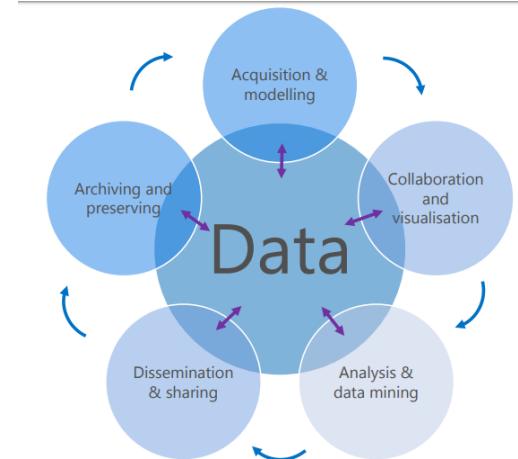
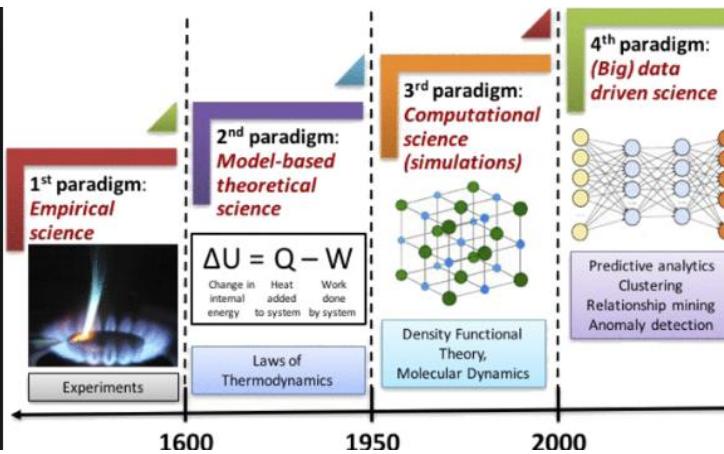
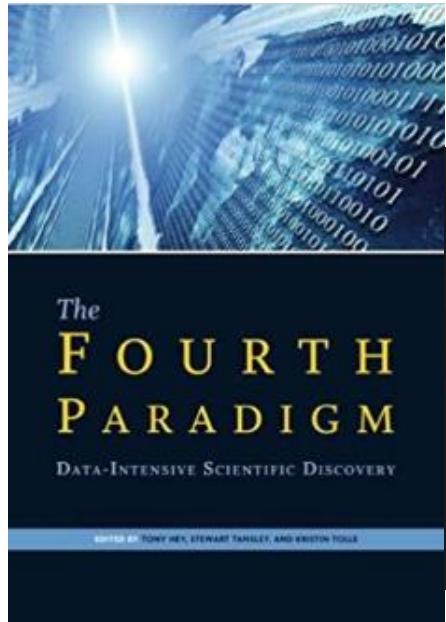


Pr. BEN LAHMAR EL Habib

What is machine learning?

- Algorithms types
- Learning techniques: an overview
- Data Preparation (Data pre-processing)
- Algorithms and models

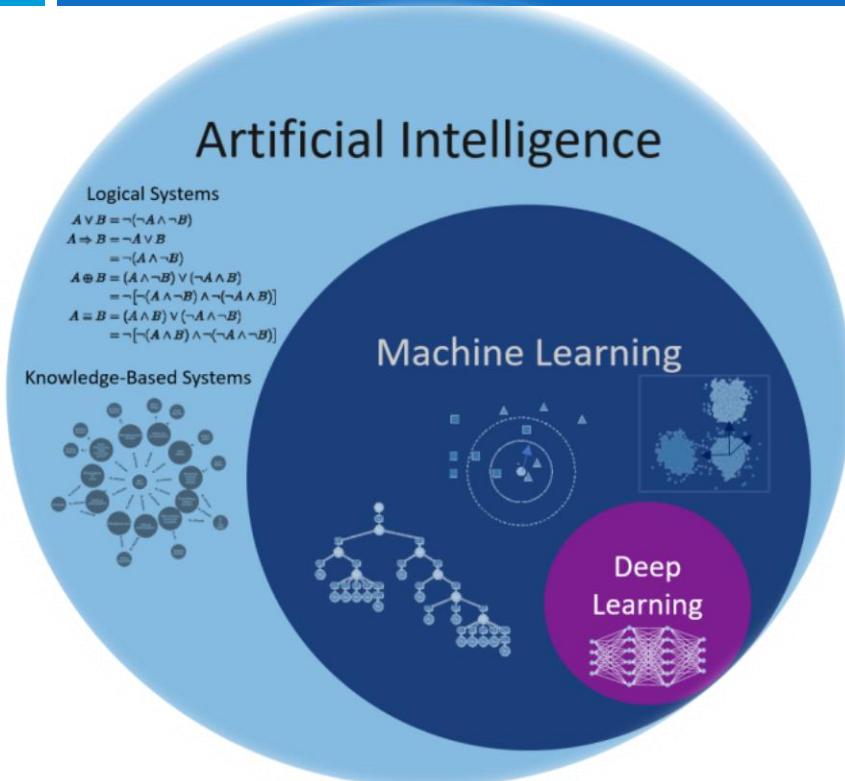
Fourth Paradigm



De quoi parlons-nous quand nous parlons d'apprentissage?

- Apprentissage de modèles généraux à partir des exemples particuliers de données ;
- Les données sont peu coûteuses et nombreuses (entrepôts de données, data marts); la connaissance est chère et rare.
- Exemple : les transactions de client à comportement de consommateur:
- Les personnes qui ont acheté «Da Vinci Code» ont également acheté «The Five People You Meet in Heaven» (www.amazon.com)
- Construisez un modèle qui constitue **une approximation valable et utile** des données.

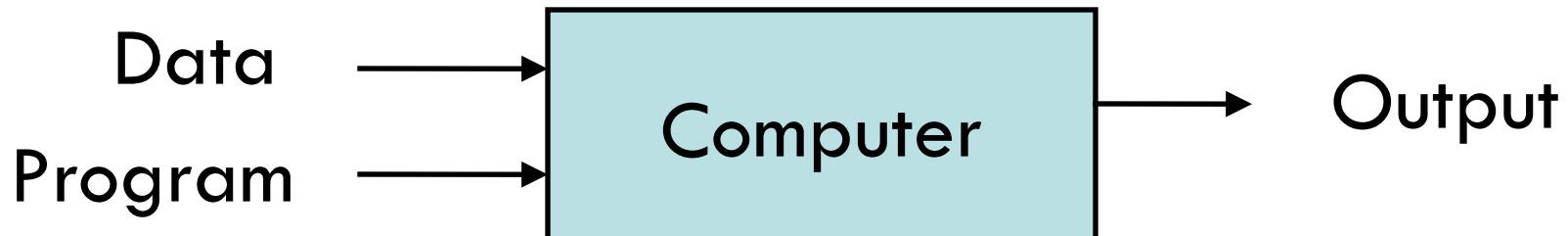
What is machine learning?



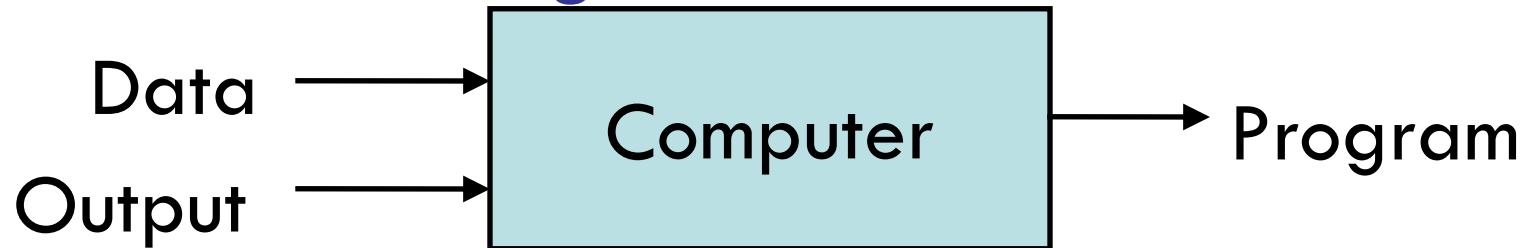
- Une branche de **l'intelligence artificielle**, préoccupée par la conception et le développement d'algorithmes permettant aux ordinateurs de faire évoluer les comportements en fonction de données empiriques.
- Comme l'intelligence requiert du savoir, il est nécessaire que les ordinateurs acquièrent ce savoir

Paradigme

Traditional Programming



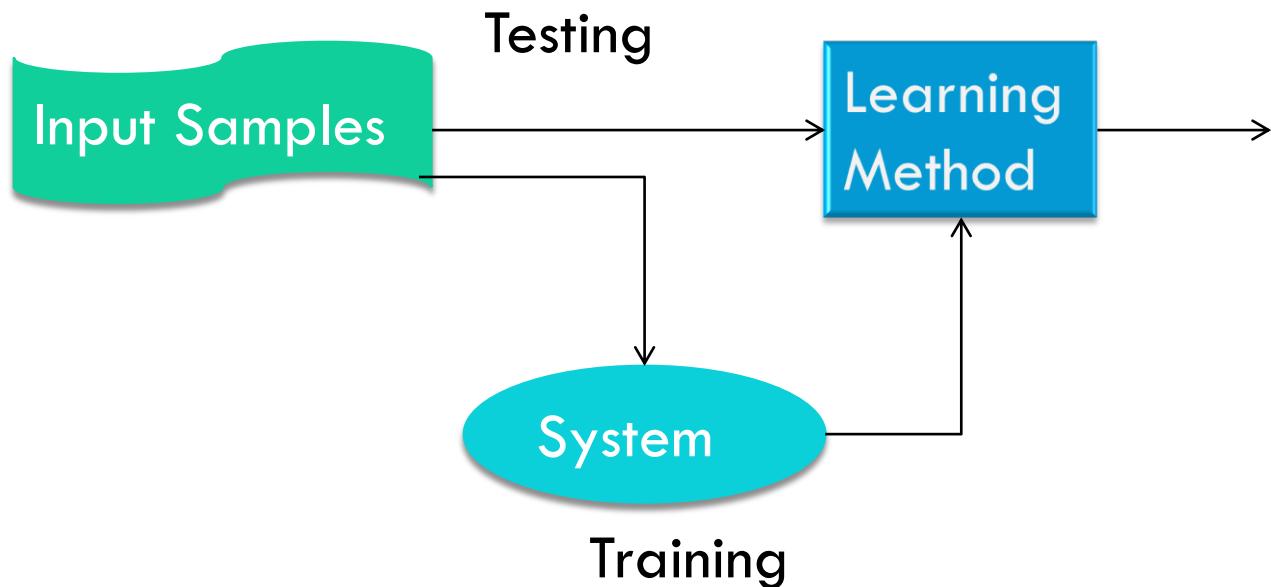
Machine Learning



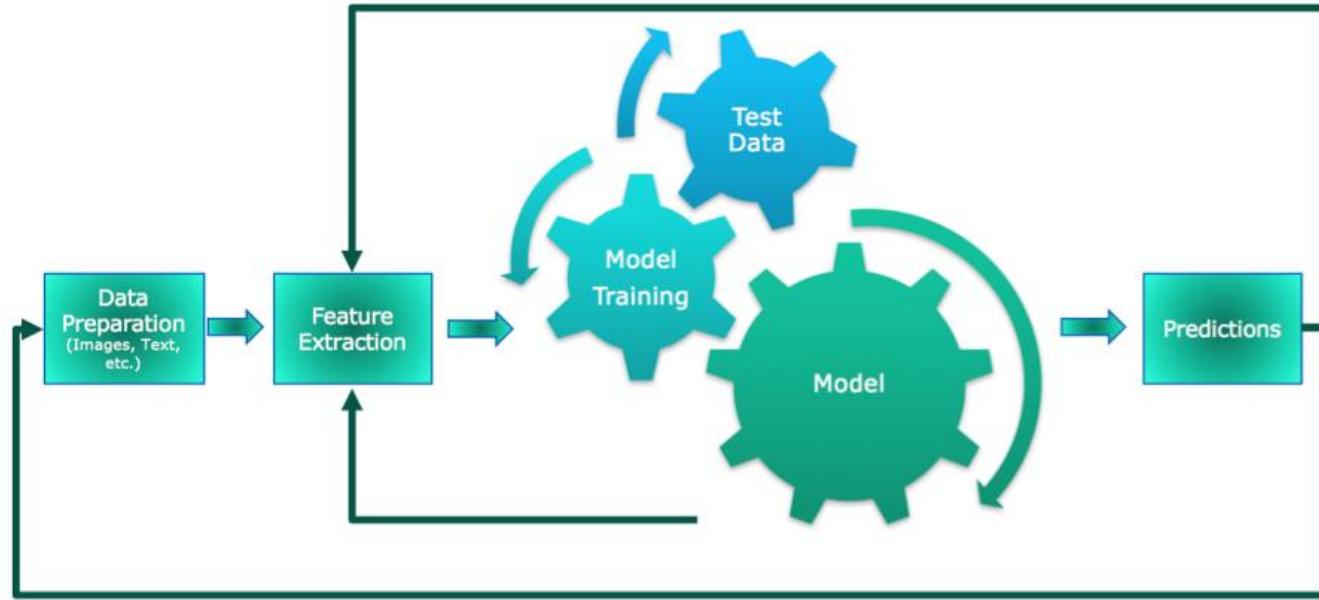
What is machine learning?

- Apprentissage automatique
 - Etude d'algorithmes qui
 - améliorer leurs performances
 - à une tâche
 - avec expérience
- Optimiser un critère de performance en utilisant des exemples de données ou une expérience passée.
- Rôle de la statistique: inférence à partir d'un échantillon
- Rôle de l'informatique: algorithmes efficaces pour
 - Résoudre le problème d'optimisation
 - Représenter et évaluer le modèle d'inférence

Learning system model

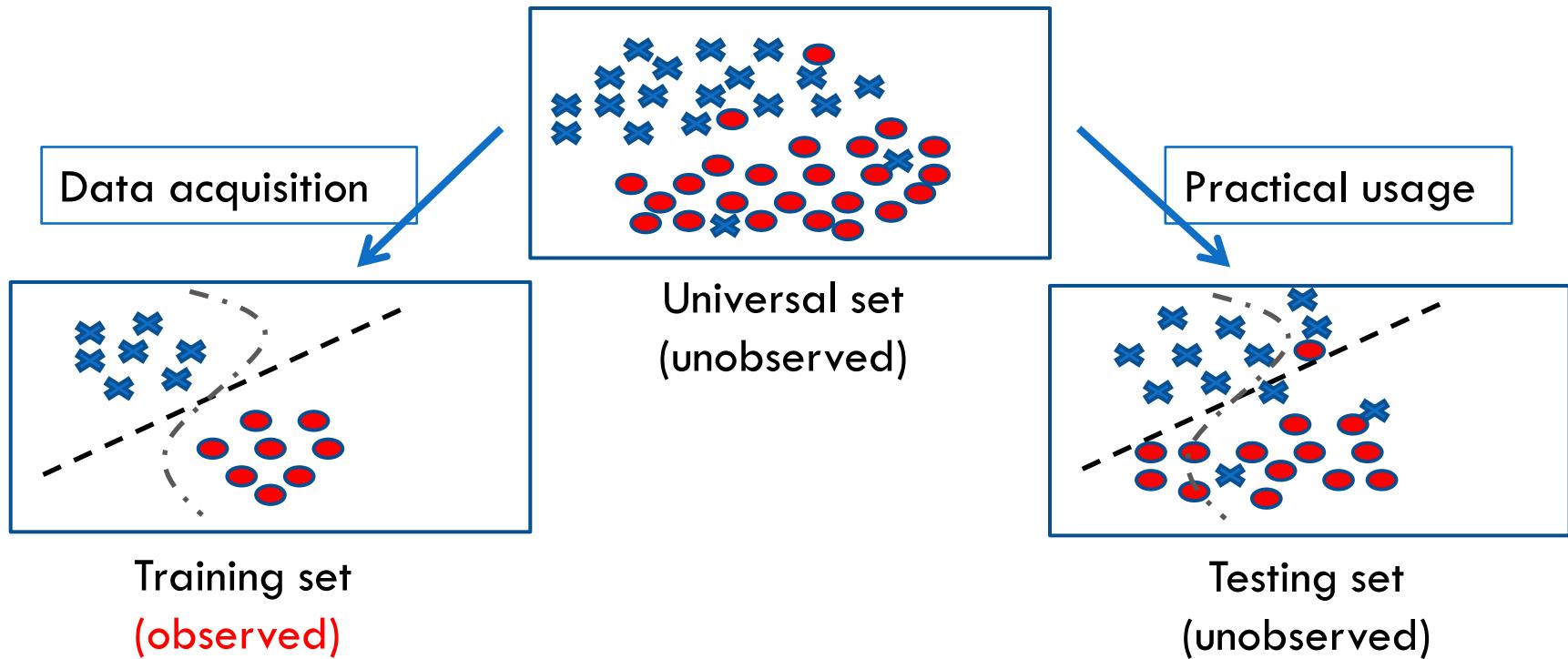


A standard Machine learning pipeline

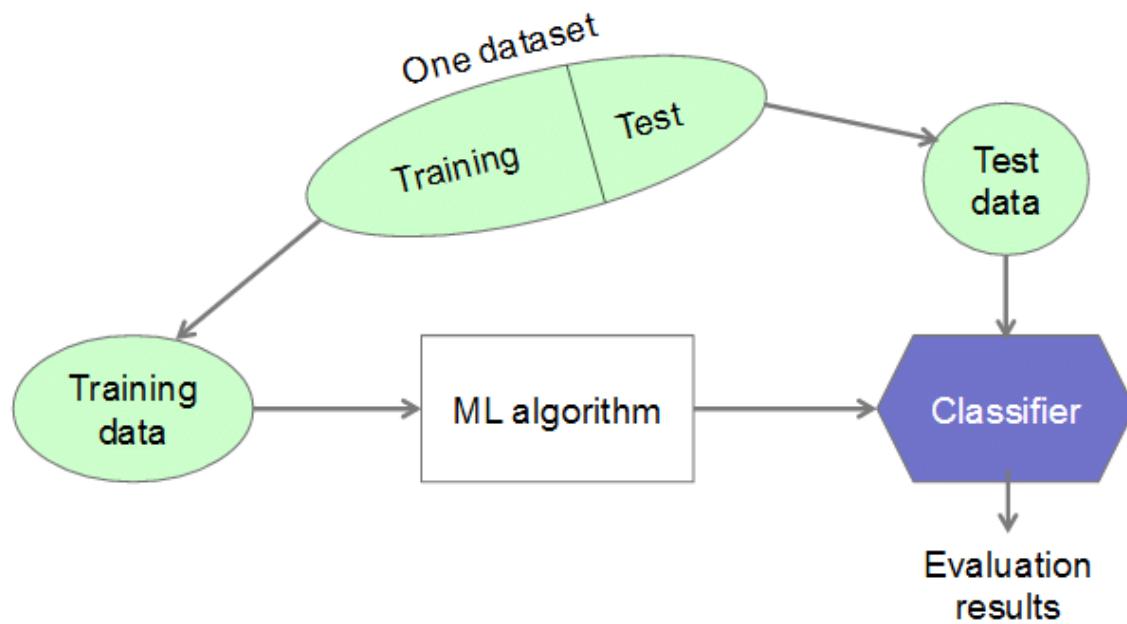


- Feature extraction is critical for machine learning pipelines (Courtesy: Western Digital)

Training and testing



Training and testing



Performance

- Plusieurs facteurs affectent la performance:
- **Types d'entraînement fournis**
- La forme et l'étendue de toute **connaissance de base initiale**
- **Le type du feedback fourni**
- Les **algorithmes d'apprentissage utilisés**

- Deux facteurs importants:
 - La modélisation
 - Optimisation

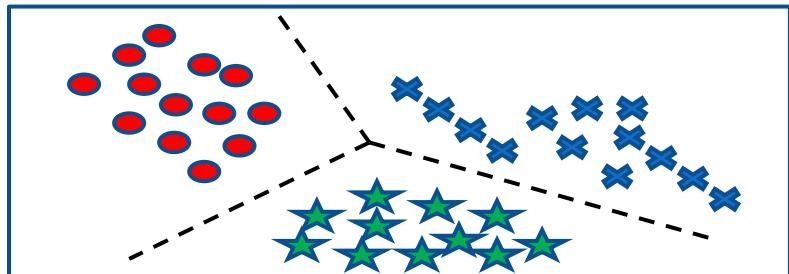
Algorithmes

- Le succès du système d'apprentissage automatique dépend également des algorithmes.
- Les algorithmes contrôlent la recherche pour trouver et construire les structures de connaissances.
- Les algorithmes d'apprentissage doivent extraire des informations utiles des exemples d'apprentissage.

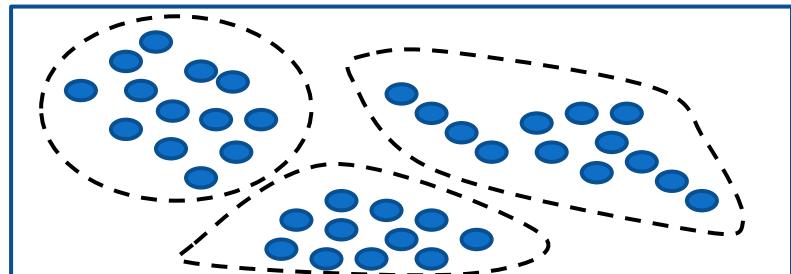
Algorithms

- **Supervised learning** ($\{x_n \in R^d, y_n \in R\}_{n=1}^N$)
 - Prediction
 - Classification (discrete labels), Regression (real values)
 - Training data includes desired outputs
- **Unsupervised learning** ($\{x_n \in R^d\}_{n=1}^N$)
 - Clustering
 - Probability distribution estimation
 - Finding association (in features)
 - Dimension reduction
 - Training data does not include desired outputs
- **Semi-supervised learning**
 - Training data includes a few desired outputs
- **Reinforcement learning**
 - Decision making (robot, chess machine)
 - Rewards from sequence of actions

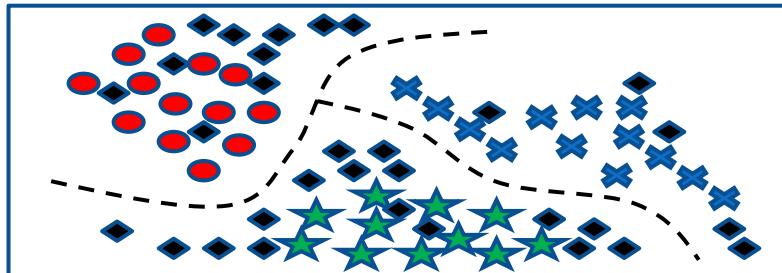
Algorithms



Supervised learning



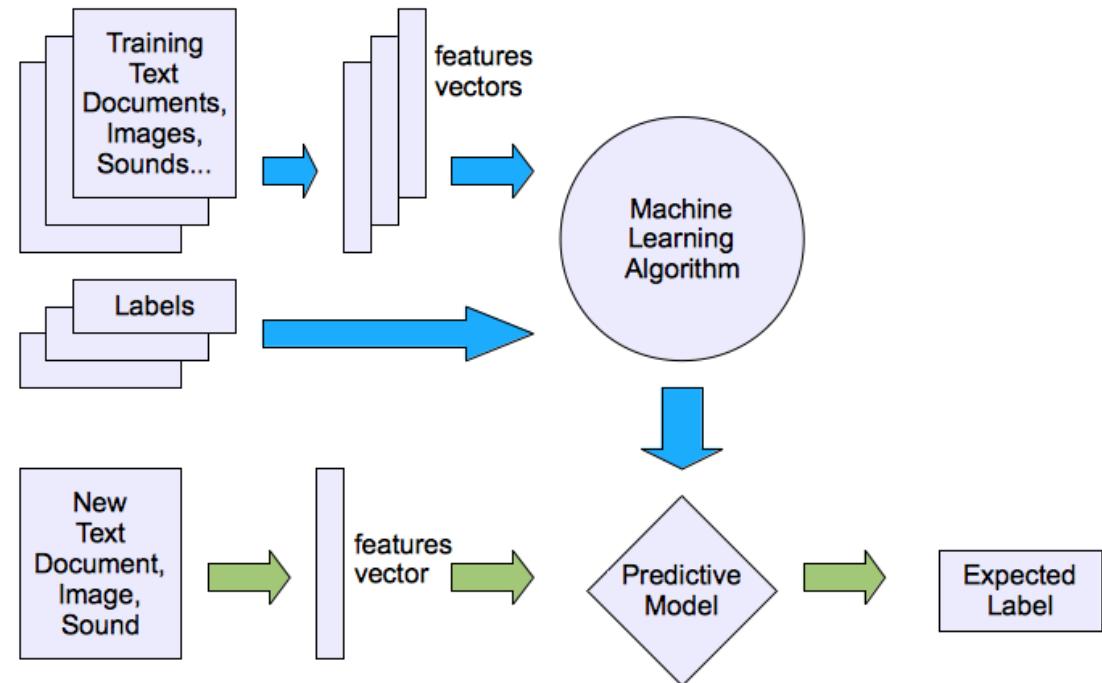
Unsupervised learning



Semi-supervised learning

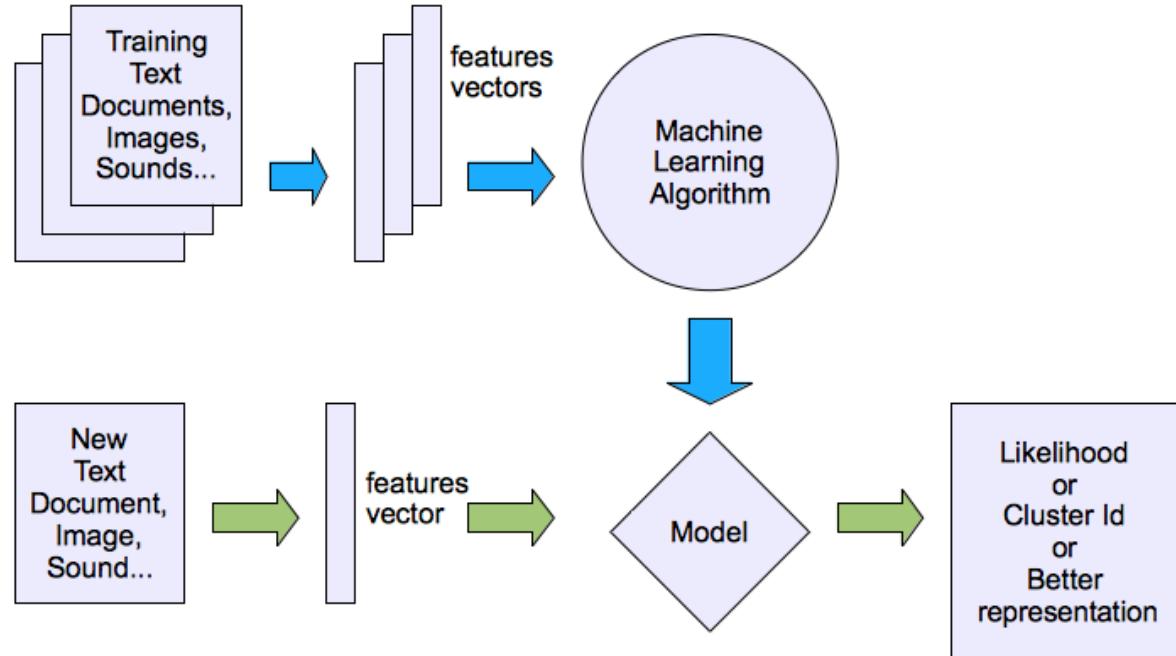
Machine learning structure

Supervised learning



Machine learning structure

Unsupervised learning



Que cherchons nous?

- Supervisé: Faible E-out ou maximisation des termes probabilistes

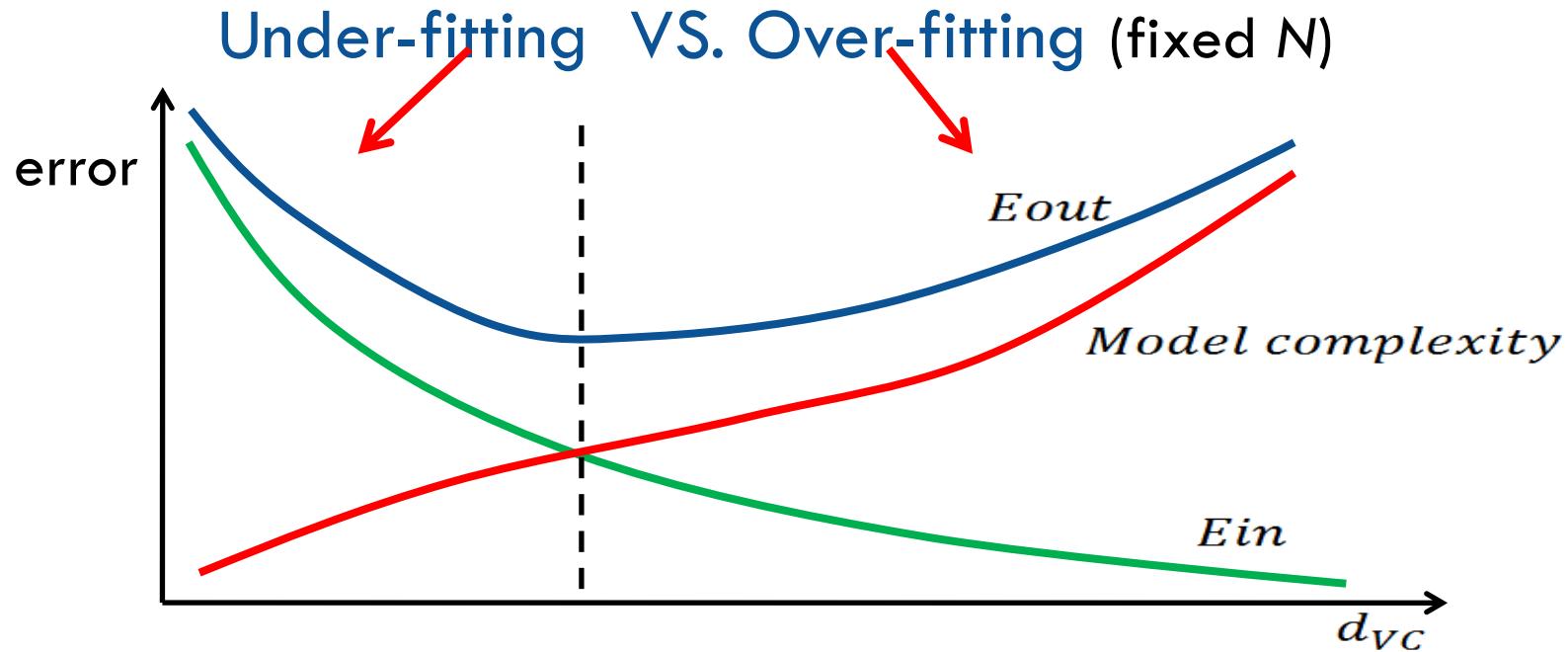
$$\text{error} = \frac{1}{N} \sum_{n=1}^N [y_n \neq g(x_n)]$$

$$Eout(g) \leq Ein(g) \pm O\left(\sqrt{\frac{d_{VC}}{N} \ln N}\right)$$

E-in: for training set
E-out: for testing set

- Non supervisé: erreur de quantification minimale, distance minimale, MLE (estimation du maximum de vraisemblance)

What are we seeking?



Learning techniques

- Supervised learning categories and techniques
 - **Linear classifier** (numerical functions)
 - **Parametric** (Probabilistic functions)
 - Naïve Bayes, Gaussian discriminant analysis (GDA), Hidden Markov models (HMM), Probabilistic graphical models
 - **Non-parametric** (Instance-based functions)
 - K-nearest neighbors, Kernel regression, Kernel density estimation, Local regression
 - **Non-metric** (Symbolic functions)
 - Classification and regression tree (CART), decision tree
 - **Aggregation**
 - Bagging (bootstrap + aggregation), Adaboost, Random forest

Learning techniques

- Unsupervised learning categories and techniques
 - **Clustering**
 - K-means clustering
 - Spectral clustering
 - **Density Estimation**
 - Gaussian mixture model (GMM)
 - Graphical models
 - **Dimensionality reduction**
 - Principal component analysis (PCA)
 - Factor analysis

Reinforcement Learning

- L'apprentissage par renforcement est né de la rencontre entre la psychologie expérimentale et les neurosciences computationnelles.
- Il tient en quelques concepts clés simples basés sur le fait que l'agent intelligent :
 - Observe les effets de ses actions
 - Déduit de ses observations la qualité de ses actions
 - Améliore ses actions futures
- Pour définir un cadre à ce processus, nous allons donc lui **apprendre à apprendre**.

Reinforcement Learning

- L'apprentissage par renforcement consiste à apprendre par interaction avec l'environnement et, en observant le résultat de certaines actions.
 - 1. L'agent observe un état d'entrée
 - 2. Une action est déterminée par une fonction de prise de décision (politique)
 - 3. L'action est effectuée
 - 4. L'agent reçoit une résultat en fonction de son environnement
 - 5. Informations sur le résultat donnée pour cette état ou action est enregistrée
- En effectuant des actions, on observe les récompenses qui en résultent, afin de déterminer la meilleure action pour un état donné.

ML in a Nutshell

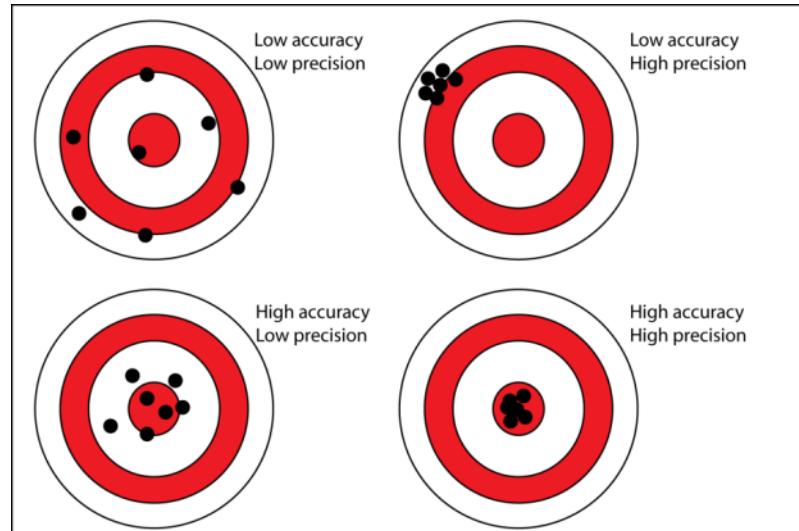
- Des dizaines de milliers d'algorithmes d'apprentissage automatique
- Des centaines de nouvelles chaque année
- Chaque algorithme d'apprentissage machine a trois composants:
 - Représentation
 - Évaluation
 - Optimisation

Representation

- Decision trees
- Sets of rules / Logic programs
- Instances
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- Etc.

Evaluation

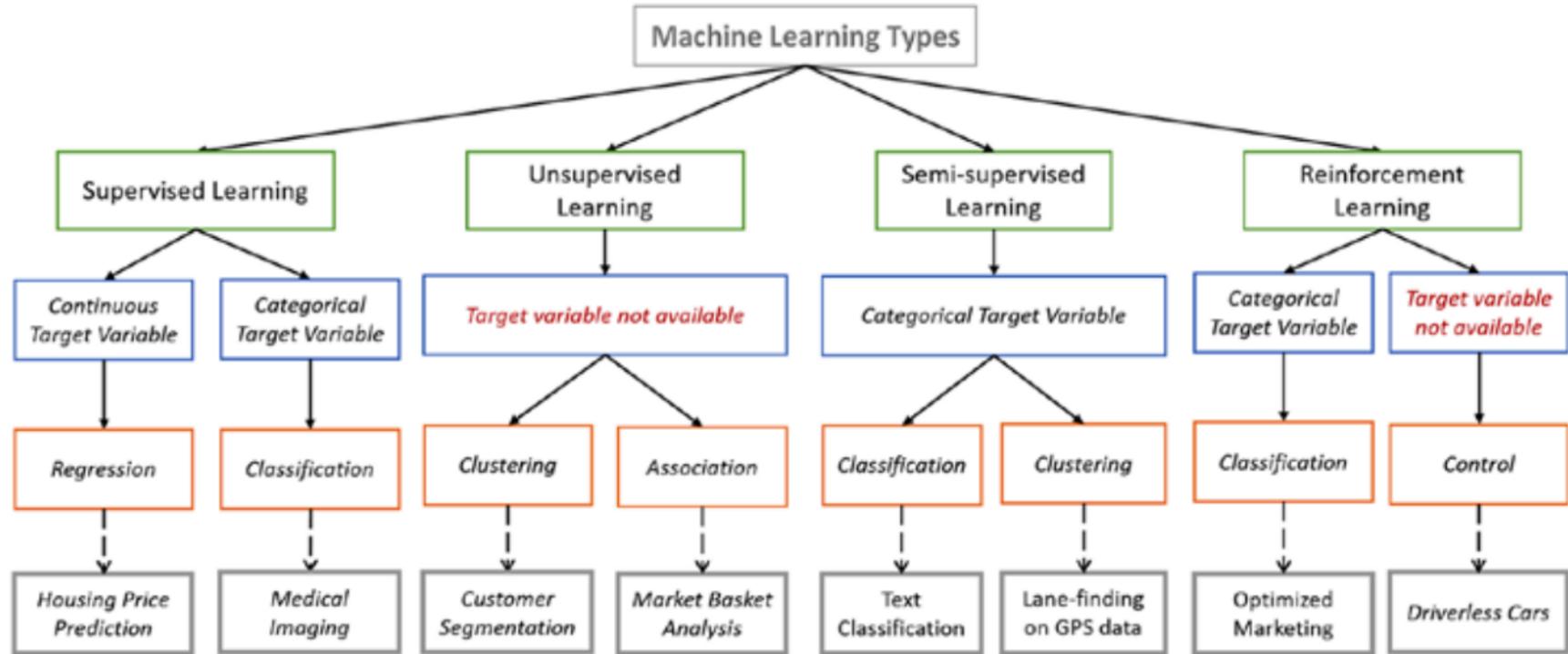
- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.



Optimization

- Optimisation combinatoire
 - E.g.: Greedy search
- Optimisation convexe
 - E.g.: Gradient descent
- Optimisation sous contrainte
 - E.g.: Linear programming

Tech. Overview



Steps in developing a machine learning application

1. Collect data.
2. Prepare the input data.
3. Analyze the input data.
4. Filter garbage
5. Train the algorithm.
6. Test the algorithm.
7. Use it.

