



# Data mining et Text mining



# Module:

## A. Data Mining

Rappel des concepts de base

Recherche des modèles fréquents, corrélations et association

Clustering

## B. Text Mining

Introduction

Prétraitement d'un corpus

Indexation

Modèles

# Data Mining

---



# Introduction

Les données augmentent à une vitesse phénoménale

Les utilisateurs attendent des informations plus sophistiquées

Comment?

découvrir des informations cachées

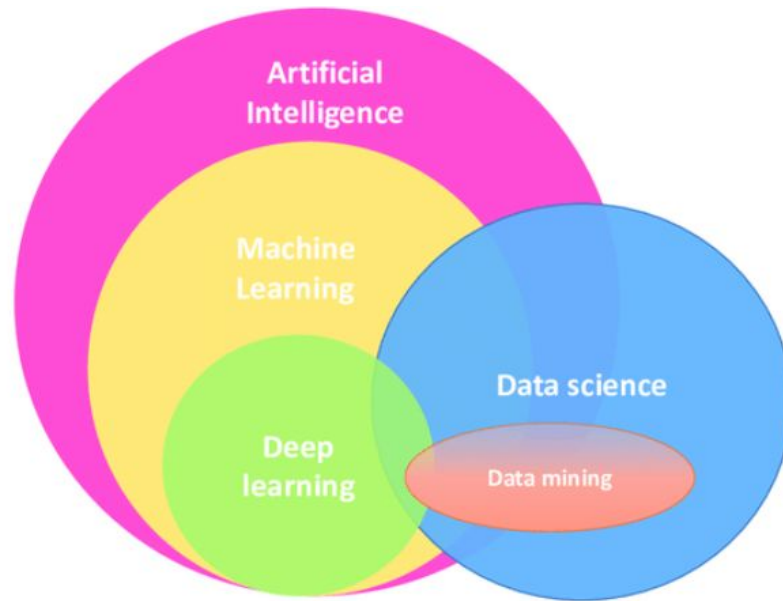
**Data Mining**

# Introduction

Bigger and Bigger volume of Data

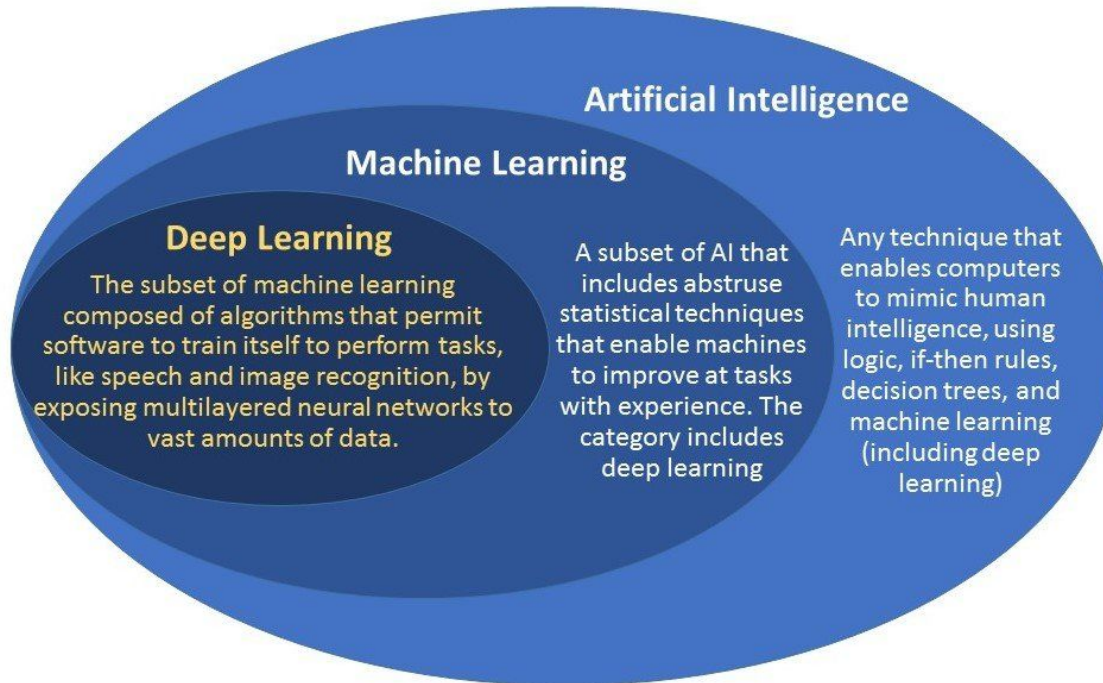


# Introduction





# Introduction





# Data Mining vs Machine Learning

- Data Mining et Machine Learning sont des domaines qui se sont inspirés l'un de l'autre, bien qu'ils aient beaucoup de choses en commun, mais ils ont des objectifs différents.
- Data Mining est effectuée par des humains sur certains ensembles de données dans le but de découvrir des modèles intéressants entre les éléments d'un ensemble de données. Le Data Mining utilise des techniques développées par le Machine Learning pour prédire les résultats.
- Le Machine Learning, quant à lui, est la capacité d'un ordinateur à apprendre à partir d'ensembles de données exploités.
- Les algorithmes de Machine Learning prennent les informations représentant la relation entre les éléments des ensembles de données et construisent des modèles afin de pouvoir prédire les résultats futurs. Ces modèles ne sont rien d'autre que des actions qui seront entreprises par la machine pour arriver à un résultat.





# Machine Learning

- Le Machine Learning est une technique qui développe des algorithmes complexes pour traiter de grandes données et fournir des résultats à ses utilisateurs. Il utilise des programmes complexes qui peuvent apprendre par l'expérience et faire des prédictions.
- Les algorithmes s'améliorent d'eux-mêmes grâce à l'apport régulier de données d'entraînement. L'objectif de l'apprentissage automatique est de comprendre les données et de construire des modèles à partir des données qui peuvent être compris et utilisés par les humains.
- Le terme "Machine Learning" a été inventé en 1959 par Arthur Samuel, un pionnier américain dans le domaine des jeux vidéo et de l'intelligence artificielle, qui a déclaré que "cela donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés".



# Machine Learning

## Apprentissage automatique non supervisé

- L'apprentissage non supervisé ne repose pas sur des ensembles de données entraînés pour prédire les résultats, mais il utilise des techniques directes telles que le regroupement et l'association afin de prédire les résultats. Les ensembles de données entraînés désignent l'entrée pour laquelle la sortie est connue.

## Apprentissage automatique supervisé

- L'apprentissage supervisé s'apparente à l'apprentissage maître-élève. La relation entre la variable d'entrée et la variable de sortie est connue. Les algorithmes d'apprentissage automatique vont prédire le résultat sur les données d'entrée qui seront comparées au résultat attendu.
- L'erreur sera corrigée et cette étape sera effectuée de manière itérative jusqu'à ce qu'un niveau de performance acceptable soit atteint.



# L'intelligence artificielle

L'intelligence artificielle est une branche de la science qui traite de la création de machines intelligentes. Ces machines sont dites intelligentes car elles ont leurs propres capacités de réflexion et de décision, comme les êtres humains.

Parmi les exemples de machines IA, citons la reconnaissance vocale, le traitement d'images, la résolution de problèmes, etc

L'intelligence artificielle, l'apprentissage automatique et l'exploration de données sont des termes fréquemment utilisés dans le monde d'aujourd'hui. Ces mots sont étroitement liés les uns aux autres et sont parfois utilisés de manière interchangeable.



# L'intelligence artificielle vs Data Mining

- L'intelligence artificielle est l'étude visant à créer des machines intelligentes capables de travailler comme les humains. Elle ne dépend pas de l'apprentissage ou de la rétroaction, mais dispose de systèmes de contrôle directement programmés. Les systèmes d'IA trouvent des solutions aux problèmes par leurs propres calculs.
- Les systèmes d'IA utilisent la technique d'exploration des données pour créer des solutions. L'exploration de données sert de base à l'intelligence artificielle. L'exploration de données fait partie de la programmation des codes avec les informations et les données nécessaires aux systèmes d'IA.



# L'intelligence artificielle vs Machine Learning

- Un grand domaine de l'intelligence artificielle est l'apprentissage automatique. On entend par là que l'IA utilise des algorithmes d'apprentissage automatique pour son comportement intelligent. On dit qu'un ordinateur apprend d'une certaine tâche si l'erreur diminue continuellement et si elle correspond aux performances souhaitées.
- L'apprentissage automatique étudiera les algorithmes qui effectueront la tâche d'extraction automatiquement. L'apprentissage automatique est issu des statistiques, mais il ne l'est pas réellement. Comme l'IA, l'apprentissage automatique a également un champ d'application très large.

---

# Rappel des concepts de base



# Rappel des concepts de base

- Définition de la fouille de données
- Processus du data mining
- Quel type de données fouiller ?
- Les tâches de la fouille de données



# Définition de la fouille de données

Le terme de Data Mining est un terme anglo-saxon qui peut être traduit par « exploration de données » ou « extraction de connaissances à partir de données ». Ainsi le Data Mining consiste en une famille d'outils -- qu'ils soient automatiques ou semi-automatiques -- permettant l'analyse d'une grande quantité de données contenues dans une base.

**Objectif :** faire apparaître des corrélations entre des phénomènes en apparence distincts afin d'anticiper des tendances.

(Knowledge discovery in databases - KDD)





# L'utilité de l'exploration de données

Aujourd'hui, le data mining est utilisé dans de nombreux secteurs d'activité comme la recherche, le marketing, le développement de produits, la santé ou encore l'éducation.

Ce processus permet de résoudre rapidement des problèmes qui, jusqu'alors, demandaient énormément de temps pour être résolues manuellement.

L'utilisation de techniques statistiques diverses pour analyser les datas permet aux utilisateurs d'identifier des modèles, des tendances et des corrélations qui n'apparaissaient pas clairement au départ. Grâce aux résultats des différentes analyses successives, ils peuvent prédire ce qui est susceptible de se produire et prendre des mesures pour influencer et maximiser les résultats commerciaux.

Lorsque le forage des datas est employé efficacement, il peut fournir aux organisations un avantage considérable par rapport à leurs concurrents. Il permet en effet de mieux comprendre les clients, de développer des stratégies marketing efficaces, d'augmenter les revenus et de réduire les coûts.



# Données, informations et savoir dans le Data Mining

## Données

Les données sont des faits, des nombres, ou des textes pouvant être traités par un ordinateur. Aujourd'hui, les entreprises accumulent de vastes quantités de données sous différents formats, dans différentes quantités de données. Parmi ces données, on distingue :

1. Les données opérationnelles ou transactionnelles telles que les données de ventes, de coûts, d'inventaire, de tickets de caisse ou de comptabilité.
2. Les métadonnées, à savoir les données concernant les données elles-mêmes, telles que les définitions d'un dictionnaire de données.



# Données, informations et savoir dans le Data Mining

## Informations

Les patterns, associations et relations entre toutes ces données permettent d'obtenir des informations. Par exemple, l'analyse des données de transaction d'un point de vente permet de recueillir des informations sur les produits qui se vendent, et à quel moment ont lieu ces ventes.



# Données, informations et savoir dans le Data Mining

## Savoir

Les informations peuvent être converties en savoir à propos de patterns historiques ou des tendances futures. Par exemple, l'information sur les ventes au détail d'un supermarché peut être analysée dans le cadre d'efforts promotionnels, pour acquérir un savoir au sujet des comportements d'acheteurs. Ainsi, un producteur ou un retailer peut déterminer quels produits doivent faire l'objet d'une promotion à l'aide du Data Mining.



# Applications potentielles

Analyse et gestion du marché

Marketing ciblé

Analyse et gestion des risques

Prévisions, fidélisation des clients, contrôle de la qualité, analyse de la concurrence

Détection et gestion des fraudes

Text Mining (groupes de discussion, courriels, documents) et analyse du Web



# Analyse et gestion du marché

Où sont les sources de données pour l'analyse ?

- Transactions par carte de crédit, cartes de fidélité, coupons de réduction, appels de réclamation des clients,
- Marketing ciblé (trouver des groupes de clients "modèles" qui partagent les mêmes caractéristiques : intérêts, niveau de revenus, habitudes de dépenses, etc.)

Déterminer les habitudes d'achat des clients au fil du temps

Analyse croisée des marchés

- Association/co-relations entre les ventes de produits
- Prédiction basée sur les informations d'association



## Analyse et gestion du marché -2-

### Profilage des clients

- L'exploration de données peut vous indiquer quel type de client achète quels produits (clustering ou classification).

### Identification des besoins des clients

- Identifier les meilleurs produits pour les différents clients
- utiliser la prédiction pour trouver les facteurs qui attireront de nouveaux clients

### Fournit des informations sommaires



# Détection et gestion des fraudes

## Applications

- Largement utilisé dans les domaines de la santé, du commerce, des services de cartes de crédit, des télécommunications (fraude à la carte téléphonique), etc.

## Approche

- Utiliser des données historiques pour construire des modèles de comportement frauduleux et utiliser l'exploration de données pour aider à identifier des cas similaires.

## Exemples

- Assurance automobile : détecter un groupe de personnes qui mettent en scène des accidents pour toucher l'assurance
- Blanchiment d'argent : détecter les transactions financières suspectes.





## Médical / Pharmaceutique

- Diagnostic assisté par ordinateur (CAD) par l'apprentissage de systèmes experts.
- Explication ou prédiction de la réponse d'un patient à un traitement.
- Étude des corrélations entre le dosage dans un traitement et l'apparition d'effets secondaires.



# Autre applications

## Prévention du crime

Plusieurs expériences ont été menées dans ce domaine. Une utilisation aux USA a par exemple été d'identifier les associations de lieu et de plages horaires auxquelles les crimes se produisaient le plus, afin de renforcer la présence policière en conséquence.

## Google, l'un des précurseurs

Google, très tôt, a été utilisateur des techniques de Data Mining, ce que l'on comprend aisément étant donné les volumes de données traités (rappel : 2 000 000 recherches/minute). Quelques outils utilisant le Data Mining :

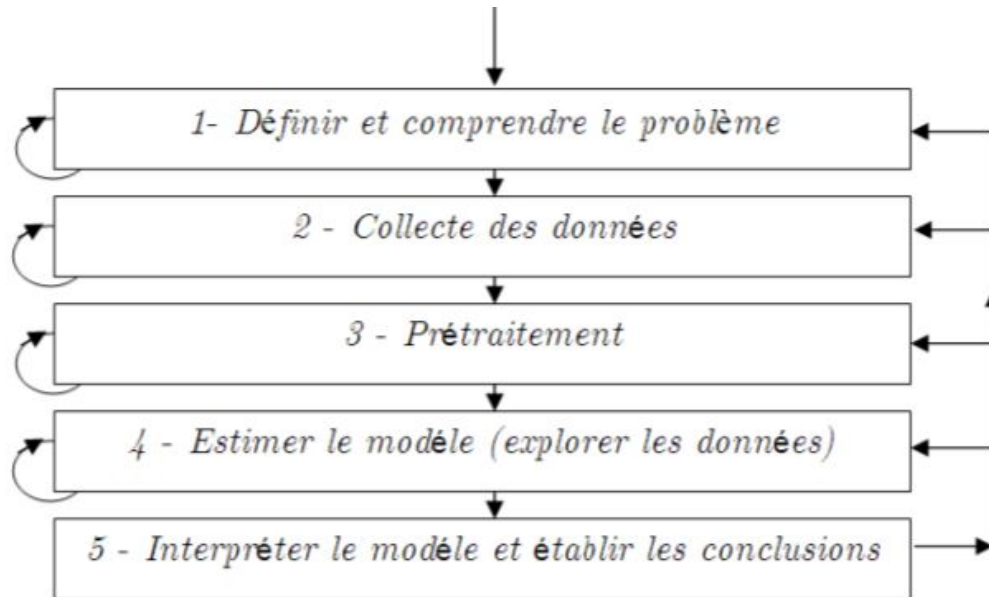
- Google spell checker : le dictionnaire est en fait constitué en fonction des recherches des utilisateurs
- Autocomplétion



# Processus du data mining

Il est très important de comprendre que le data mining n'est pas seulement le problème de découverte de modèles dans un ensemble de donnée. Ce n'est qu'une seule étape dans tout un processus suivi par les scientifiques, les ingénieurs ou toute autre personne qui cherche à extraire les connaissances à partir des données. En 1996 un groupe d'analystes définit le data mining comme étant un processus composé de cinq étapes sous le standard CRISP-DM (Cross-Industry Standard Process for Data Mining)

# Processus du data mining





## Définition et compréhension du problème

Dans la plus part des cas, il est indispensable de comprendre la signification des données et le domaine à explorer. Sans cette compréhension, aucun algorithme ne va donner un résultat fiable. En effet, Avec la compréhension du problème, on peut préparer les données nécessaires à l'exploration et interpréter correctement les résultats obtenus. Généralement, le data mining est effectué dans un domaine particulier (banques, médecine, biologie, marketing, ...etc) où la connaissance et l'expérience dans ce domaine jouent un rôle très important dans la définition du problème, l'orientation de l'exploration et l'explication des résultats obtenus. Une bonne compréhension du problème comporte une mesure des résultats de l'exploration, et éventuellement une justification de son coût. C'est-à-dire, pouvoir évaluer les résultats obtenus et convaincre l'utilisateur de leur rentabilité.



