

APPRENTISSAGE SUPERVISÉ SVM

INTRODUCTION

L'arrivée des **SVMs** en **1992** marque un tournant dans l'histoire du **machine learning**.

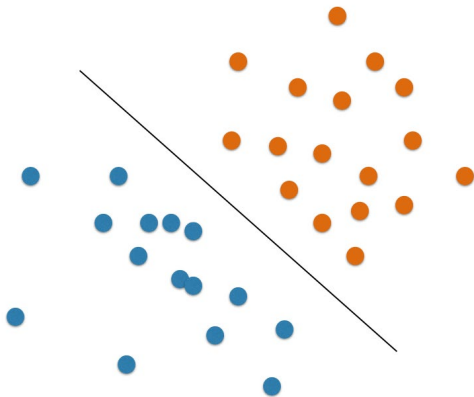
La théorie de laquelle ces méthodes sont issues est alors nouvelle, intuitive et permet de résoudre des problèmes complexes dans un **nouveau paradigme**.

Les notions mathématiques à la base des SVMs sont difficiles. Nous n'aborderons dans ce cours que les concepts.

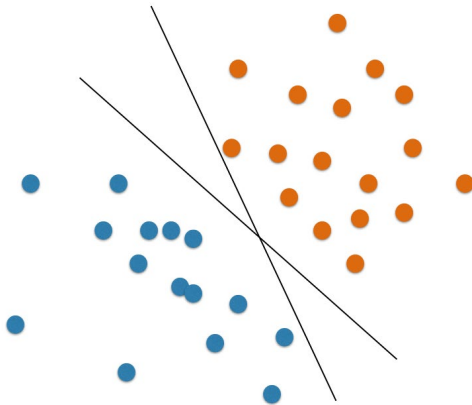
CLASSIFIEURS À VASTE MARGE : DONNÉES LINÉAIREMENT SÉPARABLES



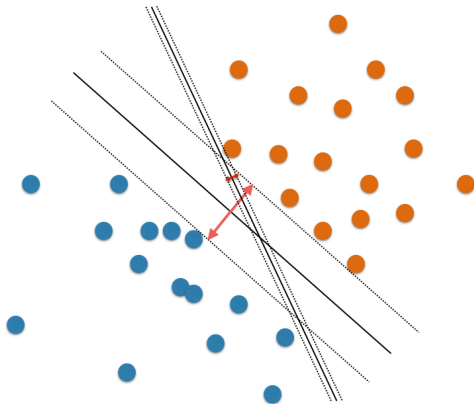
CLASSIFIEURS À VASTE MARGE: DONNÉES LINÉAIREMENT SÉPARABLES



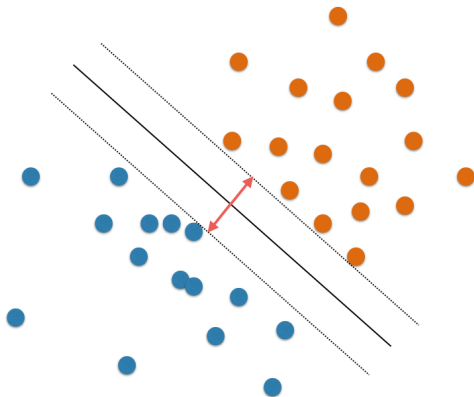
CLASSIFIEURS À VASTE MARGE: DONNÉES LINÉAIREMENT SÉPARABLES



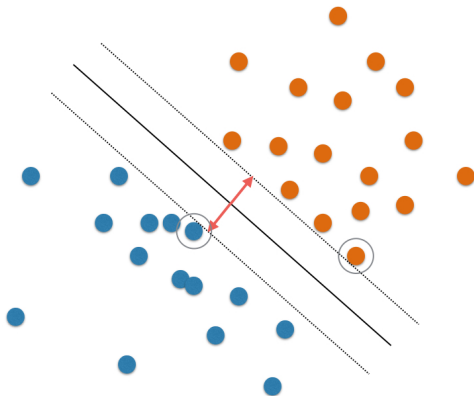
CLASSIFIEURS À VASTE MARGE: DONNÉES LINÉAIREMENT SÉPARABLES



CLASSIFIEURS À VASTE MARGE: DONNÉES LINÉAIREMENT SÉPARABLES

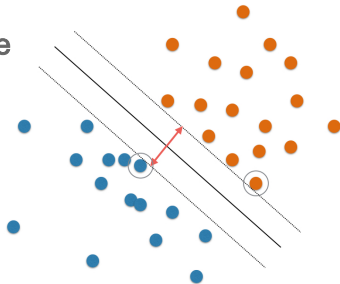


CLASSIFIEURS À VASTE MARGE: DONNÉES LINÉAIREMENT SÉPARABLES



CLASSIFIEURS À VASTE MARGE: DONNÉES LINÉAIREMENT SÉPARABLES

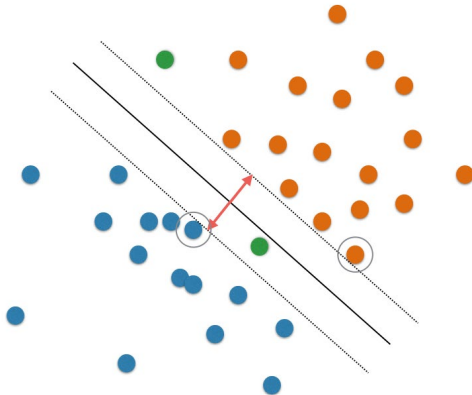
- Il existe une **infinité** de droites séparant les points.
