Une introduction à l'**Apprentissage Incrémental**

Antoine Cornuéjols
avec l'aide de **Lou Fedon**

MMIP, AgroParisTech, Paris

13 mars 2008

Pourquoi parler d'apprentissage « incrémental » ?

... maintenant ?

« Nouvelles » applications

- Très grosses bases de données
- Flux de données
- Apprentissage continu
- Résolution par transfert de tâche

Pourquoi parler d'apprentissage « incrémental » ?

... maintenant ?

Intérêt croissant

- Workshop on *Dynamically changing domains: Theory revision and context dependence issues* (ECML-**97**)
- Intelligent Data Analysis journal *Special issue on Incremental Learning Systems Capable of Dealing with Concept Drift*, Vol.8, N0.3, **2004**.
- Workshop *First International Workshop on Knowledge Discovery in Data Streams*, ECML-**04**.
- ACM Symposium on Applied Computing (SAC-**2006**) *Special Track on Data Streams*.
- Workshop *Second International Workshop on Knowledge Discovery in Data Streams*, ECML-**05**.
- Workshop à NIPS-**2006** : *Learning when test and training inputs have different distributions*

Pourquoi parler d'apprentissage « incrémental » ?
... maintenant ?

Tout apprentissage n'est-il pas essentiellement incrémental ?

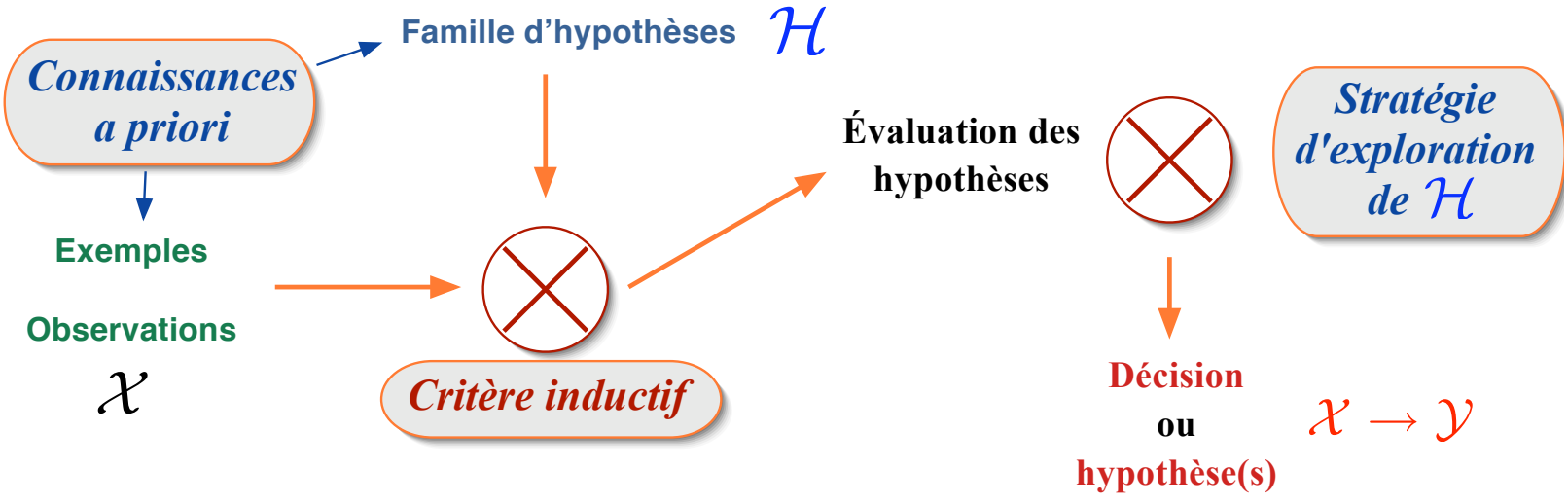
- Apprentissage de « capacités » (“skills”)
- Interprétation incrémentale d'une situation / du monde
- Construction de « théorie »

Plan

Apprentissage « batch »

Fondements

Recherche d'un bon modèle du monde



Avoir un bon modèle du monde

Identifier une dépendance cible :

- $\mathbf{P}_{\mathcal{X}\mathcal{Y}}$
- *Fonction cible* $f : \mathcal{X} \to \mathcal{Y}$

Indécidable et illusoire

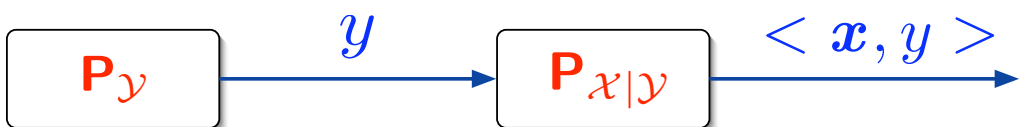


Figure: Modèle de génération des exemples.

- **Distribution stationnaire et identique** en apprentissage et en test.
- **Données i.i.d.** (identiquement et indépendamment distribuées)

Prédictions correctes (la plupart du temps)

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

Avoir un bon modèle du monde ?

Le risque réel

Fonction de perte :

$$\begin{aligned} \ell(h) : \mathcal{X} \times \mathcal{Y} &\to \mathbb{R}^+ \\ (\boldsymbol{x}, y) &\mapsto \ell(h(\boldsymbol{x}), y) \end{aligned}$$

Risque réel : espérance de perte

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

Les critères inductifs

Mais, on ne connaît pas $\mathbf{P}_{\mathcal{X}\mathcal{Y}}$

Échantillon d'apprentissage supposé représentatif

$$\mathcal{S}_m = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$$

Minimisation du Risque Empirique

Choisir l'hypothèse $\hat{h}$ telle que : $\hat{h} = \text{ArgMin}_{h \in \mathcal{H}} \left[R_m(h) \right]$

$$R_m(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h(\boldsymbol{x}_i), y_i)$$

