

CM 5

Méthodes de Simulation Informatique

Amaya Nogales Gómez
amaya.nogales-gomez@univ-cotedazur.fr

Licence 3 Informatique
Université Côte d'Azur

11 mars 2022

Plan du cours

1 Introduction

- Préliminaires
- Python: numpy, pandas

2 Base de données

- Génération des données synthétiques
- Base de données réels

3 Analyse descriptive

4 Techniques d'apprentissage supervisée

5 Techniques d'apprentissage non supervisée

6 Contrôle de connaissances

7 Techniques de validation

8 Éléments de la méthodologie scientifique

9 L^AT_EX

- Écriture de textes scientifiques
- Beamer: présentations et posters scientifiques

Apprentissage supervisé: Machines à Vecteurs de Support

- Ω : la population.
- La population est divisée en deux classes, $\{-1, +1\}$.

Apprentissage supervisé: Machines à Vecteurs de Support

- Ω : la population.
- La population est divisée en deux classes, $\{-1, +1\}$.
- Pour chaque objet dans Ω , on a
 - $x = (x^1, \dots, x^d) \in X \subset \mathbb{R}^d$: variables de prediction.
 - $y \in \{-1, +1\}$: étiquettes.

Apprentissage supervisé: Machines à Vecteurs de Support

- Ω : la population.
- La population est divisée en deux classes, $\{-1, +1\}$.
- Pour chaque objet dans Ω , on a
 - $x = (x^1, \dots, x^d) \in X \subset \mathbb{R}^d$: variables de prediction.
 - $y \in \{-1, +1\}$: étiquettes.
- L'objectif est de trouver un hyperplan $\omega^\top x + b = 0$ qui vise à séparer, si possible, les deux classes.

Apprentissage supervisé: Machines à Vecteurs de Support

- Ω : la population.
- La population est divisée en deux classes, $\{-1, +1\}$.
- Pour chaque objet dans Ω , on a
 - $x = (x^1, \dots, x^d) \in X \subset \mathbb{R}^d$: variables de prediction.
 - $y \in \{-1, +1\}$: étiquettes.
- L'objectif est de trouver un hyperplan $\omega^\top x + b = 0$ qui vise à séparer, si possible, les deux classes.
- Les objets futurs seront classés comme

$$\begin{aligned} y &= +1 && \text{si } \omega^\top x + b > 0 \\ y &= -1 && \text{si } \omega^\top x + b < 0 \end{aligned} \tag{1}$$

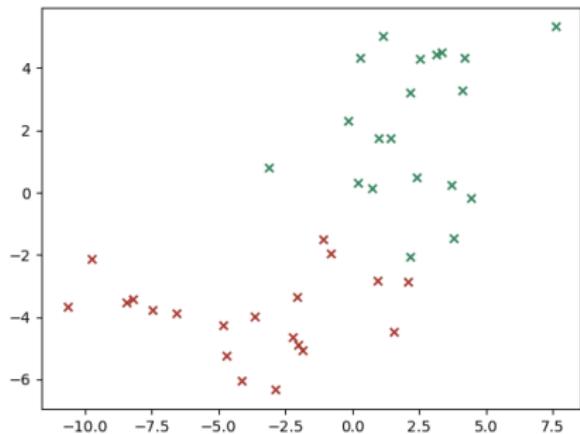
Hard-margin SVM: marge maximal

Séparation linéaire

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^d \omega_j^2$$

s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \quad \forall i = 1, \dots, n$$
$$\omega \in \mathbb{R}^d$$
$$b \in \mathbb{R}.$$



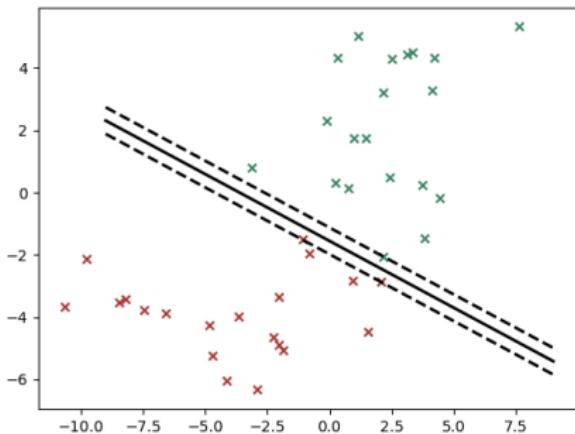
Hard-margin SVM: marge maximal

Séparation linéaire

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^d \omega_j^2$$

s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \quad \forall i = 1, \dots, n$$
$$\omega \in \mathbb{R}^d$$
$$b \in \mathbb{R}.$$



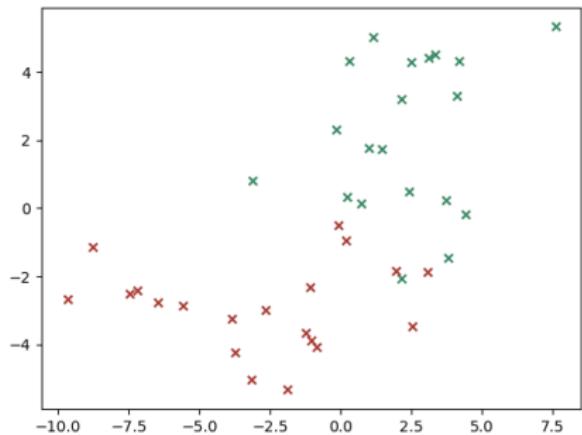
Hard-margin SVM: marge maximal

Séparation non linéaire

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^d \omega_j^2$$

s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \quad \forall i = 1, \dots, n$$
$$\omega \in \mathbb{R}^d$$
$$b \in \mathbb{R}.$$



