

30C - Informatique

Cours 2 - Machine learning

A. Ridard

A propos de ce document

- Pour naviguer dans le document, vous pouvez utiliser :
 - le menu (en haut à gauche)
 - l'icône en dessous du logo GyYv
 - les différents liens
- Pour signaler une erreur, vous pouvez envoyer un message à l'adresse suivante : anthony.ridard@eduvaud.ch

- 1 Introduction
- 2 Classification
- 3 Régression
- 4 Clustering

Entre statistique et informatique, le **machine learning**¹ est une famille d'algorithmes permettant à une machine de « résoudre » certains problèmes, de manière automatique à partir d'une « grande » quantité de données.

On distingue trois branches qui correspondent à des problèmes différents :

- l'apprentissage **supervisé** cherche à « prédire » une information (inconnue) pour un individu², à partir d'un grand nombre d'individus observés pour lesquels on dispose de cette information.
- l'apprentissage **non supervisée**³ vise à regrouper les individus qui se « ressemblent », qui ont le même « profil ».
- l'apprentissage **par renforcement**⁴ permet à un agent autonome, disons une intelligence artificielle comme ChatGPT, de trouver le meilleur comportement, au travers d'expériences itérées fournissant des récompenses positives ou négatives selon les succès ou échecs constatés.

1. apprentissage automatique en français

2. Au sens statistique du terme

3. Il s'agit du clustering que nous aborderons à la fin de ce cours

4. Nous ne traiterons pas ce type d'apprentissage qui sort largement du cadre de ce cours !

Pour présenter les deux types d'apprentissage supervisé, nous avons besoin de formaliser un peu les choses.

L'étude d'un phénomène réel « complexe » commence souvent par une phase de **modélisation** qui consiste à transformer ce problème de la « vraie vie » en un problème « plus simple ». Cette **approximation simplificatrice** de la réalité se limite à certaines caractéristiques, avec lesquelles on espère pouvoir déterminer une solution.

Ces caractéristiques sont appelées des **variables**, puisque leurs valeurs varient selon les **individus**. Plus précisément, nous supposons qu'un individu x est caractérisé par p **variables prédictives**⁵ X_1, X_2, \dots, X_p , autrement dit un individu x sera vu comme un vecteur (x_1, x_2, \dots, x_p) .

Si l'on note Y la **variable cible**, appelée aussi **étiquette**, c'est à dire l'information que l'on cherche à prédire, le problème consiste à approcher au mieux la valeur de Y pour un « nouvel » individu $x = (x_1, x_2, \dots, x_p)$, à partir d'un échantillon $x^{(1)}, x^{(2)}, \dots, x^{(N)}$ de N **individus étiquetés**.

5. On note en majuscule les variables et en minuscules les valeurs prises par ces variables

En résumé, nous souhaitons approcher le y de $x = (x_1, x_2, \dots, x_p)$ à partir des données :

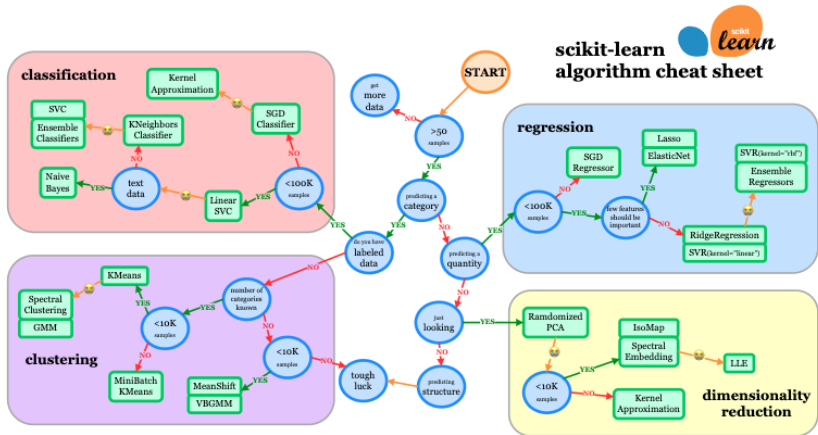
$$\begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_p^{(1)} & \left| & y^{(1)} \right. \\ x_1^{(2)} & x_2^{(2)} & \dots & x_p^{(2)} & \left| & y^{(2)} \right. \\ \vdots & \vdots & \vdots & \vdots & \left| & \vdots \right. \\ x_1^{(N)} & x_2^{(N)} & \dots & x_p^{(N)} & \left| & y^{(N)} \right. \end{pmatrix}$$