Un dilemme fondamental

Le compromis biais-variance

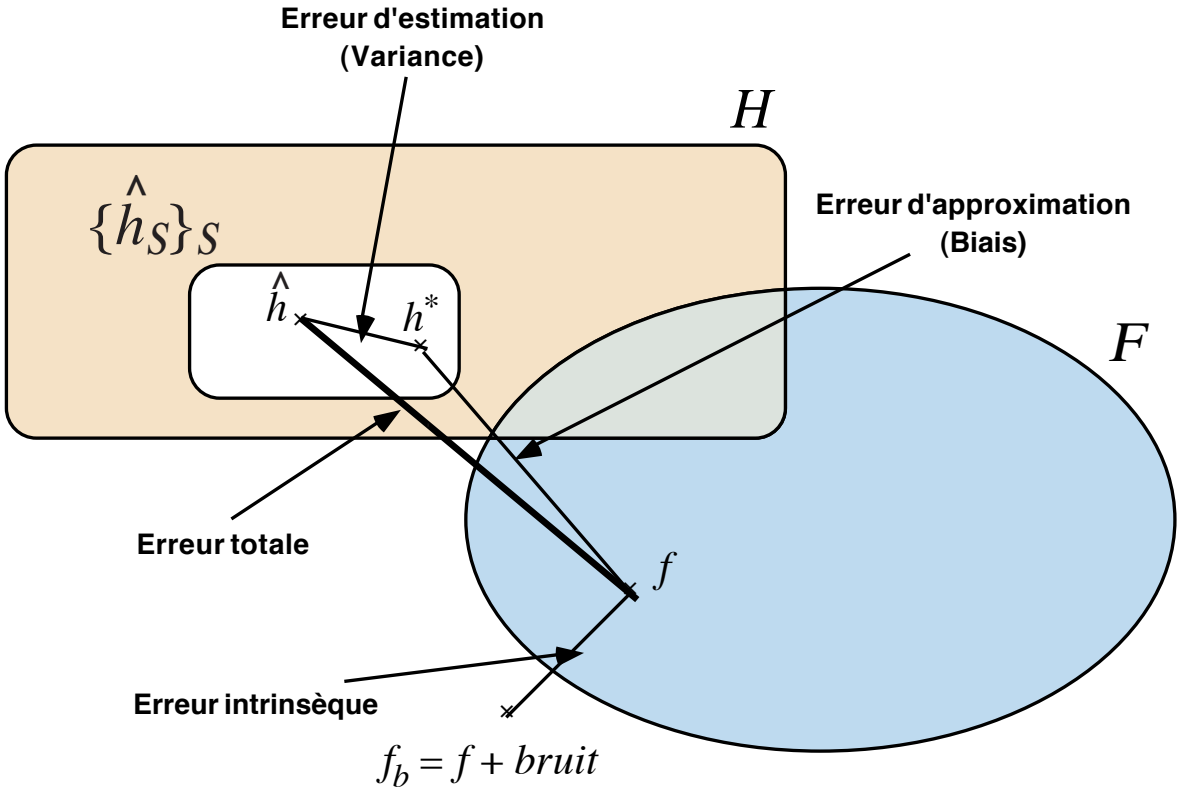


FIG.: Les différents types d'erreurs.

Critère inductif régularisé

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

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

Les techniques d'apprentissage
... sont liées à la théorie

- SVM
- Boosting

- Modèles linéaires
- Modèles bayésiens

- Réseaux de neurones ; Modèles de Markov à états cachés (HMM)

- Arbres de décision
- Règles (ILP : Induction of Logic Programs)
- Grammaires

Apprentissage en-ligne : pourquoi ?

1. **Ressources limitées :**
   - Apprentissage de très grosses bases de données (e.g. SimpleSVM : 80 millions de données, FTR, ...)
2. **Contrainte « anytime »** : Traitement de flux de données
3. **Dérive de la distribution des exemples** (et pas assez d'exemples)
4. **Apprentissage actif**
5. **Dérive de concept** (pas assez d'exemples et contrainte anytime)
6. **Transfert** d'une tâche à une autre
7. **Apprentissage guidé** par un professeur

Apprentissage incrémental

Ressources computationnelles limitées : calcul et espace

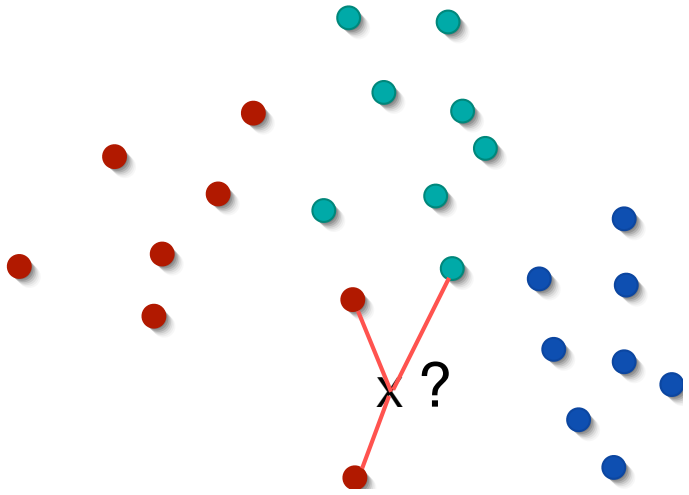
$\longrightarrow$ Impossibilité de traiter d'un seul coup

Exemples

- **k-ppv** (tout est mémorisé)
- **ID5** (idem : on pourrait reconstruire les exemples)
- **EV** (tout est mémorisé, mais on ne peut pas reconstruire les exemples)
- **ID4** (on ne peut pas reconstruire les exemples et effets de l'ordre)
- **LASVM** (choix de ce qu'il faut mémoriser)
- **RN** (pas de problème algorithmique mais oubli catastrophique)

Apprentissage incrémental
Illustration

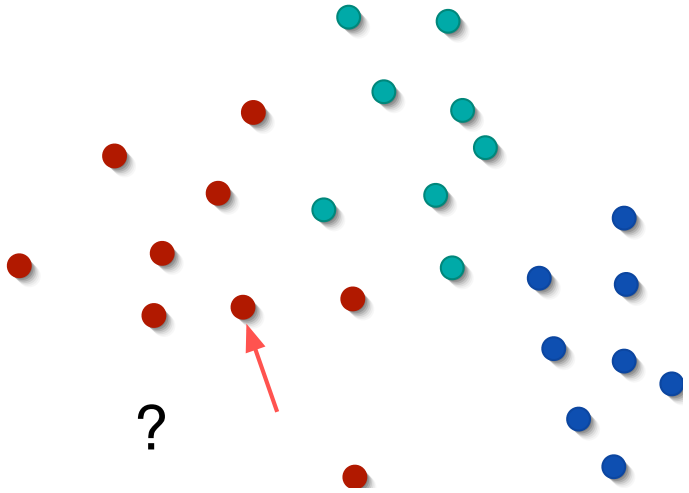
Méthodes de plus proches voisins



- Pas de problème algorithmique (“*lazy learning*”)
- Indépendant de l’ordre
- Mais temps de calcul à chaque étape : $\mathcal{O}(m)$

Apprentissage incrémental
Illustration

Méthodes de plus proches voisins (2)
avec mémoire limitée

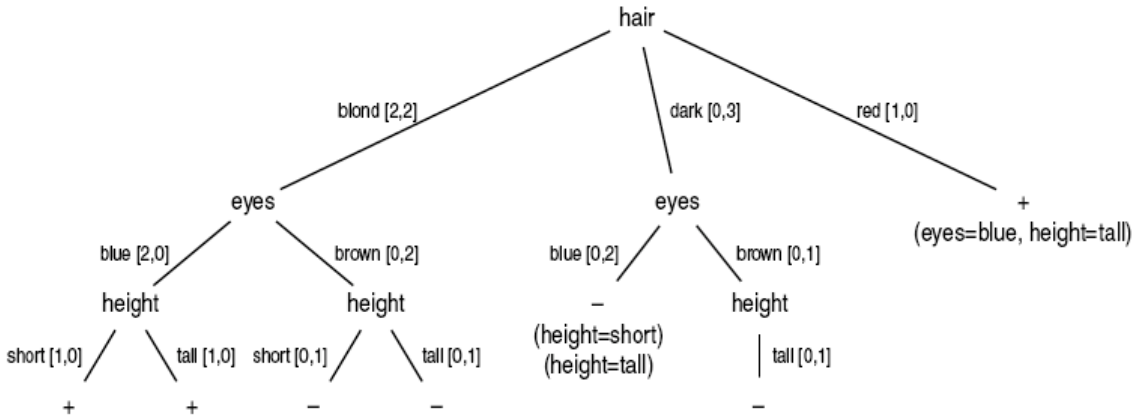


- Sélection de « prototypes »
  - Éliminer l'outlier
  - Éliminer le plus ancien
  - Retenir le centre de gravité avec le point le plus proche
  - ...
- Dépendant de l'ordre

