Apprentissage Artificiel
bases, questions directions

A. Cornuéjols

AgroParisTech – INRA MIA 518

Plan

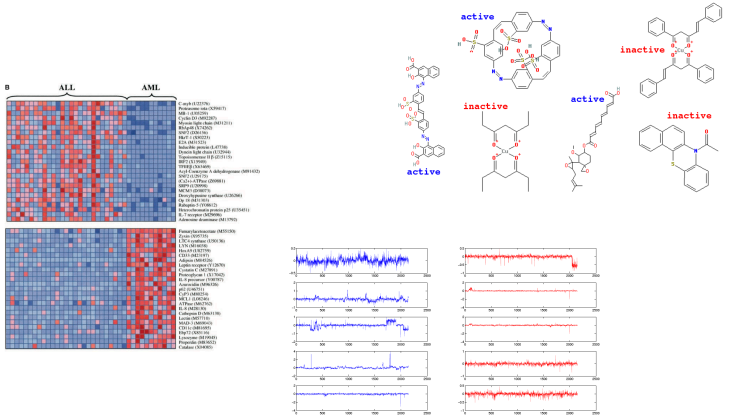
1. Notions de base sur l'Apprentissage Artificiel
2. Les nouvelles questions et directions
3. Activité à AgroParisTech

Notions de base sur l'Apprentissage Artificiel

- Apprentissage d'une **fonction** *Entrée -> Sortie* $(X \rightarrow Y)$
  - **Prédiction**
    - Cours de bourse $\Re \rightarrow \Re$
    - Prévision entretien pièce de centrale électrique $\Re^n \rightarrow \Re$
    - Météo structure → structure
  - **Classification**
    - Molécule → {active, inactive}
    - Puce à ADN → {cancer, pas cancer}
    - Série temporelle → {normale, anormale}

Bases sur l'apprentissage artificiel

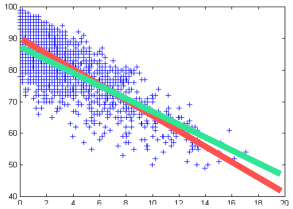
- À partir d'un échantillon d'exemples $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq m}$



Bases sur l'apprentissage artificiel

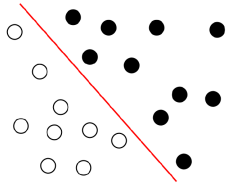
- Quelles fonctions ?
  - Modèles **linéaires**
    - ***Régression*** linéaire

$$h(\mathbf{x}) = \sum_{i=1}^{n} w_i \, \phi_i(\mathbf{x})$$



    - ***Classification*** linéaire

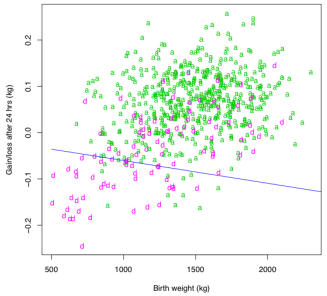
$$h(\mathbf{x}) = \text{sign}\left\{\sum_{i=1}^{d} w_i \phi_i(\mathbf{x})\right\}$$



Bases sur l'apprentissage artificiel

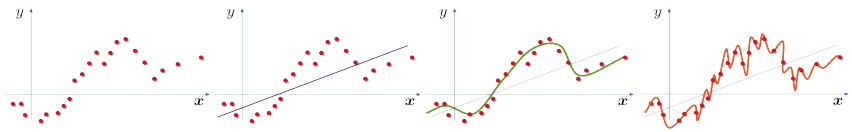
- Quelles fonctions ?
  - Modèles **linéaires**
    - ***Discrimination logistique***

$$\log \frac{\mathbf{p}(\mathbf{x}|\omega_1)}{\mathbf{p}(\mathbf{x}|\omega_2)} = \theta^\top \mathbf{x} + \theta_0$$

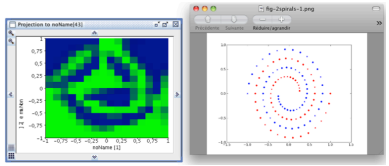


Bases sur l'apprentissage artificiel

- Quelles fonctions ?
  - Modèles **non linéaires**
    - ***Régression non linéaire***

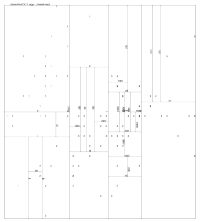


    - ***Classification non linéaire***



Bases sur l'apprentissage artificiel

- Quelles fonctions ?
  - Modèles **non linéaires**
    - ***Arbre de décision***



    - ***Expression logique***

$$\varphi_1 : \; anm(x_3, [195, 22, 3, 27, 38, 40, 92]) \wedge \neg chrg(x_3, [-0.2, 0.2]) \wedge anm(x_4, [195, 22, 3, 38, 40, 29, 92]) \wedge \neg type(x_4, [O]) \wedge \neg chrg(x_4, [-0.2]) \wedge (x_1 < x_2) \wedge (x_1 < x_3) \wedge (x_1 < x_4) \wedge (x_2 < x_3) \wedge (x_2 < x_4) \wedge (x_3 < x_4) \wedge bound(x_3, x_4) \rightarrow mutagenic,$$

$$\varphi_2 : \; \neg chrg(x_1, [-0.2]) \wedge \neg type(x_2, [N]) \wedge \neg anm(x_3, [22]) \wedge \neg chrg(x_3, [-0.6, -0.4]) \wedge \neg type(x_4, [H, N, O]) \wedge (x_1 < x_2) \wedge (x_1 < x_3) \wedge (x_1 < x_4) \wedge (x_2 < x_3) \wedge bound(x_2, x_3) \wedge (x_2 < x_4) \wedge (x_3 < x_4) \wedge bound(x_3, x_4) \rightarrow mutagenic,$$

Bases sur l'apprentissage artificiel

Sous quelles conditions l'induction est-elle valide ?

- Que signifie avoir appris ?
  - Être capable de **générer des échantillons de données** indistingables de l'échantillon « naturel »
    - Approche « **générative** »
    - Critère de Maximum de Vraisemblance
    - Critère de Maximum A Posteriori
  - Être capable de **prendre de bonnes décisions**
    - Approche « **discriminative** »
    - Critère de minimisation du risque réel

Bases sur l'apprentissage artificiel

Approche générative

Figure: Modèle de génération des exemples.

Prédictions correctes (la plupart du temps)

$$L(h) = \mathbf{P}_{\mathcal{X}\mathcal{Y}}\{h(\boldsymbol{x}) \neq y\}$$

MLE et MAP

Choisir l'hypothèse $\hat{h}$ telle que :

$$\hat{h} = \underset{h\in\mathcal{H}}{\text{ArgMax}}\, l(h) = \underset{h\in\mathcal{H}}{\text{ArgMax}}\, \ln\left[p(\mathcal{S}_m|h)\right] \quad \text{(MLE)}$$

$$\hat{h} = \underset{h\in\mathcal{H}}{\text{ArgMax}}\, p(\mathcal{S}_m|h)\, p(h) \quad \text{(MAP)}$$

Bases sur l'apprentissage artificiel

Approche discriminative

- Que signifie avoir appris ?
  - Généralisation = Minimisation du **risque réel**

$$R(h) = \mathbb{E}[\ell(h(\mathbf{x}), y)] = \int_{\mathbf{x}\in\mathcal{X}, y\in\mathcal{Y}} \ell(h(\mathbf{x}), y)\, \mathbf{P}_{\mathcal{X}\mathcal{Y}}\, d(\mathbf{x}, y)$$

inconnu

Critère inductif : **Minimisation du risque empirique**

$$h^\star = \underset{h\in\mathcal{H}}{\text{ArgMin}}\, R_m(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)$$