Maintenant que nous avons la forme des données, il convient d'en préciser la nature. Une variable est dite **quantitative** si ses valeurs sont des nombres, et **qualitative** sinon.

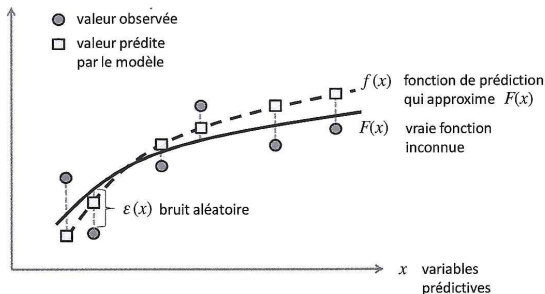
Cette différence permet de distinguer les deux types d'apprentissage supervisée :

- 1 la **régression** si la variable cible est quantitative
- 2 la **classification** si elle est qualitative

En TP, nous utiliserons la bibliothèque **scikit-learn** qui propose cet arbre de choix :



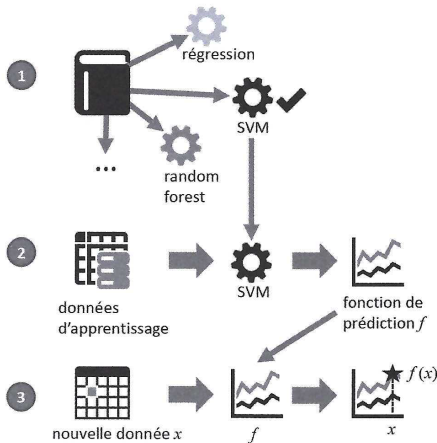
Revenons à l'apprentissage supervisée avec ce schéma ⁶ qui devrait fixer les idées.



La valeur d'une variable cible est la somme d'une **fonction déterministe** F et d'un **bruit aléatoire** ϵ . L'objectif du ML est de trouver une bonne approximation de F à partir des observations. Cette approximation f est la **fonction de prédiction** qui permet d'obtenir une estimation $f(x)$ de $F(x)$.

6. Dans sa forme, il correspond plus à une régression, mais il peut s'étendre à une classification

Donnons maintenant la démarche générale pour saisir le rôle joué par les données.

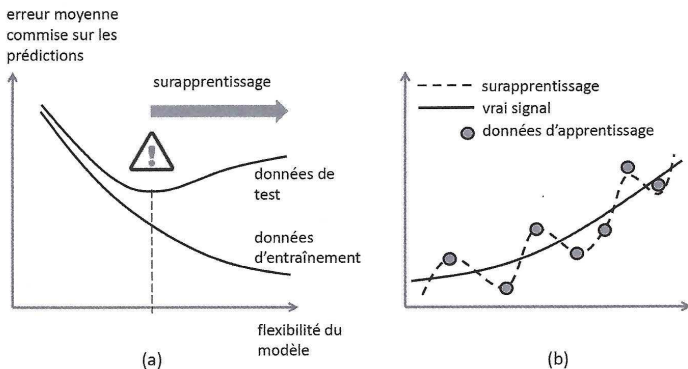


Sélection
 d'un algorithme
 de Machine Learning
 dans une librairie de ML

L'entraînement
 de l'algorithme retenu
 à partir des données
 produit une fonction
 de prédiction f

