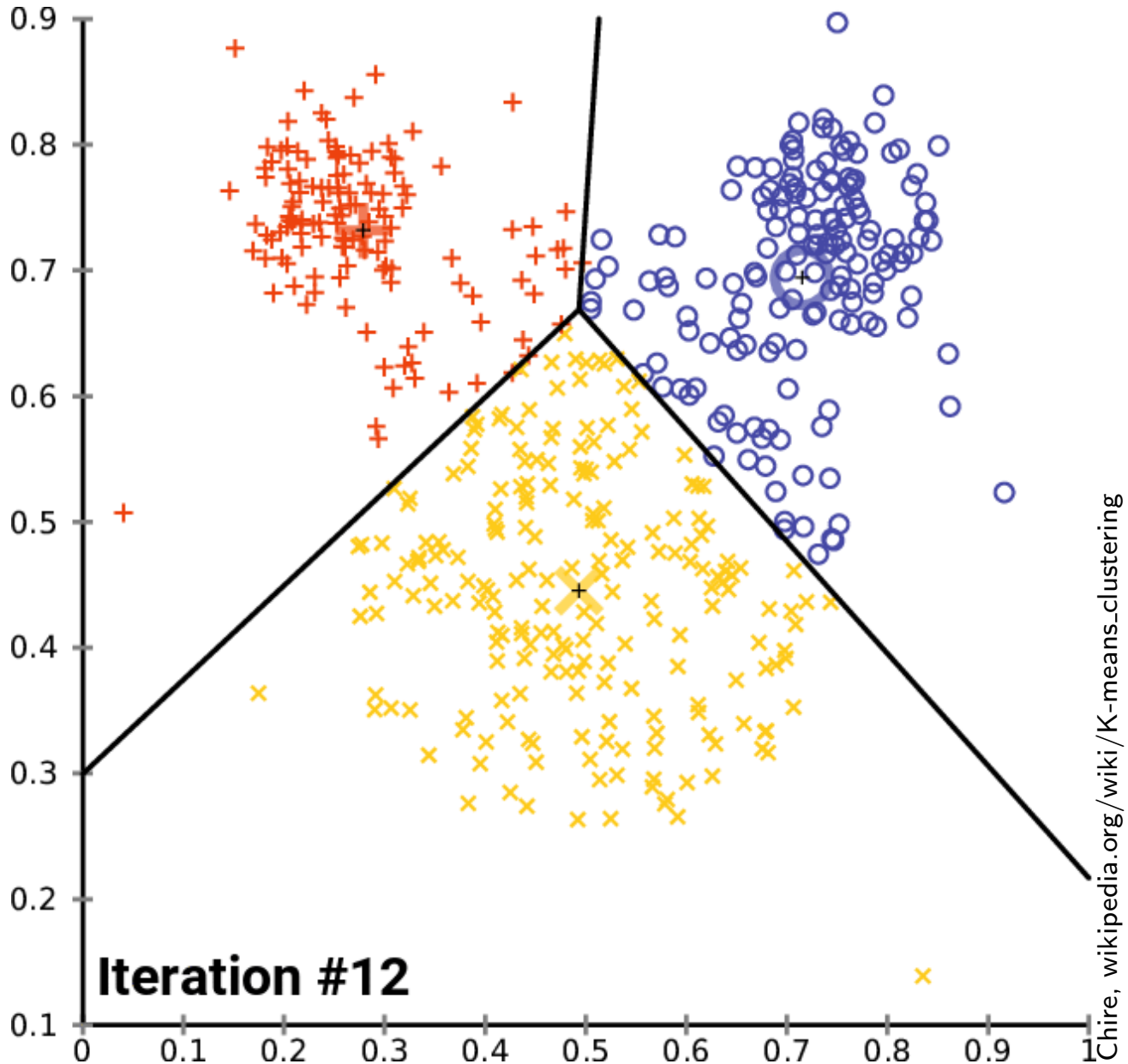
Exemple K-means (itération 10)



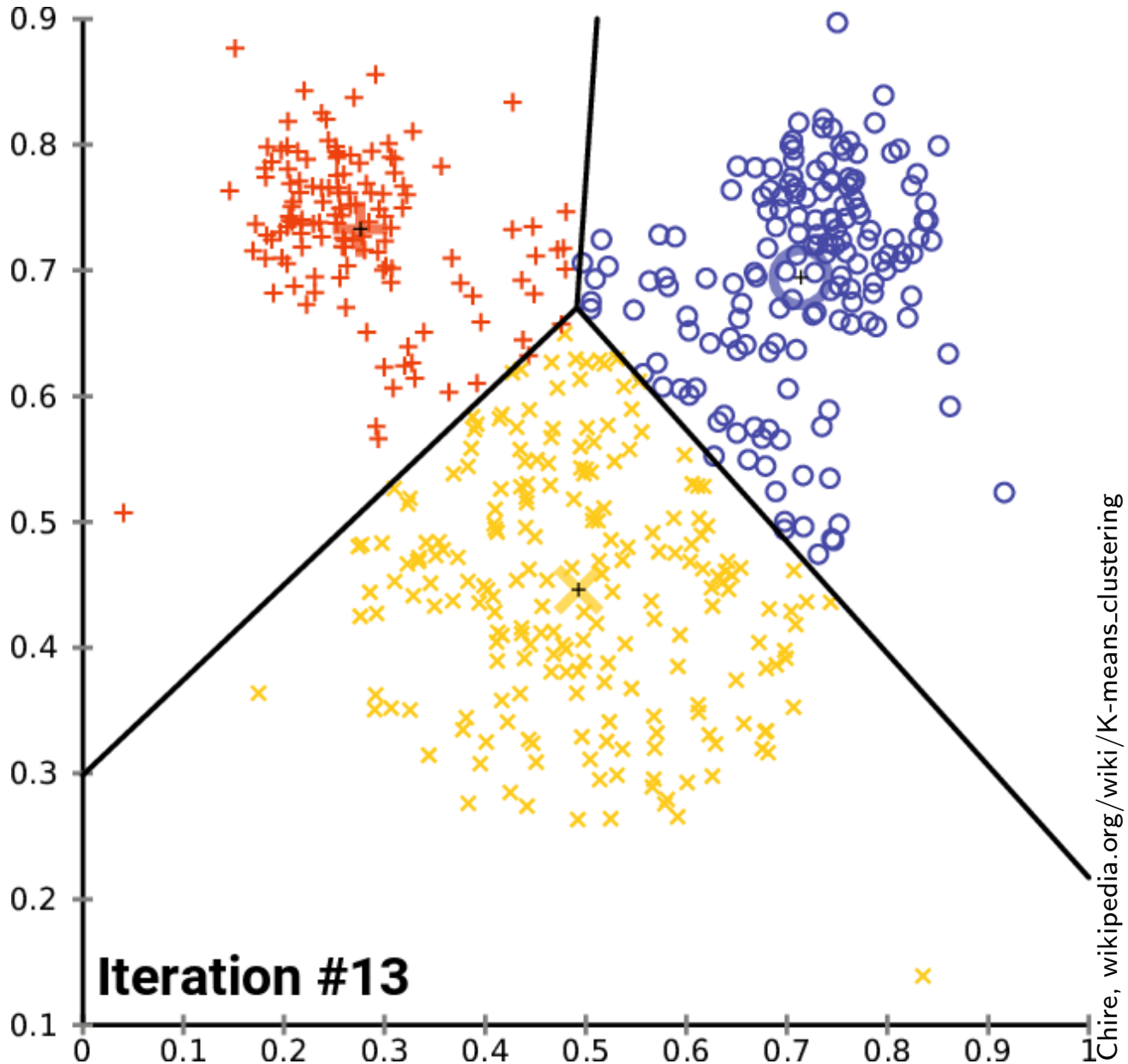
Exemple K-means (itération 11)



