

IFT3395/6390

Fondements de l'apprentissage machine

Terminologie de l'apprentissage

Professeur: Pascal Vincent

Apprendre à partir d'exemples !



“cheval”



“cheval”



“cheval”

Principe beaucoup plus général que d'écrire à la main, en partant de zéro, un algorithme pour reconnaître un cheval...

Les catégories de problèmes (tâches) standards de l'apprentissage automatique

Apprentissage supervisé

- ⦿ Classification
- ⦿ Régression

Apprentissage non supervisé

- ⦿ Estimation de densité
- ⦿ Partitionnement (*clustering*)
- ⦿ Réduction de dimensionnalité
- ⦿ Extraction de caractéristiques

Apprentissage par renforcement

Ex. de problème de classification

La reconnaissance de caractères,
(de chiffres manuscrits).

entrée x_i

2	2
2	2
2	2
2	2
2	2
3	3
3	3
3	3
3	3
3	3

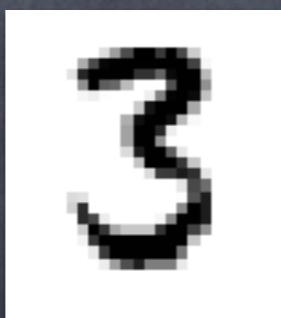
Ensemble de données
d'entraînement
(*training set*)

étiquette *label* y_i

Apprendre n'est pas
simplement
mémoriser...

C'est être capable
de généraliser à de
nouveaux cas!

Point de test:
(nouveau x)



2 ou 3?

Représentation des données

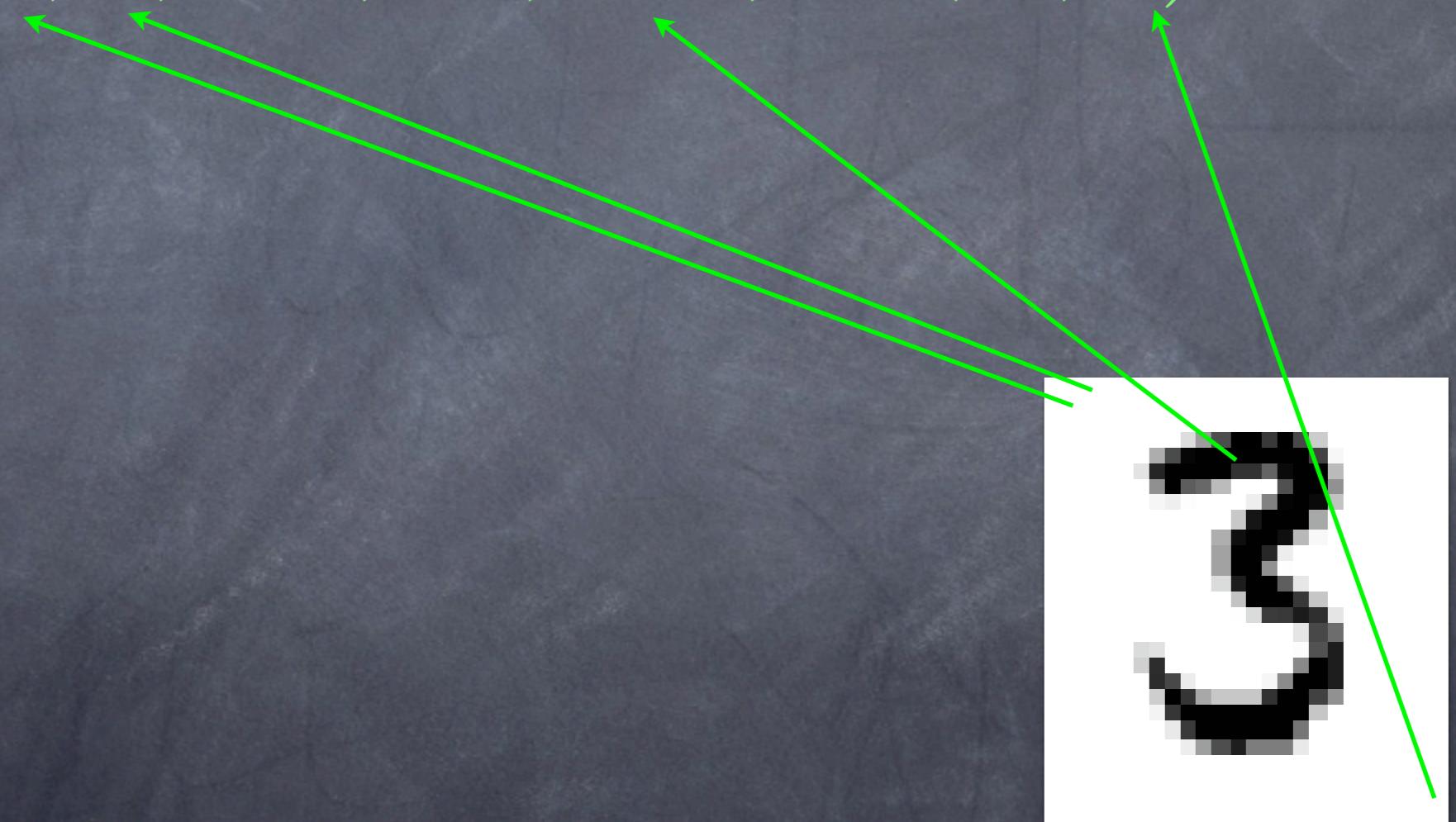
- La plupart des algorithmes d'apprentissage nécessitent une **représentation vectorielle** des exemples.
- Vecteurs numériques de taille fixe d