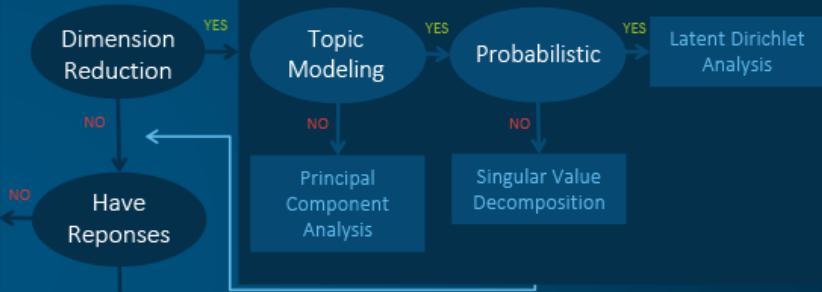
Machine Learning Algorithms Cheat Sheet

Unsupervised Learning: Clustering

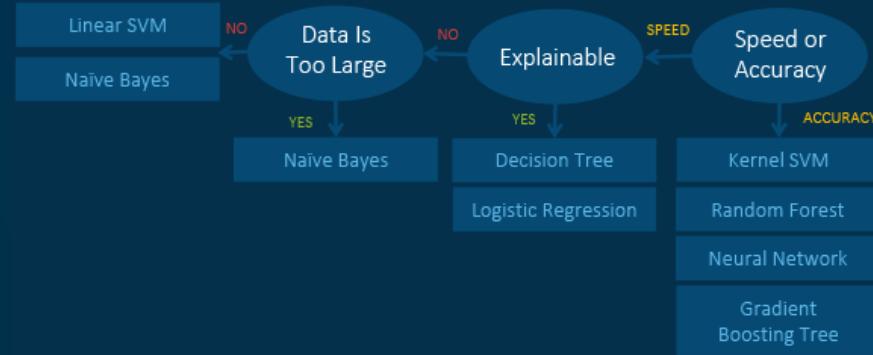


START

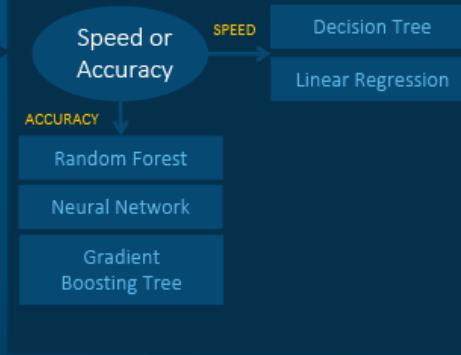
Unsupervised Learning: Dimension Reduction



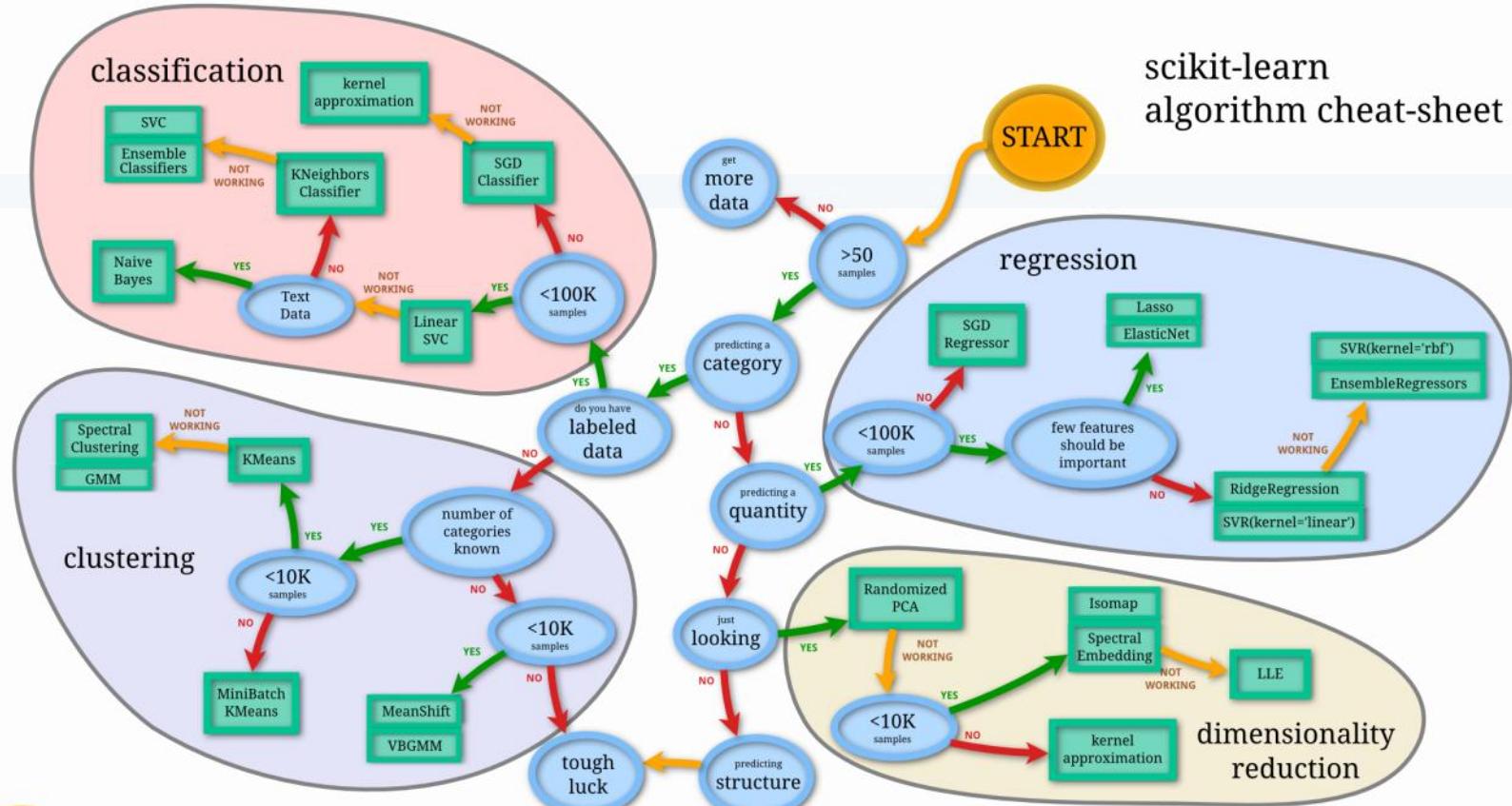
Supervised Learning: Classification



Supervised Learning: Regression



scikit-learn algorithm cheat-sheet



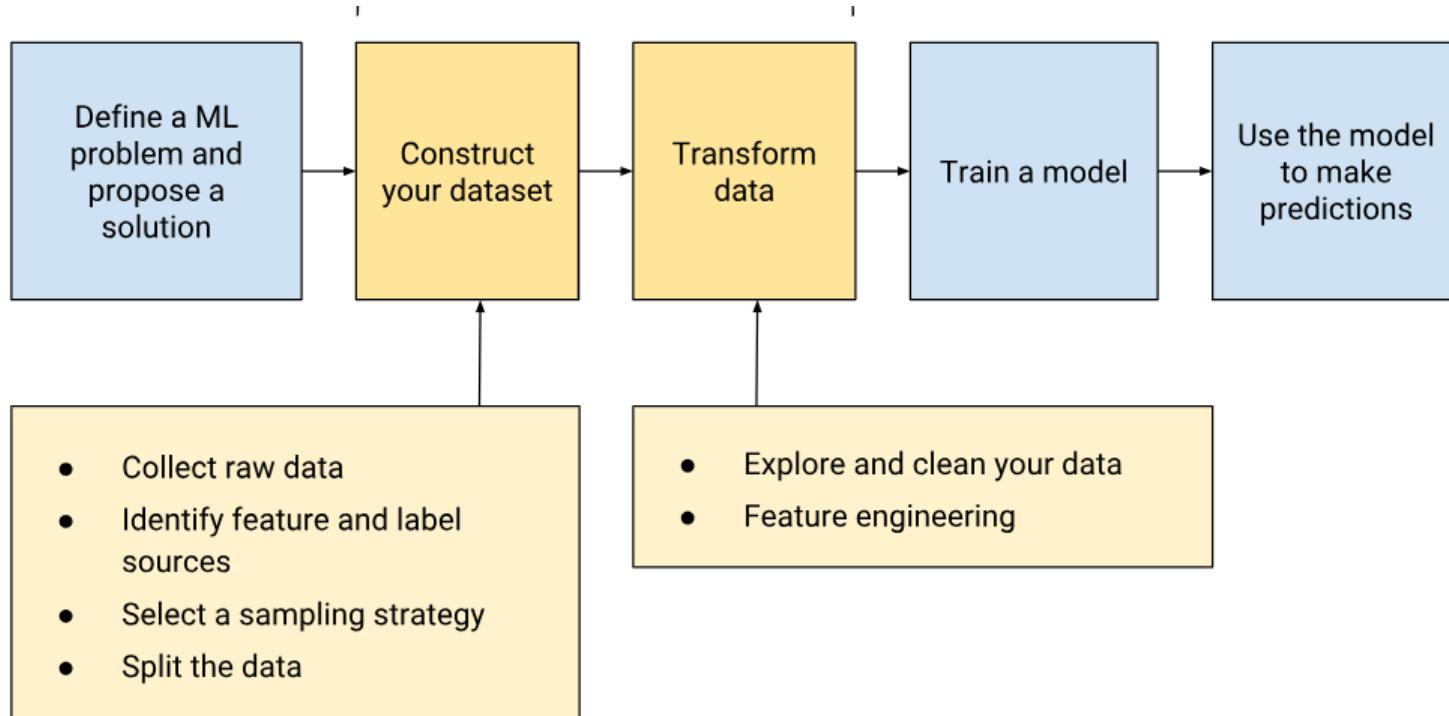
Back

scikit
learn

Data

Data Preparation (Data pre-processing)

The Process for Data Preparation and Feature Engineering



Steps to Constructing Your Dataset

- Pour construire votre jeu de données (et avant de procéder à la transformation des données), vous devez:
 1. Collecter les données brutes.
 2. Identifier les sources des caractéristiques et d'étiquettes.
 3. Sélectionnez une stratégie d'échantillonnage.
 4. Fractionner les données.
- Ces étapes dépendent beaucoup de la manière dont vous avez cadre votre problème de ML.

Self-check of Problem Framing and Data Collection Concepts

- Vous êtes sur un tout nouveau projet d'apprentissage automatique, sur le point de sélectionner vos premières caractéristiques.
Combien de fonctionnalités devriez-vous choisir?

Self-check of Problem Framing and Data Collection Concepts

Choisissez autant de caractéristiques que vous le pouvez afin de pouvoir commencer à observer quelles caractéristiques ont le plus fort pouvoir prédictif.

- Commencez plus petit. Chaque nouvelle fonctionnalité ajoute une nouvelle dimension à votre jeu de données d'entraînement. Lorsque la dimensionnalité augmente, le volume de l'espace augmente si rapidement et les données d'entraînement disponibles deviennent insuffisantes.

Choisissez entre 4 et 6 caractéristiques qui semblent avoir un fort pouvoir prédictif.

- Vous pourrez éventuellement utiliser autant de fonctionnalités, mais il est toujours préférable de commencer avec moins. Moins de caractéristiques signifie généralement moins de complications inutiles.

Choisissez 1 à 3 caractéristiques qui semblent avoir un fort pouvoir prédictif.

- Il est préférable que votre pipeline de collecte de données ne commence qu'avec une ou deux caractéristiques . Cela vous aidera à confirmer que le modèle ML fonctionne comme prévu. En outre, lorsque vous créez une base à partir de quelques caractéristiques , vous aurez l'impression de faire des progrès!

Self-check of Problem Framing and Data Collection Concepts

- Votre ami Ali est excité par les premiers résultats de son analyse statistique. Il dit que les données montrent une corrélation positive entre le nombre de téléchargements d'applications et le nombre des avis sur d'applications. Mais il ne sait pas s'ils l'auraient téléchargé sans voir les avis.
- Quelle réponse serait la plus utile pour Ali?

Self-check of Problem Framing and Data Collection Concepts

Vous pouvez effectuer une expérience pour comparer le comportement des utilisateurs qui n'ont pas vu la révision avec des utilisateurs similaires qui l'ont vu.

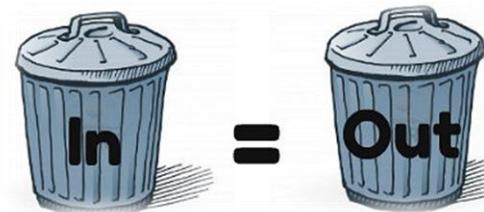
- Correct! Si Ali observe que les utilisateurs qui ont vu l'avis positif sont plus susceptibles de télécharger l'application que ceux qui ne l'ont pas encore vu, il dispose de suffisamment de preuves pour suggérer que cet avis positif encourage les utilisateurs à télécharger l'application..

Faites confiance aux données. Il est clair que cette excellente avis est la raison pour laquelle les utilisateurs téléchargent l'application.

- Incorrect. corrélation vs causalité.

The Size and Quality of a Data Set

- “Garbage in, garbage out”



- Après tout, votre modèle est aussi bon que vos données.
- Mais comment mesurez-vous la qualité de votre jeux de données et comment l'améliorez-vous? Et combien de données avez-vous besoin pour obtenir des résultats utiles? Les réponses dépendent du type de problème que vous résolvez.

The Size of a Data Set

- En règle générale, votre modèle doit s'entraîner sur au moins un ordre de grandeur supérieur à celui des paramètres pouvant être formés. Les modèles simples sur de grands jeux de données sont généralement plus sophistiqués que les modèles sophistiqués sur de petit jeux de données

Data set	Size (number of examples)
Iris flower data set	150 (total set)
MovieLens (the 20M data set)	20,000,263 (total set)
Google Gmail SmartReply	238,000,000 (training set)
Google Books Ngram	468,000,000,000 (total set)
Google Translate	trillions

The Quality of a Data Set

- Il est inutile d'avoir beaucoup de données si ce sont de mauvaises données; La qualité compte aussi.
 - Mais, que doit-on considérer comme "qualité"?
 - C'est un terme flou.
- Envisagez d'adopter une approche empirique et de choisir l'option qui produit le meilleur résultat. Dans cet état d'esprit, un ensemble de données de qualité vous permet de résoudre le problème qui vous tient à cœur. En d'autres termes, les données sont bonnes si elles remplissent la tâche à laquelle elles sont destinées.

The Quality of a Data Set

- Toutefois, lors de la collecte de données, il est utile d'avoir une définition plus concrète de la qualité. Certains aspects de la qualité ont tendance à correspondre à des modèles plus performants:
 - fiabilité
 - représentation des feautres
 - minimiser l'inclinaison

Reliability

- La **fiabilité** fait référence au degré auquel vous pouvez faire confiance à vos données. Un modèle formé sur un ensemble de données fiables est plus susceptible de produire des prévisions utiles qu'un modèle formé sur des données non fiables. En mesurant la fiabilité, vous devez déterminer:
 - 1- Quelle est la fréquence des erreurs d'étiquette? Par exemple, si vos données sont étiquetées par des humains, ils commettent parfois des erreurs.
 - 2- Vos features sont-ils bruyants? Par exemple, les mesures GPS fluctuent. Un peu de bruit est d'accord. Vous ne purgerez jamais votre ensemble de données de tout bruit. Vous pouvez aussi collecter plus d'exemples.
 - 3- Les données sont-elles correctement filtrées pour votre problème? Par exemple, votre ensemble de données devrait-il inclure des requêtes de recherche de robots?
 - Si vous construisez un système de détection de spam, la réponse est probablement oui, mais si vous essayez d'améliorer les résultats de recherche pour les humains, alors non.

Reliability

- Qu'est-ce qui rend les données peu fiables?
- de nombreux exemples dans les ensembles de données ne sont pas fiables pour l'une ou plusieurs des raisons suivantes:
 - Valeurs omises. Par exemple, une personne a oublié d'entrer une valeur correspondant à l'âge de la maison.
 - Des exemples en double. Par exemple, un serveur a téléchargé deux fois par erreur les mêmes journaux.
 - Mauvaises étiquettes. Par exemple, une personne a mal étiqueté l'image d'un chêne comme étant un érable.
 - Mauvaises valeurs de fonctionnalité. Par exemple, quelqu'un a tapé un chiffre supplémentaire ou un thermomètre a été laissé au soleil.

Feature Representation

- La représentation est la mise en correspondance des données avec des fonctionnalités utiles. Vous voudrez peut-être examiner les questions suivantes:
 - Comment les données sont-elles exposées au modèle?
 - Devez-vous normaliser les valeurs numériques?
 - Comment devriez-vous gérer les valeurs aberrantes?

Identifying Labels and Sources



Direct vs. Derived Labels

- L'apprentissage automatique est plus facile lorsque vos étiquettes sont bien définies. La meilleure étiquette est une étiquette **directe** de ce que vous voulez prédire. Par exemple, si vous voulez prédire si un utilisateur est un fan de Adil Imam, une étiquette directe serait "L'utilisateur est un fan de Adil Imam".
- Un simple test pourrait consister à déterminer si l'utilisateur a visionné une vidéo de Adil Imam sur YouTube. Le libellé "L'utilisateur a visionné une vidéo de Adil Imam sur YouTube" est un libellé dérivé car il ne mesure pas directement ce que vous souhaitez prédire. Cette étiquette **dérivée** est-elle un indicateur fiable que l'utilisateur aime Adil Imam? Votre modèle ne sera aussi bon que si le lien entre votre étiquette dérivée et votre prédiction souhaitée est bon.

Label Sources

- La sortie de votre modèle peut être un événement ou un attribut. Cela entraîne les deux types d'étiquettes suivants:
- **Libellé direct pour les événements**, tel que "L'utilisateur a-t-il cliqué sur le premier résultat de recherche?"
- **Libellé direct pour les attributs**, tel que «L'annonceur dépensera-t-il plus de X \$ la semaine prochaine?»

Échantillonnage et fractionnement des données

- Il est souvent difficile de collecter suffisamment de données pour un projet d'apprentissage automatique. Cependant, parfois, il y a trop de données et vous devez sélectionner un sous-ensemble d'exemples pour l'entraînement.

Échantillonnage et fractionnement des données

- Comment sélectionner un sous-ensemble?
- À titre d'exemple, considérons Google Search. À quelle granularité voulez-vous échantillonner ses énormes quantités de données? Voulez-vous utiliser des requêtes aléatoires? Des sessions aléatoires? Des utilisateurs aléatoires?

Sampling and Splitting Data

- En fin de compte, la réponse dépend du problème: que voulons-nous prédire et quelles caractéristique voulons-nous?
- Pour utiliser la caractéristique "requête précédente", vous devez échantillonner au niveau de la session, car les sessions contiennent une séquence de requêtes.
- Pour utiliser la caractéristique "comportement utilisateur des jours précédents", vous devez échantillonner au niveau utilisateur.

Imbalanced Data

- Un ensemble de données de classification avec des proportions de classe asymétriques est appelé déséquilibré. Les classes qui constituent une grande proportion de l'ensemble de données sont appelées classes majoritaires. Ceux qui constituent une proportion moindre sont des classes minoritaires.

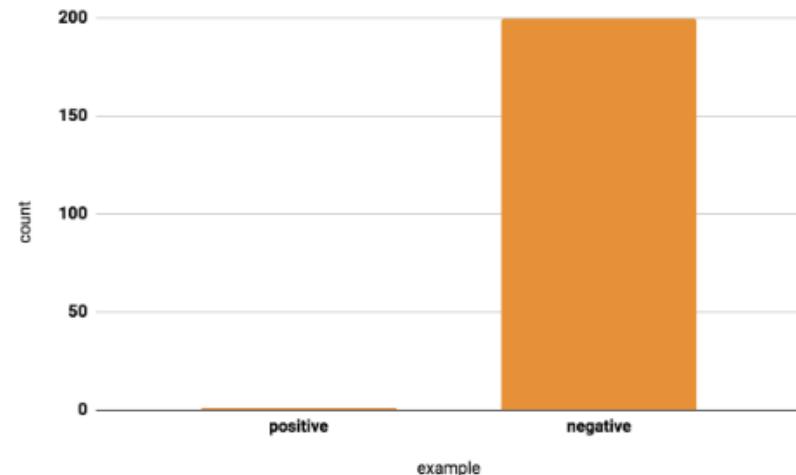
Imbalanced Data

- What counts as imbalanced? The answer could range from mild to extreme, as the table below shows.

Degree of imbalance	Proportion of Minority Class
Mild	20-40% of the data set
Moderate	1-20% of the data set
Extreme	<1% of the data set

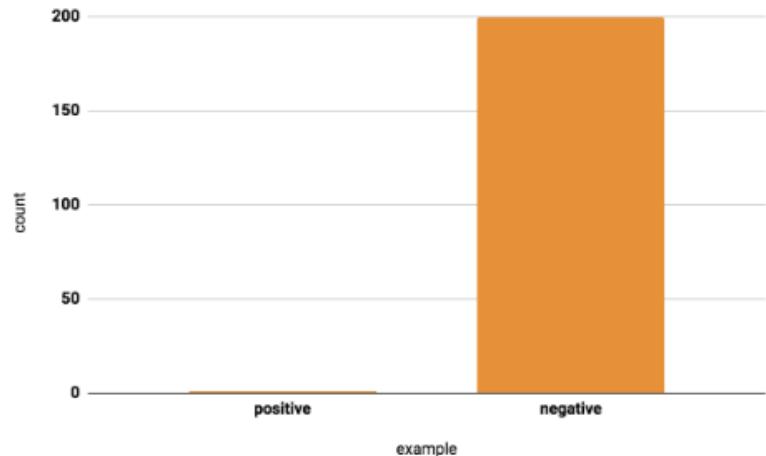
Pourquoi chercher des données déséquilibrées?

- Prenons l'exemple suivant d'un modèle qui détecte la fraude. Les cas de fraude se produisent une fois par 200 transactions dans cet ensemble de données. Ainsi, dans la vraie distribution, environ 0,5% de



Why look out for imbalanced data?

- Pourquoi cela serait-il problématique?



Data Split Example

- Après avoir collecté vos données et échantillonné si nécessaire, l'étape suivante consiste à fractionner vos données en des ensembles d'entraînement, de validation et de test.

When Random Splitting isn't the Best Approach

- Bien que le fractionnement aléatoire soit la meilleure approche pour de nombreux problèmes de ML, ce n'est pas toujours la bonne solution.
- Par exemple, considérons des ensembles de données dans lesquels les exemples sont naturellement regroupés en exemples similaires.

When Random Splitting isn't the Best Approach



Figure 1. News Stories are Clustered.

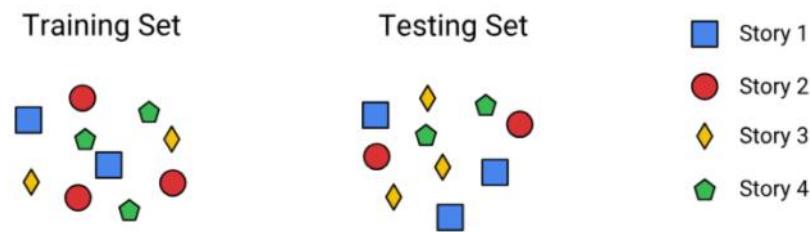


Figure 2. A random split will split a cluster across sets, causing skew.

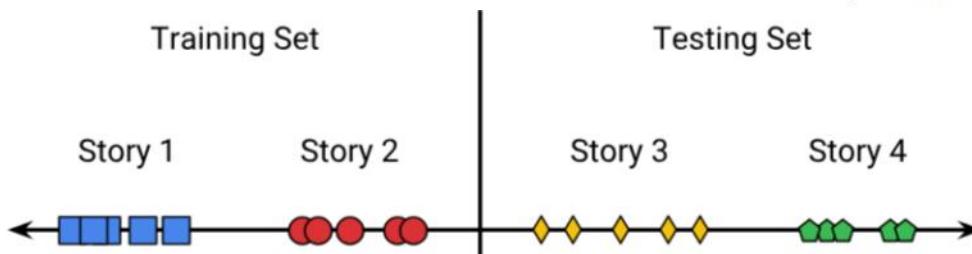


Figure 3. Splitting on time allows the clusters to mostly end up in the same set.

Why Our model perform poor sometime?

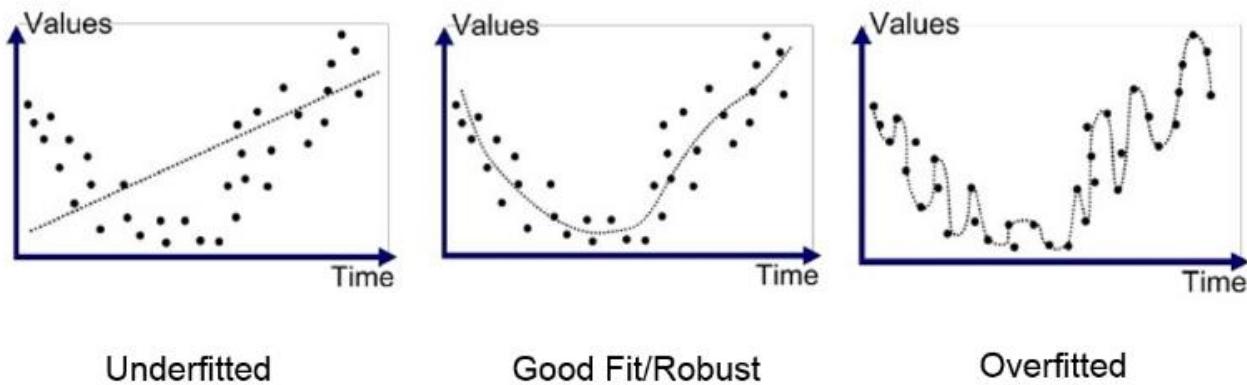
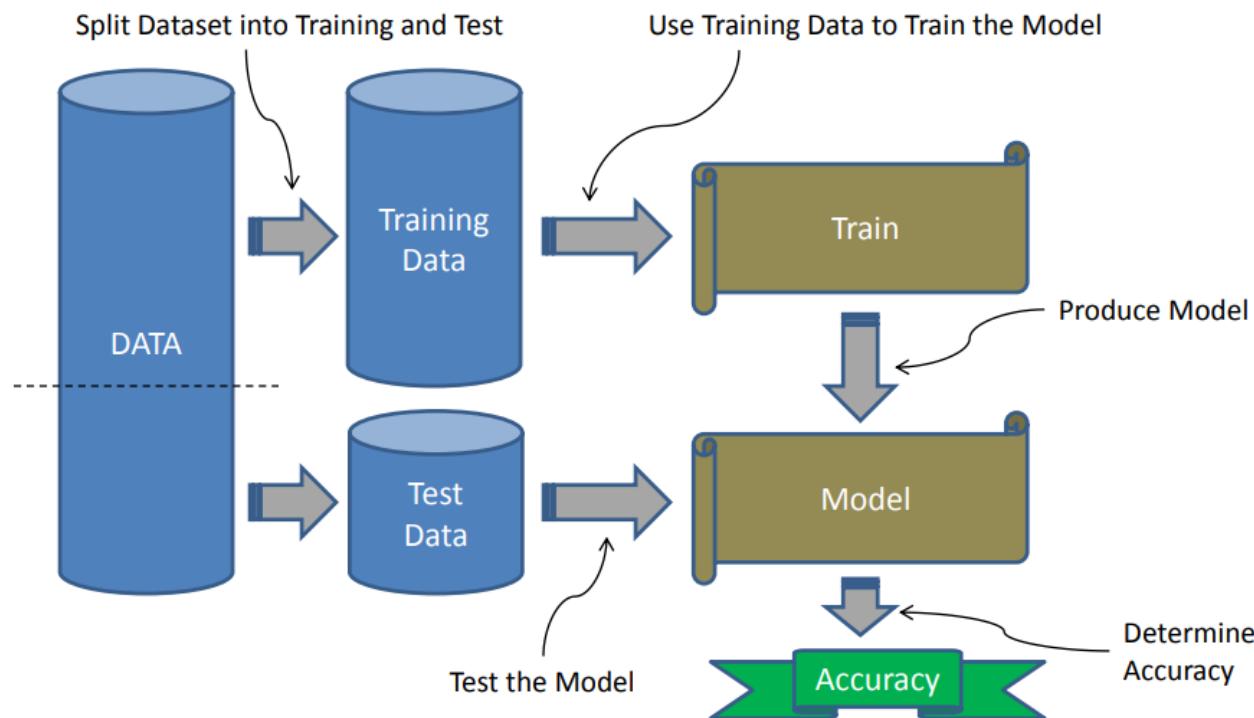


Image Source: <http://blog.algotrading101.com/design-theories/what-is-curve-fitting-overfitting-in-trading/>

Recall ...



imbalanced data - overfitting

- Si les données d'apprentissage sont trop déséquilibrées, le modèle prédit un résultat non significatif.
 - Par exemple, si le modèle est un classificateur binaire (par exemple, chat contre chien) et que presque tous les échantillons portent la même étiquette (par exemple, chat), le modèle apprendra simplement que tout est une étiquette (chat). .
- C'est ce qu'on appelle l'**overfitting**. Pour éviter les surajustements, il faut une répartition à peu près égale des échantillons d'apprentissage pour chaque classification ou plage si le libellé est une valeur réelle.

How To Limit Overfitting

- Il existe deux techniques importantes que vous pouvez utiliser lors de l'évaluation d'algorithmes d'apprentissage automatique pour limiter le sur-apprentissage :
 - Utilisez une technique de rééchantillonnage pour estimer la précision du modèle.
 - Retenir un jeu de données de validation.

Underfitting

- Le sous-apprentissage fait référence à un modèle qui ne peut ni modéliser les données d'apprentissage ni se généraliser à de nouvelles données.

Web Scraping

Web Scraping

API

- **Beautiful Soup**
 - is a tool which help programmer quickly extract valid data from web pages
- **Scrapy**
 - is a web crawling framework

Example

Features engineering



what is Feature Engineering?

- c'est l'art / science de **représenter les données** de la meilleure façon possible.
- Une bonne ingénierie des caractéristiques implique un mélange élégant de connaissances de domaine, d'intuition et de capacités mathématiques de base.

what ‘best’?

- Essentiellement, la manière dont vous présentez vos données à votre algorithme doit indiquer les **structures/propriétés** pertinentes des informations sous-jacentes de la manière la plus efficace possible.
- Lorsque vous effectuez du **Feature Engineering**, vous convertissez essentiellement vos **attributs** de données en **features** (caractéristique) de données.

Attributes

- Les attributs sont essentiellement toutes les dimensions présentes dans vos données.
- Mais tous, au format brut, représentent-ils les tendances sous-jacentes que vous souhaitez apprendre de la meilleure façon possible? Peut être pas.

Feature

- Ainsi, ce que vous faites dans l'ingénierie des caractéristiques, c'est de **prétraiter** vos données, de sorte que votre modèle / algorithme d'apprentissage doit consacrer un **minimum d'effort** à la gestion du bruit.
- Le «bruit» désigne toute information qui n'est pas pertinente pour apprendre / prédire votre objectif ultime.
- Utiliser de bonnes caractéristiques peut même vous permettre d'utiliser des modèles beaucoup plus simples.

Feature

- Comme pour toute technique d'apprentissage automatique, utilisez toujours la validation pour vous assurer que les nouvelles caractéristiques que vous introduisez améliorent réellement vos prévisions, au lieu d'ajouter une complexité inutile à votre pipeline..

Example

- Representing timestamps
- Decomposing Categorical Attributes
- Binning/Bucketing
- Feature Crosses
- Feature Selection
- Feature Scaling (data normalization)
- Feature Extraction

Representing timestamps

- Time-stamp attributes are usually denoted by the EPOCH time or split up into multiple dimensions such as (Year, Month, Date, Hours, Minutes, Seconds)
- But in many applications, a lot of that information is unnecessary.

Representing timestamps

- Consider for example a supervised system that tries to predict traffic levels in a city as a function of Location+Time.
- In this case, trying to learn trends that vary by seconds would mostly be misleading. The year wouldn't add much value to the model as well.

Representing timestamps

- Hours, day and month are probably the only dimensions you need. So when representing the time, try to ensure that your model does require all the numbers you are providing it.
- If your data sources come from different geographical sources, do remember to normalize by time-zones if needed.

Decomposing Categorical Attributes

- Certains attributs sont de type catégorie au lieu de nombre.
- Par exemple, la valeur de l'attribut ‘color’ correspond à {Red, Green, Blue}.
 - Convertissez chaque catégorie en un attribut binaire prenant une valeur sur {0, 1}. (**encodage one-hot**.)

One hot encoding

- Un **encodage one-hot** est un processus par lequel les variables catégorielles sont converties en une forme qui pourrait être fournie aux algorithmes

ML pour améliorer
le travail de prédition.

CompanyName	Categoricalvalue	Price
VW	1	20000
Acura	2	10011
Honda	3	50000
Honda	3	10000

Binning/Bucketing

- Parfois, il est plus logique de représenter un attribut numérique en tant qu'attribut catégorique.
- Exemple: considérons le problème de prédire si une personne est propriétaire d'un vêtement déterminé ou non.
 - L'âge pourrait certainement être un facteur ici. Ce qui est réellement plus pertinent, c'est **le groupe d'âge**.
 - Donc, ce que vous pourriez faire, c'est avoir des intervalles telles que 1-10, 11-18, 19-25, 26-40, etc.

Binning

- *classement ou regroupement de données (parfois appelé quantification) est un outil important dans la préparation de données numériques pour l'apprentissage automatique.*

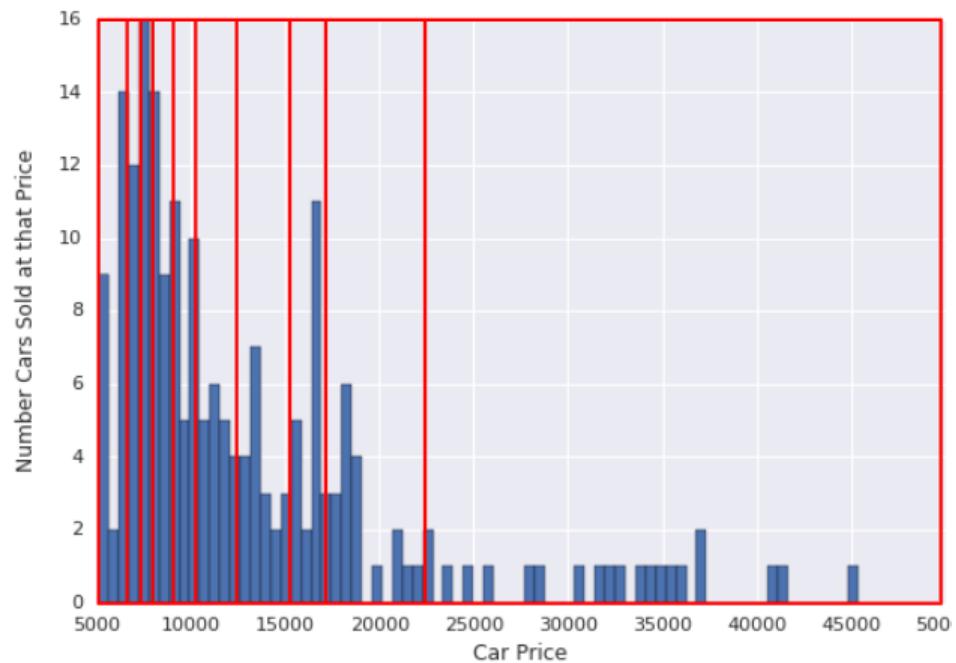
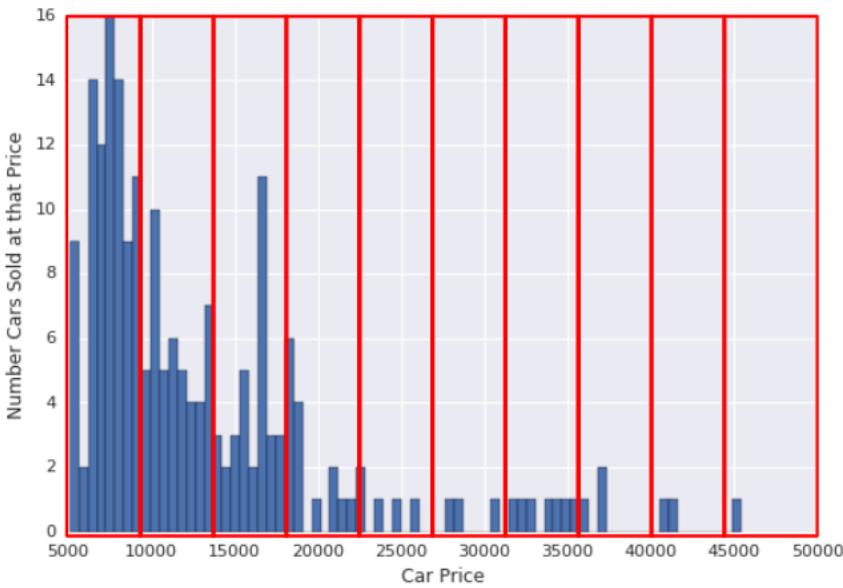
Binning

- Binning also reduces the effect of tiny errors, by 'rounding off' a given value to the nearest representative.
- Binning does *not* make sense if the number of your ranges is comparable to the total possible values, or if precision is very important to you.

Bucketing

- Bucketing makes sense when the domain of your attribute can be divided into **neat ranges**, where all numbers falling in a range imply a **common characteristic**.
 - It reduces overfitting in certain applications

Example

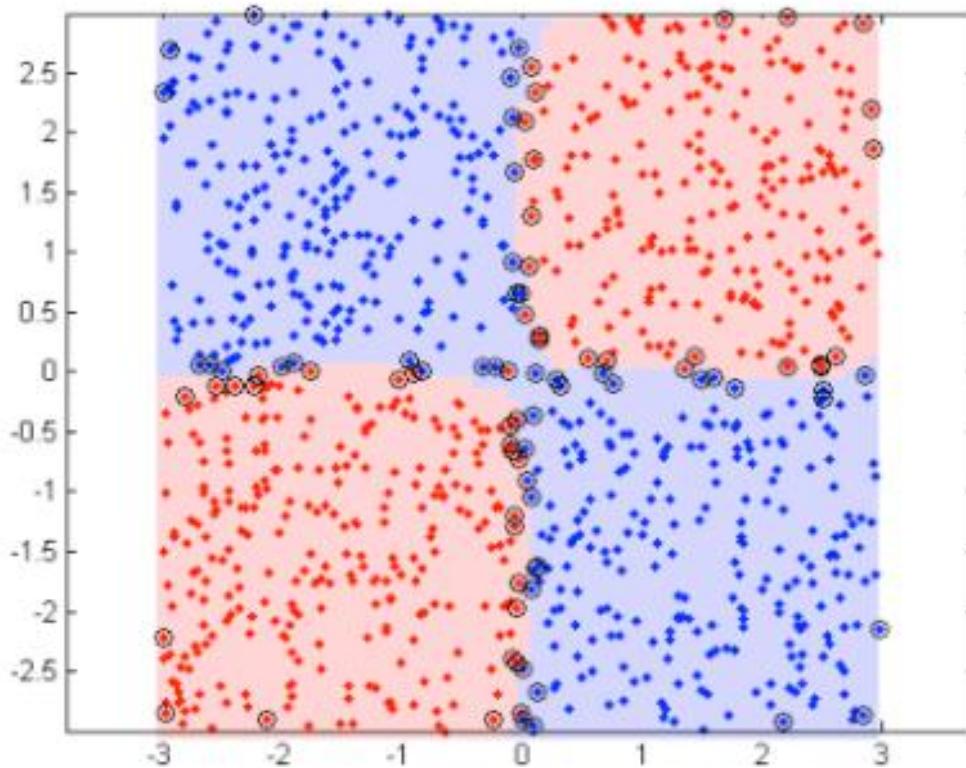


<https://developers.google.com/machine-learning/data-prep/transform/bucketing>

Feature crosses

- **Feature crosses (*croisements de caractéristiques*)** sont un moyen unique de combiner deux attributs catégoriques ou plus en un seul.
- C'est une technique extrêmement utile, lorsque certaines caractéristiques désignent ensemble une propriété mieux qu'individuellement.
 - Mathématiquement, vous faites un produit croisé entre toutes les valeurs possibles des caractéristiques catégorielles.

Example



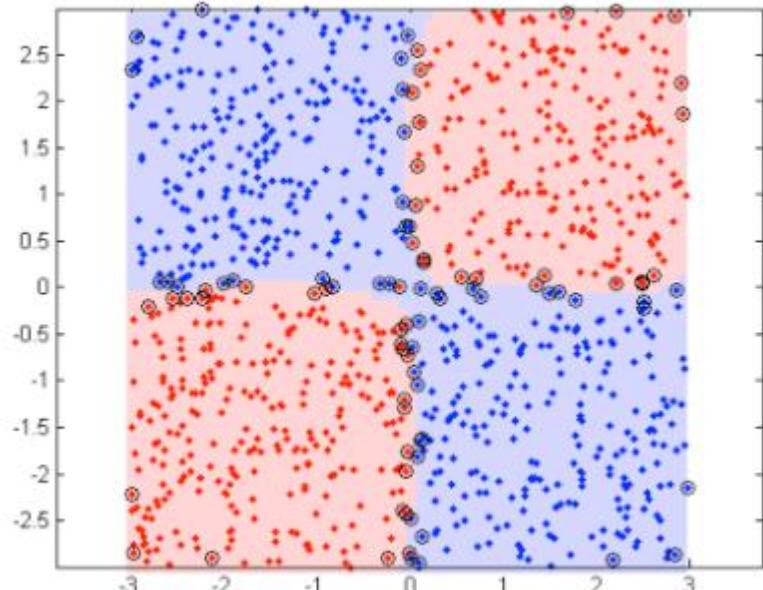
Feature crosses

- Considérons une caractéristique A, avec deux valeurs possibles {A1, A2}. Soit B une caractéristique avec deux possibilités {B1, B2}. Alors, un croisement entre A et B (appelons-le AB) prend l'une des valeurs suivantes: $\{(A1, B1), (A1, B2), (A2, B1), (A2, B2)\}$.

Example

- Tout d'abord, vous auriez intérêt à regrouper les valeurs X, Y respectivement dans
- $\{x < 0, x \geq 0\}$ & $\{y < 0, y \geq 0\}$.
- Appelons-les $\{X_n, X_p\}$ et $\{Y_n, Y_p\}$.
- Il est assez évident que les **quadrants I et III** correspondent à la classe rouge et que les **quadrants II et IV** contiennent la classe bleue.

Donc, si vous pouviez maintenant croiser les caractéristiques X et Y dans un seul quadrant, vous auriez essentiellement $\{I, II, III, IV\}$ équivalent à $\{(X_p, Y_p), (X_n, Y_p), (X_n, Y_n), (X_p, Y_n)\}$.



Exemple

- supposons que vous définissez les caractéristiques modifiées : X_{sign} , Y_{sign}
- Maintenant, vous pouvez simplement définir une nouvelle caractéristique comme suit :

$$X_{\text{sign}} = \frac{x}{|x|} \quad \text{and} \quad Y_{\text{sign}} = \frac{y}{|y|}$$

-
- $\text{Quadrant}_{\text{odd}} = X_{\text{sign}} Y_{\text{sign}}$
- Thats all! If , $\text{Quadrant}_{\text{odd}} = 1$ the class is Red. Else, Blue!

Feature selection

- Feature selection... désigne le processus de **sélection d'un sous-ensemble de caractéristiques pertinentes** à utiliser dans la construction d'un modèle
- Feature selection est un autre élément clé du processus d'apprentissage automatique appliqué, comme la sélection du modèle.
- Il est important de considérer la **sélection des caractéristiques** comme une partie du processus de sélection du modèle. Si vous ne le faites pas, vous risquez d'introduire par inadvertance des biais dans vos modèles, ce qui peut entraîner un sur-apprentissage.

Feature selection

- La sélection des caractéristiques est différente de la réduction de la dimension. Les deux méthodes cherchent à réduire le nombre d'attributs dans le jeu de données, mais une méthode de réduction de dimension consiste à créer de nouvelles combinaisons d'attributs, alors que les méthodes de sélection des caractéristiques incluant et excluant les attributs présents dans les données sans les modifier.

Feature Selection

- **Feature Selection** : Utilisation de certains algorithmes pour sélectionner automatiquement un sous-ensemble de Features d'origine pour le modèle final.
- Ici, vous ne créez pas / ne modifiez pas vos caractéristiques actuelles, vous les raffinez pour réduire le bruit / la redondance.

Voir: <http://jmlr.csail.mit.edu/papers/volume3/guyon03a/guyon03a.pdf>

Feature selection

- *Feature selection est utile en soi, mais elle agit principalement comme un filtre, en désactivant les caractéristiques inutiles.*

- Les méthodes de sélection de caractéristiques peuvent être utilisées pour identifier et désactiver les attributs inutiles, non pertinents et redondants des données qui ne contribuent pas à la précision d'un modèle prédictif ou qui peuvent en fait diminuer la précision du modèle.