# Collecte des données

Dans cette étape, on s'intéresse à la manière dont les données sont générées et collectées. Ces données n'ont pas toujours le même format et la même structure. On peut avoir des textes, des bases de données, des pages web, ...etc

Généralement les données sont subdivisées en deux parties : une utilisée pour construire un modèle et l'autre pour le tester. On prend par exemple une partie importante (suffisante pour l'analyse) des données (80 %) à partir de laquelle on construit un modèle qui prédit les données futures. Pour valider ce modèle, on le teste sur la partie restante (20 %) dont on connaît le comportement.



# Prétraitement

Les données collectées doivent être "préparées". Avant tout, elles doivent être nettoyées puisqu'elles peuvent contenir plusieurs types d'anomalies : des données peuvent être omises à cause des erreurs de frappe ou à cause des erreurs dues au système lui-même, dans ce cas il faut remplacer ces données ou éliminer complètement leurs enregistrements.

Des données peuvent être incohérentes c-à-d qui sortent des intervalles permis, on doit les écarter où les normaliser. Parfois on est obligé à faire des transformations sur les données pour unifier leur poids

Un exemple de ces transformations est la normalisation des données qui consiste à la projection des données dans un intervalle bien précis  $[0,1]$  ou  $[0,100]$  par exemple.



## Prétraitement -2-

Le prétraitement comporte aussi la réduction des données qui permet de réduire le nombre d'attributs pour accélérer les calculs et représenter les données sous un format optimal pour l'exploration. Une méthode largement utilisée dans ce contexte, est l'analyse en composantes principales (ACP).

Une autre méthode de réduction est celle de la sélection et suppression des attributs dont l'importance dans la caractérisation des données est faible, en mesurant leurs variances.

Plusieurs techniques de visualisation des données telles que les courbes, les diagrammes, les graphes,... etc, peuvent aider à la sélection et le nettoyage des données. Une fois les données collectées, nettoyées et prétraitées on les appelle entrepôt de données (data warehouse).





## Estimation du modèle

Dans cette étape, on doit choisir la bonne technique pour extraire les connaissances (exploration) des données. Des techniques telles que les réseaux de neurones, les arbres de décision, les réseaux bayésiens, le clustering, ... sont utilisées.



## Interprétation du modèle et établissement des conclusions

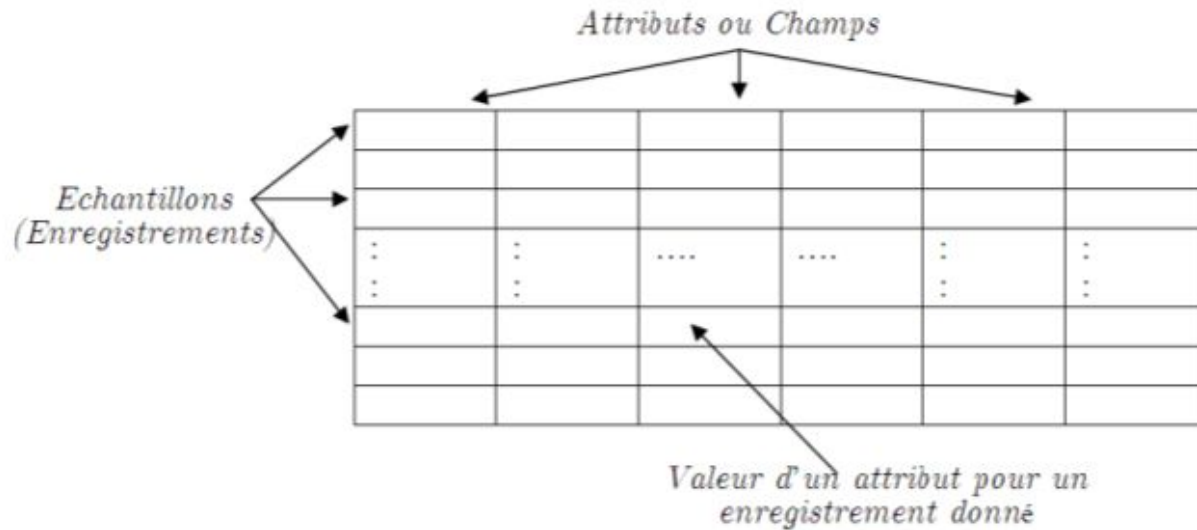
Généralement, l'objectif du data mining est d'aider à la prise de décision en fournissant des modèles compréhensibles aux utilisateurs. En effet, les utilisateurs ne demandent pas des pages et des pages de chiffres, mais des interprétations des modèles obtenus. Les expériences montrent que les modèles simples sont plus compréhensibles mais moins précis, alors que ceux complexes sont plus précis mais difficiles à interpréter.



## Quel type de données fouiller?

Le composant de base d'un processus de data mining est l'ensemble d'échantillons représentant les données à explorer. Chaque échantillon est présenté sous forme de ligne caractérisée par un ensemble d'attributs. Dans le cas des bases de données un échantillon est un enregistrement composé d'un ensemble de champs. Généralement, il convient de représenter les enregistrements sous forme de points dans un espace de  $m$  dimensions où  $m$  est le nombre d'attributs.

## Quel type de données fouiller? -2-





## Quel type de données fouiller? -3-

Les attributs ou les champs sont de deux types : numériques ou catégoriels. Les attributs numériques qui comportent les variables réelles ou entières tel que la longueur, le poids, l'âge, ... sont caractérisés par une relation d'ordre ( $5 < 7.5$ ) et une mesure de distance ( $D(5, 7.5) = 2.5$ ). Les attributs catégoriels (appelées aussi symboliques) tel que la couleur, l'adresse ou le groupe sanguin ne possèdent aucune de ces caractéristiques.



## Quel type de données fouiller? -4-

Théoriquement, plus le nombre d'échantillons est important, meilleure est la précision de l'analyse. Mais en pratique, beaucoup de difficultés peuvent être rencontrées avec les bases de données gigantesques (des milliards d'enregistrements ou des Gigabytes). En effet, Les bases de données de nos jours sont immenses au point où elles épuisent même les supports de stockage, et nécessitent pour être analysées les machines les plus puissantes et les techniques les plus performantes.



## Quel type de données fouiller? -5-

Un premier problème avec les bases de données immenses, est celui de leur préparation à l'analyse, puisque la qualité des données analysées influence directement sur les résultats d'analyse. La préparation doit prendre compte d'un certain nombre de points :

- Les données doivent être précises : les noms doivent être écrits correctement, les valeurs doivent être dans les bons intervalles et doivent être complètes,
- Les données doivent être enregistrées dans les bon formats : une valeur numérique ne doit pas être enregistrée sous format caractère, une valeur entière ne doit pas être réelle,...etc,
- La redondance doit être éliminée ou au moins minimisée, etc



# Les tâches de Data Mining

Beaucoup de problèmes intellectuels, économiques ou même commerciaux peuvent être exprimés en termes des six tâches suivantes :

- La classification.
- L'estimation.
- Le groupement par similitude (règles d'association).
- L'analyse des clusters.
- La description.





# Classification

La classification est la tâche la plus commune de la fouille de données qui semble être une tâche humaine primordiale. Afin de comprendre notre vie quotidienne, nous sommes constamment obligés à classer, catégoriser et évaluer. La classification consiste à étudier les caractéristiques d'un nouvel objet pour l'attribuer à une classe prédéfinie. Les objets à classer sont généralement des enregistrements d'une base de données, la classification consiste à mettre à jours chaque enregistrement en déterminant la valeur d'un champ de classe



## Classification -2-

Le fonctionnement de la classification se décompose en deux phases. La première étant la phase d'apprentissage. Dans cette phase, les approches de classification utilisent un jeu d'apprentissage dans lequel tous les objets sont déjà associés aux classes de références connues. L'algorithme de classification apprend du jeu d'apprentissage et construit un modèle. La seconde phase est la phase de classification proprement dite, dans laquelle le modèle appris est employé pour classer de nouveaux objets.



## L' estimation

L'estimation est similaire à la classification à part que la variable de sortie est numérique plutôt que catégorique. En fonction des autres champs de l'enregistrement l'estimation consiste à compléter une valeur manquante dans un champ particulier. Par exemple on cherche à estimer la lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang. La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation. Et par la suite nous pouvons appliquer ce modèle dans d'autres cas.



## **Le groupement par similitude (Analyse des associations et de motifs séquentiels)**

Le groupement par similitude consiste à déterminer quels attributs "vont ensemble". La tâche la plus répandue dans le monde du business, est celle appelée l'analyse d'affinité ou l'analyse du panier du marché, elle permet de rechercher des associations pour mesurer la relation entre deux ou plusieurs attributs. Les règles d'associations sont, généralement, de la forme "Si , alors ".



# L'analyse des clusters

Le clustering (ou la segmentation) est le regroupement d'enregistrements ou des observations en classes d'objets similaires. Un cluster est une collection d'enregistrements similaires l'un à l'autre, et différents de ceux existants dans les autres clusters. La différence entre le clustering et la classification est que dans le clustering il n'y a pas de variables sortantes. La tâche de clustering ne classe pas, n'estime pas, ne prévoit pas la valeur d'une variable sortantes. Au lieu de cela, les algorithmes de clustering visent à segmenter la totalité de données en des sous groupes relativement homogènes. Ils maximisent l'homogénéité à l'intérieur de chaque groupe et la minimisent entre les différents groupes.



# La description

Parfois le but de la fouille est simplement de décrire ce qui se passe sur une Base de Données compliquée en expliquant les relations existantes dans les données pour premier lieu comprendre le mieux possible les individus, les produit et les processus présents dans cette base. Une bonne description d'un comportement implique souvent une bonne explication de celui-ci. Dans la société Algériennes nous pouvons prendre comme exemple comment une simple description, "les femmes supportent le changement plus que les hommes", peut provoquer beaucoup d'intérêt et promouvoir les études de la part des journalistes, sociologues, économistes et les spécialistes en politiques.



# Techniques de préparation des données

- La qualité des résultats fournis par les techniques de datamining dépendent toujours de la qualité des données utilisées.
- Des données erronées ou manquantes ou même dupliquées peuvent conduire à de faux résultats ou trompeuses.
- Souvent, les données utilisées pour le datamining sont des produits d'interventions humaines caractérisées par les erreurs et l'incomplétude.
- Avant toute chose, on doit passer par une vérification des mauvaises saisies, des manques de données souvent à cause de leur indisponibilité, des déformations, des données bizarres et ainsi de suite.



## Techniques de préparation des données -2-

- Description et résumé des données
- Nettoyage des données
- Intégration
- Transformation
- Réduction des données
- Discrétisation





## Description et résumé des données

Pour qu'un processus de préparation aboutisse à de bons résultats, il est très important qu'on ait une image globale des données à traiter.

Les techniques de résumé des données permettent de mettre l'accent sur les données devant être considérées comme bruits ou comme extrêmes.

L'utilisateur demande souvent de voir une idée générale sur les données leur moyenne, concentrations, dispersion,...etc.



# Description et résumé des données

- Tendance centrale
- Dispersion des données



# Tendance centrale

**La moyenne** La mesure la plus utilisée en général est la moyenne arithmétique qui est donnée par la formule suivante :

$$X = \frac{\sum_{i=1}^N X_i}{N}$$

Une manière pour **accélérer** le calcul de la moyenne est sa distribution : on subdivise l'ensemble des données en sous ensembles et on calcul la moyenne de chaque sous ensemble d'une façon **indépendante** et **parallèle**.



# Tendance centrale

Le problème avec la moyenne arithmétique est sa sensibilité aux données extrêmes par exemple un nombre limité d'employés avec des payes élevées peut donner une moyenne élevée et vice versa.

**Exemple :** soit les données : 14564, 10, 23, 17, 8, 30, 1, 22, 0, 10

La moyenne arithmétique est : **1468.5** qui est loin de la majorité des valeurs.

Pour remédier ce problème on peut par exemple trier les données puis écarter les 10% valeurs supérieures et les 10% valeurs inférieures du calcul de la moyenne.

On trie les valeurs : 0, 1, 8, 10, 10, 17, 22, 23, 30, 14564. On écarte les 10% valeurs supérieures (14564) et inférieures (0), On obtient : 1, 8, 10, 10, 17, 22, 23, 30 et la moyenne devient : 15.125 qui est plus raisonnable.



# Tendance centrale

**La médiane** Une autre mesure qui permet d'éviter ce problème est la médiane : elle consiste à trier les données puis prendre celle du milieu. Elle traite des données qui ont une relation d'ordre : numérique ou symbolique (classement ou mentions A, B, C, D,...).

Exemple la médiane des données : 0, 1, 8, 10, 10, 17, 22, 23, 30, 14564 Est **13.5**,

Le problème avec cette mesure est qu'on ne peut pas partitionner son calcul. Elle nécessite toutes les données pour être calculée, ce qui rend son calcul très coûteux.



# Tendance centrale

Le **mode** représente la valeur la plus fréquente dans un ensemble de données.

Cette mesure convient bien pour les attributs symboliques qui n'ont aucune relation d'ordre.

Dans ce cas on calcule les fréquences des valeurs et on prend celle de la plus grande fréquence.

On peut tomber sur plusieurs valeurs qui ont la même fréquence maximale, on parle ici de données multimodales et on peut prendre une valeur au choix.



# Dispersion des données

La dispersion mesure le degré d'étalement des données sur leur intervalle. Les mesures les plus utilisées pour calculer la dispersion sont le rang, les quartiles, le rang interquartiles et la déviation standard. Ces mesures sont très efficaces pour détecter les données extrêmes ou étranges.



# Dispersion des données

L'**étendue** est très facile à calculer, parce qu'il s'agit simplement de la différence entre les valeurs observées les plus élevées et les plus faibles dans un ensemble de données.

$$Etendue = Val_{max} - Val_{min}$$

La valeur de l'étendue d'un ensemble de données est grandement influencée par la présence d'une seule valeur inhabituellement élevée ou faible à l'intérieur de l'échantillon (une valeur aberrante).





