A2DI: Généralités et Contexte

John Klein

Université de Lille - CRIStAL UMR CNRS 9189

Plan du chapitre

ML : défintion.

- Le **Machine Learning** (ML) est une discipline informatique connue en français sous le nom d'**apprentissage artificiel** ou **reconnaissance de formes**.
- Cette discipline a pour but l'élaboration d'algorithmes permettant à un ordinateur de prendre une **décision** quand on lui présente une nouvelle situation à examiner.
- Cette capacité de la machine à prendre des décisions (en apparence) autonomes revient à une forme d'**intelligence artificielle**.

ML : Exemples de décision à prendre :

- Etant donné les résultats annuels de l'entreprise X, quel sera le cours des actions de cette entreprise ?

régression

(images pixabay.com)

ML : Exemples de décision à prendre :

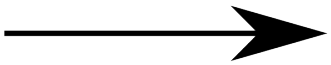
- Un objet vient de pénétrer l'espace aérien, est-ce un missile, un avion de chasse, un avion de ligne, un parapente ?

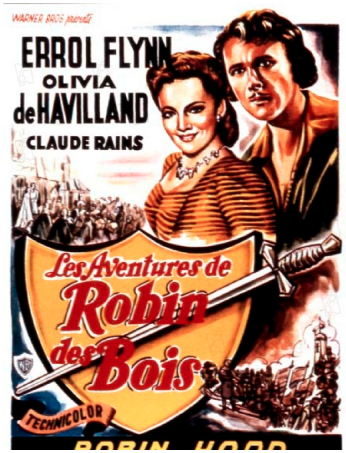
classification

(images pixabay.com)

ML : Exemples de décision à prendre :

- Un abonné a été voir *Harry Potter*, *James Bond* et *Hunger Games*, quelle liste de films peut-on lui conseiller ?





(WTF?)

recommandation

ML : Exemples de décision à prendre :

- Quel itinéraire devrais-je choisir ?

plannification

D'une manière générale, le machine learning cherche à résoudre des problèmes de prédiction. (images pixabay.com)

Quelle frontière entre du machine learning et de l'estimation en général ?

- Un algorithme de Machine Learning s'appuie sur un modèle mathématique tenant compte d'un historique.
- L'ensemble des informations contenues dans l'historique est appelé jeu de données (dataset).
- Exemple d'un estimateur qui ne relève **pas du machine learning** : un système physique représenté par une équation différentielle. Grâce à l'équation différentielle, on sait par exemple prédire la vitesse de rotation d'un moteur sachant le courant en entrée.
  Nous avons déterminé cette équation en nous basant sur les lois de la physique qui sont admises et vérifiées par des expériences rigoureuses et répétées.
  Nous ne nous sommes pas appuyés sur un historique du style *pour 2A, on avait 5000tr/mn* et *pour 3A, on avait 7500tr/mn*.

Quelle frontière entre du machine learning et de l'estimation en général ?

Je fais du machine learning si j'ai :

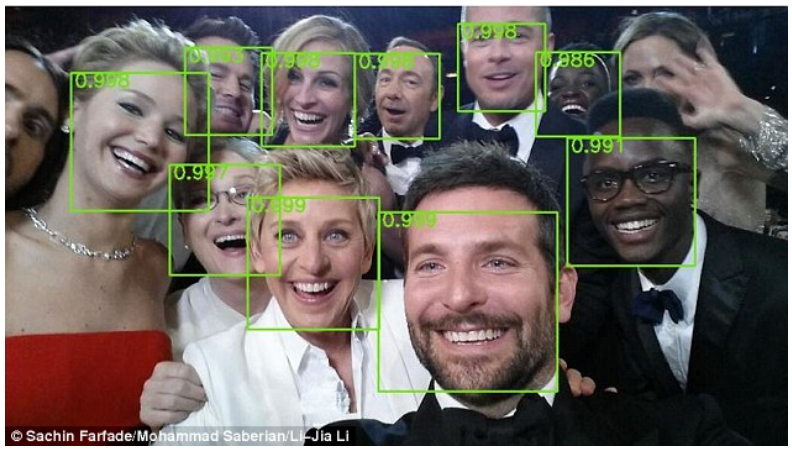
- un « pattern »,
- des données,

mais que je n'ai pas :

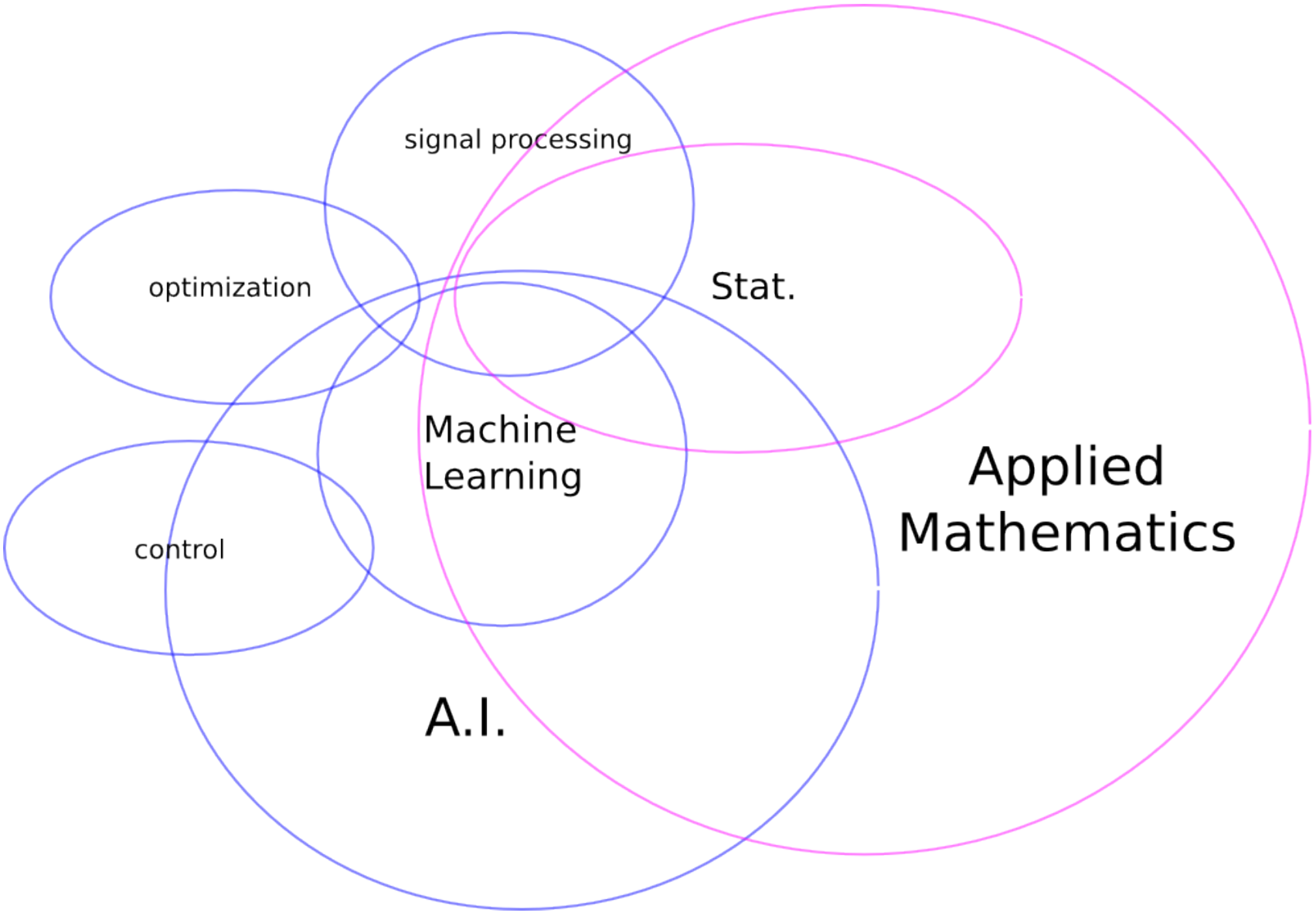
- un modèle général menant à la solution exacte (sans l'aide de données).

Le ML est partout :

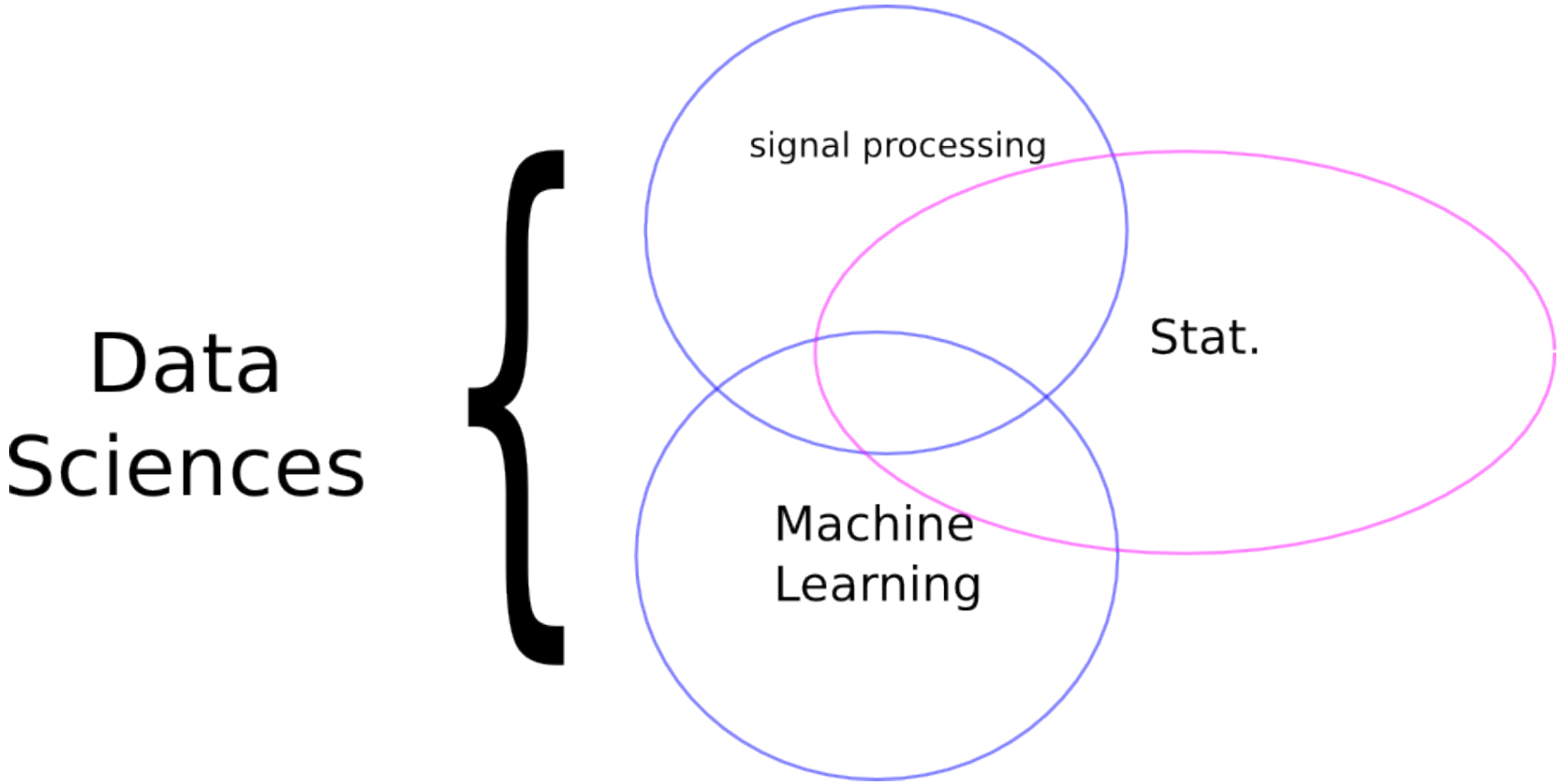
- Big data : définir un profil utilisateur en fonction de ses actions passées (commande de tel produit, clic sur telle pub, a cliqué “j”aime“ sur tel post, a tapé tel mot dans sa barre de recherche, etc.)
  rq : ce n’est pas la définition qu’un scientifique a du big data.
- Reconnaissance de visage :