Bases sur l'apprentissage artificiel

- La minimisation du risque réel par celle du risque empirique est **un problème mal-posé**
  - Il faut ajouter des **contraintes** sur l'**espace des hypothèses**

Contrôler $d_{\mathcal{H}}$

1. "Sélection de modèle"
2. Puis choix de $h \in \mathcal{H}$

$$\hat{h} = \text{ArgMin}_{h\in\mathcal{H}}\left[R_{Emp}(h) + \text{Capacité}(\mathcal{H})\right]$$

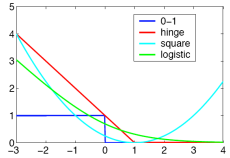
Régularisation

- Contrôler directement la complexité de h

$$\hat{h} = \text{ArgMin}_{h\in\mathcal{H}}\left[R_{Emp}(h) + \lambda\, \text{Reg}(h)\right]$$

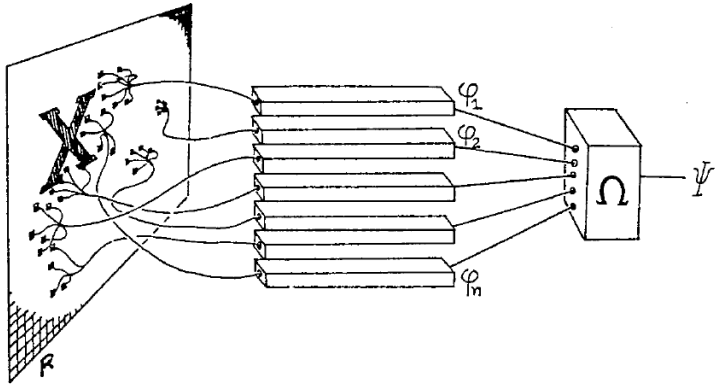
Bases sur l'apprentissage artificiel

- Plus généralement :
  - On traduit les **connaissances *a priori*** en contraintes dans le critère inductif
    - E.g. **Parcimonie** : norme l_0 sur l'expression de l'hypothèse
    - Notion de voisinage dans un graphe
  - On s'arrange pour avoir un **critère convexe**
    - Un seul optimum global
    - Fonctions de coût de substitution (« surrogate functions »)



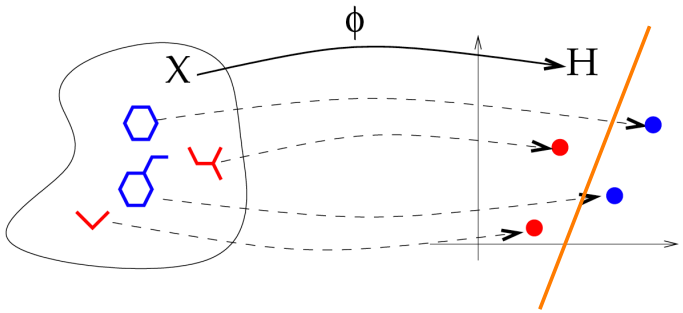
Bases sur l'apprentissage artificiel

Le choix des descripteurs *: comment faire ?*



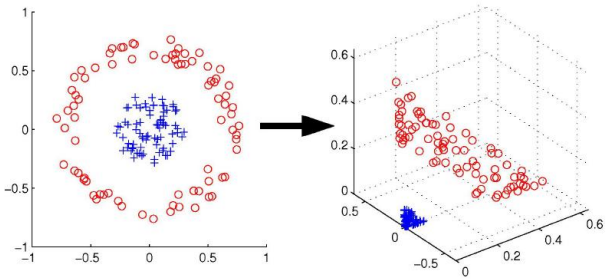
Bases sur l'apprentissage artificiel

- Changement de représentation idéal



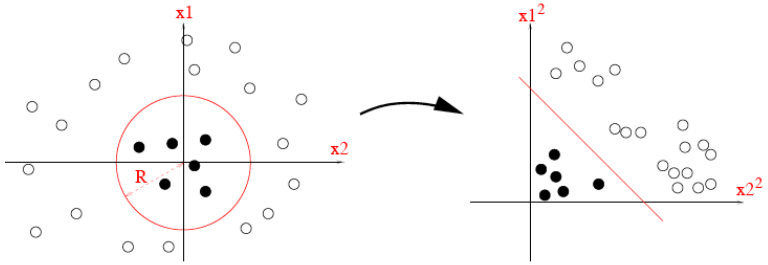
Bases sur l'apprentissage artificiel

- Changement de représentation idéal



Bases sur l'apprentissage artificiel

- Changement de représentation idéal

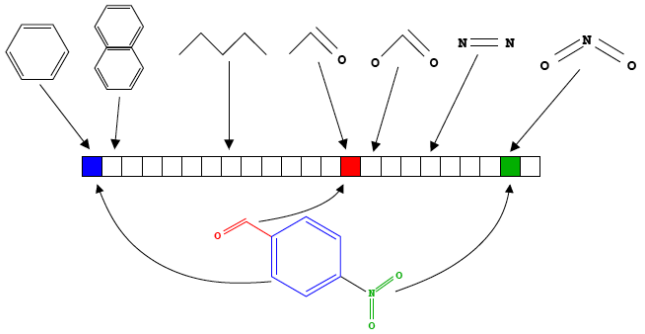


Let $\phi(\mathbf{x}) = (x_1^2, x_2^2)'$, $\mathbf{w} = (1,1)'$ and $b = 1$. Then the decision function is:

$$f(\mathbf{x}) = x_1^2 + x_2^2 - R^2 = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b,$$

Bases sur l'apprentissage artificiel

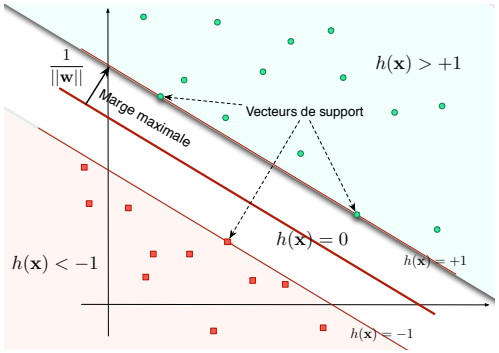
- Changement de représentation



Bases sur l'apprentissage artificiel

- **Séparateurs à Vastes Marges** (SVM)

La marge permet de contrôler la richesse de l'espace des hypothèses



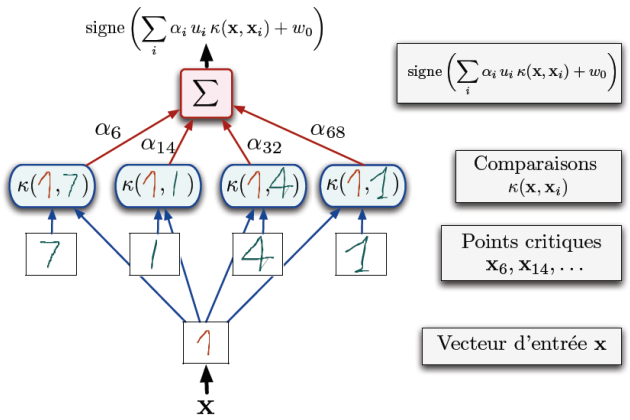
Bases sur l'apprentissage artificiel

- **Séparateurs à Vastes Marges** (SVM)

$$h^*(\mathbf{x}) = (\mathbf{w}^\star\ \mathbf{x}) + w_0^\star = \sum_{i=1}^{m} \alpha_i^\star\, u_i\, .\, \langle \mathbf{x}_i\ ,\ \mathbf{x} \rangle + w_0^\star$$