Objective

- *L'objectif de la sélection des variables est triple:*
 1. *Améliorer les performances de prédiction des prédicteurs,*
 2. *Fournir des prédicteurs plus rapides et plus économiques,*
 3. *Fournir une meilleure compréhension du processus sous-jacent qui a généré les données.*

Feature Selection Algorithms

- There are three general classes of feature selection algorithms:
 1. filter methods,
 2. wrapper methods
 3. and embedded methods.

Filter feature selection

- Ce sont des méthodes qui applique une mesure statistique pour donner un score à chaque feature .
- Les features sont classées par score et chaque features sélectionnés pour être conservés ou supprimés de l'ensemble de données.
 - Parmi les exemples de méthodes de filtrage, citons le Chi squared test, information gain et correlation coefficient scores.

Wrapper methods

- Les méthodes wrappers considèrent la sélection d'un ensemble de caractéristiques comme un problème de **recherche**, dans lequel différentes combinaisons sont préparées, évaluées et comparées à d'autres combinaisons.
- Un modèle prédictif est utilisé pour **évaluer** une combinaison de caractéristiques et attribuer un score basé sur la **précision** du modèle.
- Le processus de recherche peut être méthodique, par exemple une recherche optimale, il peut être **stochastique**, comme un algorithme aléatoire, ou il peut utiliser des méthodes **heuristiques**, telles que les passes avant et arrière pour ajouter ou supprimer des entités.
 - Exemple : Algorithme récursive d'élimination de feature.

Embedded Methods

- Les méthodes embedded déterminent quelles features contribuent le plus à la précision du modèle lors de sa création.
- Le type le plus courant de ces méthodes est la méthode de **régularisation**.
- Les méthodes de **régularisation** sont également appelées méthodes de **pénalisation**, lesquelles introduisent des contraintes supplémentaires dans l'optimisation d'un algorithme prédictif (tel qu'un algorithme de régression) qui orientent le modèle vers une complexité inférieure (moins de coefficients).
- Exemples d'algorithmes : LASSO, Elastic Net and Ridge Regression.

Features selection



FIGURE 1 – L'approche filtre.

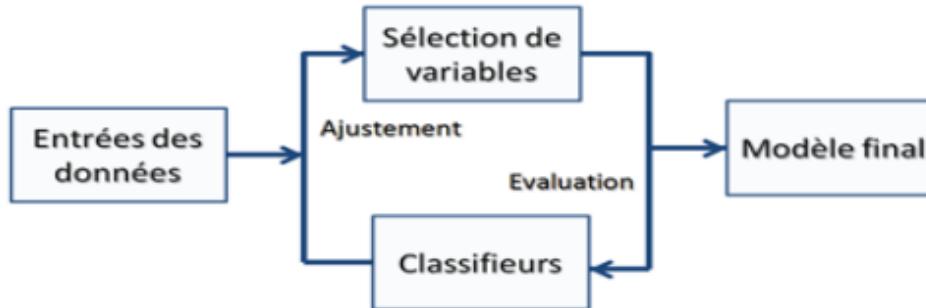


FIGURE 2 – Principe de l'approche wrapper et embedded.

sklearn.feature_selection

<code>feature_selection.GenericUnivariateSelect ([...])</code>	Univariate feature selector with configurable strategy.
<code>feature_selection.SelectPercentile ([...])</code>	Select features according to a percentile of the highest scores.
<code>feature_selection.SelectKBest ([score_func, k])</code>	Select features according to the k highest scores.
<code>feature_selection.SelectFpr ([score_func, alpha])</code>	Filter: Select the pvalues below alpha based on a FPR test.
<code>feature_selection.SelectFdr ([score_func, alpha])</code>	Filter: Select the p-values for an estimated false discovery rate
<code>feature_selection.SelectFromModel (estimator)</code>	Meta-transformer for selecting features based on importance weights.
<code>feature_selection.SelectFwe ([score_func, alpha])</code>	Filter: Select the p-values corresponding to Family-wise error rate
<code>feature_selection.RFE (estimator[, ...])</code>	Feature ranking with recursive feature elimination.
<code>feature_selection.RFECV (estimator[, step, ...])</code>	Feature ranking with recursive feature elimination and cross-validated selection of the best number of features.
<code>feature_selection.VarianceThreshold ([threshold])</code>	Feature selector that removes all low-variance features.
<code>feature_selection.chi2 (X, y)</code>	Compute chi-squared stats between each non-negative feature and class.
<code>feature_selection.f_classif (X, y)</code>	Compute the ANOVA F-value for the provided sample.
<code>feature_selection.f_regression (X, y[, center])</code>	Univariate linear regression tests.
<code>feature_selection.mutual_info_classif (X, y)</code>	Estimate mutual information for a discrete target variable.
<code>feature_selection.mutual_info_regression (X, y)</code>	Estimate mutual information for a continuous target variable.

Removing features with low variance

- VarianceThreshold est une approche de base simple pour la sélection des features. Il supprime toutes les features dont la variance ne correspond pas à un seuil. Par défaut, il supprime toutes les caractéristiques à zéro variance, c'est-à-dire les entités qui ont la même valeur dans tous les échantillons..

Exemple

- Supposons que nous ayons un jeu de données avec des features booléennes et que nous souhaitons supprimer toutes les features qui sont un ou zéro (activé ou désactivé) dans plus de 80% des échantillons. Les features booléennes sont des variables aléatoires de Bernoulli, et leur variance est donnée par: $\text{Var}[X]=p(1-p)$
- Donc nous puissions sélectionner en utilisant le seuil **.8 * (1 - .8)**

Exemple

```
Entrée [1]: In[1]: from sklearn.feature_selection import VarianceThreshold
X = [[0, 0, 1], [0, 1, 0], [1, 0, 0], [0, 1, 1], [0, 1, 0], [0, 1, 1]]
sel = VarianceThreshold(threshold=(.8 * (1 - .8)))
sel.fit_transform(X)

Out[1]: array([[0, 1],
               [1, 0],
               [0, 0],
               [1, 1],
               [1, 0],
               [1, 1]])
```

As expected, `VarianceThreshold` has removed the first column, which has a probability $p=5/6 > .8$ of containing a zero.

The Recursive Feature Elimination (RFE)

- La méthode RFE (Recursive Feature Elimination) est une approche de sélection de Feature. Cela fonctionne en supprimant récursivement les attributs et en construisant un modèle sur les attributs restants.
- Elle utilise la précision du modèle pour identifier les attributs (et la combinaison d'attributs) qui contribuent le plus à la prédiction de l'attribut cible.

The Recursive Feature Elimination (RFE)

This recipe shows the use of RFE on the Iris floweres dataset to select 3 attributes.

```
Entrée [5]: # Recursive Feature Elimination
             from sklearn import datasets
             from sklearn.feature_selection import RFE
             from sklearn.linear_model import LogisticRegression
             # Load the iris datasets
             dataset = datasets.load_iris()
             # create a base classifier used to evaluate a subset of attributes
             model = LogisticRegression()
             # create the RFE model and select 3 attributes
             rfe = RFE(model, 3)
             rfe = rfe.fit(dataset.data, dataset.target)
             # summarize the selection of the attributes
             print(rfe.support_)
             print(rfe.ranking_)
```

```
[False  True  True  True]
[2 1 1 1]
```

Feature Importance

- Methods that use ensembles of **decision trees** (like Random Forest or Extra Trees) can also compute the **relative importance of each attribute**. These importance values can be used to inform a feature selection process.

Feature Importance

This code shows the construction of an “Extra Trees ensemble” of the iris flowers dataset and the display of the relative feature **importance**.

```
Entrée [6]: ┶ # Feature Importance
             from sklearn import datasets
             from sklearn import metrics
             from sklearn.ensemble import ExtraTreesClassifier
             # Load the iris datasets
             dataset = datasets.load_iris()
             # fit an Extra Trees model to the data
             model = ExtraTreesClassifier()
             model.fit(dataset.data, dataset.target)
             # display the relative importance of each attribute
             print(model.feature_importances_)
```

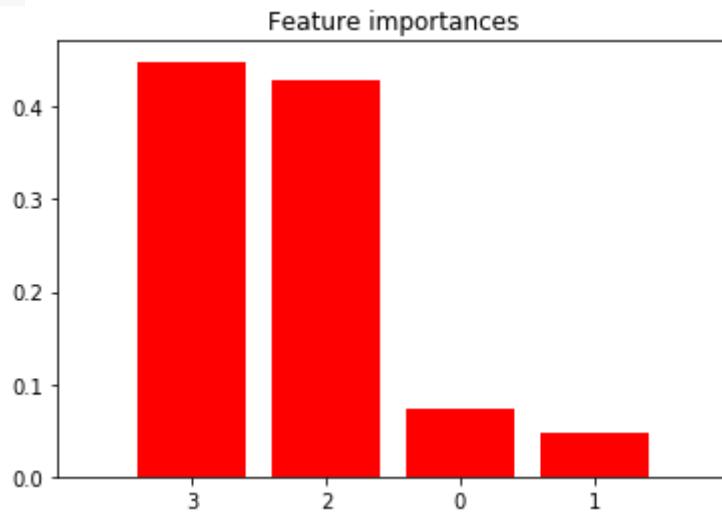
```
[0.06237856 0.04333641 0.57610379 0.31818124]
```

Feature Importance

```
importances = model.feature_importances_
std = np.std([model.feature_importances_ for tree in model.estimators_],
             axis=0)
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")
X=dataset.data
for f in range(X.shape[1]):
    print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))

# Plot the feature importances of the forest
plt.figure()
plt.title("Feature importances")
plt.bar(range(X.shape[1]), importances[indices],
        color="r", yerr=std[indices], align="center")
plt.xticks(range(X.shape[1]), indices)
plt.xlim([-1, X.shape[1]])
plt.show()
```



Le Feature Scaling

- Le Feature Scaling est une bonne pratique, pour ne pas dire obligatoire, lors de la modélisation avec du Machine Learning.
- Les algorithmes pour lesquels le feature scaling s'avère nécessaire, sont ceux pour lesquels il faudra
 - Calculer un vecteur de poids (weights) theta
 - Calculer des distances pour déduire le degré de similarité de deux items
 - Certains algorithmes de Clustering

Feature scaling

- **Feature scaling** (mise à l'échelle) des features est une méthode utilisée pour normaliser les intervalles de variables indépendantes ou des features de données.
- La plupart du temps, en machine Learning, les *Data Set* proviennent avec des ordres de grandeurs différents. Cette différence d'échelle peut conduire à des performances moindres. Pour palier à cela, des traitements préparatoires sur les données existent. Notamment le **Feature Scaling** qui comprend la **Standardisation** et la **Normalisation**.

Standardization

- La **standardisation** (aussi appelée **Z-Score normalisation**) peut- être appliquée quand les *input features* répondent à des **distributions normales** (Distributions Gaussiennes) avec des moyennes et des écart-types différents. Par conséquent, cette transformation aura pour impact d'avoir toutes nos features répondant à la même loi normale :
- La standardisation peut également être appliquée quand les features ont **des unités différentes**.

Standardisation

- La Standardisation est le processus de transformer une feature en une autre qui **répondra à la loi normale (Gaussian Distribution)** avec :
 - $\mu := 0$ La moyenne de la loi de distribution
 - $\sigma = 1$ est l'Écart-type (Standard Deviation)

Standardisation

- La formule de standardisation d'une feature est la suivante :

$$z = \frac{x - \mu}{\sigma}$$

- avec :
- x la valeur qu'on veut standardiser (input variable)
- μ la moyenne (mean) des observations pour cette feature
- σ est l'écart-type (Standard Deviation) des observations pour cette feature

Min-Max scaling

- Min-Max Scaling peut- être appliqué quand les données varient dans des échelles différentes. A l'issue de cette transformation, les features seront comprises dans un intervalle fixe [0,1].
- Le but d'avoir un tel intervalle restreint est de réduire l'espace de variation des valeurs d'une feature et par conséquent réduire l'effet des *outliers*.
- La normalisation peut- être effectuée par la technique du **Min-Max Scaling**. La transformation se fait grâce à la formule suivante :

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Exemple

- Par exemple, supposons que nous ayons les données de poids des étudiants et que leur poids s'étende à [160 cm, 200 cm]. Pour redimensionner ces données, nous soustrayons d'abord 160 du poids de chaque élève et divisons le résultat par 40 (la différence entre les poids maximum et minimum).

Mean normalization

$$x' = \frac{x - \text{average}(x)}{\max(x) - \min(x)}$$

Standardizing and normalizing - how it can be done using scikit-learn

- See Tuto [Scaling.py](#)

DataSet

Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Wine Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Using chemical analysis determine the origin of wines



Data Set Characteristics:	Multivariate	Number of Instances:	178	Area:	Physical
Attribute Characteristics:	Integer, Real	Number of Attributes:	13	Date Donated	1991-07-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	1078813

Source:

Original Owners:

Forina, M. et al, PARVUS -
An Extendible Package for Data Exploration, Classification and Correlation.
Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno,
16147 Genoa, Italy.

Donor:

Stefan Aeberhard, email: stefan '@' coral.cs.jcu.edu.au

<http://archive.ics.uci.edu/ml/datasets/Wine>

Tuto



Feature Extraction?



Source: <https://deepai.org/machine-learning-glossary-and-terms/feature-extraction>

Feature Extraction: What is ?

- L'extraction de caractéristiques est un processus de réduction de dimensionnalité par lequel un ensemble initial de données brutes est réduit à des groupes plus faciles à gérer pour le traitement.
- Une caractéristique de ces grands jeux de données est un grand nombre de variables qui nécessitent beaucoup de ressources informatiques pour être traitées.
 - L'extraction de caractéristiques est le nom de méthodes qui combinent des variables en caractéristiques, réduisant efficacement le volume de données à traiter, tout en décrivant de manière précise et complète l'ensemble de données d'origine.

Dimension reduction techniques

- With more variables, comes more trouble! And to avoid this trouble, **dimension reduction techniques** comes to the rescue.

Human Activity Recognition Using Smartphones Data Set

Download: [Data Folder](#), [Data Set Description](#)

Abstract: Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors.

Data Set Characteristics:	Multivariate, Time-Series	Number of Instances:	10299	Area:	Computer
Attribute Characteristics:	N/A	Number of Attributes:	561	Date Donated	2012-12-10
Associated Tasks:	Classification, Clustering	Missing Values?	N/A	Number of Web Hits:	773824

Source:

Jorge L. Reyes-Ortiz(1,2), Davide Anguita(1), Alessandro Ghio(1), Luca Oneto(1) and Xavier Parra(2)

1 - Smartlab - Non-Linear Complex Systems Laboratory

DITEN - Università degli Studi di Genova, Genoa (I-16145), Italy.

2 - CETpD - Technical Research Centre for Dependency Care and Autonomous Living

Universitat Politècnica de Catalunya (BarcelonaTech). Vilanova i la Geltrú (08800), Spain

activityrecognition @' smartlab.ws

General dimensionality reduction techniques :

- Independent component analysis
- Isomap
- Kernel PCA
- Latent semantic analysis
- Partial least squares
- Principal component analysis
- Multifactor dimensionality reduction
- Nonlinear dimensionality reduction
- Multilinear Principal Component Analysis
- Multilinear subspace learning
- Semidefinite embedding
- Autoencoder

Low Variance

- **Variance faible:** Imaginons un scénario dans lequel nous avons une variable constante (toutes les observations ont la même valeur, 5) dans notre ensemble de données. Pensez-vous que cela peut améliorer la puissance du modèle?
 - Bien sûr PAS, car il a une variance nulle.
 - En cas de nombre élevé de dimensions, nous devrions supprimer les variables à faible variance par rapport aux autres, car ces variables n'expliqueront pas la variation des variables cibles.

High Correlation

- **Corrélation élevée:** les dimensions présentant une corrélation plus élevée peuvent réduire les performances du modèle. De plus, il n'est pas bon d'avoir plusieurs variables d'informations ou de variations similaires, également appelées «multicolinéarité».
- Vous pouvez utiliser la matrice de corrélation de Pearson (variables continues) ou polychorique (variables discrètes) pour identifier les variables fortement corrélées et en sélectionner une à l'aide de VIF (Variance Inflation Factor). Les variables ayant une valeur plus élevée ($VIF > 5$) peuvent être supprimées.

Missing data

- Les données manquantes dans un jeu de données d'apprentissage peut réduire la puissance / l'ajustement d'un modèle ou peut conduire à un modèle anormal, car nous n'avons pas correctement analysé le comportement et la relation avec d'autres variables. Cela peut conduire à une mauvaise prédiction ou classification.



Missing values

- **1. Valeurs manquantes:** lors de l'exploration des données, si nous rencontrons des valeurs manquantes, que faisons-nous? Notre première étape devrait consister à identifier la raison, puis à imputer les valeurs manquantes / supprimer les variables à l'aide de méthodes appropriées.
- Mais que se passe-t-il si nous avons trop de valeurs manquantes? Devrions-nous assigner les valeurs manquantes ou supprimer les variables?

Complete case Analysis

- **Suppression des observations (Complete case Analysis)**
- Il s'agit de la technique la plus simple et courante. Elle consiste à **supprimer les observations** (les lignes) qui contiennent au moins une *feature manquante*. Le jeu de données résultat ne contiendra aucune observation comportant une valeur manquante. C'est le comportement par défaut dans plusieurs outils statistiques.
- Le problème de cette technique est qu'on peut être amené à supprimer un grand nombre d'observations.

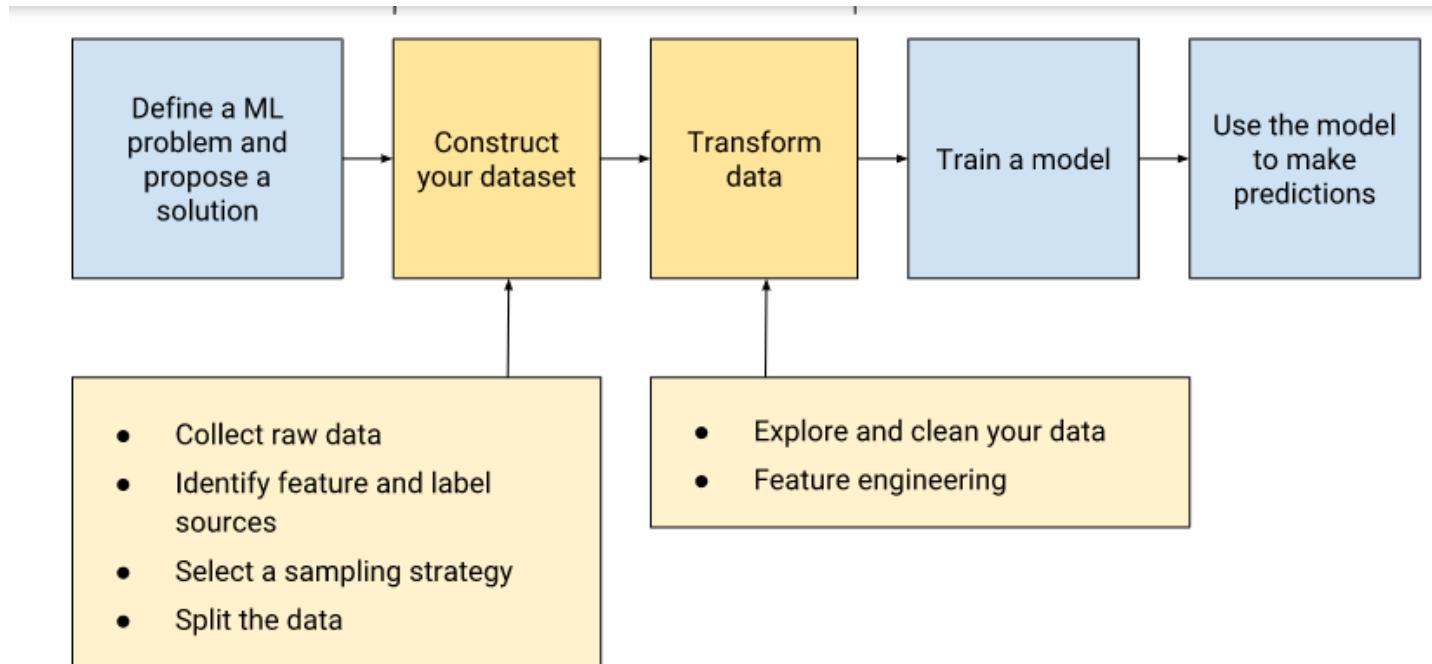
Imputation de données

- **L'imputation de données manquante** réfère au fait qu'on remplace les valeurs manquantes dans le jeu de données par des valeurs artificielles.
- Idéalement, ces remplacements ne doivent pas conduire à une altération sensible de la distribution et la composition du jeu de données.

Imputation de données

- **Imputation par règle**
 - Si on connaît le sens métier de la donnée manquante et la règle métier la régissant, on peut faire une imputation par règle.
- **Imputation par moyenne ou mode**
 - Une autre façon intuitive d'imputer les valeurs manquantes d'une feature numérique est d'utiliser par la moyenne des observations.
- **La méthode Hot Deck**
 - Traiter une valeur manquante d'une feature avec l'imputation *Hot Deck* revient à choisir aléatoirement une valeur parmi les valeurs de la même feature pour les autres observations du jeu de données.

Résumé

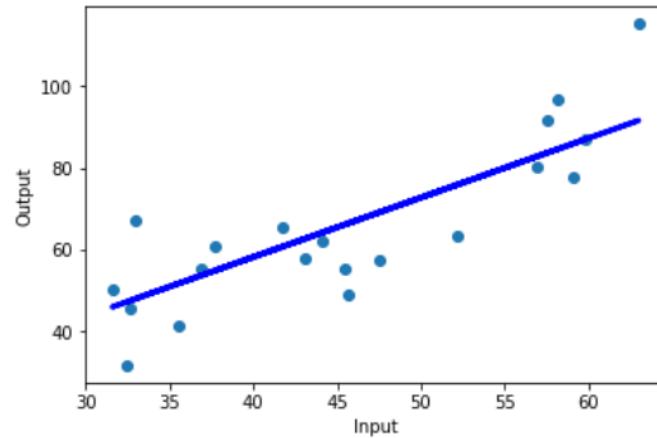




Algorithms

Linear Regression

- It is used to estimate **real values** (cost of houses, number of calls, total sales etc.) based on **continuous variable(s)**..



Linear Regression

- Linear regression is used for finding linear **relationship** between **target** and one or more **predictors**. The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

How

- Establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation $Y = a * X + b$.

$$Y' = A + B * X$$

SIMPLE REGRESSION EQUATION

- X : predictor (present in data)
- B : coefficient (estimated by regression)
- A : intercept (estimated by regression)
- Y' : predicted value (calculated from A , B and X)

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

Types

- Linear Regression is of mainly two types:
 - Simple Linear Regression;
 - Multiple Linear Regression.

Cost function & Error

- The difference between the predicted values and ground truth measures the error difference.

Cost Function

$$y = ax + b$$

- The cost function helps us to figure out the best possible values for a and b which would provide the best fit line for the data points.
- Since we want the best values for a and b , we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.

$$\text{minimize} \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

$$J = \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

Minimize the error

- The values a and b must be chosen so that they minimize the error. If sum of squared error is taken as a metric to evaluate the model, then goal to obtain a line that best reduces the error.
 - Mean Absolute Error
 - Mean Square Error
 - Mean Absolute Percentage Error
 - Mean Percentage Error
 -
- Mean Absolute Error (MAE) is the mean of the absolute value of the errors
- Mean Squared Error (MSE) is the mean of the squared errors
- Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors

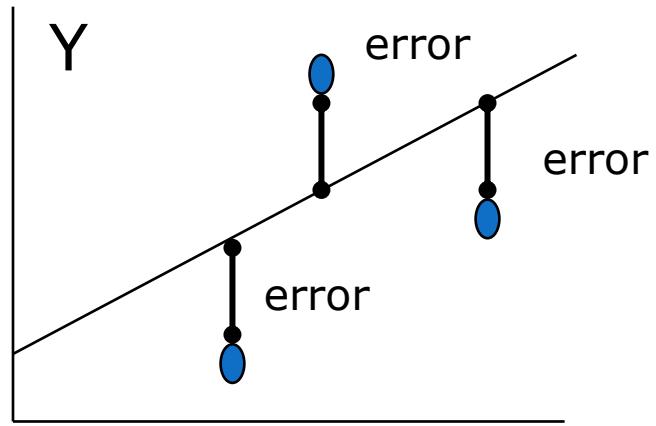
Error

$$BCE = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2$$

where N is the number of data points,
 f_i the value returned by the model and
 y_i the actual value for data point i .

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$



MSE

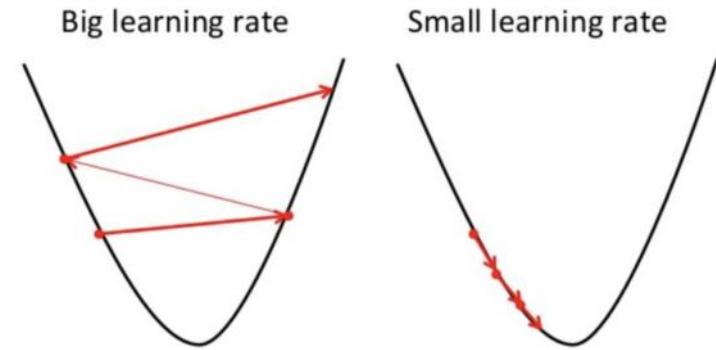
- We square the error difference and sum over all data points and divide that value by the total number of data points. This provides the average squared error over all the data points.

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2$$

where N is the number of data points, f_i the value returned by the model and y_i the actual value for data point i .

Gradient Descent

- The next important concept needed to understand linear regression is gradient descent. Gradient descent is a method of updating a and b to reduce the cost function(MSE). The idea is that we start with some values for a and b and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.



Implémentation

Python code

```
#Import Library  
  
#Import other necessary libraries like pandas, numpy...  
from sklearn import linear_model  
  
#Load Train and Test datasets  
  
#Identify feature and response variable(s) and values must be numeric and numpy arrays  
  
x_train=input_variables_values_training_datasets  
y_train=target_variables_values_training_datasets  
x_test=input_variables_values_test_datasets  
  
# Create linear regression object  
  
linear = linear_model.LinearRegression()  
  
# Train the model using the training sets and check score  
  
linear.fit(x_train, y_train)  
linear.score(x_train, y_train)  
  
#Equation coefficient and Intercept  
  
print('Coefficient: \n', linear.coef_)  
print('Intercept: \n', linear.intercept_)  
  
#Predict Output  
  
predicted= linear.predict(x_test)
```

Implémentation

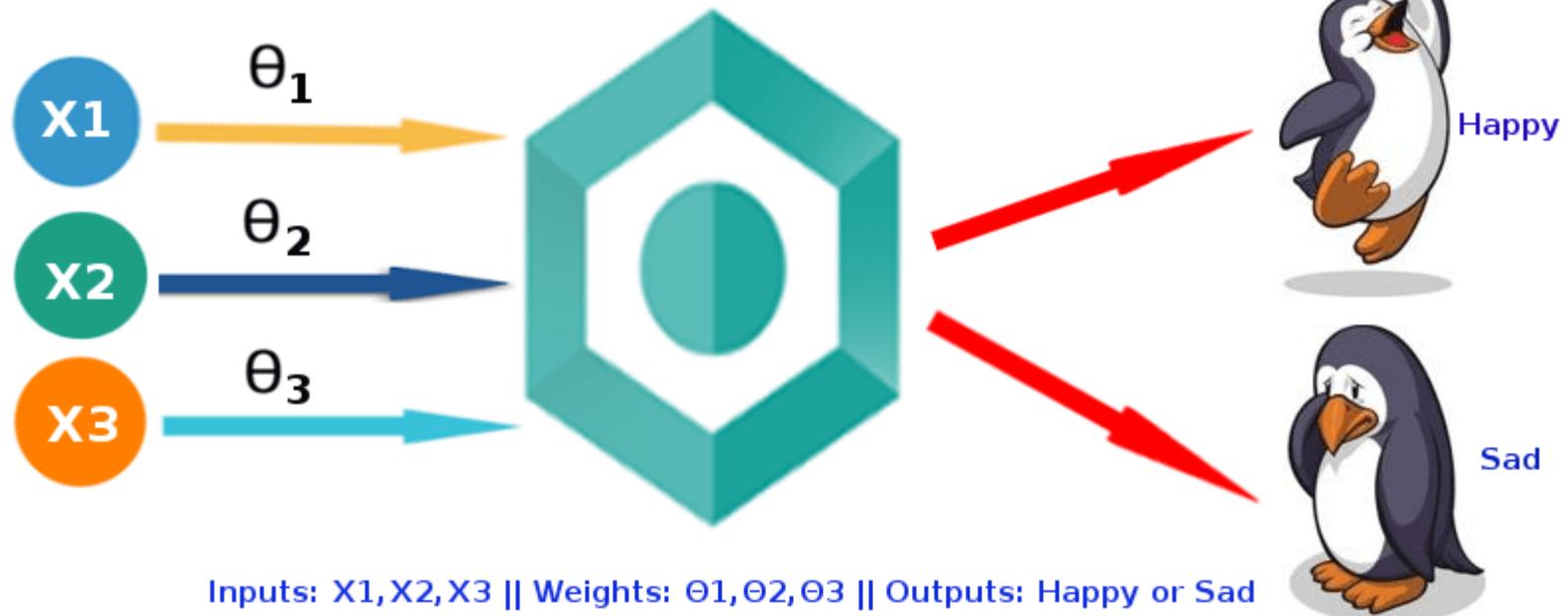
R Code

```
#Load Train and Test datasets  
  
#Identify feature and response variable(s) and values must be numeric and numpy arrays  
  
x_train <- input_variables_values_training_datasets  
y_train <- target_variables_values_training_datasets  
x_test <- input_variables_values_test_datasets  
  
x <- cbind(x_train,y_train)  
  
# Train the model using the training sets and check score  
  
linear <- lm(y_train ~ ., data = x)  
summary(linear)  
  
#Predict Output  
  
predicted= predict(linear,x_test)
```

Tuto



Logistic Regression Model

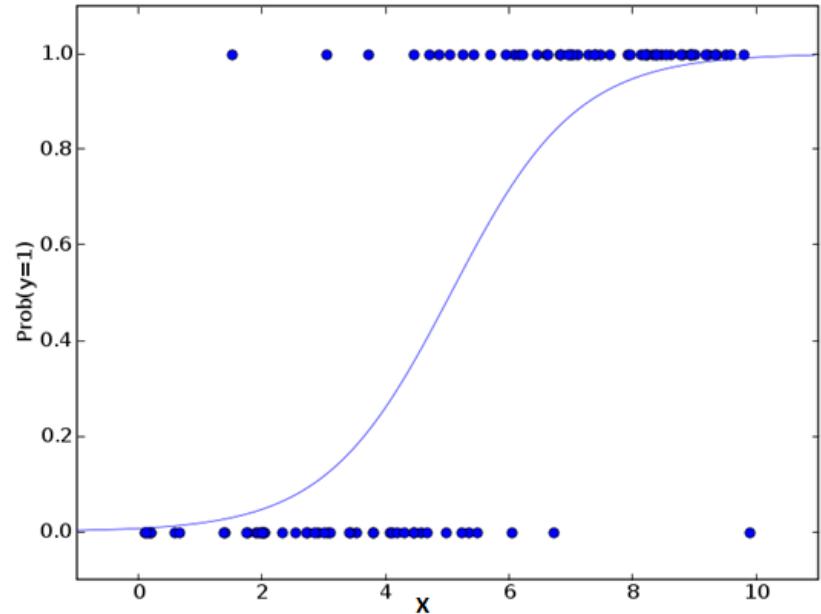


logistic regression

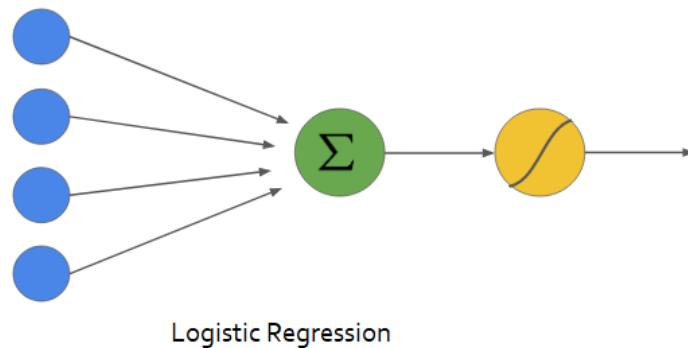
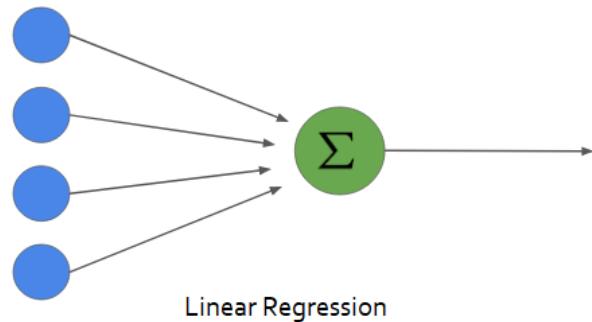
- In a lot of ways, linear regression and logistic regression are similar. But, the biggest difference lies in what they are used for.
- Linear regression algorithms are used to **predict/forecast** values but logistic regression is used for **classification tasks**.

Logistic Regression

- It is a classification not a regression algorithm.
- It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s).



LiR vs LoR



Logistic Regression

- Logistic Regression is used when the **dependent** variable(target) is categorical.
- For example,
 - To predict whether an email is spam (1) or (0)
 - Whether the tumor is malignant (1) or not (0)
 - whether a website is fraudulent (1) or not (0)

Logistic regression

- Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity.

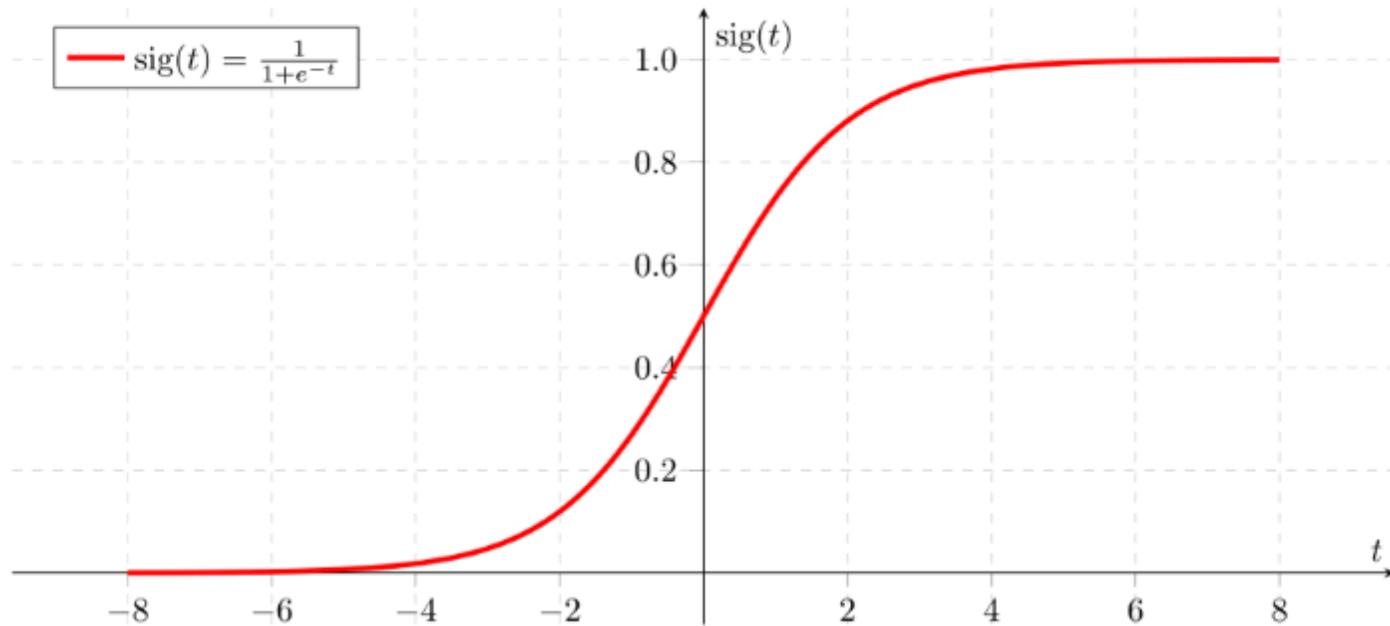
Logistic regression

- The predicted value can be anywhere between **negative infinity** to **positive infinity**. We need the output of the algorithm to be class variable, i.e 0-no, 1-yes.
- Therefore, we are squashing the output of the linear equation into a range of [0,1]. To squash the predicted value between 0 and 1, we use the sigmoid function.

Model

- Output = 0 or 1
 - Hypothesis => $Z = WX + B$
 - $h\Theta(x) = \text{sigmoid}(Z)$
-
- If 'Z' goes to infinity, Y(predicted) will become 1 and if 'Z' goes to negative infinity, Y(predicted) will become 0.

Sigmoid Activation Function



Sigmoid function

$$z = \theta_0 + \theta_1 \cdot x_1 + \theta \cdot x_2 + \dots \quad g(x) = \frac{1}{1 + e^{-x}}$$

Linear Equation and Sigmoid Function

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

Squashed output-h

Mathematically this can be written as

$$h_{\theta}(x) = P(Y=1|X; \theta)$$

Probability that $Y=1$ given X which is parameterized by 'theta'.

$$P(Y=1|X; \theta) + P(Y=0|X; \theta) = 1$$

$$P(Y=0|X; \theta) = 1 - P(Y=1|X; \theta)$$

Types of Logistic Regression

- 1. **Binary Logistic Regression**
 - The categorical response has only two possible outcomes.
Example: Spam or Not
- 2. **Multinomial Logistic Regression**
 - Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)
- 3. **Ordinal Logistic Regression**
 - Three or more categories with ordering. Example: Movie rating from 1 to 5

Cost Function

- Since we are trying to predict class values, we cannot use the same cost function used in linear regression algorithm. Therefore, we use a logarithmic loss function to calculate the cost for misclassifying.

Cost Function

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

- The above cost function can be rewritten as below since calculating gradients from the above equation is difficult.

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

Gradients

$$J = \frac{-1}{m} \cdot \left[\sum_{i=1}^m y_i \cdot \log h_i + (1 - y_i) \cdot \log 1 - h_i \right]$$

$$\frac{\partial J}{\partial \theta_n} = \frac{-1}{m} \cdot \left[\sum_{i=1}^m \frac{y_i}{h_i} \cdot h_i^2 \cdot x_n \cdot \frac{1 - h_i}{h_i} + \frac{1 - y_i}{1 - h_i} \cdot -h_i^2 \cdot x_n \cdot \frac{1 - h_i}{h_i} \right]$$

$$\frac{\partial J}{\partial \theta_n} = \frac{-1}{m} \cdot \left[\sum_{i=1}^m x_n \cdot (1 - h_i) \cdot y_i - x_n \cdot h_i \cdot (1 - y_i) \right]$$

$$\frac{\partial J}{\partial \theta_n} = \frac{1}{m} \cdot x_i \cdot \left[\sum_{i=1}^m h_i - y_i \right]$$

Implementation

Python code

```
#Import Library  
from sklearn.linear_model import LogisticRegression  
  
#Assumed you have, X (predictor) and Y (target) for training data set and x_test(predictor) of test_d  
ataset  
  
# Create logistic regression object  
model = LogisticRegression()  
  
# Train the model using the training sets and check score  
model.fit(X, y)  
  
model.score(X, y)  
  
#Equation coefficient and Intercept  
print('Coefficient: \n', model.coef_)  
print('Intercept: \n', model.intercept_)  
  
#Predict Output  
predicted= model.predict(x_test)
```

Implementation

R Code

```
x <- cbind(x_train,y_train)

# Train the model using the training sets and check score

logistic <- glm(y_train ~ ., data = x,family='binomial')

summary(logistic)

#Predict Output

predicted= predict(logistic,x_test)
```

Tuto

The screenshot shows a web browser window with the URL archive.ics.uci.edu/ml/datasets/Bank+Marketing. The page displays basic information about the dataset, including its characteristics and associated tasks.

Attribute Characteristics:	Real	Number of Attributes:	17	Date Donated	2012-02-14
Associated Tasks:	Classification	Missing Values?	N/A	Number of Web Hits:	860529

Source:

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. *Decision Support Systems*, Elsevier, 62:22-31, June 2014

Data Set Information:

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required. In order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

There are four datasets:

- 1) bank-additional-full csv with all examples (41188) and 20 inputs, ordered by date (from May 2008 to November 2010), very close to the data analyzed in [Moro et al., 2014]
 - 2) bank-additional.csv with 10% of the examples (4119), randomly selected from 1), and 20 inputs.
 - 3) bank-full.csv with all examples and 17 inputs, ordered by date (older version of this dataset with less inputs)
 - 4) bank.csv with 10% of the examples and 17 inputs, randomly selected from 3 (older version of this dataset with less inputs).
- The smallest datasets are provided to test more computationally demanding machine learning algorithms (e.g., SVM).