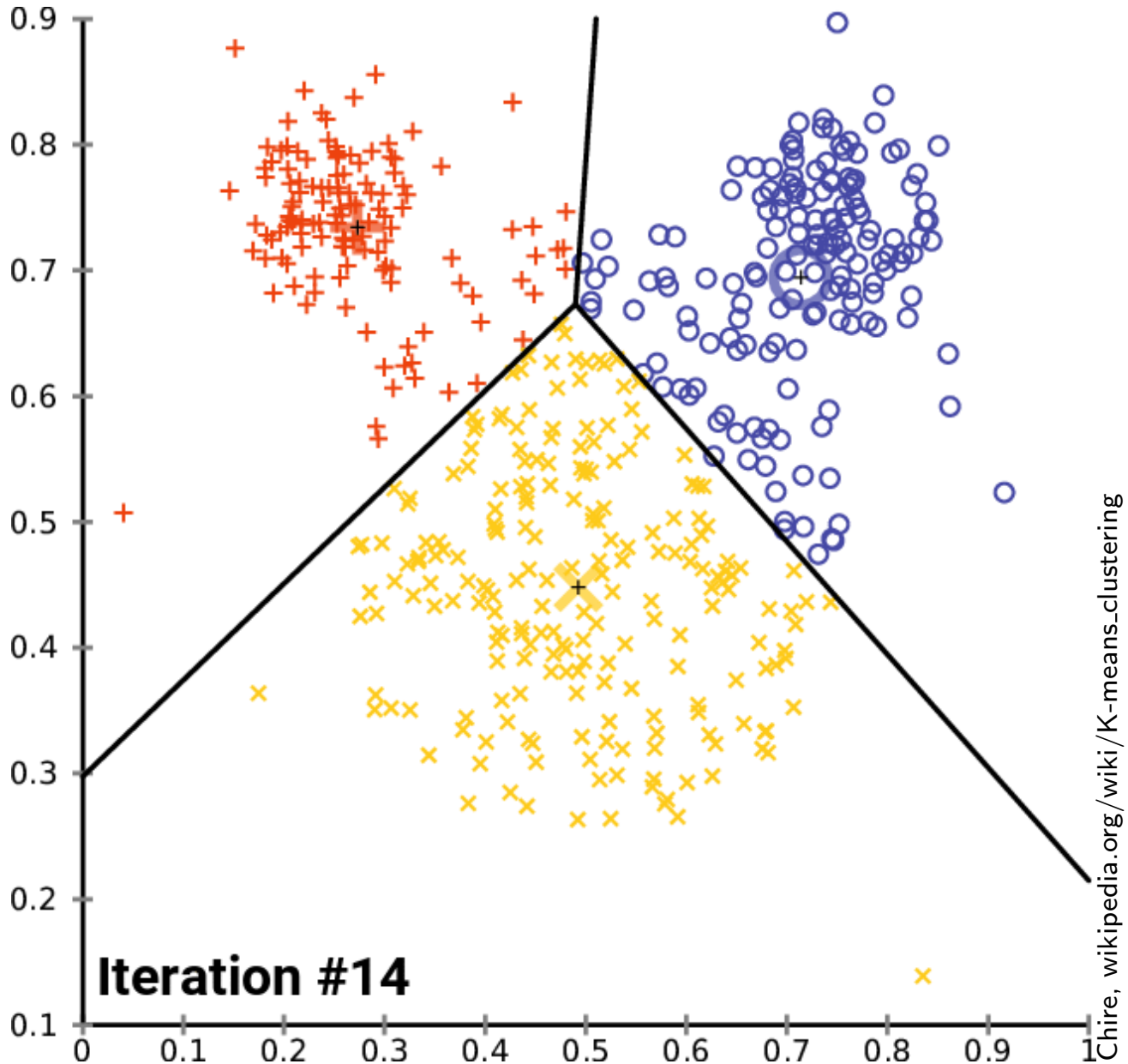
Exemple K-means (itération 12)



Exemple K-means (itération 13)



Exemple K-means (itération 14)



Algorithme de Lloyd (discussion)

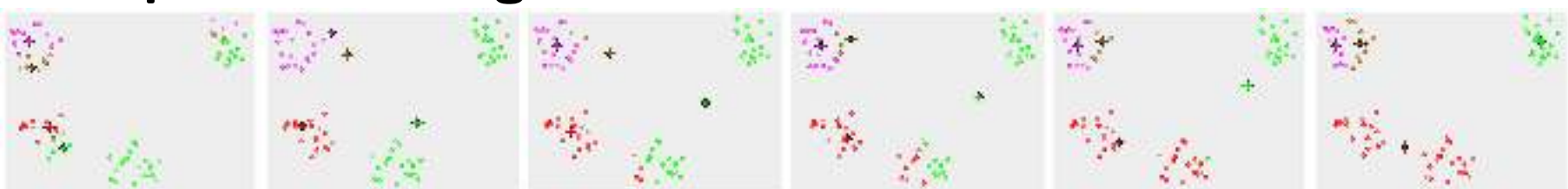
Convergence vers la solution globale

Algorithme de Lloyd suit une stratégie gloutonne (greedy, il fait, étape par étape, un choix optimum local) :

- converge en général très rapidement
- peut tomber dans un minimum local

$\Rightarrow$ pertinent de le faire tourner plusieurs fois (avec des initialisations différentes) et garder la solution qui a la plus faible variance intra-cluster

Exemple de convergence vers un minimum local



Source : Agor153 https://en.wikipedia.org/wiki/K-means_clustering

Algorithme de Lloyd (discussion, 2)

Coût de calcul

- calcul de Kn distances en p dimensions : $\mathcal{O}(npK)$
- calcul répété t itérations : $\mathcal{O}(npKt)$

Comme $K, t \ll n$ l'algorithme est linéaire en le nombre d'observations n (rappel, coût du clustering hiérarchique $\mathcal{O}(n^2)$) :
$\rightarrow$ le calcul des distances d'une observation $\vec{x}^i$ aux $n-1$ autres points est remplacé par un calcul de sa distance à K centroïdes.

Algorithme de Lloyd (discussion, 3)

Forme des clusters clusters forment des domaines avec chaque observation attachée au centroïde dont elle est le plus proche : diagramme de Voronoï, domaine convexe

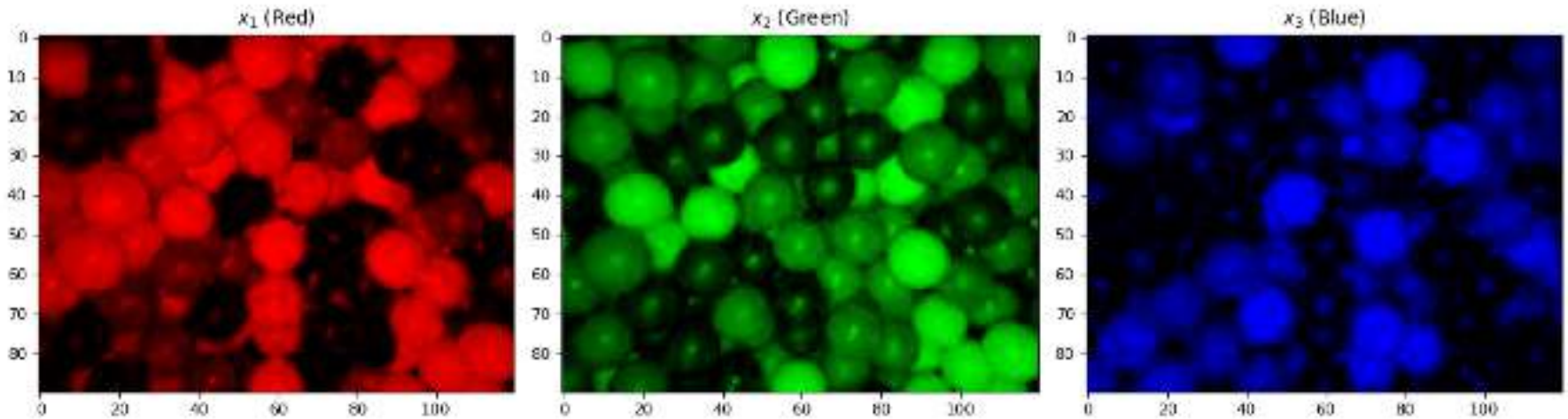
Sensibilité aux données aberrantes Données aberrantes vont tirer un cluster à elles : une observation $\vec{x}^i$ très éloignées des autres observations formera son propre cluster (reste partitionné en $K-1$ clusters) $\Rightarrow$ peut être utilisé précisément pour la détection d'observations aberrantes

Exemple d'application : clustering par couleurs

Combien de couleurs de balles dans cette image RGB ?