Apprentissage incrémental
Illustration

Induction incrémentale d'arbre de décision (ID5R)

class	height	hair	eyes
–	short	blond	brown
–	tall	dark	brown
+	tall	blond	blue
–	tall	dark	blue
–	short	dark	blue
+	tall	red	blue
–	tall	blond	brown
+	short	blond	blue



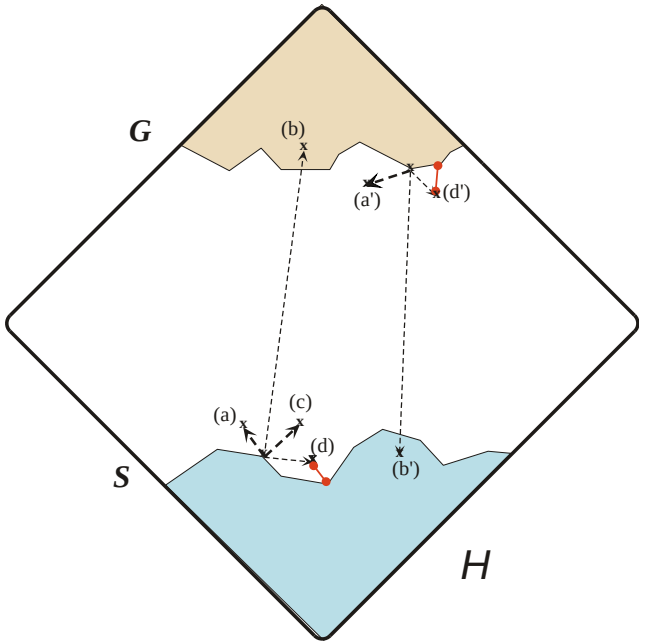
- Mémorisation de tous les exemples
- Indépendant de l'ordre
- Mais temps de calcul à chaque étape : $\mathcal{O}(m \cdot d \cdot b^d)$

UTG89 Paul Utgoff (1989) *"Incremental Induction of Decision Trees"* Machine Learning Journal, vol.4, No.2, 161-186

Apprentissage incrémental
Illustration

Calcul incrémental de l'espace des versions

Existence d'un treillis de généralisation dans l'espace des hypothèses



- On ne peut reconstruire tous les exemples
- Indépendant de l'ordre
- Mais temps de calcul à chaque étape : dépend de la taille de S et G

MIT97 Tom Mitchell (1982) *"Generalization as Search"* Artificial Intelligence, vol.18, 203-226.

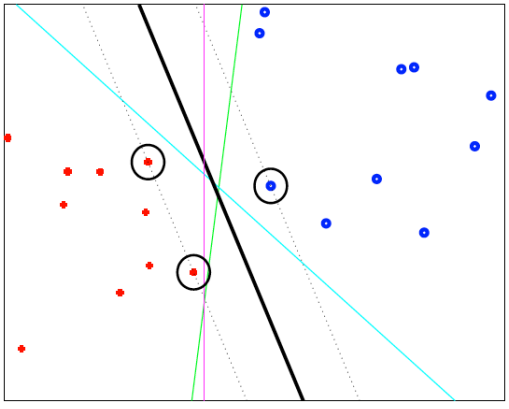
Apprentissage incrémental
Illustration

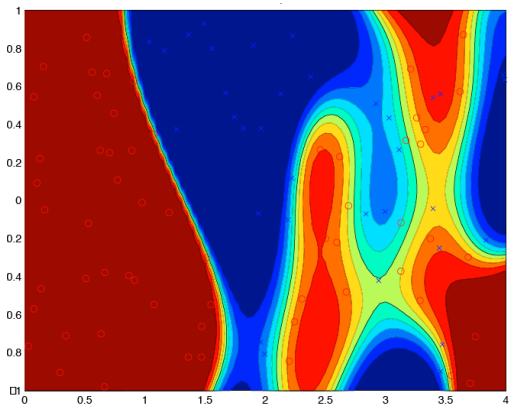
SVM incrémental : LASVM

$$\begin{aligned}\hat{y}(x) &= w^\top \Phi(x) + b \\ &= \sum_{i=1}^{n} \alpha_i K(x, x_i) + b\end{aligned}$$

Problème :

- identifier les points supports (x_i tq. $\alpha_i \neq 0$)





Apprentissage incrémental
Illustration

SVM incrémental : LASVM

Traitement incrémental :

- que mémoriser ?
  - les points de support candidats (et grandeurs associées)
- comment organiser les calculs ?
  - introduction et élimination séquentielle de points candidats
  - jusqu'à état quasi stable
  - coût : $n\,P\,\bar{S} = \mathcal{O}(n^2)$ à $\mathcal{O}(n^3)$ (n exemples, P époques, $\bar{S}$ nb moyen de points de support), mais chaque calcul (des valeurs $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$) est beaucoup plus simple. ($P = 1$ suffit généralement)

BEWB05 Bordes, Ertekin, Weston & Bottou (2005) *"Fast Kernel Classifiers with Online and Active Learning"* JMLR, vol.6:1579-1619, Sept. 2005.

Apprentissage incrémental
Illustration

SVM incrémental : LASVM

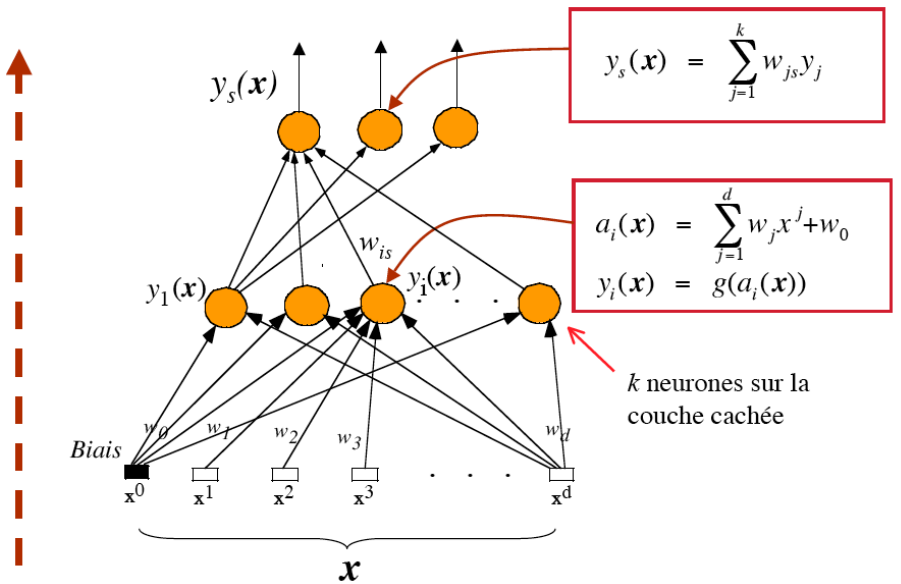
Possibilité d'apprentissage actif :

- **Sélection aléatoire**.
- **Sélection par gradient** : prendre l'exemple le moins bien classé parmi un ensemble d'exemples nouveaux.
- **Sélection active** : prendre l'exemple le plus proche de la frontière de décision.
- **Sélection autoactive** : tirer au plus 100 exemples nouveaux, mais arrêter dès que 5 d'entre eux sont à l'intérieur des marges. Parmi les 5, choisir le plus proche de la frontière de décision.