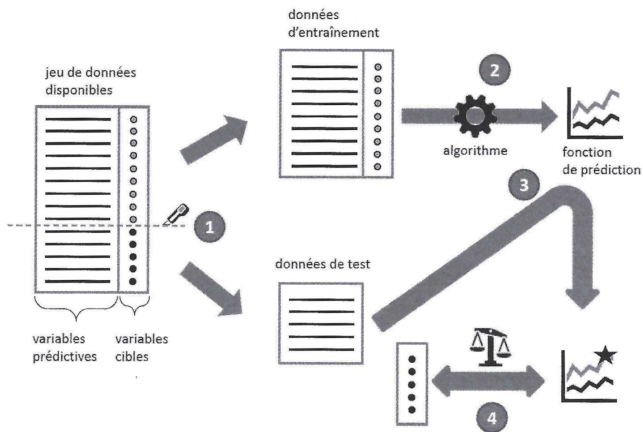
Prédiction pour une
 nouvelle observation

Un risque bien connu des *data scientists* est le **surapprentissage** qui caractérise une situation dans laquelle un modèle de ML reproduit avec une grande précision les données d'apprentissage, mais sans parvenir à **généraliser** de manière satisfaisante.



Avec un modèle « trop flexible », on prend le risque de capter les bruits lors de la phase d'entraînement, éloignant ainsi la fonction de prédiction f du signal F à découvrir pour bien généraliser ou prédire par la suite.

D'où proviennent ces **données de test** permettant d'évaluer la qualité du modèle ?



- 1 Introduction
- 2 Classification**
- 3 Régression
- 4 Clustering

1 Introduction

2 Classification

- k plus proches voisins
- Régression logistique
- SVC

3 Régression

- k plus proches voisins
- Régression linéaire (régularisée)
- SVR

4 Clustering

- Centres mobiles

Pour présenter simplement cet algorithme, nous allons utiliser le problème de la « carte scolaire » qui a pour vocation d'affecter chaque élève dans un gymnase proche de son domicile. Mais d'autres facteurs comme les capacités d'accueil des établissements et les données démographiques des quartiers peuvent intervenir pour équilibrer la répartition globale.

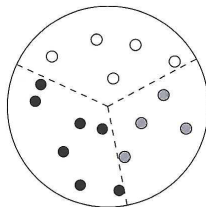
Ce problème est modélisé de la manière suivante :

- Chaque individu (élève ici) est caractérisé par une variable quantitative correspondant à sa position géographique⁷.
- La variable cible est le gymnase d'affectation, et peut prendre ici trois valeurs différentes que l'on associera à trois couleurs différentes : blanc, gris et noir.

Remarquons au passage qu'il s'agit bien d'une classification.

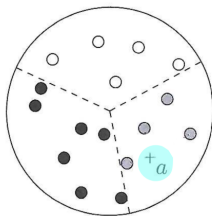
7. Voir deux si l'on distingue latitude et longitude

Une phase d'observation qui se traduit ici par un sondage dans le voisinage fournit la situation suivante.



Notons que les pointillés décrivent la réalité **inconnue** de la carte scolaire c'est à dire la fonction déterministe F que l'on cherche à approcher.

Comment prédire la couleur d'un nouvel individu à partir de ces individus étiquetés?



Comme son nom l'indique, nous allons considérer les k plus proches voisins du nouvel individu a , puis lui attribuer la valeur majoritaire⁸ pour la variable cible.

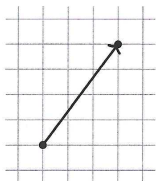
Pour cela, nous devons choisir en amont deux **hyper-paramètres**⁹ :

- une **métrique**, c'est à dire un moyen de mesurer une « distance »
- le nombre k de voisins à considérer

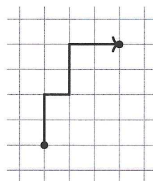
8. Si des valeurs sont ex aequo, on en choisit une au hasard.

9. Ne pas confondre avec les paramètres des modèles paramétriques que nous verrons plus loin

Concernant la métrique, nous allons opter naturellement pour la distance euclidienne, mais il en existe d'autres.