- Représentation ***non paramétrique*** : fonction des exemples d'apprentissage
  - La complexité de l'hypothèse dépend de $|\mathcal{S}|$
- **Parcimonieux**
  - Seulement les $m' \leq m$ exemples support
  - Solution la plus robuste (large marge) $\Rightarrow$ donc nécessite moins de précision (le moins de bits)
- Ne dépend pas de d, mais de m'

- **Séparateurs à Vastes Marges** (SVM)



signe $\left(\sum_i \alpha_i u_i \kappa(\mathbf{x}, \mathbf{x}_i) + w_0\right)$

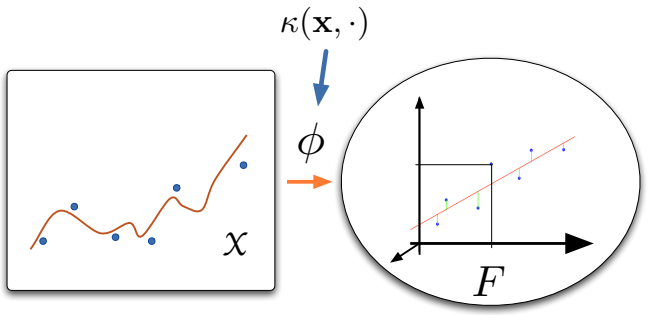
Comparaisons $\kappa(\mathbf{x}, \mathbf{x}_i)$

Points critiques $\mathbf{x}_6, \mathbf{x}_{14}, \ldots$

Vecteur d'entrée $\mathbf{x}$

- **Séparateurs à Vastes Marges** (SVM)

$$h(\mathbf{x}) = \text{sign}\left\{\sum_{i=1}^{m} \alpha_i^\star \, u_i \, \kappa(\mathbf{x}, \mathbf{x}_i) + w_0^\star\right\}$$



- **Les fonctions noyau**

- A kernel k is a function

$$\begin{array}{rccc} k: & \mathcal{X} \times \mathcal{X} & \longmapsto & \mathbb{R} \\ & (\mathbf{x}, \mathbf{y}) & \longrightarrow & k(\mathbf{x}, \mathbf{y}) \end{array}$$

- which compares two objects of a space $\mathcal{X}$, *e.g.*...
  - strings, texts and sequences,
  - images, audio and video feeds,
  - graphs, interaction networks and 3D structures
- whatever actually... time-series of graphs of images? graphs of texts?...

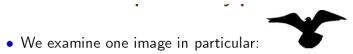
- **Les fonctions noyau**

- Suppose we have a kernel k on bird images

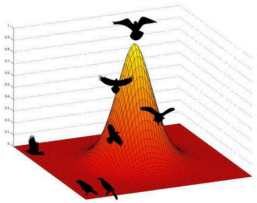
- Suppose for instance

$k\left(\right.$  ,  $\left.\right) = .32$

Les fonctions noyau



- We examine one image in particular:
- With kernels, we get a **representation** of that bird as a real-valued function, defined on the space of birds, represented here as $\mathbb{R}^2$ for simplicity.



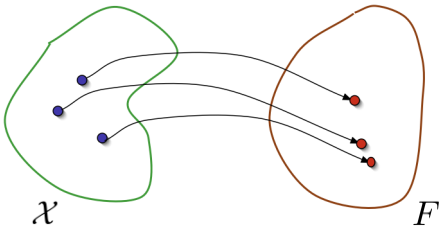
schematic plot of $k($  $, \cdot\ )$.

Les fonctions noyau

- Fonction ***k*** telle que :

$$k : \mathcal{X} \times \mathcal{X} \to \Re$$
$$\forall \boldsymbol{x}, \boldsymbol{z} \in \mathcal{X} : \quad k(\boldsymbol{x}, \boldsymbol{z}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle$$

où : $\phi : \boldsymbol{x} \mapsto \phi(\boldsymbol{x}) \in F$ Espace de redescription muni d'un produit interne



Les fonctions noyau

Soit : $\mathcal{X} \subseteq \Re^2$

$$\phi : \boldsymbol{x} = (x_1, x_2) \mapsto \phi(\boldsymbol{x}) = (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2) \in F = \Re^3$$

$$h(\boldsymbol{x}) = w_{11}\, x_1^2 + w_{22}\, x_2^2 + w_{12}\, \sqrt{2}\, x_1 x_2$$

$$\begin{aligned} \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle &= \langle (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2), (z_1^2, z_2^2, \sqrt{2}\, z_1 z_2) \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2 x_1 x_2 z_1 z_2 \\ &= (x_1 z_1 + x_2 z_2)^2 = \langle x, z \rangle^2 \end{aligned}$$

$$k(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle^2$$ **est une fonction noyau**

- Rq (non unicité de l'espace F défini par $\boldsymbol{\Phi}$) :

$$\phi : \boldsymbol{x} = (x_1, x_2) \mapsto \phi(\boldsymbol{x}) = (x_1^2, x_2^2, x_1 x_2, x_2 x_1) \in F = \Re^4$$

(le même noyau calcule le produit interne dans cet espace aussi)

Fonctions noyau pour des vecteurs

- **Noyaux polynomiaux**

$$k_{\texttt{poly1}}(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z})^d$$ ***Tous les produits d'exactement d variables***

$$k_{\texttt{poly2}}(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z} + c)^d$$ ***Tous les produits d'au plus d variables***

- **Noyaux gaussiens**

$$k_G(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{d(\boldsymbol{x}, \boldsymbol{z})^2}{2\,\sigma^2}\right)$$ ***Sorte de décomposition en série de Fourrier***

- **Noyaux sigmoïdes**

$$k(\boldsymbol{x}, \boldsymbol{z}) = \tanh(\kappa\, \boldsymbol{x}^\top \boldsymbol{z} + \theta)$$ ***Pas définie positive. Mais fonction de décision proche des réseaux connexionnistes***

- **Fonctions noyau pour des textes**

Approche classique : « sacs de mots » → ne tient pas compte de la sémantique.

Utiliser une matrice de similarité ou de sémantique **S** qui sera utilisée dans la définition du noyau :

$$\kappa(d_1, d_2) = \phi(d_1)\,\mathbf{S}\mathbf{S}^\top\,\phi(d_2)^\top$$

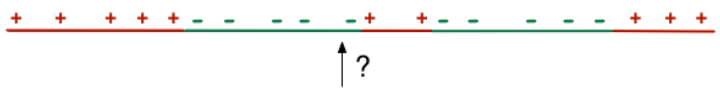
où d_1 et d_2 sont des textes ou des documents, et $\phi(d)$ est la projection de d dans un espace de redescription.

La matrice **S** peut résulter de la composition de plusieurs étapes de traitement. Par exemple, une opération de pondération des termes, qui peut utiliser l'approche *tf-idf*, et une opération mesurant la proximité entre les mots:

$$\mathbf{S} = \mathbf{RP}$$

où **R** est une matrice diagonale spécifiant le poids des termes, et **R** est la matrice de proximité, dans laquelle $\mathbf{P}_{ij} > 0$ quand les termes t_i et t_j sont sémantiquement liés.

- **Méthodes d'ensemble** : le boosting



- Quel est le meilleur séparateur linéaire ?

- Taux d'erreur = 5/20 = 0.25

Exemple simple

- Taux d'erreur (h_1) = 5/20 = 0.25

Et si je pouvais combiner avec un autre séparateur linéaire ?