Apprentissage incrémental
Illustration

Perceptron Multi-Couches

- Traitement en temps constant
- Phénomène d'oubli catastrophique



French97 Bob French (1997) *"Catastrophic Forgetting in Connectionist Networks"* Trends in Cognitive Sciences, vol.3, No.4: 128-135.

Apprentissage incrémental

... vs. apprentissage batch

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

Gradient total

$$\begin{aligned} h_t &= h_{t-1} - \Phi_t \frac{\partial R_m(h)}{\partial h}(h_{t-1}) \\ &= h_{t-1} - \frac{1}{m}\Phi_t \sum_{i=1}^{m} \frac{\partial L}{\partial h}(h_{t-1}(\boldsymbol{x})_i, y_i) \end{aligned}$$

Converge linéairement vers l'optimum $\hat{h}$ de $R_m(h)$: $(h_t - \hat{h})^2$ converge comme e^{-t}.

BLC05 L. Bottou et Y. Le Cun (2005) *"On-line learning for very large datasets"* Applied Stochastic Models in Business and Industry, 21(2):137-151, 2005.

Apprentissage incrémental

... vs. apprentissage batch

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

Gradient stochastique

$$h_t \;=\; h_{t-1} - \frac{1}{t}\,\Phi_t\,\frac{\partial L}{\partial h}(h_{t-1}(\boldsymbol{x}_t), y_t)$$

- Cvge lentement vers un optimum local de $R_m(h)$: $(h_t - \hat{h})^2$ converge comme $\frac{1}{t}$.
- En fait, rapide pour aller vers la région générale de l'optimum, mais lent ensuite en raison du gradient bruité.

Mais beaucoup **plus simple** qu'algorithme batch

Apprentissage incrémental
... vs. apprentissage batch

Complexité

- Batch
  - Mémoriser N exemples
  - Gradient en $\mathcal{O}(N)$ opérations
- En-ligne
  - Mémoriser le « passé »
  - Gradient en $\mathcal{O}(1)$ opérations

Approximation

- Batch
  - Converge vers $h* = \text{ArgMin}_{h \in \mathcal{H}} R(h)$ en $\mathcal{O}(1/t)$
- En-ligne
  - Converge vers $h* = \text{ArgMin}_{h \in \mathcal{H}} R(h)$ en $\mathcal{O}(1/t)$
  - Mais **avec davantage d'exemples !!**
    (en $\mathcal{O}(N \log N)$ si N nb exemples en batch)

Apprentissage incrémental
Bilan

De nombreux algorithmes de nature heuristique

Caractéristiques

- Contraintes sur ce qui est mémorisé
- Temps de calcul constant au cours du temps

Exigences annexes :

- Indépendance sur l'ordre des entrées
- Possibilité de prendre en compte (cas des flux de données)
  - de nouveaux descripteurs
  - de nouvelles classes

Flux de données
Illustration

Entrées **à très haut débit**

Éventuellement à partir de **sources différentes**

Exemples

- Génération automatique de **données à haut débit**
  - *IP traffic log,* *web/blog crawlers*
  - *Senseurs ambiants,* *Physique des hautes énergies, ...*

Nouveaux besoins

- **Analyse n'importe quand (*anytime*) ou en temps réel**
  - Monitoring
  - Détection de changements ou de ruptures
    - *Détection de pannes,* *Détection de dérive, ...*
  - Statistiques de résumé
  - Prédiction

Learning tasks

Gathering statistics

- Counts / Histograms
- Frequent items
- Entropy of the signal

Summarizing / Monitoring

- Association rule mining (frequent item sets)
- Change detection
- Clustering

Predicting / Modeling

- Decision trees
- Neural networks / SVMs / HMMs / ...

Flux de données

Quoi de neuf ?

- Environnement potentiellement **non stationnaire**
- Flux **éventuellement infini**
- **Une seule passe par donnée** est possible

Conséquences

- Calculs très simples
- Sans connaissance de l'avenir
  - Nombre d'items, nombre de classes, ...
- Mise à jour adaptative
  - Que conserver en mémoire ?

Covariate shift
Définition

Dérive de $\mathbf{P}_{\mathcal{X}}$

Exemples :

- **Données non stationnaires** ($\mathbf{P}_{\mathcal{X}}$ change mais pas $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$)
  - Médecine et variations saisonnières
  - Filtrage de spam (adaptation d'un groupe d'utilisateurs à un nouvel utilisateur)
- **Biais dans le processus de sélection** en apprentissage
  - Données d'apprentissage ré-équilibrées (mais pas en test)
  - Apprentissage actif
  - Interpolation vs. extrapolation (en régression)

Covariate shift

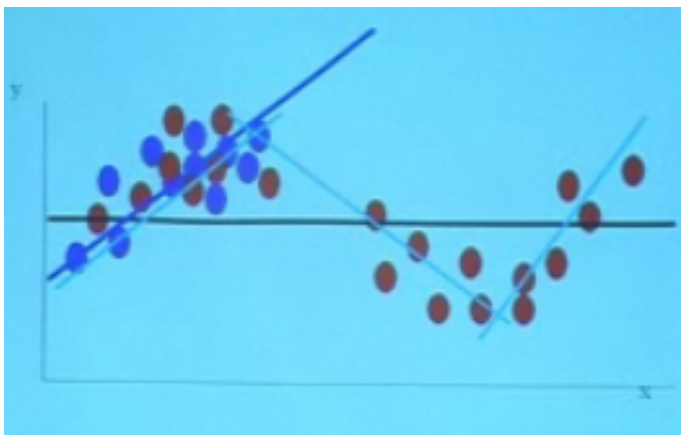
Pourquoi est-ce un problème ?