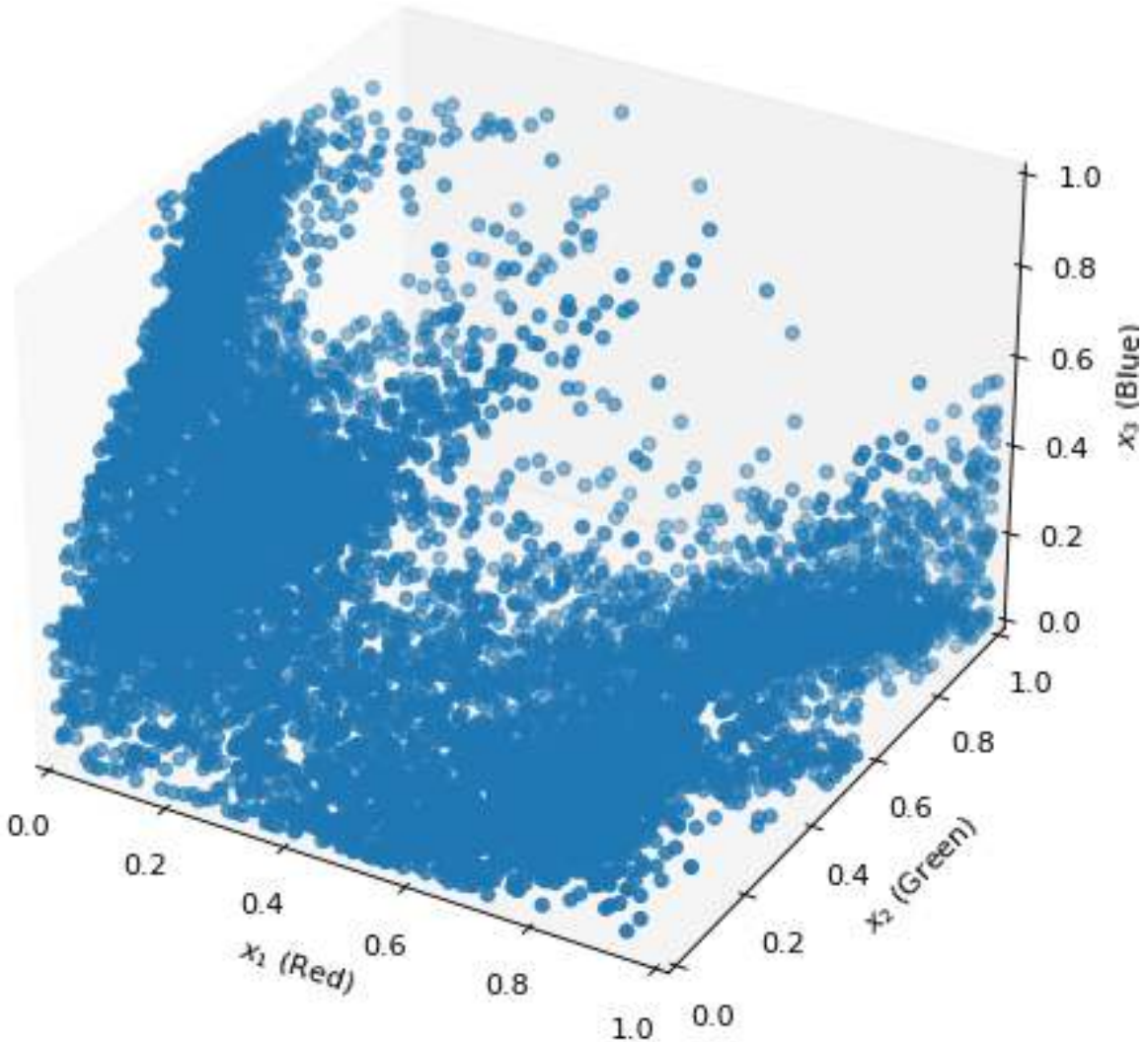
Chaque Pixel dans une image RGB est une observation dans un espace à 3 dimensions

Dans cette image, il y a $n = 90 \times 120 = 10800$ pixels $\vec{x}^i$ avec chacun $p = 3$ valeurs : $\vec{x}^i = (x_1^i, x_2^i, x_3^i) = (\text{red}, \text{green}, \text{blue})$



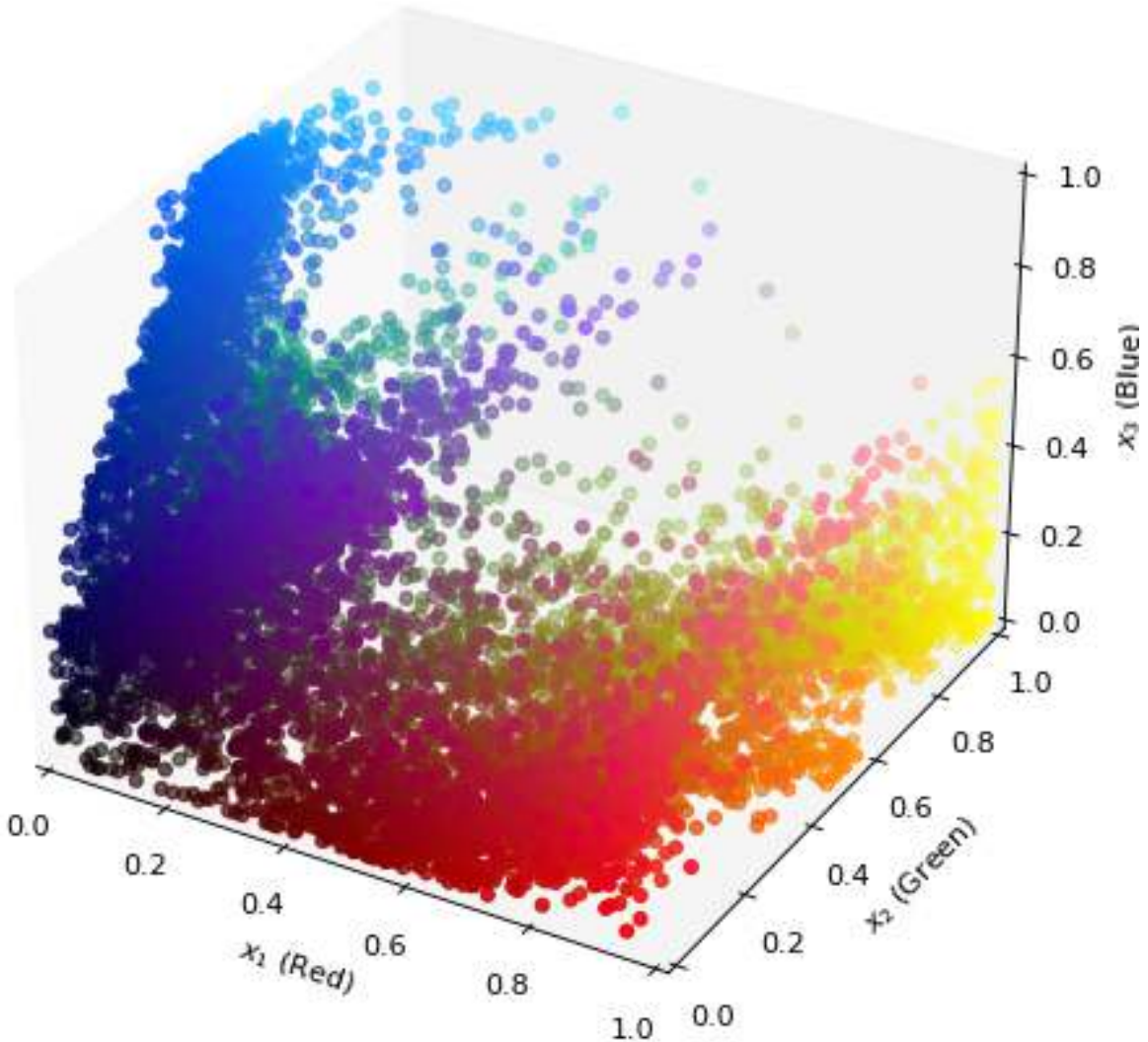
But du clustering : assigner chaque pixel $\vec{x}^i$ à l'un de K cluster selon ses valeurs RGB (segmentation d'image par clustering des couleurs).