- On cherche la **séparation linéaire** qui donne la plus grande **marge**.
- Les "bords" de cette marge s'appuient sur des **vecteurs supports**.
- On trouve cette séparation en résolvant un problème d'**optimisation convexe**.



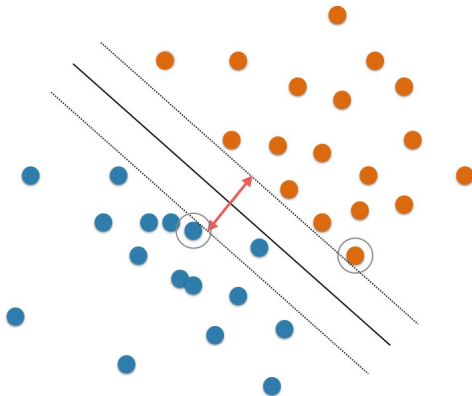
INTUITION

- Les **vecteurs supports** sont les points les plus **ambigus** et donc les plus **difficiles à classifier**.
- Ce sont ces points qui influencent le choix de la **meilleure droite** : si ces points changent, la droite change.
- En quelque sorte, on se met dans un "worst-case" scenario : puisque l'on sait classifier les points les plus ambigus, le système devrait être **robuste**.
- Plus la marge est grande, plus l'on est sûrs de nous.

PHASE DE TEST



PHASE DE TEST



PROBLÈME

Les données sont rarement aussi simples !

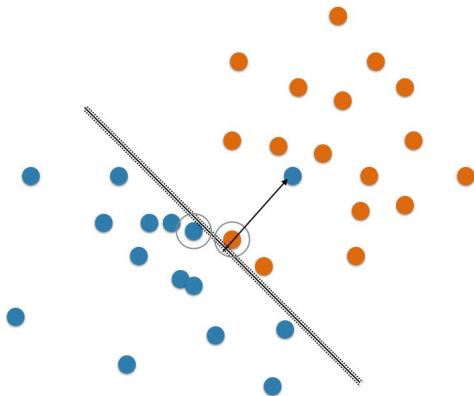
CLASSIFIEURS À VASTE MARGE: CAS NON LINÉAIREMENT SÉPARABLE



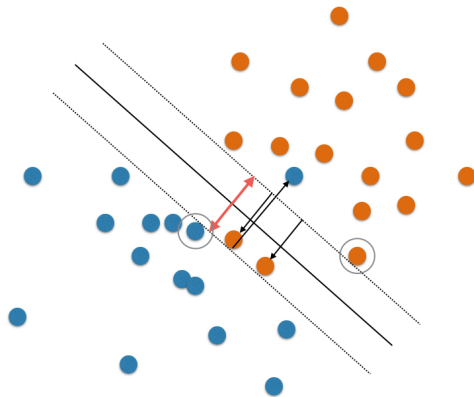
CLASSIFIEURS À VASTE MARGE: CAS NON LINÉAIREMENT SÉPARABLE

- On **autorise** des erreurs de classification sur l'**ensemble d'entraînement**. Il sera moins sensible aux **outliers (= exceptions)** et donc plus robuste.
- Mais jusqu'à quel point autoriser ces erreurs ?