$$x \in \mathbb{R}^d$$

ex: $x = (0, 0, \dots, 54, 120, \dots, 0, 0)$

Ex: représentation vectorielle brute
d'une image (bitmap) en niveaux de gris

$$x = (0, 0, \dots, 54, 120, \dots, 0, 0)$$



Autre possibilité: Traits caractéristiques décrivant l'entrée (features, descripteurs)

$$\mathbf{x} = (\text{feature 1}, \text{feature 2}, \dots, \text{feature d})$$

- ⌚ Souvent conçus manuellement pour faciliter la tâche de l'apprentissage (feature engineering).
- ⌚ Ex: pour représenter une image naturelle (vision): descripteurs SIFT.
- ⌚ Ex: pour représenter un document: fréquence d'occurrences de certains mots (sac de mots).
Ex: feature k = combien de fois le mot «avion» apparaît dans le document.

Ensemble de données d'entraînement (*training set*)

Taille de l'ensemble, nombre d'exemples:

n

entrées:



cibles:

"cheval"



etc...



"cheval"

Point de test:



→ ?

Dimensionnalité
de l'entrée:

d

entrées: X

cibles: Y

(vecteur de traits caractéristiques)

X₁

prétraitement,
extraction de
caractéristiques



preprocessing,
feature
extraction

X_n

(3.5, -2, ..., 127, 0, ...)

+1

(-9.2, 32, ..., 24, 1, ...)

-1

etc...

X_{n,2}

(6.8, 54, ..., 17, -3, ...)

+1

Y₁

Y_n

X = (5.7, -27, ..., 64, 0, ...) → ?