Ou même plusieurs autres !

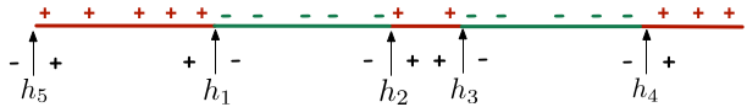
Exemple simple

Et si je pouvais combiner avec un autre séparateur linéaire ? Ou même plusieurs autres !

Par exemple en utilisant un **vote pondéré** :

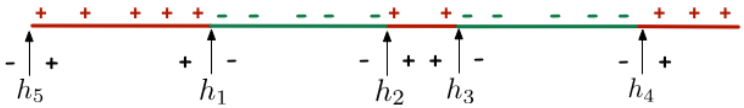
$$H(\mathbf{x}) = \text{sign}\left\{\sum_{i=1}^{l} \alpha_i \, h_i(\mathbf{x})\right\}$$

Exemple simple



H(x) = sign{ 0.549 h_1(x) + 0.347 h_2(x) +
0.310 h_3(x) + 0.406 h_4(x) + 0.503 h_5(x) }

Exemple simple



H(x) =sign{ 0.549 h_1(x) + 0.347 h_2(x) + 0.310 h_3(x) + 0.406 h_4(x)
+ 0.503 h_5(x) }

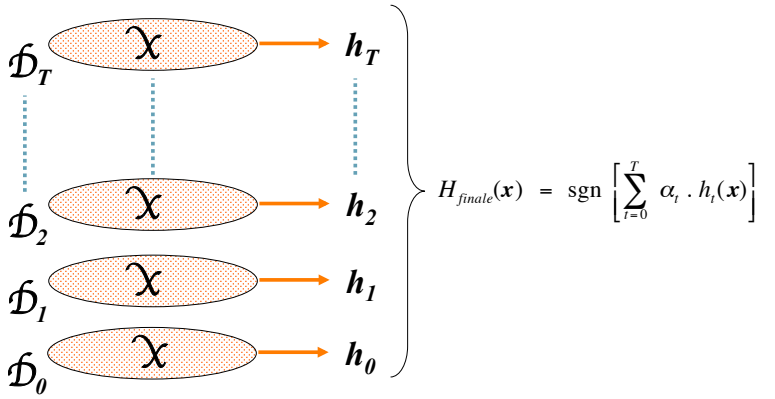
- Comment arriver à ce genre de combinaison ?

Algorithme du boosting

Boosting

- **boosting** = méthode générale pour convertir des règles de prédiction peu performantes en une règle de prédiction (très) performante
- Plus précisément :
  - Étant donné un algorithme d'apprentissage "faible" qui peut toujours retourner une hypothèse de taux d'erreur ≤1/2-γ
  - Un algorithme de boosting peut construire (de manière prouvée) une règle de décision (hypothèse) de taux d'erreur ≤ ε

Le principe général



- Comment passer de $\mathcal{D}_t$ à $\mathcal{D}_{t+1}$?
- Comment calculer la pondération α_t ?

AdaBoost [Freund&Schapire '97]

- construire D_t: $D_1(i) = \frac{1}{m}$

Étant donnée D_t et h_t:

$$D_{t+1} = \frac{D_t}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

$$= \frac{D_t}{Z_t} \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(x_i))$$

où: Z_t = constante de normalisation

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0$$

- Hypothèse finale : $H_{\text{final}}(x) = \text{sgn}\left(\sum_t \alpha_t h_t(x)\right)$

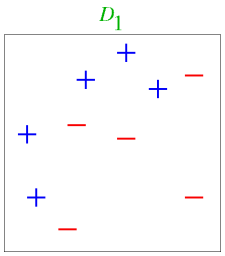
AdaBoost en plus gros

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0$$

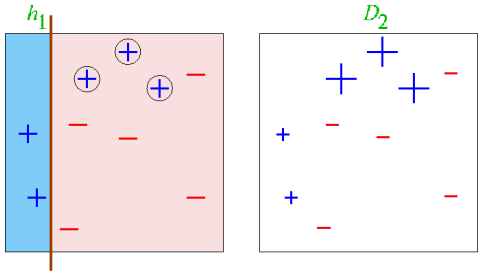
$$D_{t+1} = \frac{D_t}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

$$H_{\text{final}}(x) = \text{sgn}\left(\sum_t \alpha_t h_t(x)\right)$$

Exemple jouet

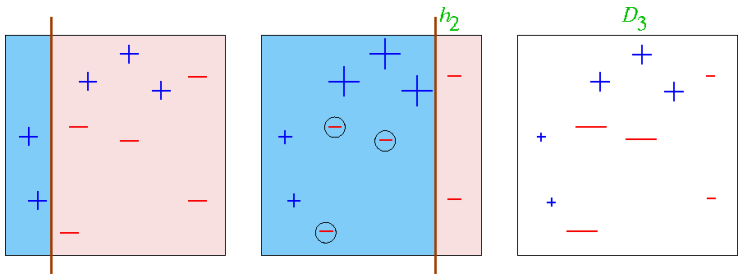


Étape 1



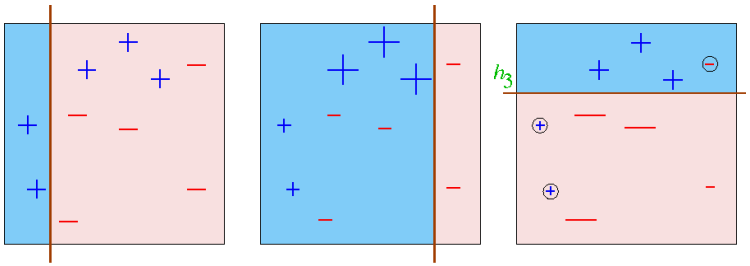
$\varepsilon_1 = 0.30$
$\alpha_1 = 0.42$

Étape 2



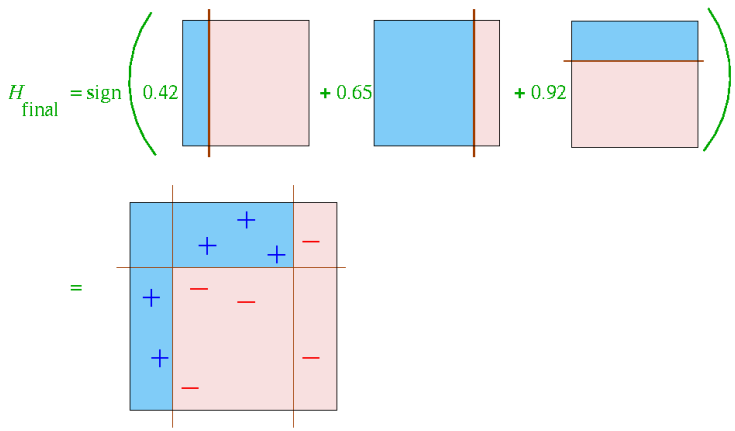
$\varepsilon_2 = 0.21$
$\alpha_2 = 0.65$

Étape 3



$\varepsilon_3 = 0.14$
$\alpha_3 = 0.92$

Hypothèse finale



Leçons et limites

- Les méthodes récentes sont des **méthodes linéaires**
  - Avec choix des descripteurs (ou des fonctions noyau)
  - Ou combinaison d'apprenants (boosting)
  - Apprentissage de dépendances fortes et non locales ?
- Données **i.i.d.**
- Apprentissage « **one-shot** »
  - Pas « life-long »
  - Pas multi-tâches
- Sorties **non structurées**