CLASSIFIEURS À VASTE MARGE: CAS NON LINÉAIREMENT SÉPARABLE

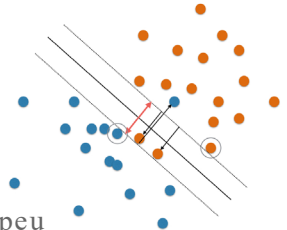


CLASSIFIEURS À VASTE MARGE: CAS NON LINÉAIREMENT SÉPARABLE

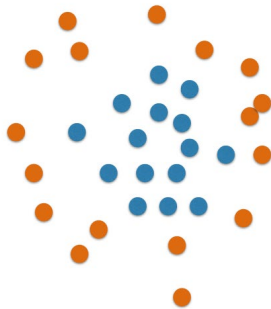


CLASSIFIEURS À VASTE MARGE: CAS NON LINÉAIREMENT SÉPARABLE

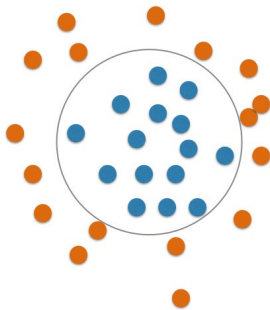
- C'est un **paramètre de régularisation** qui décide à quel point on peut faire des erreurs. Plus il est petit, plus les erreurs sont autorisées et plus la marge est large.
- On préfère des erreurs sur l'ensemble d'entraînement plutôt que du **sur-apprentissage**.
- Il faut trouver le bon **compromis** entre peu d'erreurs et une marge importante.
- Le paramètre de régularisation est optimisé en **validation croisée**.



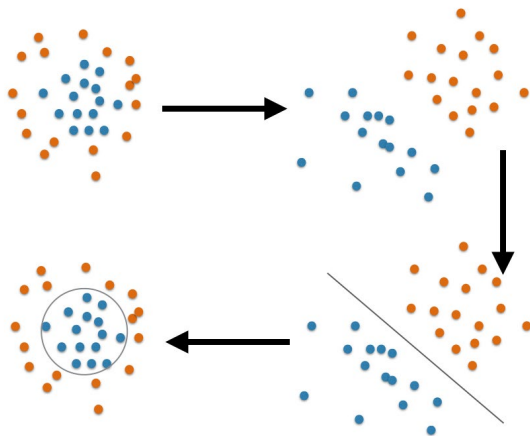
ET SI LA SÉPARATION LINÉAIRE N'EST PAS DU TOUT ADAPTÉE ?



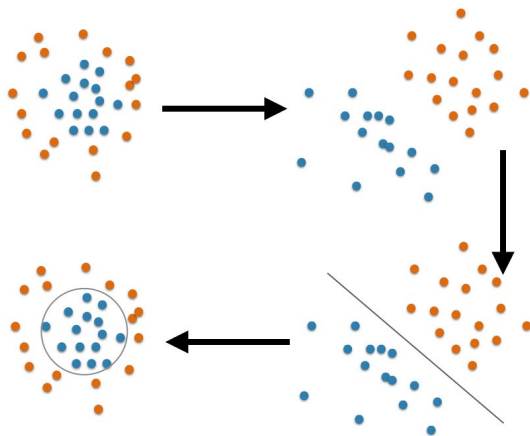
ET SI LA SÉPARATION LINÉAIRE N'EST PAS DU TOUT ADAPTÉE ?



ET SI LA SÉPARATION LINÉAIRE N'EST PAS DU TOUT ADAPTÉE ?



ET SI LA SÉPARATION LINÉAIRE N'EST PAS DU TOUT ADAPTÉE ?



ET SI LA SÉPARATION LINÉAIRE N'EST PAS DU TOUT ADAPTÉE ?