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

Attribute Information:

Input variables:

```
# bank client data:  
1 - age (numeric)  
2 - job : type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')  
3 - marital: marital status (categorical: 'divorced', 'married', 'single', 'unknown'. note: 'divorced' means divorced or widowed)  
4 - education (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')  
5 - default: has credit in default? (categorical: 'no', 'yes', 'unknown')
```

<http://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

Attribute Information:

- Input variables:

bank client data:

1 - age (numeric)

2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')

3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)

4 - education (categorical:

'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')

5 - default: has credit in default? (categorical: 'no','yes','unknown')

6 - housing: has housing loan? (categorical: 'no','yes','unknown')

7 - loan: has personal loan? (categorical: 'no','yes','unknown')

related with the last contact of the current campaign:

8 - contact: contact communication type (categorical: 'cellular','telephone')

9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')

10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')

11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Attribute Information:

- 12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
social and economic context attributes
16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

- Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes','no')



172

Régression Logistique

Régression linéaire

173

- Cette méthode se focalise sur les cas où les valeurs d'une variable à prédire sont continues
- Les valeurs à prédire peuvent être représentées par une fonction linéaire, donc une droite

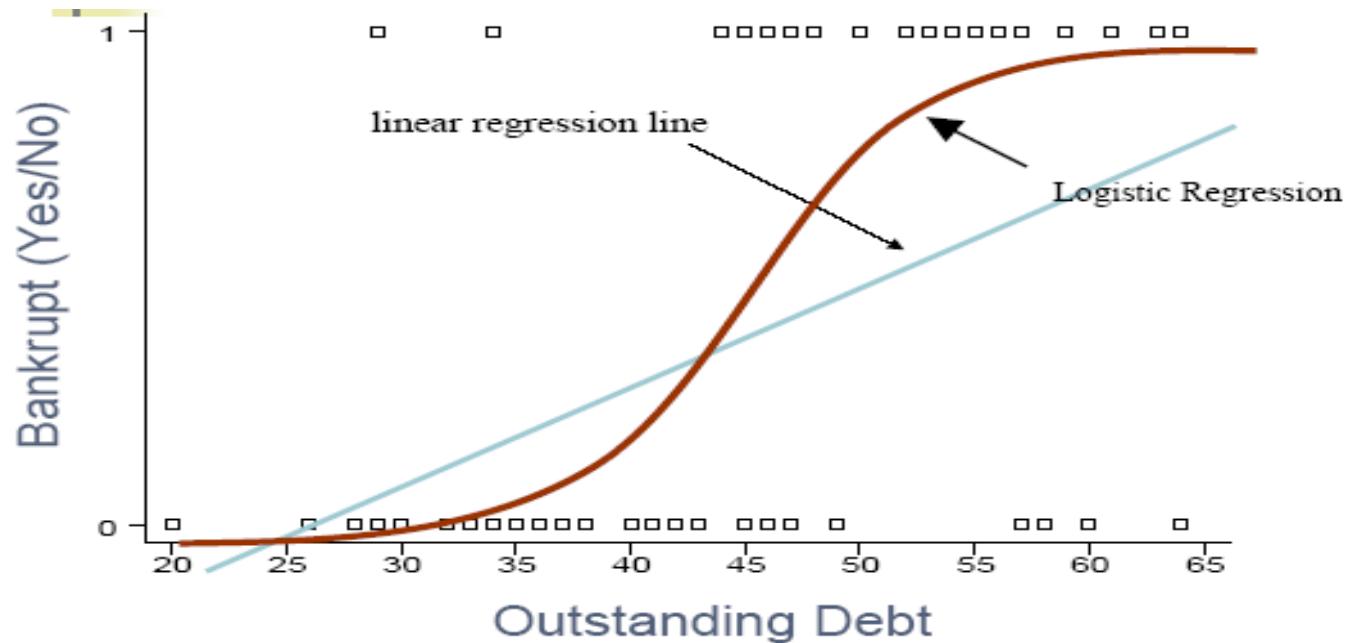
Régression logistique

174

- Cette méthode se focalise sur les situations où les valeurs d'une variable à prédire sont binaires (0 ou 1)
 - Exemple: Une variable booléenne
- Au lieu de prédire la valeur d'une variable, on prédit la probabilité de la variable à être égale à 0 et 1.
- Les probabilités décrivent une sigmoïde (courbe en forme de S) entre 0 et 1

Prédiction de banqueroute

175



Etude de cas: Compagnie de téléphone

- Adoption d'un nouveau service téléphonique (boîte vocale, accès à Internet...) suivant l'éducation, la stabilité de résidence et le salaire
- 10524 personnes ont répondu à un questionnaire sur ce thème réalisé par une compagnie de téléphone
- Comment prédire l'adoption d'un nouveau service téléphonique en fonction de l'éducation, de la stabilité de résidence et du salaire d'une personne?

Réponses au questionnaire

177

High School or below		Some College or above		
	No Change in Residence during Last five years	Change in Residence during Last five years	No change in Residence during Last five years	Change in Residence during Last five years
Low Income	$153/2160 = 0.071$	$226/1137 = 0.199$	$61/886 = 0.069$	$233/1091 = 0.214$
High Income	$147/1363 = 0.108$	$139/ 547 = 0.254$	$287/1925 = 0.149$	$382/1415 = 0.270$

Il y a 2160 personnes qui ont répondu au questionnaire qui ont un niveau d'étude inférieur ou égale au lycée, un bas salaire et qui n'ont pas changé de résidence depuis 5 ans.

Il y a 153 personnes (sur ces 2160 personnes) qui ont adopté un nouveau service téléphonique

Probabilité globale d'adoption d'un nouveau service téléphone pour

Le modèle de régression logistique

178

- Prédire la probabilité de la valeur de Y à partir de variables indépendantes x₁,..., x_k
 - Y = 1: Choisir une option
 - Y = 0: Ne pas choisir une option
- $$\text{Probability}(Y=1|x_1, x_2, \dots, x_k) = \frac{\exp(\beta_0 + \beta_1 * x_1 + \dots + \beta_k * x_k)}{1 + \exp(\beta_0 + \beta_1 * x_1 + \dots + \beta_k * x_k)}$$

Les β_i sont des constantes inconnues à déterminer. Ils sont calculés/estimés par des programmes.

Interprétation des coefficients

179

- Si $\beta_i = 0$, alors le facteur β_i n'a aucun effet sur la chance de succès
- Si $\beta_i > 0$, le facteur β_i augmente la chance de succès
- Si $\beta_i < 0$, le facteur β_i décroît la chance de succès

Poser le problème (1)

180

- On doit calculer les probabilités d'adopter un nouveau service téléphonique en fonction de l'éducation, de la stabilité de résidence et le salaire d'une personne
- Soit Y la variable représentant l'adoption d'un nouveau service téléphonique
 - $Y = 1$ si un nouveau service est adopté, et $Y = 0$ sinon

Poser le problème (2)

- On a trois variables x_1 pour l'éducation, x_2 pour la stabilité de résidence et x_3 pour le salaire
- $X_1 = 1$ pour un niveau d'étude supérieur ou égal à l'université, 0 sinon
- $X_2 = 1$ pour un changement de résidence dans les 5 dernières années, 0 sinon
- $X_3 = 1$ pour un salaire élevé, 0 sinon
- Modèle $Prob(Y = 1|x_1, x_2, x_3) = \frac{\exp(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3)}{1 + \exp(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3)}.$

Résumé des données

182

x_1	x_2	x_3	# in sample	#adopters	# Non-adopters	Fraction adopters
0	0	0	2160	153	2007	.071
0	0	1	1363	147	1216	.108
0	1	0	1137	226	911	.199
0	1	1	547	139	408	.254
1	0	0	886	61	825	.069
1	1	0	1091	233	858	.214
1	0	1	1925	287	1638	.149
1	1	1	1415	382	1033	.270
			10524	1628	8896	1.000

Calcul de β_0 , β_1 , β_2 et β_3

183

Variable	Coeff.	Std. Error	p-Value	Odds	95% Conf. Intvl. for odds	
Constant	-2.500	0.058	0.000	0.082	0.071	0.095
x_1	0.161	0.058	0.006	1.175	1.048	1.316
x_2	0.992	0.056	0.000	2.698	2.416	3.013
x_3	0.444	0.058	0.000	1.560	1.393	1.746

$$\beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3$$

Modèle

$$Prob(Y = 1|x_1, x_2, x_3) = \frac{\exp(-2.500 + 0.161 * x_1 + 0.992 * x_2 + 0.444 * x_3)}{1 + \exp(-2.500 + 0.161 * x_1 + 0.992 * x_2 + 0.444 * x_3)}.$$

x_1	x_2	x_3	# in sample	# adopters	Estimated (# adopters)	Fraction Adopters	Estimated $Prob(Y = l x_1, x_2, x_3)$
0	0	0	2160	153	164	0.071	0.076
0	0	1	1363	147	155	0.108	0.113
0	1	0	1137	226	206	0.199	0.181
0	1	1	547	139	140	0.254	0.257
1	0	0	886	61	78	0.069	0.088
1	1	0	1091	233	225	0.214	0.206
1	0	1	1925	287	252	0.149	0.131
1	1	1	1415	382	408	0.270	0.289

2160 x 0.076 = 164

Estimation du nombre de personnes qui peuvent adopter un nouveau service téléphonique

Nouvelles données

185

Calculs
d'erreurs

598 nouvelle personnes sont sondées

x_1	x_2	x_3	# in validation sample	# adopters in validation sample	Estimated (# adopters)	Error (Estimate - Actual)	Absolute Value of Error
0	0	0	29	3	2.200	-0.800	0.800
0	0	1	23	7	2.610	-4.390	4.390
0	1	0	112	25	20.302	-4.698	4.698
0	1	1	143	27	36.705	9.705	9.705
1	0	0	27	2	2.374	0.374	0.374
1	1	0	54	12	11.145	-0.855	0.855
1	0	1	125	13	16.338	3.338	3.338
1	1	1	85	30	24.528	-5.472	5.472
Totals			598	119	116.202		

$$85 \times 0.289 = 24.5$$

Estimation du nombre de personnes qui peuvent adopter un produit/éducatif

Calcul d'erreurs

186

- Total erreur: -2.8 (or $2.8 / 119 = 2.3\%$)
- La moyenne d'erreur absolue (sommes des erreurs absolues / 119): 24.9%

Tableau de contingence

187

Matrice de contingence [Kohavi, Provost, 1998]:

Prédit \ Observé	Adopteur	Non Adopteur	Total
Adopteur	103 (TP)	13 (FP)	116
Non adopteur	16 (FN)	466 (TN)	482
Total	119	479	598

TP: true positive, FP: false positive, FN: false negative, TN: true negative

Calcul de taux

□ Vrais positifs:

Cas positifs correctement prédicts

- $103 / 119 = 86.5 \%$
- Fausses positives:
 - Cas incorrectement prédicts positif
 - $13 / 479 = 2.7 \%$
- Exactitude:
 - Nombre total de prédictions correctes
 - $(103 + 466) / 598 = 95.15\%$
- Précision:
 - Proportion des prédictions positives correctes
 - $103 / (103 + 13) = 88.8 \%$
- Erreurs:
 - Proportion des prédictions incorrectes
 - $(13+16) / 598 = 4.85 \%$

Quel est le meilleur modèle?

Vrai: Offrir un crédit

Faux: Ne pas offrir un crédit

Modèle 1:

TP 600	FP 75
FN 25	TN 300

Modèle 2:

TP 600	FP 25
FN 75	TN 300

Taux d'erreur pour les 2 modèles: 10%

Le meilleur modèle est Modèle 2 car ce modèle a moins de FP

Conclusion

190

- Méthode facile à comprendre
- Méthode efficace
- Les prédictions sont faciles à réaliser
- Le bruit peut avoir un effet significatif sur la méthode
- Besoin de plusieurs mesures pour évaluer le modèle



Classification

Apprentissage supervisé

Découverte de règles ou formules (patterns) pour ranger les données dans des classes prédéfinies

- représentant un groupe d'individus homogènes
- permettant de classer les nouveaux arrivants
- Processus en deux étapes
 - construction d'un modèle sur les données dont la classe est connue (training data set)
 - utilisation pour classification des nouveaux arrivants

Applications

□ Marketing

- comprendre les critères prépondérants dans l'achat d'un produit
- segmentation automatique des clients pour le marketing direct

□ Maintenance

- aide et guidage d'un client suite à défauts constatés

□ Assurance

- analyse de risques

□ Isolation de populations à risques

- médecine



Les K plus proches voisins (KNN)

Principe de fonctionnement

- Le principe de cet algorithme de classification est très simple.
On lui fournit:
 - un ensemble de données d'apprentissage D
 - une fonction de distance d
 - et un entier k
- Pour tout nouveau point de test x , pour lequel il doit prendre une décision, l'algorithme recherche dans D les k points les plus proches de x au sens de la distance d , et attribue x à la classe qui est la plus fréquente parmi ces k voisins.

Généralités

- la méthode des k plus proches voisins est une méthode de **d'apprentissage supervisé**.
- dédiée à la **classification**.
- En abrégé k-NN ou KNN, de l'anglais *k-nearest neighbor*.
- L'algorithme KNN figure **parmi** les plus simples algorithmes **d'apprentissage artificiel**.
- L'objectif de l'algorithme est de classé les exemples **non étiquetés** sur **la base de leur similarité** avec **les exemples de la base d'apprentissage** .

Domaine d'activité

□ L'algorithme kNN est utilisée dans de nombreux domaines :

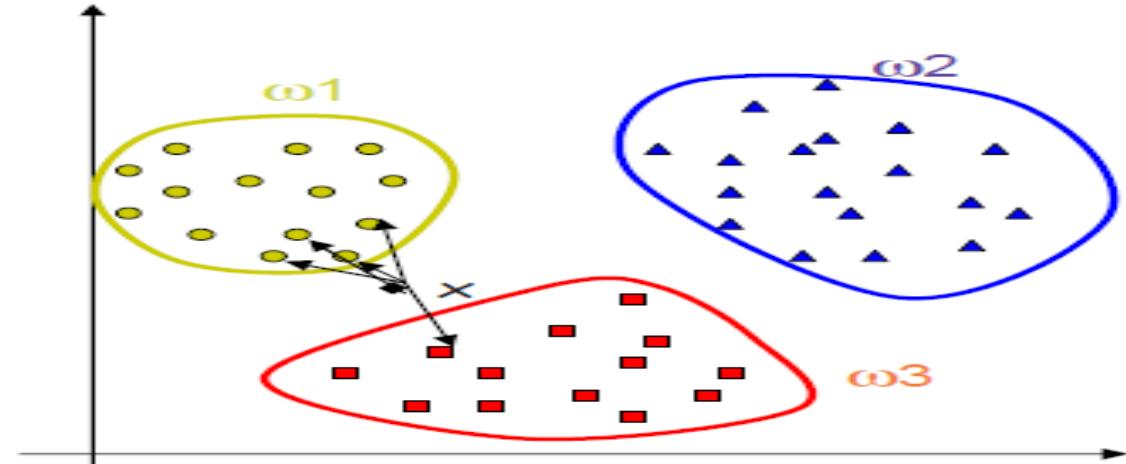
- La reconnaissance de formes.
- La recherche de nouveaux biomarqueurs pour le diagnostic.
- Algorithmes de compression.
- Analyse d'image satellite
- Marketing ciblé

Principe de fonctionnement

- Le principe de cet algorithme de classification est très simple.
On lui fournit:
 - un ensemble de données d'apprentissage D
 - une fonction de distance d
 - et un entier k
- Pour tout nouveau point de test x , pour lequel il doit prendre une décision, l'algorithme recherche dans D les k points les plus proches de x au sens de la distance d , et attribue x à la classe qui est la plus fréquente parmi ces k voisins.

Exemple

- Dans l'exemple suivant, on a 3 classes et le but est de trouver la valeur de la classe de l'exemple inconnu x .
 - On prend la distance Euclidienne et $k=5$ voisins
 - Des 5 plus proches voisins, 4 appartiennent à ω_1 et 1 appartient à ω_2 , donc la classe majoritaire



Comment choisir la valeur de K ?

□ $K=1$: frontières des classes très complexes

- très sensible aux fluctuations des données (variance élevée).
- risque de sur-ajustement.
- résiste mal aux données bruitées.

□ $K=n$: frontière rigide

- moins sensible au bruit
- plus la valeur de k est grande plus la résultat d'affectation est bien réalisée

Mesures de distance

- Mesures souvent utilisées pour la distance $dist(x_i, x_j)$
- la distance Euclidienne: qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

- la distance de Manhattan: qui calcule la somme des valeur absolue des différences entre les coordonnées de deux points .

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- la distance de Minkowski e générale.

$$d(x, y) = \sqrt[q]{\sum_{i=1}^n |x_i - y_i|^q}$$

Avantages

- Apprentissage rapide
- Méthode facile à comprendre
- Adapté aux domaines où chaque classe est représentée par plusieurs prototypes et où les frontières sont irrégulières (ex. Reconnaissance de chiffre manuscrits ou d'images satellites)

Inconvénients

- prédition lente car il faut revoir tous les exemples à chaque fois.
- méthode gourmande en place mémoire
- sensible aux attributs non pertinents et corrélés
- particulièrement vulnérable au fléau de la dimensionnalité



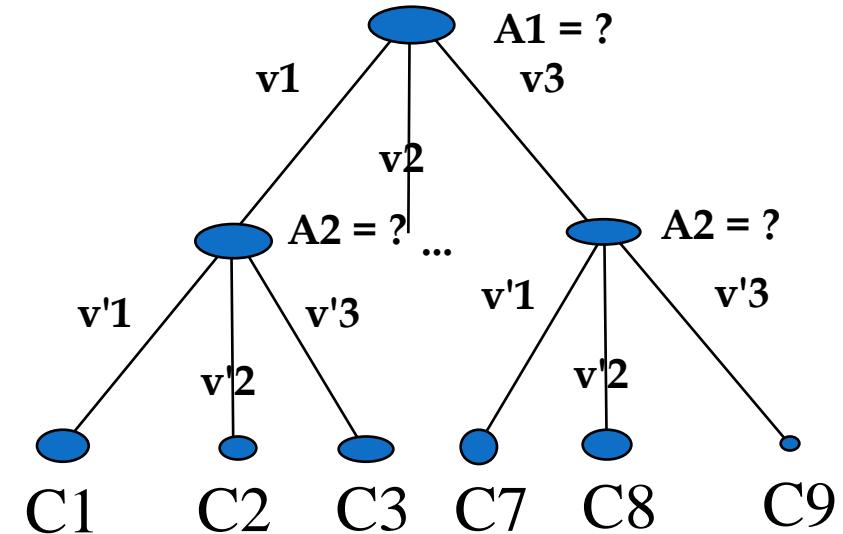
Les arbres de décision

Définition

- Arbre permettant de classer des enregistrements par division hiérarchiques en sous-classes
 - un nœud représente une classe de plus en plus fine depuis la racine
 - un arc représente un prédicat de partitionnement de la classe source
- Un attribut sert d'étiquette de classe (attribut cible à prédire), les autres permettant de partitionner

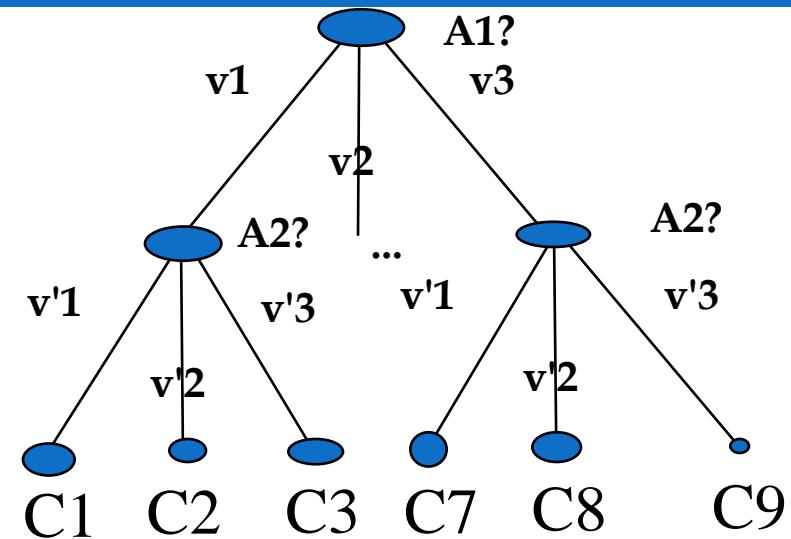
Génération de l'arbre

- Objectif:
 - obtenir des classes homogènes
 - couvrir au mieux les données
- Comment choisir les attributs (A_i) ?
- Comment isoler les valeurs discriminantes (v_i) ?



Arbre = ensemble de règles

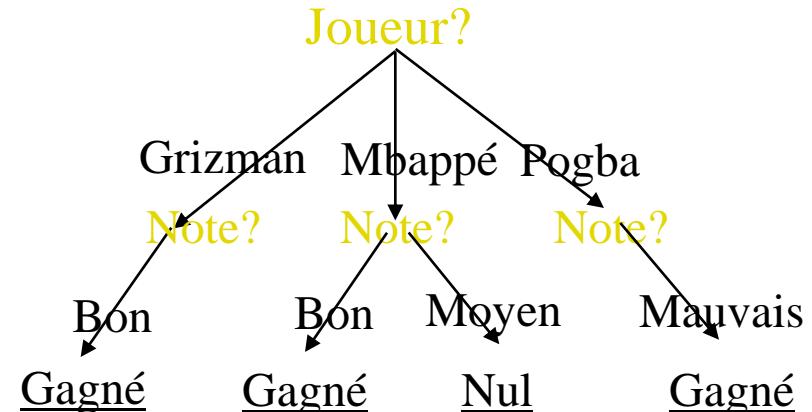
- $(A1=v1) \& (A2=v'1) \rightarrow C1$
- $(A1=v1) \& (A2=v'2) \rightarrow C2$
- $(A1=v1) \& (A2=v'3) \rightarrow C3$
- ...
- $(A1=v3) \& (A2=v'1) \rightarrow C7$
- $(A1=v3) \& (A2=v'2) \rightarrow C8$
- $(A1=v3) \& (A2=v'3) \rightarrow C9$



Exemple codant une table

Attributs ou variables

Joueur	Note	Résultat
Mbappé	Bon	Gagné
Mbappé	Moyen	Nul
Grizman	Bon	Gagné
Grizman	Bon	Gagné
Pogba	Mauvais	Gagné



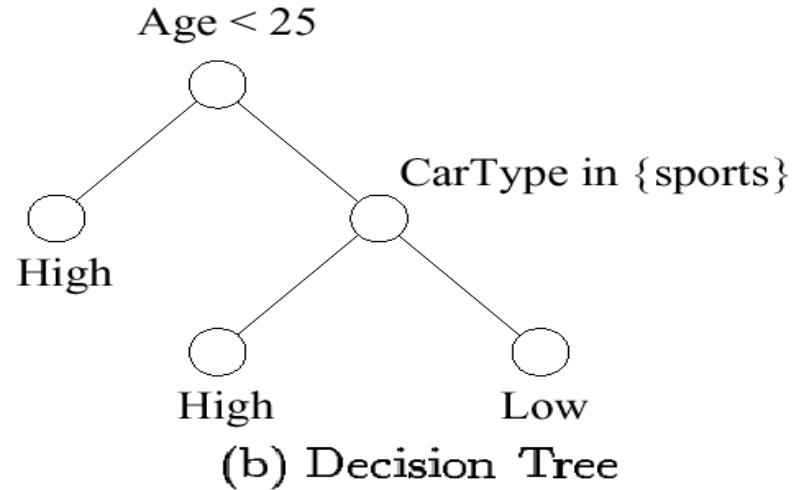
Autre Exemple

<i>rid</i>	Age	Car Type	Risk
0	23	family	High
1	17	sports	High
2	43	sports	High
3	68	family	Low
4	32	truck	Low
5	20	family	High

(a) Training Set



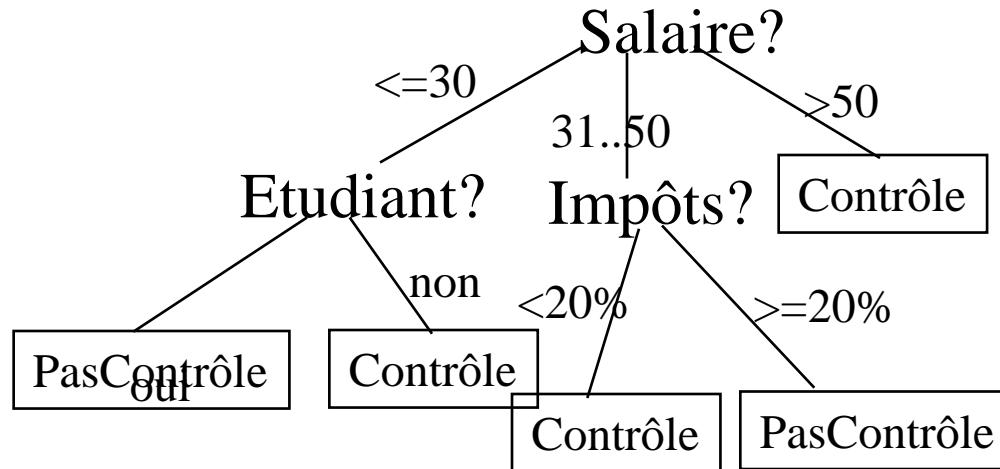
Classes cibles



(b) Decision Tree

Autre Exemple

□ Faut-il vous envoyer un contrôleur fiscal ?



Procédure de construction (1)

- recherche à chaque niveau de l'attribut le plus discriminant
- Partition (nœud P)
 - si (tous les éléments de P sont dans la même classe) alors retour;
 - pour chaque attribut A faire
 - évaluer la qualité du partitionnement sur A;
 - utiliser le meilleur partitionnement pour diviser P en P₁, P₂, ...P_n
 - pour i = 1 à n faire Partition(P_i);

Procédure de Construction (2)

□ Processus récursif

- L'arbre commence à un nœud représentant toutes les données
- Si les objets sont de la même classe, alors le nœud devient une feuille étiqueté par le nom de la classe.
- Sinon, sélectionner les attributs qui séparent le mieux les objets en classes homogènes => Fonction de qualité
- La récursion s'arrête quand:
 - Les objets sont assignés à une classe homogène
 - Il n'y a plus d'attributs pour diviser,
 - Il n'y a pas d'objet avec la valeur d'attribut



Choix de l'attribut de division

Différentes mesures introduites

- il s'agit d'ordonner le désordre
- des indicateurs basés sur la théorie de l'information
- Choix des meilleurs attributs et valeurs
 - les meilleurs tests
- Possibilité de retour arrière
 - élaguer les arbres résultants (classes inutiles)
 - revoir certains partitionnements (zoom, réduire)

Mesure de qualité

□ La mesure est appelé fonction de qualité

- Goodness Function en anglais
- Varie selon l'algorithme :
 - Gain d'information (ID3/C4.5)
 - Suppose des attributs nominaux (discrets)
 - Peut-être étendu à des attributs continus
 - Gini Index
 - Suppose des attributs continus
 - Suppose plusieurs valeurs de division pour chaque attribut
 - Peut-être étendu pour des attributs nominaux

Mesure d'impureté (variable nominale)

- Mesure des mélanges de classes d'un nœud N
 - $i(N) = \sum_i \sum_j \{ p_i * p_j \}$ avec $i \neq j$
 - p_i est la proportion d'individus de la classe i dans N .
- La réduction d'impureté de chaque division du nœud N par la variable x_j s'exprime par:
 - $\Delta N = i(N) - \sum_j p_j * i(N_j)$
 - p_j est la proportion d'individus du nœud dans le fils j
- Sur l'ensemble des n variables, la division du nœud t est effectuée à l'aide de la variable qui assure la réduction maximale de l'impureté (\sum minimum)

Mesure d'entropie

- Minimisation du désordre restant
 - π_i = fréquence relative de la classe i dans le nœud N
(% d'éléments de la classe i dans N)
- Mesure d'entropie d'un segment s
 - $E(N) = -\sum \pi_i \log_2(\pi_i)$
- Minimiser son évolution globale [Quinlan]
 - $\Delta N = E(N) - \sum_i P_i * E(N_i)$

Indices de Gini et Twoing

□ Indice de GINI

Si un ensemble de données T contient des éléments de N classes

$$\text{gini}(T) = 1 - \sum_i p_i^2 \text{ où } p_i \text{ est la fréquence relative de la classe } i \text{ dans } T$$

□ Indice de Twoing

$$G(t_g, t_d) = [((n_g/n)(n_d/n))/4][\sum_{i=1}^m |(n_{ig}/n_g) - (n_{id}/n_g)|]^2$$

- t_g : Sommet gauche issu de t.
- t_d : Sommet droit issu de t
- n_d (resp (n_g)) = card $\{t_d\}$ (resp card $\{t_g\}$).
- N : La taille de l'échantillon d'apprentissage.
- M : Le nombre de classe.
- n_{id} : (resp (n_{ig})) : l'effectif de la classe c_i dans t_d (resp (t_g)).

Exemple: Partitions de boules (1)

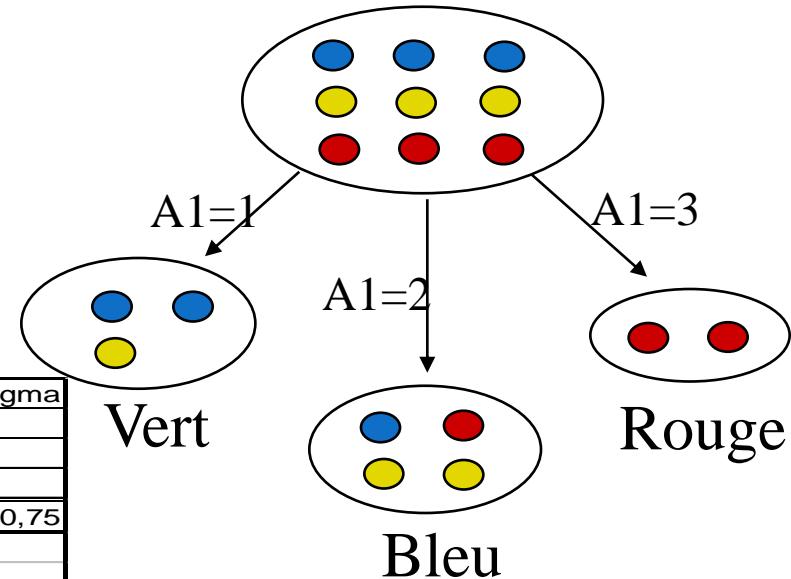
- Partition selon A1 (densité)

- Indice d'impureté :

- $i(N) = \sum_i^k \sum_j^k \{ p_i * p_j \}$ avec $i \neq j$

- P_i est la proportion d'individus de la classe i dans N .

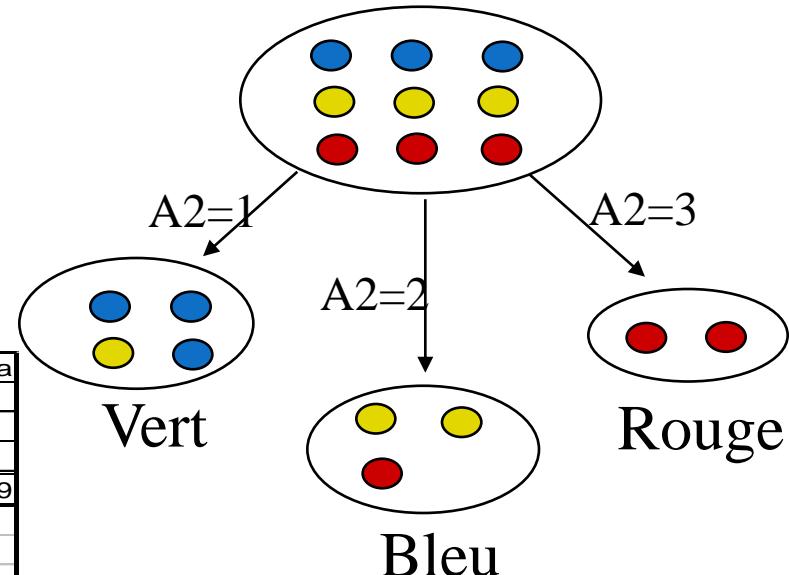
Proportion	G1	G2	G3	Sigma
Vert	0,67	0,25	0,00	
Bleu	0,33	0,50	0,00	
Rouge	0,00	0,25	1,00	
Entropie	0,92	1,00	0,00	0,75
N2 log2(N2)	-0,39	-0,50	0,00	
N3 log2(N3)	-0,53	-0,50	0,00	
N4 log2(N4)	0,00	0,00	0,00	
Impureté	0,44444444	0,625	0	0,43



Exemple: Partitions de boules (2)

- Partition selon A2
 - Position et 4 au plus par partition

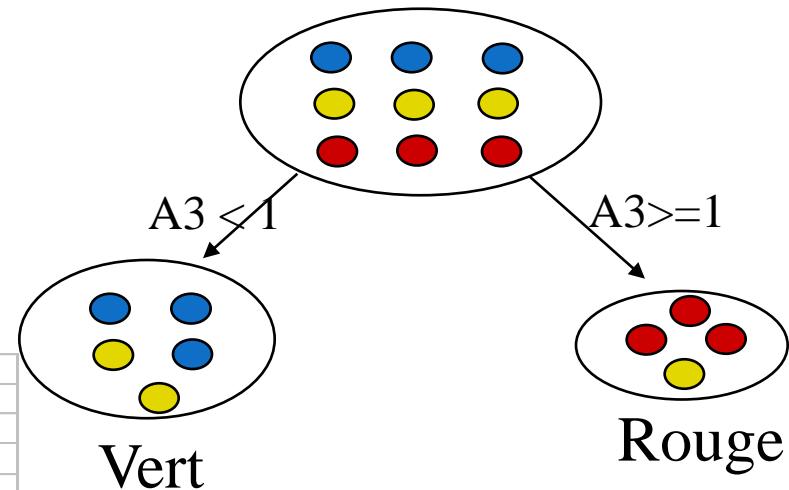
Proportion	C1	C2	C3	Sigma
Vert	0,75	0,00	0,00	
Bleu	0,25	0,67	0,00	
Rouge	0,00	0,33	1,00	
Entropie	0,81	0,39	0,00	0,49
N2 log2(N2)	-0,31	0,00	0,00	
N3 log2(N3)	-0,50	-0,39	0,00	
N4 log2(N4)	0,00	0,00	0,00	
Impureté	0,375	0,444444444	0	0,31



Exemple: Partitions de boules (3)

- Partition selon A3
 - Poids

Proportion	C1	C2	Sigma
Vert	0,60	0,00	
Bleu	0,40	0,25	
Rouge	0,00	0,75	
Entropie	0,97	0,50	0,76
N2 log2(N2)	-0,44	0,00	
N3 log2(N3)	-0,53	-0,50	
N4 log2(N4)	0,00	0,00	
Impureté	0,48	0,375	0,43



Exemple: Partitions de table (1)

Atr=?

Gain(Outlook) = 0.246

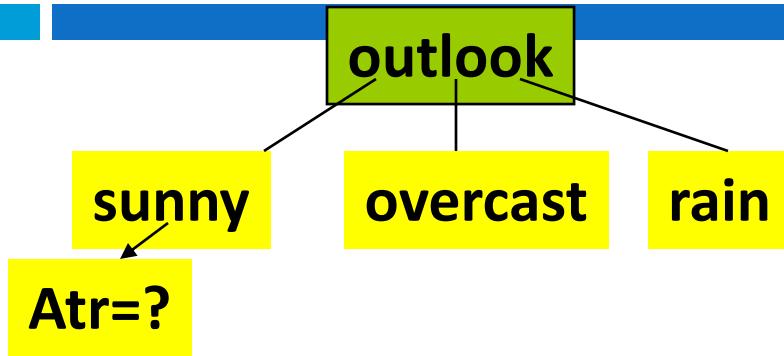
Gain(Temperature) = 0.029

Gain(Humidity) = 0.151

Gain(Windy) = 0.048

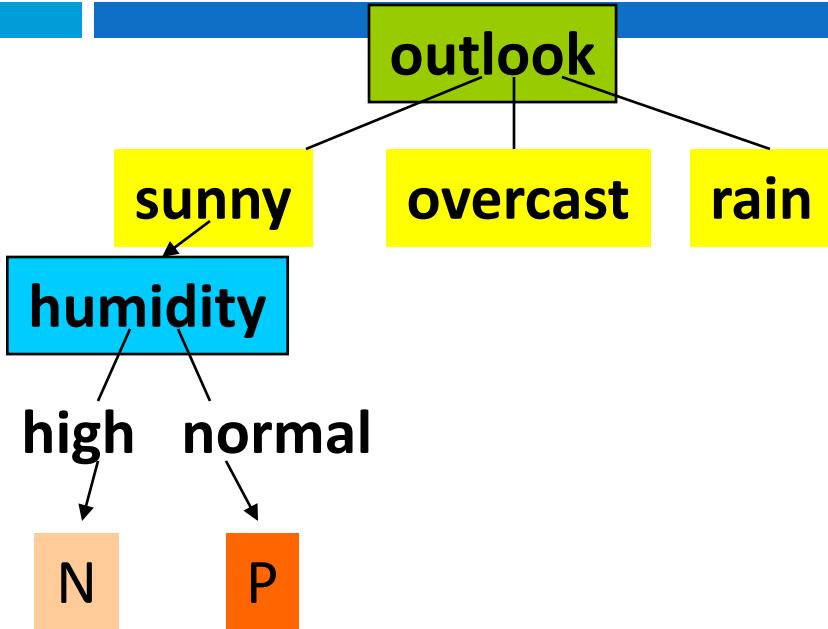
Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Exemple: Partitions de table (2)



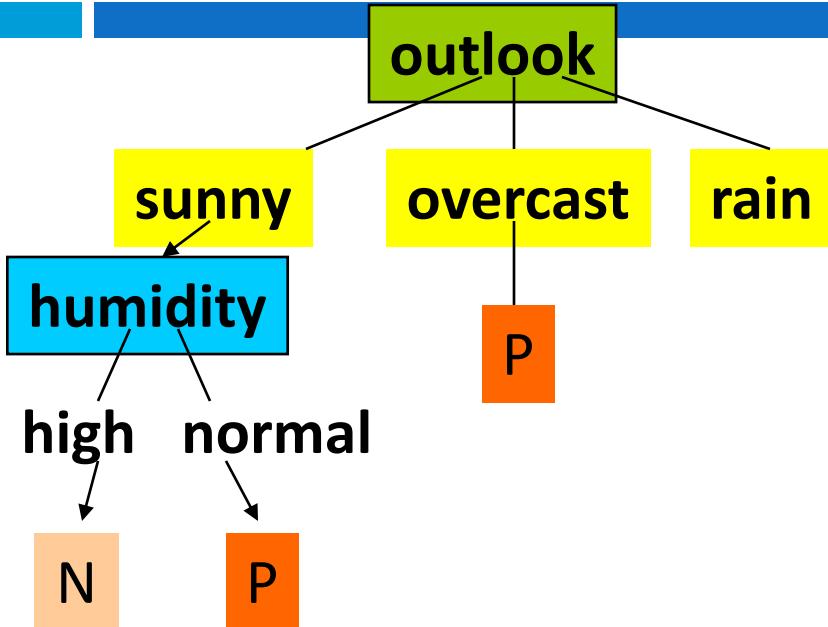
Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Exemple: Partitions de table (3)



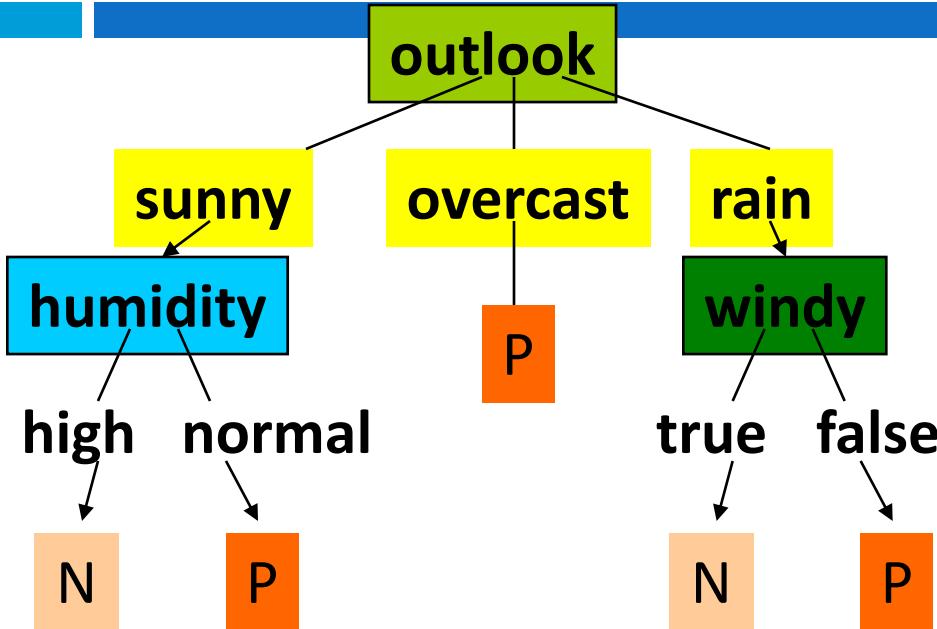
Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Exemple: Partitions de table (4)



Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Exemple: Partitions de table (5)



Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Types de tests

Binaire ou n-aire

- plus ou moins large et profond

Variable nominale

- un prédicat par valeur ou par liste de valeurs ?