ML : les cousins/cousines



ML = pattern recognition (reconnaissance de formes)
data mining $\subset$ {ML $\cup$ stat. $\cup$ trait. du signal} $\subset$ data sciences

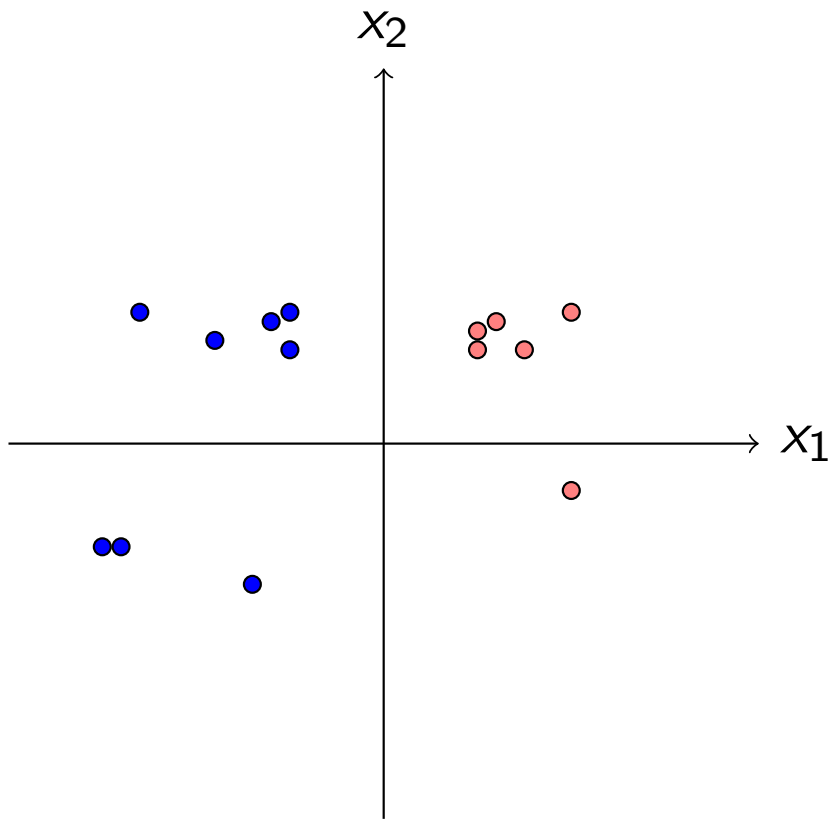


Plan du chapitre

Paradigme = catégorie de problème

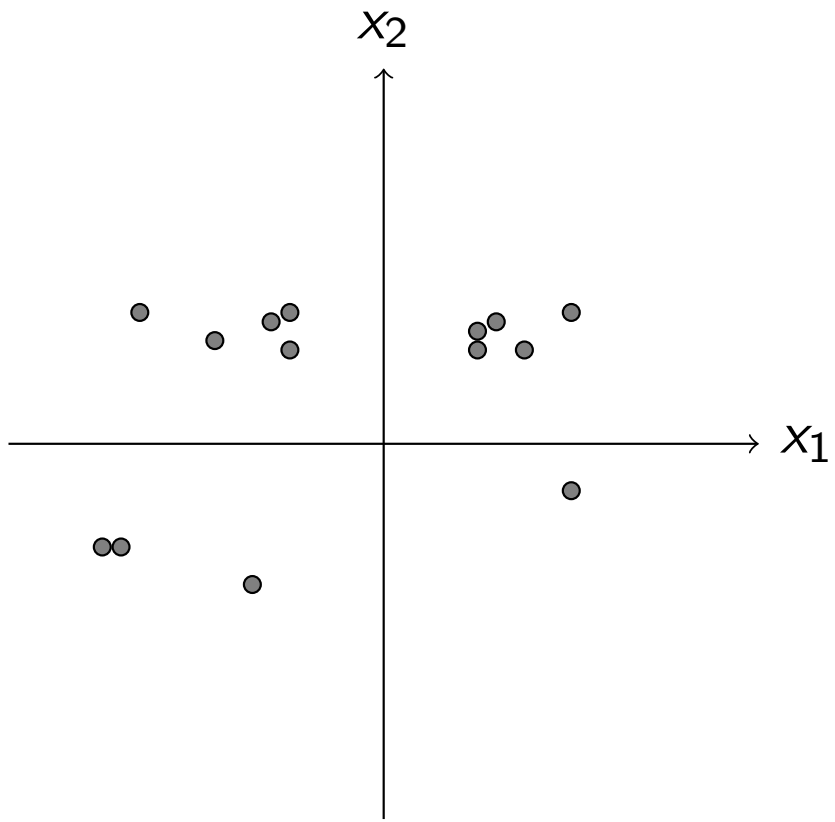
Paradigme n°1 : Apprentissage supervisé :

- On dispose d'un jeu de données entièrement étiqueté.
- Soit $\mathcal{D}$ cet ensemble. Chaque élément z dans $\mathcal{D}$ est de la forme (x, c) où x est un exemple d'apprentissage et c est l'étiquette associée à x.
- Exemple : on suppose que x est un vecteur à 2 dimensions, et $c \in \{red; blue\}$.



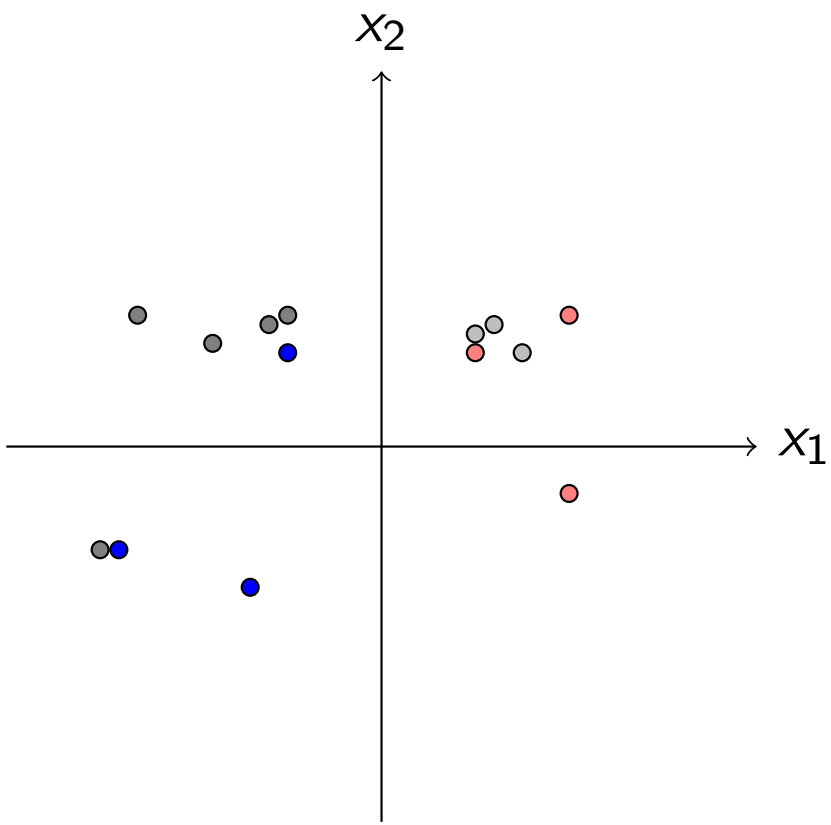
Paradigme n°2 : Apprentissage non-supervisé :

- On dispose d'un jeu de données $\mathcal{D}$ consitué uniquement d'exemples x sans les étiquettes.
- Exemple : on suppose que x est un vecteur à 2 dimensions.



Paradigme entre n°1 et 2 :

- semi-supervisé : certains exemples ont une etiquette mais d'autres non (cas le plus fréquent en pratique).



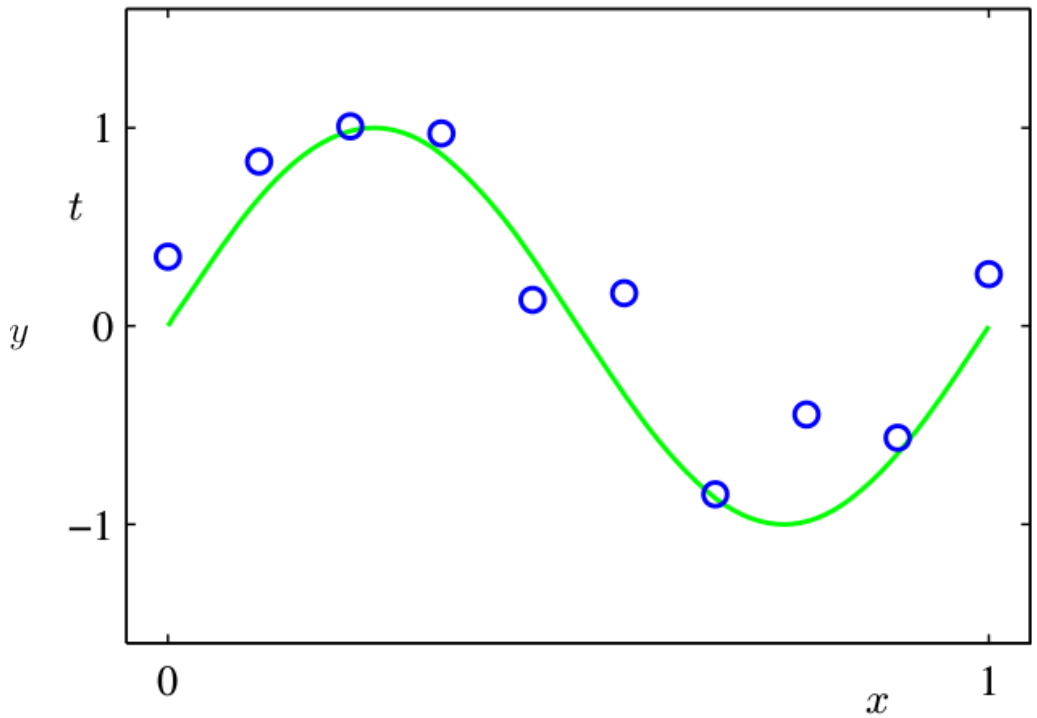
Paradigme entre n°1 et 2 :

- faiblement supervisé : les étiquettes sont mal connues.
  Supposons que l'ensemble des classes est $\{1; ..; 6\}$. Pour un exemple x_k, on va savoir que son étiquette appartient à un ensemble : $c_k \in \{1; 5; 6\}$.
  On pourrait éventuellement aussi connaître seulement une probabilité d'appartenance à une classe.

Paradigme n°1 et 2 :

Problèmes de régression et problèmes de classification : Soit $\mathcal{C}$ l'ensemble des solutions. On cherche à construire une fonction prédictrice $f : \mathbb{X} \longrightarrow \mathcal{C}$ qui associe un exemple x à une solution c.