Points clés

- Lien risque **empirique** <-> risque **réel**
- **Nouveaux critères inductifs**
  - Contrôle de la capacité de $\mathcal{H}$ *a posteriori* (e.g. vaste marge)
  - Nouvelles fonctions de coût (« surrogate functions »)
  - Convexité
  - Parcimonie
- Méthodes à **noyaux**
  - Contrôle de $\mathcal{H}$; convexité ;
    projection dans espace de combinaison de descripteurs
- Méthodes **d'ensemble**
  - Boosting ; bagging ; forêts ; …

Points de vigilance

- La **sur-adaptation**
  - Sélection de modèle
  - Régularisation
- Utilisation de **données non étiquetées**
- Choix des **descripteurs**
- Données en **très grandes dimensions**
  - Signification des résultats
  - Identification du sous-espace (non linéaire) où vivent les données

Nouvelles questions

- Apprentissage **semi-supervisé**
- **Co-apprentissage**
- Apprentissage de **tri** (ranking)
- Apprentissage de **sorties stucturées**
- Apprentissage **multi-tâches**
  - Et « life-long learning »
- Apprentissage **actif**
- Apprentissage des **descripteurs** (et de sémantique)
  - « Deep belief networks »
  - Modèles parcimonieux

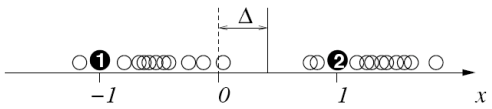
Nouvelles tendances (un choix)

- Apprentissage **semi-supervisé**
  - Comment utiliser des exemples non étiquetés
- Apprentissage **collaboratif** de tri (et de préférence)
  - Nouveaux critères inductifs ; nouvelles mesures de performances
- Textes et sémantique
  - Apprentissage de descripteurs (en hiérarchie)
- Apprentissage multi-tâche
  - Similarité
  - Apprentissage de distance
  - Transfert

Apprentissage semi-supervisé

- But : apprendre une fonction : X -> Y
- Problème
  - l'étiquetage des exemples est cher
  - E.g. *spams ; textes ; images ; …*
- Mais il existe souvent un grand nombre d'exemples non étiquetés

$$\mathcal{S} = \mathcal{S}_s \cup \mathcal{S}_{ns} = \{(\mathbf{x}_i, u_i)\}_{i=1,l} \cup \{\mathbf{x}_i\}_{i=l+1,m}$$



Apprentissage semi-supervisé

Approches

- Apprentissage **non** supervisé + données étiquetées

- Apprentissage supervisé + données **non** étiquetées
  - Auto-apprentissage
  - Co-apprentissage
  - S3SVM
  - Graphes

Co-apprentissage

Représentations multi-vues

Un même exemple peut être décrit de différentes manières

- ***Mail*** : (1) entête ($\mathcal{X}_1$); (2) contenu textuel ($\mathcal{X}_2$)
- ***Page web*** : (1) liens ($\mathcal{X}_1$); (2) contenu ($\mathcal{X}_2$)

Peut-on **combiner** l'utilisation de ces représentations

pour parvenir à un apprentissage plus performant ?

Co-apprentissage

Algorithme 2 : **Algorithme de coapprentissage**

répéter

Apprendre un classificateur A sur $\mathcal{S}_{sup}$;
Apprendre un classificateur B sur $\mathcal{S}_{sup}$;
Classer $\mathcal{S}_{nonsup}$ par A ;
Classer $\mathcal{S}_{nonsup}$ par B ;
Choisir les p_1 exemples positifs et n_1 exemples négatifs de $\mathcal{S}_{nonsup}$ les plus sûrs pour A ;
Choisir par B p_2 exemples positifs et n_2 exemples négatifs de $\mathcal{S}_{nonsup}$ les plus sûrs pour B ;
Ajouter ces $p_1 + n_1 + p_2 + n_2$ nouvellement classés à $\mathcal{S}_{sup}$;

jusqu'à *la convergence est réalisée* ;

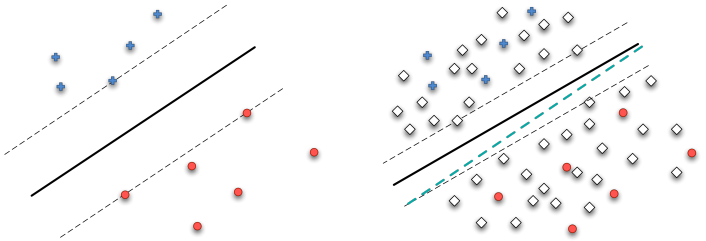
Nécessite une **mesure de certitude ou de confiance** dans l'étiquette calculée.

Co-apprentissage

- **Justification**
  - Si le classifieur **A** trouve des données non supervisées proches de données supervisées et les classe donc avec confiance,
  - cela ne signifie pas qu'ils sont proches pour **B**. D'où apport d'information possible d'un classifieur à l'autre.
  - Il est important que X_1 et X_2 soient indépendants
- Technique **efficace** dans de nombreux cas
- Supportée par **analyse théorique**

S3SVM

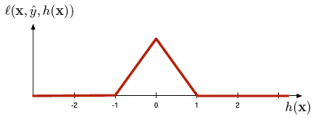
- **A priori sur la nature des données**
  - Mélange de distributions gaussiennes convexes
    -> *les changements d'étiquettes se font dans des régions de moins forte densité*



S3SVM

- **On ajoute une contrainte sur le critère inductif**

Fonction de coût pour les exemples non supervisés.



$$R_{\text{Emp}}(h) = \underset{\mathbf{w},b}{\text{ArgMin}} \Big\{ \underbrace{\sum_{i=1}^{l} \max(1 - u_i(\mathbf{w}^\top \mathbf{x}_i + b), 0) + \lambda_1 ||\mathbf{w}||^2}_{\text{Risque empirique supervisé}} + \underbrace{\lambda_2 \sum_{j=l+1}^{m} \max(1 - |\mathbf{w}^\top \mathbf{x}_j + b|, 0)}_{\text{Risque empirique non supervisé}} \Big\}$$

S3SVM

Leçons :

- En pratique, il arrive que ce critère conduise au choix d'une séparatrice mettant une grande partie des exemples non supervisés, sinon tous, dans la même classe.
  → On imposer que la proportion des classes soient approximativement la même que celle des exemples supervisés. On complète le critère inductif 3 par la contrainte :

$$\frac{1}{m-l} \sum_{j=l+1}^{m} h(\mathbf{x}) = \frac{1}{l} \sum_{i=1}^{l} u_i$$

- La fonction de perte en chapeau conduit à un **critère inductif non convexe** et de ce fait **difficile à optimiser**.
- Les S3VMs **peuvent conduire à de mauvais résultats** si le présupposé selon lequel les classes sont bien séparées est en défaut.
- Se généralise à l'utilisation d'un espace de redescription $\Phi(\mathcal{X})$.