Le problème du clustering vu dans l'espace RGB (1/4



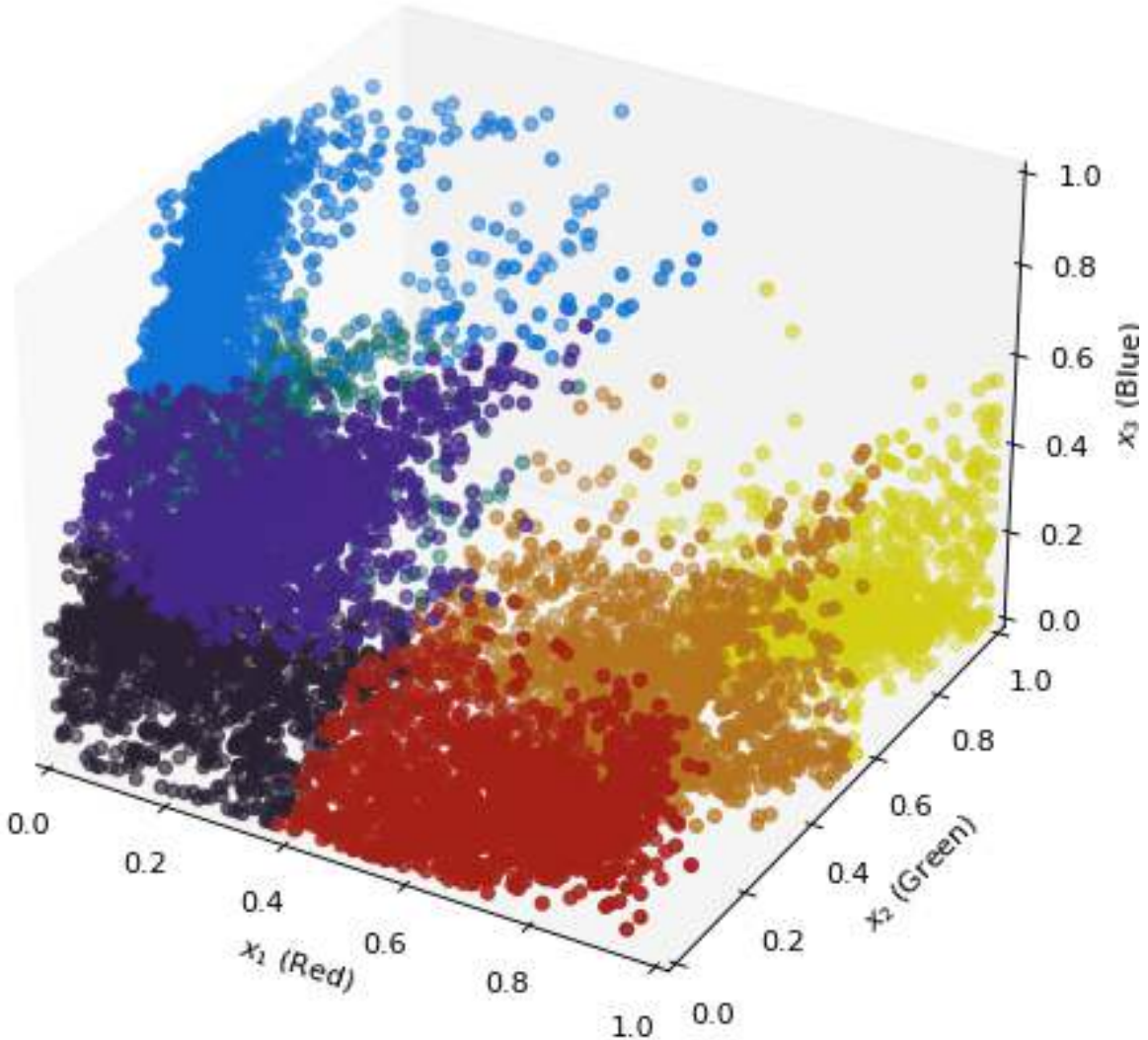
Chaque point dans cet espace RGB représente un pixel de l'image.

Le problème du clustering vu dans l'espace RGB (2/4)



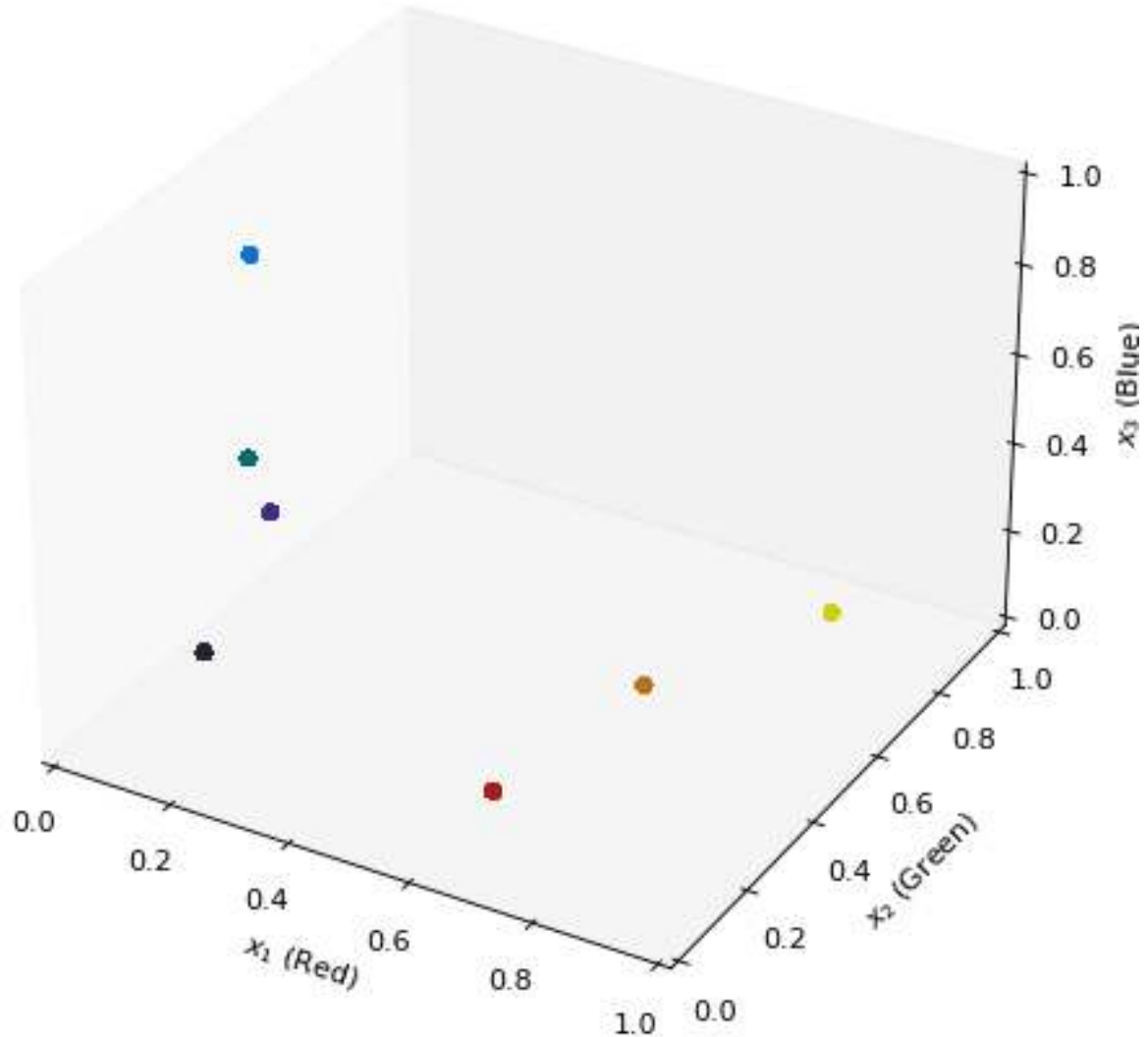
Avec la couleur correspondante pour faciliter la visualisation

