Université de Batna 2 (Mostefa Ben Boulaid)
Faculté de Technologie
Département de Génie Industriel

Apprentissage Automatique
Cours 2 (Classification)

Pr Hassen BOUZGOU

Différents problèmes de ML

	Apprentissage Supervisé	Apprentissage Non-supervisé
Discrète	Classification / Catégorisation	Clustering
Continue	Régression	Réduction de dimensionnalité

Classification

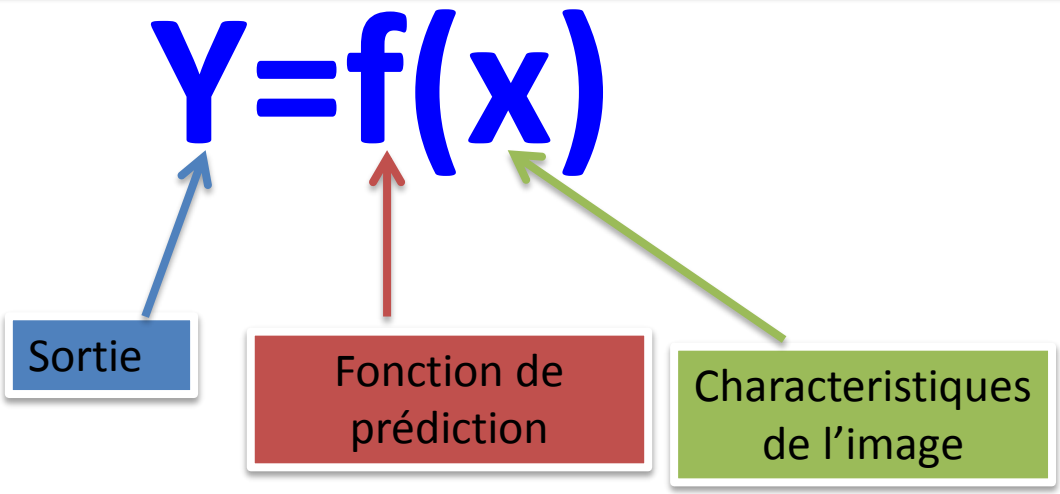
Appliquez une fonction de prédiction à une ***représentation*** de l'image pour obtenir la sortie souhaitée:

f() = pomme

f() = tomate

f() = vache

Cadre général

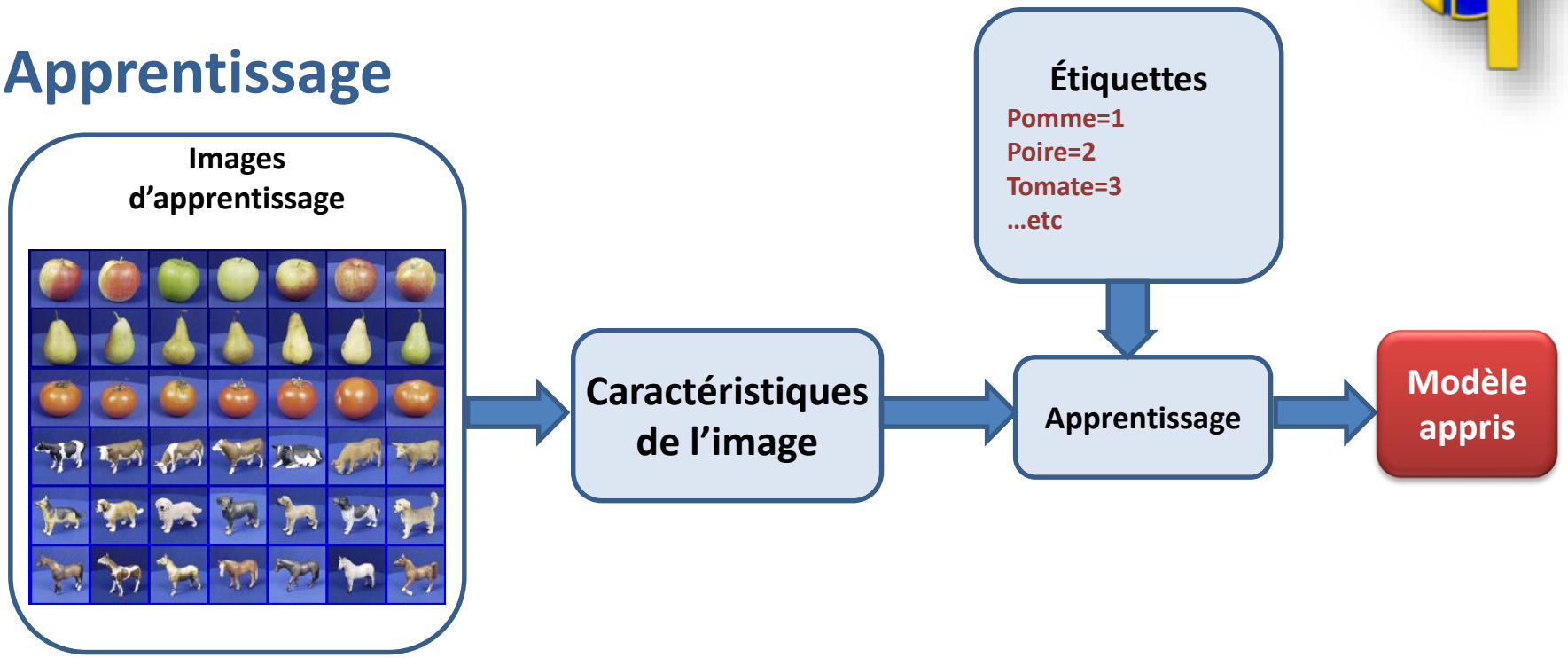


Ensemble d'apprentissage: étant donné un ensemble d'exemples étiquetés $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimer la fonction de prédiction **f** en minimisant l'erreur de prédiction sur l'ensemble d'apprentissage

Ensemble de test: appliquez **f** à un exemple de test jamais vu auparavant et affichez la valeur prédite y = f (x)

Les étapes du processus de classification

Apprentissage



Test

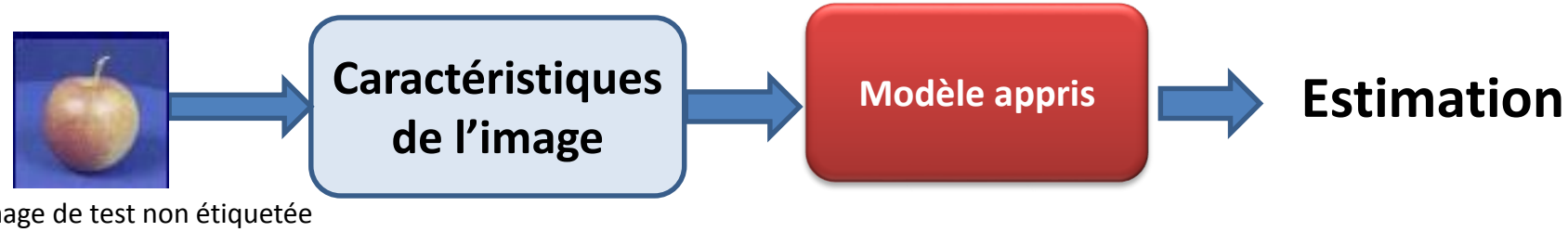


Image de test non étiquetée

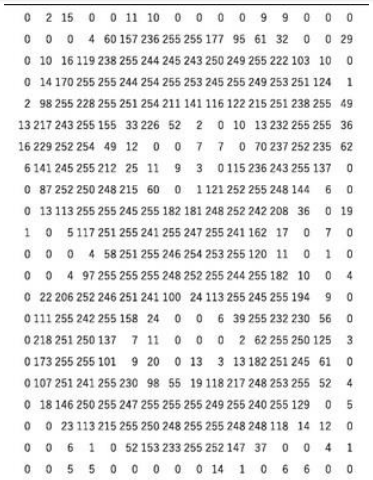
Extraction des Caractéristiques

- l'extraction de caractéristiques commence à partir d'un ensemble initial de données mesurées et construit des valeurs dérivées (caractéristiques) destinées à être **informatives** et **non redondantes**, facilitant les étapes ultérieures d'apprentissage et de généralisation. *L'extraction de caractéristiques est liée à la* ***réduction de la dimensionnalité.***

- Lorsque les données d'entrée d'un algorithme sont trop volumineuses pour être traitées et qu'ils peuvent êtres redondantes (la répétitivité des images présentées en pixels).