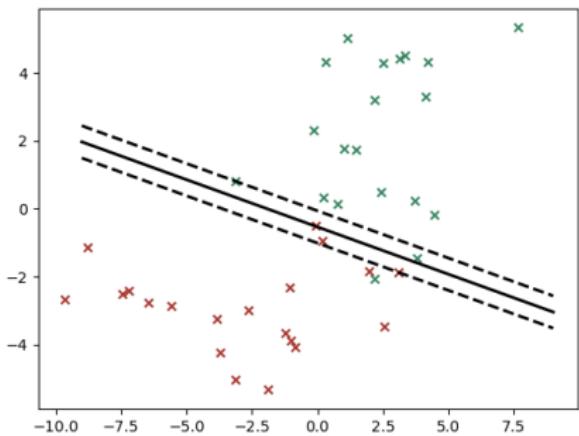
Hard-margin SVM: marge maximal

Séparation non linéaire

$$\min_{\omega, b} \frac{1}{2} \sum_{j=1}^d \omega_j^2$$

s.t.

$$y_i(\omega^\top x_i + b) \geq 1 \quad \forall i = 1, \dots, n$$
$$\omega \in \mathbb{R}^d$$
$$b \in \mathbb{R}.$$



impossible à résoudre!!

Soft-margin SVM

Séparation non linéaire

$$\min_{\omega, b, \xi} \frac{1}{2} \sum_{j=1}^d \omega_j^2 + \frac{C}{n} \sum_{i=1}^n \xi_i$$

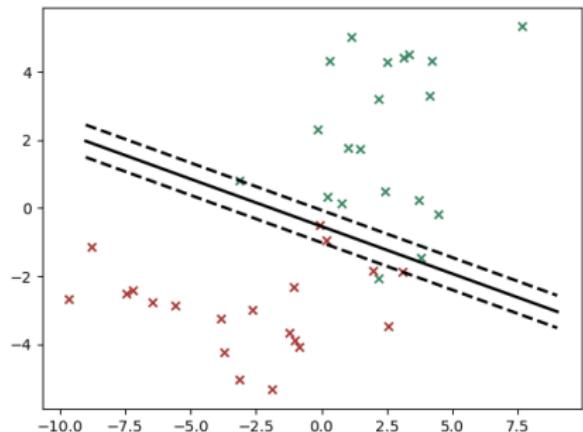
s.t.

$$y_i(\omega^\top x_i + b) \geq 1 - \xi_i \quad \forall i = 1, \dots, n$$

$$\omega \in \mathbb{R}^d$$

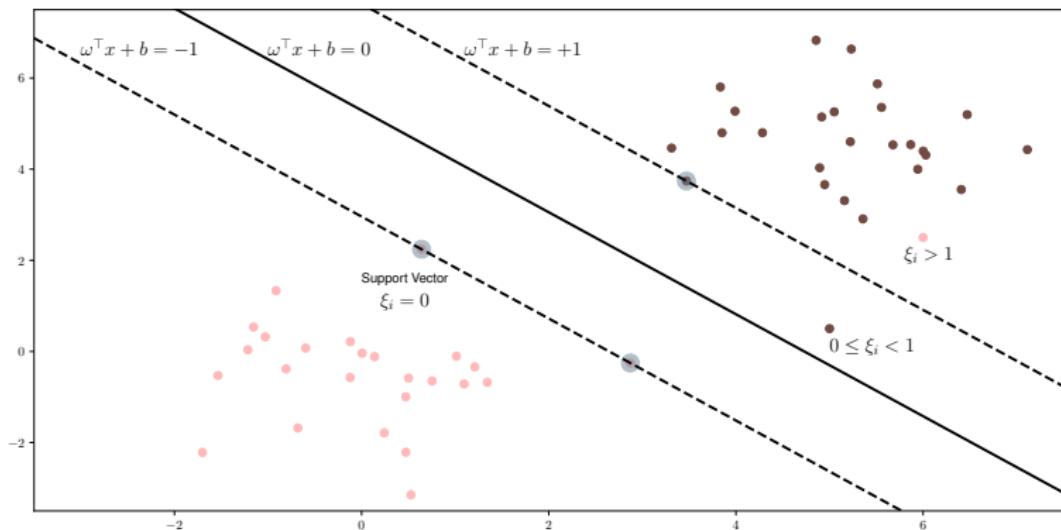
$$b \in \mathbb{R}$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n.$$



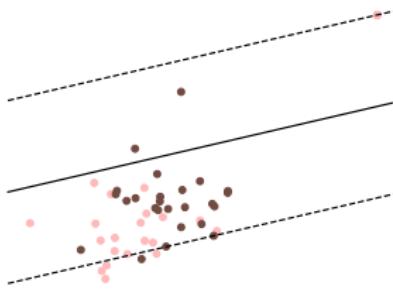
- Un objet i sera correctement classé si $0 \leq \xi_i < 1$
- Mal classé si $\xi_i > 1$.
- Dans le cas $\xi_i = 1$, nous obtenons une égalité (les objets coïncident avec l'hyperplan).