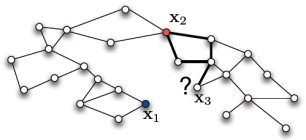
Méthodes à base de graphes

- **Principe** :
  - On suppose qu'à l'intérieur de l'espace des entrées $\mathcal{X}$, les données sont définies sur un **sous-espace** de dimension inférieure à celle de $\mathcal{X}$.
  - **On représente ce sous-espace à l'aide d'un graphe** mettant en jeu les exemples connus, supervisés et non supervisés,
  - et on suppose que **la fonction d'étiquetage** de ces points vérifie une certaine **continuité**

Méthodes à base de graphes

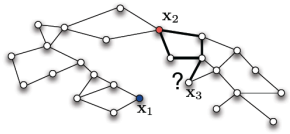
On **représente les données sous forme d'un graphe** (V, E) :

- V : ***sommets*** = exemples d'apprentissage (supervisés et non supervisés)
- E : ***arcs*** = similarité entre exemples. Matrice $\mathbf{W}$ où $\mathbf{W}_{ij}$ mesure la similarité entre l'objet i et l'objet j dans $\mathcal{S}$.
  $\mathbf{W}_{ij}$ peut être binaire ou un réel (e.g. $\mathbf{W}_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}}$).
- Les étiquettes des données supervisées sont placées sur les sommets correspondants du graphe.



Apprentissage semi-supervisé

Méthodes à base de graphes



Principe : étendre l'étiquetage à tous les sommets.

Optimiser un compromis entre :

- *fidélité* aux données d'apprentissage (les points initialement étiquetés devraient conserver leur étiquette au cours de la propagation)
- *continuité* dans l'étiquetage

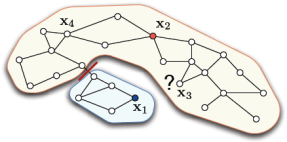
C'est en fait de l'apprentissage transductif

Apprentissage semi-supervisé

Méthodes à base de graphes : l'un des critères (*le MinCut*)

Le critère MinCut suppose que les étiquettes des données, et donc des nœuds du graphe, sont prises dans $\{+1, -1\}$.

- Les **exemples positifs** sont considérés comme des « sources »
- Les **exemples négatifs** comme des « puits »
- L'objectif est de bloquer le flot entre sources et puits en coupant un ensemble d'arcs de poids total minimal.



Inconvénient : Souvent plusieurs solutions de même valeur

Apprentissage collaboratif

- Motivation
  - De plus sollicités pour faire des choix (tris)
  - Apprentissage **individuel**
    - Lent
    - Trop limité
  - **Apprentissage collectif**
    - Quelqu'un **ayant des goûts similaires aux nôtres** pourraient nous aider dans nos choix à faire si il les a déjà faits

Apprentissage collaboratif

Les approches

- Par **modèle de l'utilisateur**
  - Construction d'un modèle de l'utilisateur
  - Exemple : *modèle régressif linéaire*

$$pred(u_l, i_j) = \alpha_1(\hat{a}ge(u_l)) + \alpha_2(revenu(u_l)) + \alpha_3(intervalle\_prix(i_j)) + \ldots$$

    - u_l : utilisateur *l*
    - i_j : item *j*
  - ***Problèmes***
    - Choix du modèle
    - Incorporation de facteurs difficiles à expliciter
      - > **Approches à variables latentes**

Apprentissage collaboratif

Les approches

- **Par similarité et plus proches voisins**
  - Un **utilisateur** est caractérisé par les appréciations qu'il a déjà émises à propos de certains items

	Item 1	*Item 2*	*Item 3*	*Item 4*	...
Jean	-	2	7	8	...
Marie	4	1	-	7	...
Christian	3	8	-	4	...

  - Questions
    - Quelle **mesure de similarité** entre utilisateurs ?
    - Combien de « **voisins** » ?
    - Quelle **formule de combinaison** d'avis ?

Apprentissage collaboratif

- **Mesure de similarité**
  - Formule de Bravais-Pearson
    - On suppose que les notes attribuées par les utilisateurs U_i et U_j sont des variables aléatoires X_i et X_j de distribution conjointe inconnue

$$\rho = \frac{Cov(X_i, X_j)}{\sqrt{Var(X_i)Var(X_j)}} \qquad Cov(X_i, X_j) = \mathbb{E}\left[(X_i - \mathbb{E}(X_i))\,(X_j - \mathbb{E}(X_j))\right]$$

    - Dans [-1, +1] : -1 (très différents) , +1 (mêmes appréciations)

$$r = \frac{\sum_k (X_i^k - \bar{X}_i)(X_j^k - \bar{X}_j)}{\sqrt{\sum_k (X_i^k - \bar{X}_i)^2}\sqrt{\sum_k (X_j^k - \bar{X}_j)^2}}$$

Apprentissage collaboratif

- **Mesure de similarité**

	Item 1	*Item 2*	*Item 3*	*Item 4*	...
Jean	-	2	7	8	...
Marie	4	1	-	7	...
Christian	3	8	-	4	...

Selon cette mesure, la similarité entre *Jean* et *Marie* est :

$$r(\text{Jean, Marie}) = \frac{(2-5)(1-4)+(8-5)(7-4)}{\sqrt{(2-5)^2+(8-5)^2}\sqrt{(1-4)^2+(7-4)^2}} = \frac{(-3)(-3)+(3)(3)}{\sqrt{9+9}\sqrt{9+9}} = \frac{18}{18} = 1$$

avec $X_{\text{Jean}} = (2+8)/2 = 5$ et $X_{\text{Marie}} = (1+7)/2 = 4$ quand on prend les articles notés en commun.

Entre *Jean* et *Christian*, elle est : $r(\text{Jean, Christian}) = \frac{-12}{\sqrt{18}\cdot\sqrt{8}} = -12/(3\cdot 2\cdot 2) = -1$

Et entre *Marie* et *Christian* : $r(\text{Marie, Christian}) = \frac{-12}{\sqrt{18}\cdot\sqrt{14}} = -4/(\sqrt{2}\cdot\sqrt{14}) \approx -0.756$

Une meilleure mesure : prendre un exposant 2.5

Apprentissage collaboratif

- **Calcul des préférences**
  - On prend les *k* plus proches voisins selon la mesure de similarité

$$Pred(u_l, i_j) = \bar{X}_l + \frac{\sum_{v=1}^{k} r(u_l, u_v)\,(X_v^j - \bar{X}_v)}{\sum_{v=1}^{k} |r(u_l, u_v)|}$$

$\bar{X}_l$: moyenne des notes attribuées par l'utilisateur u_l à *tous* les items

Supposons que nous voulions prédire la note que donnerait *Jean* à l'*item 1* et que nous prenions *Marie* et *Christian* comme voisins.

$$Pred(\text{Jean, item 1}) = \frac{17}{3} + \frac{(1\cdot(4-4)) + (-1\cdot(3-6))}{1+1} = 5.67 + 1.5 \approx 7.17$$

Intuitivement, puisque *Christian* n'a pas aimé l'*item 1*, et qu'il est négativement corrélé avec *Jean*, alors cela devrait amener à penser que *Jean* va plutôt aimer *item 1*, d'où une valeur accordée à *item 1* supérieure à la moyenne de *Jean*. L'évaluation de *Marie*, quant à elle, ne modifie rien car elle évalue l'*item 1* à sa moyenne : 4.