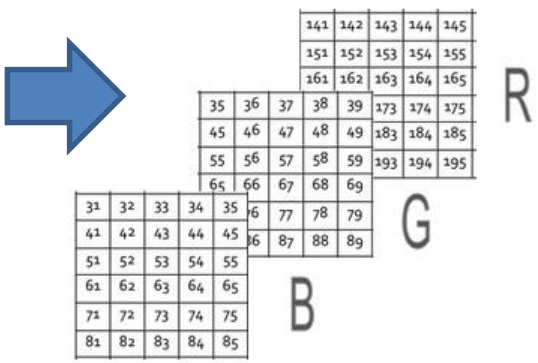
  - Transformation en un ensemble réduit de fonctionnalités (**Features**) (également appelées caractéristiques).
  - Détermination d'un sous-ensemble des caractéristiques initiales est appelée sélection de caractéristiques (**Features selection**) .
  - Les caractéristiques sélectionnées doivent contenir les informations pertinentes des données d'entrée, de sorte que la tâche souhaitée puisse être effectuée en utilisant cette représentation réduite au lieu des données initiales complètes.

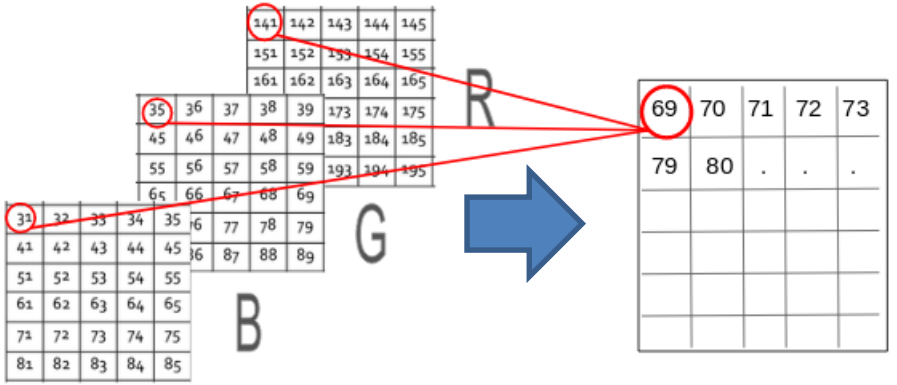
Extraction des Caractéristiques (1)



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

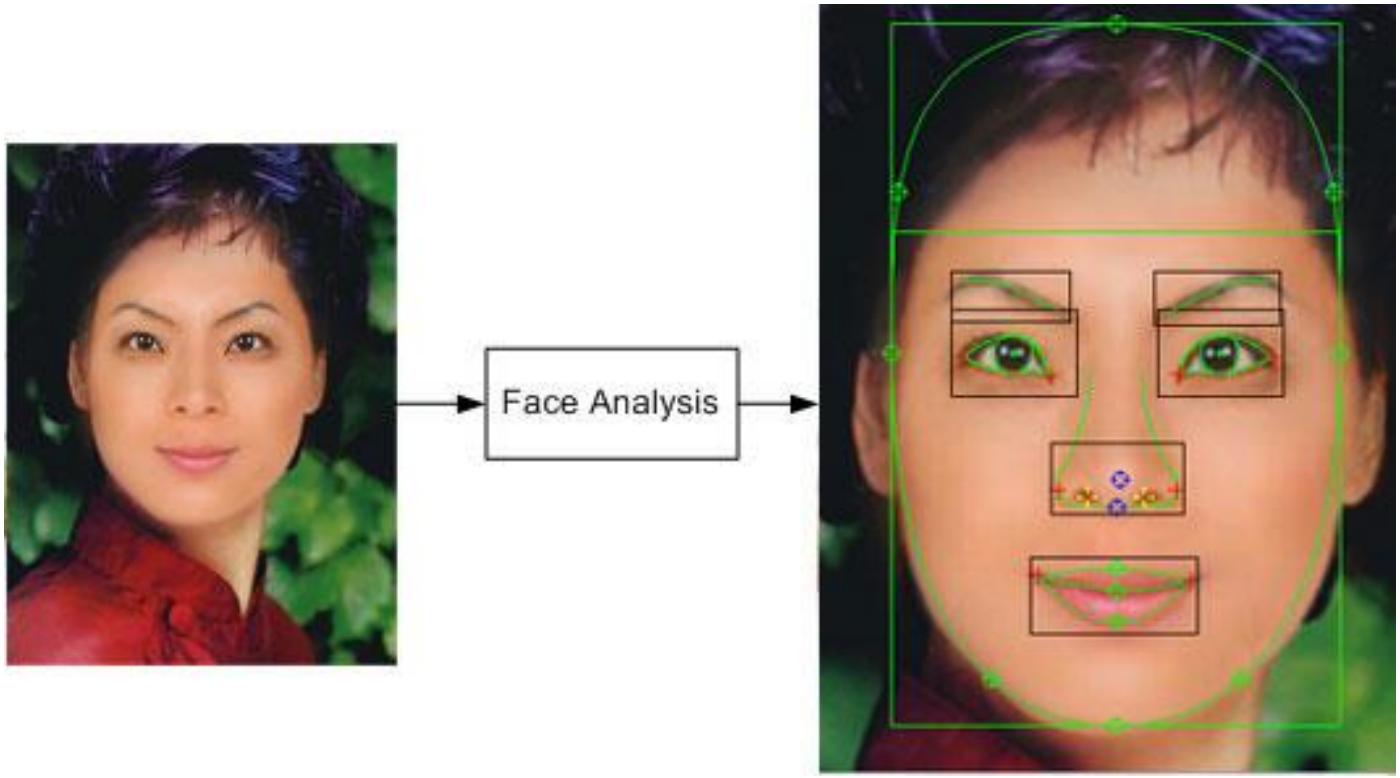
Colour Image





Extraction des Caractéristiques (2)

Caractéristiques pertinentes



Modélisation prédictive de classification

- En apprentissage automatique, la classification fait référence à un problème de modélisation prédictive où une étiquette de classe est prédite pour un exemple donné de données d'entrée.
- Nécessite un ensemble de données d'apprentissage avec de nombreux exemples d'entrées et de sorties à partir desquels le modèle apprendre.
- Un modèle utilisera l'ensemble de données d'apprentissage et calculera la meilleure façon de mapper des exemples de données d'entrée à des étiquettes de classe spécifiques.
- l'ensemble de données d'apprentissage doit être suffisamment représentatif du problème et comporter de nombreux exemples de chaque étiquette de classe.