# Dispersion des données

## Variance et écart type

La variance de N valeurs d'un attribut numérique est donnée par la formule :

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

Où  $\bar{x}$  est la moyenne arithmétique des valeurs, l'écart type est la racine carrée de la variance soit  $\sigma$ . La variance ne peut être utilisée qu'avec la moyenne (ni mode ni médiane) et avec des attributs numériques.



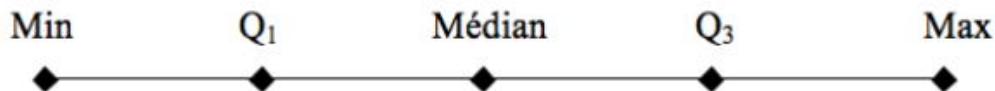
# Dispersion des données

## Rang interquartiles (RIQ)

Soit  $x_1, x_2, \dots, x_n$  un ensemble de valeurs d'un attribut numérique, le rang de cet attribut est la différence entre son max et son min.

On suppose que ces données sont triées dans l'ordre croissant, on appelle  $k$  percentile de données la valeur  $x_i$  supérieure à  $k\%$  de données.

Le médian représente le 50 percentile. Les percentiles les plus utilisés pour la description des données sont les quartiles  $Q_1$  et  $Q_3$  correspondants respectivement aux 25 percentile et 75 percentile.

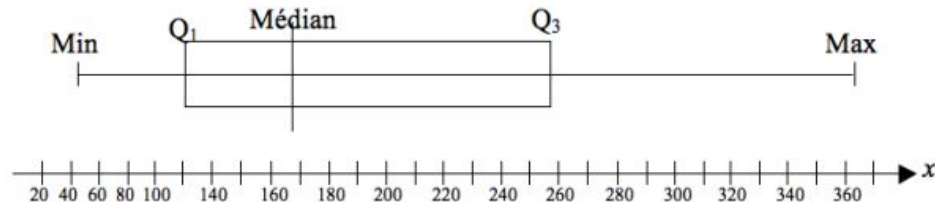


# Dispersion des données

Une première mesure qui donne une idée générale sur la dispersion des données est le rang interquartiles (RIQ) défini par :

$$\text{RIQ} = Q_3 - Q_1$$

Un exemple de l'utilisation du RIQ est de considérer étrange toute donnée qui tombent au moins  $1.5 \times$  RIQ en dessus de  $Q_3$  ou en dessous de  $Q_1$ .





# Nettoyage des données

Les données du monde réel sont caractérisées par leur incomplétude, incohérence et le bruit qui les accompagne. Le nettoyage vise à remplacer les données manquantes, compléter les données incomplètes, corriger les données erronées et filtrer les données bruitées

- Données manquantes
- Elimination du bruit (lissage)



## Données manquantes

Plusieurs raisons justifient l'absence de quelques informations d'une base de données : les erreurs de saisie, les problèmes dus au système lui-même, la non disponibilité des données, ... etc

Attribut 1	Attribut 2	Attribut 3	Attribut 4	Attribut 5
5	B+	12.5	Oui	Alger
	AB-	25.6		
13	B+	3.5	Oui	Blida
12	B+	2.3	Non	Blida



# Données manquantes

- Supprimer carrément les données incomplètes, surtout si elles contiennent plusieurs attributs manquants (supprimer l'enregistrement 2)
- Si les données sont classées, on peut remplacer la donnée manquante par la moyenne (ou le mode de) l'attribut correspondant des enregistrements de sa classe.
- Si les données ne sont pas classées, on remplace la donnée manquante par la moyenne (ou le mode) de l'attribut correspondant dans toute la base.
- Considérer l'attribut de la donnée manquante comme une classe puis construire un modèle de décision pour prédire la donnée manquante
- Laisser les données telles qu'elles et opter pour une technique d'analyse résistante aux manques de données



## Elimination du bruit (lissage)

Le bruit présent dans les données peut se présenter sous forme d'une erreur ou d'une variance aléatoire d'un attribut. Il peut être dû à plusieurs causes :

- Les instruments de mesure utilisés peuvent être défectueux et génèrent des données erronées
- Les erreurs de saisie
- Les erreurs de transmission
- Incohérence dans les conventions de nommage



## Elimination du bruit (lissage) -2-

Le bruit enregistré peut être observé sous différentes formes

- Enregistrement dupliqués
- Données incomplètes (existence d'une partie de la donnée)
- Données incohérentes (Grade = directeur ; Salaire = 20000)
- Données étranges (age = 300 !)





## Elimination du bruit (lissage) -3-

Le lissage consiste à éliminer ou au moins minimiser ces anomalies. Il peut être effectué en plusieurs manières :

- **Par partitionnement (binning)** : en triant les données puis les partitionner en des groupes de taille fixe puis remplacer toutes les données d'un groupe par la moyenne, la médiane ou les bornes de ce groupe
- **Par clustering** en détectant les groupes homogènes et éliminer les données étranges.
- **Par inspection humaine** et informatique combinées : on détecte les valeurs suspectes et on les vérifie manuellement,



# Intégration

Le problème désintégration des données se pose lors de l'utilisation des données de sources multiples : bases de données de différents formats, fichiers, pages web, ...etc. pour construire une base de données unique (datawarehouse) pour l'analyse.

Comment savoir par exemple si un attribut d'une base de données est le même attribut dans une autre base ou un fichier ?

La redondance peut être trouvée aussi si un attribut peut être dérivé d'un autre et doit être éliminé de la base finale.



# Intégration

Le coefficient de Pearson:

$$r_{x,y} = \frac{\sum_{i=1}^n (x_i y_i) - n \bar{x} \bar{y}}{n \sigma_x \sigma_y}$$

Où  $n$  est le nombre d'exemples,  $x_i$ ,  $y_i$  les valeurs des deux attributs dans l'enregistrement  $i$  et  $\sigma_x$ ,  $\sigma_y$  leurs écart-types.

- Si  $r$  se rapproche du zéro, les deux attributs sont faiblement corrélés, et doivent exister tous les deux dans l'analyse.
- Si  $r$  est élevé (proche de 1), la corrélation est forte et l'un des deux peut être écarté de l'analyse.



# Transformation

La transformation des données consiste à les mettre dans des intervalles appropriés à l'analyse. Plusieurs techniques d'analyse nécessitent que les données soient présentées dans des intervalles bien précis tel que  $[-1, 1]$ ,  $[0, 1]$ ,  $[0, 100]$ .

- Échelle décimale
- Normalisation par écart type



# Réduction des données

Le prétraitement comporte aussi la réduction des données qui permet de réduire le volume des données pour accélérer les calculs et **représenter les données** sous un **format optimal** pour l'exploration. La fouille de données peut être très longue sur les données complètes.

**La réduction des données** permet d'obtenir une représentation réduite du jeu de données, plus petite en volume, mais qui produit les mêmes (ou presque) résultats analytiques. Plusieurs stratégies de réduction peuvent être appliquées en data mining :



## Réduction des exemples

- Supprimer les exemples similaires, au sens d'une distance
- Supprimer les exemples étrangers (non significatifs dans certaines analyses)
- Représenter des exemples proches (clusters) par leur moyennes
- Faire un clustering et prendre uniquement les clusters les plus importants



# Sélection des attributs

Les ensembles de données à analyser peuvent contenir des centaines d'attributs voire des milliers, plusieurs parmi eux peuvent être sans importance pour l'analyse ou redondants.

L'**objectif** de la sélection des attributs est de trouver **un ensemble minimum d'attributs** tels que la distribution de probabilité résultant des classes de données soit **aussi proche que possible de la répartition initiale** obtenue en utilisant tous les attributs.



# Sélection des attributs

Les méthodes les plus utilisées pour la sélection des attributs sont les suivantes :

- Sélection progressive (ascendante)
- Élimination progressive (descendante)
- Induction par arbre de décision





# Réduction de dimension

Dans la réduction de dimensions, les données originales sont transformées pour obtenir une forme réduite ou comprimée des données originales.

Parmi les méthodes les plus utilisées dans ce contexte on trouve l'analyse en composantes principales (ACP) qui consiste à remplacer l'ensemble d'attributs d'origine par de nouveaux à variance maximale, non corrélés deux à deux et qui sont des combinaisons linéaires des variables d'origine.

Ces nouvelles variables, appelées composantes principales, définissent des plans factoriels qui servent de base à une représentation graphique plane des variables initiales.



# Échantillonnage

L'échantillonnage peut être utilisé comme une technique de réduction des données, car il permet à un grand ensemble d'être représenté par un échantillon aléatoire (ou sous-ensemble) beaucoup plus petit. Supposons un grand ensemble de données  $D$  contenant  $N$  exemples.





## Discrétisation -2-

Pour réaliser une discrétisation, il faut choisir le nombre de classes et les bornes de classe. La question est donc : comment choisir le nombre  $K$  d'intervalles et les bornes (seuils) de découpage ?

Dans certaines conditions, cela peut être fait en se basant sur les connaissances du domaine, l'expert peut proposer le découpage le plus adapté au problème

---

# Recherche des modèles fréquents, corrélations et associations



## Recherche des modèles fréquents, corrélations et associations

- Concepts de base
- Méthodes efficaces pour la recherche des modèles fréquents
- Types de motifs fréquents
- Passage aux règles d'association
- Analyse des corrélations
- Motifs rares



## Recherche des modèles fréquents, corrélations et associations

Les motifs fréquents sont des motifs ou patterns (tel que les ensembles d'items, les sous séquences, ou les sous structures) qui apparaissent fréquemment dans un ensemble de données.

Par exemple, un ensemble d'items tel que le lait et le pain qui apparaissent souvent dans une base de transactions dans un supermarché, est un ensemble d'items fréquent.

Trouver de tels motifs fréquents joue un rôle essentiel dans la fouille des associations et des corrélations, et représente une tâche importante en data mining et constitue toujours un thème qui attire beaucoup de recherches.



## Recherche des modèles fréquents, corrélations et associations

L'analyse des motifs fréquents trouve son application dans plusieurs domaines :

- L'analyse du panier du marché, pour comprendre les habitudes des clients afin de mieux organiser les rayons d'articles, organiser les promotions, ...etc.
- L'analyse d'ADN en biologie afin de comprendre les propriétés génétiques des espèces.
- L'analyse du climat en météorologie afin de mieux orienter l'agriculture ou choisir l'orientation des pistes des aérodrômes.





# Concepts de base

- Base de données formelle
- Motif
- Connexion de Galois
- Support d'un motif
- Motif fréquent



## Base de données formelle

La version de base de l'extraction de motifs fréquents permet de faire la fouille dans une table d'une base de données relationnelle dont les valeurs sont des booléens indiquant la présence ou l'absence d'une propriété. Une telle base est appelée base de données formelle.

Une base de données formelle est définie par un triplet  $(O, P, R)$  où :

- $O$  est un ensemble fini d'objets.
- $P$  est un ensemble fini de propriétés.
- $R$  est une relation sur  $O \times P$  qui permet d'indiquer si un objet  $x$  a une propriété  $p$  (noté  $xRp$ ) ou non.

## Base de données formelle -2-

Par exemple dans le cas d'analyse du panier dans un supermarché,  $O$  est l'ensemble des transactions d'achat,  $P$  est l'ensemble d'articles et  $R$  est la relation indiquant si un article  $a$  est acheté dans la transaction  $t$ .

- $O = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ .
- $P = \{a, b, c, d, e\}$ .
- $xRp$  si et seulement si la ligne de  $x$  et la colonne de  $p$  se croisent sur un 1 (et pas sur un 0), par exemple :  $x_1Ra$ ,  $x_1Rc$  et  $x_1Rd$ .

$R$	$a$	$b$	$c$	$d$	$e$
$x_1$	1	0	1	1	0
$x_2$	0	1	1	0	1
$x_3$	1	1	1	0	1
$x_4$	0	1	0	0	1
$x_5$	1	1	1	0	1
$x_6$	0	1	1	0	1



# Motif

Un motif d'une base de données formelle  $(O, P, R)$  est un sous-ensemble de  $P$ .

L'ensemble de tous les motifs d'une base est donc l'ensemble des parties de  $P$ , noté  $2^P$ .

On dira qu'un objet  $x \in O$  possède un motif  $m$  si  $\forall p \in m, xRp$ . Pour la base de données en exemple, on a donc :



# Motif

- Motif de taille 0 =  $\emptyset$  ( $C 0 5 = 0$  motifs).
- Motifs de taille 1 =  $\{a\}, \{b\}, \{c\}, \{d\}$  et  $\{e\}$ , qu'on notera, pour simplifier, a, b, c, d et e. ( $C 1 5 = 5$  motifs).
- Motifs de taille 2 = ab, ac, ad, ae, bc, bd, be, cd, ce, de ( $C 2 5 = 10$  motifs)
- Motifs de taille 3 = abc, abd, abe, acd, ace, ade, bcd, bce, bde, cde ( $C 3 5 = 10$  motifs).
- Motifs de taille 4 = abcd, abce, abde, acde, bcde ( $C 4 5 = 5$  motifs).
- Motifs de taille 5 = abcde ( $C 5 5 = 1$  motifs).

Dans la base formelle précédente, x1 possède les motifs :  $\emptyset$ , a, c, d, ac, ad, cd et acd. Parmi l'ensemble global de 2 p motifs, on va chercher ceux qui apparaissent fréquemment. Pour cela, on introduira les notions de **connexion de Galois** et de support d'un motif.



## Connexion de Galois

La connexion de Galois associée à une base de données formelle  $(O, P, R)$  est le couple de fonctions  $(f, g)$  définies par :

$$\left\{ \begin{array}{l} f : 2^P \rightarrow 2^O \\ m \rightarrow f(m) = \{x \in O / x \text{ possède } m\} \\ g : 2^O \rightarrow 2^P \\ x \rightarrow g(x) = \{p \in P / \forall x \in O \ xRp\} \end{array} \right.$$

$g$  est dite duale de  $f$  et  $f$  duale de  $g$ . On dit parfois que  $f(m)$  est l'image du motif  $m$ .



## Support d'un motif

Soit  $m \in 2^P$ , un motif. Le support de  $m$  est la proportion d'objets dans  $O$  qui possèdent le motif :

$$\text{Support} : 2^P \rightarrow [0, 1]$$

$$m \rightarrow \text{Support}(m) = \frac{|f(m)|}{|O|}$$

Par exemple dans la base précédente, on a :

$$\text{Support}(\underline{a}) = \frac{3}{6},$$

$$\text{Support}(\underline{b}) = \frac{5}{6},$$

$$\text{Support}(\underline{ab}) = \frac{2}{6},$$

$$\text{Support}(\emptyset) = 1,$$

$$\text{Support}(P) = 0.$$



# Support d'un motif

## Propriété fondamentale :

Le support est décroissant de  $(2^p, \subseteq)$  dans  $([0, 1], \leq)$ . Autrement dit, si  $m$  est un sous-motif de  $m_0$  ( $m \subseteq m_0$ ) alors  $\text{Support}(m) \geq \text{Support}(m_0)$ . Le support mesure la fréquence d'un motif : plus il est élevé, plus le motif est fréquent. On distingue les motifs fréquents des motifs non fréquents à l'aide d'un seuil  $\sigma_s$ .





## Motif fréquent

Soit  $\sigma_s \in [0, 1]$ .

Un motif  $m$  **est fréquent** (sous-entendu, relativement au seuil  $\sigma_s$ ) si **Support( $m$ )  $\geq \sigma_s$** .

Sinon, il est dit non fréquent.



## Méthodes efficaces pour la recherche des modèles fréquents

Une approche naïve pour l'extraction des motifs fréquents consiste à parcourir l'ensemble de tous les motifs, à calculer leurs nombres d'occurrences (support) et à ne garder que les plus fréquents.