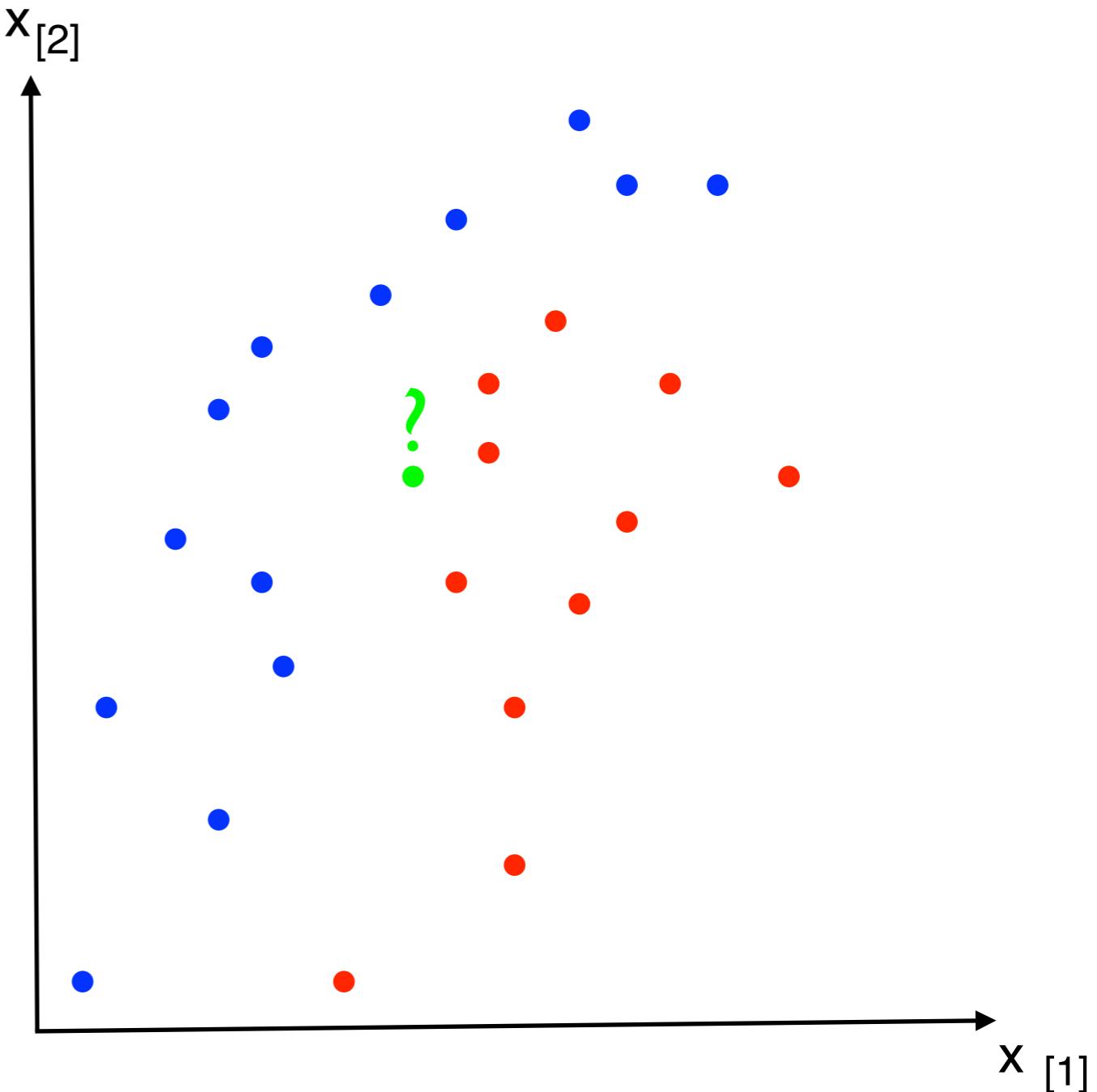
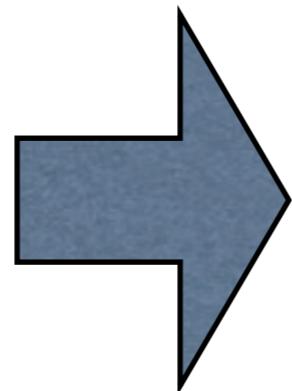
Importance des Dimensions du problème

- ⇒ Déterminent les algorithmes d'apprentissage applicables ou non (en raison de leur complexité algorithmique).
 - Nombre d'exemples: **n**
(parfois plusieurs millions)
 - Dimensions de l'entrée: **d**
combien de traits caractéristiques
(souvent de l'ordre de 100 à 1000, parfois des dizaines de milliers)
 - Dimensions de la cible qu'on veut prédire
ex. nombre de classes **m** (souvent faible, parfois des milliers)
- Un ensemble de donnée sera souvent organisé dans une matrice $n \times (d+1)$ ou $n \times (d+m)$

Représentation “géométrique” d'un ensemble de données

Chaque exemple de
l'ensemble de donné
est représenté par
un point x de \mathbb{R}^d

Ex: $x = (0, 0, \dots, 140, 0, \dots)$



Terminologie de l'apprentissage supervisé

Une entrée est généralement représentée par un vecteur de dimension d .

$x \in \mathbb{R}^d$

dimensionalité de l'entrée

Ensemble de données d'entraînement (*training set*)

1	entrée, observation, <i>input</i> , x_1	cible, <i>target</i> , sortie désirée, y_1
2	entrée, observation, <i>input</i> , x_2	cible, <i>target</i> , sortie désirée, y_2
3	entrée, observation, <i>input</i> , x_3	cible, <i>target</i> , sortie désirée, y_3
:	etc...	:
n	entrée, observation, <i>input</i> , x_n	cible, <i>target</i> , sortie désirée, y_n

point de test



? ? ?

taille de l'ensemble, nombre d'exemples, d'échantillons.