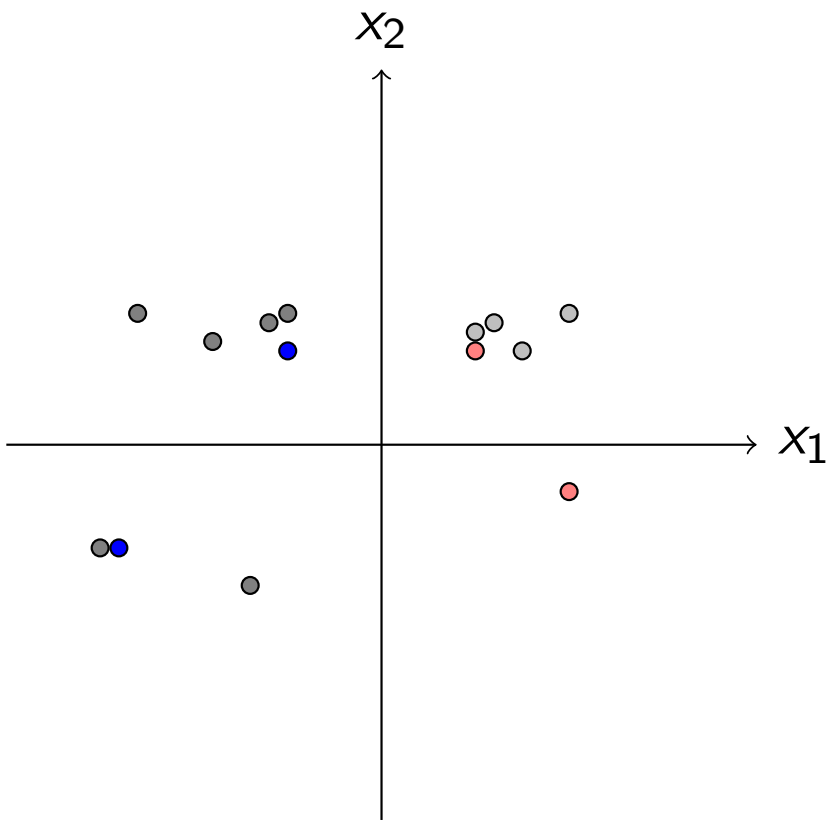
- Si $\mathcal{C}$ est discret (et fini en pratique), on parle de problème de classification et les éléments c sont appelés classes.
- Si $\mathcal{C}$ a la puissance du continu (et donc infini), on parle de problème de régression. Typiquement, $\mathcal{C} = \mathbb{R}^d$.



[Bishop 2006]

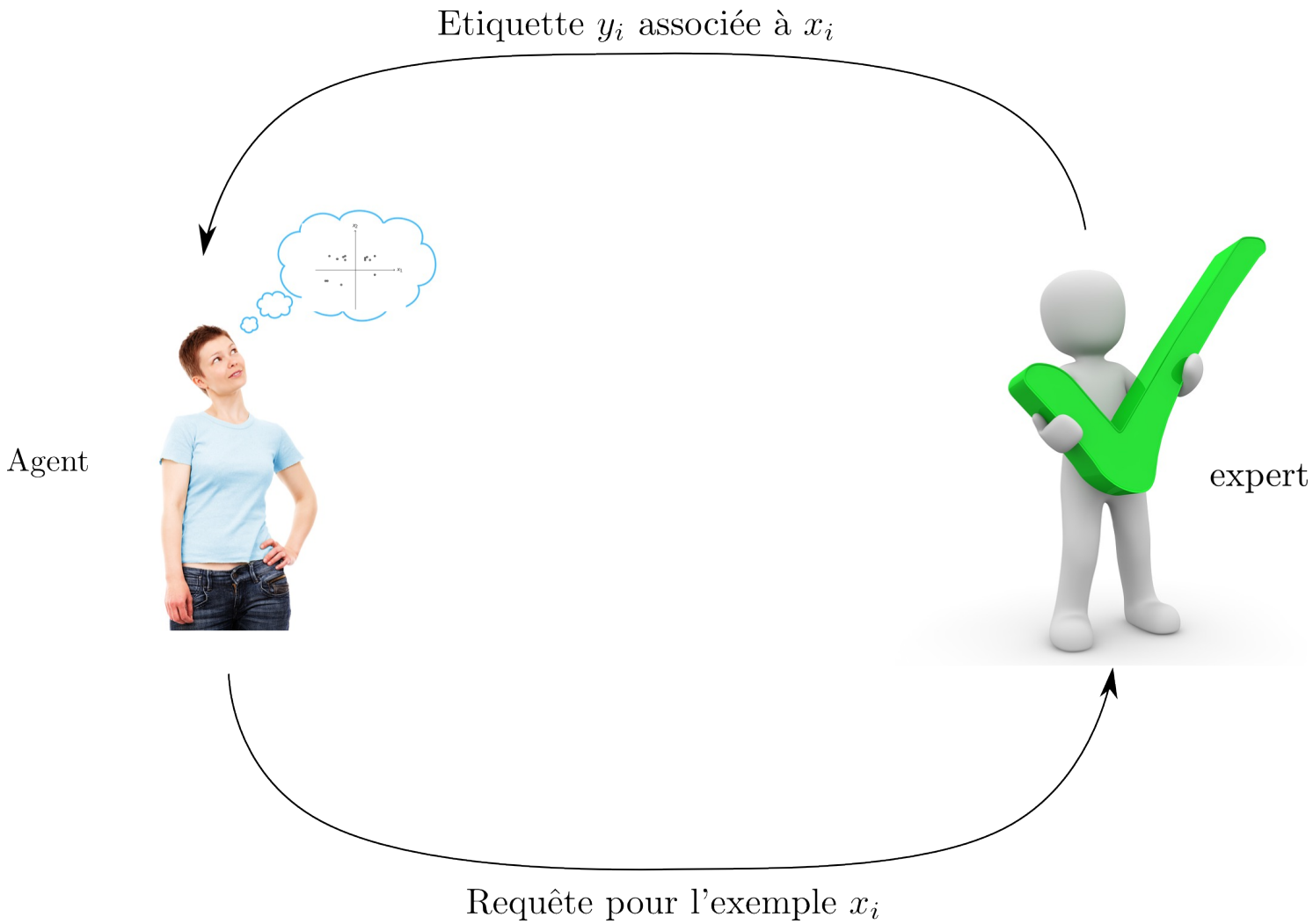
Paradigme n°3 : l'apprentissage actif :

- l'apprenant a le droit de demander n_0 parmi $n > n_0$ exemples.



Paradigme n°3 : l'apprentissage actif :

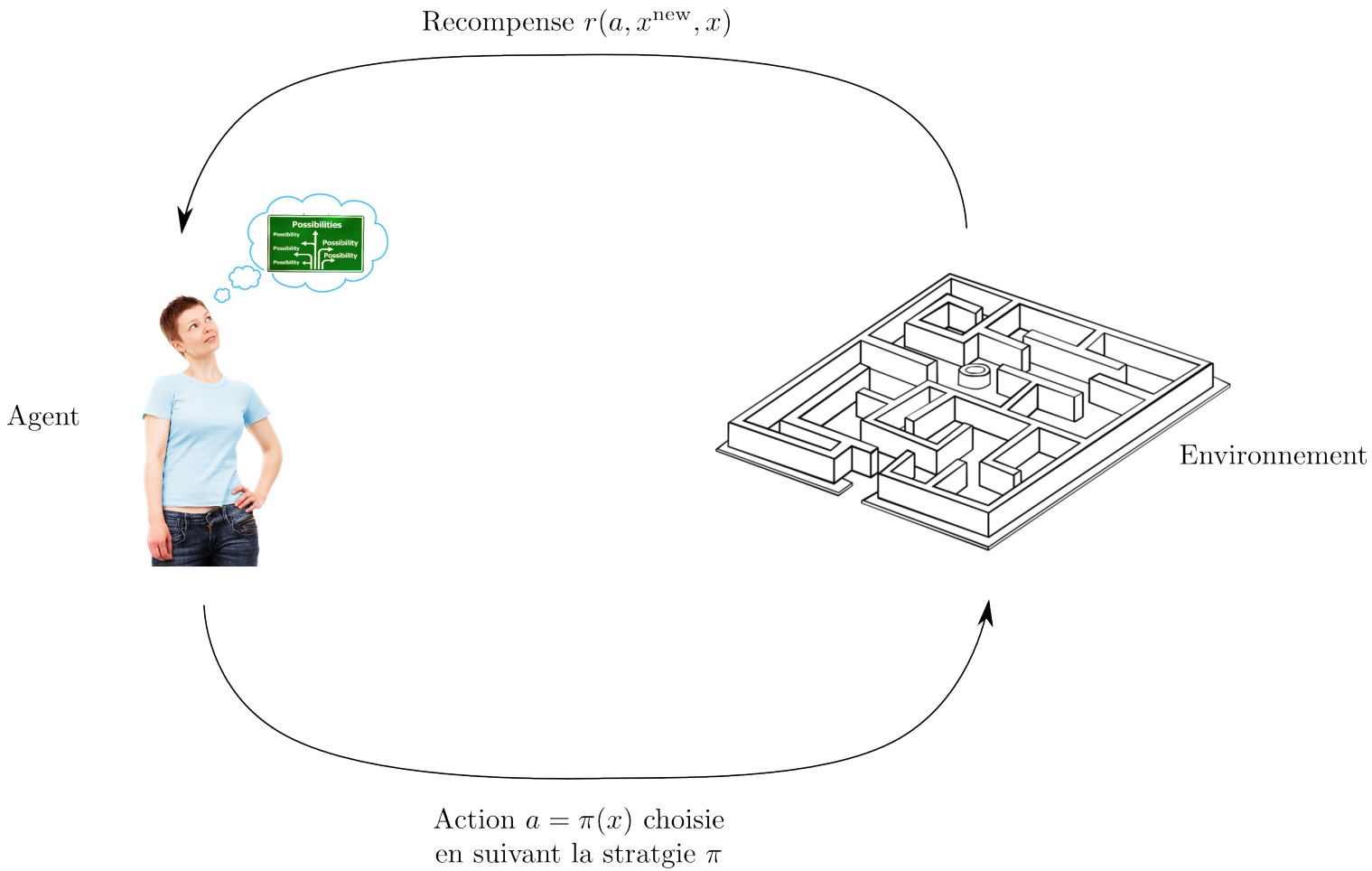
- l'apprenant a le droit de demander n_0 parmi $n > n_0$ exemples.



(images pixabay.com)

Paradigme n°4 : l'apprentissage par renforcement :

- Ce principe est proche d'un principe bien connu en psychologie animale.



(images pixabay.com)

Paradigme n°4 : l'apprentissage par renforcement :

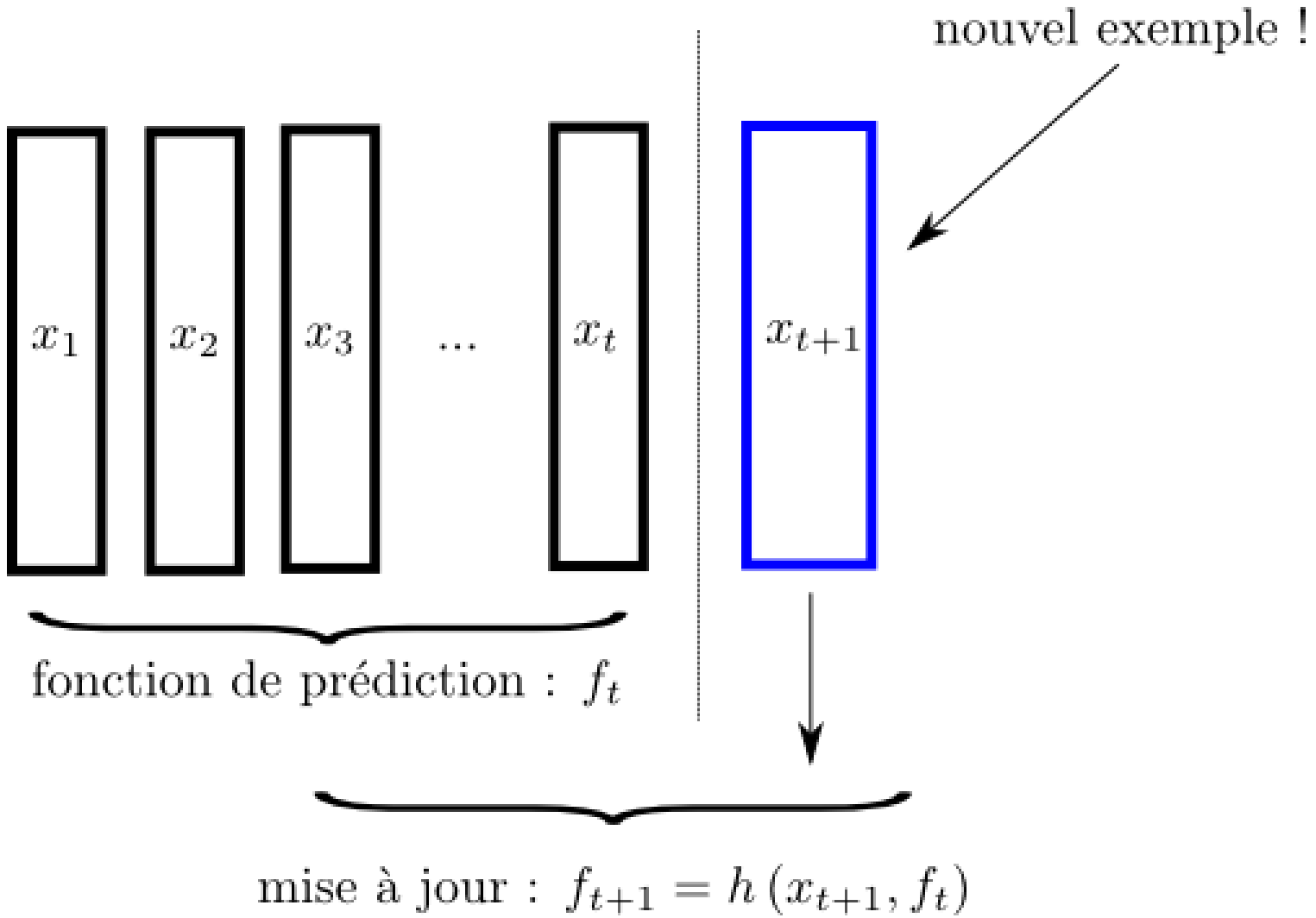
- Ici, on ne dispose pas vraiment d'un historique d'exemples au départ, mais on va s'en construire un en testant différentes solutions et en observant si j'obtiens satisfaction.
- L'apprentissage par renforcement ne s'applique pas à toute situation, il faut que les expériences soient répétables à souhait.