Malheureusement, cette approche est trop consommatrice en temps et en ressources.

En effet, le nombre de motifs est  $2^p$  ( $p$  est le nombre de propriétés), et en pratique, on veut manipuler des bases ayant un grand nombre d'attributs.



# L'algorithme d'Agrawal

L'idée est d'effectuer une extraction par niveaux selon le principe suivant :

- On commence par chercher les motifs fréquents de longueur 1 ;
- On combine ces motifs pour obtenir des motifs de longueur 2 et on ne garde que les fréquents parmi eux ;
- On combine ces motifs pour obtenir des motifs de longueur 3 et on ne garde que les fréquents parmi eux ;
- ... continuer jusqu'à la longueur maximale.



# L'algorithme d'Agrawal

Cette approche s'appuie sur les deux principes fondamentaux suivants (qui reposent sur la décroissance du support) :

1. Tout sous-motif d'un motif fréquent est fréquent.
2. Tout sur-motif d'un motif non fréquent est non fréquent.



# L'algorithme d'Agrawal

---

**Algorithme 1** Apriori

---

**Require:** Base de données de transactions  $D$ , Seuil de support minimum  $\sigma$

**Ensure:** Ensemble des items fréquents

$i \leftarrow 1$

$C_1 \leftarrow$  ensemble des motifs de taille 1 (un seul item)

**while**  $C_i \neq \emptyset$  **do**

    Calculer le Support de chaque motif  $m \in C_i$  dans la base

$F_i \leftarrow \{m \in C_i | \text{support}(m) \geq \sigma\}$

$C_{i+1} \leftarrow$  toutes les combinaisons possibles des motifs de  $F_i$  de taille  $i + 1$

$i \leftarrow i + 1$

**end while**

retourner  $\cup_{(i \geq 1)} F_i$

---

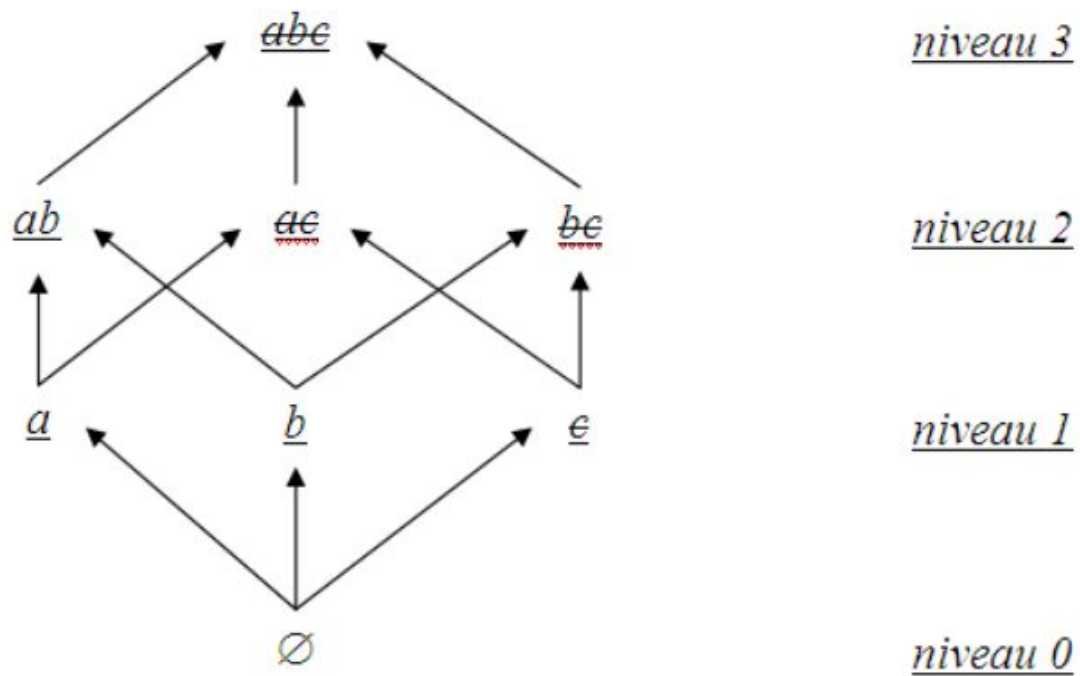


## Exemple

L'application de l'algorithme sur la base donnée en exemple avec  $\sigma = 0.25$  se passe comme suit :

1. Génération de candidats de taille 1 : –  $C1 = \{a, b, c, d, e\}$  – Supports :  $3/6, 5/6, 5/6, 1/6, 5/6$   
D'où  $F1 = \{a, b, c, e\}$  (aucun motif fréquent ne contiendra d).
2. Génération de candidats de taille 2 : Combiner 2 à 2 les candidats de taille 1 de  $F1$  : –  $C2 = \{ab, ac, ae, bc, be, ce\}$  – Supports :  $2/6, 3/6, 2/6, 4/6, 5/6, 4/6$   
 $F2 = C2$  : tous les motifs de  $C2$  sont fréquents.
3. Génération de candidats de taille 3 : Combiner 2 à 2 les candidats de taille 2 de  $F2$  : –  $C3 = \{abc, abe, ace, bce\}$  – Supports :  $2/6, 2/6, 2/6, 4/6$   
 $F3 = C3$  : tous les motifs de  $C3$  sont fréquents.
4. Génération de candidats de taille 4 : –  $C4 = \{abce\}$  – Supports :  $2/6$
5. Génération de candidats de taille 5 :  $C5 = \emptyset$ . Donc,  $F5 = \emptyset$
6. L'algorithme retourne alors l'ensemble des motifs fréquents :  $F1 \cup F2 \cup F3 \cup F4$

Supposons par exemple que  $P = \{a, b, c\}$ . Le treillis des parties de  $P$  est :





## Remarques

Il est parcouru par niveau croissant à partir du niveau  $i = 1$ . Quand un motif n'est pas fréquent, tous ses sur-motifs sont non fréquents. Dans notre exemple  $c$  n'est pas fréquent (il a été barré) et, par conséquent, aucun de ses sur-motifs n'est considéré. On a ainsi élagué le parcours du treillis.

Le seuil  $\sigma$  est fixé par l'analyste. Celui-ci peut suivre une approche itérative en fixant un seuil au départ et, en fonction du résultat, changera la valeur du seuil : Si trop de motifs fréquents ont été trouvés, il augmentera le seuil ; dans le cas inverse, il le diminuera.





# Optimisations

L'algorithme AprioriTID

L'algorithme apriori partitionné

Algorithme de comptage dynamique



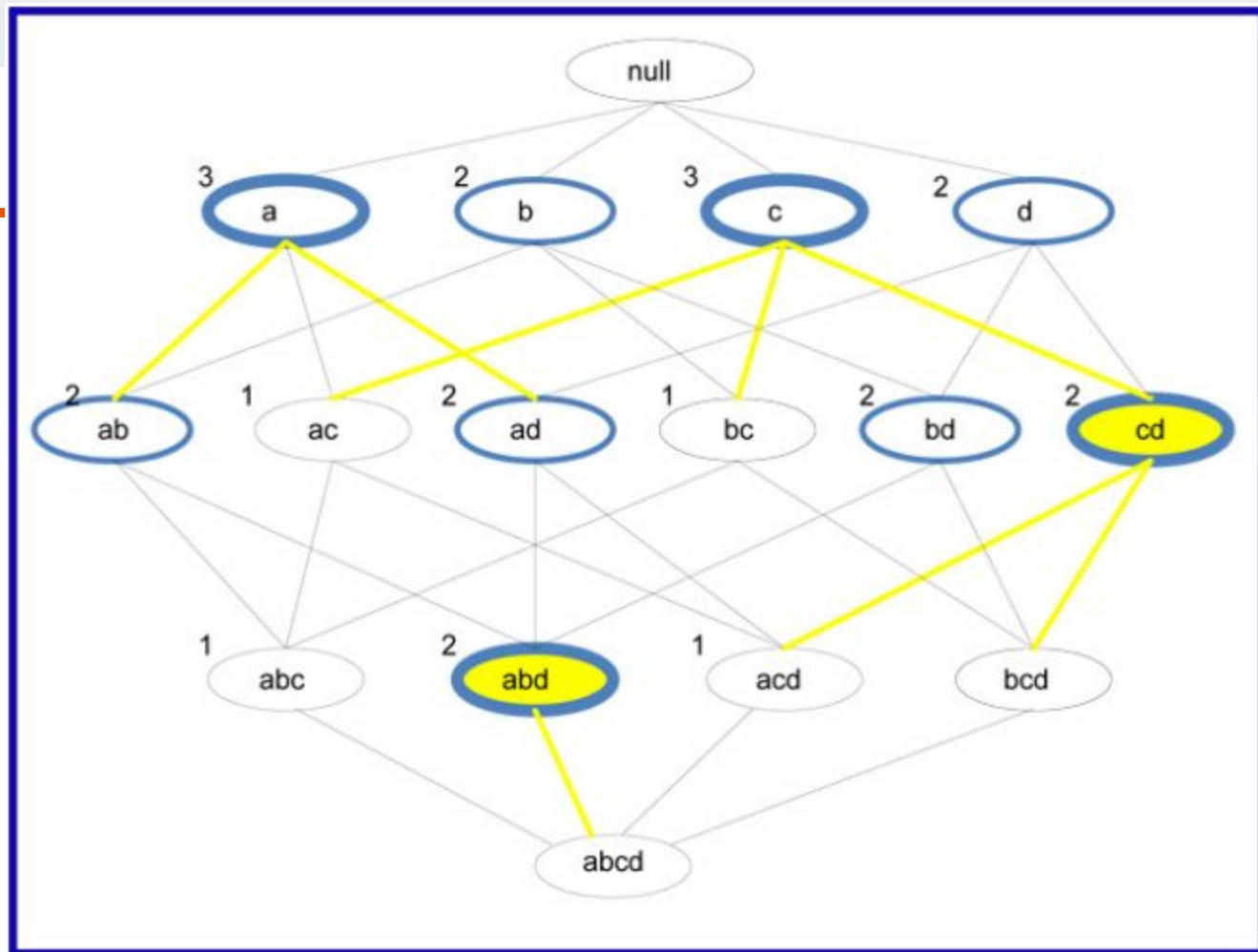
# Types de motifs fréquents

## Motif fréquent fermé:

Un motif fréquent est dit fermé s'il ne possède aucun sur-motif qui a le même support

## Motif fréquent maximal:

Un motif fréquent est dit Maximal si aucun de ses sur-motifs immédiats n'est fréquent.





## Types de motifs fréquents

- Les motifs encerclés par les lignes minces ne sont pas fréquents, les autres le sont.
- Les motifs encerclés par des lignes plus épais sont fermés.
- Les motifs colorés sont maximaux.

Il est clair que : Les motifs maximaux  $\subset$  Les motifs fermés  $\subset$  Les motifs fréquent



## Passage aux règles d'association

Trouver toutes les règles qui existent entre les items fréquents. Par exemple pour une règle telle que  $\{x_1, x_2, x_3\} \Rightarrow x_4$ , il faut premièrement que  $\{x_1, x_2, x_3\}$  et  $x_4$  sont fréquents.

Ensuite, il faut que la confiance de la règle dépasse un certain seuil. La confiance est calculée comme suit :

$$\begin{aligned} \text{Confidence}(\{x_1, x_2, x_3\} \Rightarrow x_4) &= P(x_4 / \{x_1, x_2, x_3\}) \\ &= \frac{\text{Support}(\{x_1, x_2, x_3\} \cup x_4)}{\text{Support}(\{x_1, x_2, x_3\})} \end{aligned}$$

Où le  $\text{Support}(\{x_1, x_2, x_3\})$  est le nombre d'enregistrement où apparaissent les items  $x_1, x_2, x_3$  ensemble.



## Définition d'une règle d'association

Soit  $m_1$  et  $m_2$  deux motifs. Une règle d'association est une implication de la forme :

$$m_1 \Rightarrow m_2$$

$$\text{Où } m_1, m_2 \in 2^P, \text{ et } m_1 \cap m_2 = \emptyset$$

La règle  $m_1 \Rightarrow m_2$  est vérifiée dans la base de donnée  $D$  avec un support  $s$ , où  $s$  est le pourcentage d'objets dans  $D$  contenant  $m_1 \cup m_2$  :

$$\text{Support}(m_1 \Rightarrow m_2) = \frac{\text{Nombre de transactions contenant}(m_1 \cup m_2)}{\text{Nombre total de transactions}}$$

La confiance de la règle  $m_1 \Rightarrow m_2$  est définie par le pourcentage de transactions qui contiennent  $m_1 \cup m_2$  dans les transaction contenant  $m_1$ .

$$\text{Confidence}(m_1 \Rightarrow m_2) = \frac{\text{Support}(m_1 \cup m_2)}{\text{Support}(m_1)} = \frac{\text{Nombre de transactions contenant}(m_1 \cup m_2)}{\text{Nombre de transactions contenant } m_1}$$



## Définition d'une règle d'association

Les règles qui dépassent un minimum de support et un minimum de confiance sont appelées **règles solides**.

Une fois les items fréquents dans une base de donnée sont extraits, il devient simple de générer les règles d'association qui vérifient un minimum de support et un minimum de confiance, comme suit :

- Pour chaque motif fréquent  $I$ , générer tous les sous ensembles non vides de  $I$ ,
- Pour chaque sous-ensemble non vide  $s$  de  $I$ , enregistrer la règle  $(s \Rightarrow I - s)$  si :

$$\text{Confidence}(s \Rightarrow I - s) \geq \text{Min\_Conf}$$

Où  $\text{Min\_Conf}$  est un seuil minimum de confiance.

Puisque les règles sont générées des motifs fréquents, chacune vérifie automatiquement le support minimum.



## Analyse des corrélations

La plupart des algorithmes de recherche des règles d'association utilise un seuil minimum de confiance. Cependant, plusieurs règles intéressantes peuvent être trouvées en utilisant un seuil très faible.

Bien que l'utilisation d'un support minimum empêche l'exploration d'un grand nombre de règles intéressantes, plusieurs règles ainsi trouvées restent inutiles pour l'utilisateur.

On s'intéresse maintenant à la recherche de paires d'items dont le support est faible mais dont la présence est fortement corrélée.

Les données peuvent être vues comme une matrice  $X = (x_{i,j})$  dont chaque ligne représente un individu et chaque colonne représente un item.





## Analyse des corrélation -2-

Typiquement la matrice  $x_{i,j}$  est très creuse : pour chaque individu, il n'y qu'une faible proportion d'items présents (souvent bien moins qu'1 %).

On cherche des paires de produits souvent présents ensemble, i.e. des colonnes similaires.

Le problème à résoudre est donc de trouver le plus efficacement possibles ces paires de colonnes similaires, en faisant l'hypothèse que les items apparaissent rarement (support faible).



## Calcul de la corrélation

Informellement, deux colonnes sont similaires si elles ont généralement des 1 aux mêmes lignes.

On mesure la corrélation de deux colonnes  $x_{.j}$  et  $x_{.k}$  par le rapport entre le nombre de lignes où  $x_{i,j}$  et  $x_{i,k}$  sont égaux à 1 en même temps par le nombre de lignes où l'un des deux seulement vaut 1 :

$$Cor(x_{.j}, x_{.k}) = \frac{|x_{i,j} \wedge x_{i,k}|}{|x_{i,j} \vee x_{i,k}|}$$

La Complexité du calcul de Cor : (O) pour deux colonnes données ; il y a  $O(P^2)$  paires de colonnes ; donc  $O(O \times P^2)$  pour l'ensemble des données.