On cherche un algorithme qui produit une sortie (*output*) qui est une bonne prédiction de la cible. Cet algorithme trouve une bonne fonction $x \rightarrow y$

Terminologie de l'apprentissage supervisé

- ➊ Lorsque la **cible** est une **étiquette de classe**, une **variable catégorique** (indiquant à quelle classe ou catégorie l'entrée appartient, parmi plusieurs) on dit qu'on a affaire à un problème de **CLASSIFICATION**. (on utilise souvent un **entier** comme étiquette).
- ➋ Lorsque la **cible** est une (ou plusieurs) **valeur réelle** à prédire, on parle de problème de **RÉGRESSION**.

Quand on n'a pas de cible explicite, on est dans le cadre de l'apprentissage **non supervisé**.

Tâches de l'apprentissage

Apprentissage supervisé = predire une cible y à partir de l'entrée x

- y représente une catégorie ou “classe”
➡ classification (binaire ou multiclasse)
- y est une valeur réelle
➡ régression

} Modèles prédictifs

Apprentissage non supervisé: pas de cible (étiquette) explicite y

- modéliser la distribution de x
➡ estimation de densité
- découvrir une structure sous-jacente dans x
➡ groupements (clustering)
➡ réduction de dimensionalité (pour visualisation)

} Modèles descriptifs

Tâches de l'apprentissage

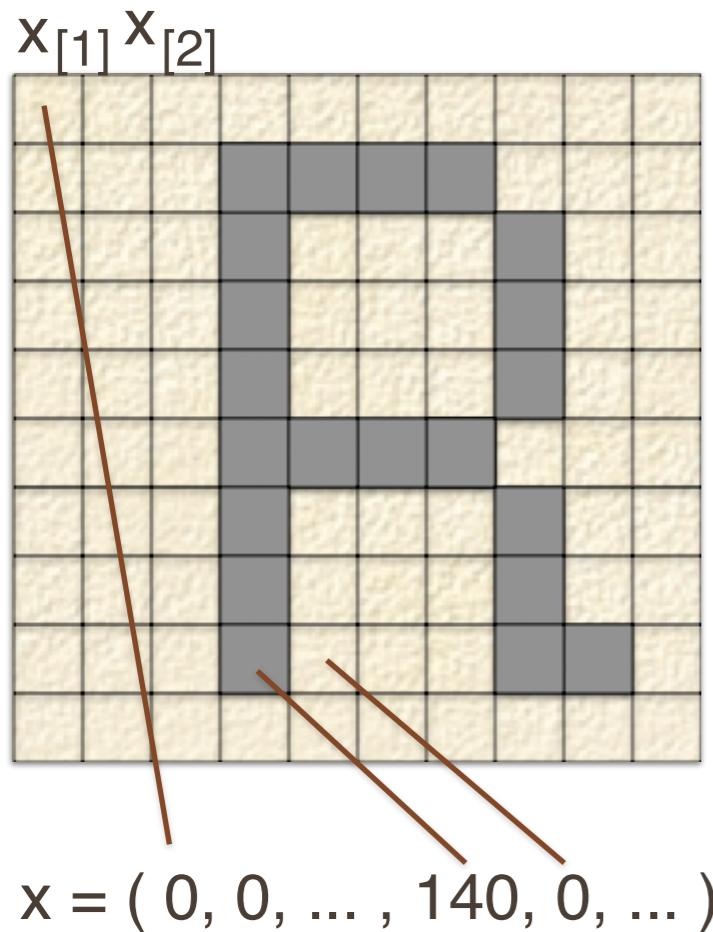
Apprentissage semi-supervisé:

même but que supervisé mais utilisant à la fois des exemples étiquetés (avec cible) et non étiquetés.