Choix par niveau ou par classe

- mêmes tests pour chaque nœud interne d'un niveau

- arbres balancés ou non

Élimination de classes

- vides ou presque, peu représentatives

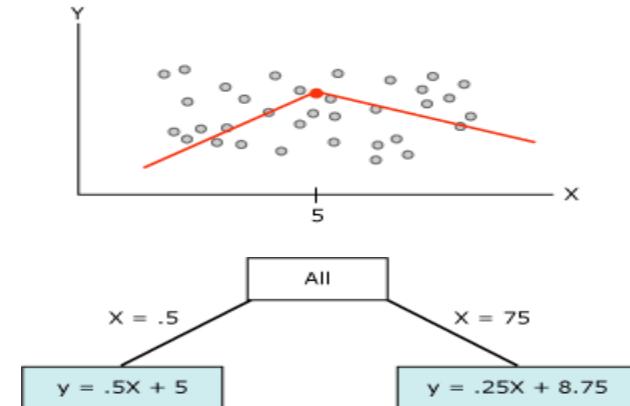
Problème des attributs continus

□ Certains attributs sont continus

- exemple : salaire
- découper en sous-ensembles ordonnés (e.g., déciles)
 - division en segments $[a_0, a_1[, [a_1, a_2[, \dots, [a_{n-1}, a_n]$
- utiliser moyenne, médiane, ... pour représenter
- minimiser la variance, une mesure de dispersion ...
- investiguer différents cas et retenir le meilleur
 - exemple : 2, 4, 8, etc. par découpe d'intervalles en 2 successivement

Attributs continus: Régression

- Partitionnement par droite de régression
- Chaque nœud est représenté par une formule de régression
- Séparation des données = point de non linéarité
- 1 ou plusieurs régresseurs
- Exemple :
 - $\text{salaire} = a + b * \text{tranche_age}$



Procédure d'élagage

- Les arbres trop touffus sont inutiles
- Intérêt d'un élagage récursif à partir des feuilles
 - S'appuie sur un modèle de coût d'utilité
- Possibilité de l'appliquer sur l'ensemble des données ou sur un sous-ensemble réservé à la validation

Exemple d'élagage

□ Exemple :

- arbres vus comme encodage de tuples
- partition utile si gain supérieur à un seuil
- coût d'un partitionnement
 - CP bits pour coder les prédictats de partition
 - Entropie_Après bits pour coder chaque tuple
- partitionnement à supprimer si :
 - $\text{Gain} = n * \text{Entropie_Après} + \text{CP} - n * \text{Entropie_Avant} < \text{seuil}$
- Ce test peut être appliquer lors de la création

Types d'arbres

	Description	Critère de coupe
Segmentation	binaire	<i>variance si variable continue, Indice de pureté si variable nominale</i>
ID3	<i>n</i> -aire	<i>Gain informationnel par entropie de Shannon</i>
C4.5	<i>n</i> -aire dérivé de ID3 + élagage, règles simplifiées, données manquantes,...	<i>Ratio du gain informationnel</i>
CART	binaire régression + élagage	<i>Indice de Gini si 2 valeurs en conclusion</i> <i>Indice de Twoing sinon</i>

Méthodes ID3 et C4.5

□ ID3

- Le pouvoir discriminatoire (ou gain informationnel) d 'une variable \leq une variation d '« entropie de Shannon » lors de la partition de S
- C4.5 (ID3++)
 - Support des variables continues
 - Introduit un facteur «Gain ratio » visant à pénaliser la prolifération des nœuds
- Critères d'arrêt :
 - Seuils de gain informationnel, d'effectif dans un nœud
 - Test statistique d'indépendance des variables (Ki2)

Méthode CART

□ Principes

- si problème à 2 classes, cherche la bi-partition minimisant l'indice d'impureté de Gini
- si problème à N classes, cherche celle maximisant le gain d'information donné par l'indice de Towing
- Critères d 'arrêt :
 - Seuil de gain informationnel
 - Seuil d 'effectif dans un nœud
 - Procédure d'élagage

Méthodes passant à l'échelle

- La plupart des algorithmes de base supposent que les données tiennent en mémoire
- La recherche en bases de données a proposé des méthodes permettant de traiter de grandes BD
- Principales méthodes:
 - ▣ SLIQ (EDBT'96 -- Mehta et al.'96)
 - ▣ SPRINT (VLDB96 -- J. Shafer et al.'96)
 - ▣ RainForest (VLDB98 -- J. Hekankho et al.'98)
 - ▣ PUBLIC (VLDB'98 -- R. Rastogi et al.'98)

Méthode SLIQ

□ SLIQ (EDBT'96 -- Mehta et al.'96)

- Supervised Learning In Quest
- Classificateurs CART et C4.5 :
 - Développe l'arbre en profondeur d'abord
 - Tri les données de manière répétée à chaque nœud
- SLIQ:
 - Remplace le tri répété par 1 seul tri par attribut
 - Utilise une nouvelle structure de données (**class-list**)
 - S'applique sur des attributs numériques ou nominaux
- Indicateur: maximiser $\text{gini}_{\text{split}}(T) = \sum_i [n_i/n] \text{gini}(T_i)$

Méthode SPRINT

■ SPRINT (VLDB96 -- J. Shafer et al.'96)

- Scalable PaRallelizable INduction of decision Tree
- SLIQ nécessite de garder la class-list en mémoire
- SPRINT
 - Ne nécessite pas de structure résidente en mémoire
 - Version parallèle passant à l'échelle

Comparaison avec SLIQ

- SLIQ ne divise pas les listes d'attributs lors du split

- Repère le nœud par un pointeur dans la class-list

- Avantages

- Pas de recopie des listes d'attributs lors du split
- Ré-allocation d'articles par déplacement de pointeur

- Désavantage

- La liste des références (class-list) de taille le nombre d'articles doit tenir en mémoire

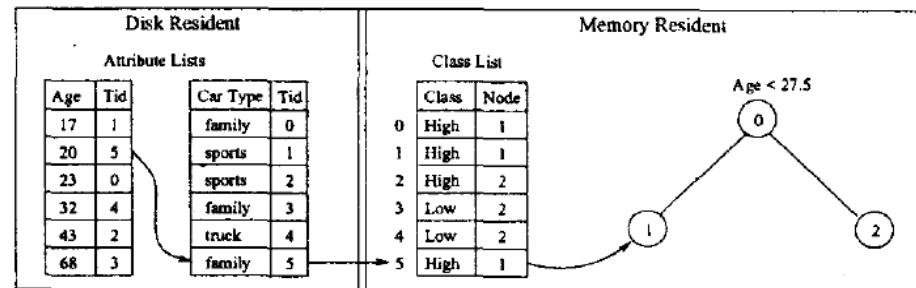


Figure 7: Attribute and Class lists in SLIQ

- SPRINT peut être facilement parallélisé
 - pas de structures partagées en mémoire

Bilan

- De nombreux algorithmes de construction d'arbre de décision
- SPRINT passe à l'échelle et traite des attributs nominaux ou continus
- Autres algorithmes proposés
 - Encore plus rapides ?



Le modèle de Bayes

La classification de Bayes

- Une méthode simple de classification supervisée
- Basée sur l'utilisation du Théorème de Bayes:

$$\Pr[H | E] = \frac{\Pr[E | H] \Pr[H]}{\Pr[E]}$$

Où H est l'hypothèse à tester, et E est l'évidence associée à l'hypothèse

- $\Pr(E | H)$ et $\Pr(H)$ sont facilement calculables
- $\Pr(H)$ est une probabilité a priori: la probabilité de H avant la présentation de l'évidence
- Il n'est pas nécessaire de calculer $\Pr(E)$

Note: $\Pr(E) = P(E)$

Méthode

- Une évidence E est donnée
- On calcule $P(H \mid E)$ pour toutes les valeurs de H
- Si $P(H = h \mid E)$ est maximum, alors on choisit:
 $H=h$

Étude de cas: Météo et match de foot

Données

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Calcul: $\Pr[\text{yes} | E]$

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Evidence E

$$\Pr[\text{yes} | E] = \Pr[\text{Outlook} = \text{Sunny} | \text{yes}] \times$$

Probability for class “yes”

$$\Pr[\text{Temperature} = \text{Cool} | \text{yes}] \times$$
$$\Pr[\text{Humidity} = \text{High} | \text{yes}] \times$$
$$\Pr[\text{Windy} = \text{True} | \text{yes}] \times \frac{\Pr[\text{yes}]}{\Pr[E]}$$

$$= \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{\Pr[E]}$$

$$= 0.0053 / \Pr[E]$$

Calcul: $\Pr[\text{No} | E]$

$$\boxed{\Pr[\text{No} | E] =}$$

$$= \Pr[\text{Outlook} = \text{Sunny} | \text{No}] \times \Pr[\text{Temperature} = \text{Cool} | \text{No}]$$

$$\times \Pr[\text{Humidity} = \text{High} | \text{No}] \times \Pr[\text{Windy} =$$

$$\text{True} | \text{No}] \times \Pr[\text{No}] / \Pr[E]$$

$$= 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 / \Pr[E]$$

$$= 0.0205 / \Pr[E]$$

Conclusion de l'exemple

On compare $\Pr[\text{Yes} | E]$ et $\Pr[\text{No} | E]$

- $\Pr[\text{Yes} | E] = 0.0053 / \Pr[E]$
- $\Pr[\text{No} | E] = 0.0205 / \Pr[E]$
- Donc le match ne va pas avoir lieu, car $0.0205 > 0.0053$

Cas d'un numérateur égal à 0

- Pour éviter d'avoir un numérateur égal à 0 et donc une probabilité égale à 0, dans le cas où le nombre d'attributs ayant une certaine valeur serait 0, on ajoute une constante k à chaque valeur au numérateur et au dénominateur
 - Un rapport n/d est transformé en $(n + kp)/(d+k)$, où p est une fraction du nombre total des valeurs possibles de l'attribut
 - K est entre 0 et 1
 - Estimation de Laplace: $k = 1$

Exemple: $\Pr[\text{No} \mid E]$

□ $K = 1$

□ $\Pr[\text{No} \mid E] =$

$$= \Pr[\text{Outlook} = \text{Sunny} \mid \text{No}] \times \Pr[\text{Temperature} = \text{Cool} \mid \text{No}] \times \Pr[\text{Humidity} = \text{High} \mid \text{No}] \times \Pr[\text{Windy} = \text{True} \mid \text{No}] \times \Pr[\text{No}] / \Pr[E]$$

$$= (3+1/3)/(5+1) \times (1+1/3)/(5+1) \times (4+1/2)/(5+1) \times (3+1/2)/(5+1) \times 5/14 / \Pr[E]$$

$$= 0.7539 / \Pr[E]$$

Données manquantes

- Les données manquantes sont traitées de façon satisfaisante par la méthode de Bayes
- Les valeurs manquantes sont ignorées, et une probabilité de 1 est considérée

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

Le match n'aura pas lieu

$$\text{Likelihood of "yes"} = 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$$

$$\text{Likelihood of "no"} = 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$$

$$P(\text{"yes"}) = 0.0238 / (0.0238 + 0.0343) = 41\%$$

$$P(\text{"no"}) = 0.0343 / (0.0238 + 0.0343) = 59\%$$

Données numériques

- Une fonction de densité des probabilités $f(x)$ représente la distribution normale des données de l'attribut numérique x en fonction d'une moyenne μ et d'une déviation standard σ

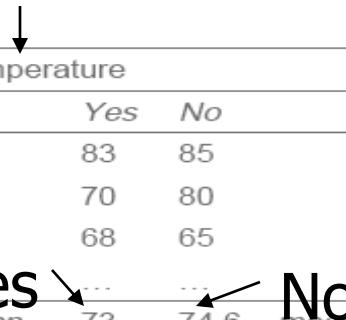
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2}$$

- Les valeurs manquantes ne sont pas incluses dans les calculs des moyennes et des déviations standards

Exemple de calcul de $f(x)$



Outlook		Temperature				Humidity				Windy		Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3		83	85		86	85	False	6	2	9	5
Overcast	4	0		70	80		96	90	True	3	3		
Rainy	3	2		68	65		80	70					
			mean	73	74.6	mean	79.1	86.2					
Sunny	2/9	3/5	std dev	6.2	7.9	std dev	10.2	9.7	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5							True	3/9	3/5		
Rainy	3/9	2/5											

Temperature: $x = 66$, $\mu = 73$, $\sigma = 6.2$ pour yes

$$f(\text{temperature} = 66 \mid \text{yes}) = \frac{1}{\sqrt{2\pi} 6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

Exemple de classification

Outlook	Temp.	Humidity	Windy	Play
Sunny	66	90	true	?

Likelihood of "yes" = $2/9 \times 0.0340 \times 0.0221 \times 3/9 \times 9/14 = 0.000036$

Likelihood of "no" = $3/5 \times 0.0291 \times 0.0380 \times 3/5 \times 5/14 = 0.000136$

$P(\text{"yes"}) = 0.000036 / (0.000036 + 0.000136) = 20.9\%$

$P(\text{"no"}) = 0.000136 / (0.000036 + 0.000136) = 79.1\%$

Le match n'aura pas lieu

Relations entre densité et probabilité

$$\Pr[c - \frac{\varepsilon}{2} < x < c + \frac{\varepsilon}{2}] \approx \varepsilon * f(c)$$

$$\Pr[a \leq x \leq b] = \int_a^b f(t)dt$$

Conclusion

□ Méthode efficace

- La classification ne demande pas des estimations exactes des probabilités, mais seulement que la probabilité maximum soit donnée à la bonne classe
- Les numériques ne sont pas toujours distribués normalement, on a donc besoin d'autres estimations
 - Kernel density estimator

4. Réseaux Bayésiens

■ Classificateurs statistiques

- Basés sur les probabilités conditionnelles
- Prévision du futur à partir du passé
- Suppose l'indépendance des attributs

Fondements

Dérivé du théorème de Bayes

- permet de calculer une probabilité à postériori $P(C_i/X)$ d'un événement C_i sachant que X s'est produit à partir d'une probabilité à priori $P(C_i)$ de production de l'événement C_i
- $$P(C_i/X) = P(X/C_i)*P(C_i) / \sum P(X/C_j)*P(C_j)$$
- Plus simplement si E est l'événement:
 - $$P(E/X) = P(X/E)*P(E)/P(X)$$

Bayésien Naïf

□ Chaque enregistrement est un tuple

- $X = (x_1, x_2, \dots, x_n)$ sur $R(A_1, A_2, \dots, A_n)$
- Il s'agit de classer X parmi m classes C_1, \dots, C_m
- L'événement C_i est l'appartenance à la classe C_i
- Assignation de la classe la plus probable
 - Celle maximisant $P(C_i/X) = P(X/C_i) * P(C_i)/P(X)$
 - $P(X)$ est supposé constant (équi-probabilité des tuples)
- On cherche la classe maximisant :
 - $P(X/C_i) * P(C_i)$ pour $i = 1 \text{ à } m$

On calcule la probabilité de chaque classe étant donné le tuple X

Calcul de $P(X/C_i)$

■ $P(C_i)$ est déduite de l'échantillon :

- Comptage "training set" = $\text{Taille}(C_i) / \text{Taille}(\text{Ech})$
- $P(X/C_i)$ est approchée comme suit :
 - Indépendance des attributs →
 - $P(X/C_i) = \prod_k P(x_k/C_i)$
- $P(x_k/C_i)$ est estimé comme suit:
 - variable nominale = $\text{Taille}(t=x_k \text{ de } C_i)$
 - distribution gaussienne si variable continue

$P(x_k/C_i)$ est la probabilité d'avoir une valeur donnée x_k pour un attribut d'un tuple dans la classe C_i ; Calculée sur le training set

Exemple de problème

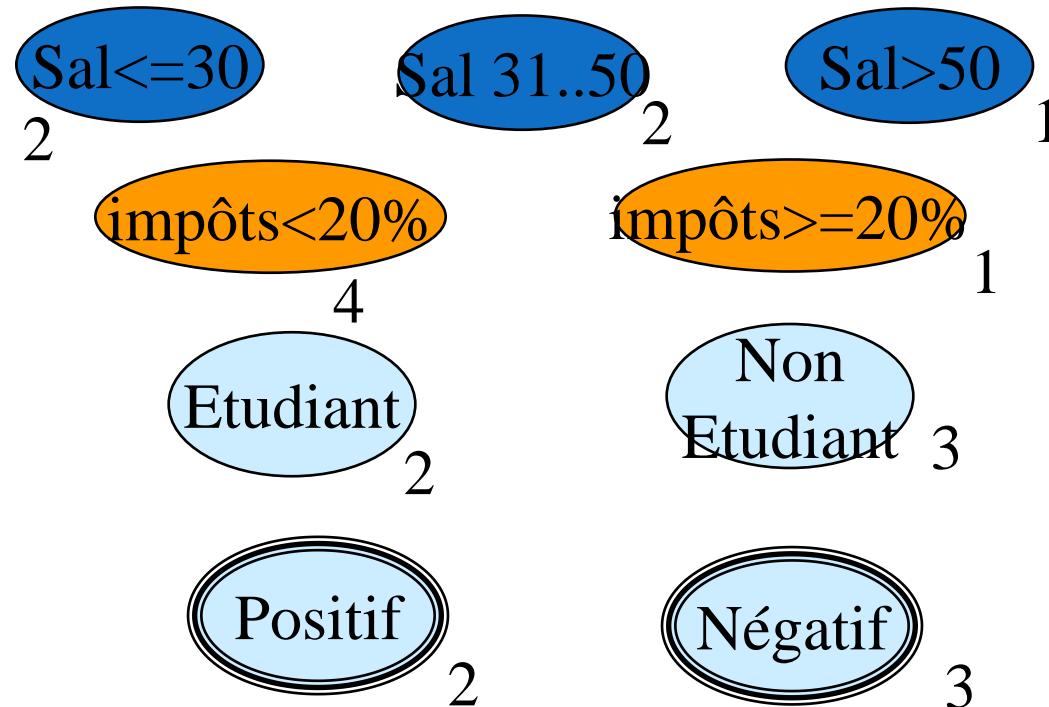
- Faut-il effectuer un contrôle fiscal ?
 - Échantillon de contrôlés

- Faut-il contrôler un nouvel arrivant ?

Salaire	Impôts	Etudiant	Contrôle
20	0	oui	négatif
30	0	non	positif
40	5	oui	positif
40	10	non	négatif
60	10	non	positif

35	2	oui	???
----	---	-----	-----

Les classes nominales



Calcul de Probabilités

- Il s'agit de choisir C_i maximisant $P(C_i/X)$:
 - ▣ $P(\text{Positif}/X) = P(X/\text{Positif})P(\text{Positif})/P(X)$
 - ▣ $P(\text{Négatif}/X) = P(X/\text{Négatif})P(\text{Négatif})/P(X)$
 - ▣ $P(X)$ est supposé constant
- Donc, choisir le plus grand de $\{P(X/\text{Positif})P(\text{Positif}), P(X/\text{Négatif})P(\text{Négatif})\}$
 - ▣ $P(X/\text{Positif}) = \prod_k P(X_k/\text{Positif}) = P(\text{sal30..50}/\text{Positif}) * P(\text{impots}<20\%/\text{Positif}) * P(\text{Etudiant}/\text{Positif}) = 2/3 * 1 * 1/3 = 2/9; P(\text{Positif}) = 3/5$
➔ Produit = 0.13
 - ▣ $P(X/\text{Négatif}) = \prod_k P(X_k/\text{Négatif}) = P(\text{sal30..50}/\text{Négatif}) * P(\text{impots}<20\%/\text{Négatif}) * P(\text{Etudiant}/\text{Négatif}) = 1/2 * 1 * 2/3 = 1/8;$
 $P(\text{Négatif}) = 2/5$ ➔ Produit = 0.05
- On effectuera donc un contrôle !

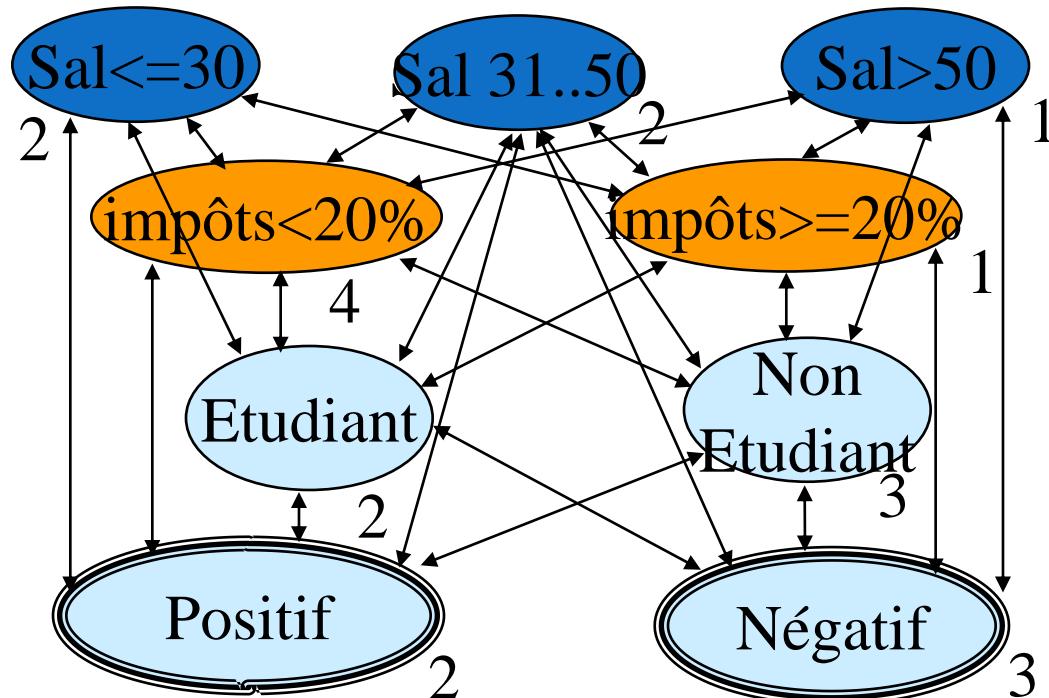
Réseau Bayésien

- Nœuds = Variables aléatoires
- Structure
 - Graphe direct acyclique de dépendance
 - $X \rightarrow Y$ signifie que X est un parent de Y
 - $X \rightarrow \rightarrow Y$ signifie que X est un descendant de Y
 - Les variables non liées sont indépendantes
- Classes à déterminer
 - Nœuds singuliers du réseau
- Probabilités connues
 - à priori et conditionnelles (arcs)

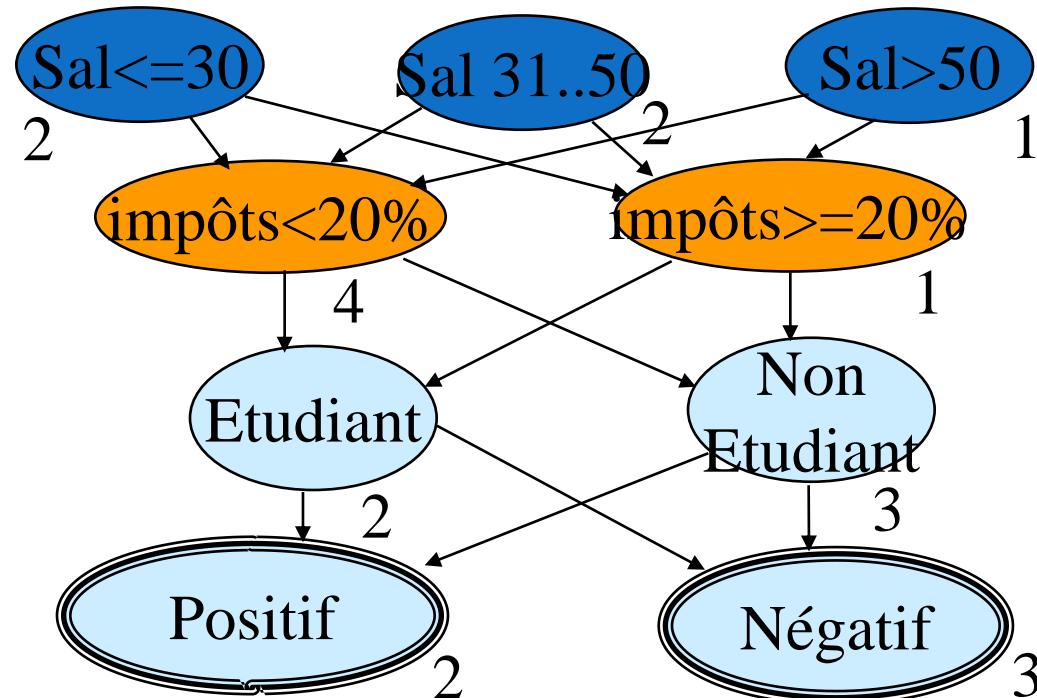
Calculs

- L'instanciation des variables non classes permet de calculer la probabilité des classes
- Application des calculs classiques de probabilité et du théorème de bayes
- Apprentissage à partir d'une base d'échantillons
- Peut être complexe si structure inconnue

Exemple complet



Structure de connaissance



Autre exemple

- Classification de pannes d'ordinateurs
 - Couleur de voyant (Rouge, Vert)
 - Équipement défaillant (UC,MC,PE)
- Envoie d'un dépanneur selon la classe
- Calcul de probabilités sur le training set

P(Couleur/Panne)	Rouge	Vert	P(Panne)
UC	0,70	0,30	0,20
MC	0,40	0,60	0,10
PE	0,20	0,80	0,70
P(Couleur)	0,32	0,68	1,00

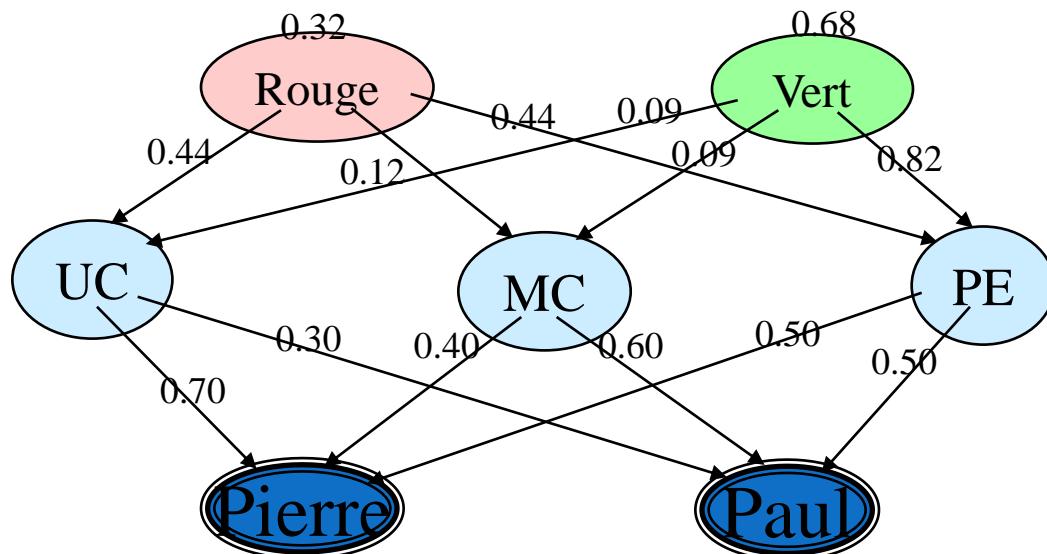
Exemple de réseau

Voyant

Panne

Dépanneur

Rouge



?

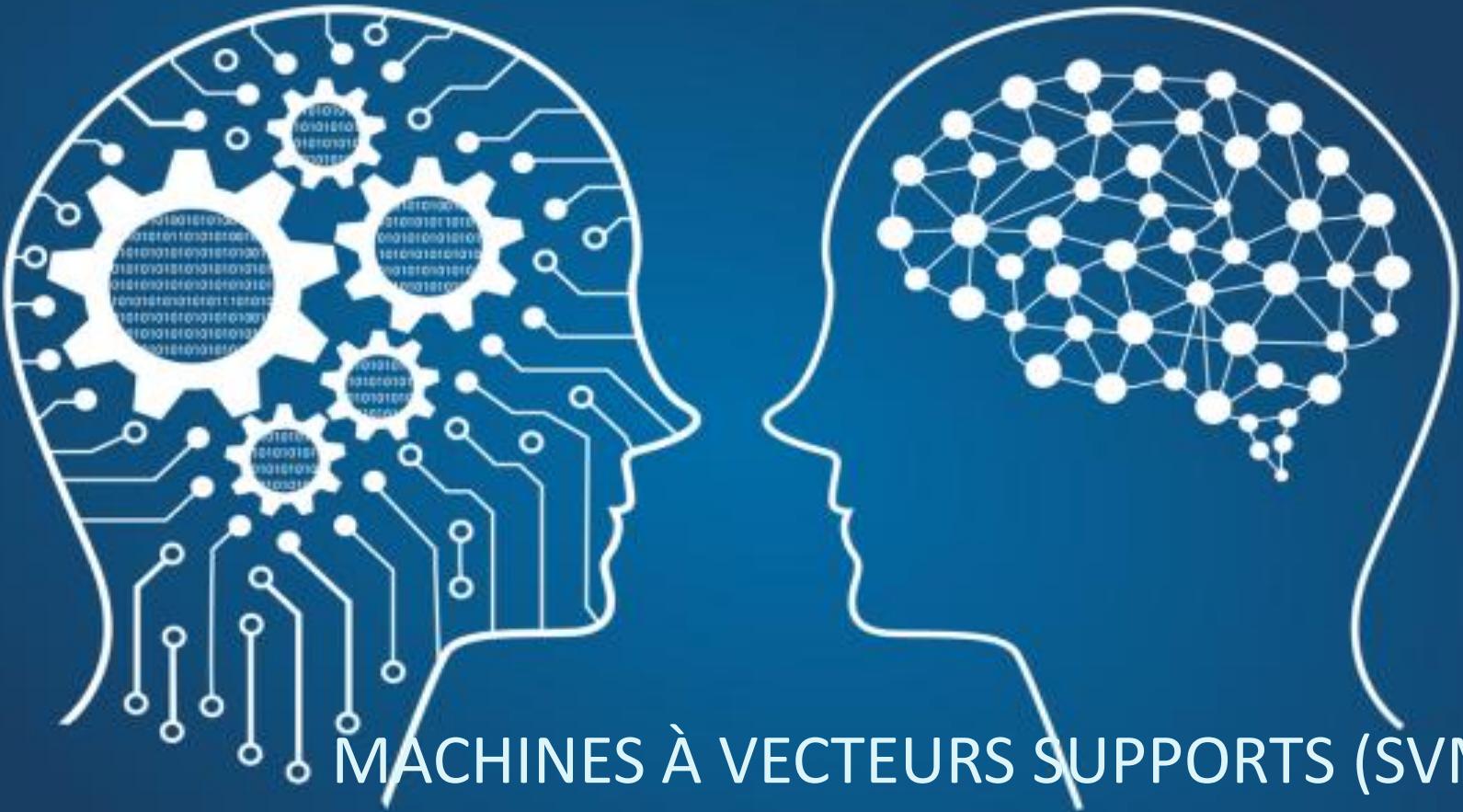
Intérêt

- Permet d'inférer les probabilités dans le réseau
 - méthode d 'inférence du futur à partir du passé
 - les événements X_i doivent être indépendants
 - méthode assez peu appliquée en Data Mining
- Problèmes
 - Comment choisir la structure du réseau ?
 - Comment limiter le temps de calcul ?

Bilan

□ Apprentissage

- si structure connue = calculs de proba.
- si inconnue = difficile à inférer
- Baysien naïf
 - suppose l'indépendance des variables
- Réseaux baysiens
 - permettent certaines dépendances
 - nécessitent des tables d'apprentissage réduites



MACHINES À VECTEURS SUPPORTS (SVM)

Plan

279

Historique

- Qu'est-ce qu'une bonne frontière de séparation pour deux classes linéairement séparables ?
 - La solution SVM
- Adaptation aux cas non linéairement séparables: l'astuce des fonctions noyau
- Exemples d'application
- Conclusion

Historique du SVM

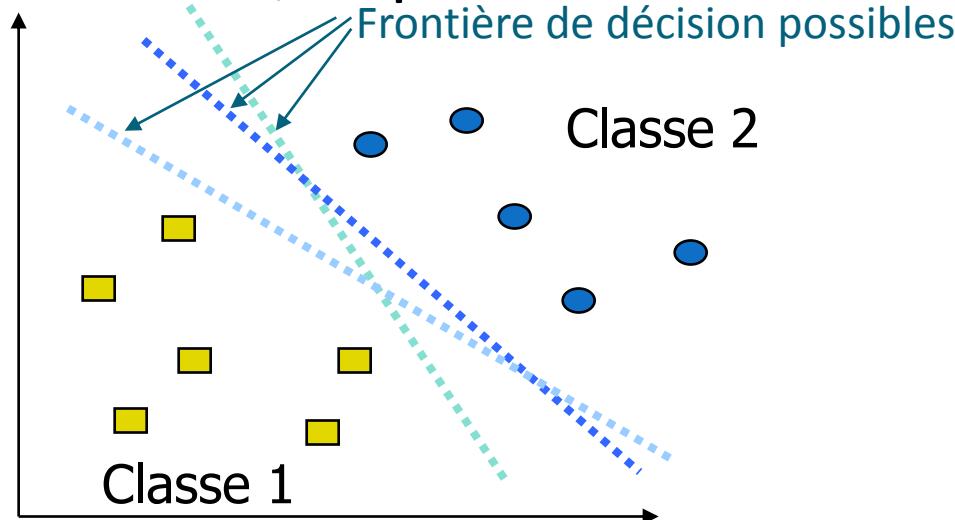
280

- Classifieur devenu populaire depuis que, partant d'images formées de pixels, il a permis des performances égales aux RNA pour reconnaître l'écriture manuscrite.
- Proche de :
 - Séparateurs à vastes marges
 - Méthodes à fonctions noyau
 - Réseaux de neurones à bases radiales

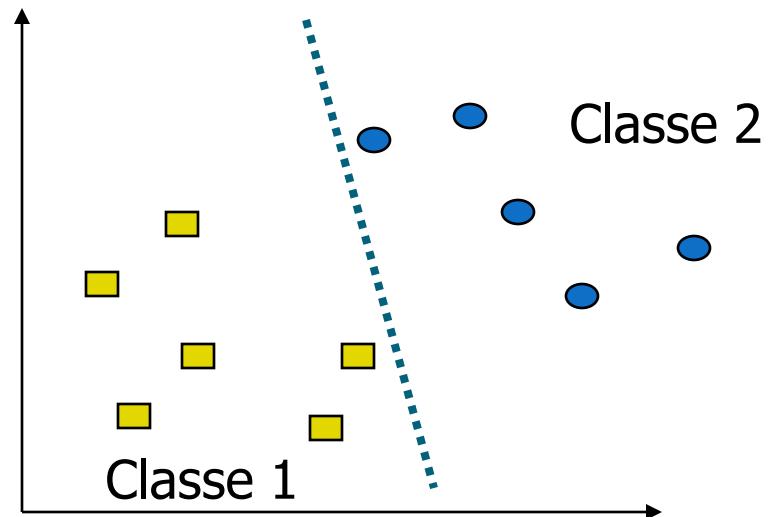
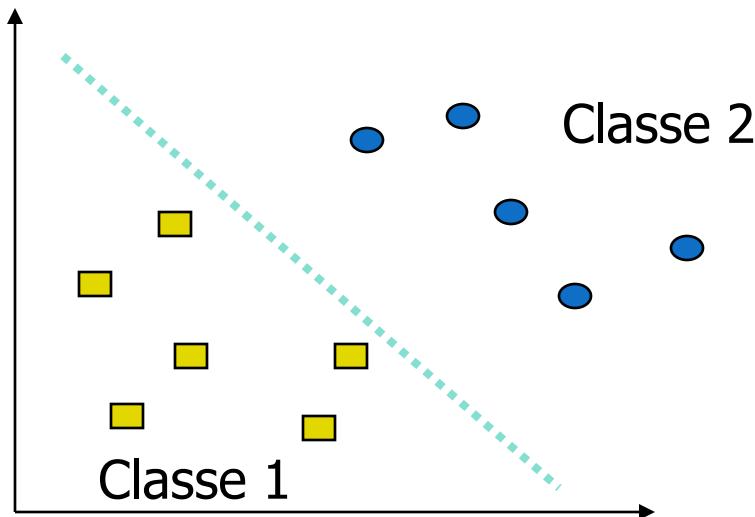
Frontières à deux classes non linéairem séparables

281

- Plusieurs surfaces de décision existent pour séparer les classes ; laquelle choisir ?

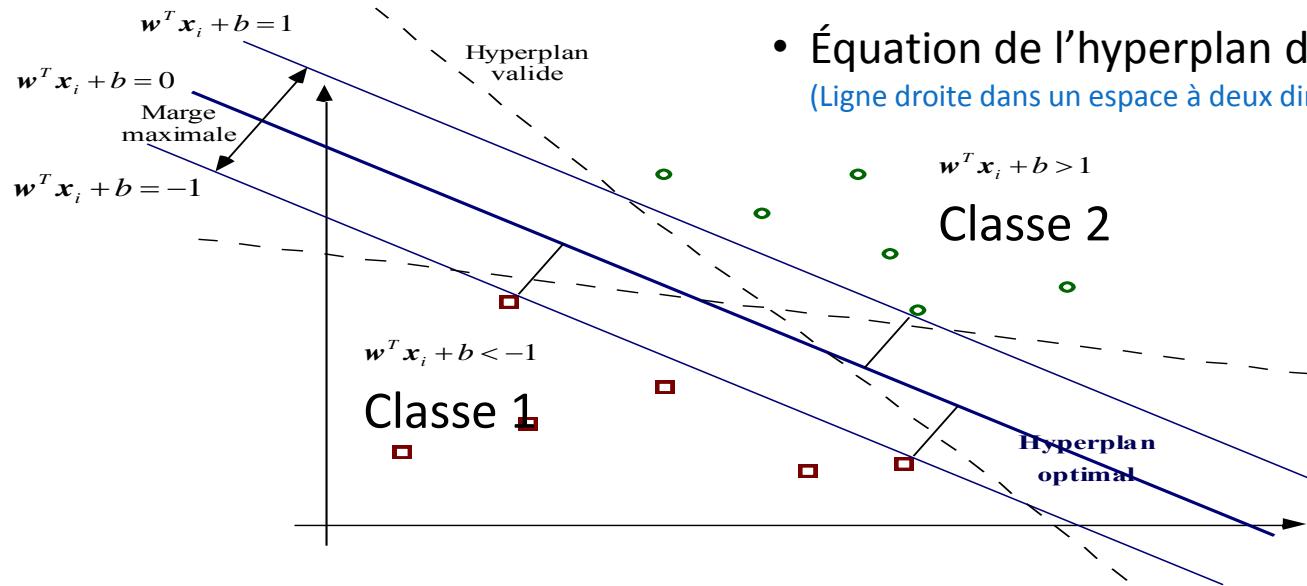


Exemples de choix mal avisés



- ▶ Pour minimiser la sensibilité au bruit, la surface de décision doit être aussi éloignée que possible des données les proches de chaque classe

Hyperplan de plus vaste marge



- Équation de l'hyperplan de séparation : $y = w^T \mathbf{x} + b$
(Ligne droite dans un espace à deux dimensions)

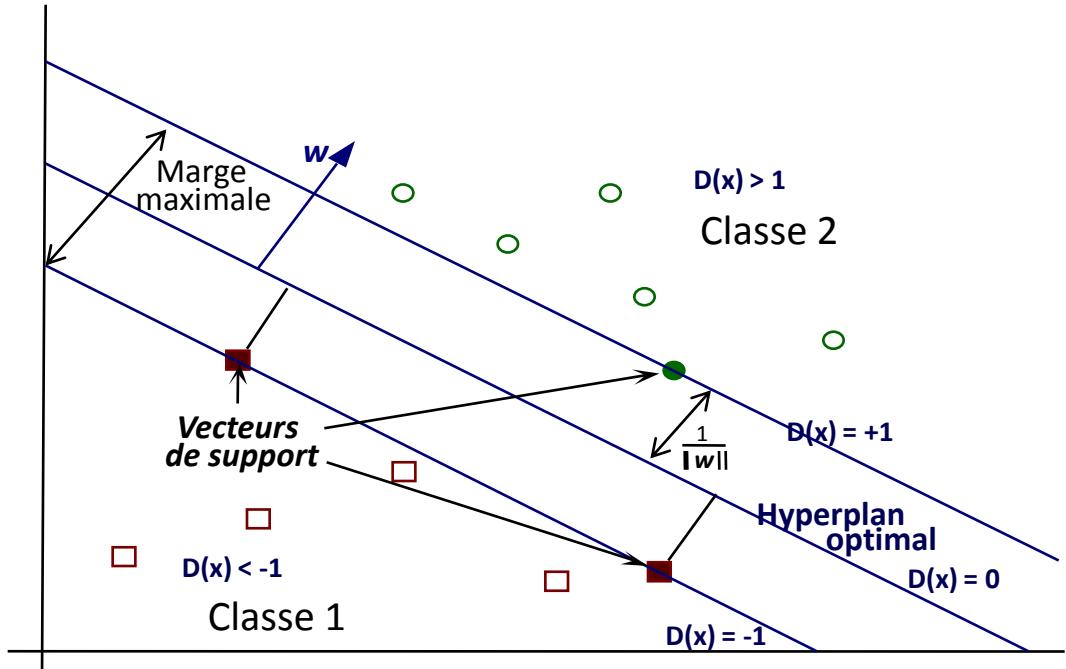
- ▶ Si $\{x_i\} = \{x_1, \dots, x_n\}$ est l'ensemble des données et $y_i \in \{1, -1\}$ est la classe de chacun
$$y_i(w^T \mathbf{x}_i + b) \geq 1, \quad \forall i$$

tout en ayant une distance optimale entre x_i et le plan de séparation

Optimisation de la marge

$$|\mathbf{w}^T \mathbf{x} + b| = 1$$

- ▶ Distance d'un point à l'hyperplan :
$$D(\mathbf{x}) = \frac{|\mathbf{w}^T \mathbf{x} + b|}{\|\mathbf{w}\|}$$
- ▶ Marge max. avant d'atteindre $m = \frac{2}{\|\mathbf{w}\|}$ unitières des deux classes :
- ▶ Maximiser m revient à minimiser $\|\mathbf{w}\|$ tout en préservant le pouvoir de classification :
$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 \text{ sous la contrainte } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \forall i$$



Problème d'optimisation quadratique

- ▶ Maximiser le pouvoir de généralisation du classeur revient donc à trouver w et b tels que :

$$\frac{1}{2} \|w\|^2$$

est minimum

et

$$y_i(w^T x_i + b) \geq 1, \quad i=1, \dots, n$$

- ▶ Si d est la dimension des x_i (nombre d'entrées), cela revient à régler $d+1$ paramètres (les éléments de w , plus b)
 - ▶ Possible par des méthodes d'optimisation classiques ([optimisation quadratique](#)) seulement si d pas trop grand (< qqs 10³)

Les multiplicateurs de Lagrange en 30 s

- Problème : maximiser ou minimiser $f(x)$ sous la contrainte $g(x)=0$
- Solutions possibles :
 - Résoudre $g(x)=0$ et substituer la/les racines trouvées dans $f(x)$; résoudre alors $f'(x) = 0$: pas toujours facile !
 - Considérer $f(x)$ et $g(x)=0$ évoluent pareillement au voisinage de l'extrémum recherché. Leurs tangentes sont alors colinéaires et on a :
$$f'(x) = \alpha g'(x) \quad (\text{ou } f'(x) - \alpha g'(x) = 0) \quad \alpha \text{ étant à déterminer}$$
 - La méthode des multiplicateurs de Lagrange regroupe la fonction à optimiser et la contrainte en une seule fonction $\Lambda(x, \alpha) = f(x) - \alpha g(x)$
 - La solution de $d\Lambda(x, \alpha)/dx = 0$ donne le point où les tangentes sont colinéaires; en même temps, $d\Lambda(x, \alpha)/d\alpha$ répond à la contrainte.