  Par exemple, si un médecin doit choisir entre différents protocoles de soins, on ne peut pas créer un nouveau patient pour rajouter un exemple ou repartir de la situation initiale.

  Par contre, si j'ai un système commandable et un signal de consigne à atteindre, je peux tenter plusieurs commandes, et je serai d'autant plus satisfait que ma consigne a été atteinte rapidement.

Paradigme n°5 (transversal) : l'apprentissage incrémental :

- Il s'agit d'apprendre dans un contexte où les données arrivent au fur et à mesure (streaming).



Plan du chapitre

Algorithmes paramétriques vs algorithmes non-paramétriques :

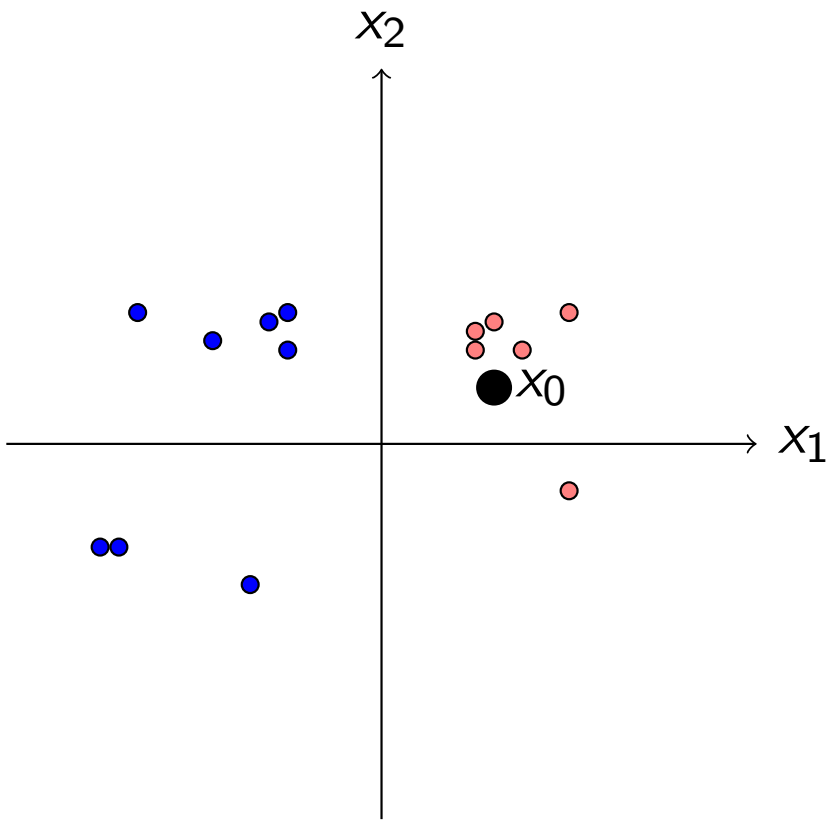
- Contrairement à l'intuition, on dit d'un modèle qu'il est non paramétrique si le nombre de paramètres à régler pour l'utiliser ne croit pas avec la dimension des données (mais il possède bel et bien des paramètres magré tout).

- Les paramètres qui ne sont pas liés à la dimension des données sont appelés hyper-paramètres qu'on notera $\boldsymbol{\lambda}$.

- Pour un algorithme paramétrique, la fonction de prédiction f a une forme prédéterminée entièrement définie par les autres paramètres qu'on notera $\boldsymbol{\theta}$.
  Chercher la bonne valeur des paramètres $\boldsymbol{\theta}$ = apprendre.

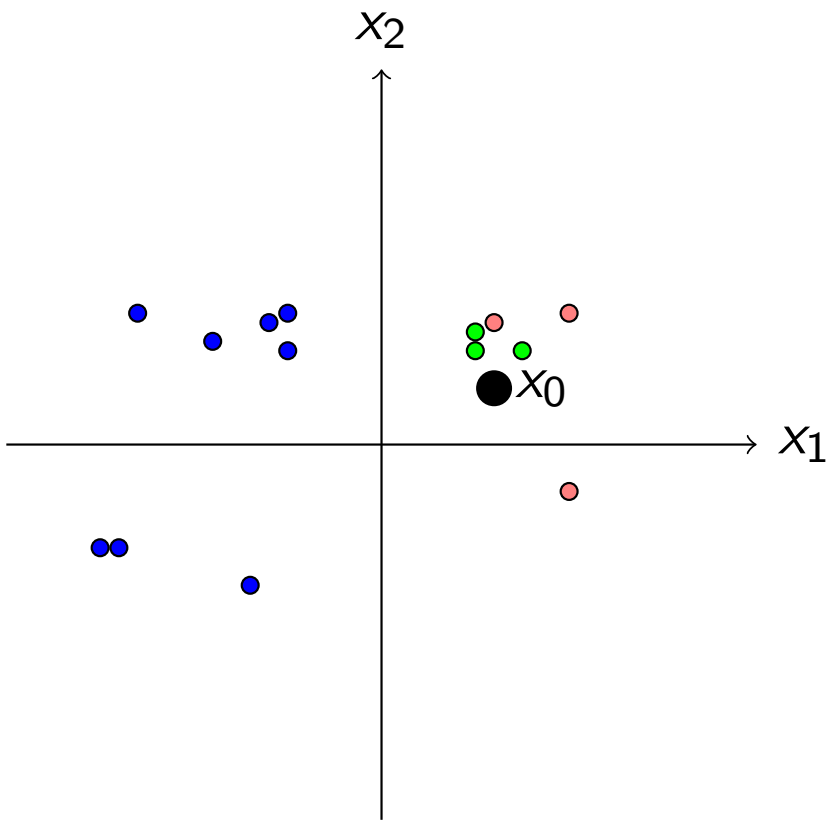
Exemple d'algorithme de classification supervisée non-paramétrique : k-PPV

- les exemples connus sont ceux appartenant à l'ensemble d'apprentissage $\mathcal{D}$. Un nouvel exemple "non-vu" précedemment arrive : x_0. Plaçons cet exemple dans $\mathbb{X}$:



k-PPV : principe (suite)

- Supposons que $k = 3$, en utilisant la distance euclidienne, on sait trouver les 3 plus proches voisins de x_0 :



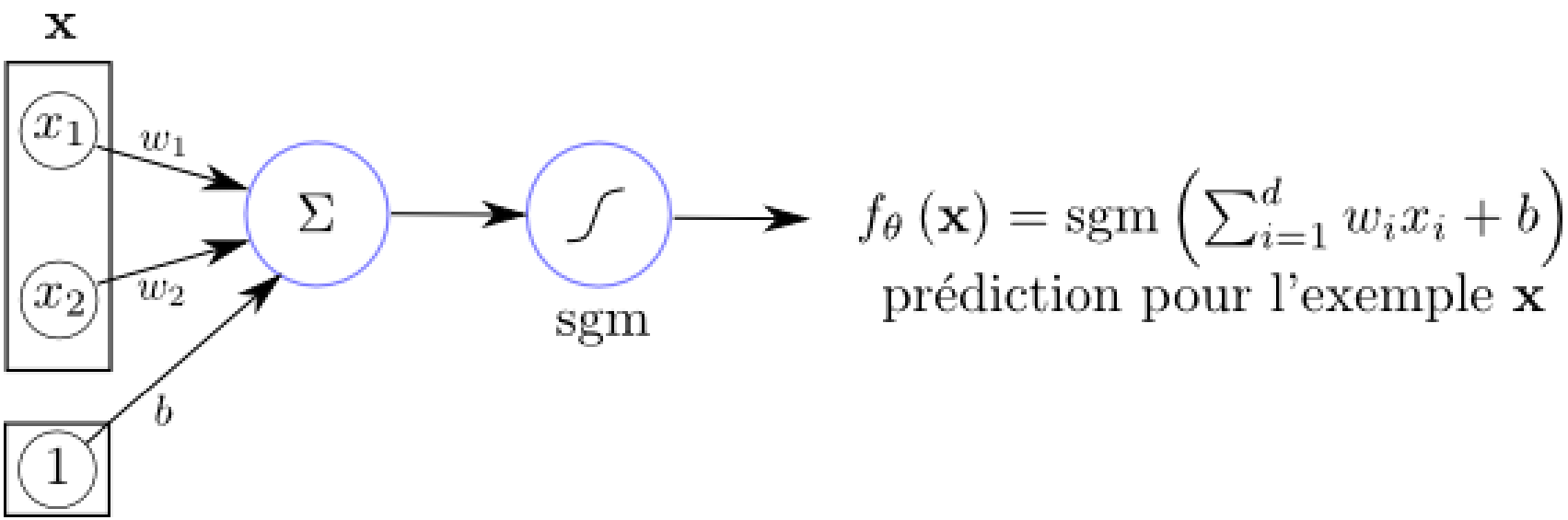
k-PPV : principe (suite)

- Parmi les 3 voisins, on procède à un vote à la majorité : sans ambiguité, on prédit que x_0 appartient à la classe *red*.
- On choisit souvent k impair pour éviter des *ex aequo* dans le vote.

- Quels sont les hyper-paramètres de k-PPV ?
- Que se passe-t-il si on passe à $\dim(\mathbb{X}) = 3$?
- Choisir $k = 1$ est risqué car si une erreur a été faite lors de la mesure d'un exemple, celui-ci va fausser toutes les prédictions d'exemples proche de lui. On parle de sur-apprentissage ou overfitting.
- Choisir k très grand revient à toujours choisir la classe qui a le plus de représentant dans $\mathcal{D}$, ce qui intuitivement n'est pas bon. On parle à l'inverse de sous-apprentissage ou underfitting.