Modélisation prédictive de classification(1)

- Les étiquettes de classe sont souvent des valeurs de chaîne, par ex. « spam », «non-spam » et doivent être transformées en valeurs numériques avant d'être fournis à un algorithme pour la modélisation. Ceci est souvent appelé **codage d'étiquette**, où un entier unique est attribué à chaque étiquette de classe, par ex. "spam" = 0, « non-spam" = 1.

- Il n'y a pas de théorie précise sur la façon de mapper des algorithmes sur des types de problèmes ; au lieu de cela, il est généralement recommandé d'utiliser des expériences contrôlées et découvrir quel algorithme et quelle configuration d'algorithme donnent les meilleures performances pour une tâche de classification donnée.

- Quatre principaux types de tâches de classification existent:
  1. Classification binaire
  2. Classification multi-classes
  3. Classification multi-étiquettes
  4. Classification déséquilibrée

Classification Binaire

- La classification binaire fait référence aux tâches de classification qui ont deux étiquettes de classe.
  1- Détection de spam par courrier électronique (spam ou non).
  2- Prédiction de maladie (malade ou non).
  3- Diagnostic d'un système industriel (marche ou panne)
  3- …..etc.

- Les tâches de classification binaire impliquent une classe qui est **l'état normal** et une autre classe qui est **l'état anormal**. Par exemple, « pas de spam » est l'état normal et « spam » est l'état anormal. Un autre exemple est « cancer non détecté » est l'état normal d'une tâche qui implique un test médical et « cancer détecté » est l'état anormal. La classe pour l'état normal reçoit *généralement* l'étiquette de classe 0 et la classe avec l'état anormal reçoit l'étiquette de classe 1.

- Les algorithmes populaires qui peuvent être utilisés pour la classification binaire:
  - Régression logistique
  - Les k- voisins les plus proches
  - Arbres de décision
  - Machine à vecteur de support
  - Naïf Bayes

Classification multi-classes

- La classification multi-classe fait référence aux tâches de classification qui ont plus de deux étiquettes de classe.
  1- Classification des visages.
  2- Classification des espèces végétales.
  3- Reconnaissance optique de caractères. …etc.
- Contrairement à la classification binaire, la classification multi-classe n'a pas la notion de résultats normaux et anormaux. Au lieu de cela, les exemples sont classés comme appartenant à l'une parmi une gamme de classes connues.
- Le nombre d'étiquettes de classe peut être très important sur certains problèmes. Par exemple, un modèle peut prédire qu'une photo appartient à un parmi des milliers ou des dizaines de milliers de visages dans un système de reconnaissance faciale.
- Les algorithmes populaires qui peuvent être utilisés pour la classification multi-classes incluent :
  1- Les k- voisins les plus proches (kNN)
  2- Arbres de décision
  3- Naïf Bayes
  4- Forêt aléatoire
  …etc.

Classification multi-étiquettes

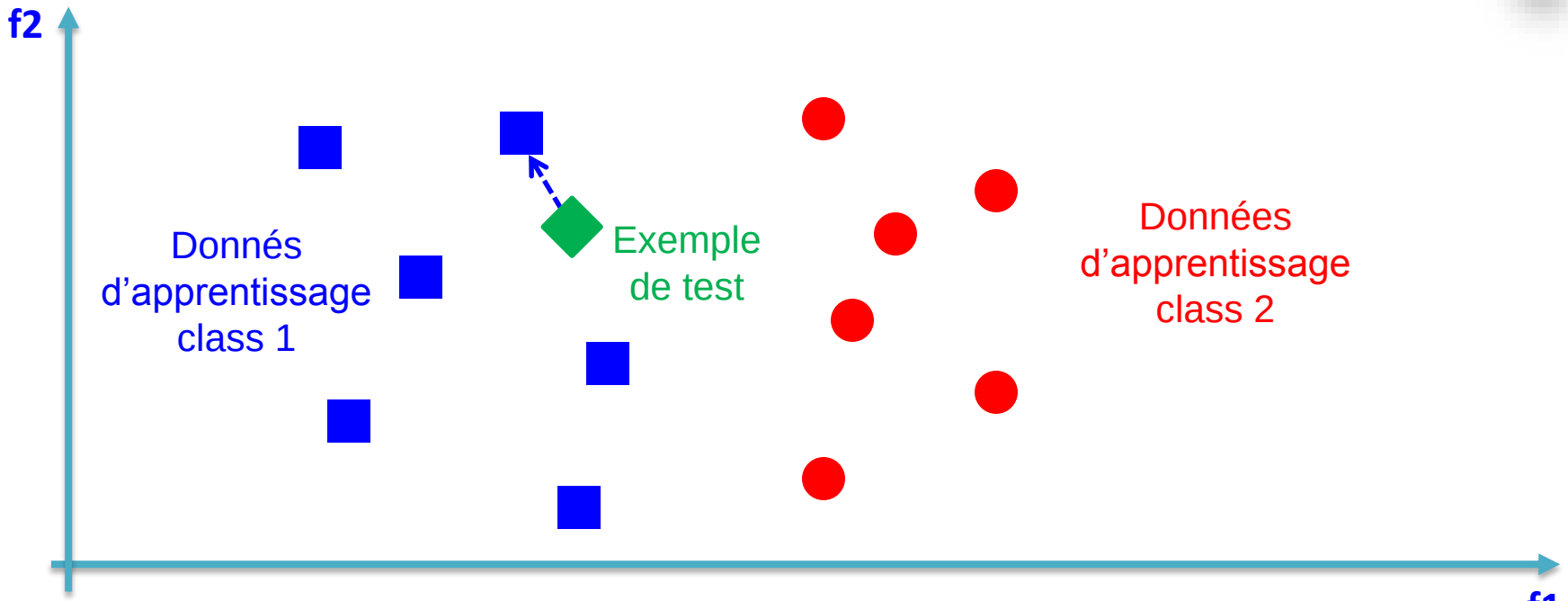
- La classification multi-étiquette fait référence aux tâches de classification qui ont deux ou plusieurs étiquettes de classe, où une ou plusieurs étiquettes de classe peuvent être prédites pour chaque exemple.
- Prenons l'exemple de la classification des photos, où une photo donnée peut avoir plusieurs objets dans la scène et un modèle peut prédire la présence de plusieurs objets connus sur la photo, tels que « vélo », « pomme », « personne », etc.
- Ceci est différent de la classification binaire et de la classification multi-classes, où une seule étiquette de classe est prédite pour chaque exemple.
- Il est courant de modéliser les tâches de classification multi-étiquettes avec un modèle qui prédit plusieurs sorties, chaque sortie étant prédite comme une distribution de **probabilité de Bernoulli**. Il s'agit essentiellement d'un modèle qui fait plusieurs prédictions de classification binaire pour chaque exemple.
- Des versions spécialisées d'algorithmes de classification standard peuvent être utilisées, appelées versions multi-étiquettes des algorithmes, notamment :
  1- Arbres de décision multi-étiquettes
  2- Forêts aléatoires multi-étiquettes
  3- Renforcement des gradients multi-étiquettes

la **loi de Bernoulli**, désigne la loi de probabilité d'une variable aléatoire discrète qui prend la valeur 1 avec la probabilité p et 0 avec la probabilité $q = 1 – p$.