Le problème du clustering vu dans l'espace RGB (3/4)



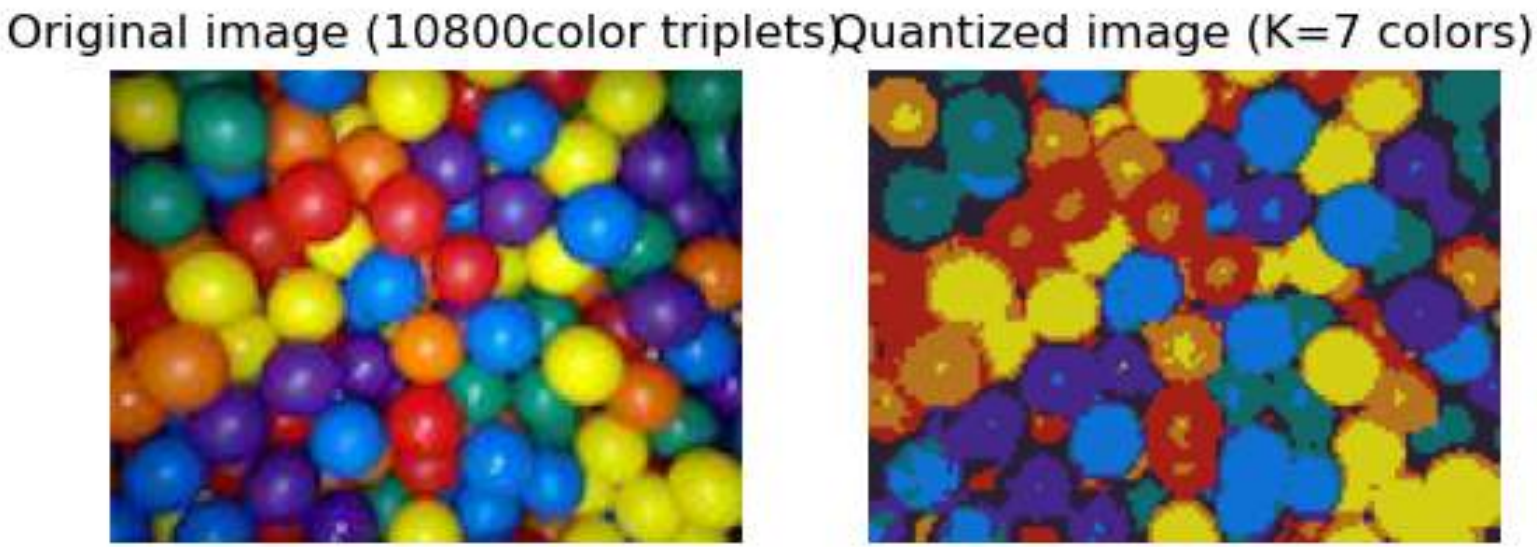
Après clustering K-means ($K = 7$), la couleur des pixels ...

Le problème du clustering vu dans l'espace RGB (4/4)

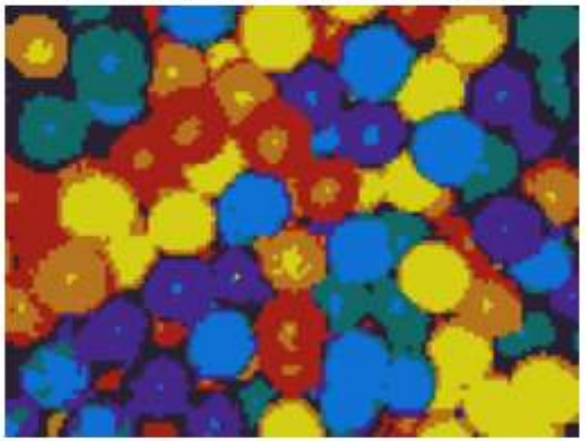


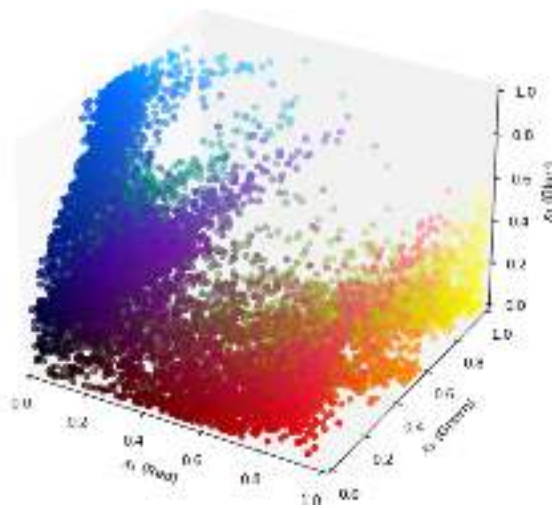
. . . est celle du centroïde du cluster.