- L'idée est donc d'appliquer une **transformation** aux données pour se trouver dans un cas où le problème redevient linéaire.
- Cette transformation passe par ce qu'on appelle un **noyau**.
- Un noyau est une mesure de **similarité** souvent notée K (kernel) qui mesure à quel point deux vecteurs sont proches.
- En réalité, on n'applique pas la transformation à chaque vecteur directement mais à leur similarité deux à deux.
- Il existe différents types de noyaux. Il faut choisir celui qui est le plus adapté aux données.

Démo

<http://cs.stanford.edu/people/karpathy/svmjs/demo/>

COMMENT DÉFINIR LE BON NOYAU ?

Il existe des noyaux “offthe shelf”, tous prêts à être utilisés. On peut en essayer différents pour voir ce qui fonctionne le mieux. Mais ce n'est pas l'utilisation optimale de ce modèle.

Pour entraîner un SVM, on a “juste” besoin du noyau. On n'a pas besoin de savoir représenter les objets eux-mêmes.

Donc on utilise un SVM en particulier quand on sait définir la similarité entre deux objets (similarité exprimée dans le noyau), mais qu'on n'a pas de “features” pour les objets.

QUELQUES NOYAUX

- Le noyau **linéaire**

$$K(x_i, x_j) = x_{i,1}x_{j,1} + x_{i,2}x_{j,2} + \dots + x_{i,p}x_{j,p}$$

- Le noyau **RBF**

$$K(x_i, x_j) = e^{-\gamma(x_{i,1}-x_{j,1})^2 + (x_{i,2}-x_{j,2})^2 + \dots + (x_{i,p}-x_{j,p})^2}$$

- Le noyau **polynomial**

$$K(x_i, x_j) = (x_{i,1}x_{j,1} + x_{i,2}x_{j,2} + \dots + x_{i,p}x_{j,p} + c)^p$$

- Et surtout : le noyau customisé ! Sans même savoir décrire les objets, il suffit de savoir décrire leur similarité par un noyau pour faire fonctionner un SVM.

Les noyaux **déforment** donc les distances entre les points et changent par conséquent la forme du problème.

CONSIDERATIONS FONCTIONNELLES

- Fonctionnellement un réseau SVM et un PMC avec une couche cachée
- Spécifiquement, le noyau joue le rôle de fonction d'activation
- Les vecteurs de supports jouent le rôle des poids entre la couche d'entrée et la couche cachée.
- Le nombre de vecteurs de support définit le nombre de neurones dans la couche cachée
- Il existe un noyau «tangente hyperbolique» et donc les SVM sont très proches des PMC standards

EXEMPLE PERSONNALISE

SVM pour classifier des protéines. On peut comparer deux séquences comme on comparerait 2 chaînes de caractères:

Protéine 1: AGGGCTTACTA

Protéine 2: GGGCTACTAGGGGCC

EXEMPLE PERSONNALISE

SVM pour classifier des protéines. On peut comparer deux séquences comme on comparerait 2 chaînes de caractères :

Protéine 1: A**GGGCTT**ACTA

Protéine 2: GGGCT ACTAGGGGCC

Par exemple une distance de Levenshtein, avec trois opérations possibles: insertion, suppression, remplacement.

EXEMPLE PERSONNALISE

SVM pour classifier des protéines. On peut comparer deux séquences comme on comparerait 2 chaînes de caractères :

Protéine 1: A**GGGCTT**ACTA

Protéine 2: GGGCT ACTAGGGGCC

Puisque vous êtes capables de calculer cette distance pour toutes les paires de protéines, vous avez tout ce qu'il vous faut pour entraîner un SVM avec ce noyau. A aucun moment vous n'avez eu besoin de trouver des features pour représenter chaque protéine.

CONCLUSION SUR LES SVMS

Avantages :

- Très performants.
- Ne nécessitent que la similarité des objets en entrée et non les objets en eux-mêmes. On peut donc traiter images, séquences biologiques, vidéos... Il n'y a pas de limite.
- Très efficaces en grande dimension.

Inconvénients :

- Il faut pouvoir fournir un noyau adapté, sans quoi le modèle échoue.
- Difficiles à interpréter.
- Dépendent souvent de paramètres à optimiser.
- Parfois long !