Exemple d'algorithme de classification supervisée paramétrique : réseau de neurones

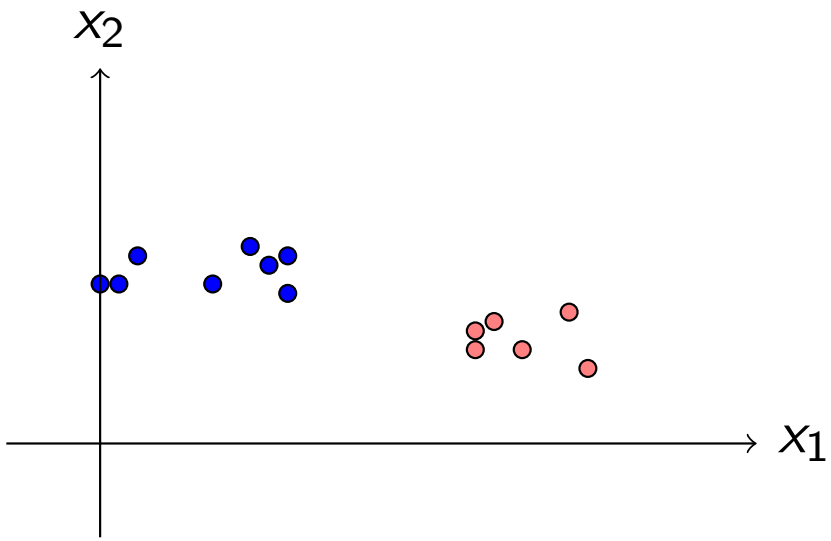
- On suppose à un problème à 2 classes : $\mathcal{C} = \{c_1, c_2\}$.
- Chaque neurone effectue une projection affine suivie d'une non-linéarité.



$\boldsymbol{\theta} =$

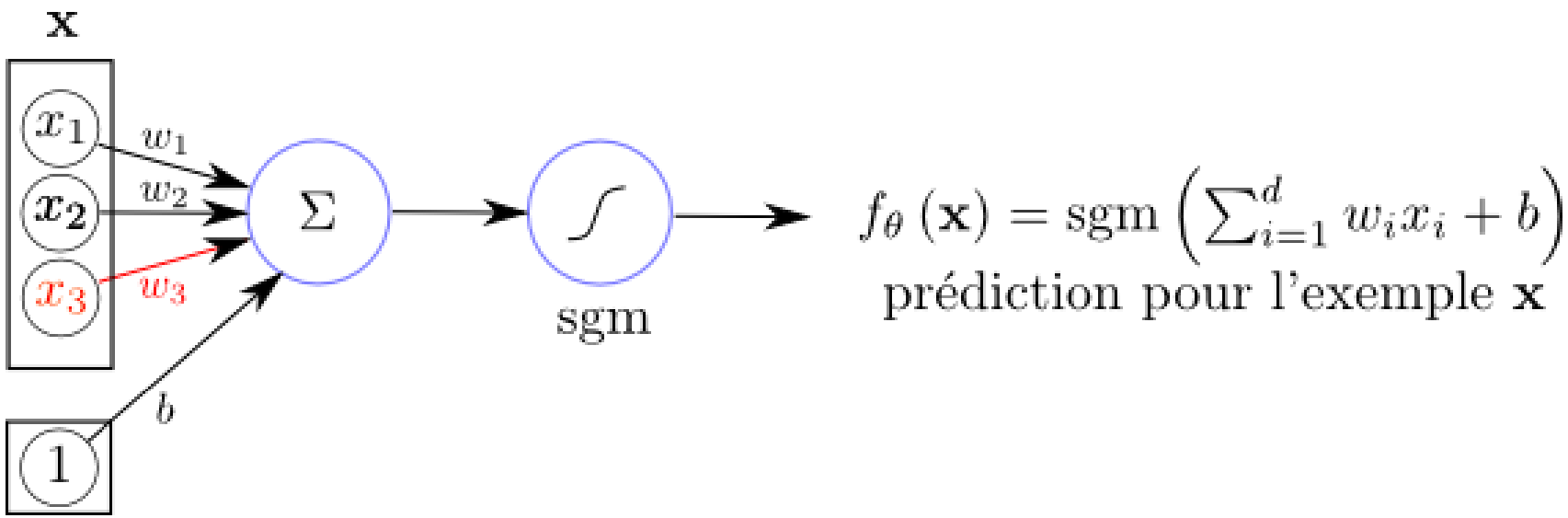
$\boldsymbol{\lambda} =$

Exemple d'algorithme de classification supervisée paramétrique : réseau de neurones



Exemple d'algorithme de classification supervisée paramétrique : réseau de neurones

- Que se passe-t-il si on passe à $\dim(\mathbb{X}) = 3$?

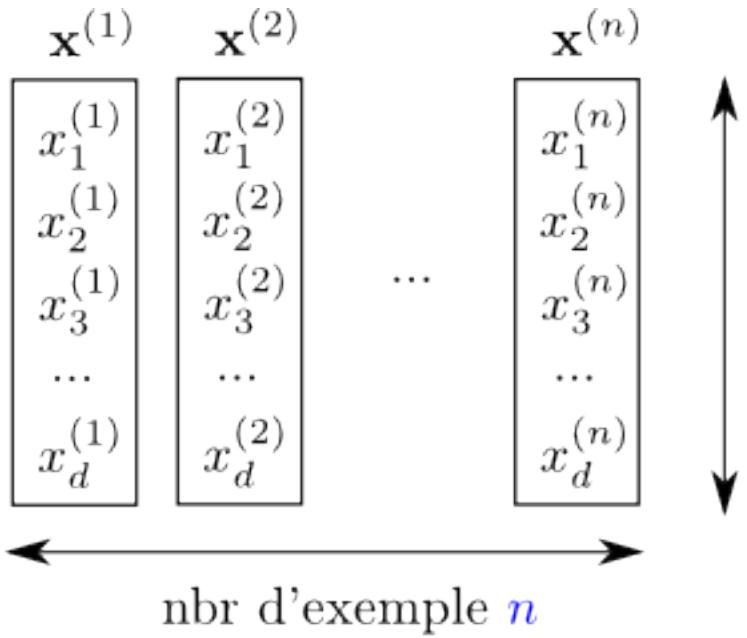


Plan du chapitre

Challenge n°1 : les données $\mathcal{D}$.

- En supervisé, $\mathcal{D} = \left\{ \left(\mathbf{x}^{(i)}, y^{(i)} \right) \right\}_{i=1}^{n} = (\mathbf{X}, \mathbf{y})$

matrice des exemples $\mathbf{X}=$



vecteur des étiquettes $\mathbf{y}^T = \left(y^{(1)} \; y^{(2)} \; \ldots \; y^{(n)} \right)^T$

Challenge n°1 : les données $\mathcal{D}$.

- En non-supervisé, $\mathcal{D} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n} = \mathbf{X}$

matrice des exemples $\mathbf{X}=$

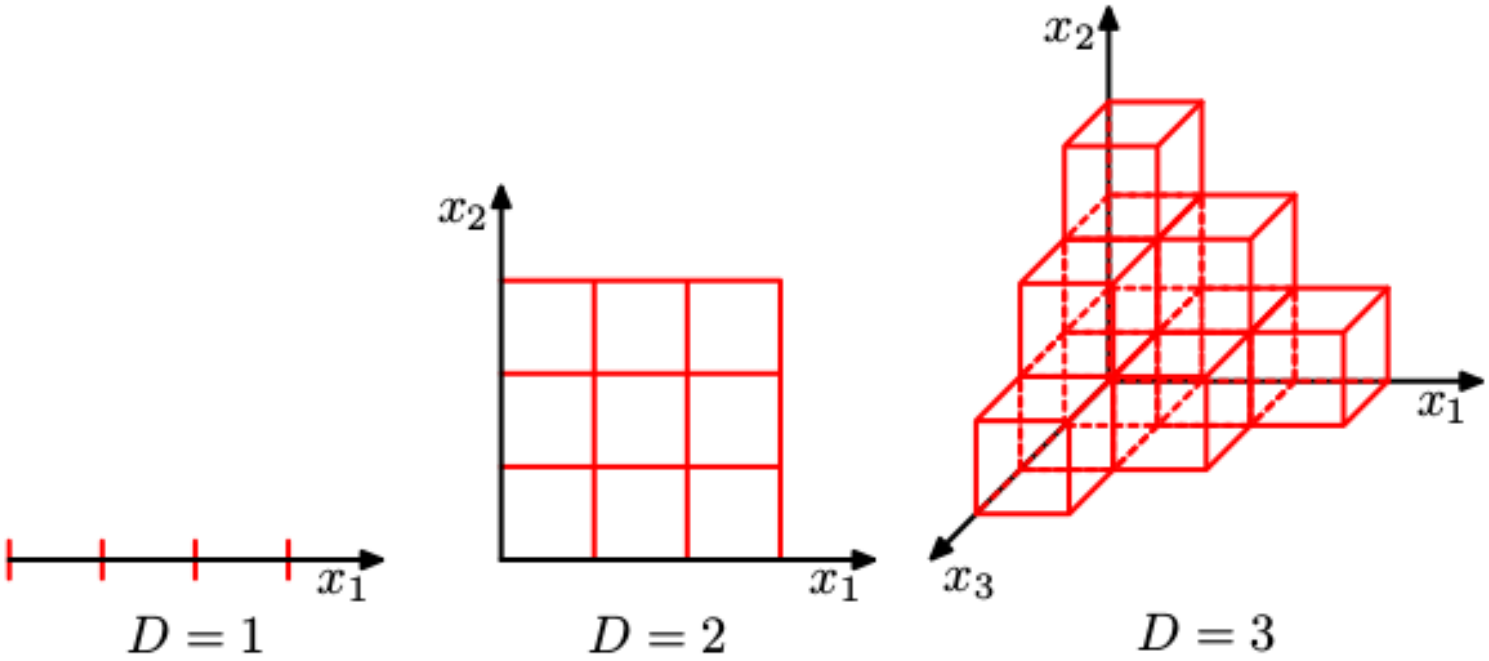
$$\begin{array}{c|ccc|c} \mathbf{x}^{(1)} & \mathbf{x}^{(2)} & & \mathbf{x}^{(n)} & \\ \hline x_1^{(1)} & x_1^{(2)} & & x_1^{(n)} & \\ x_2^{(1)} & x_2^{(2)} & & x_2^{(n)} & \\ x_3^{(1)} & x_3^{(2)} & \cdots & x_3^{(n)} & \text{dimension } d \\ \cdots & \cdots & & \cdots & \\ x_d^{(1)} & x_d^{(2)} & & x_d^{(n)} & \end{array}$$

nbr d'exemple n

~~vecteur des étiquettes $\mathbf{y}^T = \left(y^{(1)}\ y^{(2)}\ \ldots\ y^{(n)}\right)^T$~~

Challenge n°1.1 : la malédiction de la dimension

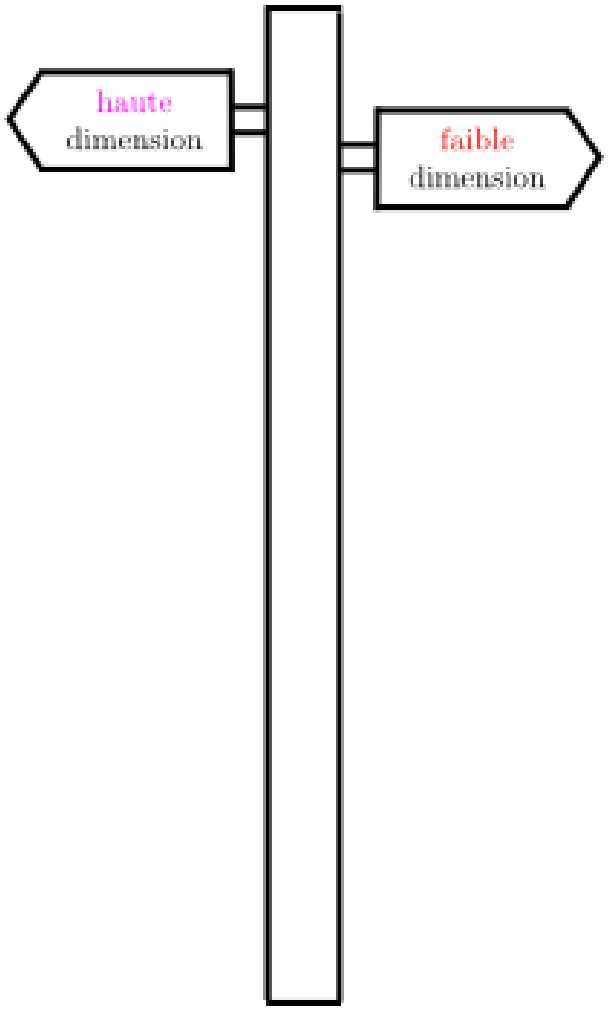
- elle touche aussi bien les algos non-paramétriques que paramétriques.