## Calcul de la corrélation

On définit le type de deux colonnes pour une ligne donnée par a, b, c ou d comme suit :

Pour une paire de colonnes, On note respectivement a, b, c et d le nombre de lignes de type a, b, c et d. On a :

$$Cor(x_{.,j}, x_{.,k}) = \frac{|x_{i,j} \wedge x_{i,k}|}{|x_{i,j} \vee x_{i,k}|} = \frac{a}{a + b + c + d}$$

Type	$x_{i,j}$	$x_{i,k}$
a	1	1
b	1	0
c	0	1
d	0	0



## Calcul de la corrélation

Les applications de cette recherche de paires fortement corrélées à faible support sont :

- Lignes et colonnes sont des pages web et  $x_{i,j} = 1$  si la page  $i$  pointe sur la page  $j$ . Dans ce cas, des colonnes similaires peuvent être des pages traitant d'un même sujet : ces pages sont pointées par les mêmes pages.
- Lignes et colonnes sont des pages web et  $x_{i,j} = 1$  si la page  $j$  pointe sur la page  $i$ . Dans ce cas, des colonnes similaires sont des pages miroirs.
- Les lignes sont des pages web ou des documents, les colonnes sont des mots. Des colonnes similaires indiquent des mots qui apparaissent souvent ensemble dans les mêmes pages.
- Les lignes sont des pages web ou des documents, les colonnes sont des phrases. Les colonnes similaires indiquent des pages miroir ou des plagats.



## Motifs rares

Les motifs rares représentent les motifs qui apparaissent rarement dans un ensemble de données tel que les symptômes non usuels ou les effets indésirables exceptionnels qui peuvent se déclarer chez un patient pour une pathologie ou un traitement donné.

De même que pour les motifs fréquents, certains phénomènes rares dans les bases de données peuvent également véhiculer des connaissances.

La découverte des motifs rares peut se révéler très intéressante, en particulier en médecine et en biologie.



## Définitions

Un motif est dit rare ou infrequent si son support est inférieur ou égal à un support maximum (noté  $\text{max\_supp}$ ).

Généralement, on considère que  $\text{max\_supp} = \text{min\_supp} - 1$ , c'est-à-dire qu'un motif est rare s'il n'est pas fréquent.

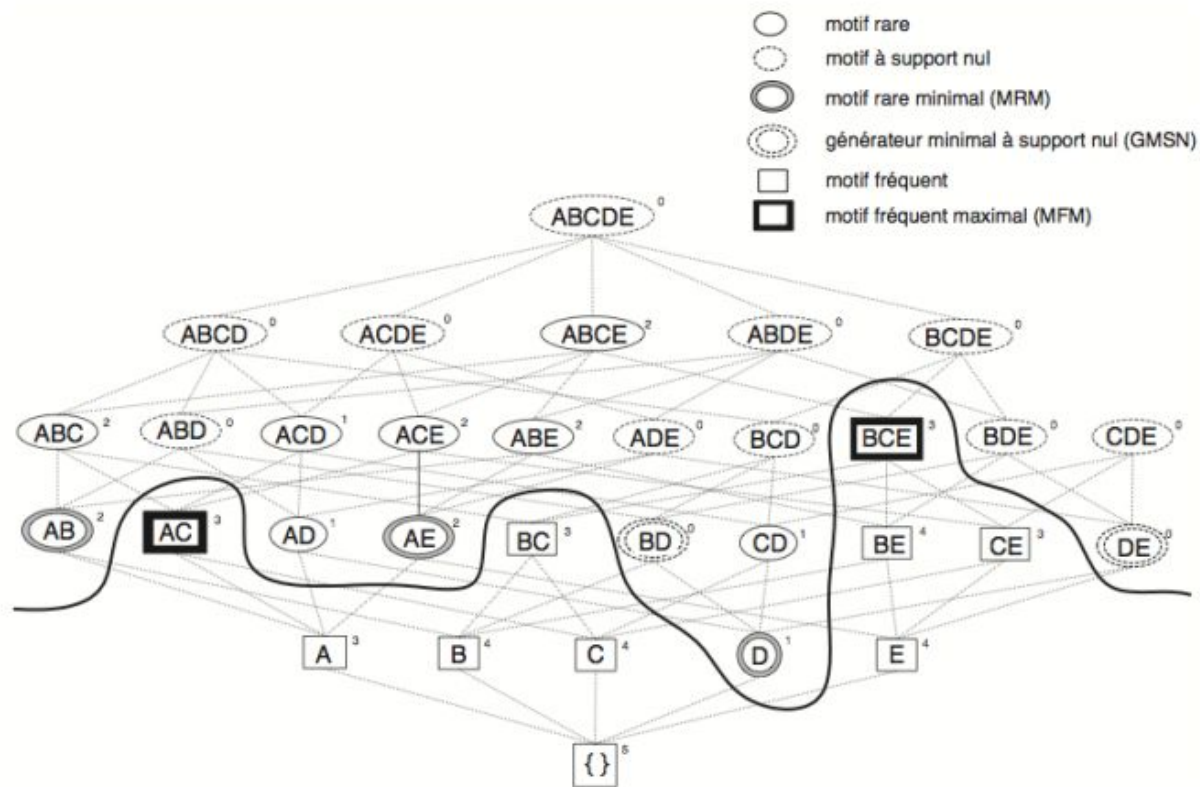


## Recherche des motifs rares

Une première approche pour la recherche des motifs rares est celle de treillis. Prenons l'exemple de la base suivante

	A	B	C	D	E
1	X		X	X	
2		X	X		X
3	X	X	X		X
4	X				X
5	X	X	X		X

En fixant  $\text{min\_supp}$  à 3 ( $\text{max\_supp} = 2$ ), On obtient le treillis suivant :







## Apriori-Rare

De manière surprenante, les motifs rares minimaux peuvent être trouvés simplement à l'aide de l'algorithme bien connu Apriori.

Il est conçu pour trouver les motifs fréquents, mais, puisque nous sommes dans le cas où non fréquent signifie rare, cela a pour "effet collatéral" d'explorer également les motifs rares minimaux.

Quand Apriori trouve un motif rare, il ne générera plus tard aucun de ses sur-motifs car ils sont de manière sûre rares.



## Exercice 1

Soit la table des transactions suivante :

- (a) Représenter cette table par une table formelle.
- (b) Trouver, en utilisant l'algorithme Apriori, les motifs fréquents sachant que  $\sigma_s = 2/9$ .
- (c) Calculer à partir du motifs fréquent le plus long, les règles d'associations les plus fortes sachant que le seuil de confiance est de 70%.

TID	List des items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



## Exercice 2

Une base de données possède cinq transactions.

On suppose que le support minimum est de 60% et la confiance minimale est de 80%.

- (a) Représenter cette table par une table formelle.
- (b) Trouver, en utilisant l'algorithme Apriori, les motifs fréquents.
- (c) Lister toutes les règles d'association fortes correspondant à la métarègle suivante, où X représente les clients et les itemi dénotent les articles achetés.

TID	List des items
T100	M, O, N, K, E, Y
T200	D, O, N, K, E, Y
T300	M, A, K, E
T400	M, U, C, K, Y
T500	C, O, O, K, I, E

$\text{Acheter}(X, \text{item1}) \wedge \text{Acheter}(X, \text{item2}) \Rightarrow \text{Acheter}(X, \text{item3})$

---

# Régression



## Définition

La régression est la méthode utilisée pour l'estimation des valeurs continues. Son objectif est de trouver le meilleur modèle qui décrit la relation entre une variable continue de sortie et une ou plusieurs variables d'entrée. Il s'agit de trouver une fonction  $f$  qui se rapproche le plus possible d'un scénario donné d'entrées et de sorties.

# Régression

Le modèle le plus utilisée actuellement est le modèle linéaire qui est utilisé pour décrire la relation entre une seule variable de sortie et une ou plusieurs variables d'entrée. Ce modèle est appelé la régression linéaire.

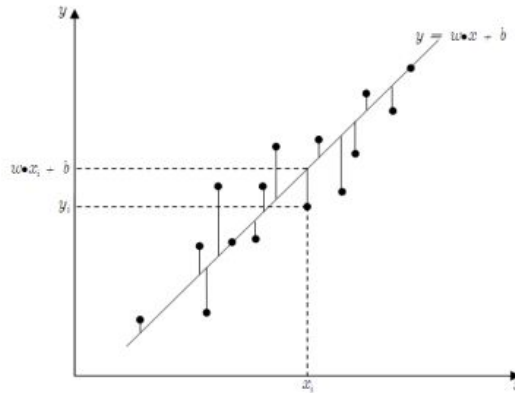


FIGURE 4.1 – Régression linéaire simple



# Régression

Dans les problèmes de régression, les exemples d'entraînement sont associés à des valeurs numériques plutôt qu'à des étiquettes discrètes. Le problème est formulé comme suit : Soit  $D$  l'ensemble d'entraînement défini par :

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \subset \mathbb{R}^m \times \mathbb{R}$$

Le problème consiste à trouver, en utilisant  $D$ , une fonction  $\hat{f} : \mathbb{R}^m \rightarrow \mathbb{R}$  qui vérifie  $\hat{f}(x_i) \approx y_i; \forall i = 1..n$ .  
pratique une telle fonction est difficile à trouver, on recherche plutôt une fonction qui rapproche des  $y_i$ , en d'autre terme qui minimise la différence entre les  $f(x_i)$  et les  $y_i$  :

$$\text{Min} \sum_{i=1}^n (y_i - \hat{f}(x_i))^2$$



# Régression linéaire simple

Souvent,  $\hat{f}$  est considérée comme fonction linéaire :  $\hat{f} = \langle w, x \rangle + b$ , où  $w$  est un vecteur et  $b$  est un scalaire. Le problème revient donc à trouver un hyperplan caractérisé par  $w^*$  et  $b^*$  qui minimise l'écart global entre  $\hat{f}$  et les  $y_i$ :

$$(w^*, b^*) = \underset{w, b}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \langle w, x_i \rangle - b)^2$$

Dans le cas où  $m = 1$ , on parle de régression linéaire simple, sinon on parle de régression linéaire multiple. L'équation à minimiser est :

$$(w, b) = \underset{w, b}{\operatorname{argmin}} \sum_{i=1}^n (y_i - wx_i - b)^2$$





## Régression linéaire simple -2-

Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

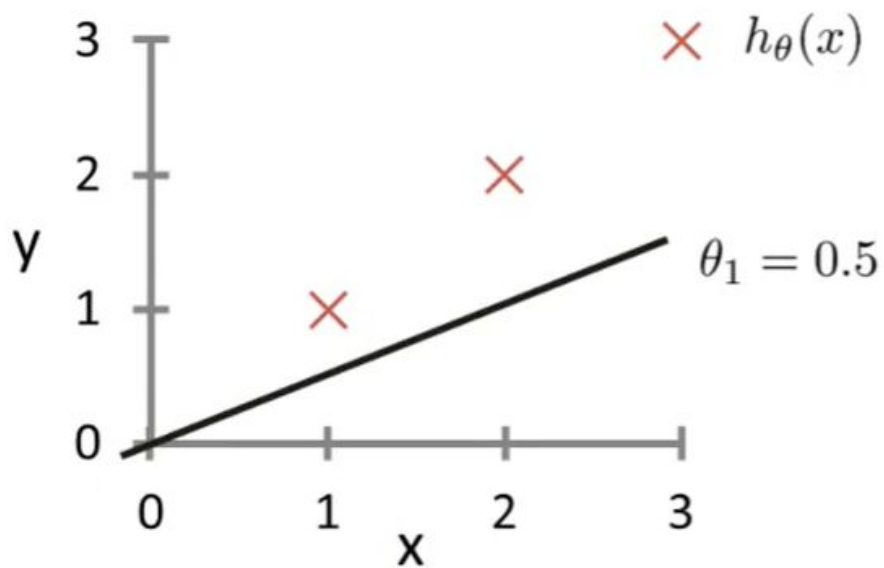
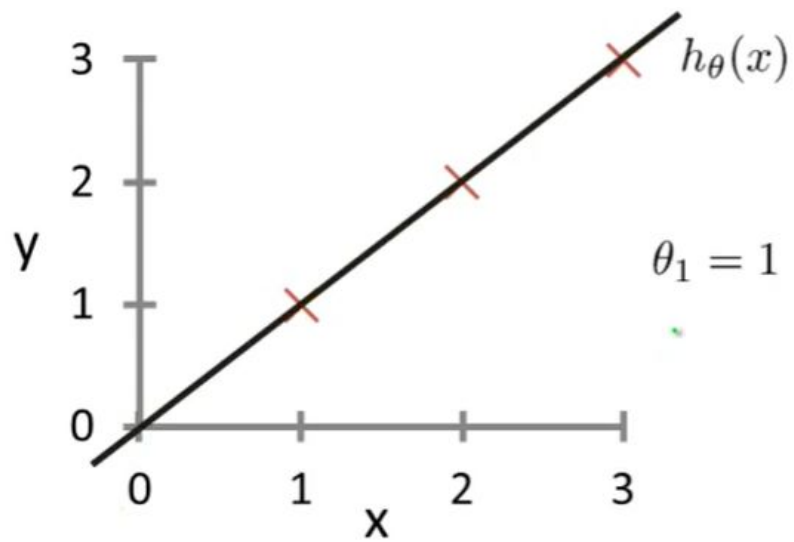
Parameters:

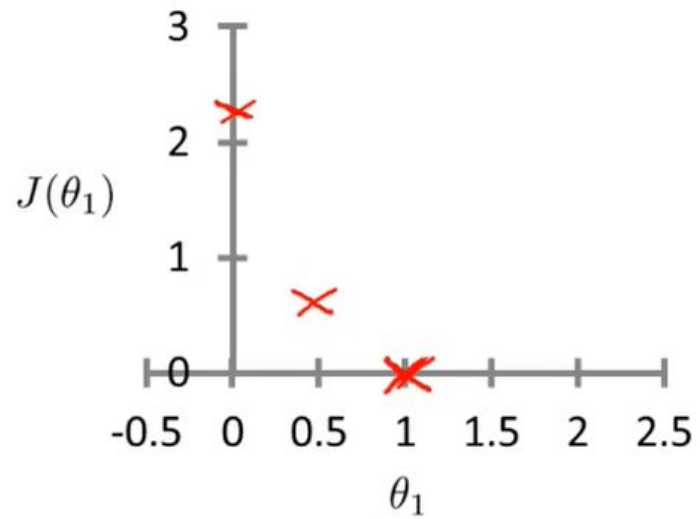
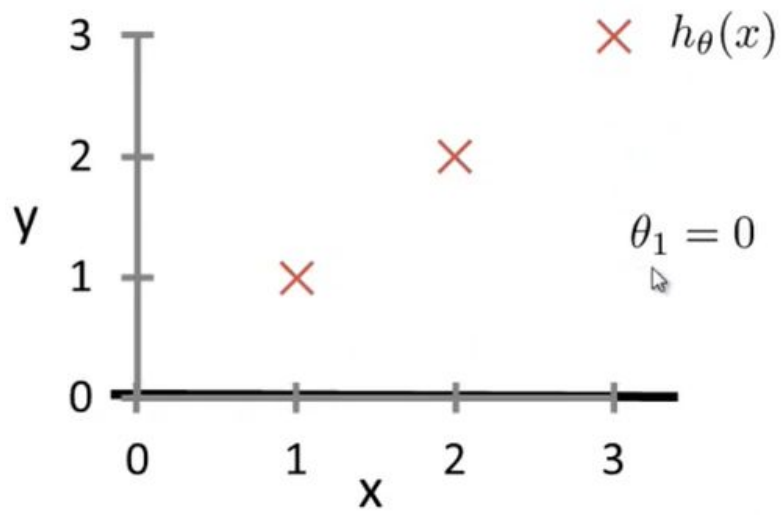
$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Goal: minimize  $J(\theta_0, \theta_1)$   
 $\theta_0, \theta_1$