Slack variables



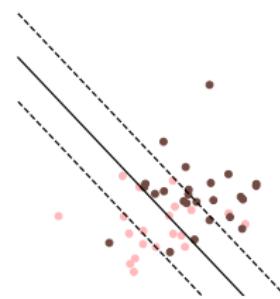
Robustesse aux données aberrantes

SVM avec Hinge Loss



Précision de 44%

SVM avec the Ramp Loss



Précision de 78%

SVM avec la Ramp Loss

$$\min_{\omega, b, \xi, z} \frac{1}{2} \sum_{j=1}^d \omega_j^2 + \frac{C}{n} \left(\sum_{i=1}^n \xi_i + 2 \sum_{i=1}^n (1 - z_i) \right)$$

s.t.

$$(y_i(\omega^\top x_i + b) - 1 + \xi_i) \cdot \textcolor{blue}{z}_i \geq 0 \quad \forall i = 1, \dots, n$$

$$0 \leq \xi_i \leq 2 \quad \forall i = 1, \dots, n$$

$$z \in \{0, 1\}^n$$

$$\omega \in \mathbb{R}^d$$

$$b \in \mathbb{R}.$$

Précision

Étant donné un objet i , il est classé dans la classe positive ou négative selon la valeur de la fonction score, $\text{sign}(\omega^\top x_i + b)$, tandis que pour le cas $\omega^\top x_i + b = 0$, l'objet est classé au hasard. La précision de la classification est définie comme le pourcentage d'objets correctement classés par le classifieur sur une base de données.

$$\begin{aligned} \text{Précision} &= \frac{\text{correct predictions}}{\text{total predictions}} = \\ &= P(\omega^\top x_i + b \geq 0 \wedge y_i = +1) + P(\omega^\top x_i + b < 0 \wedge y_i = -1) \end{aligned}$$

Sensibilité

La sensibilité est connue sous le nom de taux de vrais positifs. Essentiellement, cela nous informe sur la proportion de cas positifs réels qui ont été prédits comme positifs par notre modèle. C'est le rapport des vrais positifs à tous les positifs.

$$\text{Sensibilité} = \frac{\text{Vrai Positif}}{\text{Vrai Positif} + \text{Faux Negatif}} =$$

Spécificité

La spécificité est connue sous le nom de taux de vrais négatifs. Il nous informe sur la proportion de cas négatifs réels qui ont été prédits comme négatifs par notre modèle. C'est le rapport des vrais négatifs à tous les négatifs.

$$\text{Spécificité} = \frac{\text{Vrai Negatif}}{\text{Vrai Negatif} + \text{Faux Positif}} =$$

L'astuce du noyau: The kernel trick

- Soft-margin SVM¹:

$$\min_{\omega, b, \xi} \frac{1}{2} \sum_{j=1}^d \omega_j^2 + \frac{C}{n} \sum_{i=1}^n g_i(\xi_i)$$

s.t.

$$y_i(\omega^\top \phi(x_i) + b) \geq 1 - \xi_i \quad \forall i = 1, \dots, n$$

$$\omega \in \mathbb{R}^d$$

$$b \in \mathbb{R}$$

$$\xi_i \geq 0 \quad \forall i = 1, \dots, n,$$

- Astuce du noyau. Exemple: $x \in \mathbb{R}^3, \phi(x) \in \mathbb{R}^{10}$

$$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

¹ Cortes, C., Vapnik, V. *Support-vector networks*. Machine learning, 20(3), 273-297.

Definition (Noyau)

$k : X \times X \rightarrow \mathbb{R}$ est un noyau si

- ① k est symétrique: $k(x_1, x_2) = k(x_2, x_1)$.
- ② k est semi-défini positif, c'est-à-dire $\forall x_1, x_2, \dots, x_n \in X$, la "Matrice de Gram" K défini par $K_{ij} = k(x_i, x_j)$ est positif semi-défini. (Une matrice $M \in \mathbb{R}^{n \times n}$ est semi-défini positif si $\forall a \in \mathbb{R}^n$, $a'Ma \geq 0$.)

Noyaux les plus utilisés

Noyau $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle = \phi(x_1)^\top \phi(x_2)$. Noyaux les plus populaires:

- Linear

$$k(x_1, x_2) = \langle x_1, x_2 \rangle$$

- Radial Basis Function (RBF)

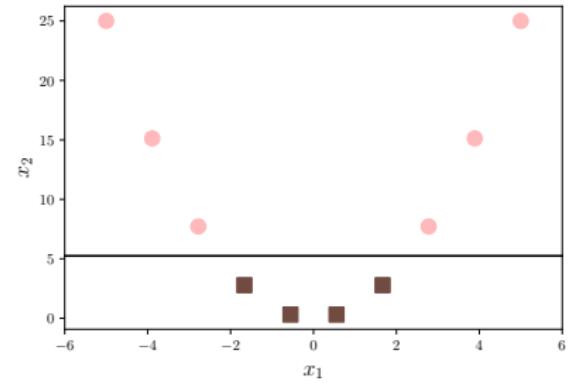
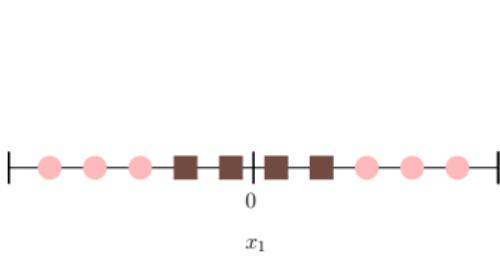
$$k(x_1, x_2) = e^{-\gamma \|x_1 - x_2\|^2}$$

- Noyau polynomial (de dimension d):

$$k(x_1, x_2) = (x_1^\top x_2 + c)^d$$

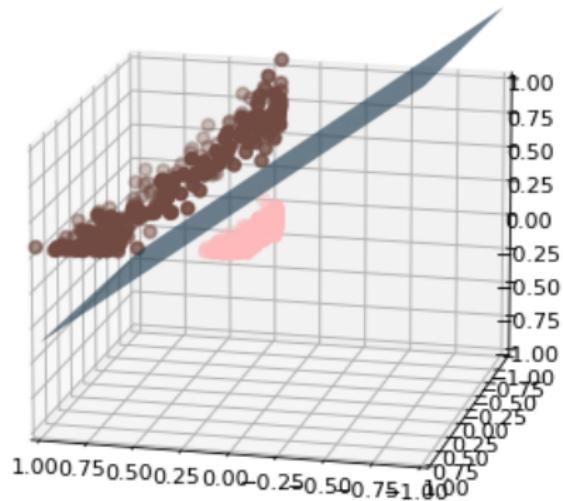
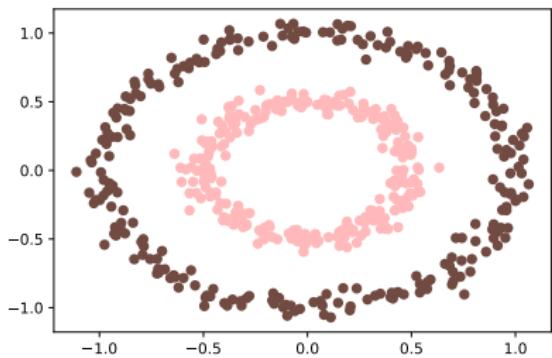
Comment savons-nous que les noyaux aident à séparer les données?