[Bishop 2006]

Challenge n°1.1 : la malédiction de la dimension

Risque de manque de quantité d'informations. (besoin d'un n grand)



Risque de manque de qualité d'informations. (pouvoir discriminant des dimensions choisies)

Challenge n°1.1 : la malédiction de la dimension

Des raisons d'espérer quand même :

- Même si d est grand, les données vivent en général dans une région confinée de $\mathbb{X}$, il n'est donc pas nécessaire d'avoir des points dans tout l'espace $\mathbb{X}$.
- La pertinence de certaines dimensions peut être évaluée. On peut donc partir d'un espace $\mathbb{X}$ de grande dimension puis trouver une fonction $\phi : \mathbb{X} \longrightarrow \mathbb{V}$ qui calculera de nouveaux vecteurs dans un espace $\mathbb{V}$ de bien plus petite taille.
- Les entrées d'un vecteur $\mathbf{v} \in \mathbb{V}$ sont appelées attributs (features).
- Les entrées de $\mathbf{x}$ sont parfois appelées attributs brutes (raw features).

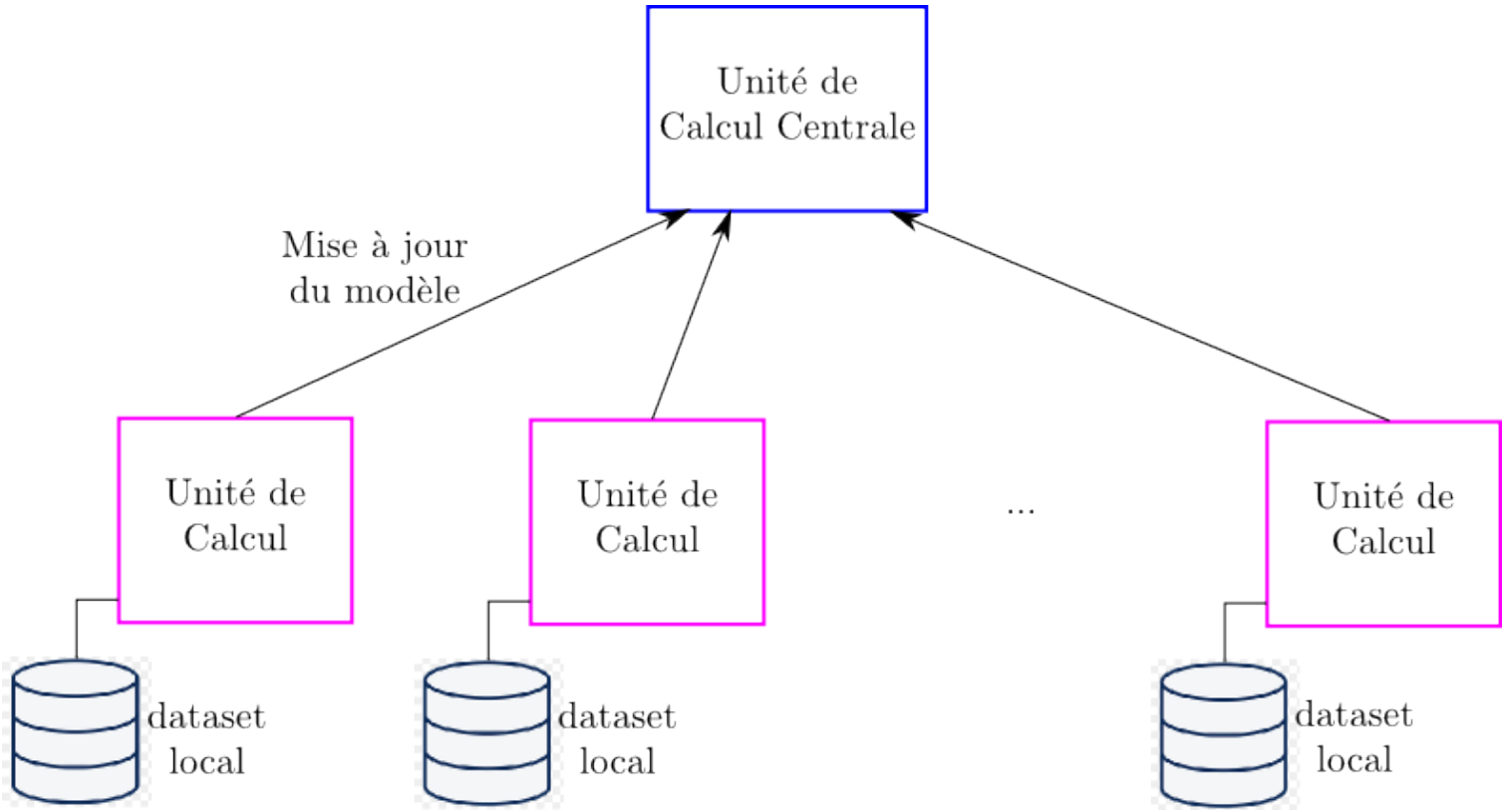
Challenge n°1.2 : données massives (Big Data, le vrai..)

- Les données massives posent des problèmes pragmatiques plutôt que théoriques.
- Certains datasets sont si volumineux qu'ils ne peuvent tenir sur une seule machine, ex : base Imagenet = 50 millions d'images de résolution moyenne 350x450 pixels $\approx$ 8To.

(images pixabay.com)

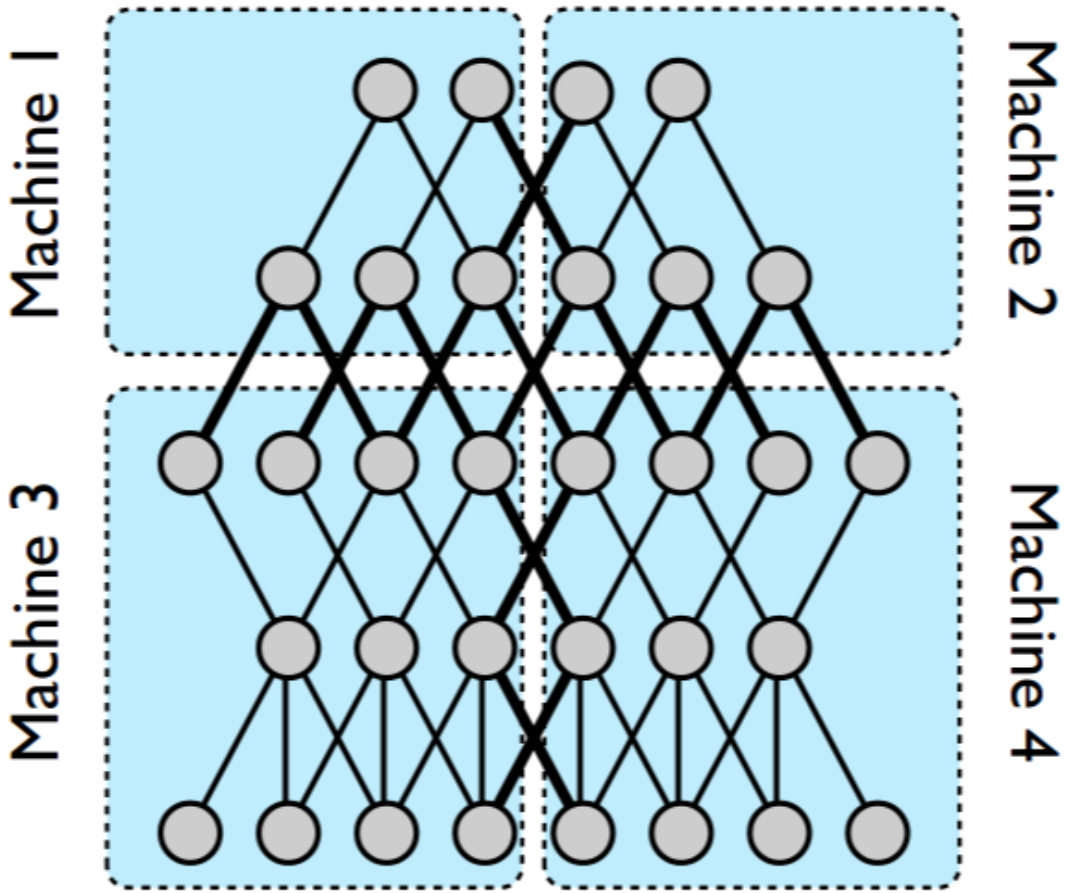
Challenge n°1.2 : données massives (Big Data, le vrai..)

- Certains datasets sont (par nature) distribués ou décentralisés sur différents datacenters (Google, Facebook, ..).
- Dans les 2 cas, la solution consiste à distribuer aussi les calculs.



Challenge n°1.2 : données massives (Big Data, le vrai..)

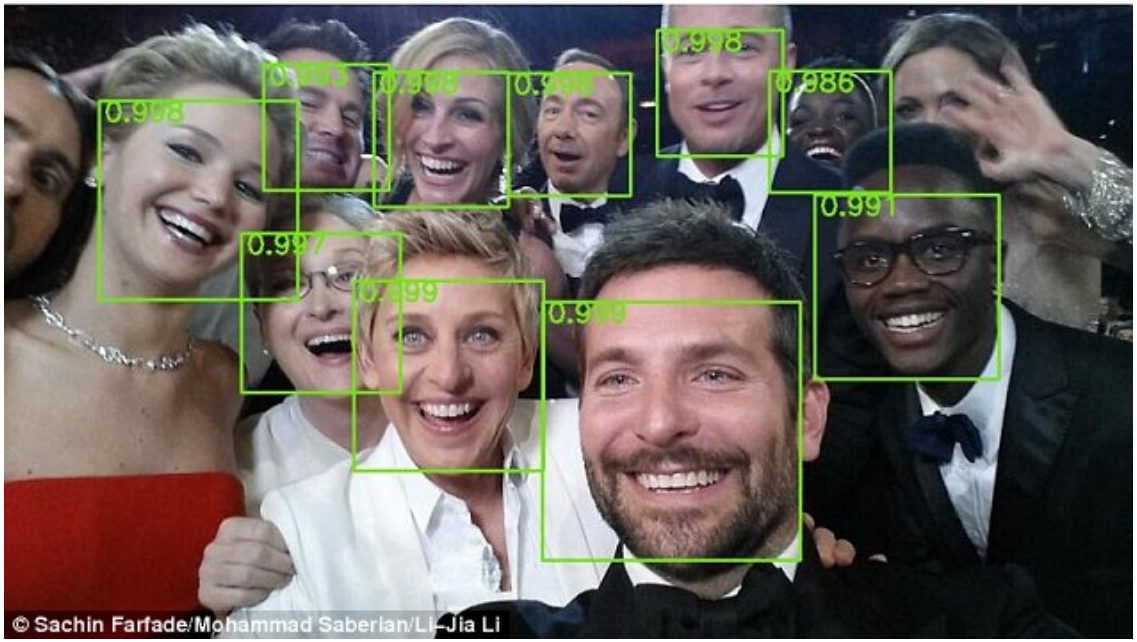
- Certains datasets sont (par nature) distribués ou décentralisés sur différents datacenters (Google, Facebook, ..).
- Dans les 2 cas, la solution consiste à distribuer aussi les calculs.



Dean, J. *et. al.*, Large scale distributed deep networks, NIPS, 2012.

Challenge n°2 : des tâches de plus en plus complexes

- Même problème que par le passé mais en plus dur ex : reconnaissance de visage.
- Facebook : 1 milliard d'utilisateurs = 1 milliard de classes ! !