Distance euclidienne : 5



Distance Manhattan : 7

Ces distances sont définies de la manière suivante :

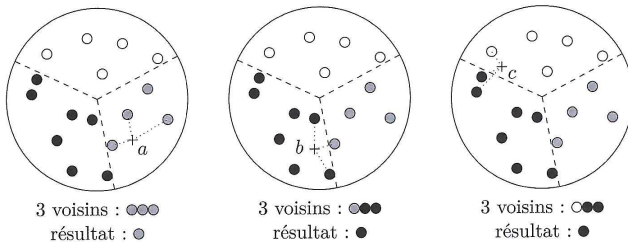
- Distance euclidienne¹⁰ : $d(A, B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$
- Distance Manhattan¹¹ : $d(A, B) = |x_B - x_A| + |y_B - y_A|$

10. A vol d'oiseau

11. A travers les rues de la ville de Manhattan

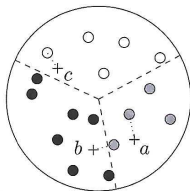
Dans cette méthode, le choix de l'hyper-paramètre k est certainement le plus délicat.

Prenons par exemple $k = 3$:



Dans ce cas, seul l'individu c est mal classé.

Puis comparons avec $k=1$ et $k=5$:

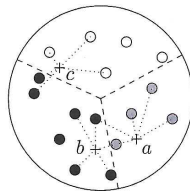


$k=1$

a : ●

b : ●

c : ○



$k=5$

a : ●●●●●

b : ●●●●●

c : ○●●○○

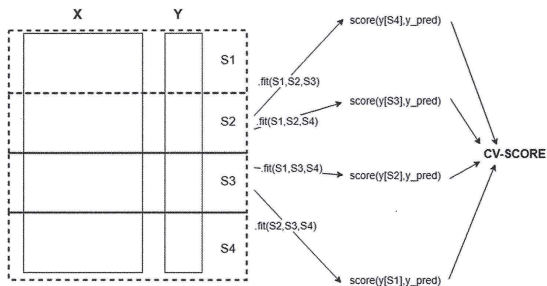
Avec $k=1$, seul l'individu b est mal classé.

Avec $k=5$, tous les individus sont bien classés.

Attention, une trop grande valeur de k a aussi ses défauts.

Que se passe-t-il pour l'individu a avec $k=9$?

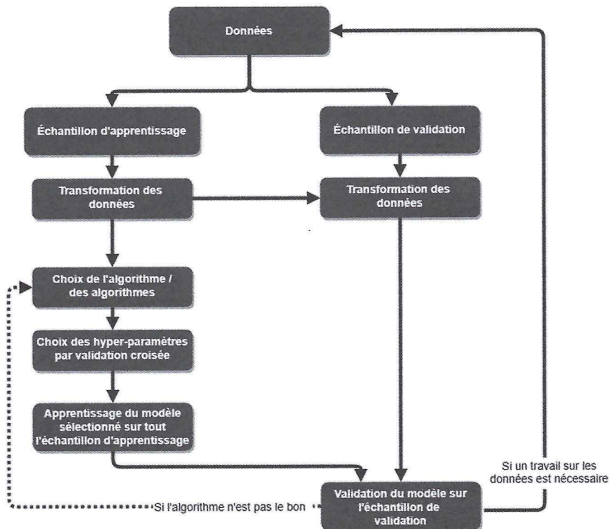
Dans la pratique, pour choisir le meilleur k , nous avons recours à la **validation croisée** qui permet de tester sur une partie des données d'entraînement.



Pour chaque valeur de k , nous obtenons ainsi un **cv-score** qui mesure la qualité¹² de notre modèle, et permet de choisir le meilleur k .

12. Nous précisons plus tard les différentes manières de mesurer cette qualité

Voici à nouveau la démarche générale, mais un peu plus détaillée.