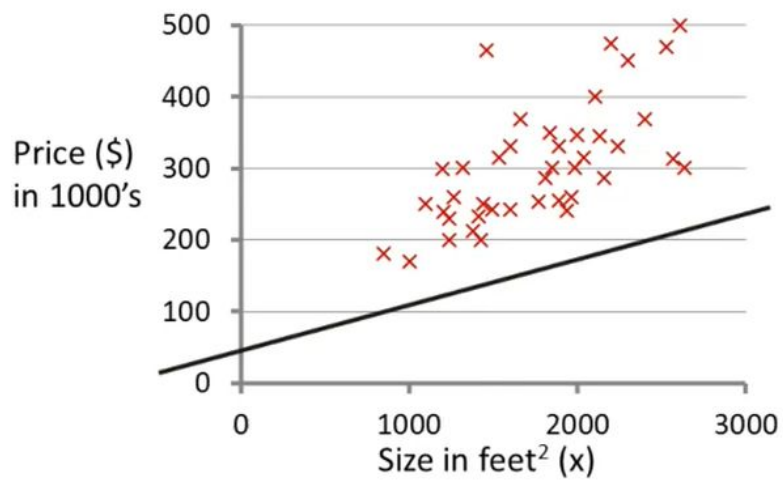




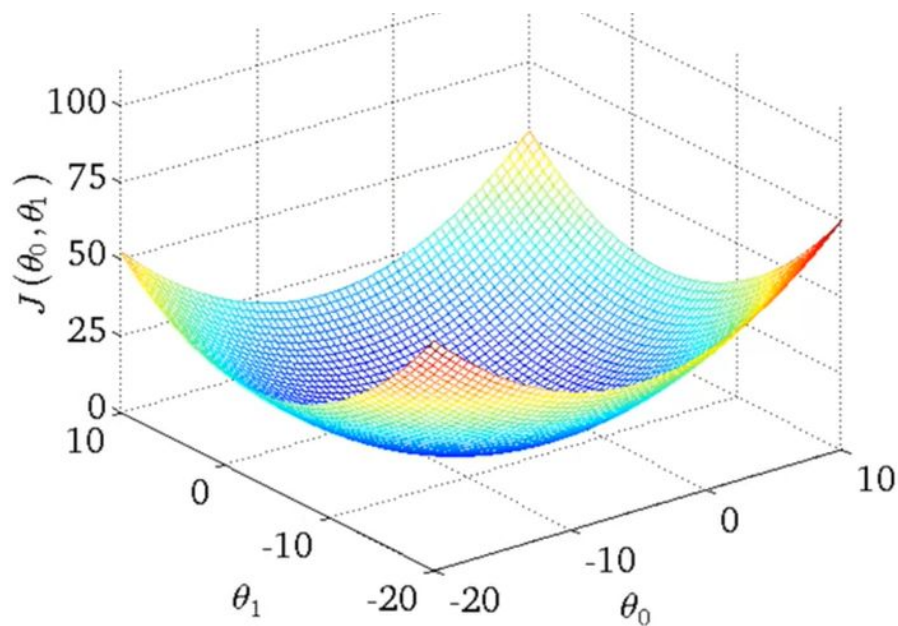


$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )

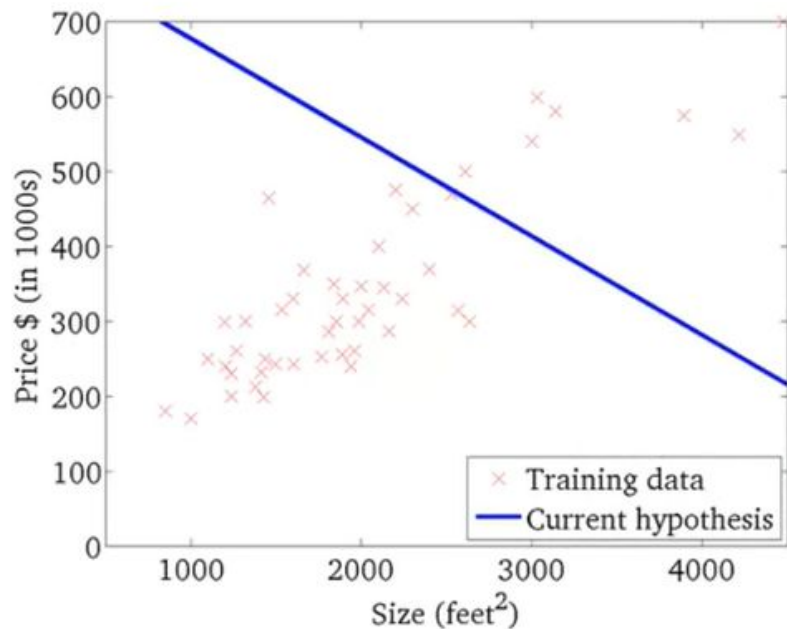


$$h_{\theta}(x) = 50 + 0.06x$$



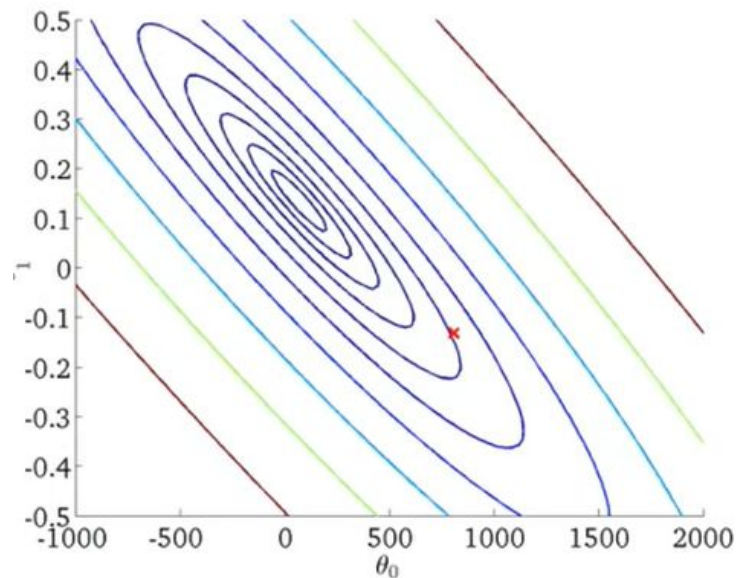
$$h_{\theta}(x)$$

(for fixed  $\theta_0, \theta_1$ , this is a function of  $x$ )



$$J(\theta_0, \theta_1)$$

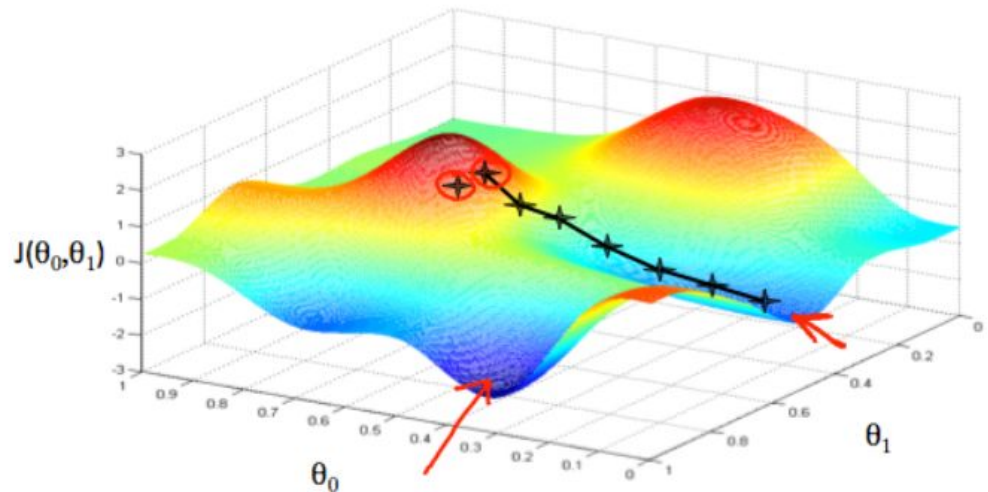
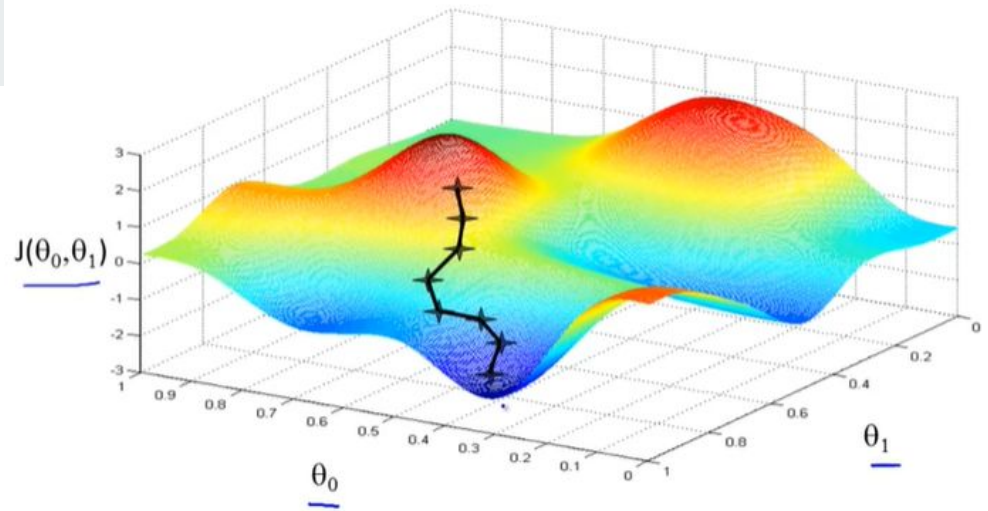
(function of the parameters  $\theta_0, \theta_1$ )



# Parameter Learning

## Gradient Descent

Nous saurons que nous avons réussi lorsque notre fonction de coût se trouve tout en bas des creux de notre graphique, c'est-à-dire lorsque sa valeur est minimale. Les flèches rouges indiquent les points minimums du graphique.





# Gradient Descent

Pour ce faire, nous prenons la dérivée (la ligne tangentielle à une fonction) de notre fonction de coût. La pente de la tangente est la dérivée à ce point et elle nous donne une direction vers laquelle aller. Nous descendons la fonction de coût dans la direction où la pente est la plus forte. La taille de chaque étape est déterminée par le paramètre  $\alpha$ , qui est appelé **the learning rate**.

Par exemple, la distance entre chaque "étoile" dans le graphique ci-dessus représente un pas déterminé par notre paramètre  $\alpha$ . Un  $\alpha$  plus petit entraîne un pas plus petit et un  $\alpha$  plus grand un pas plus grand. La direction dans laquelle le pas est pris est déterminée par la dérivée partielle de  $J$

Selon l'endroit où l'on commence sur le graphique, on peut se retrouver à différents points.



# The gradient descent algorithm

répétez jusqu'à la convergence :

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

où  $j=0,1$  représente le numéro d'indice du feature.

A chaque itération  $j$ , on doit simultanément mettre à jour les paramètres theta. La mise à jour d'un paramètre spécifique avant le calcul d'un autre paramètre sur la  $j$  itération conduirait à une implémentation erronée.

$$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := \text{temp0}$$

$$\theta_1 := \text{temp1}$$





## The gradient descent

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

$$j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) =$$

$$j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) =$$



# Multivariate linear regression

La régression linéaire avec des variables multiples est également connue sous le nom de "régression linéaire multivariée".

Nous introduisons maintenant la notation pour les équations où nous pouvons avoir un nombre quelconque de variables d'entrée.

$x_j^{(i)}$  = value of feature  $j$  in the  $i^{th}$  training example

$x^{(i)}$  = the input (features) of the  $i^{th}$  training example

$m$  = the number of training examples

$n$  = the number of features



# Multivariate linear regression

La forme multivariable de la fonction prenant en compte ces multiples caractéristiques est la suivante :

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

En utilisant la définition de la multiplication matricielle, notre fonction d'hypothèse multivariable peut être représentée de manière concise comme :

$$h_{\theta}(x) = [\theta_0 \quad \theta_1 \quad \dots \quad \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

**Remarque :** Notons que pour des raisons de commodité dans ce cours, nous supposons  $x_0=1$  pour  $(i \in 1, \dots, n)$ . Cela nous permet d'effectuer des opérations matricielles avec  $\theta$  et  $x$ .



## Gradient Descent for Multiple Variables

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

...

}

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \quad \text{for } j := 0 \dots n$$

}


---

# Classification




## Définition

La classification est une tâche très importante dans le data mining, et qui consomme beaucoup de recherches pour son optimisation. La classification supervisée est l'une des techniques les plus utilisées dans l'analyse des bases de données. Elle permet d'apprendre des modèles de décision qui permettent de prédire le comportement des exemples futurs.



La classification est un processus à deux étapes : une étape d'apprentissage (entraînement) et une étape de classification (utilisation).

Dans l'étape d'apprentissage, un classifieur (une fonction, un ensemble de règles, ...) est construit en analysant (ou en apprenant de) une base de données d'exemples d'entraînement avec leurs classes respectives. Un exemple  $X = (x_1, x_2, \dots, x_m)$  est représenté par un vecteur d'attributs de dimension  $m$ . Chaque exemple est supposé appartenir à une classe prédéfinie représentée dans un attribut particulier de la base de donnée appelé attribut de classe. Puisque la classe de chaque exemple est donnée, cette étape est aussi connue par l'apprentissage supervisé.



Avant de passer à l'utilisation, le modèle doit être testé pour s'assurer de sa capacité de généralisation sur les données non utilisées dans la phase d'entraînement.

Le modèle obtenu peut être testé sur les données d'entraînement elles-mêmes, la précision (le taux de reconnaissance) est généralement élevée mais ne garantit pas automatiquement une bonne précision sur les nouvelles données.

En effet, les données d'entraînement peuvent contenir des données bruitées ou erronées (outliers) qui ne représentent pas le cas général et qui tire le modèle vers leurs caractéristiques. Ce cas est appelée le sur-apprentissage ou en anglais "**over fitting**" et qui peut être évité en testant le modèle sur une base de données différente appelée base de test.

La base de test est un ensemble d'exemples ayant les mêmes caractéristiques que ceux de la base d'entraînement et qui sont écartés au départ de l'entraînement pour effectuer les tests.





On dispose d'un ensemble  $X$  de  $N$  données étiquetées représentant un sous ensemble de l'ensemble  $D$  de toutes les données possibles.

Chaque donnée  $x_i$  est caractérisée par  $P$  attributs et par sa classe  $y_i \in Y$ .

Dans un problème de classification, la classe prend sa valeur parmi un ensemble fini. Le problème consiste alors, en s'appuyant sur l'ensemble d'exemples  $X = \{(x_i, y_i) / i \in \{1, \dots, N\}\}$ , à prédire la classe de toute nouvelle donnée  $x \in D$ .

On parle de classification binaire quand le nombre de classes  $|Y|$  est 2 ; il peut naturellement être quelconque.

Un exemple est donc une donnée dont on dispose de sa classe. On utilise donc un ensemble d'exemples classés pour prédire les classes des nouvelles données ; c'est une tâche d'"apprentissage à partir d'exemples", ou "apprentissage supervisé".



# Binary classification problem

Email: Spam/Not Spam ?

Online Transactions: Fraudulent (yes/no)?