Classification déséquilibrée

- La classification déséquilibrée fait référence aux tâches de classification où le nombre d'exemples dans chaque classe est inégalement réparti.
- En règle générale, les tâches de classification déséquilibrées sont des tâches de classification binaires où la majorité des exemples de l'ensemble de données d'apprentissage appartiennent à la classe normale et une minorité d'exemples appartiennent à la classe anormale.
- Les exemples comprennent:
  - Détection de fraude
  - Détection des valeurs aberrantes (outliers)
  - Tests de diagnostic médical
- Des techniques spécialisées peuvent être utilisées pour modifier la composition des échantillons dans l'ensemble de données d'apprentissage en sous-échantillonnant la classe majoritaire ou en suréchantillonnant la classe minoritaire. Les exemples comprennent:
  - Sous-échantillonnage aléatoire.
  - Suréchantillonnage SMOTE.
- Des algorithmes de modélisation spécialisés peuvent être utilisés pour accorder plus d'attention à la classe minoritaire lors de l'ajustement du modèle à l'ensemble de données d'apprentissage, tels que des algorithmes d'apprentissage automatique sensibles aux coûts (cost-sensitive):
  - Régression logistique sensible aux coûts
  - Arbres de décision sensibles aux coûts
  - Machines à vecteurs de support sensibles aux coûts

Classification (exemple de base)

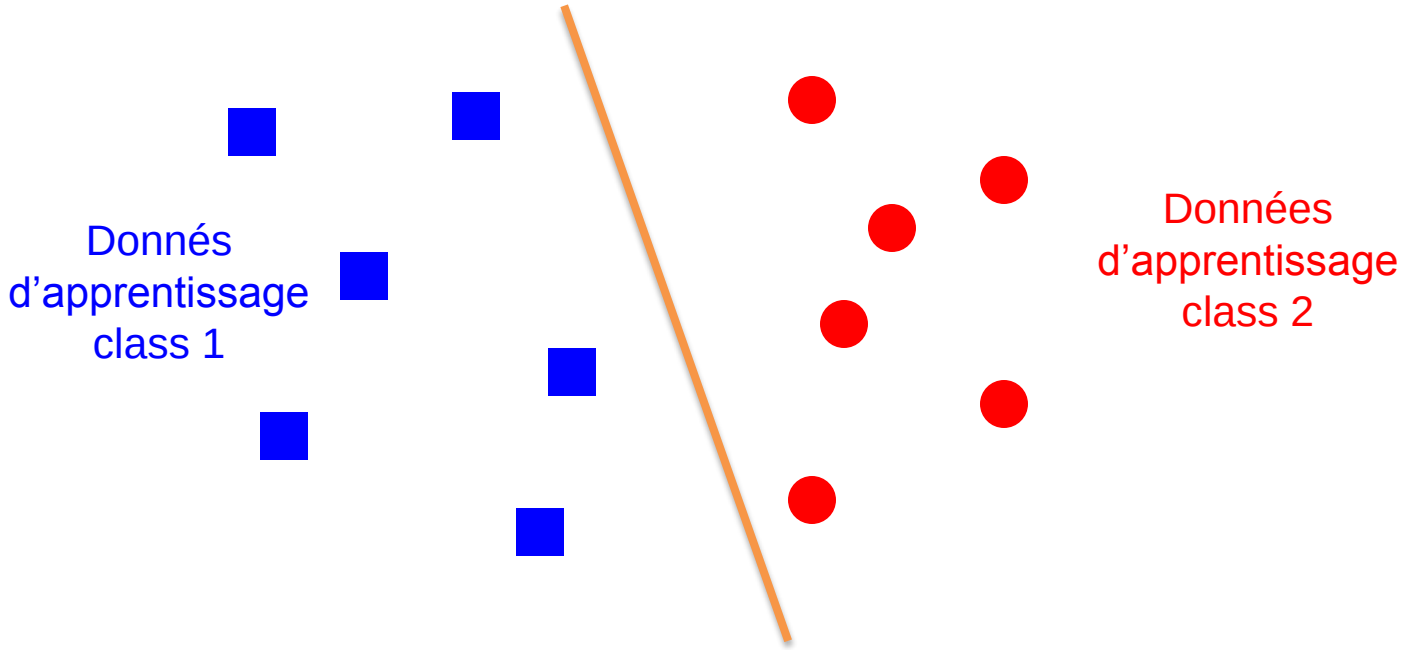


On suppose que f1 est la position du pixel et f2 est la valeur du pixel d'une image binaire (noir et blanc)

f(**x**) = étiquette de l'exemple d'apprentissage le plus proche de x

Tout ce dont nous avons besoin est une **fonction de distance** pour nos entrées $\Rightarrow$ **Aucun apprentissage!**

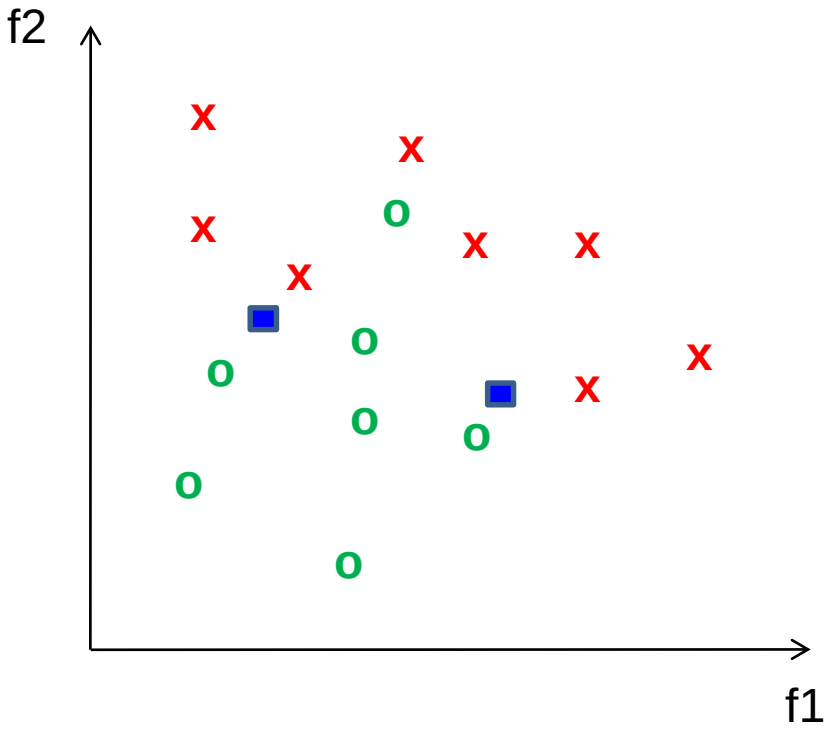
Classifieur linéaire



Trouver une fonction linéaire pour séparer les classes:

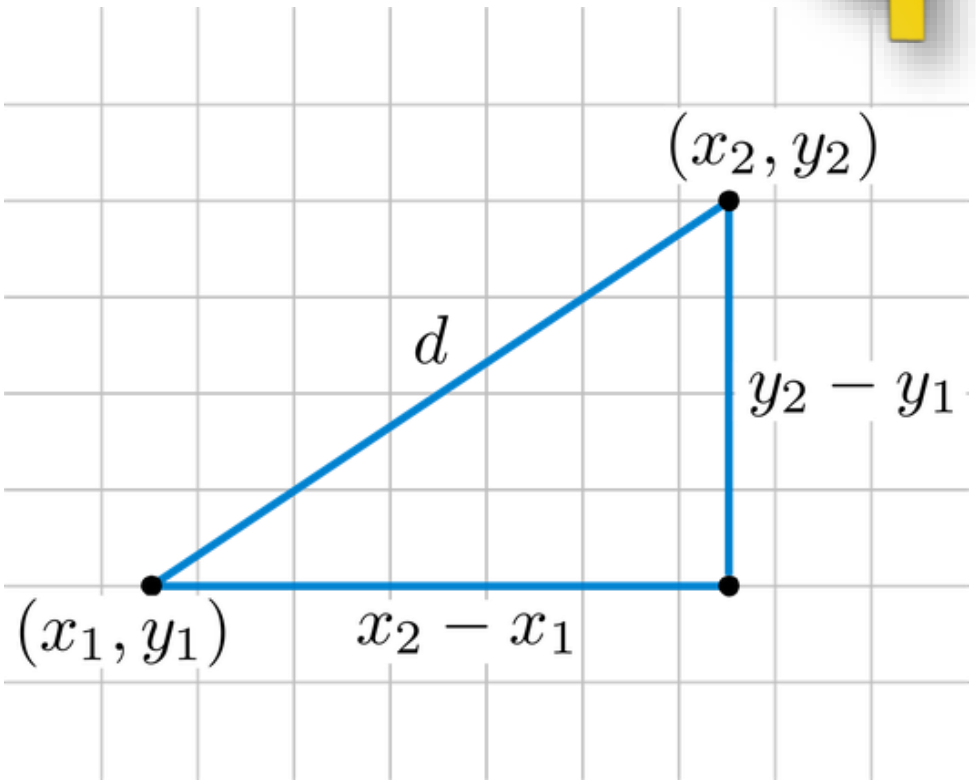
$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

K-Plus Proche Voisin (KNN)



Problème: **déterminer les classes des deux instances (carrés en bleu ■)**

K-Plus Proche Voisin (1)



Distance euclidienne

Cas 2-d

$$D2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Cas 3-d

$$D3 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Cas N-d

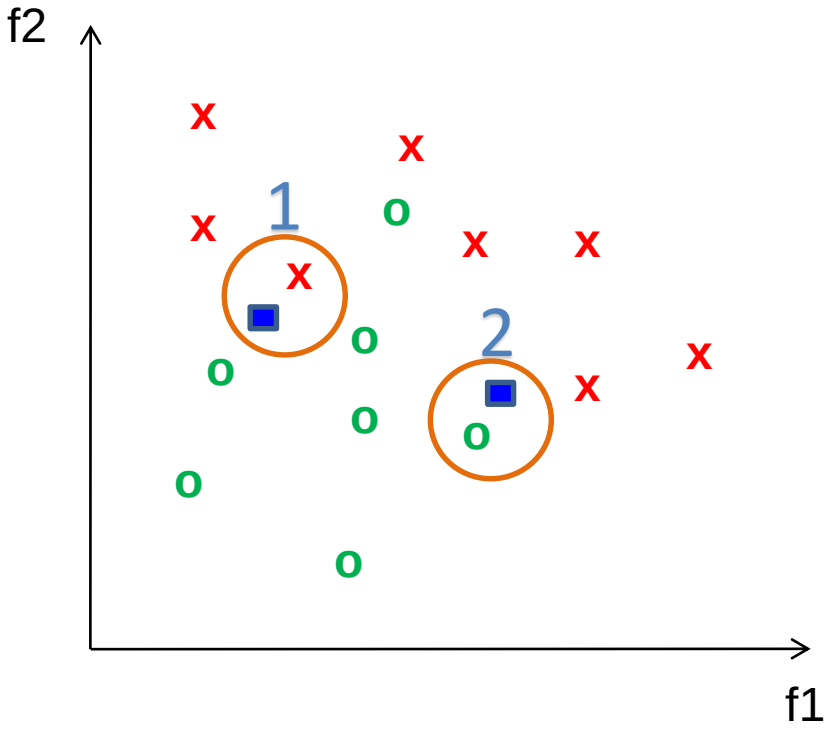
$$D(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

K-Plus Proche Voisin (2)

1-Plus proche

1 ∈ X

2 ∈ O

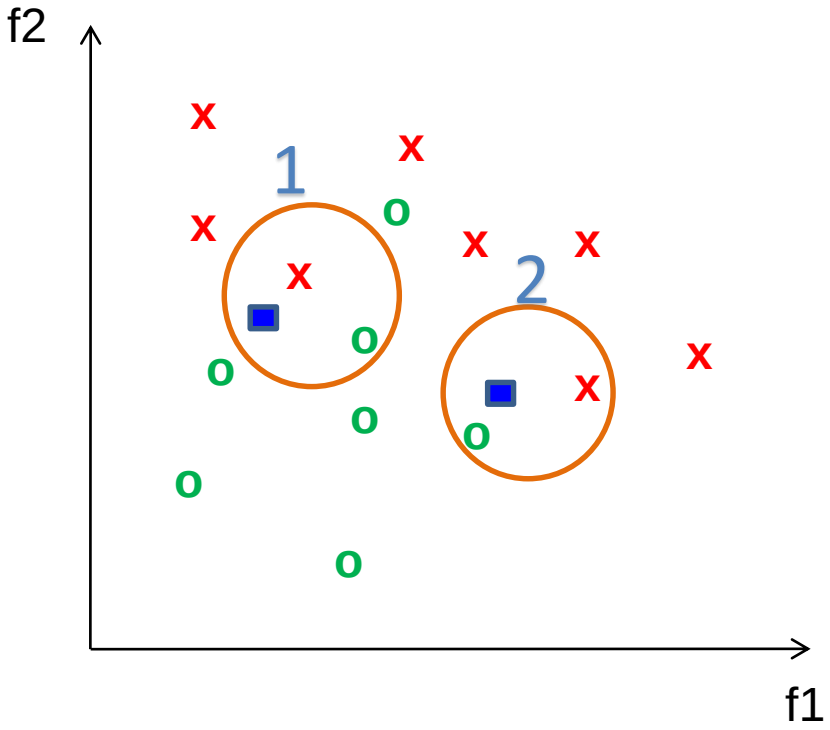


K-Plus Proche Voisin (3)

2-Plus proche

$1 \in ?$

$2 \in ?$

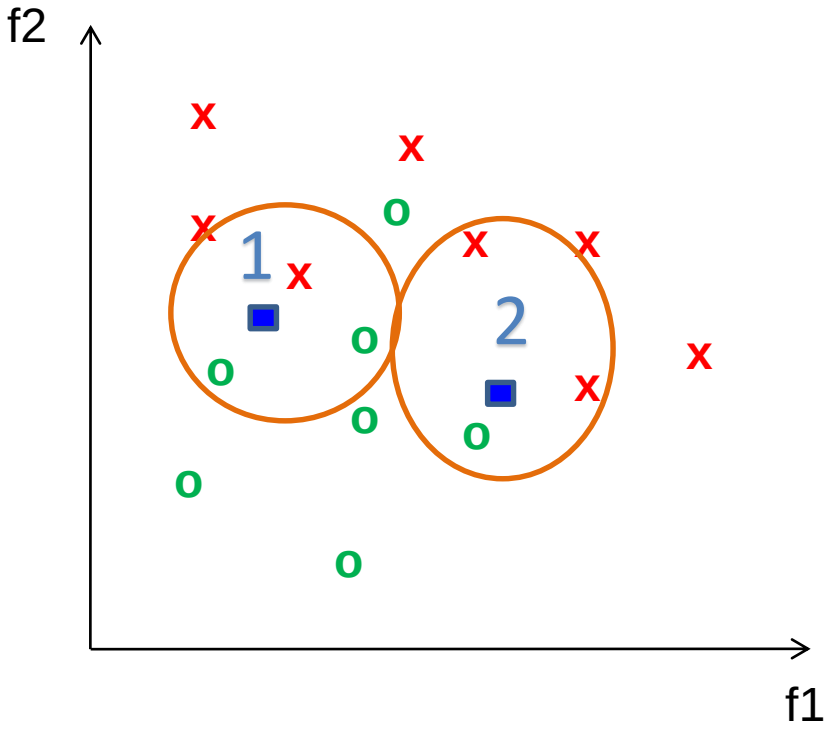


K-Plus Proche Voisin (4)