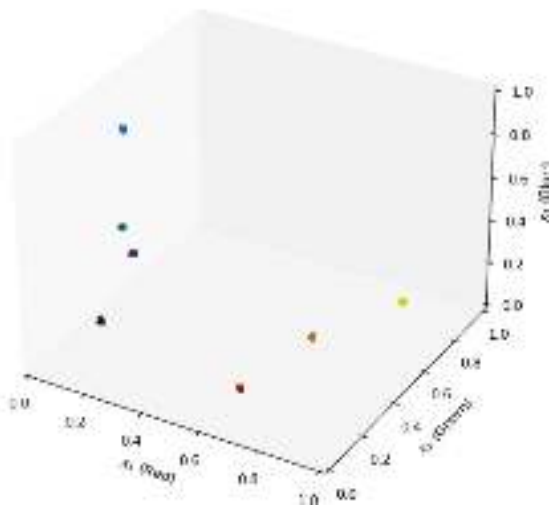
Clustering avec $K = 7$

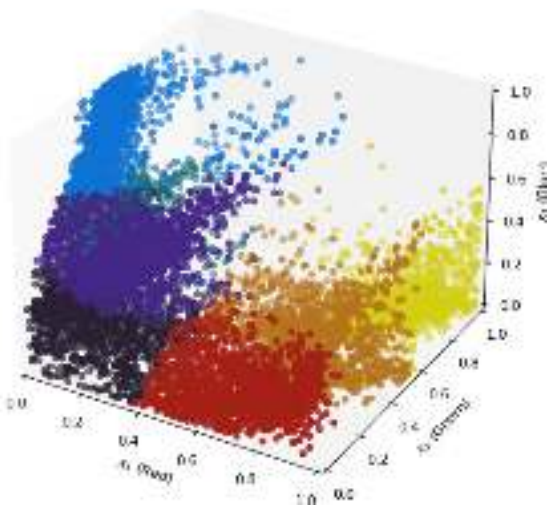


K-means color clustering $K = 7$

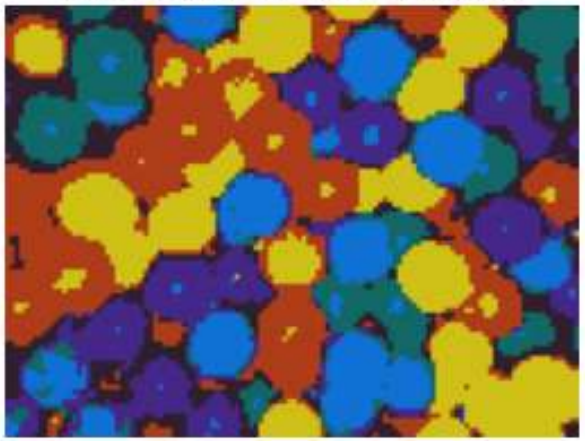


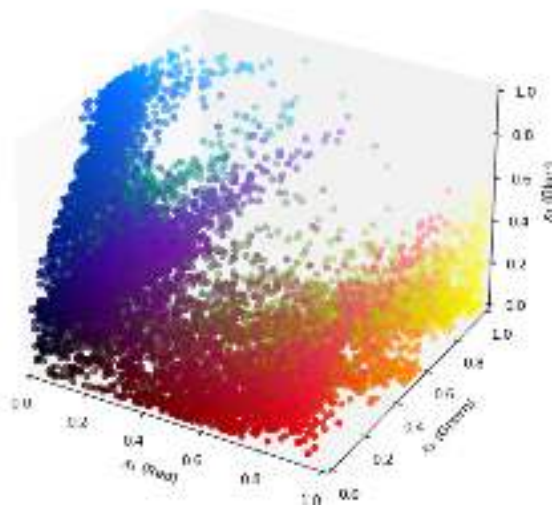


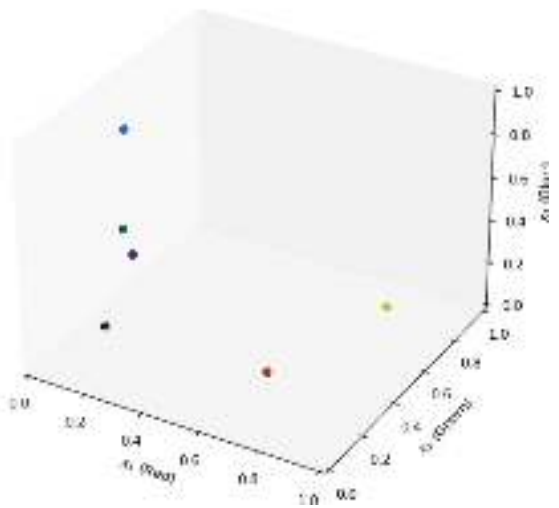


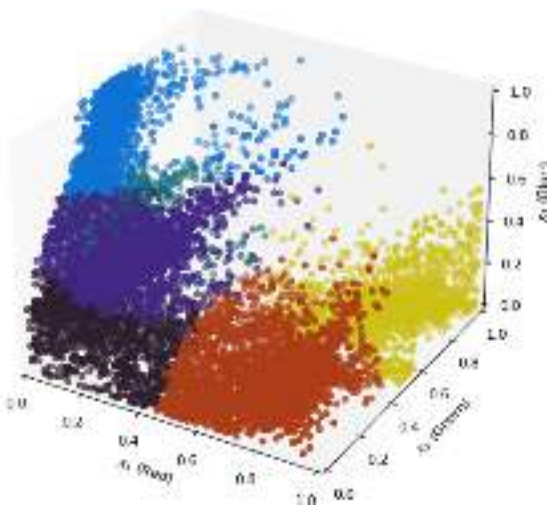


K-means color clustering $K = 6$

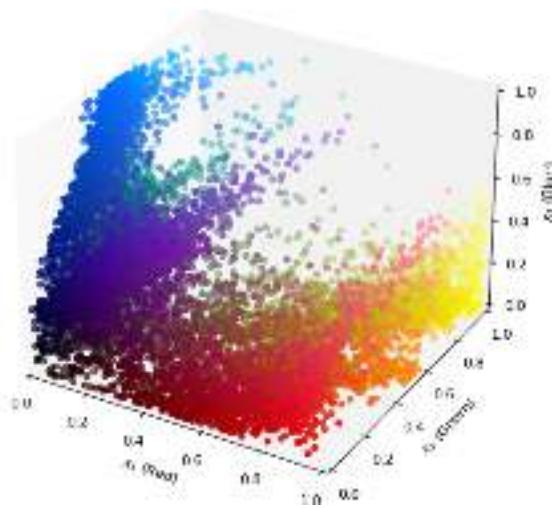


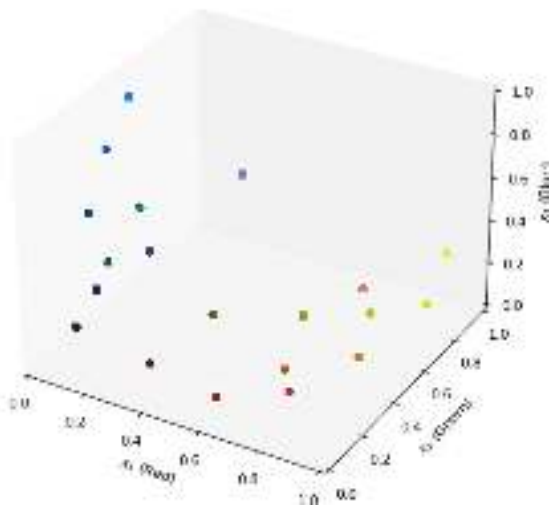


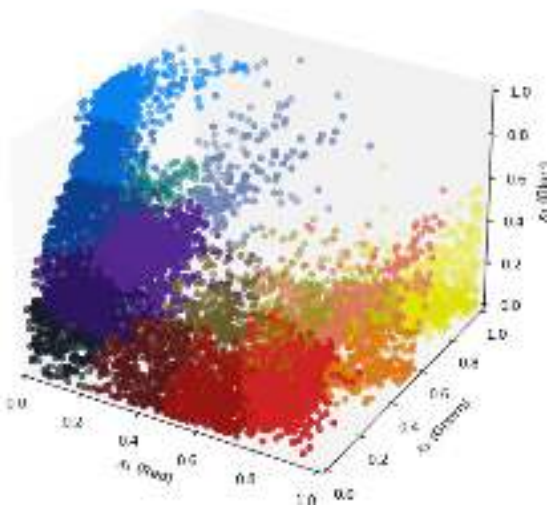




K-means color clustering $K = 20$







Variantes du k-means : k-means++

L'algorithme du k-means est stochastique : on peut obtenir des résultats différents selon l'initialisation, et certains de ces résultats peuvent avoir une inertie bien plus grande que la solution optimale. Pour éviter ce problème, l'algorithme k-means++ commence par initialiser les centroïdes de manière à les disperser au maximum parmi les données. Plus précisément, la procédure consiste à

1. Choisir un premier centroïde $\vec{x}^1$ aléatoirement parmi les observations $\mathscr{D}$
2. Pour $k = 2, \ldots, K$: Choisir le k-ème centroïde $\vec{u}^k$ parmi $\mathscr{D} \setminus \vec{u}^{k-1}$, en suivant une loi proportionnelle au carré de la distance à $\vec{u}^{k-1}$, c'est-à-dire que $\vec{u}^k$ aura de fortes chances d'être éloigné de $\vec{u}^{k-1}$