Plus de lien « direct » garanti entre risque empirique et risque réel

Modifier le critère inductif

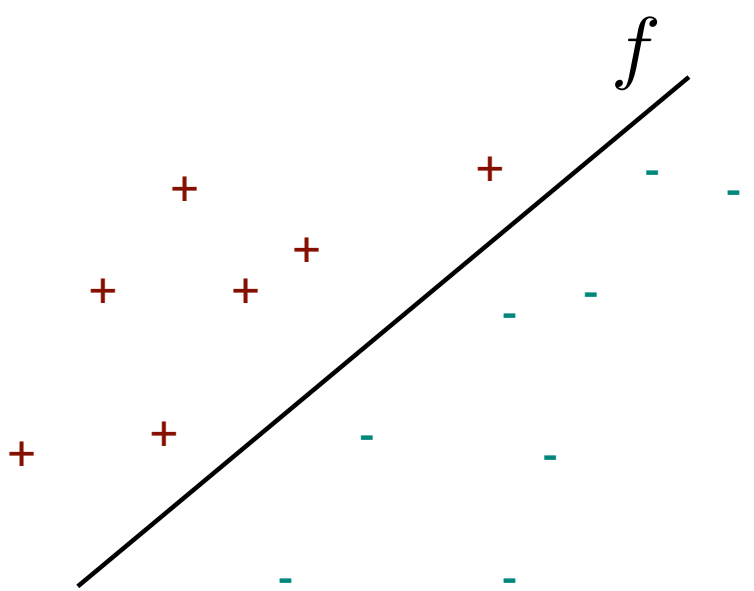
La performance suivant $\mathbf{P}'_{\mathcal{X}}$ (*généralisation*) dépend de :

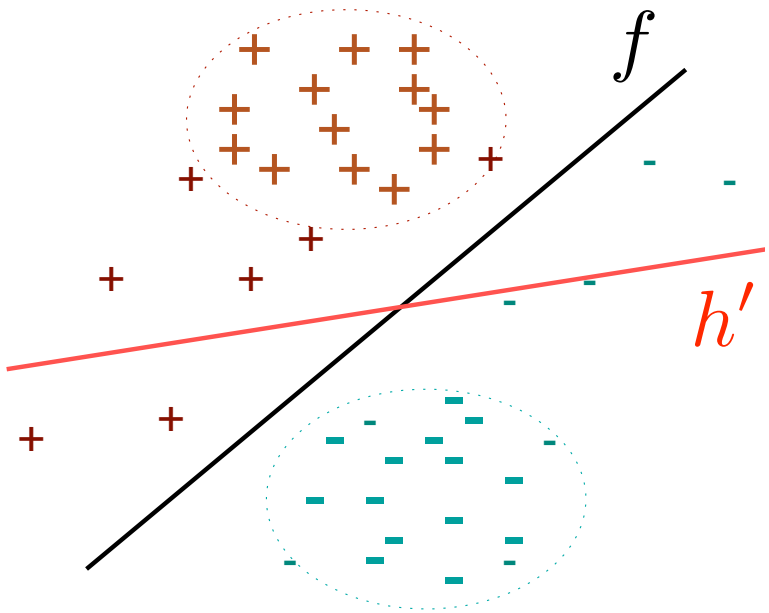
- La performance suivant $\mathbf{P}_{\mathcal{X}}$ (*apprentissage*)
- La similarité entre $\mathbf{P}_{\mathcal{X}}$ et $\mathbf{P}'_{\mathcal{X}}$



Covariate shift

Le problème





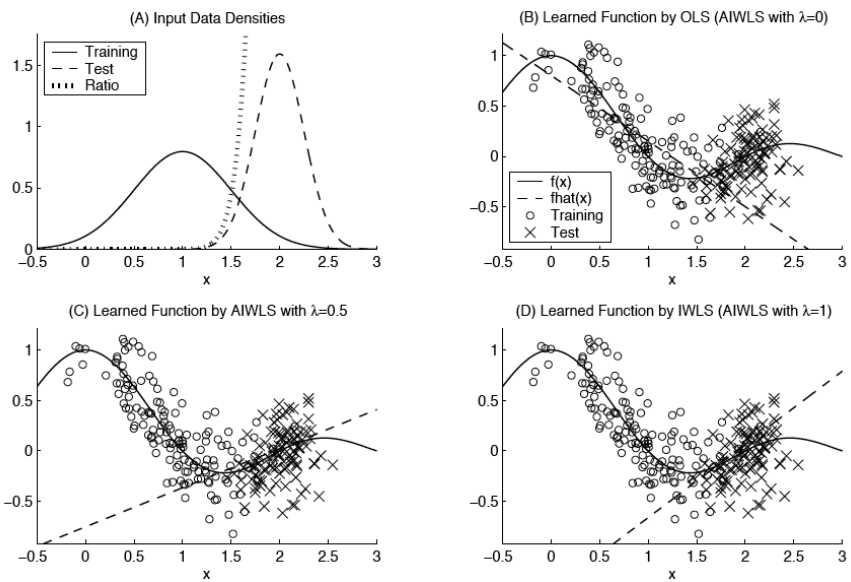
Covariate shift
Approches

"Importance weighted" inductive criterion

Principe : Pondérer l'ERM classique

$$R_{Cov}(h) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\mathbf{P}_{\mathcal{X}'}(\boldsymbol{x}_i)}{\mathbf{P}_{\mathcal{X}}(\boldsymbol{x}_i)}\right)^{\lambda}(h(\boldsymbol{x}_i - y_i)^2$$

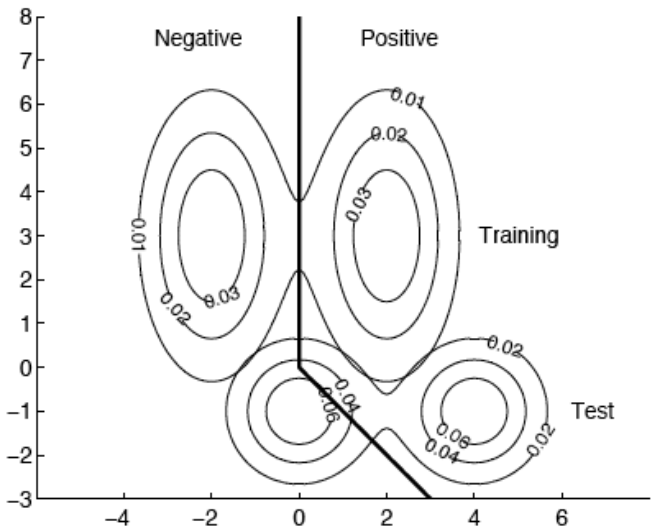
λ contrôle la stabilité / consistance (absence de biais)



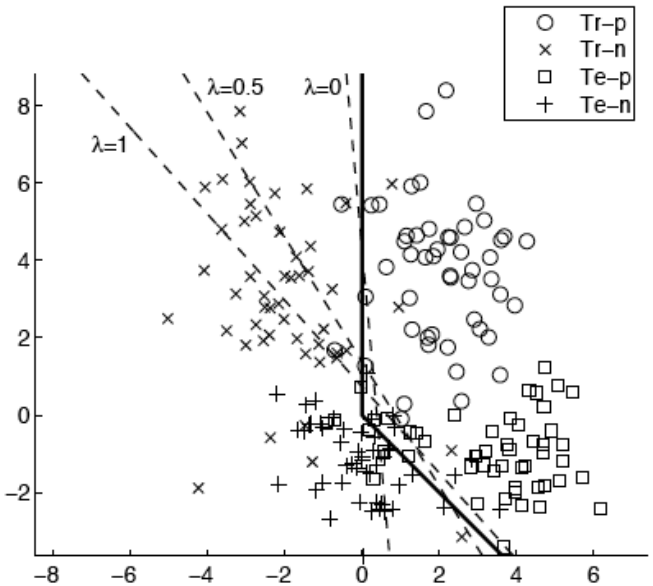
SKM07 M. Sugiyama and M. Kraudelat and K.-R. Müller (2007) *"Covariate Shift Adaptation by Importance Weighted Cross Validation"* Journal of Machine Learning Research, vol.8: 985-1005.