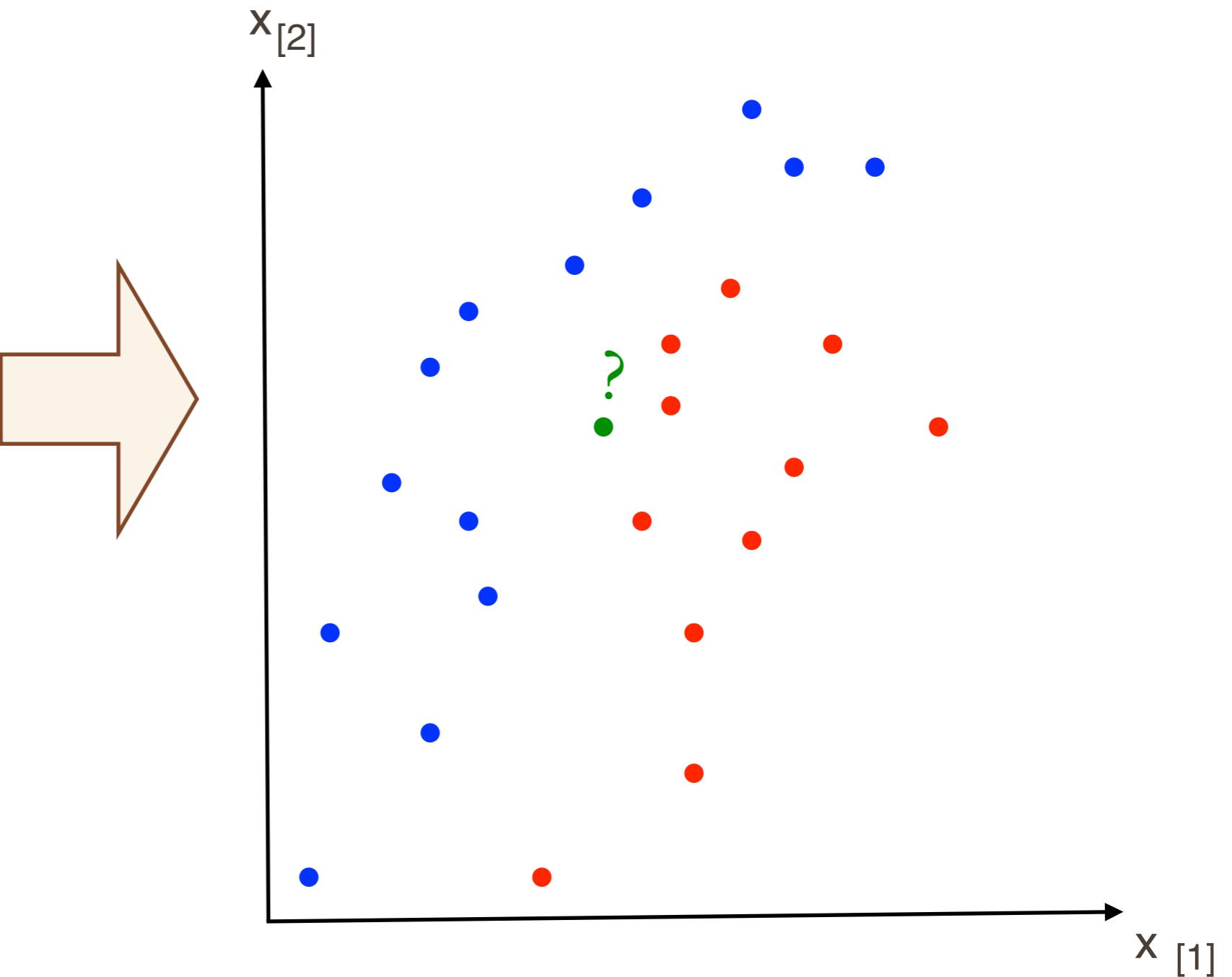
Apprentissage par renforcement:

un agent artificiel doit apprendre à décider quelles actions effectuer dans un *environnement changeant* afin de maximiser une *récompense totale*.