- Dans \mathbb{R}^d , tous les vecteurs indépendants d sont linéairement séparables.
- Si le noyau k est définie positive \rightarrow données linéairement séparables.
- Exemple: $x_1 \in \mathbb{R}$, $\Phi(x_1) = (x_1, x_1^2) \in \mathbb{R}^2$



Exemple: $\mathbb{R}^2 \rightarrow \mathbb{R}^3$

$$x \in \mathbb{R}^2, \phi(x) \in \mathbb{R}^3, \phi(x) = (x_1^2, \sqrt{2}x_1, x_2, x_2^2)$$



Sélection des paramètres

- Étape importante du cycle ML.
- Paramètres: C , paramètres des noyaux.
- Exemple:

$$\gamma \text{ in } e^{-\gamma \|x-y\|^2}$$

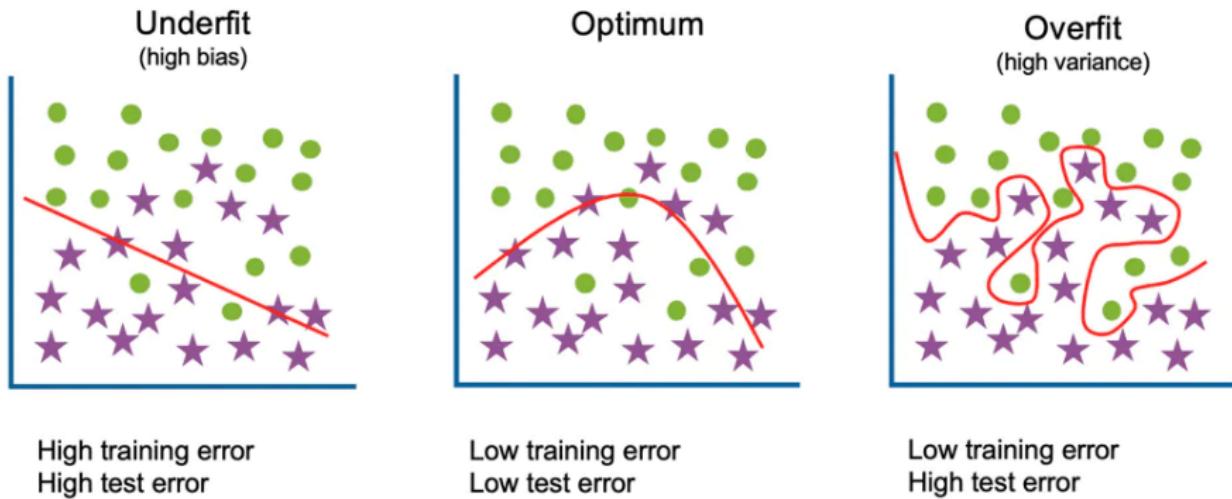
$$c, d \text{ in } (x^\top y + c)^d$$

- Comment les sélectionner?

Et aussi:

- Comment sélectionner les noyaux ? RBF, polynomial,...
- Comment sélectionner les méthodes ? SVM, arbres de décision,...

Underfitting et overfitting

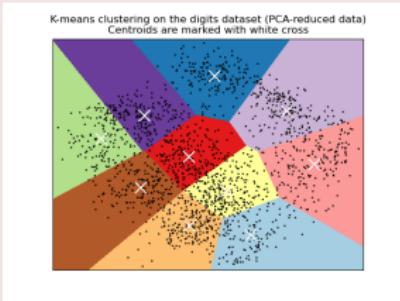


Source d'image: <https://www.ibm.com/cloud/learn/underfitting>

Algorithmes d'apprentissage automatique

- Apprentissage supervisé
 - Les données d'entraînement incluent les résultats souhaités
 - Ensemble de données composé d'exemples étiquetés
- Apprentissage non supervisé
 - Les données d'entraînement n'incluent pas les résultats souhaités
 - Trouver une structure dans certains exemples (pas d'étiquettes!)
- Apprentissage par renforcement
 - Récompenses de la séquence d'actions
 - Prise de décision séquentielle basée sur la rétroaction

Clustering

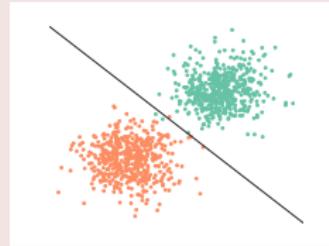


Exemples:

- Systèmes de recommandation
- Analyse des réseaux sociaux

Classification

Sortie: règle de séparation.