Challenge n°2 : des tâches de plus en plus complexes

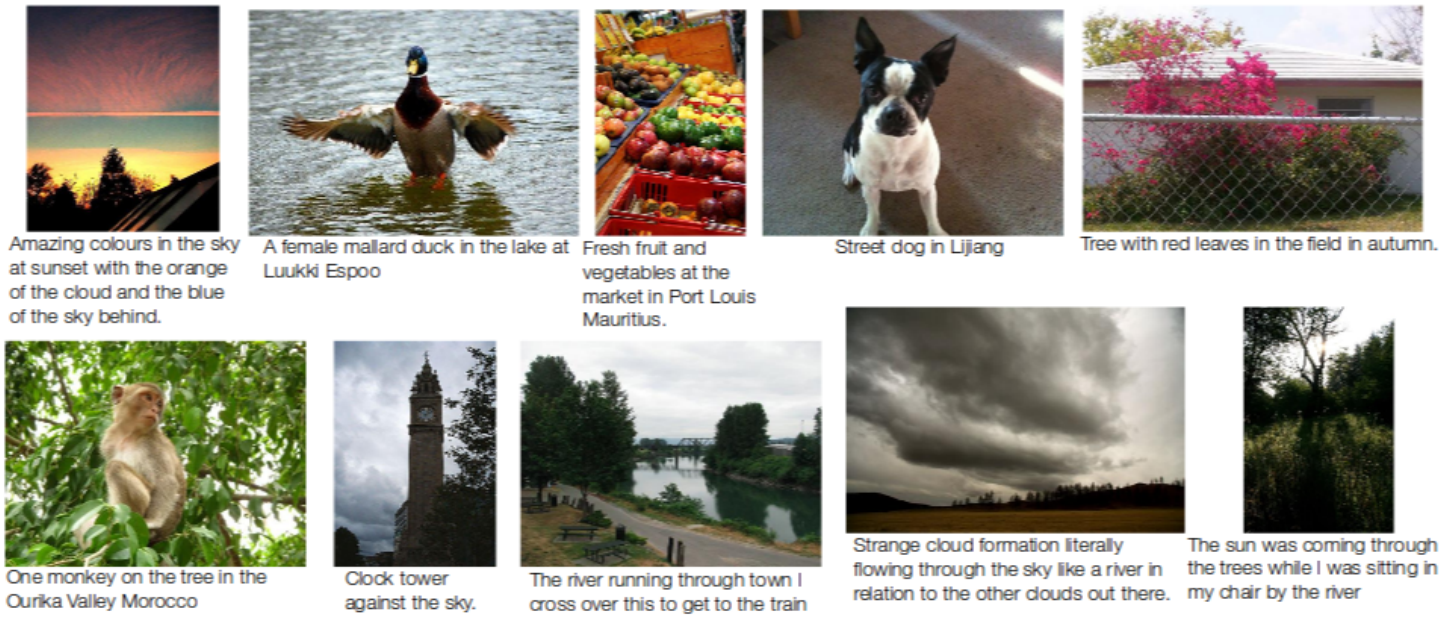
- des tâches hiérarchiques et/ou séquentielles : traduction automatique.

Type	Sentence
Our model	Ulrich UNK , membre du conseil d' administration du constructeur automobile Audi , affirme qu' il s' agit d' une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d' administration afin qu' ils ne soient pas utilisés comme appareils d' écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d' administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu' ils ne puissent pas être utilisés comme appareils d' écoute à distance , est une pratique courante depuis des années .

Sutskever, I. *et. al.*, Sequence to sequence learning with neural networks, NIPS 2014

Challenge n°2 : des tâches de plus en plus complexes

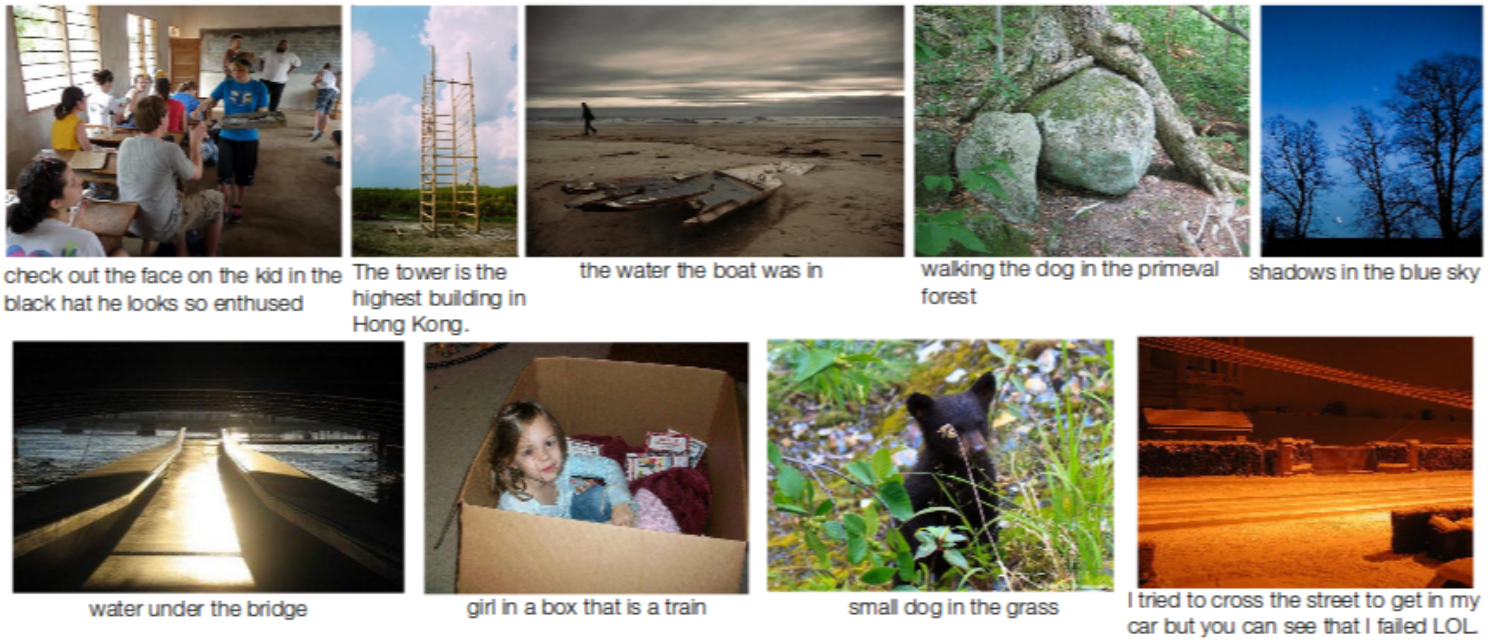
- des tâches abstraites : décrire une image.



Ordonez, V. *et. al.*, Im2text : describing images using 1 million captioned photographs, NIPS 2011

Challenge n°2 : des tâches de plus en plus complexes

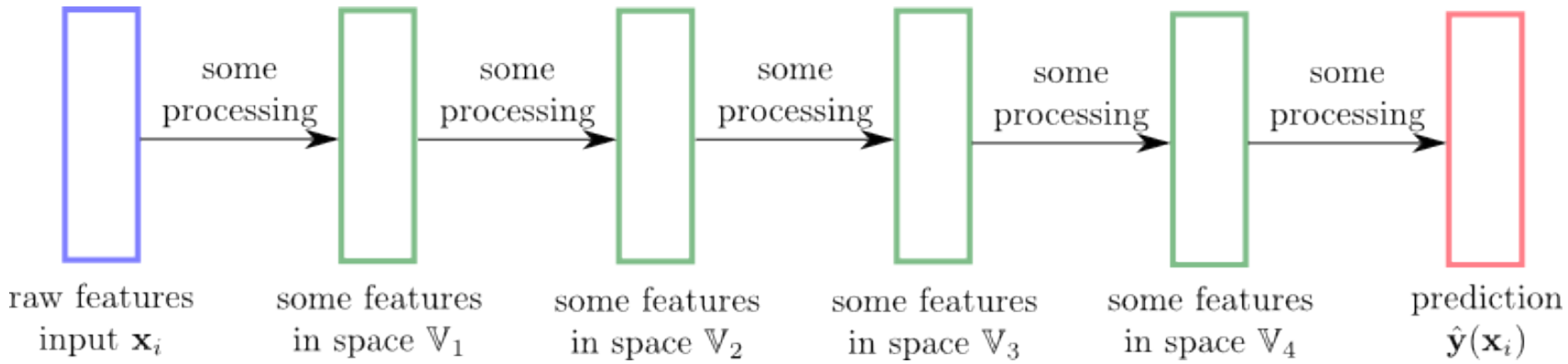
- des tâches abstraites : décrire une image.



Ordonez, V. *et. al.*, Im2text : describing images using 1 million captioned photographs, NIPS 2011

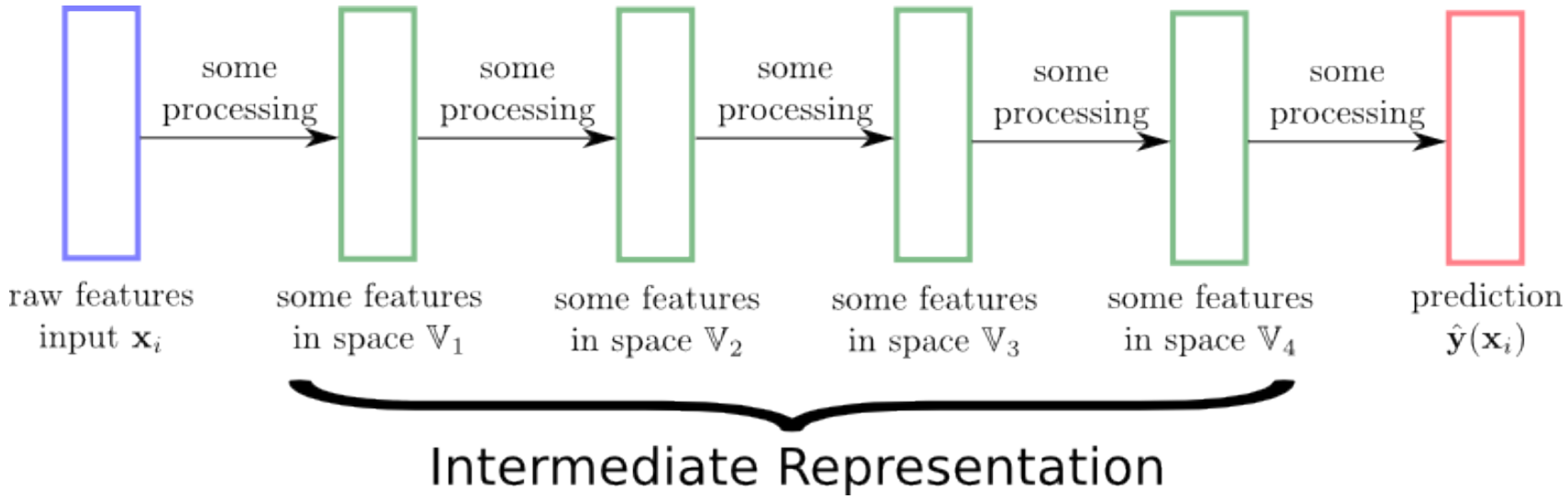
Challenge n°3 : des systèmes réutilisables

- avant : 1 tâche = 1 système
- maintenant/futur : apprentissage multi-tâches.
- Que se passe-t-il avec 1 tâche (supervisée) ?