Représentation des données



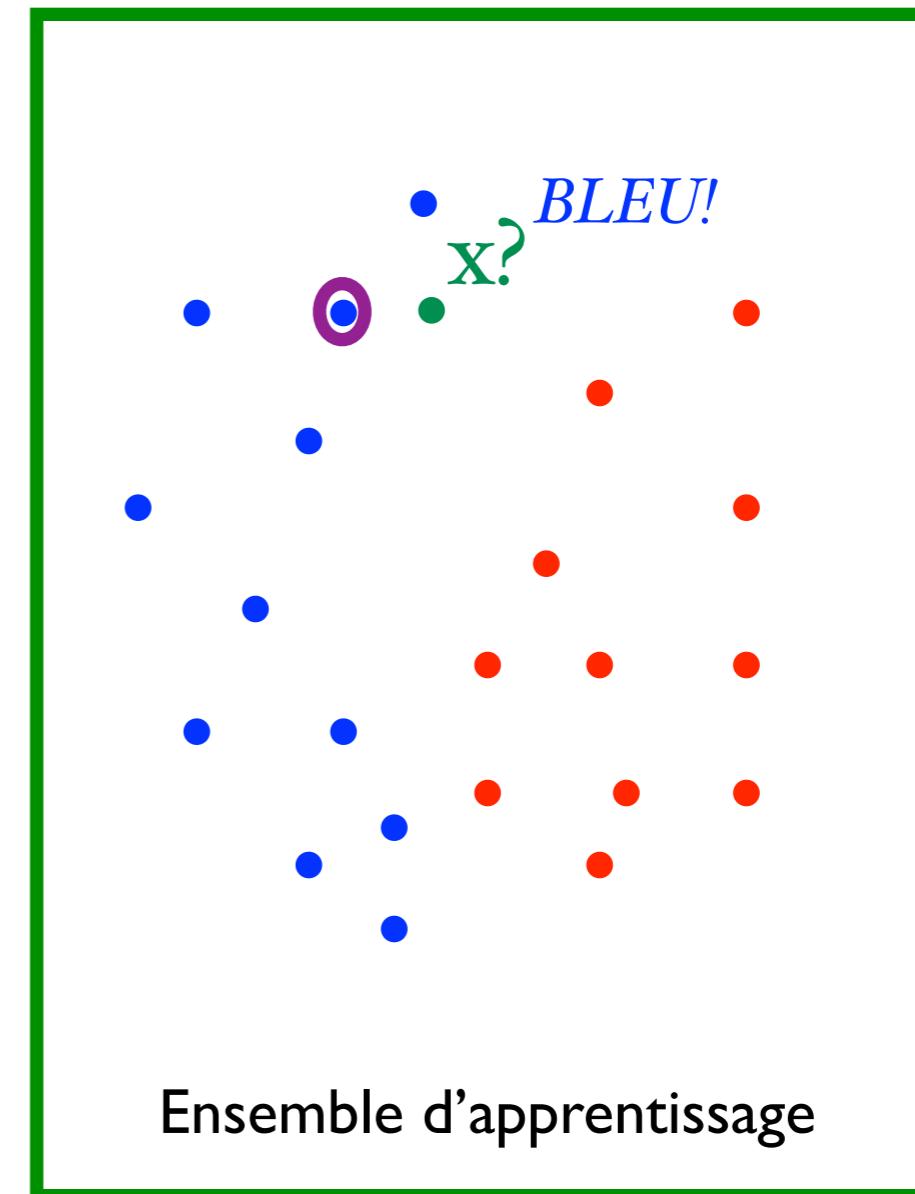
x point de \mathbb{R}^d



L'algorithme du plus proches voisin

Pour un point test x :