3-Plus proche

1 ∈ o

2 ∈ x

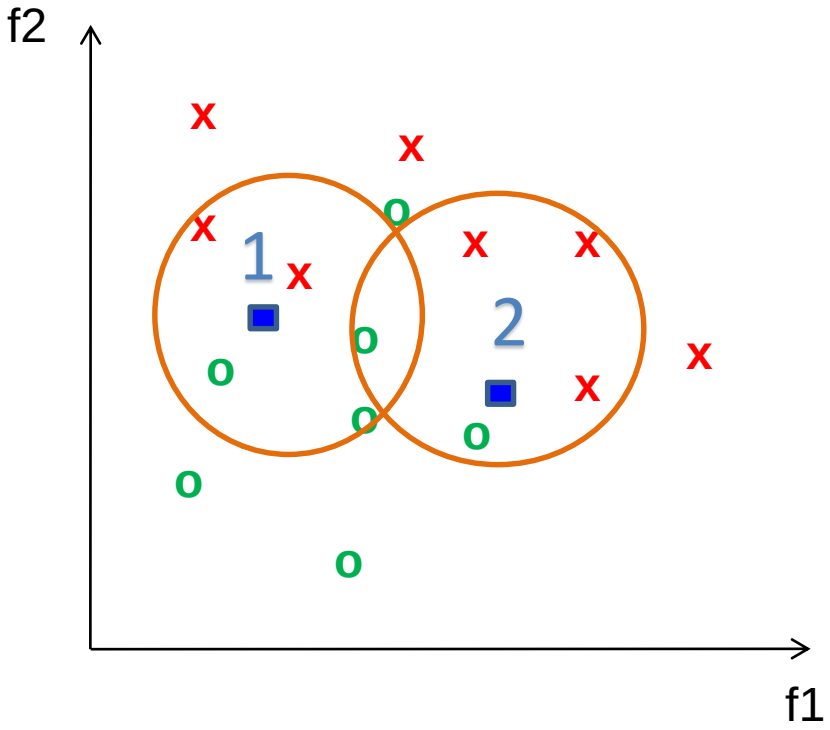


K-Plus Proche Voisin (5)

5-Plus proche

1 ∈ o

2 ∈ x



K-Plus Proche Voisin (6)

Comment déterminer la bonne valeur pour k?

- Expérimentalement;

-Commencez par k = 1 et utilisez un jeu de validation pour valider le taux d'erreur du classifieur

-Répéter avec k = k + 2

-Choisissez la valeur de k pour laquelle le taux d'erreur est minimum

Note: k devrait être un nombre impair pour éviter les égalités

Support Vector Machines (SVMs)

- 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.

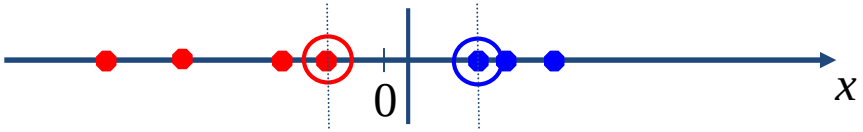
- Classifieur dérivé de la théorie statistique de l'apprentissage par Vapnik and Chervonenkis

  Vladimir Vapnik. *Statistical learning theory. 1998*. Wiley, New York, 1998.

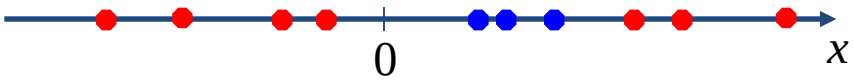
- Devenu populaire depuis que, partant d'images formées de pixels, il a permis des performances égales aux RNA pour reconnaitre l'écriture manuscrite.

SVM Non linéaires (1)

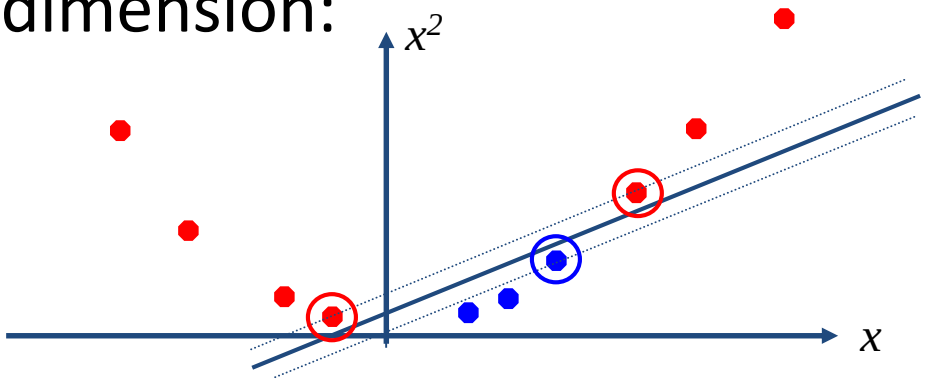
- Cas des données linéairement séparable:



Mais que faire si le jeu de données est difficile à séparer?