Covariate shift
Approches

"Importance weighted" inductive criterion (en classification)



(a) Contours of training and test input densities.



(b) Optimal decision boundary (solid line) and learned boundaries (dashed lines). '○' and '×' denote the positive and negative training samples, while '□' and '+' denote the positive and negative test samples. Note that the test samples are not given in the training phase; they are plotted in the figure for illustration purposes.

SKM07 M. Sugiyama and M. Kraudelat and K.-R. Müller (2007) *"Covariate Shift Adaptation by Importance Weighted Cross Validation"* Journal of Machine Learning Research, vol.8: 985-1005.

Covariate shift
Approches

"Importance weighted" inductive criterion

Comment obtenir $\frac{\mathbf{P}_{\mathcal{X}'}(\boldsymbol{x}_i)}{\mathbf{P}_{\mathcal{X}}(\boldsymbol{x}_i)}$?

Estimation empirique

Apprentissage semi-supervisé

Apprentissage semi-supervisé

Définition

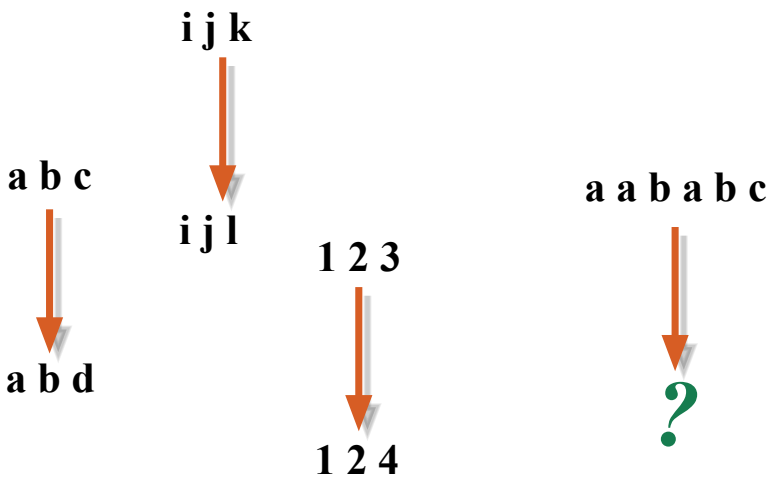
- Étant donné un échantillon d'apprentissage $\mathcal{S}_m = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$, **et** la connaissance de points non étiquetés $\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+l}$ supposés issus i.i.d. de $\mathbf{P}_{calX}$
- Trouver la meilleure fonction de décision $h \in \mathcal{H}$

Idée : Ces données supplémentaires fournissent une information sur $\mathbf{P}_{\mathcal{X}}$

Transduction
Définition

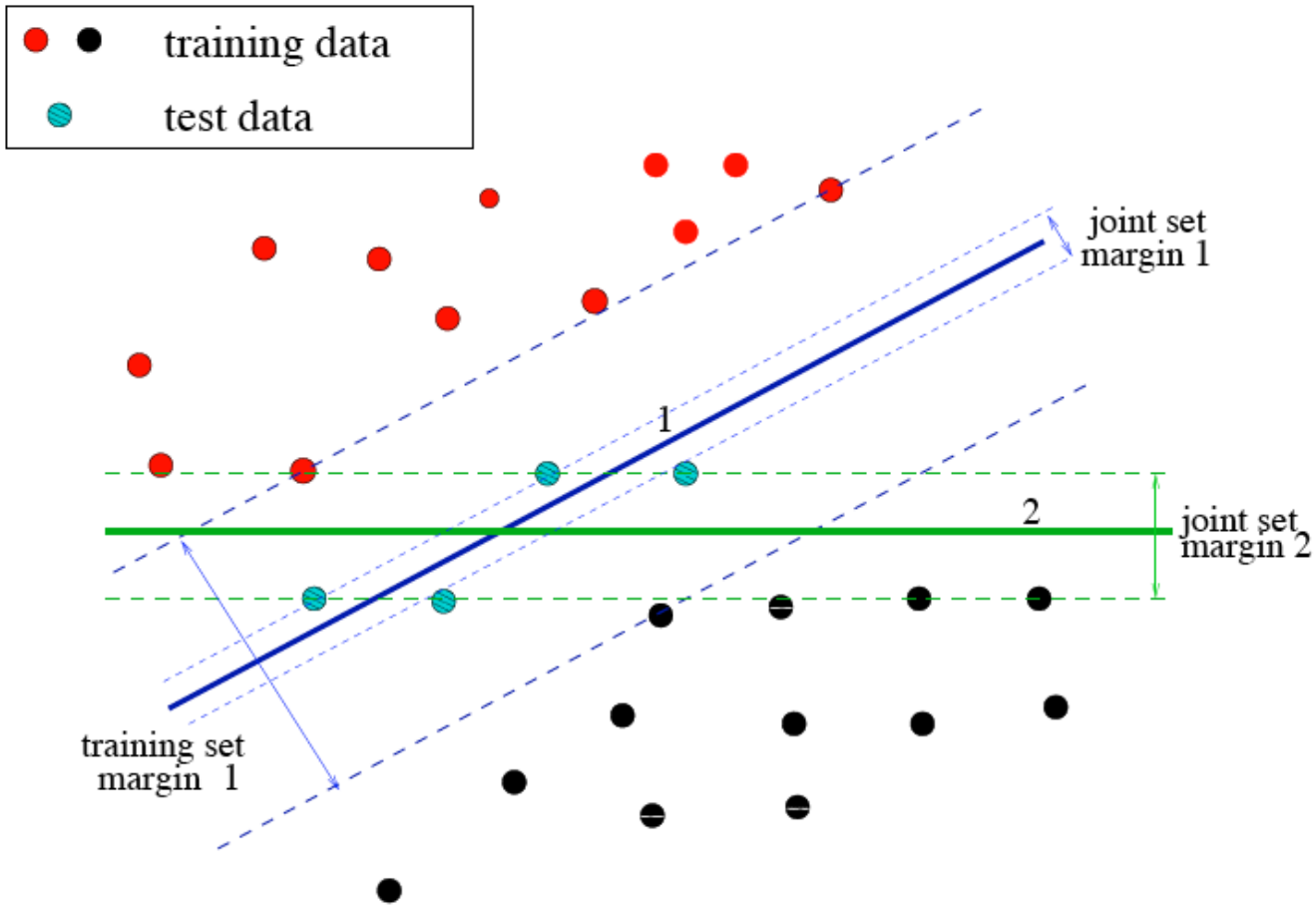
- Étant donné un échantillon d'apprentissage $\mathcal{S}_m = ((\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m))$, **et** la connaissance des points test $\boldsymbol{x}_{m+1}, \ldots, \boldsymbol{x}_{m+k}$
- Trouver le meilleur vecteur de classification $y_{m+1}, \ldots, y_{m+k}$ parmi un ensemble de vecteurs possibles $Y \in \mathcal{Y}^k$

On ne cherche même plus une fonction de décision !!