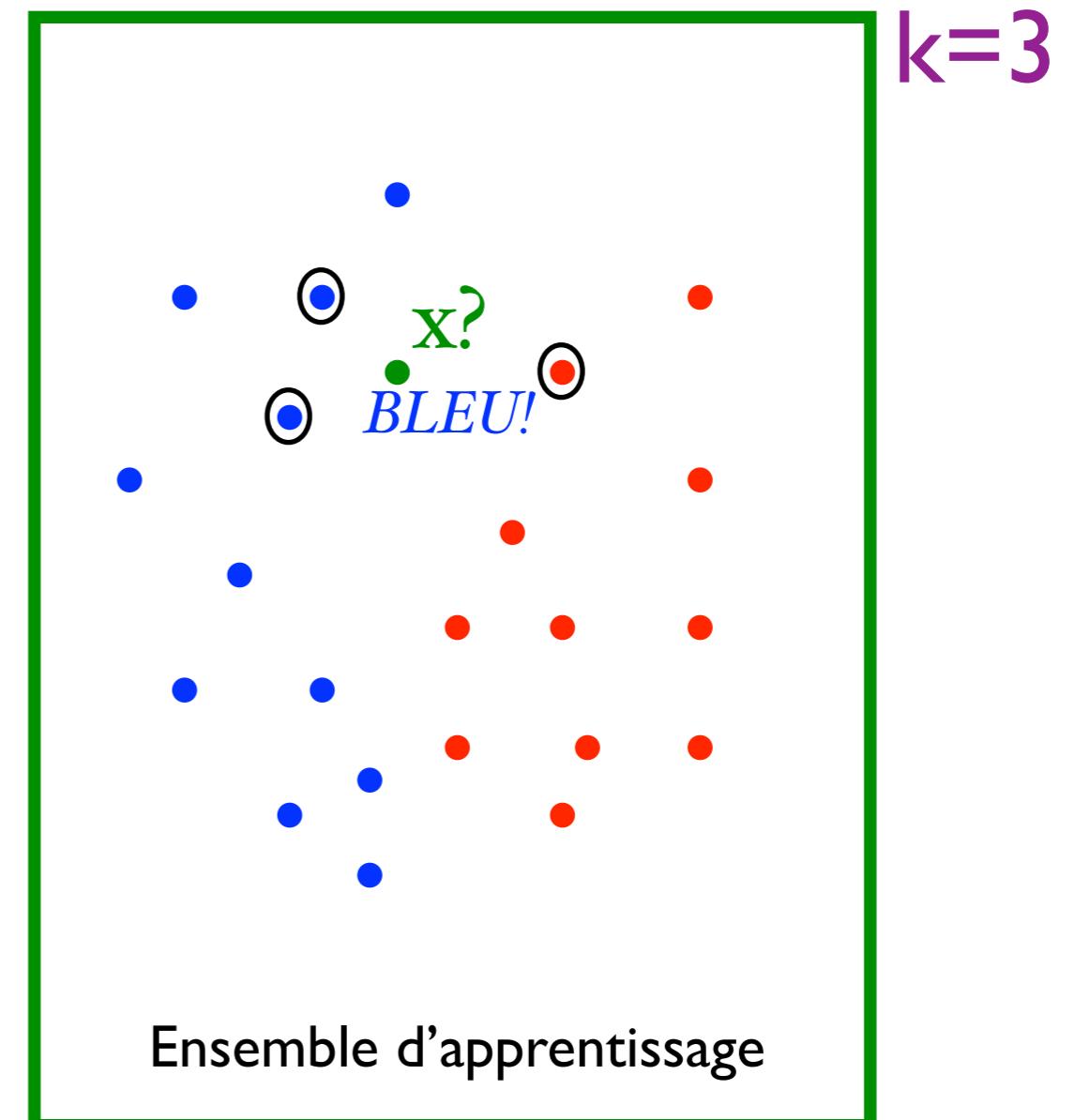
- On trouve **le plus proche voisin** de x parmi l'ensemble d'apprentissage selon une certaine mesure de distance (ex: distance Euclidienne).
- On associe à x la classe de ce plus proche voisin.



L'algorithme classique des k plus proches voisins (kNN)

Pour un point test x :

- On trouve les k plus proches voisins de x parmi l'ensemble d'apprentissage (typiquement selon la distance Euclidienne).
- On associe à x la classe **majoritaire** parmi ses k voisins