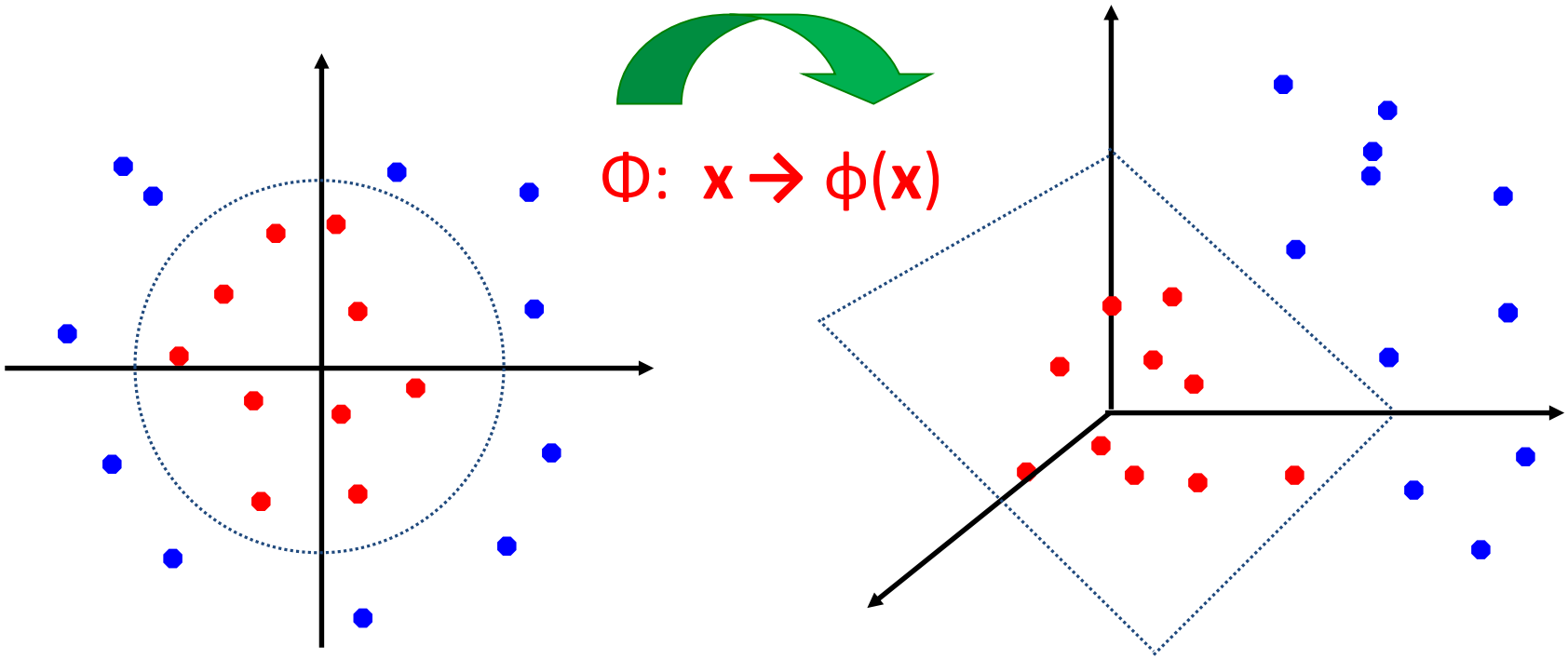


Nous pouvons le mapper à un espace de plus grande dimension:



SVM Non linéaires (2)

- **Idée générale:** l'espace d'entrée d'origine peut toujours être mappé à un espace d'entités de plus grande dimension où l'ensemble d'apprentissage est **séparable**:



Kernel Function

L'astuce du noyau s'utilise dans un algorithme qui ne dépend que du produit scalaire entre deux vecteurs d'entrée *x* et *y*. Après passage à un espace de redescription par une transformation φ, l'algorithme n'est plus dépendant que du produit scalaire :

$$\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

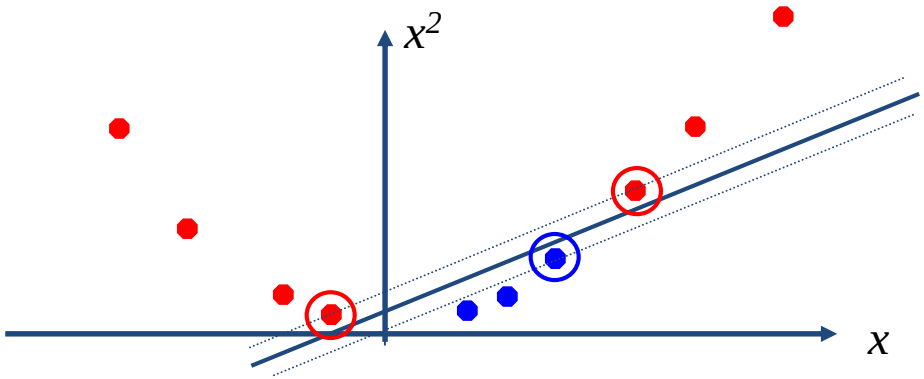
Le problème de ce produit scalaire est qu'il est effectué dans un espace de grande dimension, ce qui conduit à des calculs impraticables. L'idée est donc de remplacer ce calcul par une fonction noyau de la forme :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