Apprentissage collaboratif

- Les **difficultés**
  - **Dimension** de la matrice *utilisateurs-items* souvent **énorme**
    - Précalculs ou clustering préalable
  - Matrice **très creuse** (typiquement moins de 1%)
  - Choix de la mesure de similarité
  - **Efficacité** (souvent moins de 2 s.)
  - **Mesure de la performance**
    - Validation croisée (lourde)
    - Mais classement seulement
    - Il faut que le système suggère aussi des « nouveautés »

Apprentissage et textes

- **Buts**
  - « Traduire » un texte (des textes) en un langage adapté à la machine pour des **tâches**
    - De **recherche d'information**
    - **Questions / réponses**
    - **Résumés**
    - **Enrichissement** d'une base de connaissances
    - …
  - C'est de l'**IA !!**

Natural Language Processing (ou TAL)

- **Good Old Fashioned AI**
  - Recherches sur **la représentation des connaissances**
    - Roger Schank, Kristian Hammond, Michael Dyer, Janet Kolodner, John Sowa, …
    - Scripts, plans, goals, MOPs, TAUs, …
  - **Systèmes d'analyse de textes** (QA, traduction, …)
    - Indexation
    - Pattern matching
    - Inférences
  - Mais comment apprendre ces structures ? (souvent ad hoc)

Text Mining and Information Retrieval

- **Tâches** plus limités
  - Part-Of-Speech tagging, chunking, parsing
  - Word sense disambiguation, semantic role labeling, named entity extraction, anaphora resolution
- **Modèles** beaucoup plus **simples**
  - Modèles linéaires statistiques
  - SVM linéaires
- **Descripteurs** issus des experts
  - Souvent ad hoc
  - Processus d'identification empirique

TAL : nouveau programme

- **Apprendre les représentations** nécessaires
  - À partir d'exemples (très nombreux) (y compris non étiquetés)
  - Adaptées à des tâches multiples
    - En TAL
    - Mais pourquoi pas en traitement d'images
- Approche
  - Les « deep belief networks »

TAL : tâches de benchmark

- **Part-Of-Speech tagging**
  - Etiqueter chaque mot avec son rôle syntaxique (e.g. *nom pluriel, adverbe, …*)
  - **État de l'art**
    - Classifieurs entraînés sur des fenêtres de textes avec algorithmes de décodage bidirectionnels
    - ~ 97% de précision sur les mots (Wall Street Journal)

TAL : tâches de benchmark

- **Chunking**
  - Étiqueter des segments de phrases avec ses constituants syntaxiques (e.g. *groupe nominal, groupe verbal, …*)
  - **État de l'art**
    - Sur la tâche CoNLL 2000 (Wall Street Journal)
    - Score F1 de ~94% (SVM à deux classes et fenêtres autour des mots d'intérêt)
    - Méthode d'ensemble avec classifieurs binaires -> 95%

TAL : tâches de benchmark

- **Named Entity Recognition**
  - Étiqueter les éléments de la phrase en catégories (e.g. `Personne, Lieu, …`)
  - **État de l'art**
    - Sur la tâche CoNLL 2003 (données Reuters)
    - Score F1 de ~89%
    - Apprentissage semi-supervisé sur un large corpus (27 M de mots)
    - Descripteurs experts (i.e. ad hoc) + gros dictionnaire (« gazetteer »)

TAL : tâches de benchmark

- **Semantic Role Labeling**
  - Associer un rôle sémantique à un constituant syntaxique de la phrase
    (e.g. $[\text{John}]_{ARG0}$ $[\text{ate}]_{REL}$ $[\text{the apple}]_{ARG1}$ (où ate est reconnu comme un prédicat))
  - État de l'art
    - Sur la tâche CoNLL 2005 (Wall Street Journal)
    - Production d'un arbre de parsing, identification du nœud de l'arbre, classification de ces nœuds pour le calcul du rôle sémantique (modèle statistique)
    - Descripteurs ad hoc
    - ~77.3% de score F1

Les « deep belief networks »

- Motivation
  - Disposer d'un espace d'hypothèses adapté : langage de concepts
  - L'apprendre
- Moyen proposé
  - Réseaux de neurones « profonds »
  - Permettant de **décomposer** en descripteurs
  - Avec **interdépendances** complexes
- Approches
  - Réseaux à convolution (LeCun et al.)
  - « Deep belief networks » (Hinton et al.)

Les « deep beliefs networks »

- En un transparent

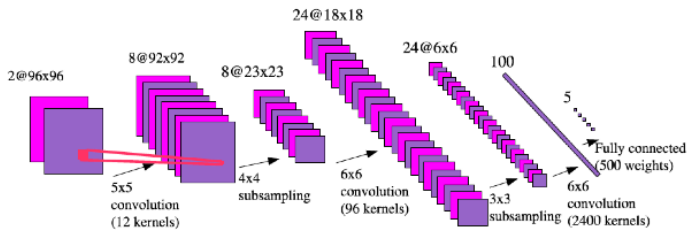
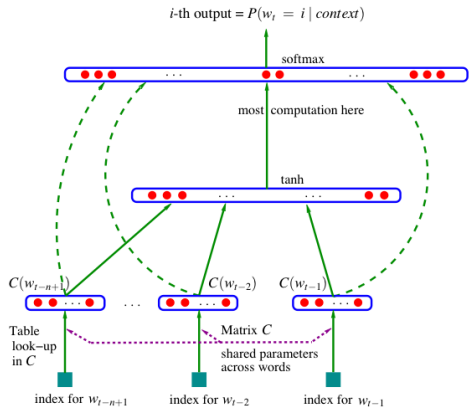


FIG. 10.15: *Architecture d'un réseau connexionniste utilisé pour traiter des images de la base NORB. L'entrée consiste en une paire d'images, dont le système extraie 8 descripteurs de taille 92 × 92 calculés sur des imagettes 5 × 5. Les sorties de ces descripteurs sont reprises par 8 descripteurs 23 × 23, 24 descripteurs 18 × 18, 24 descripteurs 6 × 6 et une couche de 100 neurones complètement connectés aux 5 neurones de sortie qui donnent la distance avec les vecteurs cibles. (Repris de [BL07].)*

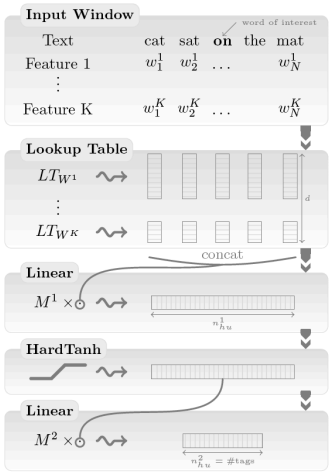
Apprentissage d'une « word embedding matrix »



- [Bengio et al., JMLR, 2003]

Apprentissage de représentations hiérarchiques

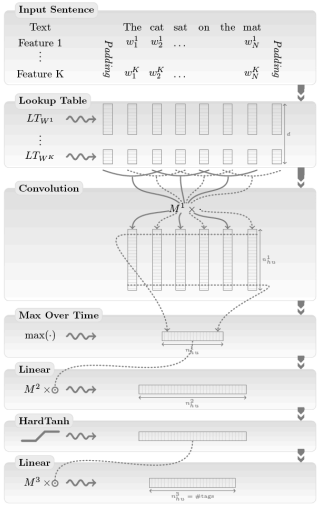
- L'approche par fenêtre
  - Présuppose que l'étiquette d'un mot dépend des mots voisins
  - Plusieurs couches de combinaison linéaire + non-linéarité
  - Couche de sortie : un neurone / étiquette



- [Collobert et al., JMLR, 2011]

Apprentissage de représentations hiérarchiques

- L'approche par phrase
  - Si dépendances à plus longue portée (e.g. pour le SRL)
  - Réseaux à convolution fenêtre de fenêtres (combinaison linéaire)



- [Collobert et al., JMLR, 2011]

Apprentissage de représentations hiérarchiques

- **Apprentissage**
  - Maximisation d'une vraisemblance sur l'échantillon d'apprentissage

$$\theta^\star = \underset{\theta}{\text{ArgMax}} \sum_{(\mathbf{x},y)\in\mathcal{S}} \log p(y|\mathbf{x},\theta)$$