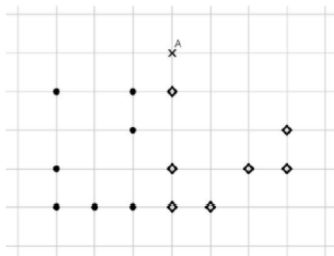
Cet algorithme est apprécié pour sa simplicité conceptuelle, mais il est particulièrement sensible au bruit, et au delà de dix variables prédictives, il est conseillé d'utiliser des techniques de réduction dimensionnelle¹³ pour limiter le coût des calculs. Malgré tout, il est qualifié de « paresseux » car sa prédiction est lente !

13. Tout comme l'apprentissage par renforcement, ces techniques ne seront pas traitées dans ce cours



Dans le quadrillage ci-dessous, 14 points sont représentés :

- 7 de la classe C_1 (ronds)
- 7 de la classe C_2 (losanges)



Pour la distance Manhattan, déterminer la classe du nouveau point A avec :

- 1 $k = 1$
- 2 $k = 3$
- 3 $k = 5$



Considérons les données d'apprentissage suivantes :

$$\left(\begin{array}{ccccc|c} 3 & 5 & 4 & 6 & 1 & 1 \\ 4 & 6 & 10 & 3 & 2 & 2 \\ 8 & 3 & 4 & 2 & 6 & 3 \\ 2 & 1 & 4 & 3 & 6 & 3 \\ 2 & 5 & 1 & 4 & 8 & 2 \end{array} \right)$$

Déterminer^a la classe (1, 2 ou 3) du nouvel individu $x = (3, 12, 4, 7, 8)$ avec :

- 1 $k = 1$
- 2 $k = 3$

a. A l'aide d'un tableur et pour la distance euclidienne

1 Introduction

2 Classification

- k plus proches voisins
- Régression logistique
- SVC

3 Régression

- k plus proches voisins
- Régression linéaire (régularisée)
- SVR

4 Clustering

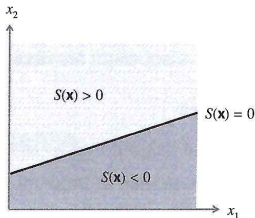
- Centres mobiles

Nous cherchons avec ce modèle à prédire une variable cible **binaire**¹⁴.

Commençons par indiquer qu'il s'agit d'un modèle **linéaire** dans le sens où il utilise une combinaison linéaire des variables prédictives :

$$S(x) = a_1x_1 + a_2x_2 + \dots + a_px_p$$

Cette combinaison linéaire, appelée **Score**¹⁵, cherche à partager l'espace des variables prédictives en deux parties, à l'aide d'un hyperplan¹⁶ : l'une espérant capter les individus prenant la valeur 0, et l'autre ceux prenant la valeur 1.



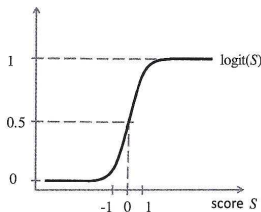
14. A deux valeurs que l'on peut toujours *renommer* 0 et 1

15. Ne pas confondre avec celui qui mesure la qualité d'un modèle !

16. Une droite dans le plan, un plan dans l'espace, ...

Lors de la phase d'apprentissage (ajustement), la machine va chercher à optimiser¹⁷ les paramètres a_1, a_2, \dots, a_p du modèle¹⁸ de sorte que $S(x)$ soit positif lorsque les chances d'appartenir au groupe 1 sont grandes¹⁹, et négatif sinon.

Pour calculer cette probabilité d'appartenance au groupe 1, le modèle utilise la fonction logistique appliquée au score S : $\text{logit}(S) = \frac{1}{1 + e^{-S}}$



-
17. A l'aide de méthodes numériques
 18. Voici donc un modèle paramétrique !
 19. Supérieures à 0,5

Contrairement à l'algorithme des k plus proches voisins, la classification d'un nouvel individu est ici extrêmement rapide puisqu'elle se résume à l'évaluation de la fonction linéaire S . Mais ce modèle possède aussi l'inconvénient de son avantage, puisqu'il n'est pas toujours possible de séparer correctement les individus de manière linéaire!