Transduction

Méthodes

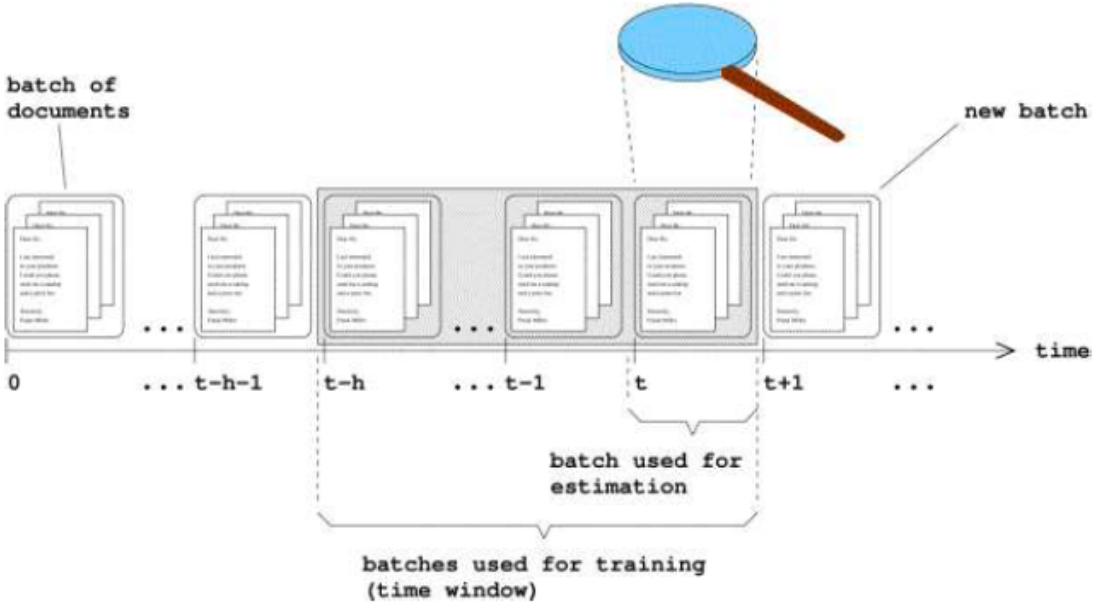


Dérive de concept
Définition

Dérive de $\mathbf{P}_{\mathcal{Y}|\mathcal{X}}$

Exemples :

- Profils de clients (achats en fonction de *revenu, âge, ...*)
- Filtrage de documents en fonction des intérêts de l'utilisateur
- Tracking



Dérive de concept

Problèmes

- Détecter les changements de régime
  - vs. bruit
- Suivre les évolutions mais rester robuste
  - Contrôler l'oubli

Critère de performance : tout au long de la séquence

$$L = \sum_t \ell(h_t(\boldsymbol{x}_t), y_t)$$

ou :
Critère de performance : espérance sur le prochain exemple

$$L = \int_{\mathcal{X}\mathcal{Y}} \ell(h_t(\boldsymbol{x}_{t+1}), y_{t+1}) \mathbf{P}_{\mathcal{X}_{t+1}\mathcal{Y}_{t+1}} d\boldsymbol{x}dy$$

Dérive de concept
Approches

- Plus une donnée est ancienne, plus elle est susceptible d'être périmée

Mais :

- Plus grand est l'échantillon, meilleure est la généralisation

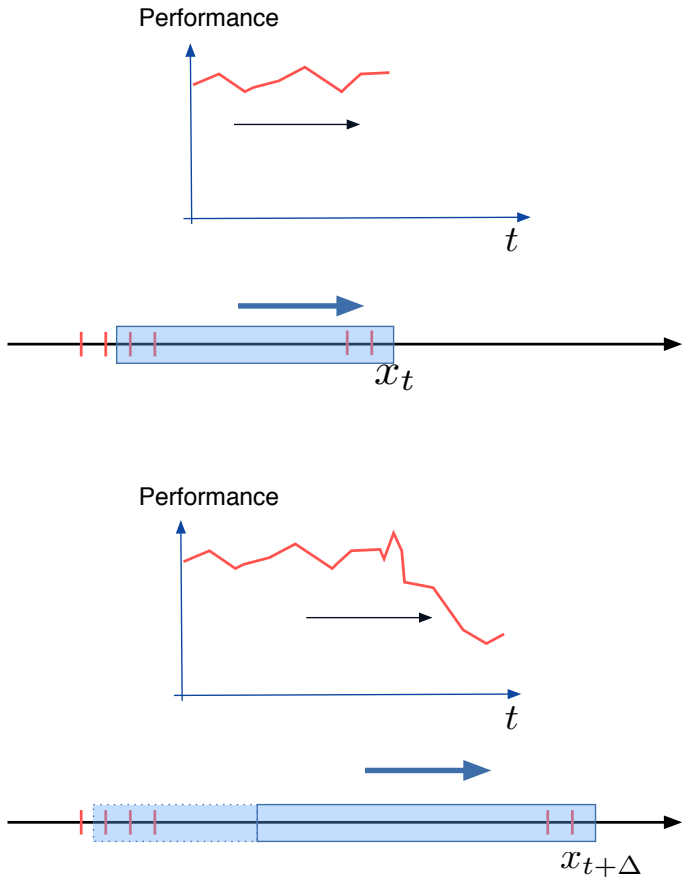
Approches heuristiques

- Utilisation de fenêtres. *Problème : régler leur taille*
- Pondération des exemples en fonction du temps. *Problème : régler le poids*

Dérive de concept

Approche par fenêtres glissantes

Principe :

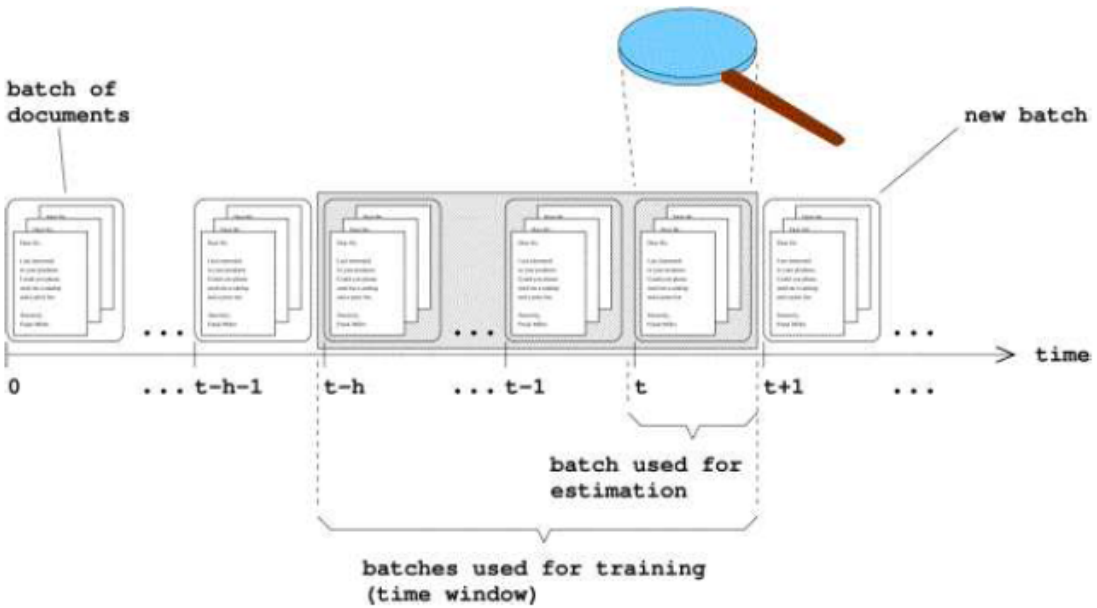


WK96 G. Widmer and M. Kubat (1996) *"Learning in the presence of concept drift ans hidden contexts"* Machine Learning 23: 69–101, 1996.