Examples:

- Rembourser un prêt
- Acceptation d'entrée à l'université
- Classement des images

Clustering

Trouvez des sous-types ou des groupes qui ne sont pas définis a priori en fonction des mesures.

apprentissage non supervisé

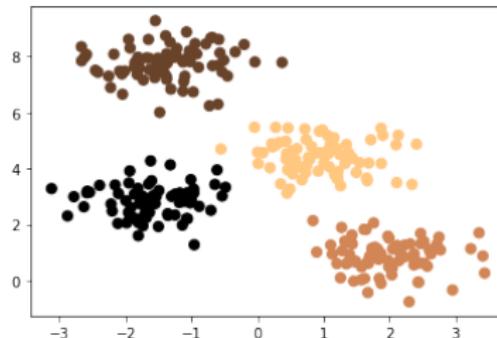
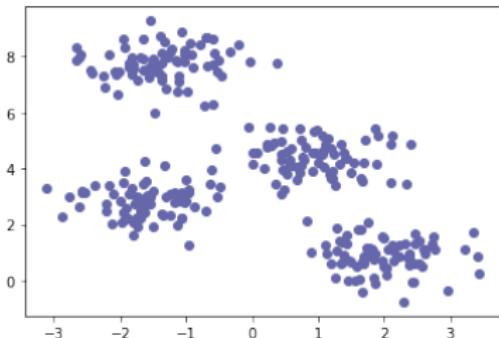
Classification

Utilisez des étiquettes de groupe a priori dans l'analyse pour attribuer de nouvelles observations à un groupe ou à une classe en particulier.

apprentissage supervisé

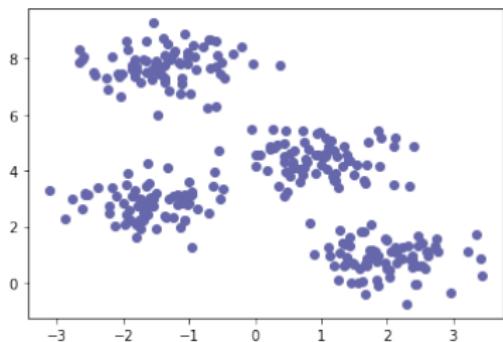
Apprentissage non supervisé

L'algorithme d'apprentissage non supervisé mets en évidence les groupes "naturels" c.-à -d. qui se démarquent significativement les uns des autres.



- ① Combien de groupes?
- ② Calcul de la délimitation des groupes

K-means: Une algorithme de regroupement itératif



- Initialiser: Choisissez K points au hasard comme centres de cluster
- Réitérer:
 - ① Affectez des points de données au centre du cluster le plus proche
 - ② Remplacer le centre du cluster pour la moyenne de ses points affectées
- Arrêtez quand aucun affectation de point change

Partitionnement K-moyennes

Étant donné un ensemble de points (x_1, x_2, \dots, x_n) , on cherche à partitionner les n points en k ensembles (clusters) $C = \{C_1, C_2, \dots, C_k\}$ ($k \leq n$) en minimisant la distance entre les points à l'intérieur de chaque partition :

$$\arg \min_C \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

où μ_i est moyenne des points dans C_i , c'est-a-dire, $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.

Partitionnement K-moyennes: Objective

$$\arg \min_C \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

① Fixer μ , optimiser C :

$$\min_C \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

② Fixer C , optimiser μ :

$$\min_\mu \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

On calcule la dérivée partielle de μ et on la mets à zéro:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

- à partir de la visualisation des données, nous pouvons voir le nombre optimal de clusters.
- Mais la visualisation des données seules ne peut pas toujours donner la bonne réponse.
- Par conséquent, nous définissons:

Inertia

La inertia est la somme au carré des distances des données à leur centre de cluster le plus proche. C'est aussi appelé Within-Cluster Sum of Squares (WCSS).

$$\text{Inertia} = \sum_{i=1}^k \sum_{x_j \in C_i} \|x_j - \mu_i\|^2$$

Exemple en une dimension

Imaginez que vous disposiez de données que vous pouvez tracer sur une ligne et que vous savez que vous deviez les répartir en 3 clusters.



Exemple en une dimension

Dans ce cas, les données forment trois groupes relativement évidents.



Exemple en une dimension

Dans ce cas, les données forment trois groupes relativement évidents.



Mais, plutôt que de se fier à notre œil, voyons si nous pouvons obtenir mathématiquement les trois mêmes clusters.

Pour ce faire, nous utiliserons l'algorithme de clustering K -means

Exemple en une dimension

- ① Sélectionnez le nombre de clusters que vous souhaitez identifier dans vos données. C'est la valeur K dans K -means.



Exemple en une dimension

- ① Sélectionnez le nombre de clusters que vous souhaitez identifier dans vos données. C'est le valeur K dans K -means.



Dans ce cas, nous sélectionnerons $k=3$, c'est-à-dire, nous voulons identifier 3 clusters.

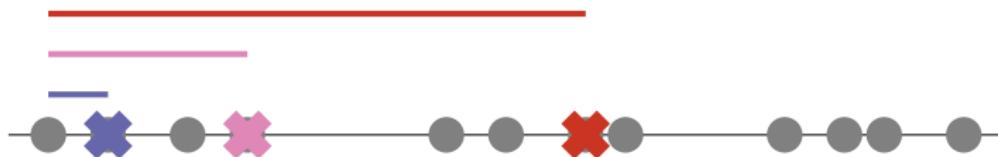
Exemple en une dimension

- ② Sélectionnez au hasard 3 points de données distincts.



Exemple en une dimension

- ③ Mesurer la distance entre le premier point et les trois centres initiaux des clusters.



Exemple en une dimension

- ④ Attribuer le premier point au cluster le plus proche.



Exemple en une dimension

Mesurer la distance entre le deuxième point et les trois centres initiaux des clusters.



Exemple en une dimension

Attribuer le deuxième point au cluster le plus proche.



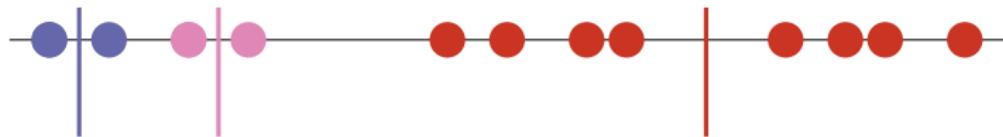
Exemple en une dimension

Les autres points sont plus proches des cluster rouge, donc ils sont donc affectés à celui-ci.