1 Introduction

2 Classification

- k plus proches voisins
- Régression logistique
- SVC

3 Régression

- k plus proches voisins
- Régression linéaire (régularisée)
- SVR

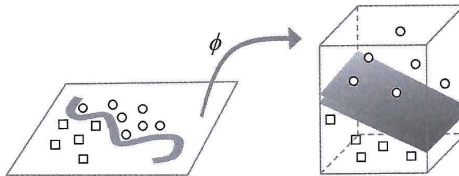
4 Clustering

- Centres mobiles

Comme précédemment, nous cherchons avec ce modèle à prédire une variable cible binaire, mais cette fois de manière **non linéaire**.

Le principe des **SVM** (séparateurs à vaste marge) consiste à construire une bande séparatrice non linéaire de largeur maximale pour séparer les deux groupes (0 et 1), et à l'utiliser pour faire les prédictions.

Pour y parvenir, les SVM utilisent une transformation ϕ qui envoie les points (individus) $x^{(1)}, x^{(2)}, \dots, x^{(N)}$ de l'espace initial à p dimensions dans un espace de dimension plus grande, où ils seront plus faciles à séparer de manière linéaire.



Nous ne donnerons pas plus de détails, mais nous serons amenés à utiliser ces modèles extrêmement puissants en TP.

- 1 Introduction
- 2 Classification
- 3 Régression**
- 4 Clustering

1 Introduction

2 Classification

- k plus proches voisins
- Régression logistique
- SVC

3 Régression

- **k plus proches voisins**
- Régression linéaire (régularisée)
- SVR

4 Clustering

- Centres mobiles

Revenons à notre premier algorithme des k plus proches voisins, et voyons comment il peut être adapté pour résoudre un problème de régression cette fois.

Imaginons que nous souhaitions estimer le prix de vente d'un bien immobilier, la variable cible (le prix au m^2) n'est plus qualitative, mais bien quantitative. Comme ce prix dépend de l'emplacement, de l'emplacement, et aussi de l'emplacement²⁰, c'est plutôt naturel d'utiliser le prix observé des ventes voisines. De manière à comparer tout de même ce qui est comparable, on peut enrichir notre description d'un individu (bien immobilier) en considérant d'autres variables que l'emplacement comme le type de bien (maison ou appartement), sa superficie, son année de construction et sa classe énergétique. Les voisins ne seront plus simplement des voisins au sens géographique du terme, mais des biens qui se ressemblent compte tenu de toutes les caractéristiques.

20. La règle d'or pour un achat immobilier ;)

Maintenant que nous sommes en mesure de choisir les k plus proches voisins, et en imaginant que nous travaillons avec $k = 5$, comment le modèle doit-il estimer le prix à partir des 5 prix voisins suivants ?

2240, 2240, 2339, 2340, 2341

La règle de la majorité n'a évidemment plus beaucoup de sens ici, on préférera utiliser la moyenne de ces valeurs, éventuellement pondérée pour donner plus d'importance aux voisins très proches.

Ce modèle a les mêmes avantages et inconvénients que dans le cas de la classification.

1 Introduction

2 Classification

- *k* plus proches voisins
- Régression logistique
- SVC

3 Régression

- *k* plus proches voisins
- Régression linéaire (régularisée)
- SVR

4 Clustering

- Centres mobiles

Ce modèle paramétrique suppose²¹ que la fonction de prédiction f est de la forme :

$$f(x) = a_1x_1 + a_2x_2 + \dots + a_px_p + b$$

que l'on peut réécrire de manière affine à l'aide du produit scalaire :

$$f(x) = a \cdot x + b$$

et même de manière linéaire en ajoutant une variable X_0 constante égale à 1 :

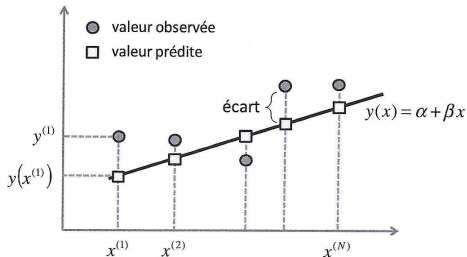
$$f(x) = a_0x_0 + a_1x_1 + a_2x_2 + \dots + a_px_p = a \cdot x$$

avec $a_0 = b$.