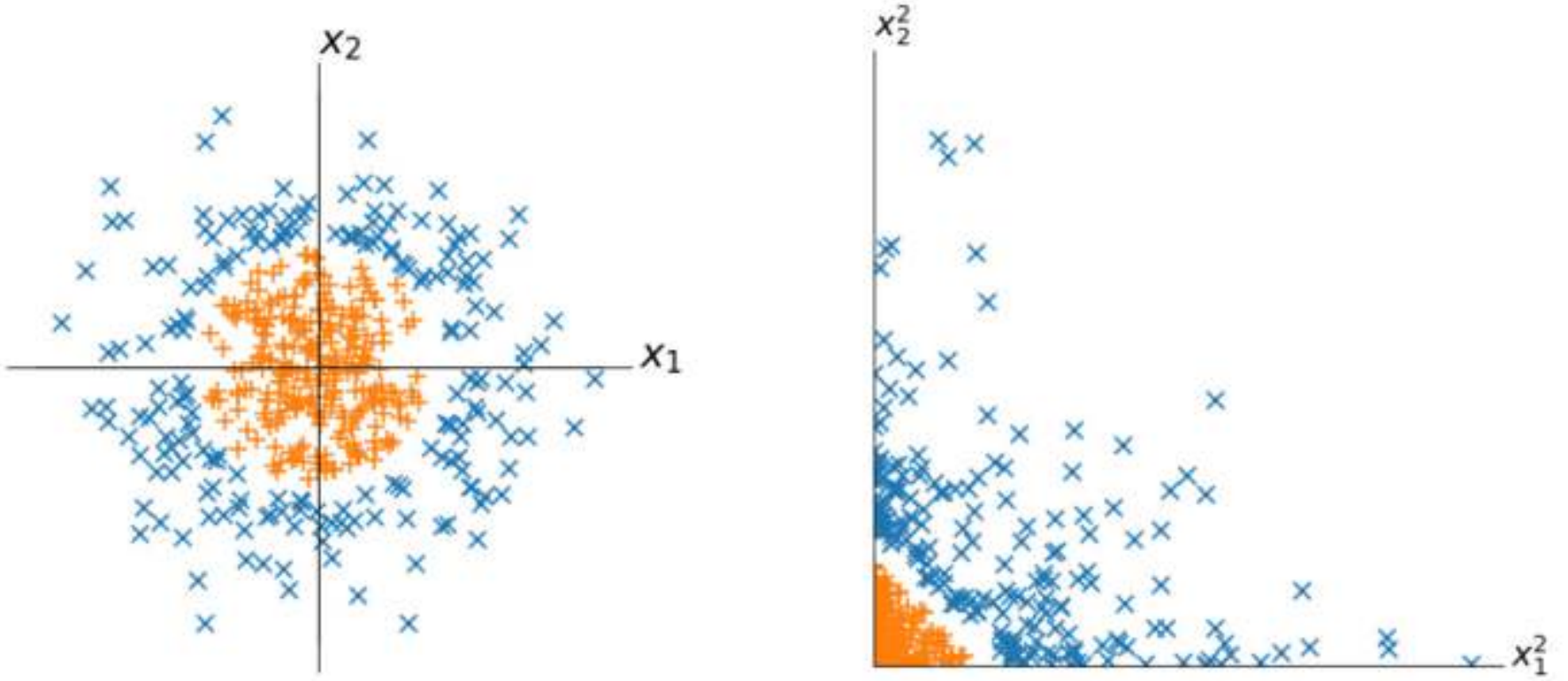
Cette approche ne rend pas le k-means déterministe, mais permet d'éviter les « pires » solutions.

Méthode des k-moyennes avec noyau

La méthode des k-moyennes requiert de décrire les données dans un espace euclidien, et ne peut former que des clusters convexes, ce qui peut être limitant. Cependant, l'astuce du noyau (cf. section 10.3.3) s'applique à l'algorithme de Lloyd.

Rappel : Cas non linéaire : SVM à noyau

Les fonctions linéaires ne sont pas toujours appropriées pour séparer les données...



(A) Un cercle semble bien mieux indiqué qu'une droite pour séparer ces données.

(B) Après transformation par l'application $\phi : (x_1, x_2) \mapsto (x_1^2, x_2^2)$, les données sont linéairement séparables dans l'espace de redescription.

FIGURE 10.4 – Transformer les données permet de les séparer linéairement dans un espace de redescription. Azencott

Idée : définir un espace de redescription dans lequel la fonction de séparation est linéaire.

Rappel : Espace de redescription : un exemple

Exemple : la fonction

$$f : \mathbb{R}^2 \to \mathbb{R}$$
$$\vec{x} \mapsto x_1^2 + x_2^2 - R^2$$

n'est pas linéaire en $\vec{x} = (x_1, x_2)$ mais elle est linéaire en (x_1^2, x_2^2). On peut donc définir

$$\phi : \mathbb{R}^2 \to \mathbb{R}^2$$
$$(x_1, x_2) \mapsto \left(x_1^2, x_2^2\right)$$

La fonction de décision f est linéaire en $\phi\left(\vec{x}\right)$:

$$f\left(\vec{x}\right) = \left(\phi\left(\vec{x}\right)\right)_1 + \left(\phi\left(\vec{x}\right)\right)_2 - R^2$$

et nous pouvons l'apprendre en utilisant une SVM *sur les images des données* par l'application ϕ.

Rappel : Espace de redescription : cas général

Dans le cas général, les observations sont dans un espace quelconque $\mathcal{X}$:

- $\mathcal{X} = \mathbb{R}^p$
- $\mathcal{X} =$ ensemble des chaînes de caractères sur un alphabet donné
- $\mathcal{X} =$ espace de tous les graphes
- $\mathcal{X} =$ espace de fonctions

Définition 10.8 (Espace de redescription) On appelle espace de redescription l'espace de Hilbert $\mathcal{H}$ dans lequel il est souhaitable de redécrire les données, au moyen d'une application $\phi : \mathcal{X} \to \mathcal{H}$, pour y entraîner une SVM sur les images des observations du jeu d'entraînement.

La redescription des données dans un espace de Hilbert nous permet d'utiliser un algorithme linéaire, comme la SVM à marge souple, pour résoudre un problème non linéaire.

Rappel : Définition du noyau

Définition 10.10 (Noyau) Nous appelons noyau toute fonction k de deux variables s'écrivant sous la forme d'un produit scalaire des images dans un espace de Hilbert de ses variables. Ainsi, un noyau est une fonction continue, symétrique, et semi-définie positive :

$$\forall N \in \mathbb{N}, \forall \left(\vec{x}^1, \vec{x}^2, \ldots, \vec{x}^N\right) \in \mathcal{X}^N \text{ et } (a_1, a_2, \ldots, a_N) \in \mathbb{R}^N,$$
$$\sum_{i=1}^{N} \sum_{\ell=1}^{N} a_i a_\ell k\left(\vec{x}^i, \vec{x}^\ell\right) \geq 0.$$

Définition 10.11 (Matrice de Gram) Étant données n observations $\left(\vec{x}^1, \vec{x}^2, \ldots, \vec{x}^N\right) \in \mathcal{X}^N$ et un noyau k sur $\mathcal{X}$, on appelle matrice de Gram de ces observations la matrice $K \in \mathbb{R}^{n \times n}$ telle que

$$K_{i\ell} = k\left(\vec{x}^i, \vec{x}^\ell\right)$$

Cette matrice est semi-définie positive.

Méthode des k-moyennes avec noyau : dérivation

Soit $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ un noyau.

Il existe un espace de Hilbert $\mathcal{H}$ et une application $\phi : \mathcal{X} \to \mathcal{H}$ telle que, pour toute paire $\vec{x}, \vec{x}' \in \mathcal{X} \times \mathcal{X}$:

$$\kappa(\vec{x}, \vec{x}') = \langle \phi(\vec{x}), \phi(\vec{x}') \rangle_{\mathcal{H}}.$$

Pour appliquer l'algorithme de Lloyd aux images $\{\phi(\vec{x}^1), \phi(\vec{x}^2), \ldots, \phi(\vec{x}^n)\}$ des éléments de $\mathcal{D}$ dans $\mathcal{H}$, nous aurions besoin de calculer, à chaque itération, la distance de $\phi(\vec{x}^i)$ à chacun des K centroïdes $\vec{h}_1$, $\vec{h}_2$, ..., $\vec{h}_K$.