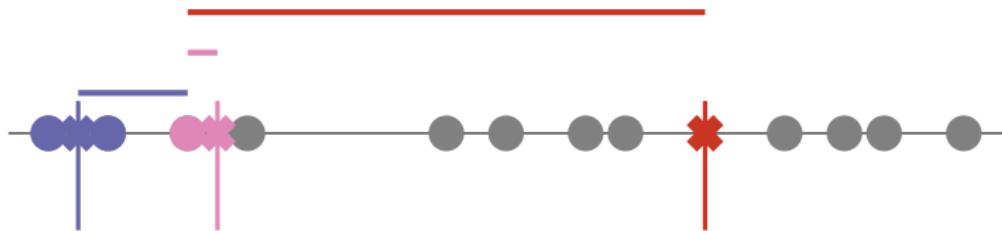
Exemple en une dimension

- ➅ Calculer la moyenne de chaque cluster.



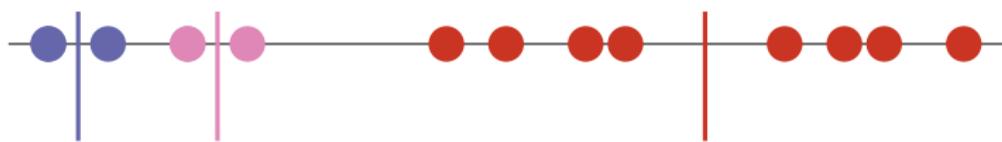
Exemple en une dimension

Ensuite, nous répétons ce que nous venons de faire (mesurer et regrouper) en utilisant les nouveaux centres (valeurs moyennes).



Exemple en une dimension

On obtient le même regroupement. Donc, quand on recalcule les moyennes, les centres ne changent pas, on a trouvé notre solution!



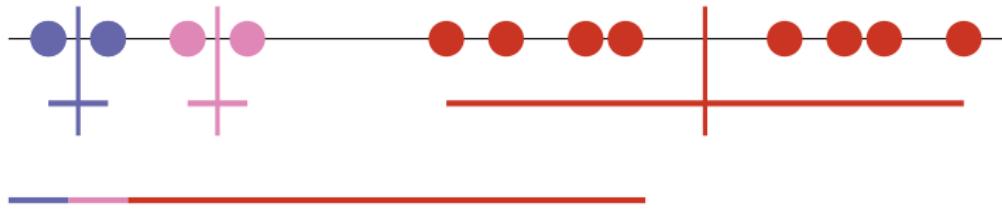
Exemple en une dimension



Le clustering K -means est assez terrible par rapport à ce que nous avons fait à l'œil



Exemple en une dimension



Nous pouvons évaluer la qualité du clustering en additionnant la variation au sein de chaque cluster, en comparant la métrique d'inertia.

Comme K -means est resté bloqué aux optima locaux, la seule option est de garder une trace des différents regroupements, de leur inertia, et de recommencer tout le processus pour différents points de départ.

Exemple en une dimension

Nous choisissons des autres centres initiaux...



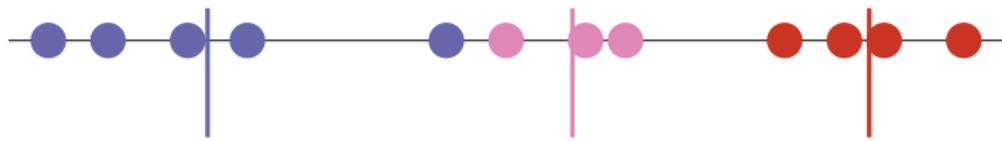
Exemple en une dimension

Et puis regroupez tous les points restants...



Exemple en une dimension

Et puis regroupez tous les points restants, calculez la moyenne de chaque cluster...



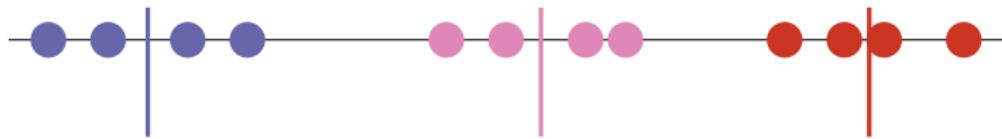
Exemple en une dimension

Et puis regroupez tous les points restants, calculez la moyenne de chaque cluster, puis regroupez en fonction des nouvelles moyennes...



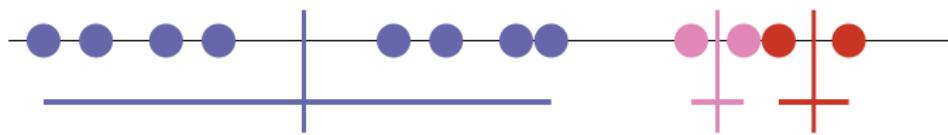
Exemple en une dimension

Et puis regroupez tous les points restants, calculez la moyenne de chaque cluster, puis regroupez en fonction des nouvelles moyennes. Et on répète jusqu'à que les centres ne changent plus.



Exemple en une dimension

À ce stade, K -means sait que le deuxième regroupement est le meilleur à ce jour. Mais il ne sait pas si c'est le meilleur dans l'ensemble, donc il fera quelques clusters supplémentaires (il en fera autant que vous lui dites de faire), puis reviendra et renverra le meilleur



Cluster 1

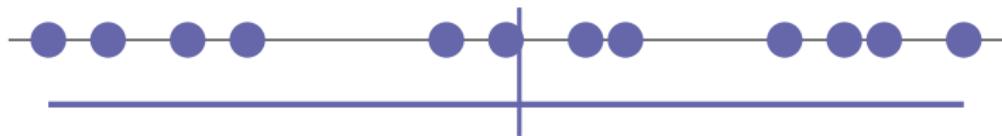
Cluster 2

Cluster 3

← Meilleur clustering!!