Cas à plusieurs dimensions

- On veut minimiser (ou maximiser) une fonction $f(\mathbf{x})$ en respectant des contraintes $g_i(\mathbf{x})=0, i=1,\dots,n$

- On peut monter qu'à l'extréumum recherché :
(égalité des tangentes)

$$\nabla f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \nabla g_i(\mathbf{x})$$

ou encore : $\nabla f(\mathbf{x}) - \sum_{i=1}^n \alpha_i \nabla g_i(\mathbf{x}) = 0$ où les coefficients α_i sont à déterminer

- Si on forme la fonction (lagrangien) : $\Lambda(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) - \sum_{i=1}^l \alpha_i g_i(\mathbf{x})$

Alors : $\nabla_{\mathbf{x}} \Lambda(\mathbf{x}, \boldsymbol{\alpha}) = \nabla f(\mathbf{x}) - \sum_{i=1}^l \lambda_i \nabla g_i(\mathbf{x})$

et $\nabla_{\boldsymbol{\alpha}_i} \Lambda(\mathbf{x}, \boldsymbol{\alpha}) = g_i(\mathbf{x})$

=> la solution de $\nabla_{\mathbf{x}} \Lambda(\mathbf{x}, \boldsymbol{\alpha}) = 0$ mène à un extréumum qui respecte les contraintes.

Forme duale du problème d'optimisation

- ▶ Dans notre cas, la fonction à optimiser sous contrainte est celle donnant la marge maximale, ce qui revient à trouver les paramètres (w, b) correspondants.
- ▶ Donc, partant d'un ensemble de données $\{(x_i, y_i)\}$, de l'ensemble de contraintes (w, b) , on a:
$$\begin{cases} \Lambda(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i \{(x_i^T w + b) y_i - 1\} \\ \forall i \quad \alpha_i \geq 0 \end{cases}$$
 et des paramètres à optimiser $\nabla_{w,b} \Lambda(w, b, \alpha) = 0$

Le minimum recherché est donné par la solution de $\min_{\alpha} \max_x \Lambda(x, \alpha) = \max_x (\min_{\alpha} \Lambda(x, \alpha))$

- ▶ Il existe un problème dual plus facile à résoudre: $\nabla_{w,b} \Lambda(w, b, \alpha) = 0$
 - Théorème de Kuhn-Tucker :

=> on peut aussi trouver w et b en solvant

sujet aux

Forme duale du problème d'optimisation

289

□ Avantage de résoudre $\nabla_{\alpha} \Lambda(\mathbf{w}, b, \boldsymbol{\alpha}) = 0$ au lieu de $\nabla_{w,b} \Lambda(\mathbf{w}, b, \boldsymbol{\alpha}) = 0$

- La complexité du problème d'optimisation devient proportionnelle à n (nombre de paires d'apprentissage (\mathbf{x}_i, y_i)) et non d (dimension de chaque \mathbf{x}_i)
- Possible d'obtenir des solutions pour des problèmes impliquant $\approx 10^5$ exemples
- C'est aussi un problème pour lequel le maximum global des α_i peut toujours être trouvé

Formulation du problème dual

- ▶ Partant de $\begin{cases} \Lambda(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \{ (\mathbf{x}_i^T \mathbf{w} + b) y_i - 1 \} \\ \forall i \quad \alpha_i \geq 0 \end{cases}$
- $\nabla_{\mathbf{w}, b} \Lambda(\mathbf{w}, b, \boldsymbol{\alpha}) = 0$ donne $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$
- $\sum_{i=1}^n \alpha_i y_i = 0$
- et $\Lambda(\mathbf{w}, b, \boldsymbol{\alpha})$

- ▶ On $\boxed{\begin{cases} \Lambda(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j) \\ \forall i \quad \alpha_i \geq 0 \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}}$
- $\Lambda(\mathbf{w}, b, \boldsymbol{\alpha})$ $\nabla_{\boldsymbol{\alpha}} \Lambda(\mathbf{w}, b, \boldsymbol{\alpha}) = 0$

Solution du problème d'optimisation

$$\left\{ \begin{array}{l} \hat{\mathbf{w}} = \sum_{i=1}^{n_s} \hat{\alpha}_i y_i \mathbf{x}_i \\ \hat{b} = y_s - \sum_{i=1}^{n_s} \hat{\alpha}_i y_i (\mathbf{x}_i^T \mathbf{x}_s) \\ D(\mathbf{x}) = (\hat{\mathbf{w}}^T \mathbf{x} + \hat{b}) \end{array} \right.$$

- $\hat{\cdot}$: estimé
- n_s : nombre de vecteurs de support (\mathbf{x}_i avec $\alpha \neq 0$)
- (\mathbf{x}_S, y_S) : vecteur de support arbitraire (pour trouver $\hat{\mathbf{w}}$ et \hat{b})

- ▶ Les données \mathbf{x}_i avec $\alpha \neq 0$ sont appelées vecteurs de support. Ils correspondent aux points les plus proches de la surface de séparation $\hat{\mathbf{w}}$ et \hat{b}
- Dans l'expression du Lagrangien pour déterminer $\hat{\mathbf{w}}$ et \hat{b} , seuls interviennent les produits scalaires entre les données

$$L(\mathbf{w}, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{x}_j)$$

Caractéristiques de la solution

292

- Puisque plusieurs α_i sont nuls, w est une combinaison linéaire d'un petit nombre de données
- La surface de décision est uniquement déterminée par les n_s vecteurs de support trouvés:

$$\hat{w} = \sum_{i=1}^{n_s} \hat{\alpha}_i y_i \mathbf{x}_i$$

$$\hat{b} = y_s - \sum_{i=1}^{n_s} \hat{\alpha}_i y_i (\mathbf{x}_i^T \mathbf{x}_s)$$

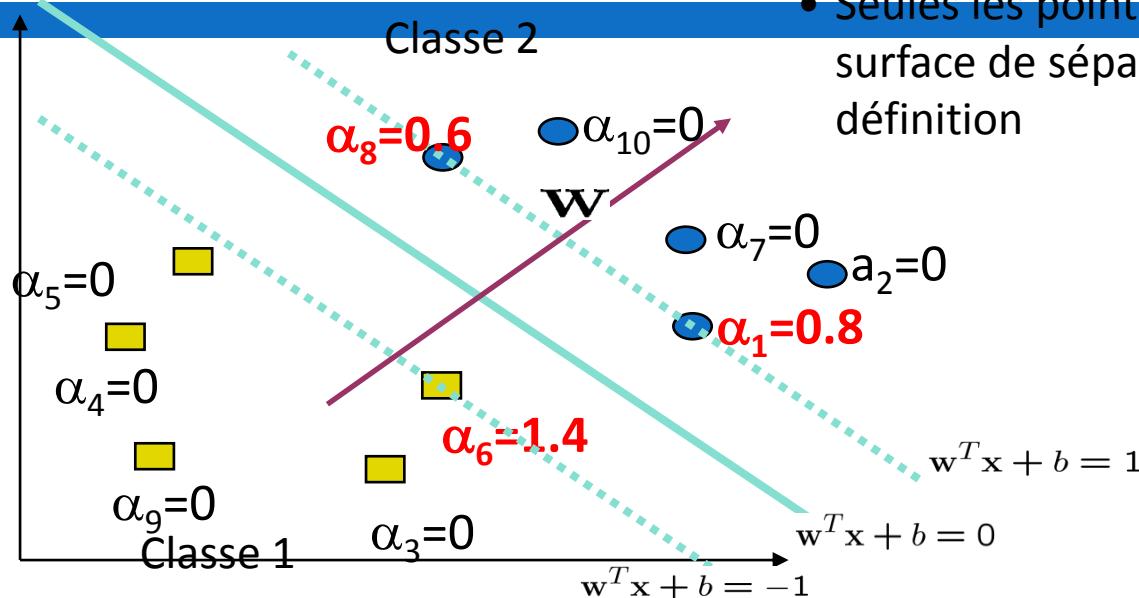
$$\hat{w}^T z + \hat{b} = \sum_{i=1}^{n_s} \hat{\alpha}_i y_i (\mathbf{x}_i^T z) + \hat{b}$$

- Pour classer une nouvelle donnée z

- Calculer et classer z dans la classe 1 si le résultat est positif, la classe 2 s'il est négatif

Interprétation géométrique

293



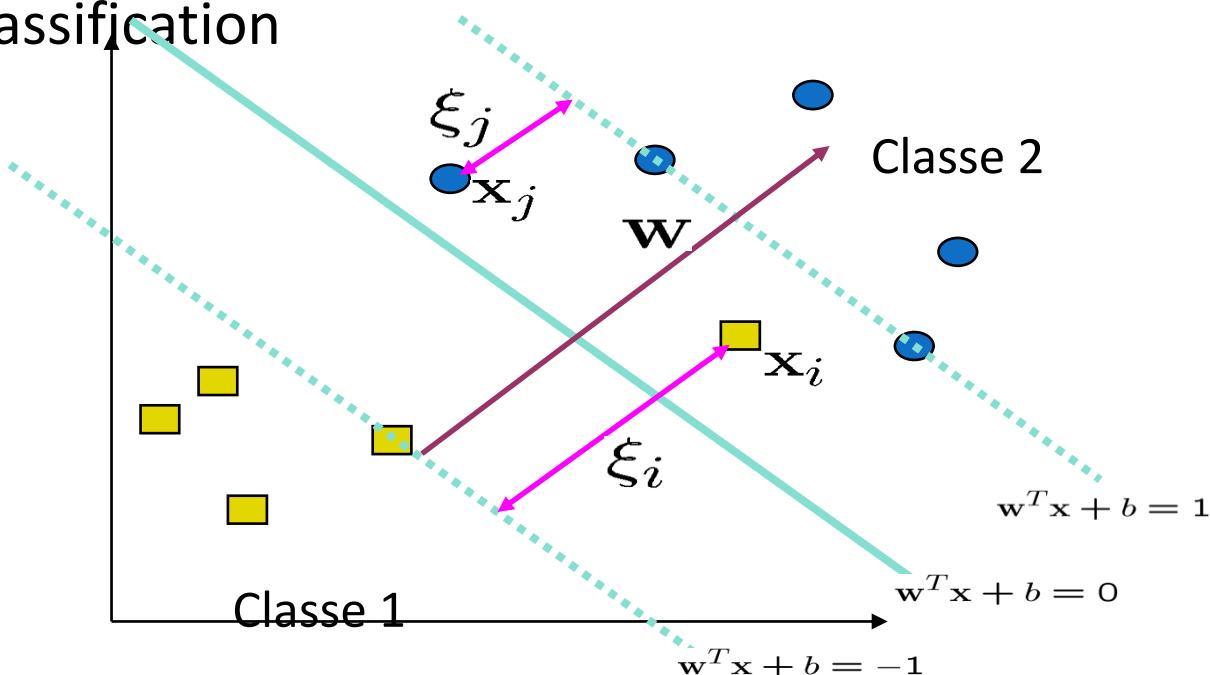
- Seules les points les plus proches de la surface de séparation influent sur sa définition

- Il existe des limites théoriques pour l'erreur de classification de données nouvelles
 - Plus grande la marge, plus petite la limite
 - Plus petit le nombre de SV, plus petite la limite

Et pour un cas non linéairement séparable ?

294

- On peut introduire une marge d'erreur ξ_i pour la classification



Hyperplan à marges douces

295

- $\xi_i = 0$ s'il n'existe pas d'erreur pour x_i

- ξ_i sont des variables qui donnent du "mou" aux marges optimales

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq 1 - \xi_i & y_i = 1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 + \xi_i & y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases}$$

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

- Nous voulons minimiser

$$\text{Minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$

- C : parameter to $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \dots, \xi_i \geq 0$ a marge
- Le problème d'optimisation devient

Détermination de l'hyperplan de séparation

296

- La forme duale du problème est

$$\text{max. } W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

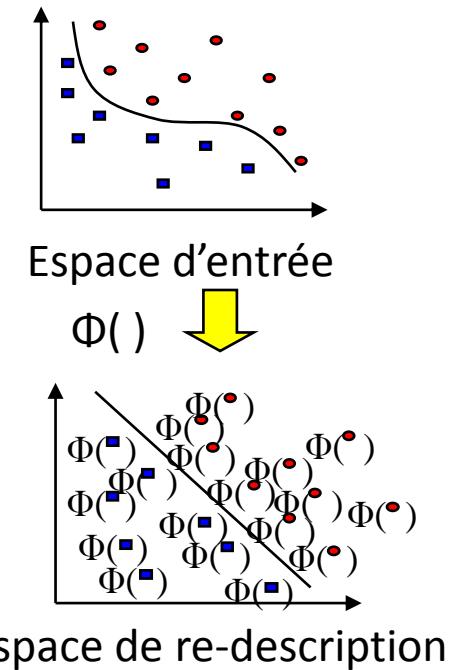
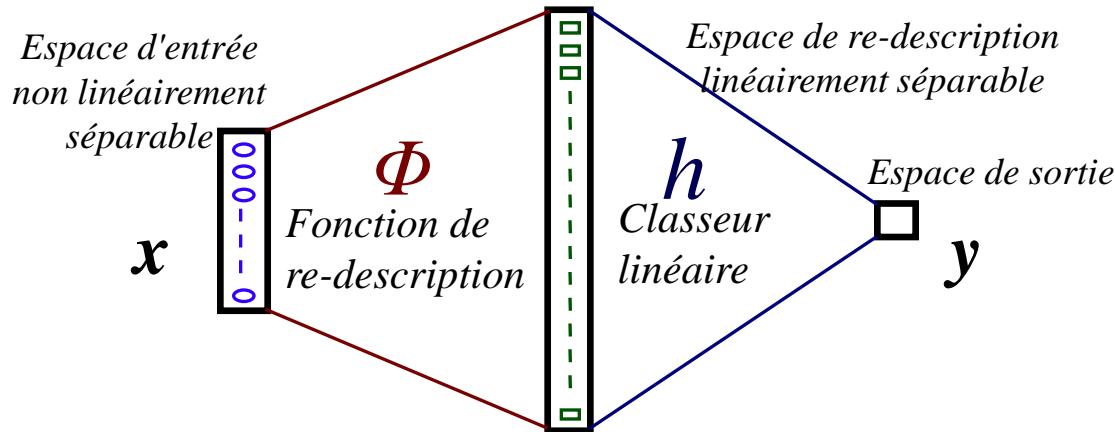
$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$$

$$\mathbf{w} = \sum_{j=1}^s \alpha_j y_j \mathbf{x}_j$$

- \mathbf{w} est aussi donné par
- La seule différence avec le cas linéairement séparable est qu'il existe une limite supérieure C aux α_i

Extension à une surface de séparation non-linéaire

- « Simplifier les choses » en projetant les x_i dans un nouvel espace où ils sont linéairement séparables



Modification due à la transformation

- Substituer les arguments transformés dans les produits scalaires lors de la phase d'apprentissage,

Problème

original :

$$\max. \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$

Après

xformation :

$$\max. \quad W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j \boldsymbol{\Phi}(\mathbf{x}_i)^T \boldsymbol{\Phi}(\mathbf{x}_i)$$

subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$

- Mais trouver $\Phi()$ pas évident !

Modification due à la transformation

299

- Les nouvelles données z sont toujours classées dans la classe 1 si $f \geq 0$, la classe 2 sinon :

Original : $\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
 $f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$

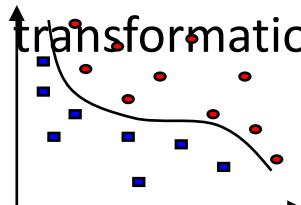
Après transformation : $\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \phi(\mathbf{x}_{t_j})$
 $f = \langle \mathbf{w}, \phi(\mathbf{z}) \rangle + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \Phi(x_{t_j})^T \Phi(z) + b$

$$D(\mathbf{x}) = \sum_{j=1}^s \hat{\alpha}_j y_j K(\mathbf{x}_j, \mathbf{x}) + \hat{b}$$

Extension à une surface de séparation non-linéaire

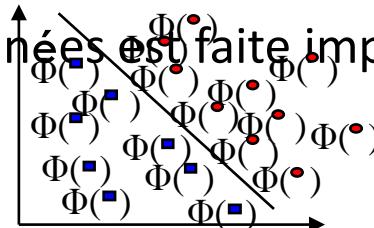
□ Problèmes cependant :

- ▶ $\Phi = ?$
- ▶ Grand effort de calcul potentiel (*d* explose !)
- SVM à fonctions noyaux résout les deux problèmes
 - ▣ Efficacité computationnelle
 - ▣ La transformation désirée des données est faite implicitement !



$\Phi()$

Espace d'entrée



Espace de re-description

La transformation désirée des données est faite implicitement !

L'astuce des fonctions noyau

301 □ Définition d'une fonction noyau :

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

- ⇒ La connaissance de $K(\cdot)$ permet de calculer indirectement un produit scalaire où intervient $\Phi(\cdot)$, sans connaître l'expression de $\Phi(\cdot)$
- Or, seuls des produits scalaires interviennent dans la solution du problème d'optimisation
 - Un autre avantage d'utiliser $K()$ est qu'il représente intuitivement la similarité entre les x et y , obtenue de nos connaissances a priori
 - Cependant, $K(x,y)$ doit satisfaire certaines conditions (conditions de Mercer) pour que le $\Phi(\cdot)$ correspondant existe

Les conditions de Mercer

- Pour une fonction K symétrique, il existe une fonction Φ telle que :

$$K(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') = \sum_{i=1}^m g_i(\mathbf{x}) \cdot g_i(\mathbf{x}')$$

$$\int f(\mathbf{x})^2 d\mathbf{x} \text{ est fini}$$

ssi, pour toute fonction f telle que $d\mathbf{x} d\mathbf{x}' \geq 0$

l'on a :

- Si cette condition est vérifiée, on peut appliquer la fonction noyaux dans le SVM

Exemple de d'utilisation

303

- Définissons la fonction noyau $K(x,y)$ telle que, pour toute paire de vecteurs

$$\mathbf{x} = (x_1, x_2) \text{ et } \mathbf{y} = (y_1, y_2) :$$

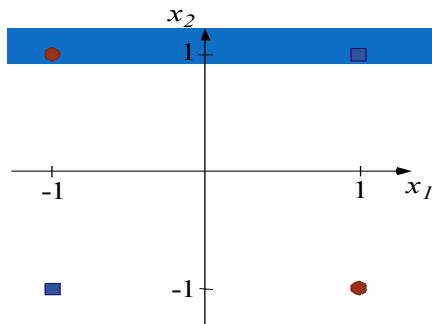
$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1 y_1 + x_2 y_2)^2$$

- Considérons maintenant une transformation Φ qui prend un vecteur de dimension 2 et le projette dans un espace de dimension 6 :

$$\begin{aligned}\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle &= (1 + x_1 y_1 + x_2 y_2)^2 \\ &= K(\mathbf{x}, \mathbf{y})\end{aligned}$$

On peut voir en effectuant le calcul que

Illustration : le cas du XOR



□ Il faut résoudre :

$$\begin{cases} \max_{\alpha} \left(\sum_{i=1}^4 \alpha_i - \frac{1}{2} \sum_{i=1}^4 \sum_{j=1}^4 \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right) \\ \forall i \quad 0 \leq \alpha_i \leq C \\ \sum_{i=1}^4 \alpha_i y_i = 0 \end{cases}$$

Index i	\mathbf{x}_i	y
1	(1,1)	1
2	(1,-1)	-1
3	(-1,-1)	1
4	(-1,1)	-1

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1 y_1 + x_2 y_2)^2$$

□ $Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$
 Si on reprend la fonction noyau
 , on obtient les équations suivantes pour le
 Lagrangien: $9\alpha_2^2 - 2\alpha_2\alpha_3 + 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2$

$$\alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0$$

Illustration : le cas du XOR

305

- Le maximum de $Q(\alpha)$ est obtenu en prenant ses dérivées par rapport aux α_i et en trouvant les valeurs de α_i qui les annulent :

$$\begin{cases} 1 - 9\alpha_1 + \alpha_2 - \alpha_3 + \alpha_4 = 0 \\ 1 + \alpha_1 - 9\alpha_2 + \alpha_3 - \alpha_4 = 0 \\ 1 - \alpha_1 + \alpha_2 - 9\alpha_3 - \alpha_4 = 0 \\ 1 + \alpha_1 - \alpha_2 + \alpha_3 - 9\alpha_4 = 0 \end{cases}$$

- La valeur optimale des multiplicateurs de Lagrange est :

$$\hat{\alpha}_1 = \hat{\alpha}_2 = \hat{\alpha}_3 = \hat{\alpha}_4 = \frac{1}{8}$$

- Les 4 données du où exclusif sont donc des vecteurs de support, puisque aucune valeur trouvée de α n'est nulle

Illustration : le cas du XOR

- Dans l'espace de Espace de re-description :

$$\begin{cases} \hat{\mathbf{w}} = \sum_{i=1}^{n_s} \hat{\alpha}_i y_i \Phi(\mathbf{x}_i) \\ \hat{b} = y_s - \sum_{i=1}^{n_s} \hat{\alpha}_i y_i K(\mathbf{x}_i^T \mathbf{x}_s) \\ D(\mathbf{x}) = \sum_{j=1}^s \hat{\alpha}_j y_j K(\mathbf{x}_j, \mathbf{x}) + \hat{b} \end{cases}$$

- Donc :

$$\begin{aligned} \hat{\mathbf{w}} &= \frac{1}{8} [-\Phi(\mathbf{x}_1) + \Phi(\mathbf{x}_2) + \Phi(\mathbf{x}_3) - \Phi(\mathbf{x}_4)] \\ &= \frac{1}{8} \left\{ -\begin{pmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{pmatrix} \right\} = \begin{pmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

(on connaît $\Phi()$ dans cet exemple, mais il n'est pas requis en général, car l'équation de la marge dépend seulement de $K()$)

$$\hat{b} = 1 - \frac{1}{8} \sum_{j=1}^4 y_j K(\mathbf{x}_j, \mathbf{x}_1) = 1 + \frac{1}{8} \sum_{j=1}^4 (-1)^j K(\mathbf{x}_j, \mathbf{x}_1) = 0$$

et $D(\mathbf{x}) = \frac{1}{8} \sum_{j=1}^4 y_j K(\mathbf{x}_j, \mathbf{x}) = -\frac{1}{8} \sum_{j=1}^4 (-1)^j K(\mathbf{x}_j, \mathbf{x}) = -x_1 x_2$

(on aurait obtenu le même résultat en utilisant $\Phi()$:

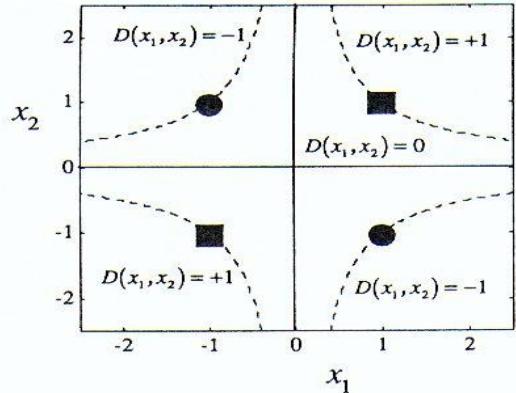
$$\hat{\mathbf{w}}^T \Phi(\mathbf{x}) = \left(0, 0, \frac{-1}{\sqrt{2}}, 0, 0, 0 \right) \begin{pmatrix} 1 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \end{pmatrix} = -x_1 x_2$$

$\frac{1}{2} \|\hat{\mathbf{w}}\|^2 = \frac{1}{2} \hat{\mathbf{w}}^T \hat{\mathbf{w}} = \frac{1}{2} \left(\sum_{i=1}^4 \hat{\alpha}_i y_i \Phi(\mathbf{x}_i) \right)^T \left(\sum_{j=1}^4 \hat{\alpha}_j y_j \Phi(\mathbf{x}_j) \right)$

► La marge optimale est : $\frac{1}{2} \cdot \frac{4}{8^2} \sum_{i=1}^4 \sum_{j=1}^4 y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{4} \Rightarrow \|\hat{\mathbf{w}}\| = \frac{1}{\sqrt{2}}$

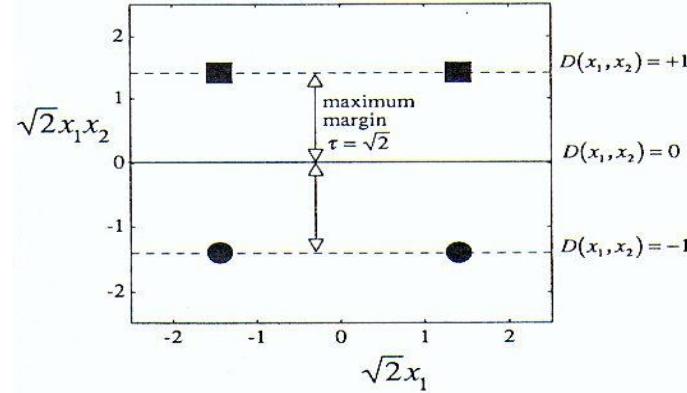
Illustration : le cas du XOR

307



Séparatrice dans l'espace
d'entrée

$$D(x) = -x_1 x_2$$



Séparatrice dans l'espace

$$\sqrt{2} \Phi(x) = 0$$

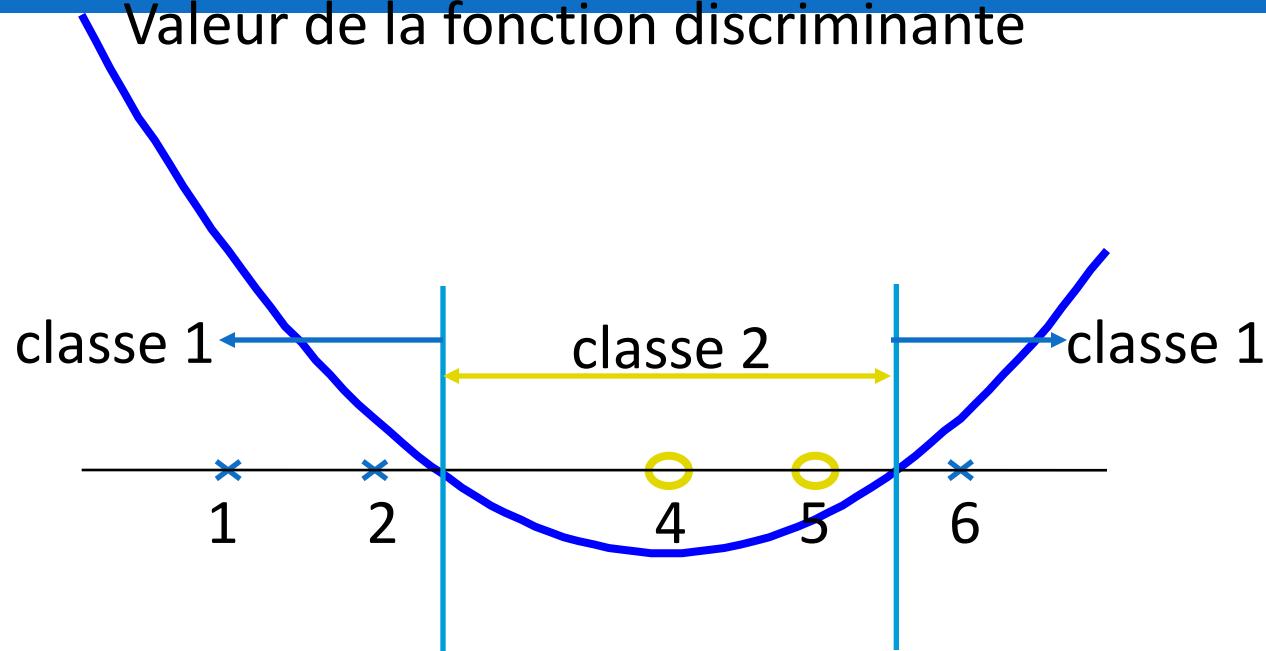
Autre Exemple

- Supposons 5 nombres $x_1=1, x_2=2, x_3=4, x_4=5, x_5=6$, avec
 - ▣ $1, 2, 6 \in$ classe 1 ($y=1$)
 - ▣ $4, 5 \in$ classe 2 ($y=-1$)
 - ▣ Donc: $\{(x_i, y_i)\}_{i=1, \dots, 5} = \{(1, 1), (2, 1), (4, -1), (5, -1), (5, 1)\}$
- Utilisons à nouveau le noyau polynomial de degré 2
 - ▣ $K(x, y) = (1 + x^T y)^2$
 - ▣ $\max_{\sum \alpha_i = 100} \sum \alpha_i \sum_{j=1}^5 \alpha_j y_i y_j (x_i x_j + 1)^2$
- Trouver $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$
 subject to $100 \geq \alpha_i \geq 0, \sum_{i=1}^5 \alpha_i y_i = 0$

Exemple

309

Valeur de la fonction discriminante



Exemples de fonctions noyaux

310

- Noyau polynomial de degré d

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^d$$

- Noyau à $K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$ tension σ

- Très proche de $K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x}^T \mathbf{y} + \theta)$ dans à base radiale

- Sigmoïde avec paramètres κ et θ

- Ne satisfait pas la condition de Mercer pour tous κ et θ

- La recherche d'autres fonctions noyau pour diverses

Classification multi-classes

311

- SVM est à la base un classifieur binaire
- On peut changer la formulation pour permettre la classification multi-classe
 - ▣ L'ensemble des données est divisé en deux parts de multiples façons, et classé ensuite
 - Un contre tous ou un contre chaque alternative
 - Un SVM séparé est formé pour chaque division
 - ▣ La classification multi-classes est accomplie en combinant la sortie de tous les SVM

Sommaire: étapes de la classification

312

- Préparer la matrice des patrons
- Choisir la fonction noyau à utiliser
- Choisir les paramètres de la fonction noyau et la valeur de C (valeurs suggérées par le logiciel SVM ou essai-erreur).
- Exécuter l'algorithme d'apprentissage pour trouver α_i

Effet des paramètres de contrôle.

313

- Apprentissage de données en damier
 - Apprentissage de deux classes
 - SVM à fonction noyau gaussienne

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}}$$

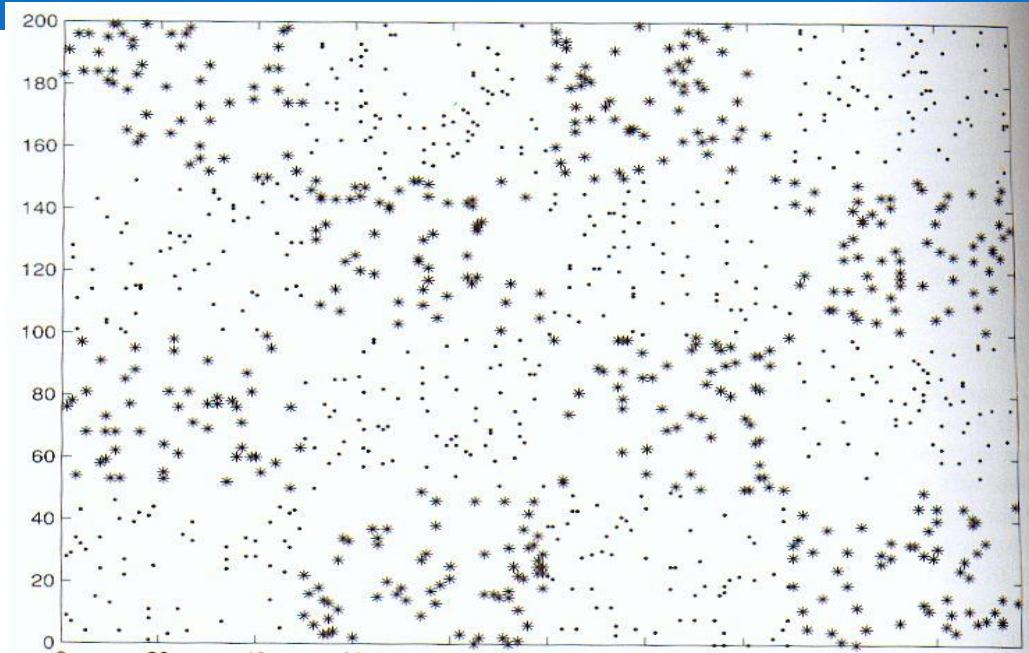


Figure 10.2 Training points for checkerboard pattern

Effet des paramètres de contrôle

314

□ Apprentissage de deux classes

- exemples tirés uniformément sur l'échiquier

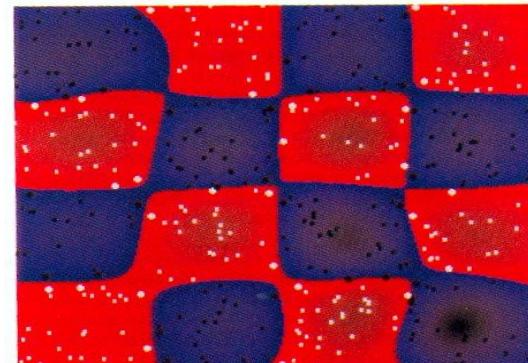
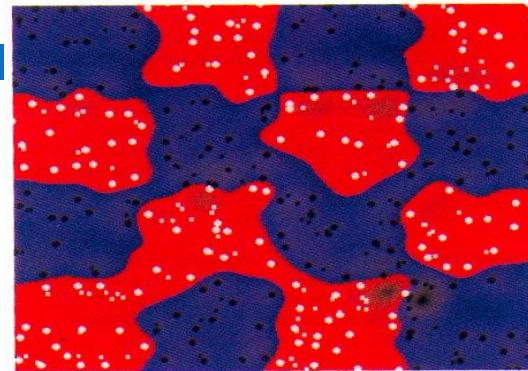
□ SVM à fonctions noyau gaussienne

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}}$$

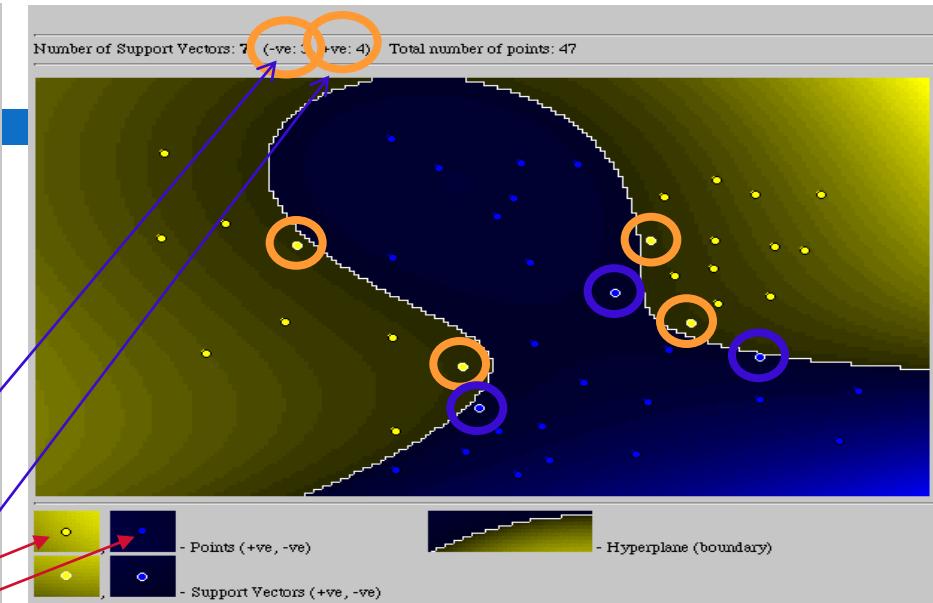
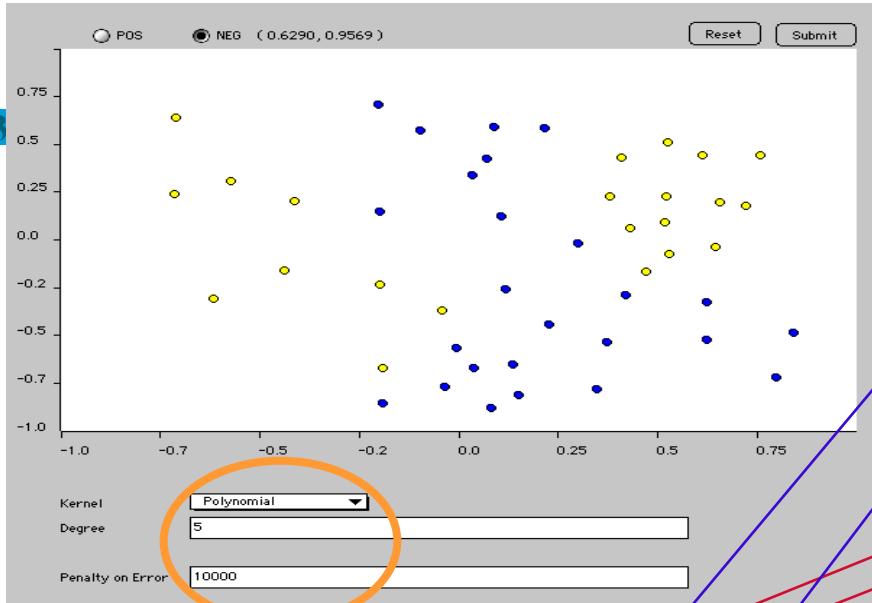
□ Ici deux valeurs de σ

- En haut : petite valeur
- En bas : grande valeur

□ Les gros points sont des exemples critiques



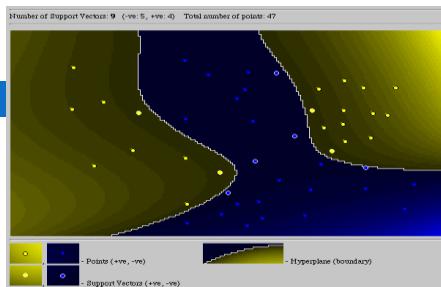
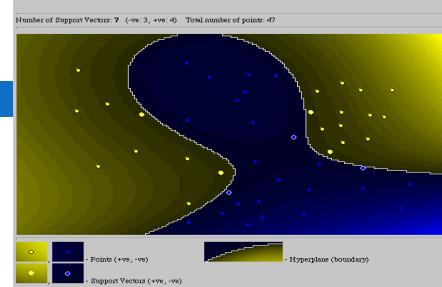
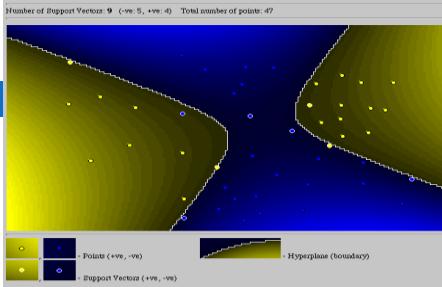
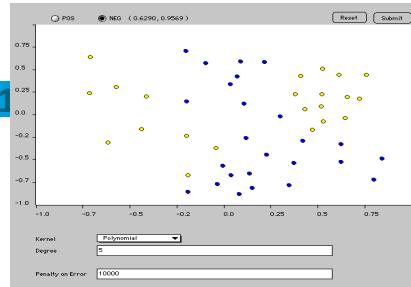
Une applette de démonstration



- <http://svm.cs.rhul.ac.uk/pagesnew/GPat.shtml>
- 47 exemples (22 +, 25 -)
- Exemples critiques : 4 + et 3 -
- Ici fonction polynomiale de degré 5 et $C = 10000$

Paramètres de contrôle : les fonctions noyau

31



(5-, 4+)

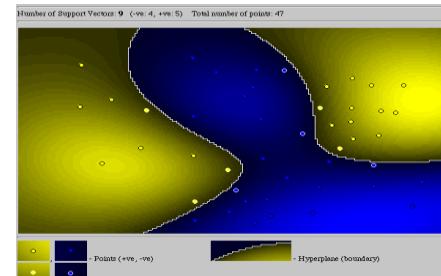
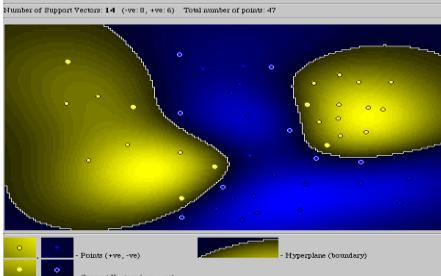
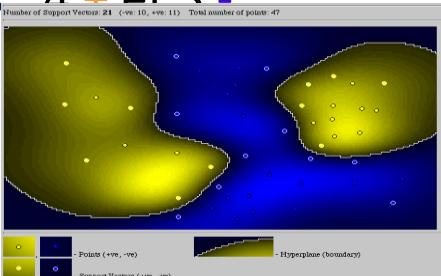
(3-, 4+)

(5-, 4+)

Ici *fonction polynomiale* de degré 2, 5, 8 et $C = 10000$

- 47 exemples (22 +, 25 -)

- Exemples critiques · 1 + et 3 -



(10-, 11+)

(8-, 6+)

(4-, 5+)

Ici *fonction Gaussienne* de $\sigma = 2, 5, 10, 20$ et $C = 10000$

Domaines d'application des SVMs

317

□ Traitement d'images

- Reconnaissance de caractères manuscrits
 - Reconnaissance de scènes naturelles
 - Reconnaissance de visages
-
- *Entrées* : image bidimensionnelle en couleur ou en tons de gris
 - *Sortie* : classe (chiffre / personne)

Domaines d'application des SVMs

318

□ Catégorisation de textes

- Classification d'e-mails
- Classification de pages web

- *Entrées* : document (texte ou html)
 - Approche « sac de mots »
 - Document = vecteur de mots (lemmatisés pondérés par tf-idf)
- *Sortie* : catégorie (thème, spam/non-spam)
- *Noyau* :
 - Produit scalaire des vecteurs