La position d'un centroïde $\vec{h}_k$ se calcule(rait) comme la moyenne des images des observations appartenant à $\mathcal{C}_k$:

$$\vec{h}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\vec{x}^i \in \mathcal{C}_k} \phi(\vec{x}^i)$$

Méthode des k-moyennes avec noyau : dérivation (suite)

Or la distance de l'image d'une observation $\vec{x}^i$ à un centroïde peut se calculer comme :

$$\begin{aligned}\left\|\phi\left(\vec{x}^i\right)-\vec{h}_k\right\|_2^2 &= \left\|\phi\left(\vec{x}^i\right)-\frac{1}{|\mathscr{C}_k|}\sum_{\vec{x}^i\in\mathscr{C}_k}\phi\left(\vec{x}^i\right)\right\|_2^2 \\ &= \left\langle \phi\left(\vec{x}^i\right)-\frac{1}{|\mathscr{C}_k|}\sum_{\vec{x}^i\in\mathscr{C}_k}\phi\left(\vec{x}^i\right), \phi\left(\vec{x}^i\right)-\frac{1}{|\mathscr{C}_k|}\sum_{\vec{x}^i\in\mathscr{C}_k}\phi\left(\vec{x}^i\right)\right\rangle_{\mathscr{H}} \\ &= \kappa\left(\vec{x}^i,\vec{x}^i\right)-\frac{2}{|\mathscr{C}_k|}\sum_{\vec{u}\in\mathscr{C}_k}\kappa\left(\vec{u},\vec{x}^i\right)+\frac{1}{|\mathscr{C}_k|^2}\sum_{\vec{u}\in\mathscr{C}_k}\sum_{\vec{v}\in\mathscr{C}_k}\kappa\left(\vec{u},\vec{v}\right)\end{aligned}$$

On peut ainsi calculer l'affectation des observations à l'étape 2 (Slide 32) :

$$k\left(\vec{x}^i\right)=\underset{k=1,\ldots,K}{\operatorname{argmin}}\left\|\phi\left(\vec{x}^i\right)-\vec{h}_k\right\|_2^2$$

sans devoir calculer ϕ et sans avoir à jamais calculer les centroïdes de manière explicite (l'étape 3 n'est plus nécessaire, on itère sur l'étape 2) :

Algorithme de Lloyd avec noyau

Etant données n observations dans $\mathbb{R}^p$, un nombre K de clusters, et un noyau $\kappa\left(\vec{x}, \vec{x}'\right)$:

1. choisir K observations $\vec{\mu}_1$,$\vec{\mu}_2$,..., $\vec{\mu}_K$ parmi les n observations (centroïdes initiaux) et affecter chaque observation $\vec{x}^i \in \mathscr{D}$ au centroïde tranformé initial $\vec{h}_k = \phi\left(\vec{\mu}_k\right)$ pour lequel la transformée $\phi\left(\vec{x}^i\right)$ est la plus proche :

$$\begin{aligned} k\left(\vec{x}^i\right) &= \underset{k=1,\ldots,K}{\operatorname{argmin}} \left\| \phi\left(\vec{x}^i\right) - \vec{h}_k \right\|_2^2 \\ &= \underset{k=1,\ldots,K}{\operatorname{argmin}} \left(\kappa\left(\vec{x}^i, \vec{x}^i\right) - 2\kappa\left(\vec{x}^i, \vec{\mu}_k\right) + \kappa\left(\vec{\mu}_k, \vec{\mu}_k\right) \right) \end{aligned}$$

2. Affecter chaque observations $\vec{x}^i \in \mathscr{D}$ au centroïde transformé $\vec{h}_k = \phi\left(\vec{\mu}_k\right)$ pour lequel sa transformation $\phi\left(\vec{x}^i\right)$ est le plus proche :

$$\begin{aligned} k\left(\vec{x}^i\right) &= \underset{k=1,\ldots,K}{\operatorname{argmin}} \left\| \phi\left(\vec{x}^i\right) - \vec{h}_k \right\|_2^2 \\ &= \underset{k=1,\ldots,K}{\operatorname{argmin}} \left(\kappa\left(\vec{x}^i, \vec{x}^i\right) - \frac{2}{|\mathscr{C}_k|} \sum_{\vec{u} \in \mathscr{C}_k} \kappa\left(\vec{u}, \vec{x}^i\right) + \frac{1}{|\mathscr{C}_k|^2} \sum_{\vec{u} \in \mathscr{C}_k} \sum_{\vec{v} \in \mathscr{C}_k} \kappa\left(\vec{u}, \vec{v}\right) \right) \end{aligned}$$

3. ~~Recalculer les centroïdes de chaque cluster~~ $\leftarrow$ les centroïdes en 2 sont implicites, l'astuce du noyau ne requiert pas leur calcul explicite !

4. Répéter opération 2 jusqu'à convergence (affectations ne changent plus)

Version à noyau de la méthode des k-moyennes : remarques

• il faut initialiser les clusters (centroïdes aléatoires, calcul explicite de l'étape 2)

• La version à noyau de la méthode des k-moyennes ne permet généralement pas de connaître les centroïdes des clusters, puisqu'ils vivent dans l'espace de redescription $\mathscr{H}$ qui n'est pas accessible sans connaître ϕ.

Clustering par densité

(A) Il nous semble naturel de partitionner ces données en 3 cercles concentriques.

(B) Partitionnement en 3 clusters par clustering agglomératif (lien moyen).

(C) Partitionnement en 3 clusters par k-moyennes.

FIGURE 12.4 – Motivation du clustering par densité.

Azencott

Clustering par densité (suite)

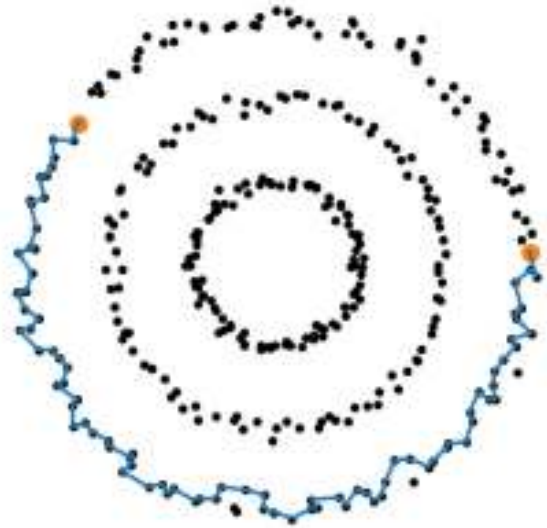


FIGURE 12.5 – Il existe un chemin entre voisins proches permettant de passer d'un point à un autre du même cluster.

Azencott

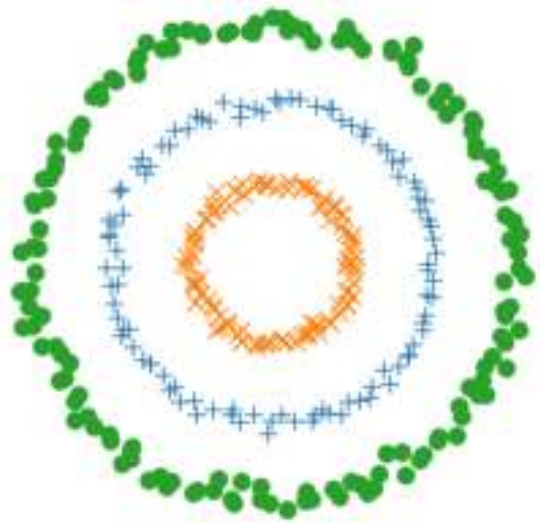


FIGURE 12.6 – Partitionnement des données de la figure 12.4a par DBSCAN.

Azencott

Clustering : conclusions

- clustering = partitionnement de données cherche à identifier des classes sans utiliser d'étiquettes
- en l'absence d'étiquettes, la qualité d'une partition peut s'évaluer sur la base de critères de séparabilité et d'homogénéité
- Le clustering hiérarchique partitionne les données de manière itérative. Son résultat peut être visualisé sur un dendrogramme.
- Le clustering par la méthode des k-moyennes s'effectue grâce à l'algorithme de Lloyd ou une de ses variantes. Il permet de trouver efficacement K clusters convexes.
- La version à noyau de la méthode des k-moyennes permet de l'appliquer pour découvrir des clusters non convexes.
- Le clustering par densité permet d'identifier des régions denses du jeu de données, c'est-à-dire des observations qui peuvent former un ensemble non convexe mais qui sont proches les unes des autres.

Guide de lecture pour ce cours

Chloé-Agathe Azencott “Introduction au Machine Learning”,
Dunod, 2019, ISBN 978-210-080153-4
Chapitre 12 : Clustering