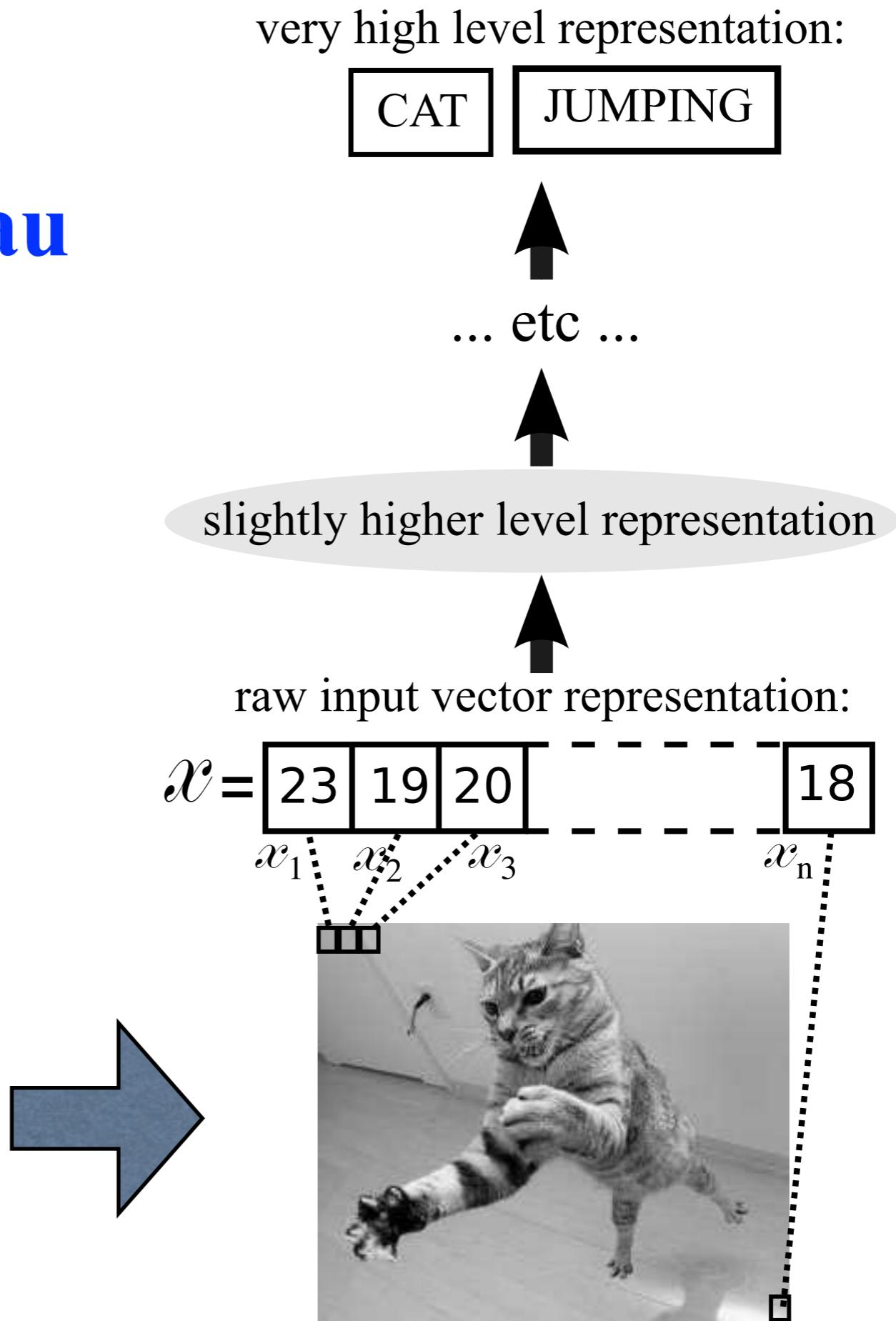


Petit rappel sur les vecteurs

(en dimension d)

- qu'est-ce qu'un vecteur?
représentations mathématiques,
informatiques, graphiques...
- distance Euclidienne
- norme
- produit scalaire
- calcul efficace de la distance d'un
point à un ensemble de points

La notion de niveau de représentation



Etapes d'un projet utilisant des algorithmes d'apprentissage

- ➊ Le **problème** à résoudre peut-il être **reformulé** sous la forme d'une des tâches standard de l'apprentissage? (classification, régression, estimation de densité, partitionnement, réduction de dimensionnalité, ...) Quel coût veut-on réellement optimiser dans ce problème?
- ➋ Examiner les **données** dont on peut disposer pour l'entraînement/test. Y en a-t-il suffisamment?, bien comprendre leur format, leur sémantique.
- ➌ Concevoir et coder les étapes de **prétraitement des données**. Le but est de les transformer dans une forme appropriée pour les algos d'apprentissage qu'on va utiliser.
- ➍ Entrainer/tester et évaluer correctement la **performance** "hors échantillon" des algorithmes considérés. (ex: découpage entraînement/test, validation croisée, ou bootstrap).

Etapes pratiques d'un projet de data-mining

- Identifier clairement le problème à résoudre
 - Le formaliser comme une tâche d'apprentissage spécifique basée sur les données disponibles.
 - Extraire les données et les regrouper (dans un fichier, une table).
 - Prétraiter les données pour obtenir une représentation appropriée pour les algorithmes d'apprentissage envisagés.
 - Modélisation: appliquer plusieurs algorithmes d'apprentissage sur les données.
 - Évaluation de la performance de chaque algorithme, pour choisir la meilleure approche.
 - Déployer le système opérationnel chez le client.
- “data plumbing”
- sélection de modèle