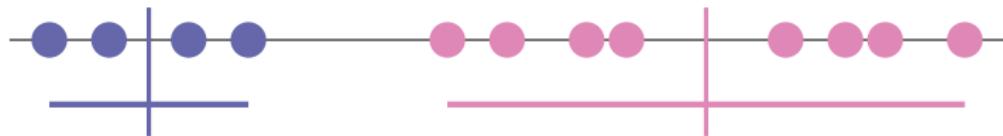
Exemple en une dimension: choix de k

On commence avec $K = 1$

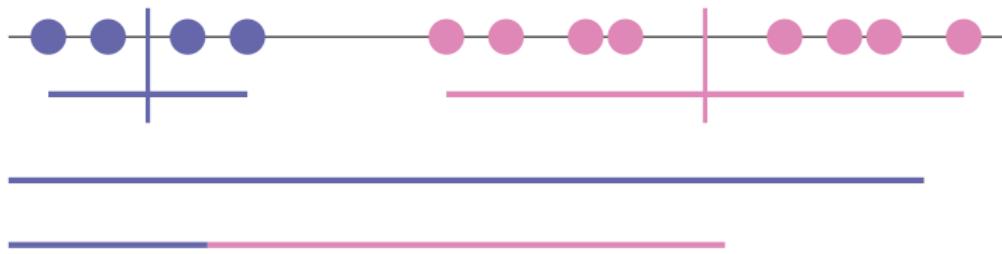


Exemple en une dimension: choix de k

Maintenant on essaie avec $K = 2$



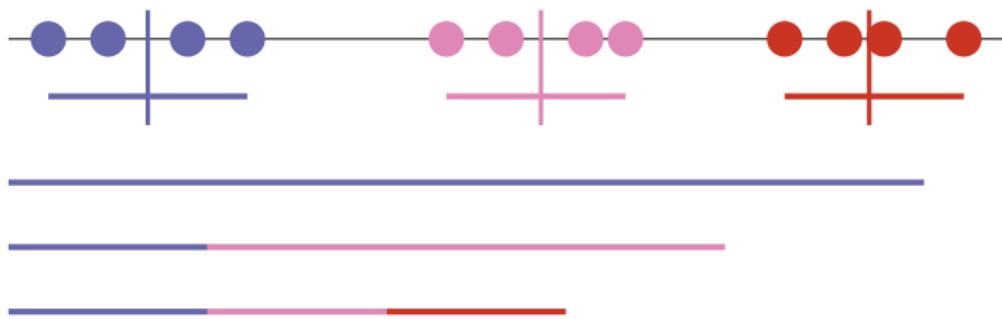
Exemple en une dimension: choix de k



Nous pouvons voir que $K = 2$ est meilleur en comparant la valeur d'inertie.

Exemple en une dimension: choix de k

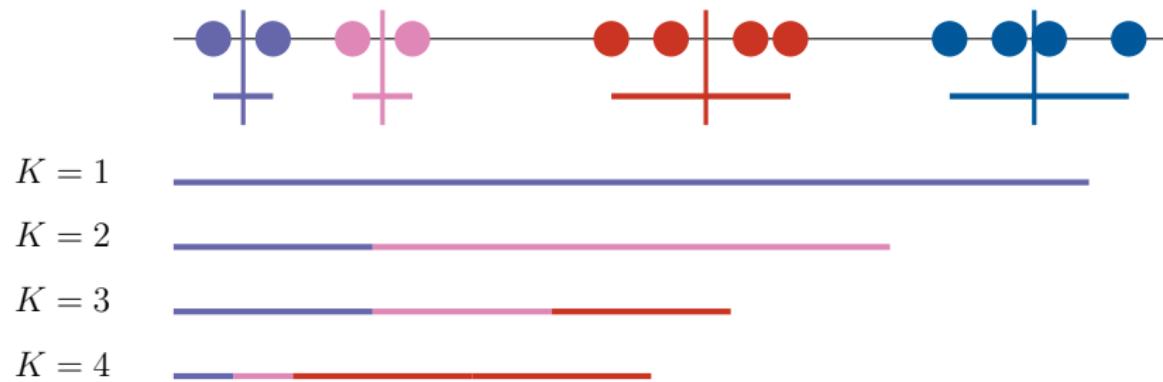
Maintenant on essaie avec $K = 3$



$K = 3$ c'est encore mieux!

Exemple en une dimension: choix de k

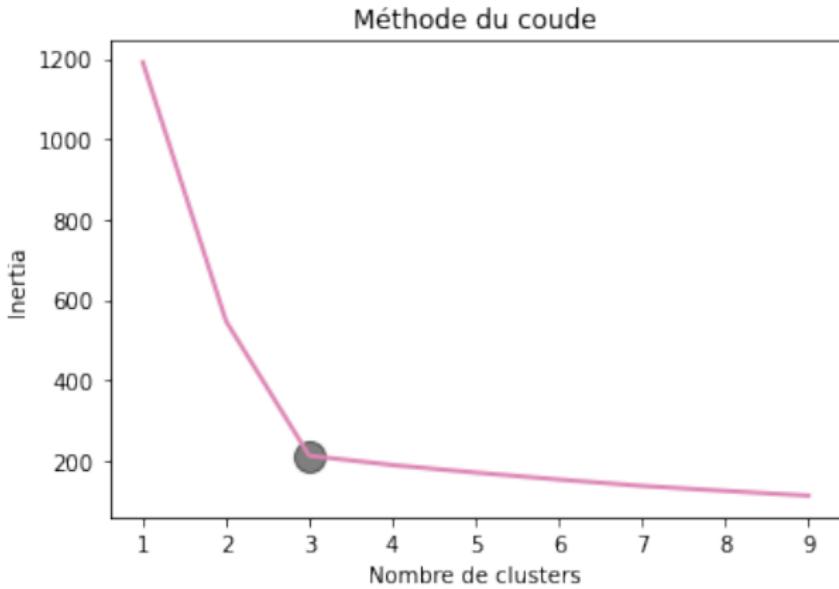
Maintenant on essaie avec $K = 4$



Le valeur d'inertia pour $K = 4$ est encore mieux que pour $K = 3$.