pour être valide, la fonction du noyau doit satisfaire à la condition de Mercer

Fonction du Noyau NL

Considérons le noyau $\varphi(x) = (x, x^2)$

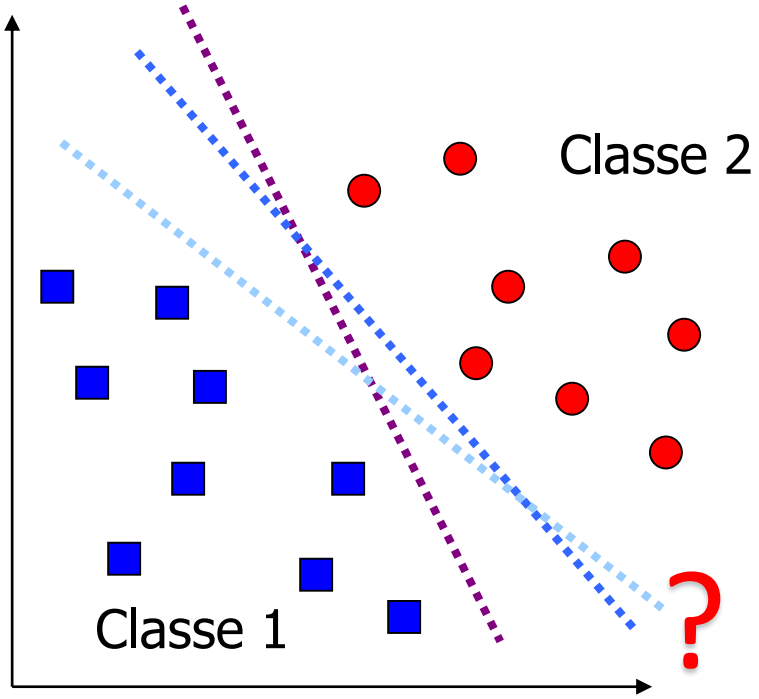


$$\varphi(x) \cdot \varphi(y) = (x, x^2) \cdot (y, y^2) = xy + x^2 y^2$$

$$K(x, y) = xy + x^2 y^2$$

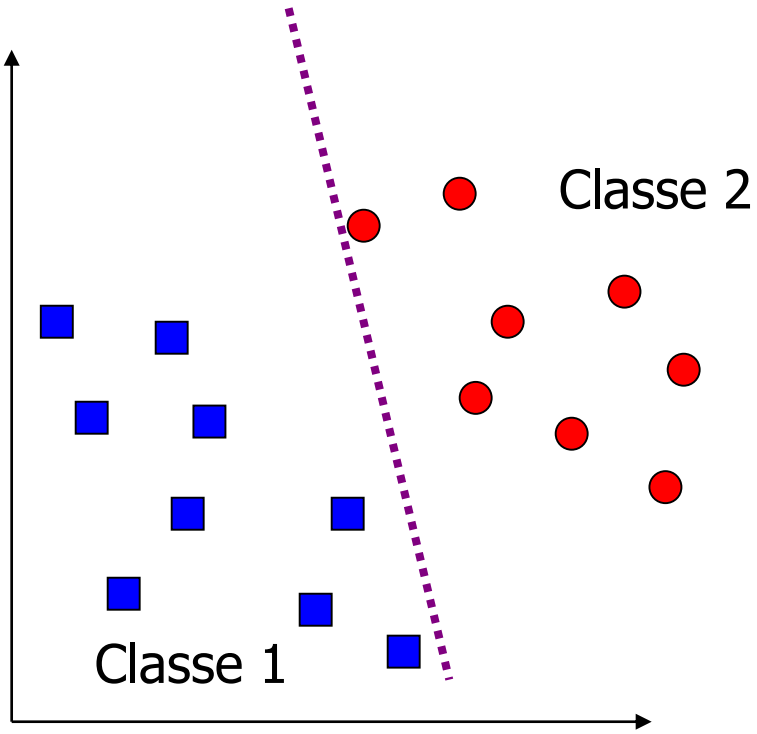
Hyperplan de décision

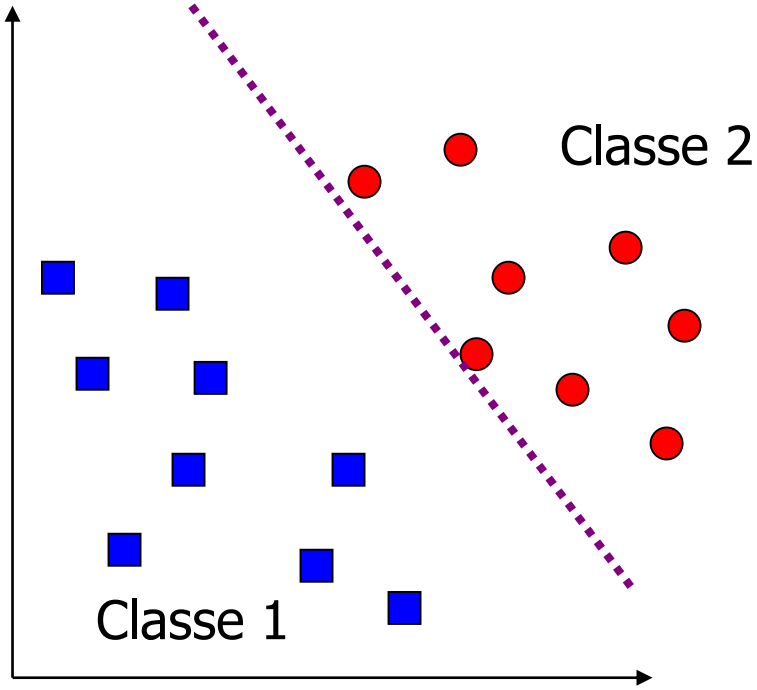
Plusieurs surfaces de décision existent pour séparer les classes ; laquelle choisir ?



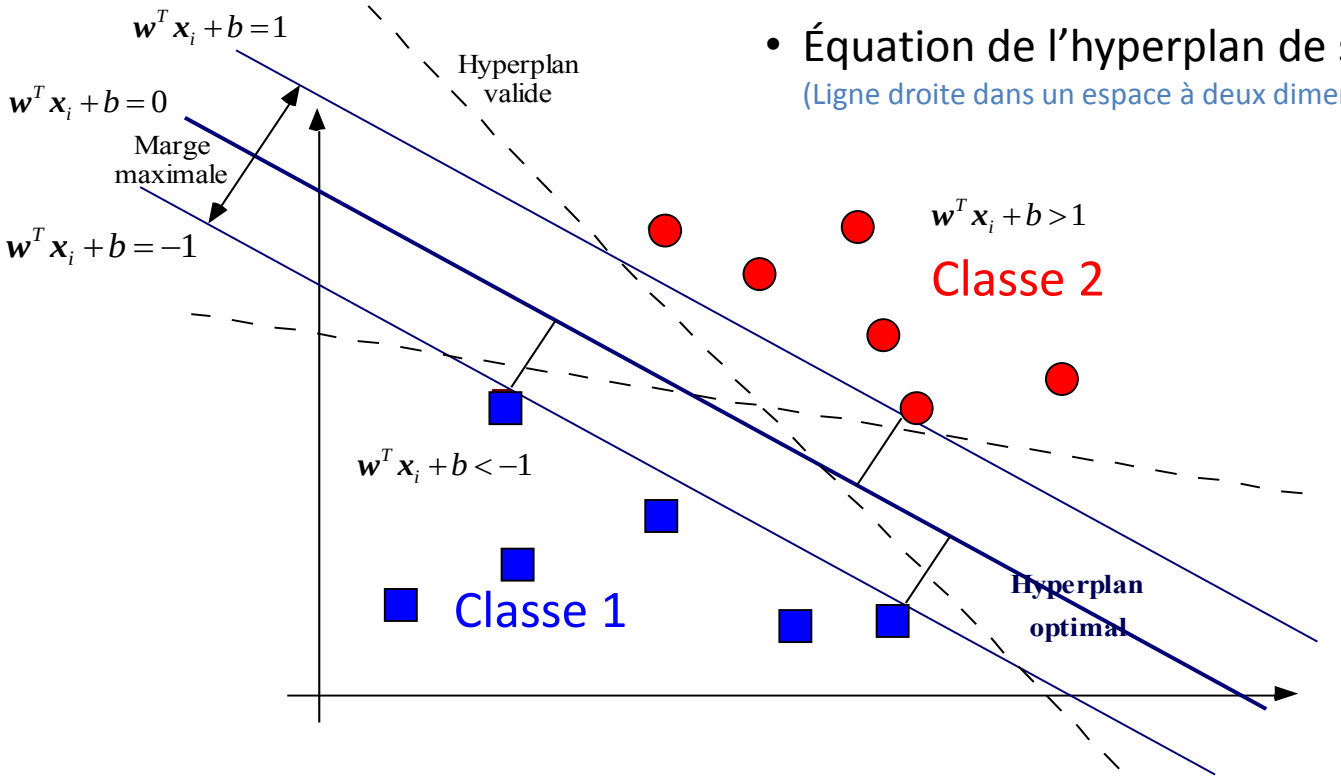
Hyperplan de décision (2)

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





Marge Maximale



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

- Si $\{x_i\} = \{x_1, \ldots, x_n\}$ est l'ensemble des données et $y_i \in \{1,-1\}$ est la classe de chacune, on devrait avoir :

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \qquad \forall i$$

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

Zones de l'espace de représentation

- Plusieurs zones sont définies dans l'espace de représentation
- f(x) = 0, on est sur la frontière
- f(x) > 0, on classe «+»
- f(x) < 0, on classe «-»
- f(x) = +1 ou -1, on est sur les droites délimitant des vecteurs de support

SVM Multi-classes

- Malheureusement, il n'y a pas de formulation SVM multi-classe « définitive »
- En pratique, nous devons obtenir un SVM multi-classes en combinant plusieurs SVM à deux classes

✓ **Un vs les autres**

- Apprentissage: apprendre un SVM pour chaque classe par rapport aux autres
- Test: appliquez chaque exemple SVM à un exemple de test et attribuez-lui la classe du SVM qui renvoie la valeur de décision la plus élevée

✓ **Un contre un**

- Apprentissage : apprenez un SVM pour chaque paire d'apprentissage
- Test: chaque SVM appris "vote" pour une classe à assigner à l'exemple de test

SVMs: Avantages et inconvénients

✓ Avantages

- ❑ Une bonne généralisation
- ❑ Le noyau est très puissant, flexible
- ❑ Les SVM fonctionnent très bien dans la pratique, même avec de très petites tailles d'échantillons de formation

✓ Les inconvénients

- ❑ Pas de SVM multi-classes "direct", il faut combiner des SVM à deux classes
- ❑ Calcul, mémoire
  - ❖ Pendant le temps d'apprentissage, on doit calculer la matrice des valeurs du noyau pour chaque paire d'exemples
  - ❖ L'apprentissage peut prendre beaucoup de temps pour des problèmes à grande taille

Evaluation du modèle de classification

❑ Précision

$$Précision_i = \frac{Nombre\ d'instances\ correctement\ attribuées\ à\ la\ classe\ i}{Nombre\ d'instances\ attribuées\ à\ la\ classe\ i}$$

❑ Rappel

$$Rappel_i = \frac{Nombre\ d'instances\ correctement\ attribuées\ à\ la\ classe\ i}{Nombre\ d'instances\ appartenant\ à\ la\ classe\ i}$$

❑ Mesure F

$$F = 2.\frac{Précision\ .\ Rappel}{Précision + Rappel}$$