Domaines d'application des SVMs

319

□ Diagnostic médical

- Évaluation du risque de cancer
- Détection d'arythmie cardiaque
- Évaluation du risque d'accidents cardio-vasculaires à moins de 6 ans

- *Entrées* : état du patient (sexe, age, bilan sanguin, ...)
- *Sortie* :

Extensions

320

- Leçon à retenir des SVM:
 - *Un algorithme linéaire dans l'espace de re-description peut remplacer un algorithme non-linéaire dans l'espace d'entrée*
 - Les algorithmes linéaires classiques peuvent être généralisés en des versions non-linéaires en allant vers l'espace de re-description
 - ACP à noyaux, k-moyennes à noyaux, etc.
 - Régression

SVM et régression

321

□ Fonction de perte : $|y - f(\mathbf{x})|_\varepsilon = \max\{0, |y - f(\mathbf{x})| - \varepsilon\}$

■ Régression linéaire :

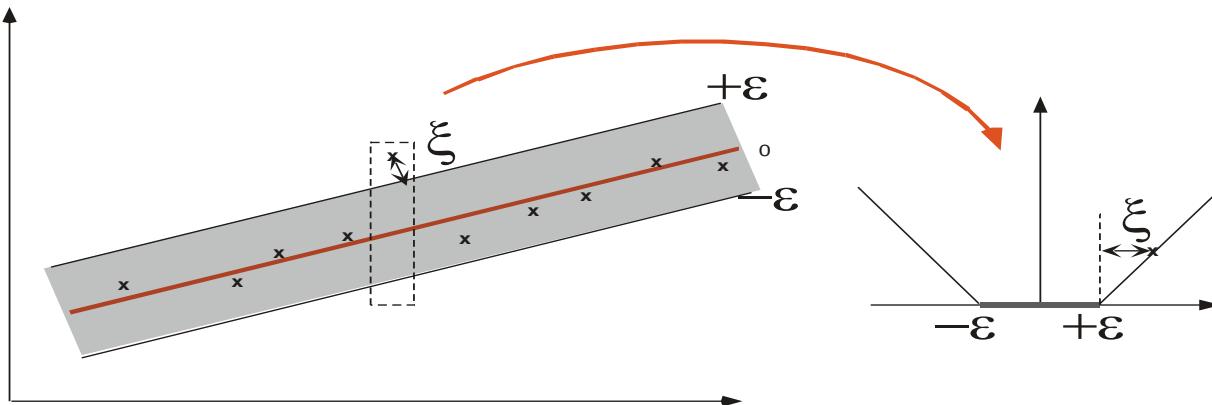
$$f(\mathbf{x}) := (\mathbf{w} \cdot \mathbf{x}) + w_0$$

■ Soit à minimiser :

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m |y_i - f(\mathbf{x}_i)|_\varepsilon$$

■ Généralisation :

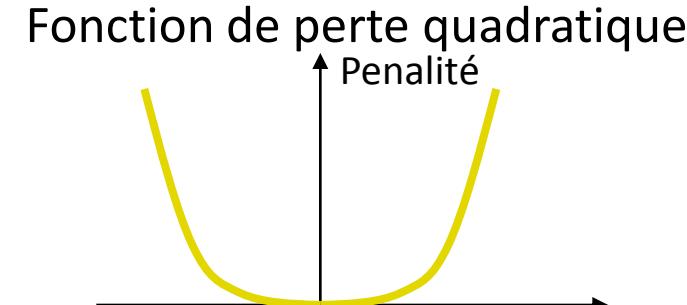
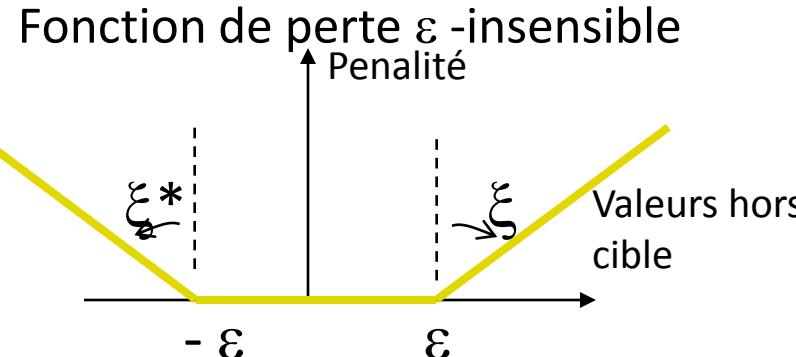
$$f(\mathbf{x}) = \sum_{i=1}^m (\alpha_i^\star - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + w_0$$



Régression vectorielle à support ε (ε -SVR)

322

- Régression linéaire dans l'espace de redescription
- À l'encontre de la régression par moindres carrés, la fonction d'erreur est une fonction de perte ε -insensible
 - Intuitivement, une erreur inférieure à ε est ignorée
 - Cela mène à des points de marge terres similaires à SVM



Régression vectorielle à support ε (ε -SVR)

323

- Soit un ensemble de données $\{x_1, \dots, x_n\}$ avec valeurs cibles $\{u_1, \dots, u_n\}$, on veut réaliser ξ -SVR

- Le problème d'optimisation est

$$\text{Min } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*)$$

subject to

$$\begin{cases} u_i - w^T x_i - b \leq \epsilon + \xi_i \\ w^T x_i + b - u_i \leq \epsilon + \xi_i^* \\ \xi_i \geq 0, \xi_i^* \geq 0 \end{cases}$$

- Formulation similaire à SVM, donc peut être résolu en tant

Régression vectorielle à support ε (ε -SVR)

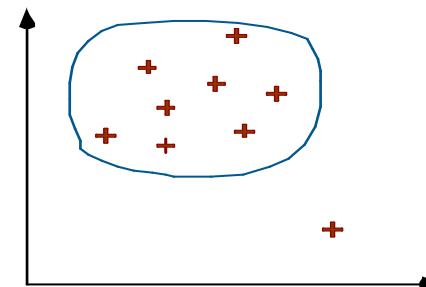
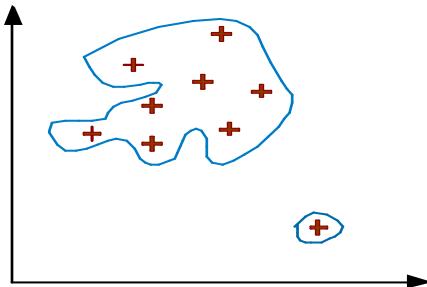
324

- C permet de contrôler l'influence de l'erreur
- Le terme $\frac{1}{2} \|w\|^2$ sert à contrôler la complexité de la fonction de régression
- Après l'apprentissage (solution du problème QP), on trouve les valeurs α_i and α_i^* , qui sont toutes deux zéros si x_i ne contribue pas à la séparation
- Pour calculer $f(z) = \sum_{j=1}^s (\alpha_{t_j} - \alpha_{t_j}^*) K(x_{t_j}, z) + b$

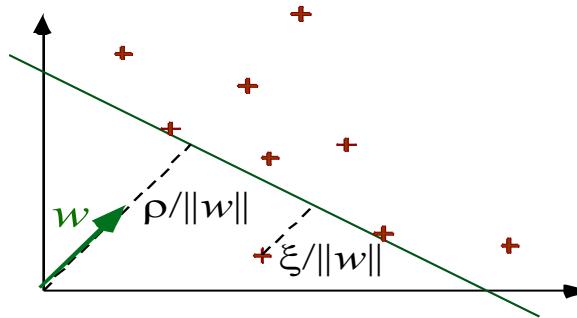
SVM et apprentissage non supervisé

□ Détection de « nouveautés »

325



On cherche à séparer au maximum le nuage de points de l'origine



Pourquoi ça marche ?

La marge est liée à la capacité en généralisation

- Normalement, la classe des hyperplans de \mathbb{R}^d est de $d_H = d + 1$
- Mais la classe des hyperplans de marge
est bornée par :
$$\frac{1}{\|\omega\|} \text{ tq. } \|\omega\|^2 \leq c$$
- où R est le rayon de la plus petite sphère englobant l'échantillon
d'apprentissage S
- ↳ Peut être beaucoup plus petit que la dimension d de l'espace d'entrée X

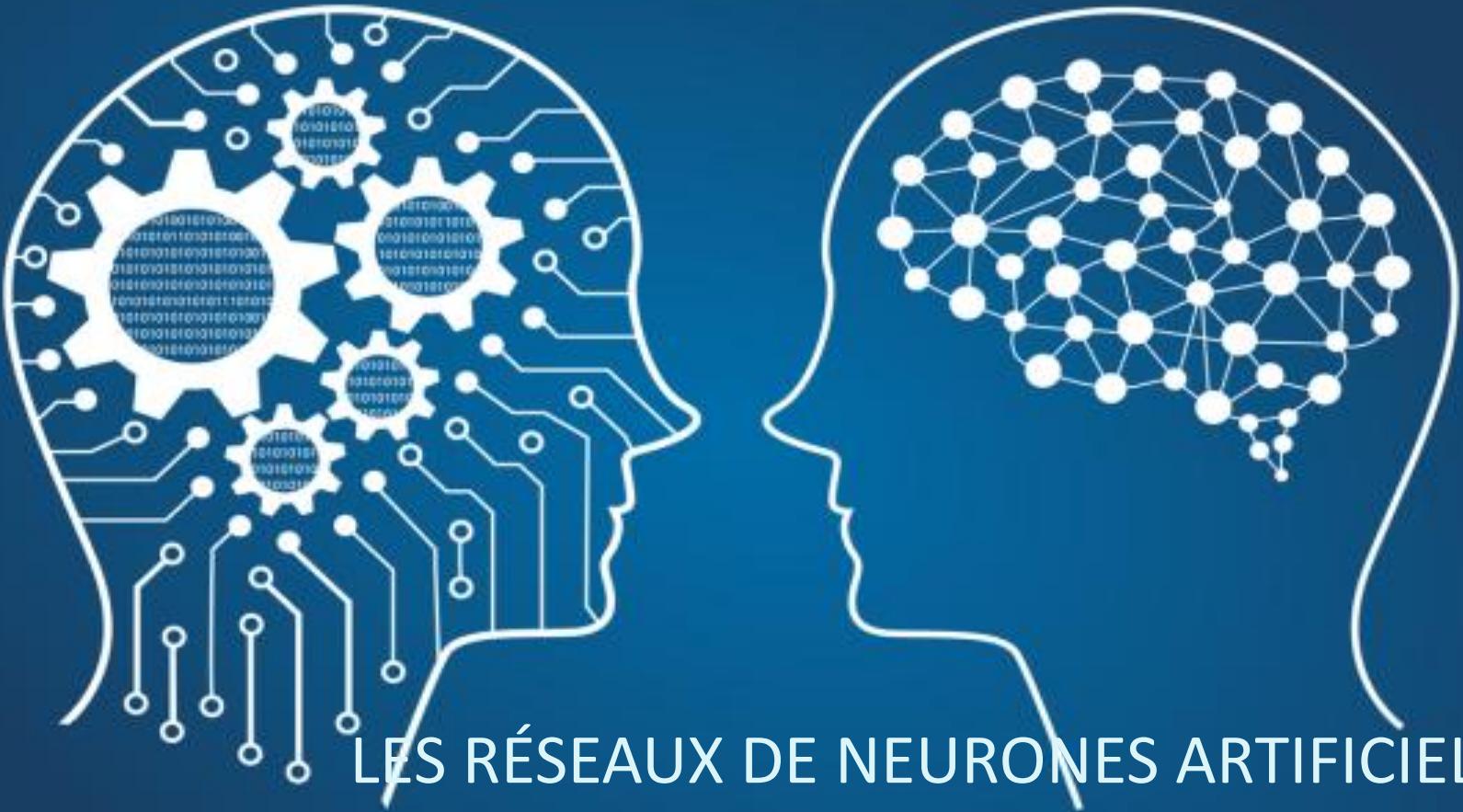
Forces et faiblesses des SVM

□ Forces

- L'apprentissage est relativement facile
 - Pas de minima locaux, comme pour les RNA
- L'algorithme est robuste face aux changements d'échelle
- Le compromis entre la complexité du classifieur et l'erreur de classification peut être gérée explicitement
- Méthode générale
 - Des données non conventionnelles, telles des chaînes et des arbres peuvent servir d'entrées au SVM, à la place des vecteurs de traits
- Résultats en général **équivalents et souvent meilleurs**

□ Faiblesses

- Il faut trouver la “bonne” fonction noyau
- Problèmes i.i.d. (données indépendantes et identiquement distribuées)
- Deux classes à la fois



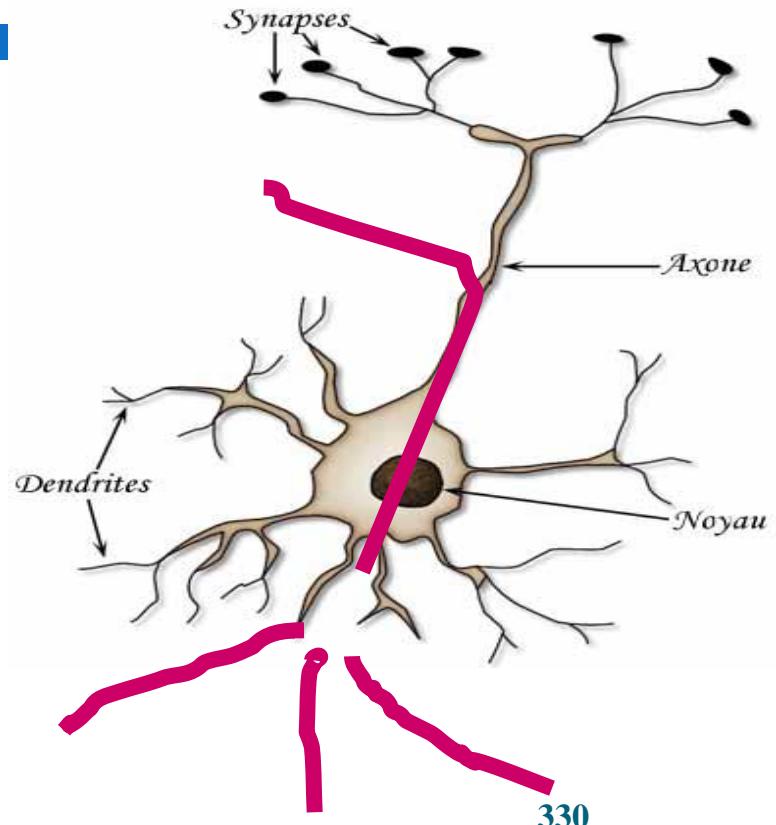
LES RÉSEAUX DE NEURONES ARTIFICIELS

Réseaux de neurones

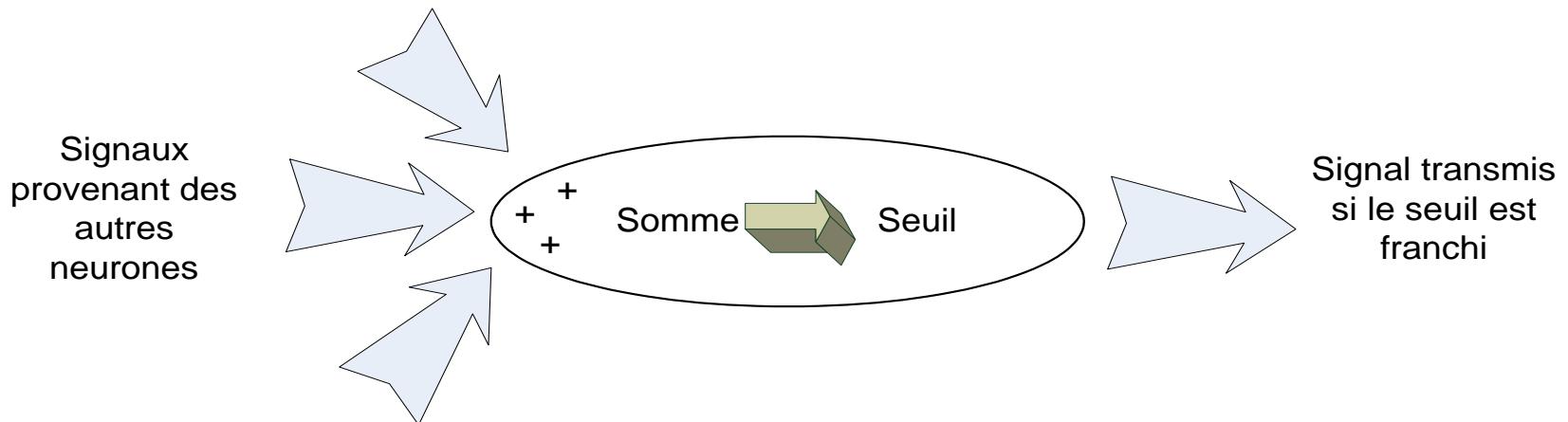
- Tentative de reproduction des structures du cerveau afin de raisonner
- Ensemble d'unités transformant des entrées en sorties (neurones) connectées, où chaque connexion à un poids associé
- La phase d'apprentissage permet d'ajuster les poids pour produire la bonne sortie (la classe en classification)

Analogie avec le cerveau

- Le cerveau humain contient environ 100 milliards de neurones, et chacun est connecté à environ 10.000 autres
- Un neurone reçoit des impulsions électriques de ses voisins via les dendrites. Si la somme des signaux dépasse un certain seuil, il se produit une décharge électrique de type tout ou rien appelée potentiel d'action. Le potentiel d'action se propage le long de l'axone, qui se ramifie en une multitude de dendrites.
- La terminaison d'une dendrite est une petite usine de production chimique. Elle diffuse des neurotransmetteurs chimiques dans un espace appelé synapse, qui rejoint un autre neurone.

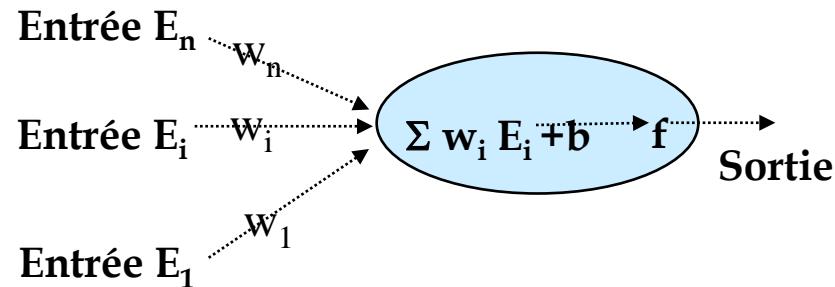


Modélisation du neurone

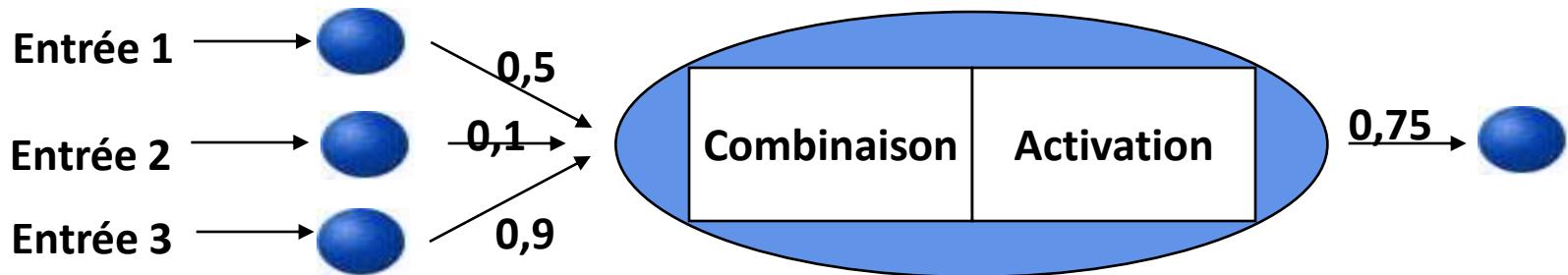


Plus précisément ...

- Induit une valeur en sortie à partir d'un ensemble de valeurs en entrée
- Les liens sont pondérés par des poids
- Réalise une combinaison linéaire des entrées suivie d'une fonction de transfert (fonction à seuil)
 - Fonction Sigma ($\sum w_i E_i$)
 - Biais optionnel b
 - Fonction Sigmoïde $f(\sum) = 1/(1+e^{-\sum})$



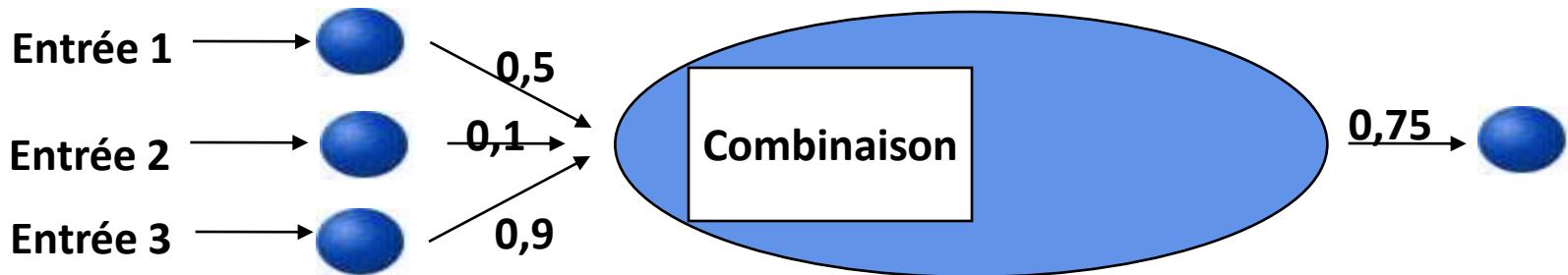
Combinaison/Activation



Phase de combinaison : combine les entrées et produit une valeur en sortie

Phase d'activation : prend en entrée la sortie de la fonction de combinaison et déduit la valeur de sortie

Combinaison



Fonctions de combinaison :

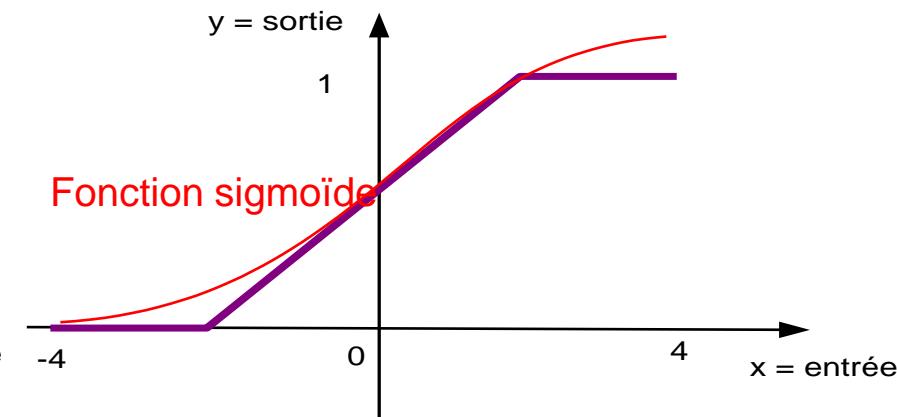
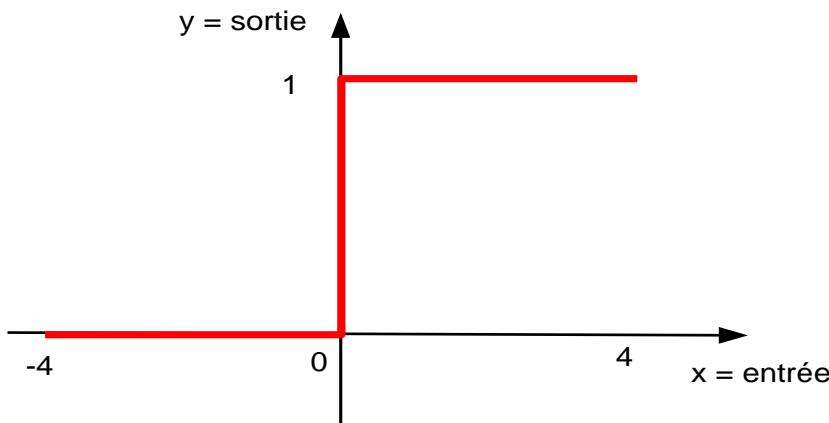
- Produit scalaire
- Norme euclidienne

$$\begin{array}{c} \xrightarrow{\hspace{1cm}} \\ \left(\begin{array}{c} E_1 \\ E_2 \\ E_3 \end{array} \right) \end{array} \bullet \begin{array}{c} \xrightarrow{\hspace{1cm}} \\ \left(\begin{array}{c} 0,5 \\ 0,1 \\ 0,9 \end{array} \right) \end{array} \quad \left\| \left(\begin{array}{c} E_1 \\ E_2 \\ E_3 \end{array} \right) \right\|$$

Activation

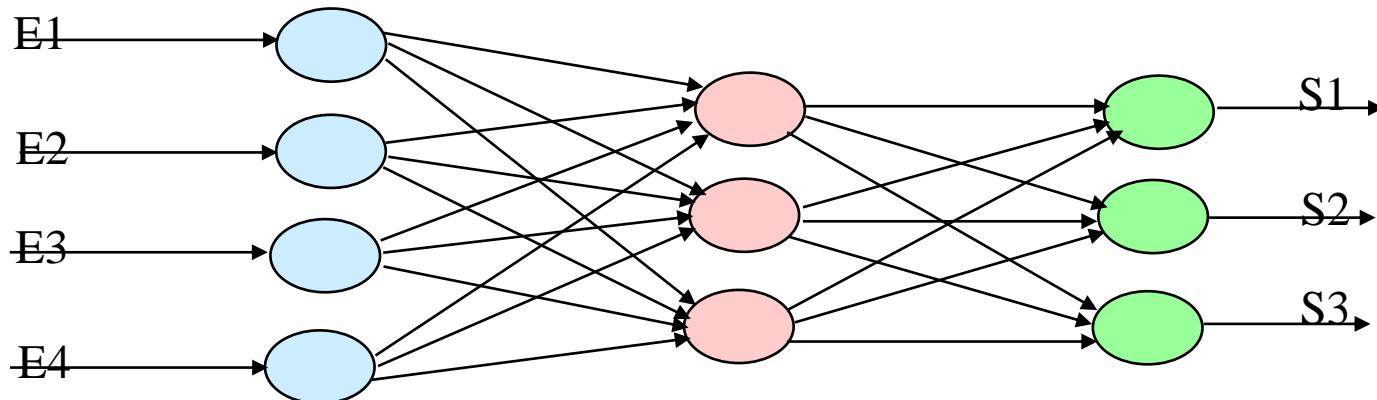
Trois intervalles :

- en dessous du seuil : neurone non actif
- aux alentours du seuil : phase de transition
- au dessus du seuil : neurone actif



Organisation en réseau

- Réseau multi-couches totalement connecté
- Entrées, Calculs (cachés), Sorties

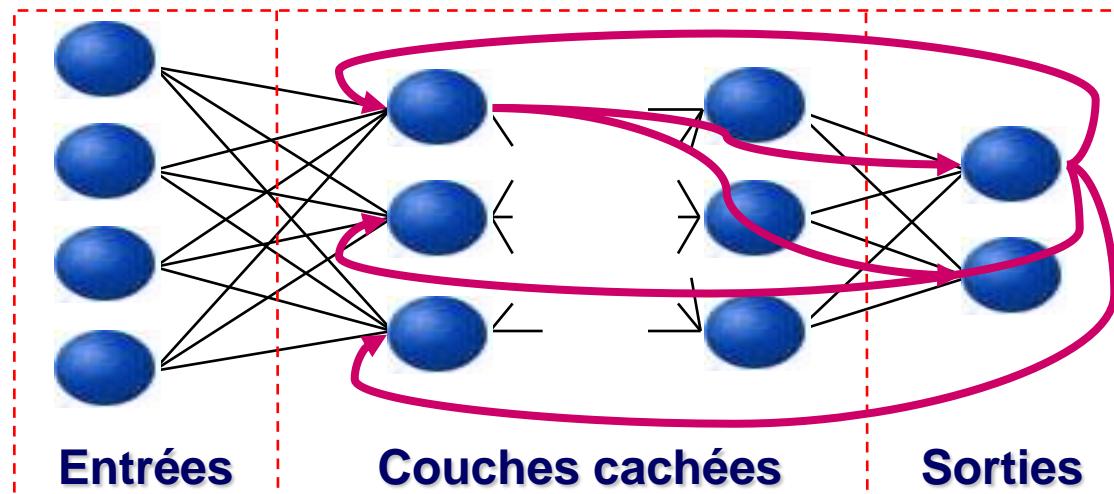


Topologie

Choix du nombre de couches

- entrées, 1 ou 2 couches cachées, sorties
- Choix du nombre de neurones par couche
 - dépend des entrées et sorties
 - couches cachées intermédiaires
- Normalisation des variables d'entrées
 - Variable continue centrée réduite $[-1, +1]$
 - Variable discrète codée ou valeurs attribuées aux entrées
 - Sorties booléenne codant les classes

Perceptron multicouche



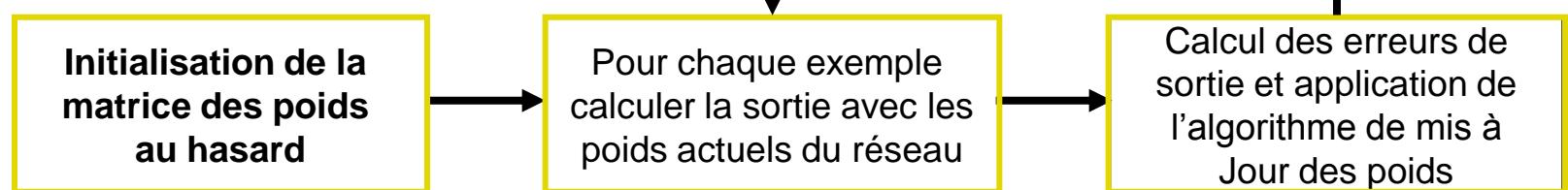
Apprentissage

■ Découverte de modèles complexes avec affinage progressif

- Le réseau s'adapte lors de la phase d'apprentissage
- Plusieurs algorithmes possibles
 - le plus utilisé = rétropropagation
 - modification des poids w_i par rétropropagation

Principe

- Off-Line ou Batch : après tous les exemples
- On-Line ou Stochastique : après chaque exemple



Rétropropagation

Initialiser les poids et les biais

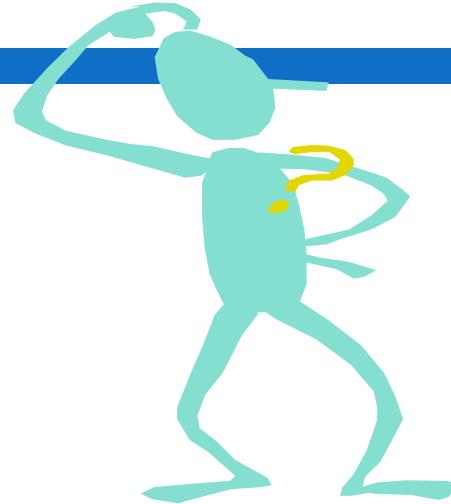
- tirage aléatoire sur [-1,+1]
- Propager les entrées en avant
 - Un exemple est appliqué aux entrées
 - Le réseau calcul les sorties
- Propager les erreurs en arrière
 - Sortie devant délivrer T : $\text{Err} = O(1-O)(T-O)$
 - Cellule cachée : $\text{Err} = O(1-O) \sum_k w_k * \text{Err}_k$
- Corriger poids et biais de sorte à réduire les erreurs
 - $\Delta w_{ij} = \lambda * \text{Err}_j * O_i ; \Delta b_j = \lambda * \text{Err}_j$

Forces et Faiblesses

- Permet d'approcher toute sorte de fonction
- Coûteux en apprentissage
 - ▣ calculs complexes
 - ▣ possibilité d'élaguer le réseau en connexions
 - ▣ peu applicable sur de larges BD
- Effet boite noire
 - ▣ comportement difficile à expliquer
- Autres applications possibles
 - ▣ prédiction, décodage, reconnaissance de formes, etc.

Bilan Classification

- De nombreuses techniques dérivées de l'IA et des statistiques
 - Autres techniques
 - ▣ règles associatives, raisonnement par cas, ensembles flous, ...
 - Problème de passage à l'échelle
 - ▣ arbre de décisions, réseaux
 - Tester plusieurs techniques pour résoudre un problème
-
- Y-a-t-il une technique dominante ?





LÉ CLUSTERING

Qu'est ce qu'un bon regroupement ?

- Une bonne méthode de regroupement permet de garantir
 - Une grande similarité intra-groupe
 - Une faible similarité inter-groupe
- La qualité d'un regroupement dépend donc de la mesure de similarité utilisée par la méthode et de son implémentation

Structures de données

Matrice de données

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

Matrice de similarité

$$\begin{bmatrix} \mathbf{o} & & & & \\ d(2,1) & \mathbf{o} & & & \\ d(3,1) & d(3,2) & \mathbf{o} & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & \mathbf{0} \end{bmatrix}$$

Mesurer la qualité d'un clustering

- Métrique pour la similarité: La similarité est exprimée par le biais d'une mesure de distance
- Une autre fonction est utilisée pour la mesure de la qualité
- Les définitions de distance sont très différentes que les variables soient des intervalles (continues), catégories, booléennes ou ordinales
- En pratique, on utilise souvent une pondération des variables

Types des variables

- Intervalles:
- Binaires:
- catégories, ordinale, ratio:
- Différents types:

Intervalle (discrètes)

□ Standardiser les données

- Calculer l'écart absolu moyen:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf}).$$

où

- Calculer la mesure standardisée (z-score)

Exemple

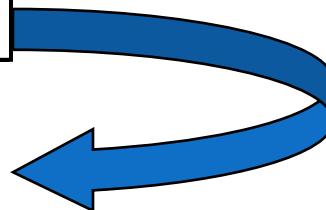
	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

$$M_{Age} = 60$$

$$S_{Age} = 5$$

$$M_{salaire} = 11074$$

$$S_{salaire} = 148$$



	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	2

Similarité entre objets

■ Les distances expriment une similarité

- Ex: la distance de Minkowski :

$$d(i, j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \dots + |x_{i_p} - x_{j_p}|^q)}$$

où $i = (x_{i_1}, x_{i_2}, \dots, x_{i_p})$ et $j = (x_{j_1}, x_{j_2}, \dots, x_{j_p})$ sont deux objets

p -dimensionnels et q un entier positif

- Si $q = 1$, d est la distance de Manhattan

Similarité entre objets(I)

- Si $q = 2$, d est la distance Euclidienne :

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

- Propriétés

- $d(i, j) \geq 0$
- $d(i, i) = 0$
- $d(i, j) \equiv d(j, i)$

Exemple: distance de Manhattan

	Age	Salaire
Personne1	50	11000
Personne2	70	11100
Personne3	60	11122
Personne4	60	11074

$$d(p1, p2) = 120$$

$$d(p1, p3) = 132$$

Conclusion: p1 ressemble plus à p2 qu'à p3 ☹

	Age	Salaire
Personne1	-2	-0,5
Personne2	2	0,175
Personne3	0	0,324
Personne4	0	0

$$d(p1, p2) = 4,675$$

$$d(p1, p3) = 2,324$$

Conclusion: p1 ressemble plus à p3 qu'à p2 ☺

Variables binaires

- Une table de contingence pour données binaires

		Objet j		sum	$a = \text{nombre de positions où } i$ $\text{a } 1 \text{ et } j \text{ a } 1$
		1	0		
Objet i	1	a	b	$a+b$	
	0	c	d	$c+d$	
sum		$a+c$	$b+d$	p	

- Exemple $o_i=(1,1,0,1,0)$ et $o_j=(1,0,0,0,1)$

Mesures de distances

- Coefficient d'appariement (matching) simple (invariant pour variables symétriques):

$$d(i, j) = \frac{b + c}{a + b + c + d}$$

Exemple $\mathbf{o}_i = (1, 1, 0, 1, 0)$ et $\mathbf{o}_j = (1, 0, 0, 0, 1)$

$$d(\mathbf{o}_i, \mathbf{o}_j) = 3/5$$

- Coefficient de Jaccard

$$d(\mathbf{o}_i, \mathbf{o}_j) = 3/4$$

$$d(i, j) = \frac{b + c}{a + b + c}$$

Variables binaires (I)

- Variable symétrique: Ex. le sexe d'une personne, i.e coder masculin par 1 et féminin par 0 c'est pareil que le codage inverse
- Variable asymétrique: Ex. Test HIV. Le test peut être positif ou négatif (0 ou 1) mais il y a une valeur qui sera plus présente que l'autre. Généralement, on code par 1 la modalité la moins fréquente
 - 2 personnes ayant la valeur 1 pour le test sont *plus similaires* que 2 personnes ayant 0 pour le test

Variables binaires(II)

□ Exemple

Nom	Sexe	Fièvre	Toux	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- Sexe est un attribut symétrique
- $d(jack, mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33$
- Les autres attributs sont asymétriques
- Y et P sont ~~pas~~ pas = 0, la distance n'est mesurée que sur les asymétriques

$$d(jim, mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

Les plus similaires sont Jack et Mary \Rightarrow atteints du même mal

Variables Nominales

- Une généralisation des variables binaires, ex: rouge, vert et bleu
- Méthode 1: Matching simple

- m : # d'appariements, p : # total de variables

$$d(i, j) = \frac{P - m}{p}$$

- Méthode 2: utiliser un grand nombre de variables binaires
 - Créer une variable binaire pour chaque modalité (ex: variable rouge qui prend les valeurs vrai ou faux)

Variables Ordinales

■ Une variable ordinaire peut être discrète ou continue

- L'ordre peut être important, ex: classement
- Peuvent être traitées comme les variables intervalles
 - remplacer x_{if} par son rang $r_{if} \in \{1, \dots, M_f\}$
 - Remplacer le rang de chaque variable par une valeur dans [0, 1] en remplaçant la variable f dans l'objet I par

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- Utiliser une distance pour calculer la similarité

En Présence de Variables de différents Types

- Pour chaque type de variables utiliser une mesure adéquate.
Problèmes: les clusters obtenus peuvent être différents
- On utilise une formule pondérée pour faire la combinaison

$$d(i, j) = \frac{\sum_{f=1}^P \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^P \delta_{ij}^{(f)}}$$

- f est binaire ou nominale:

$$d_{ij}^{(f)} = 0 \text{ si } x_{if} = x_{jf}, \text{ sinon } d_{ij}^{(f)} = 1$$

- f est de type intervalle: utiliser une distance normalisée

- f est ordinaire

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- calculer les rangs r_{if} et

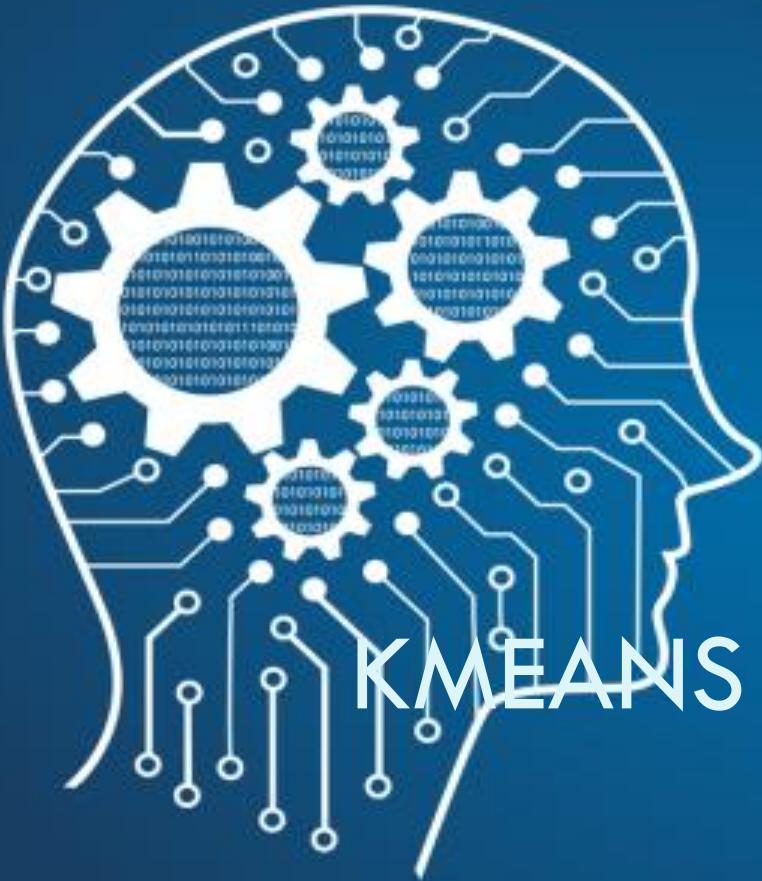
- Ensuite traiter z_{if} comme une variable de type intervalle

Approches de Clustering

- Algorithmes de Partitionnement: Construire plusieurs partitions puis les évaluer selon certains critères
- Algorithmes hiérarchiques: Créer une décomposition hiérarchique des objets selon certains critères
- Algorithmes basés sur la densité: basés sur des notions de connectivité et de densité
- Algorithmes de grille: basés sur un structure à multi-niveaux de granularité
- Algorithmes à modèles: Un modèle est supposé pour chaque cluster ensuite vérifier chaque modèle sur chaque groupe pour choisir le meilleur

Algorithmes à partitionnement

- Construire une partition à k clusters d'une base D de n objets
- Les k clusters doivent optimiser le critère choisi
 - ▣ Global optimal: Considérer toutes les k -partitions
 - ▣ Heuristic methods: Algorithmes k -means et k -medoids
 - ▣ k -means (MacQueen'67): Chaque cluster est représenté par son centre
 - ▣ k -medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Chaque cluster est représenté par un de ses objets



KMEANS



Means)