Tumor: Malignant/benign ?

$y \in \{0,1\}$

0: Negative Class

1: Positive Class



# Linear regression ?

Threshold classifier output  $h(x)$  at 0.5:

If  $h_{\theta}(x) \geq 0.5$ , predict "y = 1"

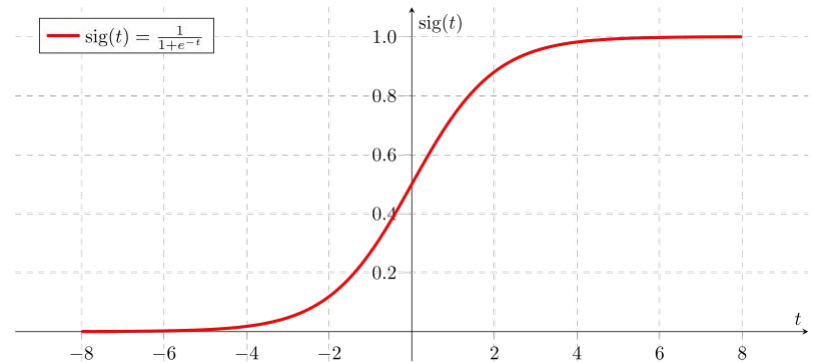
If  $h_{\theta}(x) < 0.5$ , predict "y = 0"

# Logistic regression

On veut  $0 \leq h(x) \leq 1$

$$h(x) = g(\theta^T x)$$

$$= 1/(1+\exp(-\theta^T x)) \quad \Rightarrow \text{Sigmoid function}$$






# Logistic Regression

Example if

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

$h(x) = 0.7 \Rightarrow$  Tell patient that 70% chance of tumor being malignant


Probability that  $y = 1$  given  $x$ , parameterized by  $\theta$   $P(y=1|x;\theta)$


$$h_{\theta}(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Suppose predict “ $y = 1$ ” if  $h_{\theta}(x) \geq 0.5$

predict “ $y = 0$ ” if  $h_{\theta}(x) < 0.5$

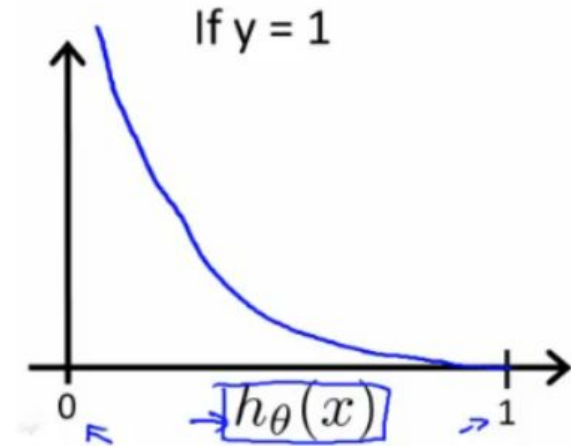


Training Set =  $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$

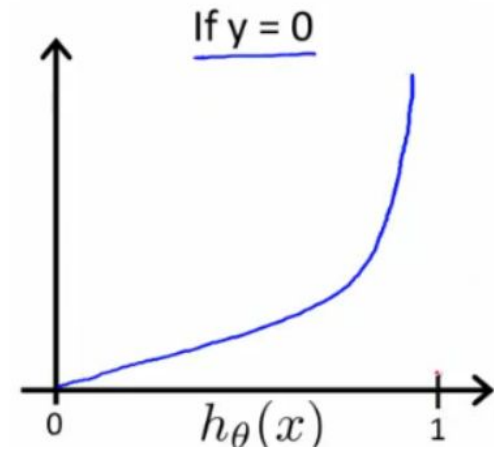
$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad \underline{x_0} = 1, y \in \{0, 1\}$$
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

How to choose Theta?

When  $y = 1$ , we get the following plot for  $J(\theta)$  vs  $h_\theta(x)$ :



Similarly, when  $y = 0$ , we get the following plot for  $J(\theta)$  vs  $h_\theta(x)$ :



## Logistic Regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \quad \text{if } y = 1$$

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \quad \text{if } y = 0$$

$$\text{Cost}(h_\theta(x), y) = 0 \text{ if } h_\theta(x) = y$$

$$\text{Cost}(h_\theta(x), y) \rightarrow \infty \text{ if } y = 0 \text{ and } h_\theta(x) \rightarrow 1$$

$$\text{Cost}(h_\theta(x), y) \rightarrow \infty \text{ if } y = 1 \text{ and } h_\theta(x) \rightarrow 0$$





## Simplified Cost Function

Nous pouvons comprimer les deux cas conditionnels de notre fonction de coût en un seul cas :

$$\text{Cost}(h_{\theta}(x), y) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x))$$

Remarquez que lorsque  $y$  est égal à 1, alors le deuxième terme  $(1-y)\log(1-h_{\theta}(x))$  sera égal à zéro et n'affectera pas le résultat. Si  $y$  est égal à 0, alors le premier terme  $-y\log(h_{\theta}(x))$  sera nul et n'affectera pas le résultat.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$



## Gradient Descent

Rappelez-vous que la forme générale de la descente du gradient est :

$$\begin{aligned} & \textit{Repeat} \{ \\ & \quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \\ & \} \end{aligned}$$

On peut calculer la partie dérivée en utilisant le calcul pour obtenir:

$$\begin{aligned} & \textit{Repeat} \{ \\ & \quad \theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \\ & \} \end{aligned}$$

Remarquez que cet algorithme est identique à celui que nous avons utilisé dans la régression linéaire. Nous devons toujours mettre à jour simultanément toutes les valeurs de theta



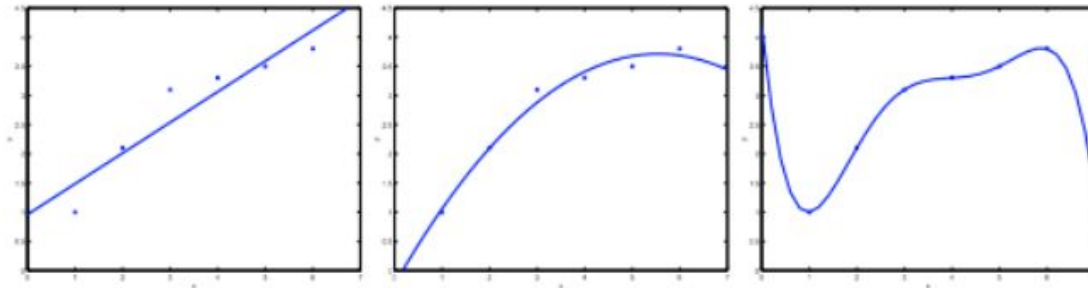
## Le problème du Overfitting

Considérons le problème de la prédiction de  $y$  à partir de  $x \in \mathbb{R}$ . La figure la plus à gauche ci-dessous montre le résultat de l'ajustement de  $y = \theta_0 + \theta_1 x$  à un ensemble de données. On voit que les données ne sont pas vraiment en ligne droite, et donc que l'ajustement n'est pas très bon.

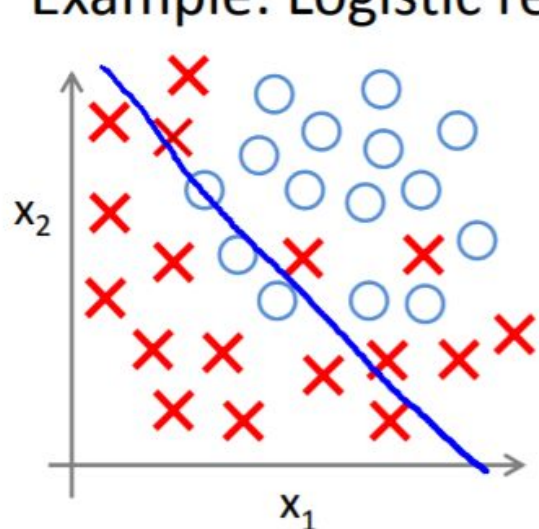
# Le problème du Overfitting

Au lieu de cela, si nous avons ajouté une caractéristique supplémentaire  $x^2$ , et ajusté  $y = \theta_0 + \theta_1 x + \theta_2 x^2$ , nous aurions obtenu un ajustement légèrement meilleur aux données (voir figure du milieu).

Naïvement, il pourrait sembler que plus on ajoute de caractéristiques, mieux c'est. Cependant, il y a aussi un danger à ajouter trop de caractéristiques : La figure la plus à droite est le résultat de l'ajustement d'un polynôme d'ordre 5  $y = \sum_{j=0}^5 \theta_j x^j$



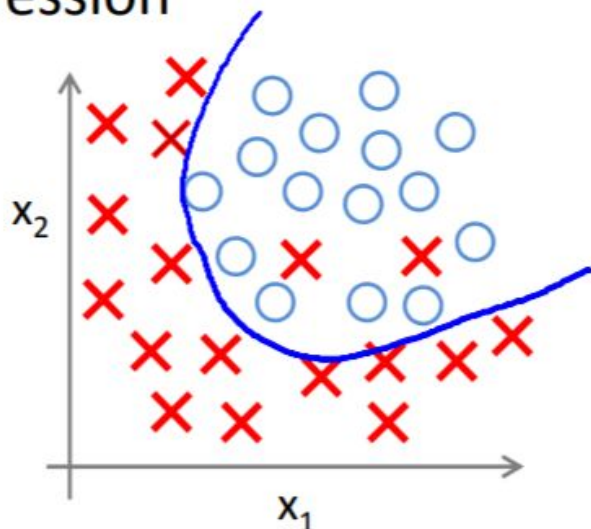
## Example: Logistic regression



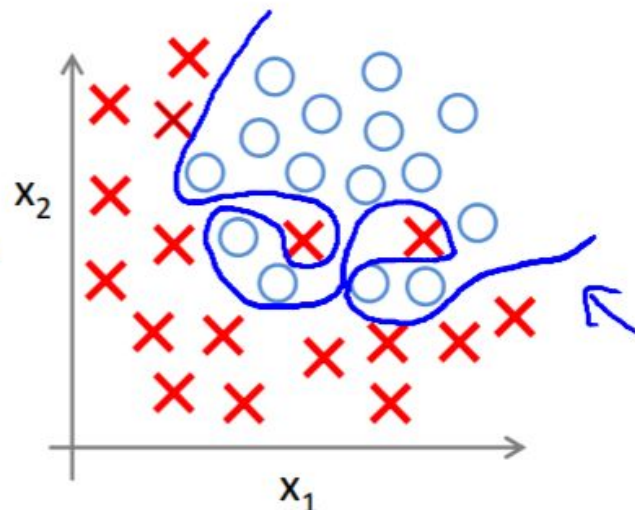
$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

( $g$  = sigmoid function)

"Underfit"



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

"Overfit"



# Le problème du Overfitting

Nous constatons que, même si la courbe ajustée passe parfaitement par les données, nous ne nous attendons pas à ce qu'elle soit un très bon prédicteur, par exemple, des prix des logements ( $y$ ) pour différentes zones d'habitation ( $x$ ).

Sans définir formellement la signification de ces termes, nous dirons que la figure de gauche montre un exemple de **underfitting** - dans lequel les données montrent clairement une structure non prise en compte par le modèle - et que la figure de droite est un exemple de **overfitting**.



# Le problème du Overfitting

On parle de **underfitting**, ou de biais élevé, lorsque la forme de notre fonction d'hypothèse  $h$  correspond mal à la tendance des données.

Il est généralement causé par une fonction trop simple ou utilisant trop peu de caractéristiques.

À l'autre extrême, **overfitting**, ou variance élevée, est causé par une fonction d'hypothèse qui s'adapte aux données disponibles mais qui ne se généralise pas bien pour prédire de nouvelles données.

Il est généralement causé par une fonction compliquée qui crée un grand nombre de courbes et d'angles inutiles sans rapport avec les données.



# Le problème du Overfitting

Cette terminologie s'applique à la fois à la régression linéaire et à la régression logistique. Il existe deux options principales pour résoudre le problème de l'overfitting :

## 1) Réduire le nombre de fonctionnalités :

- Sélectionnez manuellement les caractéristiques à conserver.
- Utilisez un algorithme de sélection de features.

## 2) Régularisation

- Gardez toutes les caractéristiques, mais réduisez la magnitude des paramètres  $\theta_j$
- La régularisation fonctionne bien lorsque nous avons beaucoup de caractéristiques légèrement utiles.





# Régularisation

En cas du overfitting de notre fonction d'hypothèse, nous pouvons réduire le poids de certains des termes de notre fonction en augmentant leur coût.

Supposons que nous voulions rendre la fonction suivante plus quadratique :

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Nous voudrions éliminer l'influence de  $\theta_3 x^3$  et  $\theta_4 x^4$ . Sans réellement se débarrasser de ces caractéristiques ou changer la forme de notre hypothèse, nous pouvons à la place modifier notre fonction de coût :

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + 1000 \cdot \theta_3^2 + 1000 \cdot \theta_4^2$$



# Régularisation

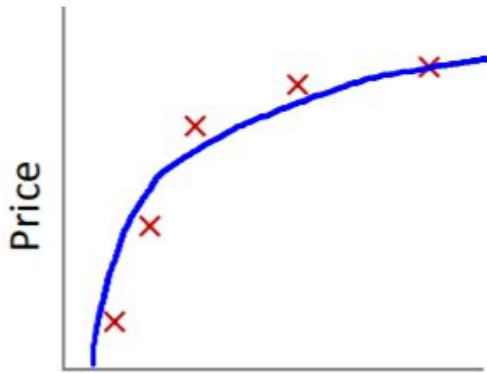
Nous avons ajouté deux termes supplémentaires à la fin pour gonfler le coût de  $\theta_3$  et  $\theta_4$ .

Maintenant, pour que la fonction de coût se rapproche de zéro, nous devons réduire les valeurs de  $\theta_3$  et  $\theta_4$  à presque zéro.

Cela va à son tour réduire considérablement les valeurs de  $\theta_3 \cdot x^3$  et  $\theta_4 \cdot x^4$  dans notre fonction d'hypothèse.

En conséquence, nous voyons que la nouvelle hypothèse (représentée par la courbe rose) ressemble à une fonction quadratique mais s'adapte mieux aux données grâce aux petits termes supplémentaires  $\theta_3 \cdot x^3$  et  $\theta_4 \cdot x^4$ .

# Régularisation



Size of house

$$\theta_0 + \theta_1 x + \theta_2 x^2$$



Size of house

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$



# Régularisation

Nous pourrions également régulariser tous nos paramètres  $\theta$  en une seule sommation comme :

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2$$

Le  $\lambda$ , ou lambda, est le paramètre de régularisation. Il détermine de combien les coûts de nos paramètres  $\theta$  sont gonflés.

En utilisant la fonction de coût ci-dessus avec la sommation supplémentaire, nous pouvons lisser la sortie de notre fonction d'hypothèse pour réduire l'overfitting.

Si lambda est choisi trop grand, il peut trop lisser la fonction et provoquer underfitting.