Chaque fois que nous ajoutons un nouveau cluster, la valeur d'inertia diminue. Et lorsqu'il n'y a qu'un seul point par cluster, l'inertia est égale à zéro.

C'est ce qu'on appelle le tracé du coude, et vous pouvez choisir le k optimal en trouvant le coude dans le tracé.



Il y a une énorme réduction de l'inertia lorsque $K = 3$, mais après cela,
l'inertia ne diminue pas aussi rapidement

Exemple: segmentation d'images

Computer vision

- La segmentation d'image est le processus de partitionnement d'une image en plusieurs segments.
- Le but de la segmentation d'une image est de changer la représentation d'une image en quelque chose de plus significatif et plus facile à analyser.
- Il est généralement utilisé pour localiser des objets et créer des limites.

Exemple: segmentation d'images

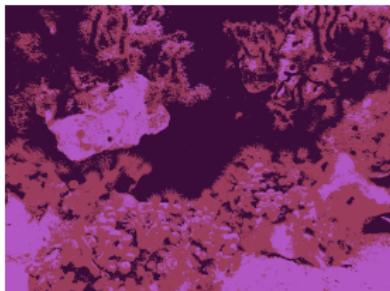
- Dans une image entière nombreuses parties peuvent ne pas contenir d'informations utiles.
- En segmentant l'image, nous pouvons utiliser uniquement les segments importants pour le traitement.
- Une image est un ensemble de pixels donnés. Dans la segmentation d'image, les pixels qui ont des attributs similaires sont regroupés.

K-means pour segmentation: Exemples

- Voitures autonomes. La conduite autonome n'est pas possible sans la détection d'objets qui implique une segmentation.
- Santé. Utile pour segmenter les cellules cancéreuses et les tumeurs à l'aide desquelles leur gravité peut être évaluée.

Exemple: *K*-means pour segmentation

$K = 2$



L'objectif de la segmentation est partitionner une image en régions chacune de qui a raisonnablement visuel homogène apparence.

Original



Exemple: K -means pour segmentation

$K = 2$



Original



Exemple: K -means pour segmentation

$K = 2$



$K = 3$



Original



Exemple: K -means pour segmentation

$K = 2$



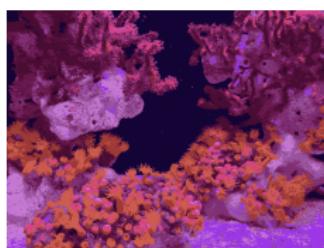
$K = 3$



$K = 10$



Original



Remarque 1: Nombre de clusters k

- Le nombre de clusters k doit être spécifié par l'utilisateur.
- Choisir la meilleure valeur de k pour un ensemble de données est donc un problème.
- Le nombre possible de clusters peut être ambigu dans les applications de clustering d'images, audio, vidéo, etc.
- Il n'existe aucun moyen global de savoir quelle devrait être la valeur de k . Nous pouvons essayer avec des valeurs successives de k .
- Le processus est arrêté lorsque deux valeurs k consécutives produisent des résultats plus ou moins identiques (par rapport à l'inertia).

Remarque 2: Choix des centres

- Les centres initiaux de chaque cluster doivent être spécifiés par l'utilisateur.
- Le choix initial des centres influence la qualité finale du cluster.
- Le résultat peut être bloqué dans des optima locaux, si les centroïdes initiaux ne sont pas choisis correctement.
- Une technique généralement suivie pour éviter le problème ci-dessus consiste à choisir des centroïdes initiaux dans plusieurs exécutions, chacune avec un ensemble différent de centroïdes initiaux choisis au hasard, puis à sélectionner le meilleur clustering (par rapport à certains critères de mesure de la qualité, par exemple inertia).
- Cependant, cette stratégie souffre du problème d'explosion combinatoire en raison du nombre de toutes les solutions possibles.

Remarque 3: Mesure de distance

- Pour attribuer un point au centroïde le plus proche, nous avons besoin d'une mesure de proximité qui devrait quantifier la notion de "plus proche" pour les objets sous clustering.
- Habituellement, la distance euclidienne (norme L2) est la meilleure mesure lorsque les points d'objet sont définis dans un espace euclidien à n dimensions.
- De plus, il peut y avoir d'autres types de mesures de proximité appropriées dans le contexte des applications.
- Par exemple, la distance Manhattan (norme L1), etc.

Remarque 4: Type d'objets à partitionner

- L'algorithme K -means ne peut être appliqué que lorsque la moyenne du cluster est définie (d'où son nom K -means).
- Le calcul de la moyenne suppose que chaque objet est défini avec un ou des attribut(s) numérique(s).
- Nous ne pouvons pas appliquer les K -means aux objets qui sont définis avec des attributs catégoriels.
- L'algorithme K -means n'est pas applicable aux données catégorielles, car les variables catégorielles sont discrètes et n'ont pas d'origine naturelle. Donc, calculer la distance euclidienne pour un tel espace n'a pas de sens.
- K-mode,...

Bibliography

- Cortes, C., Vapnik, V. *Support-vector networks*. Machine learning, 20(3), 273-297.
- Brooks, J.P. *Support vector machines with the ramp loss and the hard margin loss*. Operations Research: 59(2), 467-479 (2011).