Le caractère linéaire du modèle est ainsi justifié !

21. Il s'agit d'une hypothèse très forte qui aura tendance à faire du sous-apprentissage !

Lors de la phase d'apprentissage (ajustement), la machine va chercher à optimiser²² les paramètres $a_0, a_1, a_2, \dots, a_p$ du modèle, de manière à minimiser les erreurs²³ de prédiction (sur le jeu de données d'apprentissage).



22. En résolvant un système d'équations linéaires

23. On cherche à minimiser l'erreur quadratique moyenne, c'est à dire à obtenir les *moindres carrés* entre valeur observée et valeur prédite

Sans contrainte supplémentaire sur l'optimisation des paramètres, ces derniers peuvent prendre de grandes valeurs, rendant l'ajustement du modèle particulièrement sensible dans le sens où de faibles changements dans les données d'apprentissage peuvent produire des modèles très différents !

Pour limiter ce problème, on a recours aux méthodes de régularisation :

- Ridge²⁴ diminue l'importance de certains paramètres
- Lasso²⁵ est plus sévère que Ridge, elle annule certains paramètres
- ElasticNet est une combinaison des deux précédentes

24. basée sur la norme 2

25. basée sur la norme 1

1 Introduction

2 Classification

- k plus proches voisins
- Régression logistique
- SVC

3 Régression

- k plus proches voisins
- Régression linéaire (régularisée)
- SVR

4 Clustering

- Centres mobiles

Comme pour la classification (SVC), les SVM fournissent des modèles de régression (SVR) non linéaires extrêmement puissants que nous utiliserons en TP.

- 1 Introduction
- 2 Classification
- 3 Régression
- 4 Clustering**

1 Introduction

2 Classification

- k plus proches voisins
- Régression logistique
- SVC

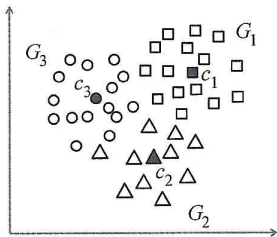
3 Régression

- k plus proches voisins
- Régression linéaire (régularisée)
- SVR

4 Clustering

- Centres mobiles

Cet algorithme des centres mobiles, appelé aussi algorithme des k -moyennes, cherche à regrouper les N individus $x^{(1)}, x^{(2)}, \dots, x^{(N)}$ dans k classes ou clusters **homogènes**²⁶.



26. Plus précisément, il cherche à minimiser l'inertie intra-classe, c'est à dire à minimiser dans chaque classe la somme des carrés des écarts au centre de gravité

Ce partitionnement en classes homogènes est un problème NP-complet, l'algorithme des centres mobiles est une heuristique qui en fournit une solution approchée :

- 1 On initialise le procédé en choisissant au hasard k points (individus) dits centraux.
- 2 On affecte chacun des N points au plus proche des k points centraux.
On a ainsi défini k clusters intermédiaires.
- 3 On calcule les centres de gravité des k clusters intermédiaires qui deviennent les nouveaux points centraux.
- 4 On itère les points 2 et 3 jusqu'à stabilisation²⁷ des points centraux.

La solution obtenue dépend de l'initialisation, il est donc conseillé de répéter plusieurs fois le procédé, mais cela n'est pas un frein puisque cet algorithme est très rapide, même pour un grand volume de données.

27. Il n'existe pas de garantie théorique de convergence de l'algorithme, mais en pratique celle-ci est aisée à vérifier.

Concernant le choix de l'hyper-paramètre k , nous cherchons le plus grand nombre de classes diminuant de manière « significative » l'inertie, c'est à dire un « coude » sur la courbe ci-dessous.