  - Gradient stochastique
  - Sorties interprétées comme des probabilités
    - **Force les mots à ne plus être traités indépendamment**
    - Prise en compte d'un **coût de transition entre étiquettes**
      - Minimisation du coût du chemin suivi (Viterbi)
    - **Critère** de coût **complexe**
    - **Calculs très coûteux** (plusieurs semaines sur énorme corpus)

Apprentissage de représentations hiérarchiques

Approach	POS (PWA)	Chunking (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99

- Performances un peu en retrait
- Mais descripteurs appris
- Et généricité de l'approche

Apprentissage de représentations hiérarchiques

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
454	1973	6909	11724	29869	87025
PERSUADE	THICKETS	DECADENT	WIDESCREEN	ODD	PPA
FAW	SAVARY	DIVO	ANTICA	ANCHIETA	UDDIN
BLACKSTOCK	SYMPATHETIC	VERUS	SHABBY	EMIGRATION	BIOLOGICALLY
GIORGI	JFK	OXIDE	AWE	MARKING	KAYAK
SHAHEED	KHWARAZM	URBINA	THUD	HEUER	MCLARENS
RUMELIA	STATIONERY	EPOS	OCCUPANT	SAMBHAJI	GLADWIN
PLANUM	ILIAS	EGLINTON	REVISED	WORSHIPPERS	CENTRALLY
GOA'ULD	GSNUMBER	EDGING	LEAVENED	RITSUKO	INDONESIA
COLLATION	OPERATOR	FRG	PANDIONIDAE	LIFELESS	MONEO
BACHA	W.J.	NAMSOS	SHIRT	MAHAN	NILGIRIS

Table 6: Word embeddings in the word lookup table of a SRL neural network trained from scratch, with a dictionary of size 100,000. For each column the queried word is followed by its index in the dictionary (higher means more rare) and its 10 nearest neighbors (arbitrarily using the Euclidean metric).

Apprentissage de représentations hiérarchiques

- En apprenant la matrice d'embedding à partir de
  - English Wikipedia : 631 M de mots
  - + Reuters CV1 : 221 M de mots
  - Approche par fenêtre
  - Critère ne favorisant pas les phrases communes / phrases rares mais correctes

Apprentissage de représentations hiérarchiques

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
454	1973	6909	11724	29869	87025
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Table 7: Word embeddings in the word lookup table of the language model neural network LM1 trained with a dictionary of size 100,000. For each column the queried word is followed by its index in the dictionary (higher means more rare) and its 10 nearest neighbors (using the Euclidean metric, which was chosen arbitrarily).

Apprentissage de représentations hiérarchiques

- **Apprentissage semi-supervisé**

Approach	**POS** (PWA)	**CHUNK** (F1)	**NER** (F1)	**SRL** (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
NN+WLL	96.31	89.13	79.53	55.40
NN+SLL	96.37	90.33	81.47	70.99
NN+WLL+LM1	97.05	91.91	85.68	58.18
NN+SLL+LM1	97.10	93.65	87.58	73.84
NN+WLL+LM2	97.14	92.04	86.96	58.34
NN+SLL+LM2	97.20	93.63	88.67	74.15

Table 8: Comparison in generalization performance of benchmark NLP systems with our (NN) approach on POS, chunking, NER and SRL tasks. We report results with both the word-level log-likelihood (WLL) and the sentence-level log-likelihood (SLL). We report with (LMn) performance of the networks trained from the language model embeddings (Table 7). Generalization performance is reported in per-word accuracy (PWA) for POS and F1 score for other tasks.

Apprentissage de représentations hiérarchiques

- **Apprentissage multi-tâches**

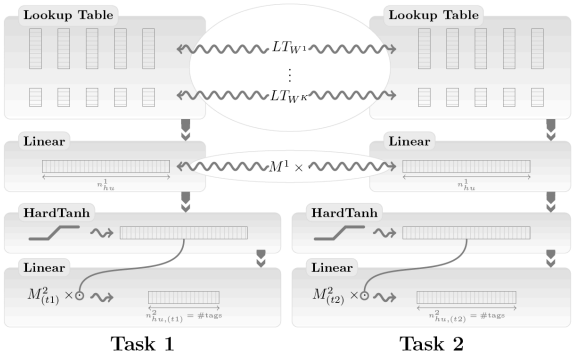


Figure 5: Example of multitasking with NN. Task 1 and Task 2 are two tasks trained with the window approach architecture presented in Figure 1. Lookup tables as well as the first hidden layer are shared. The last layer is task specific. The principle is the same with more than two tasks.

Apprentissage de représentations hiérarchiques

- **Apprentissage multi-tâches**

Approach	POS (PWA)	CHUNK (F1)	NER (F1)	SRL (F1)
Benchmark Systems	97.24	94.29	89.31	77.92
	Window Approach			
NN+SLL+LM2	97.20	93.63	88.67	–
NN+SLL+LM2+MTL	97.22	94.10	88.62	–
	Sentence Approach			
NN+SLL+LM2	97.12	93.37	88.78	74.15
NN+SLL+LM2+MTL	97.22	93.75	88.27	74.29

Table 9: Effect of multi-tasking on our neural architectures. We trained POS, CHUNK NER in a MTL way, both for the window and sentence network approaches. SRL was only included in the sentence approach joint training. As a baseline, we show previous results of our window approach system, as well as additional results for our sentence approach system, when trained separately on each task. Benchmark system performance is also given for comparison.

Apprentissage de représentations hiérarchiques

- **Apprentissage multi-tâches + méthode d'ensemble**

Approach		POS (PWA)	CHUNK (F1)	NER (F1)
Benchmark Systems		97.24	94.29	89.31
NN+SLL+LM2+POS	worst	97.29	93.99	89.35
NN+SLL+LM2+POS	mean	97.31	94.17	89.65
NN+SLL+LM2+POS	best	97.35	94.32	89.86
NN+SLL+LM2+POS	voting ensemble	97.37	94.34	89.70
NN+SLL+LM2+POS	joined ensemble	97.30	94.35	89.67

Table 11: Comparison in generalization performance for POS, CHUNK and NER tasks of the networks obtained using by combining ten training runs with different initialization.

Apprentissage multi-tâches

- **Tâches différentes**
  - Même espaces d'entrée et sorties, mais régularités différentes
  - Mêmes espaces d'entrées, mais pas de sortie (multi-objectif)
  - Mêmes espaces de sortie, mais pas d'entrée
  - Pas les mêmes espaces
- **Questions**
  - Comment mesurer la **similarité entre tâches** ?
  - Que **transférer** ?
- **État de l'art**
  - Beaucoup de travaux
  - Presque **pas encore de théorie**

Apprentissage multi-tâches

- Intérêt
  - Apprentissage **à long terme**
    - Subsume apprentissage incrémental
    - Apprentissage en-ligne
  - Données et tâches **non i.i.d.**
  - Prise en compte de **structure** dans les mesures de similarité
    - Analogie
    - « Priming effects »

Leçons

- Apprentissage **semi-supervisé**
  - Exploitation d'énormes corpus
- Apprentissage
  - De distances
  - De **représentations**
- Apprentissage **multi-tâches**
  - Généralisation des apprentissages de représentation
  - Données non i.i.d.
    - Mémoire ; transfert

L'apprentissage artificiel à AgroParisTech

- Apprentissage **en-ligne** (à partir de flux de données)
  - Dérive de concepts
  - Par méthode d'ensemble
  - Apprentissage par renforcement
- Apprentissage à partir d'**exemples positifs seuls**
  - Fouille de données
    - Amarrage de protéines
- **Ranking**
- Analyse de **corpus**
  - Projet Holyrisk
    - Etiquetage de parties de textes