- L'algorithme *k-means* est en 4 étapes :
 1. Choisir k objets formant ainsi k clusters
 2. (Ré)affecter chaque objet O au cluster C_i de centre M_i tel que $\text{dist}(O, M_i)$ est minimal
 3. Recalculer M_i de chaque cluster (le barycentre)

K-Means :Exemple

- $A = \{1, 2, 3, 6, 7, 8, 13, 15, 17\}$. Créer 3 clusters à partir de A
- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. Ça donne $C_1 = \{1\}$, $M_1 = 1$, $C_2 = \{2\}$, $M_2 = 2$, $C_3 = \{3\}$ et $M_3 = 3$
- Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche. 6 est affecté à C_3 car $\text{dist}(M_3, 6) < \text{dist}(M_2, 6)$ et $\text{dist}(M_3, 6) < \text{dist}(M_1, 6)$

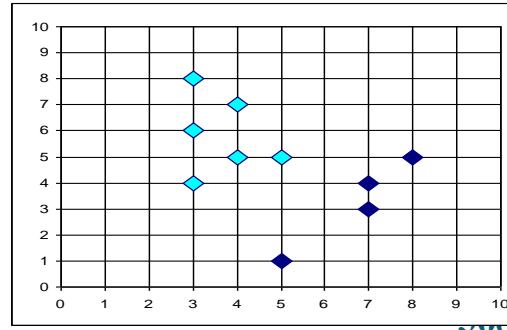
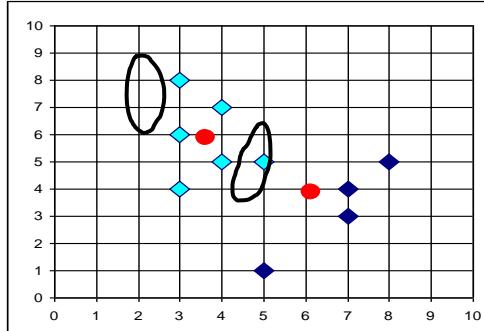
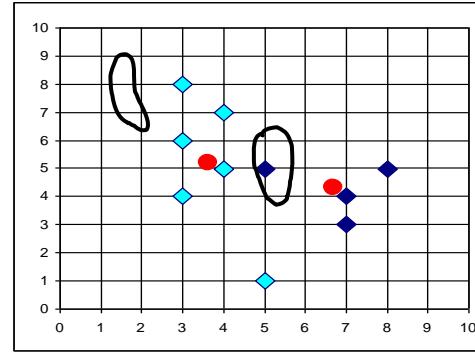
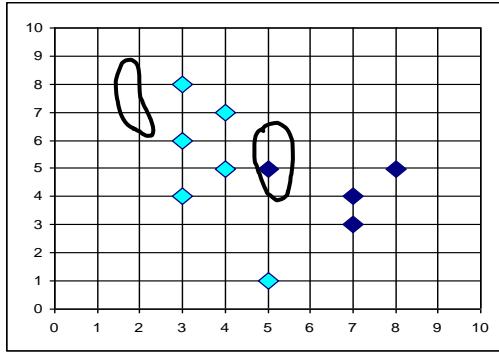
On a $C_1 = \{1\}$, $M_1 = 1$,

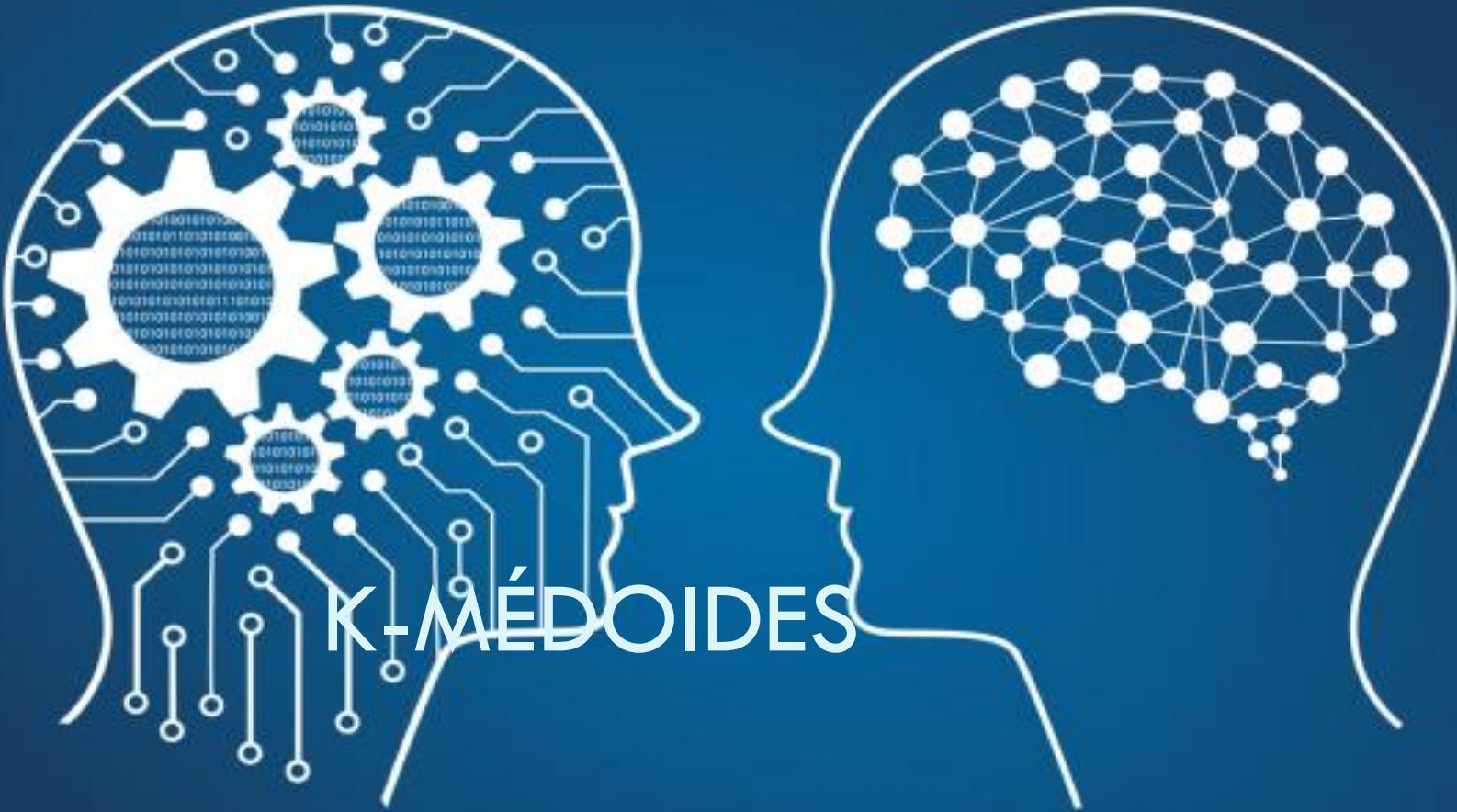
K-Means: Exemple (suite)

- $\text{dist}(3, M_2) < \text{dist}(3, M_3) \rightarrow 3$ passe dans C_2 . Tous les autres objets ne bougent pas.
 $C_1 = \{1\}$, $M_1 = 1$, $C_2 = \{2, 3\}$, $M_2 = 2.5$, $C_3 = \{6, 7, 8, 13, 15, 17\}$ et $M_3 = 66/6 = 11$
- $\text{dist}(6, M_2) < \text{dist}(6, M_3) \rightarrow 6$ passe dans C_2 . Tous les autres objets ne bougent pas.
 $C_1 = \{1\}$, $M_1 = 1$, $C_2 = \{2, 3, 6\}$, $M_2 = 11/3 = 3.67$, $C_3 = \{7, 8, 13, 15, 17\}$, $M_3 = 12$
- $\text{dist}(2, M_1) < \text{dist}(2, M_2) \rightarrow 2$ passe en C_1 . $\text{dist}(7, M_2) < \text{dist}(7, M_3) \rightarrow 7$ passe en C_2 . Les autres ne bougent pas. $C_1 = \{1, 2\}$, $M_1 = 1.5$, $C_2 = \{3, 6, 7\}$, $M_2 = 5.34$, $C_3 = \{8, 13, 15, 17\}$, $M_3 = 13.25$
- $\text{dist}(3, M_1) < \text{dist}(3, M_2) \rightarrow 3$ passe en 1. $\text{dist}(8, M_2) < \text{dist}(8, M_3) \rightarrow 8$ passe en 2 367

Algorithme K-Means

Exemple





K-MÉDOIDES

La méthode des K-Medoids (PAM)

- Trouver des objets représentatifs (medoïdes) dans les clusters (au lieu de la moyenne)
- Principe
 - Commencer avec un ensemble de medoïdes puis itérativement remplacer un par un autre si ça permet de réduire la distance globale
 - Efficace pour des données de petite taille

Algorithme des k-Medoides

Choisir arbitrairement k medoides

Répéter

affecter chaque objet restant au medoide le plus proche

Choisir aléatoirement un non-medoide O_r

Pour chaque medoide O_i

 Calculer le coût TC du remplacement de O_i par O_r

 Si $TC < 0$ alors

 Remplacer O_i par O_r

 Calculer les nouveaux clusters

 Finsi

FinPour

Jusqu'à ce ce qu'il n'y ait plus de changement

PAM (Partitioning Around Medoids) (1987)

Choisir arbitrairement k objets représentatifs

- Pour toute paire (h,i) d'objets t.q h est choisi et i non, calculer le coût TC_{ih} du remplacement de i par h
 - Si $TC_{ih} < 0$, i est remplacé par h
 - Puis affecter chaque objet non sélectionné au medoïde qui lui est le plus similaire
- Répéter jusqu'à ne plus avoir de changements

La méthode des K-Medoids

TC_{jh} représente le gain en distance globale que l'on va avoir en remplaçant h par j

- Si TC_{jh} est négatif alors on va perdre en distance. Ca veut dire que les clusters seront plus compacts.
- $TC_{jh} = \sum_i dist(j,h) - dist(j,i) = \sum_i C_{ijh}$

La méthode des K-Medoids: Exemple

Soit $A = \{1, 3, 4, 5, 8, 9\}$, $k=2$ et $M = \{1, 8\}$ ensemble des medoides

$\rightarrow C_1 = \{1, 3, 4\}$ et $C_2 = \{5, 8, 9\}$

$$E_{\{1,8\}} = \text{dist}(3,1)^2 + \text{dist}(4,1)^2 + \text{dist}(5,8)^2 + \text{dist}(5,9)^2 + \text{dist}(9,8)^2 = 39$$

Comparons 1 et 3 $\rightarrow M = \{3, 8\} \rightarrow C_1 = \{1, 3, 4, 5\}$ et $C_2 = \{8, 9\}$

$$E_{\{3,8\}} = \text{dist}(1,3)^2 + \text{dist}(4,3)^2 + \text{dist}(5,3)^2 + \text{dist}(9,8)^2 = 10$$

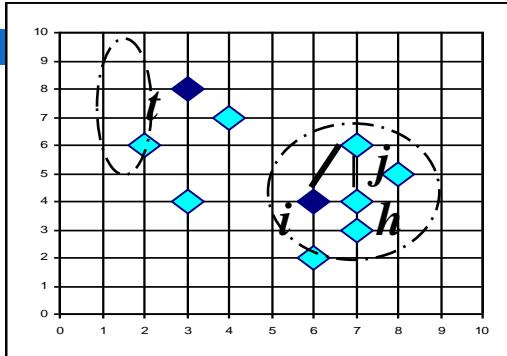
$E_{\{3,8\}} - E_{\{1,8\}} = -29 < 0$ donc le remplacement est fait.

Comparons 3 et 4 $\rightarrow M = \{4, 8\} \rightarrow C_1$ et C_2 inchangés et

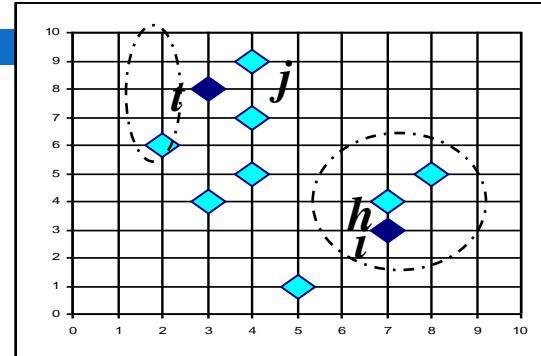
$$E_{\{4,8\}} = \text{dist}(1,4)^2 + \text{dist}(3,4)^2 + \text{dist}(5,4)^2 + \text{dist}(8,9)^2 = 12 \rightarrow 3$$
 n'est pas remplacé par 4

Comparons 3 et 5 $\rightarrow M = \{5, 8\} \rightarrow C_1$ et C_2 inchangés et $E\{5,8\} > E\{3,8\}$

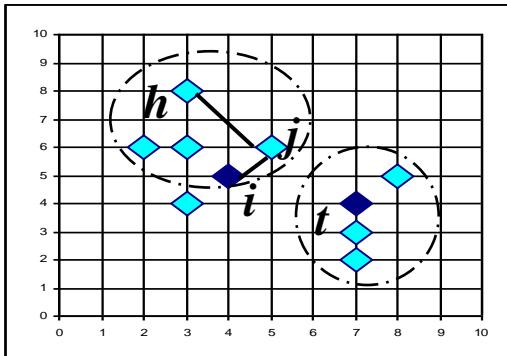
PAM Clustering: $TC_{ih} = \sum_i C_{jih}$



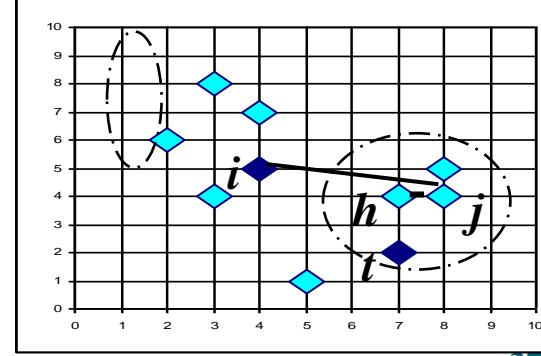
$$C_{jih} = d(j, h) - d(j, i)$$



$$C_{jih} = 0$$



$$C_{jih} = d(j, t) - d(j, i)$$



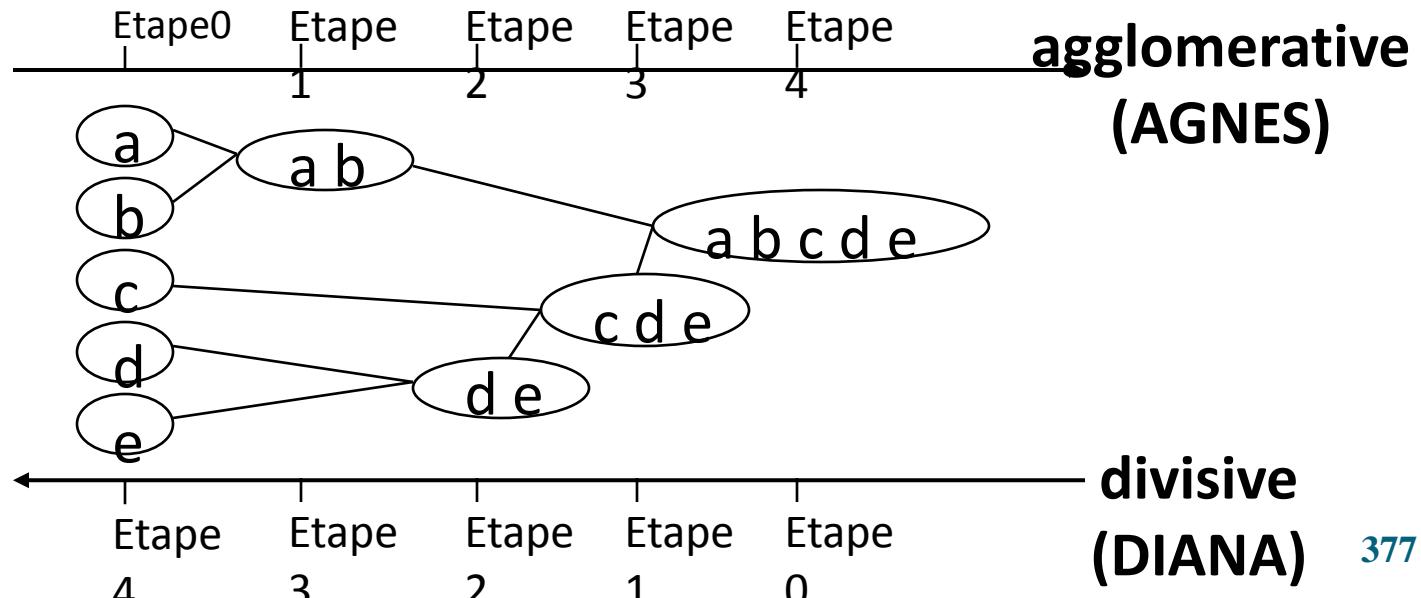
$$C_{jih} = d(j, h) - d(j, t)$$
375



CLASSIFICATION HIÉRARCHIQUE

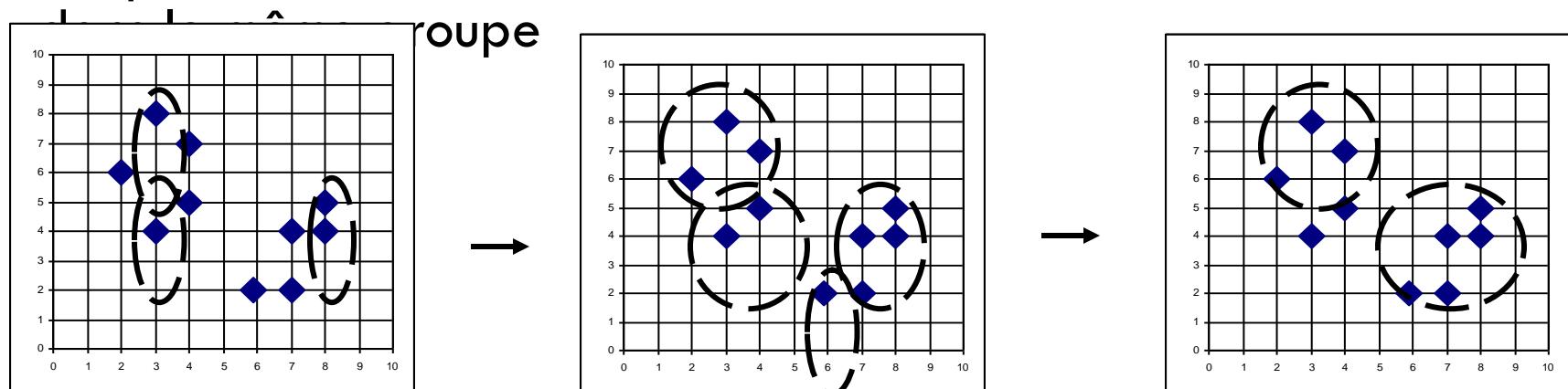
Clustering Hiérarchique

- Utiliser la matrice de distances comme critère de regroupement.
 k n'a pas à être précisé, mais a besoin d'une condition d'arrêt



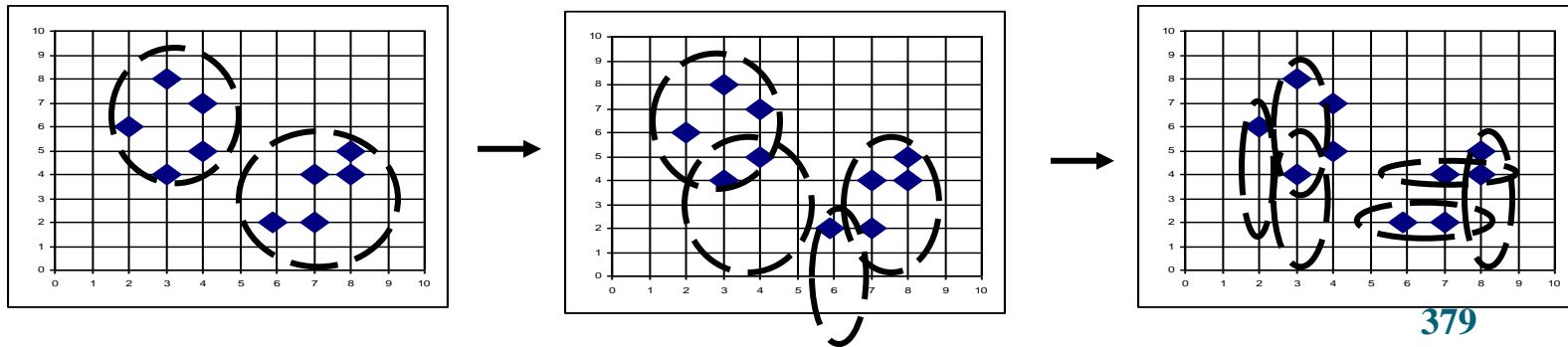
AGNES (Agglomerative Nesting)

- Utilise la matrice de dissimilarité.
- Fusionne les nœuds qui ont la plus faible dissimilarité
- On peut se retrouver dans la situation où tous les nœuds sont



DIANA (Divisive Analysis)

- L'ordre inverse de celui d'AGNES
- Il se peut que chaque objet forme à lui seul un



Critères de fusion-éclatement

- Exemple: pour les méthodes agglomératives, C1 et C2 sont fusionnés si
 - il existe $o_1 \in C_1$ et $o_2 \in C_2$ tels que $\text{dist}(o_1, o_2) \leq \text{seuil}$, ou $C_1 \cup C_2 = \frac{n_1 * n_2}{\sum_{o_1 \in C_1, o_2 \in C_2} \text{dist}(o_1, o_2)}$
 - il n'existe pas $o_1 \in C_1$ et $o_2 \in C_2$ tels que $\text{dist}(o_1, o_2) \geq \text{seuil}$, ou
 - distance entre C_1 et $C_2 \leq \text{seuil}$ avec

BIRCH (1996)

Birch: Balanced Iterative Reducing and Clustering using Hierarchies

- Construit incrémentalement un arbre (CF-tree : Clustering Feature), une structure hiérarchique où chaque niveau représente une phase de clustering
- Phase 1: scanner la base pour construire le CF-tree dans la mémoire
- Phase 2: utiliser n'importe quel algorithme de clustering sur les feuilles du CF-tree
- Avantage: trouve les clusters en une seule passe sur la BD 381

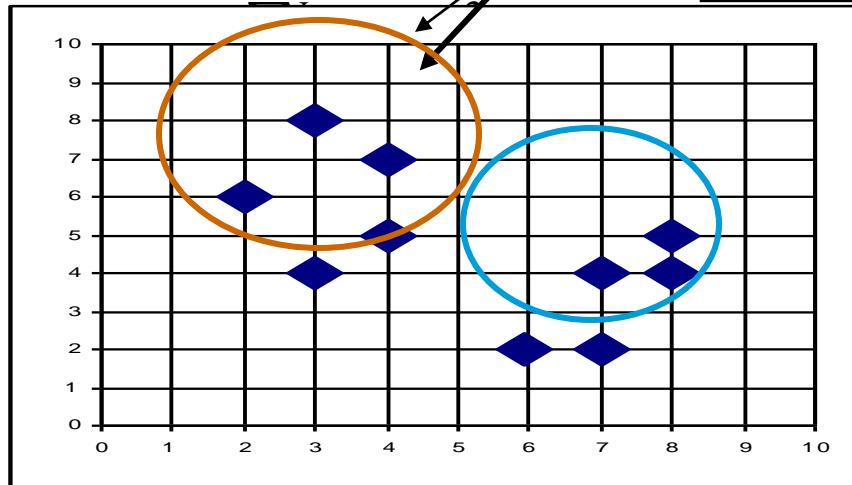
Clustering Feature Vector

Clustering Feature: $CF = (\overrightarrow{N}, \overrightarrow{LS}, SS)$

N: Number of data points

$$LS: \sum_{i=1}^N = \vec{X}_i$$

$$CF = (5, (16,30), (54,190))$$



(3,4)

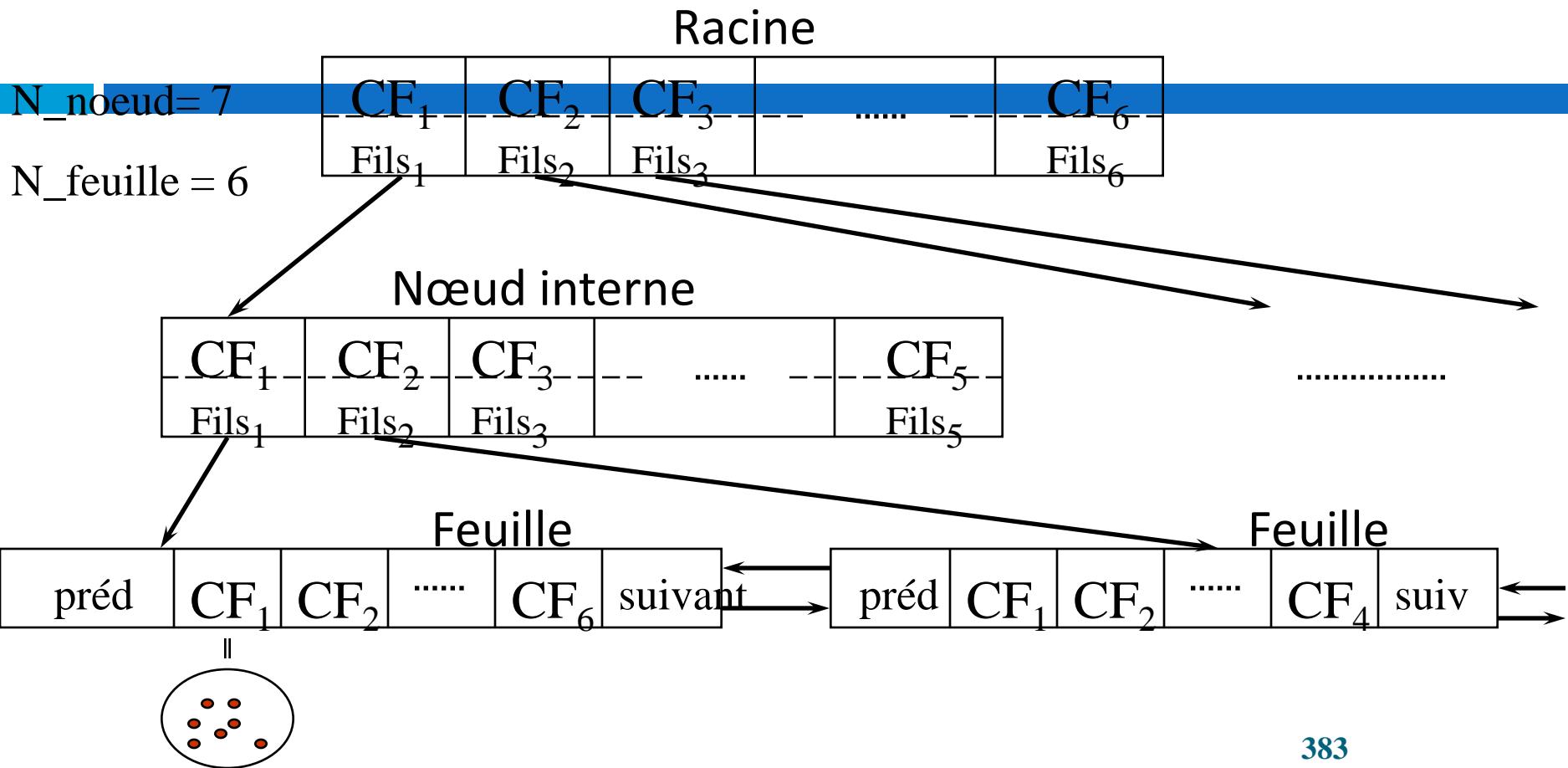
(2,6)

(4,5)

(4,7)

(3,8) 382

CF Tree



CURE (Clustering Using REpresentatives)



- Les méthodes précédentes donnent les groupes (b)
- CURE: (1998)
 - Arrête la création de clusters dès qu'on en a k
 - Utilise plusieurs points représentatifs clusters

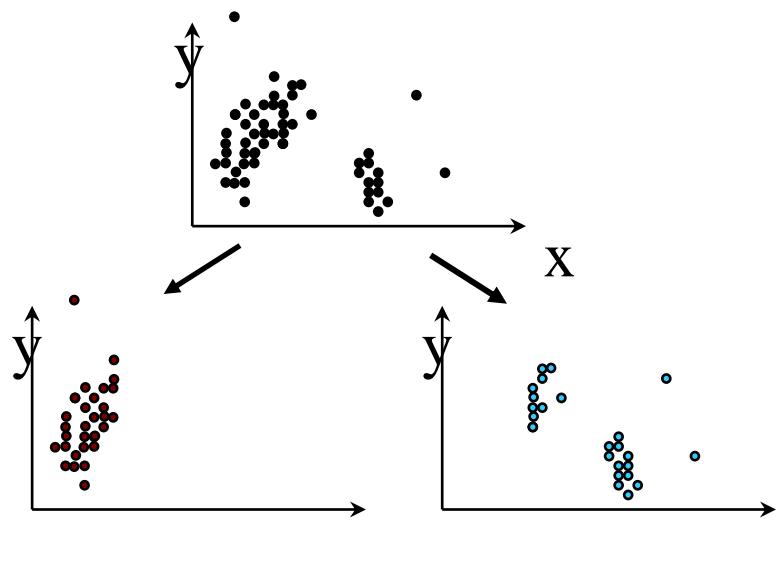
Cure: l'algorithme

- Prendre un sous-ensemble s
- Partitionner s en p partitions de taille s/p
- Dans chaque partition, créer s/pq clusters
- Eliminer les exceptions (points aberrants)
- Regrouper les clusters partiels

Partitionnement et Clustering

$s = 50$

- $p = 2$
- $s/p = 25$



■ $s/pq = 5$

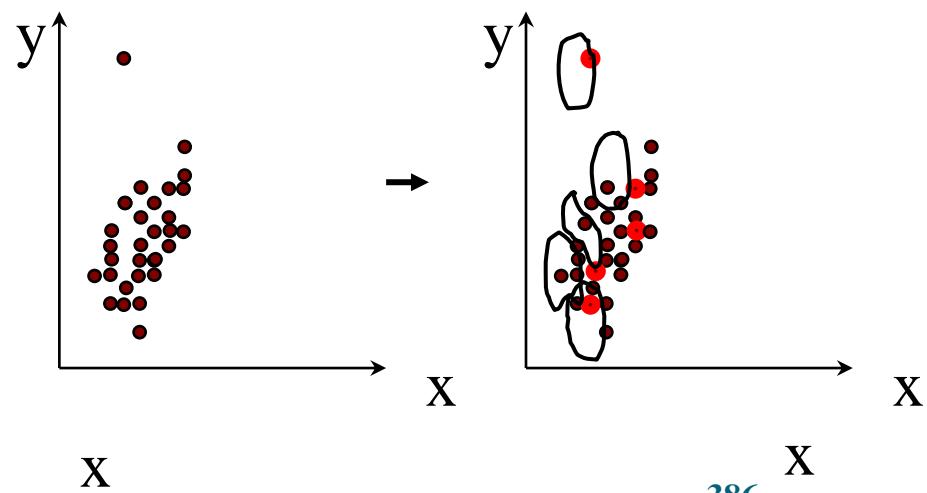
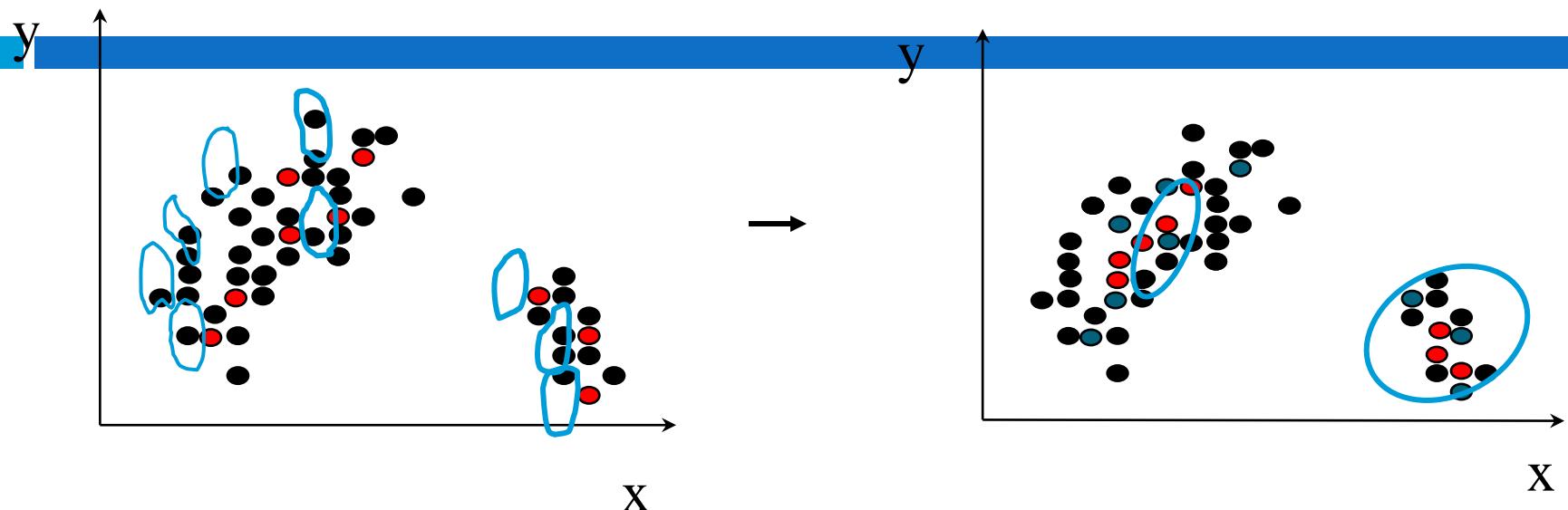


Figure: Rapprochement des points représentatifs



- Rapprocher les points représentatifs vers le centre de gravité par un facteur α .
- Plusieurs points représentatifs permettent de figurer la forme du cluster

Clustering de données Catégorielles : ROCK

- ROCK: Robust Clustering using links
 - Utilise les liens pour mesurer la similarité/proximité
 - N'est pas basé sur la notion de distance
- Idée :
 - Fonction de similarité et voisins:

Let $T_1 = \{1, 2, 3\}$, $T_2 = \{3, 4, 5\}$

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|} = \frac{1}{5} = 0.2$$

Rock

- Considérons 4 transactions et 6 produits t.q

$$T1=\{1,2,3,5\} \quad T2=\{2,3,4,5\}$$

$$T3=\{1,4\} \text{ et } T4=\{6\}$$

- $T1$ peut être représentée par $\{1,1,1,0,1,0\}$

$\text{dist}(T1, T2)=2$ qui est la plus petite distance entre 2 transactions $\rightarrow T1$ et $T2$ dans même cluster. La moyenne de $C1=(0.5,1,1,0.5,1,0)$.

$C2=\{T3, T4\}$ car $\text{dist}(T3, T4)=3$. Or $T3$ et $T4$ n'ont aucun produit en commun !

Idée : se baser sur le nombre d'éléments en commun

Ce n'est pas suffisant $\{1,2\}$ est plus proche de $\{1,2,3\}$ que de $\{1,2,3,4,5,6\}$

Rock: l'algorithme

- Liens: Le nombre de voisins communs de 2 points

$\{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}$
 $\{1,4,5\}, \{2,3,4\}, \{2,3,5\}, \{2,4,5\}, \{3,4,5\}$

$\{1,2,3\} \xleftarrow[3]{\quad} \{1,2,4\}$

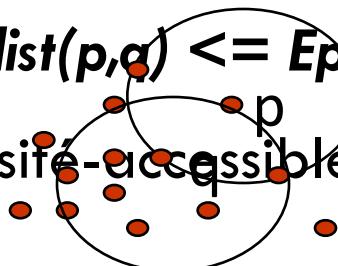
- Algorithme

- Prendre un sous ensemble

- Regrouper avec les liens

Clustering basé sur la densité

- Voit les clusters comme des régions denses séparées par des régions qui le sont moins (bruit)
- Deux paramètres:
 - **Eps:** Rayon maximum du voisinage
 - **MinPts:** Nombre minimum de points dans le voisinage-Eps d'un point
- **Voisinage :** $V_{Eps}(p) : \{q \in D \mid dist(p,q) \leq Eps\}$
- Un point p est directement densité-accessible à partir de q resp. à $Eps, MinPts$ si
- 1) $p \in V_{Eps}(q)$



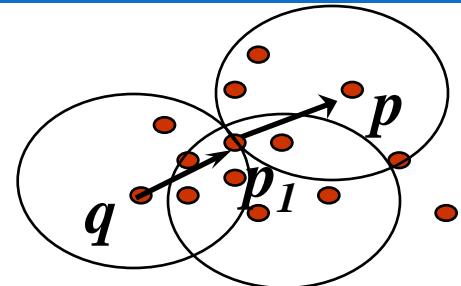
MinPts = 5

Eps = 1 cm

Clustering basé sur la densité

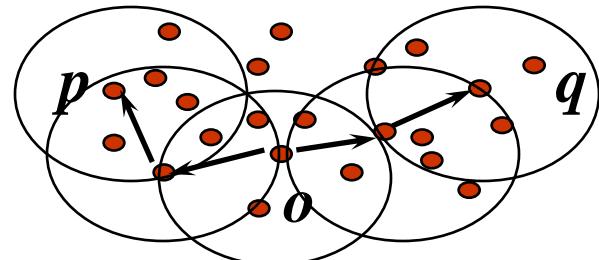
□ Accessibilité:

- p est accessible à partir de q resp. à $Eps, MinPts$ si il existe p_1, \dots, p_n , $p_1 = q$, $p_n = p$ t.q p_{i+1} est directement densité accessible à partir de p_i



□ Connexité

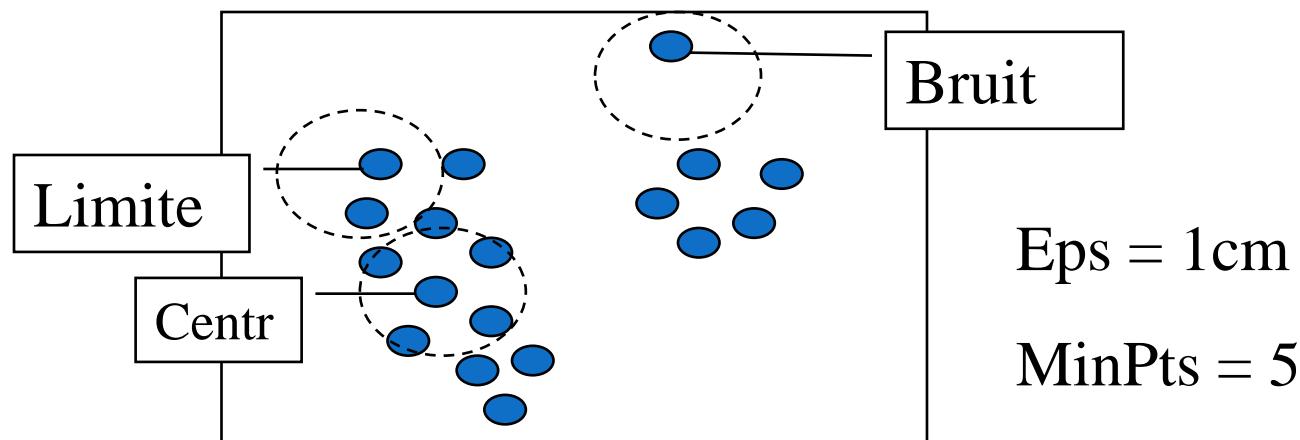
- p est connecté à q resp. à $Eps, MinPts$ si il existe un point o t.q p et q accessibles à partir de o resp. à Eps et $MinPts$.





DBSCAN: Density Based Spatial Clustering of Applications with Noise

- Un *cluster* est l'ensemble maximal de points connectés
- Découvre des clusters non nécessairement convexes



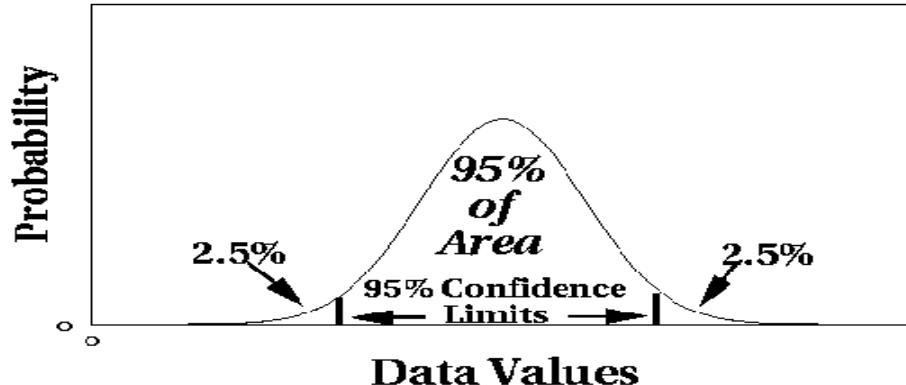
DBSCAN: l'algorithme

- Choisir p
- Récupérer tous les points accessibles à partir de p resp. Eps et $MinPts$.
- Si p est un centre, un cluster est formé.
- si p est une limite, alors il n'y a pas de points accessibles de p : passer à un autre point
- Répéter le processus jusqu'à épuiser tous les points. 395

Découverte d'exceptions

- Ce sont les objets qui sont considérablement différents du reste, exemple: ornithorynque, kiwi
- Problème
 - Trouver n objets qui sont les plus éloignés du reste
- Applications:
 - fraude
 - Analyse médicale
 - ...

Approche statistique



- On suppose que les données suivent une loi de distribution statistique (ex: loi normale)
- Utiliser les tests de discordance
 - $\text{Proba}(X_i=\text{val}) < \beta$ alors X est une exception
- Problèmes

Distance

- Une (α, β) -exception est un object O dans T tel qu'il y a au moins α objets O' de T avec $\text{dist}(O,O') > \beta$



Thank you

BEN LAHMAR EL Habib