## Regularized Linear Regression

Nous allons modifier notre fonction de descente de gradient pour séparer  $\theta_0$  du reste des paramètres car nous ne voulons pas pénaliser  $\theta_0$ .

$$\begin{aligned} &\text{Repeat } \{ \\ &\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)} \\ &\quad \theta_j := \theta_j - \alpha \left[ \left( \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)} \right) + \frac{\lambda}{m} \theta_j \right] \quad j \in \{1, 2 \dots n\} \\ &\} \end{aligned}$$

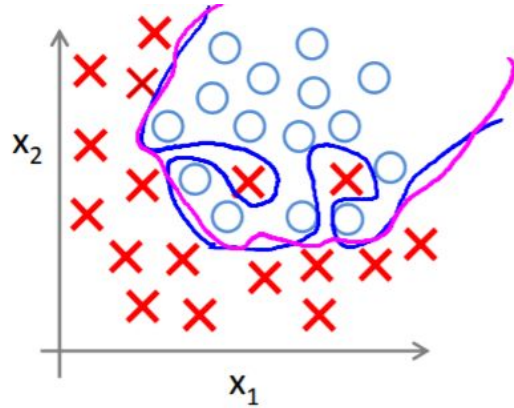


## Regularized Linear Regression

$$\theta_j := \theta_j(1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})x_j^{(i)}$$

Le premier terme de l'équation ci-dessus,  $1 - \alpha \lambda/m$ , sera toujours inférieur à 1. Intuitivement, vous pouvez le voir comme réduisant la valeur de  $\theta_j$  d'une certaine quantité à chaque mise à jour. Remarquez que le deuxième terme est maintenant exactement le même qu'avant.

## Regularized Logistic Regression



$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

---

# Clustering





# Introduction

Le clustering regroupe un ensemble de techniques qui visent à regrouper les enregistrements d'une base de données en des groupes selon leur rapprochement les uns des autres en ne se basant sur aucune information antérieure, c'est un **apprentissage non supervisé**.

Un système d'analyse en clusters prend en entrée un ensemble de données et une mesure de similarité entre ces données, et produit en sortie un ensemble de partitions décrivant la structure générale de l'ensemble de données.



# Introduction

Plus formellement, un système de clustering prend un tuple  $(D, s)$  où  $D$  représente l'ensemble de données et  $s$  la mesure de similarité, et retourne une partition  $P = (G_1, G_2, \dots, G_m)$  tel que les  $G_i (i = 1..m)$  sont des sous ensembles de  $D$  qui vérifient :

$$\begin{cases} G_1 \cup G_2 \cup \dots \cup G_m = D \\ G_i \cap G_j = \Phi, \quad i \neq j \end{cases}$$

Chaque  $G_i$  est appelé un cluster qui représente une ou plusieurs caractéristiques de l'ensemble  $D$ .



# Introduction

Un problème à traiter:

-Le nombre de clusters à chercher: Dans certains cas c'est un expert du domaine d'application qui fournit le nombre de clusters qui forment l'ensemble de données. Mais dans la plupart des cas, même l'expert ne connaît pas le nombre de clusters et cherche à le savoir et le comprendre.

Dans la majorité des cas, on définit une mesure de stabilité du processus d'analyse à base de laquelle on peut atteindre le meilleur nombre de clusters qui décrit mieux les données.



## Mesures de similarités

Une bonne méthode de clustering est une méthode qui maximise la ressemblance entre les données à l'intérieur de chaque cluster, et minimise la ressemblance entre les données des clusters différents.

C'est pourquoi les résultats d'une technique de clustering dépendent fortement de la mesure de similarité choisie par son concepteur, qui doit la choisir avec prudence.

La mesure de similarité repose sur le calcul de la distance entre deux données, sachant que chaque donnée est composée d'un ensemble d'attributs numériques et/ou catégoriels.

Plus la distance est importante, moins similaires sont les données et vice versa.



# Attributs numériques

Pour mesurer la distance entre les données à attributs numériques, plusieurs formules existent :

- La distance Euclidienne :

$$D_n(x_i, x_j) = \sqrt{\sum_{k=1}^{n_n} (x_{ik} - x_{jk})^2}$$

- La distance City blocs :

$$D_n(x_i, x_j) = \sum_{k=1}^{n_n} |x_{ik} - x_{jk}|$$

- La distance de Minkowski :

$$D_{np}(x_i, x_j) = \left( \sum_{k=1}^{n_n} (x_{ik} - x_{jk})^p \right)^{\frac{1}{p}}$$



## Attributs catégoriels

Le problème qui se pose lors du calcul de la distance entre les attributs catégoriels, c'est qu'on ne dispose pas d'une mesure de différence.

La seule mesure qui existe, dans l'absence de toute information sur la signification des valeurs, c'est l'égalité ou l'inégalité. La distance utilisée est alors :

$$\begin{cases} D_c(x_i, x_j) = \frac{1}{n_c} \sum_{k=1}^{n_c} f(x_{ik}, x_{jk}) \\ f(x_{ik}, x_{jk}) = 1 \quad \text{si } x_{ik} \neq x_{jk}, \quad 0 \quad \text{sinon} \end{cases}$$



Il faut en fin la normaliser avec les attributs numériques et le nombre d'attributs catégoriels.

La distance entre deux données  $x_i$  et  $x_j$ , composées d'attributs numériques et catégoriels, est donc


$$D(x_i, x_j) = D_n(x_i, x_j) + D_c(x_i, x_j)$$



En se basant sur la distance entre deux attributs, plusieurs distances peuvent être calculées

- **Distance entre deux clusters** : permet de mesurer la distance entre deux clusters pour une éventuelle fusion en cas où ils soient trop proches. Cette distance peut être prise entre les centres des deux clusters, entre les deux données les plus éloignées (ou plus proches) des deux clusters ou la distance moyenne de leurs données.
- **Distance intracluster** : c'est la distance moyenne entre les données à l'intérieur d'un cluster, elle peut être utile pour maintenir un seuil d'éloignement maximum dans le cluster au dessus duquel on doit scinder ce cluster.
- **Distance intercluster** : c'est la distance moyenne entre les clusters, elle permet de mesurer l'éloignement moyen entre les différents clusters.
- **Distance intraclusters moyenne** : permet avec la distance interclusters de mesurer la qualité du clustering.:





La mesure de similarité peut être utilisée par un algorithme de clustering pour trouver le partitionnement optimal des données. Parmi ces algorithmes on peut citer :

- **Clustering hiérarchique**
- **Clustering partitionnel**
- **Clustering incrémental**
- **Clustering basé densité**



# Clustering hiérarchique

Dans ce type de clustering le nombre de clusters ne peut être connu à l'avance.

Le système prend en entrée l'ensemble de données et fournit en sortie une arborescence de clusters.

Il existe deux classe de ce type d'algorithmes :

- **Les algorithmes divisibles** qui commencent à partir d'un ensemble de données et le subdivisent en sous ensembles puis subdivisent chaque sous ensemble en d'autres plus petits, et ainsi de suite, pour générer en fin une séquence de clusters ordonnée du plus général au plus fin.
- **Les algorithmes agglomératifs** qui considèrent chaque enregistrement comme étant un cluster indépendant puis rassemblent les plus proches en des clusters plus importants, et ainsi de suite jusqu'à atteindre un seul cluster contenant toutes les données.



# Algorithme Agglomératif

Un algorithme agglomératif suit généralement les étapes suivantes :

1. Placer chaque enregistrement dans son propre cluster ;
2. Calculer une liste des distances interclusters et la trier dans l'ordre croissant ;
3. Pour chaque seuil de niveau de similitude préfixé  $d_k$
4. Relier tous les clusters dont la distance est inférieure à  $d_k$  par des arrêtes à un nouveau cluster ;
5. Pour
6. Si tous les enregistrements sont membres d'un graphe connecté alors fin sinon aller à 3 ;
7. Le résultat est un graphe qui peut être coupé selon le niveau de similarité désiré ;



## Exemple

Soient les données suivantes :  $X_1(0, 2)$ ,  $X_2(0, 0)$ ,  $X_3(1.5, 0)$ ,  $X_4(5, 0)$ ,  $X_5(5, 2)$  On utilise la distance euclidienne pour mesurer la distance entre les données:

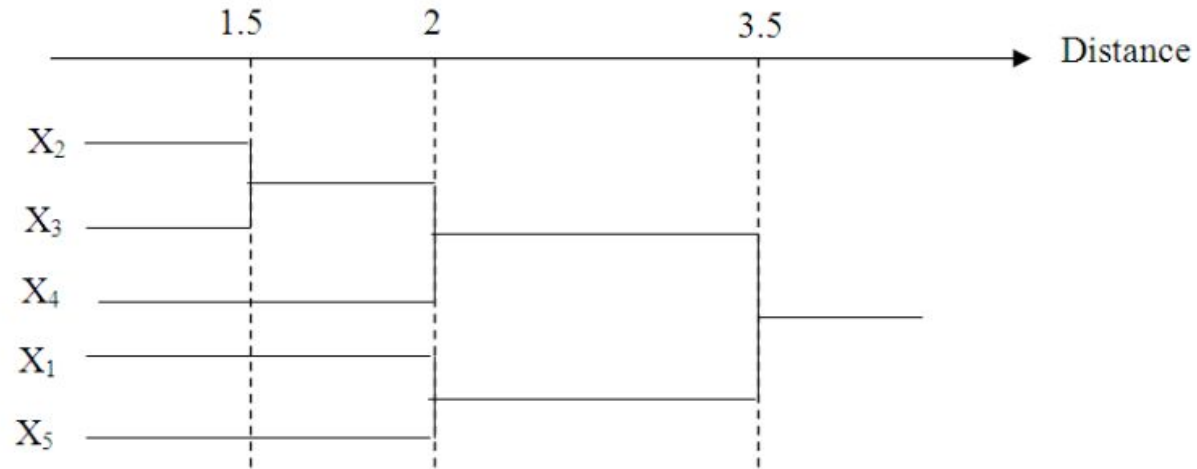
$$D(X_1, X_2) = \sqrt{(0 - 0) \times 2 + (2 - 0) \times 2} = 2, \quad D(X_1, X_3) = 2.5, \quad D(X_2, X_3) = 1.5,$$

On trie ces distances, on trouve que  $X_2$  et  $X_3$  sont les plus proches, on les rassemble dans le même cluster et on calcule son centre  $(0.75, 0)$ ,

On refait le calcul des distances en remplaçant  $X_2, X_3$  par le nouveau cluster, on trie puis on choisit la plus courte distance et ainsi de suite.

## Exemple

On obtient à la fin le dendrogramme suivant représentant le rassemblement hiérarchique des données





## Clustering partitionnel

Un algorithme partitionnel de clustering obtient à sa fin une seule partition des données au lieu de plusieurs tel que dans le cas précédent.

Pour ce faire, il essaye d'optimiser la distance moyenne entre les données de chaque cluster et son centre, on utilise généralement l'erreur quadratique qui calcule l'erreur entre les données et le centre :

$$e_k^2 = \sum_{i=1}^{N_k} (x_{ik} - \bar{x}_k)^2$$

Où  $e_k^2$  est l'erreur quadratique moyenne du cluster  $k$ ,  $N_k$  est le nombre d'enregistrements dans le cluster,  $x_{ik}$  est l'enregistrement numéro  $i$  du cluster  $k$  et  $\bar{x}_k$  est la moyenne du cluster  $k$ .



# Clustering partitionnel

L'objectif d'un algorithme partitionnel est de trouver une partition qui minimise  $E^2$  tel que

$$e_K^2 = \sum_{k=1}^K e_k^2$$

Le plus simple et le plus utilisé des algorithmes partitionnels est l'algorithme **K-means**, il démarre d'une partition arbitraire des enregistrements sur les  $k$  clusters, à chaque itération il calcule les centres de ces clusters puis il effectue une nouvelle affectation des enregistrements aux plus proches centres.

Il s'arrête dès qu'un critère d'arrêt est satisfait, généralement on s'arrête si aucun enregistrement ne change de cluster.



# K-means

L'algorithme de K-means est le suivant:

1. Sélectionner une partition initiale contenant des enregistrements choisis arbitrairement, puis calculer les centres des clusters ;
2. Calculer une liste des distances interclusters et la trier dans l'ordre croissant ;
3. Générer une nouvelle partition en affectant chaque enregistrement au cluster du centre le plus proche ;
4. Calculer les nouveaux centres des clusters ;
5. Répéter 2 et 3 jusqu'à ce que les enregistrements se stabilisent dans leurs clusters ;





## Exemple

Prenons le même ensemble de données précédent :  $X_1(0, 2)$ ,  $X_2(0, 0)$ ,  $X_3(1.5, 0)$ ,  $X_4(5, 0)$ ,  $X_5(5, 2)$

On commence par choisir une affectation arbitraire des données:  $C_1 = X_1, X_2, X_4$  et  $C_2 = X_3, X_5$ .

On calcule les deux centres :

$$M_1 = ((0 + 0 + 5)/3, (2 + 0 + 0)/3) = (1.66, 0.66)$$

$$M_2 = ((1.5 + 5)/2, (0 + 2)/2) = (3.25, 1)$$

On calcule la distance entre chaque donnée  $X_i$  et les centres  $M_1$  et  $M_2$ , puis on affecte chaque données au cluster le plus proche.

La nouvelle affectation est  $C_1 = \{X_1, X_2, X_3\}$ ,  $C_2 = \{X_4, X_5\}$ .

On recalcule les nouveaux centres, puis on réaffecte jusqu'à ce qu'aucune donnée ne change de cluster.



# Clustering incrémental

Les deux algorithmes présentés ci-dessus nécessitent la présence de tout l'ensemble de données analysées en mémoire, ce qui n'est pas pratique avec de larges bases de données avec des millions d'enregistrements.

Pour palier ce problème le clustering incrémental traite une partie des données (selon la mémoire disponible) puis ajoute itérativement des données en modifiant chaque fois si nécessaire le partitionnement obtenu.



# Algorithme du Clustering incrémental

L'algorithme suivant résume les étapes suivies :

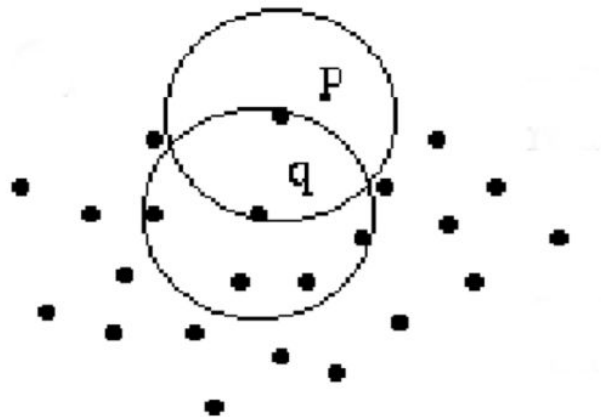
1. Affecter le premier enregistrement au premier cluster ;
2. Pour un nouvel enregistrement : soit l'affecter à un cluster existant, soit l'affecter à un nouveau cluster. Cette affectation est effectuée selon un certain critère. Par exemple la distance du nouvel enregistrement des centres des clusters existants. Dans ce cas à chaque affectation les centres des clusters sont recalculés ;
3. Répéter l'étape 2 jusqu'à ce que tous les enregistrements soient clusterés ;

# Clustering basé densité

Ce type de clustering se base sur l'utilisation de la densité à la place de la distance. On dit qu'un point est dense si le nombre de ses voisins dépasse un certain seuil.

Un point est voisin d'un autre point s'il est à une distance inférieure à une valeur fixée.

Dans la figure suivante q est dense mais pas p :





# L'algorithme DBSCAN

L'algorithme DBSCAN (Density-Based Spatial Clustering of Applications with Noise) est un exemple des algorithmes à base de densité.

Il utilise deux paramètres : la distance et le nombre minimum de points MinPts devant se trouver dans un rayon pour que ces points soient considérés comme un cluster.

Les paramètres d'entrées sont donc une estimation de la densité de points des clusters.

L'idée de base de l'algorithme est ensuite, pour un point donné, de récupérer son -voisinage et de vérifier qu'il contient bien MinPts points ou plus.

Ce point est alors considéré comme faisant partie d'un cluster. On parcourt ensuite l'-voisinage de proche en proche afin de trouver l'ensemble des points du cluster.