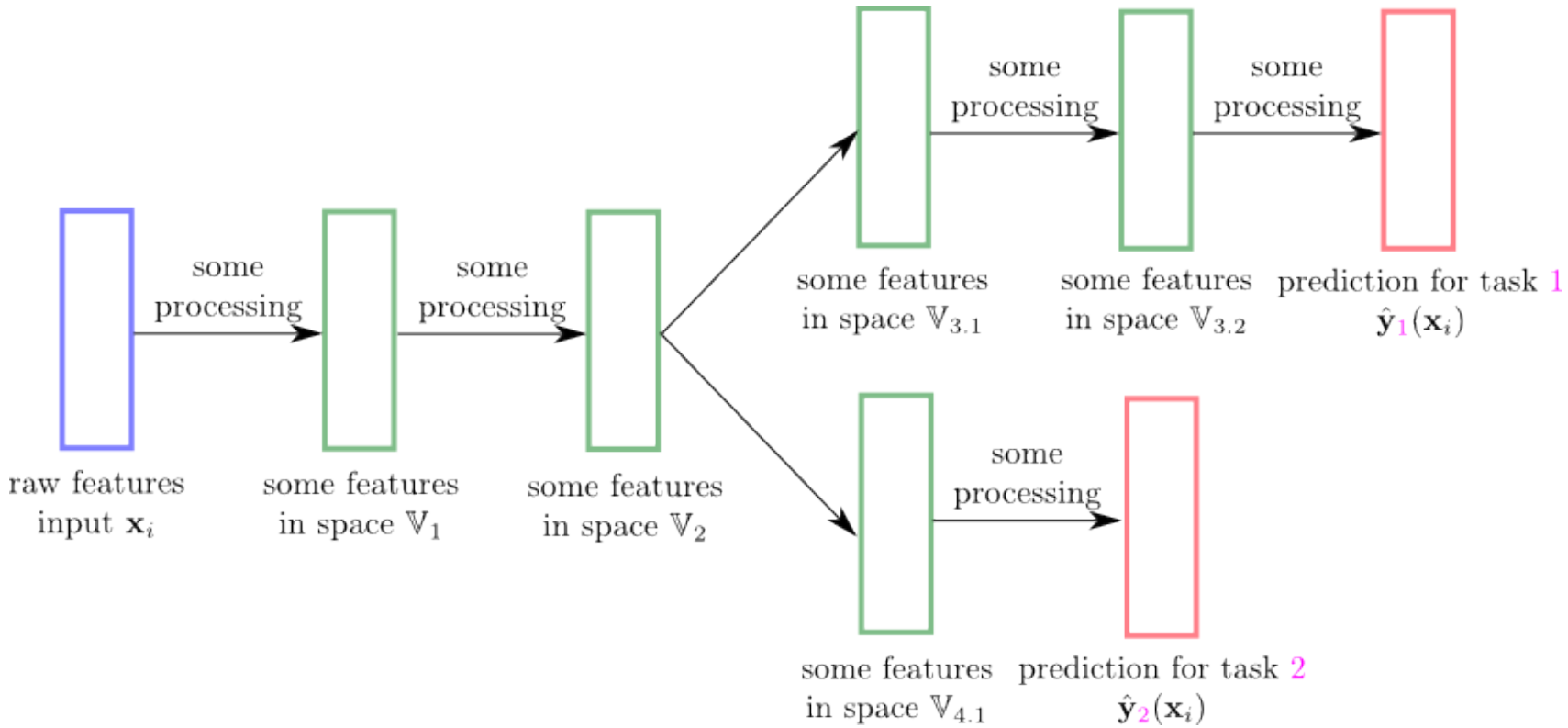
Challenge n°3 : des systèmes réutilisables

- avant : 1 tâche = 1 système
- maintenant/futur : apprentissage multi-tâches.
- Que se passe-t-il avec 1 tâche (supervisée) ?



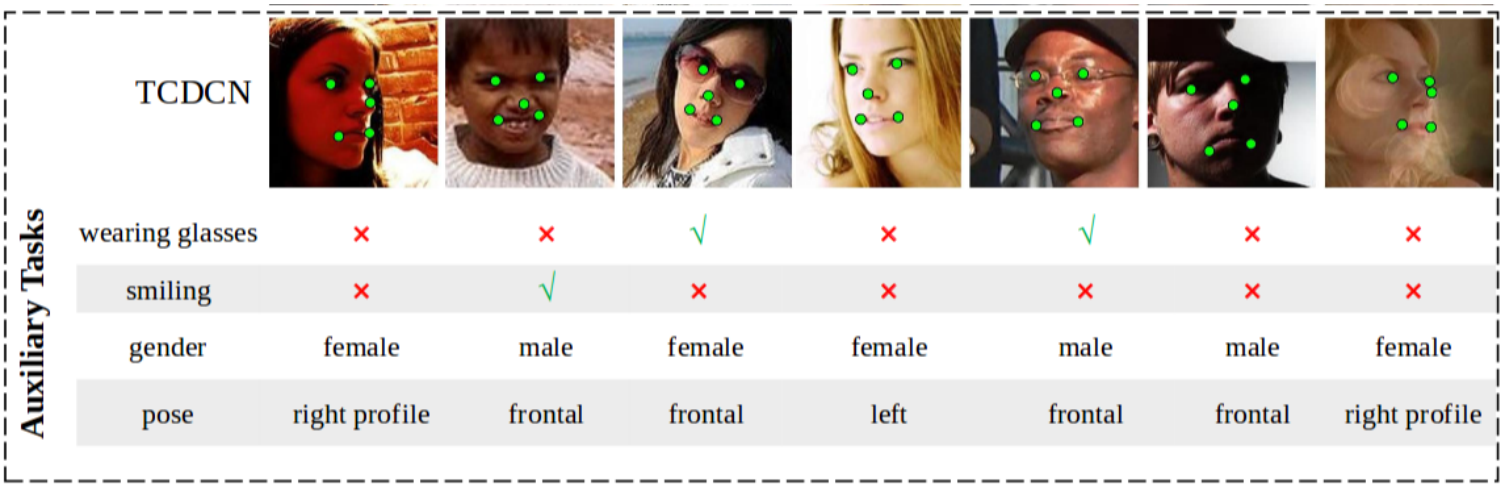
Challenge n°3 : des systèmes réutilisables

- Que se passe-t-il avec 2 tâches (supervisées) ?



Challenge n°3 : des systèmes réutilisables

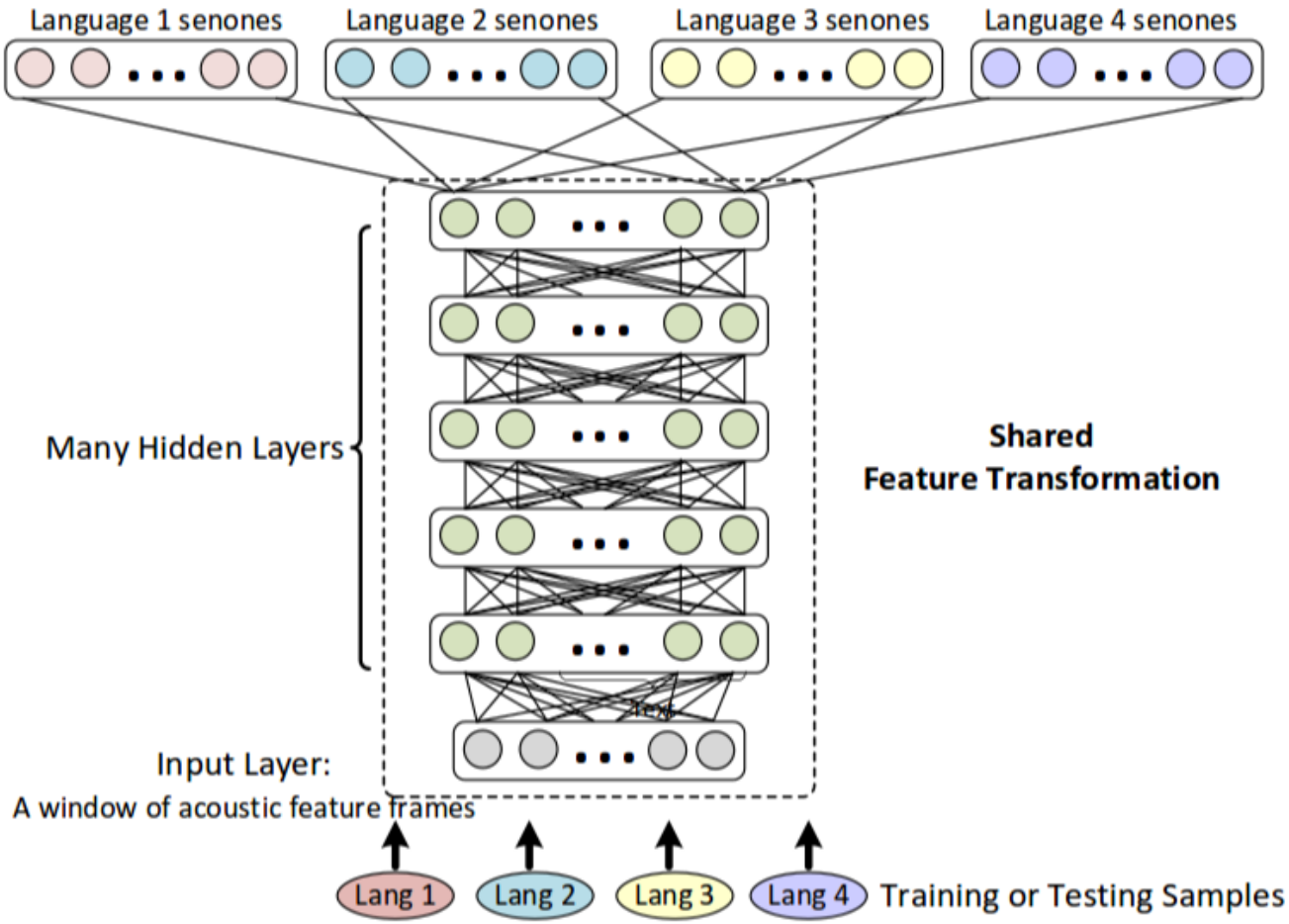
- Exemple d'apprentissage multi-tâches.



Zhang, Z. *et. al.*, Facial detection by deep multi-task learning, ECCV 2014

Challenge n°3 : des systèmes réutilisables

- Exemple d'apprentissage multi-tâches.



Huang, J. *et. al.*, Cross-language Knowledge Transfer using Multilingual Deep Neural Network with Shared Hidden Layers, ICASSP 2013

Challenge n°4 : des garanties théoriques

- Comment s'assurer du comportement moyen d'un algorithme ?
- Statistical Learning Theory :
  Soit $\hat{h} \in \mathcal{H}$ mon modèle obtenu en minimisant le risque empirique sur mes données.
  Soit $h^* \in \mathcal{H}$ le meilleur modèle possible parmi ceux que j'examine.
  Avec probabilité au moins $1 - \delta$, j'ai

$$\text{Risque}\left(\hat{h}\right) - \text{Risque}\left(h^*\right) \leq \text{borne.} \tag{1}$$

  La borne dépend de la complexité de $\mathcal{H}$, du nombre d'exemples d'apprentissage n et de δ.

Plan du chapitre

Déroulement :

- 18 séances de 3h
- 1 séance $\approx$ 1h30 de cours + 1h30 de TP

Evaluation :

- 50% Exercices sur machine (TP / comptétition)
- 50% Examen final (2 sessions)

Contenu :

- 70% de supervisé
- 10% de non supervisé
- 20% de renforcement

Livres :

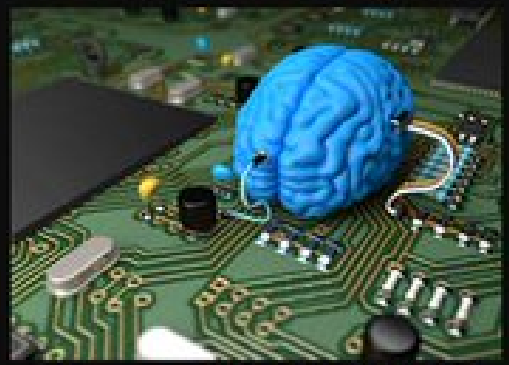
- Murphy, Machine Learning a probabilistic perspective, MIT press, 2012.
- Bishop, Pattern Recognition and Machine Learning, springer, 2006.
- Sutton, R. and Barto, A., Reinforcement Learning : An Introduction, MIT press, 1998.

Video :

- H. Larochelle (youtube channel)
- Y. Abu-Mostafa (youtube channel)
- http ://videolectures.net

Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

What I think I do

```
from theano import *
```

What I actually do