Dérive de concept

Approche par fenêtres glissantes

- Théoriquement : cherche à minimiser l'espérance d'erreur sur le prochain exemple
- Estimation empirique :
  - Test toutes les tailles de fenêtres
  - sur les exemples du dernier lot ("batch")

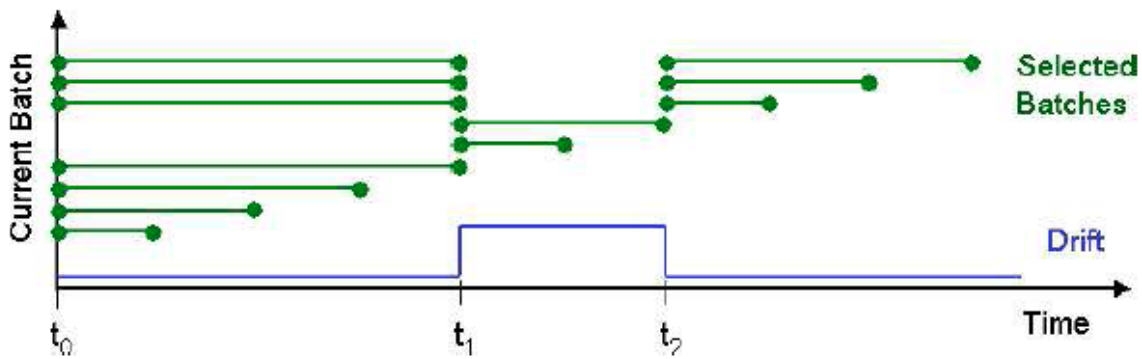


Dérive de concept

Approche par fenêtres glissantes

Deuxième méthode de sélection de fenêtres

- Apprendre un classifieur sur le dernier lot
- Le tester sur toutes les fenêtres précédentes
- Retenir toutes les fenêtres pour lesquelles l'erreur est $< \varepsilon$



SK M. Scholz and R. Klinkenberg (1996) *"Boosting classifiers for drifting concepts"* Intelligent Data Analysis (IDA) Journal, Volume 11, Number 1, March 2007.

Dérive de concept

Approche par pondération des exemples

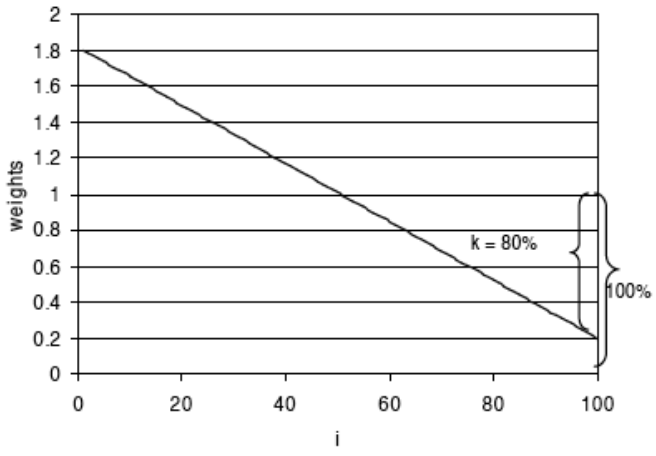


Figure: $n = 100; k = 80\%$

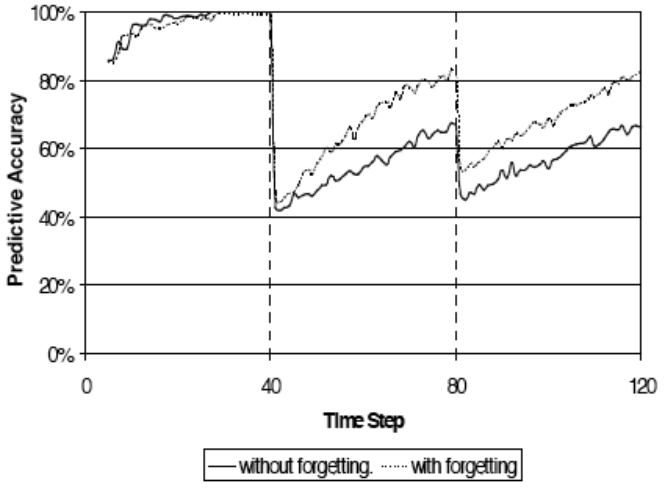


Figure: Test sur le problème STAGGER avec C4.5

$$w_i = -\frac{2k}{n-1}(i-1) + 1 + k$$

Koy00 I. Koychev (2000) *"Gradual forgetting for adaptation to concept drift"* European Conf. On Artificial Intelligence (ECAI-00) , Workshop 'Current Issues in Spatio-Temporal Reasoning'.

Dérive de concept

Approche par Boosting

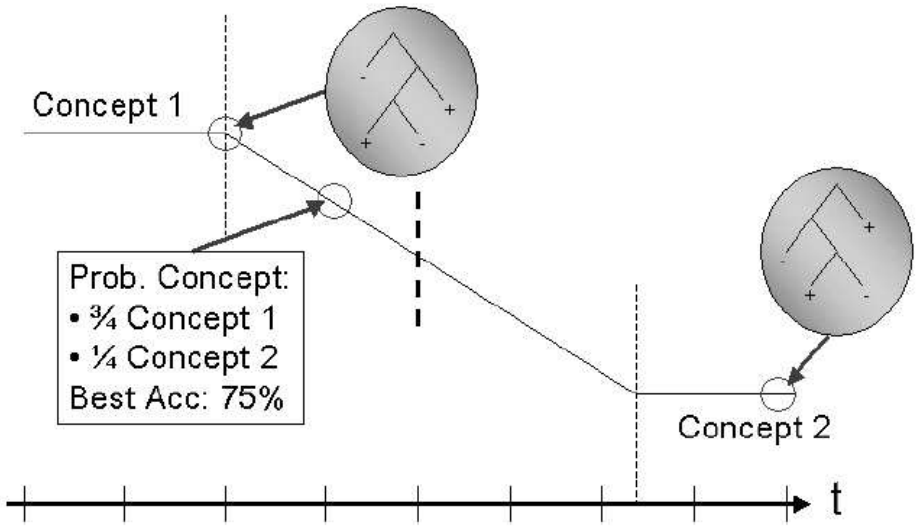


Fig. 4. Continuous concept drift, starting with a pure *Concept 1* and ending with a pure *Concept 2*. In between, the target distribution is a probabilistic mixture. It is optimal to predict *Concept 1* before the dotted line, and *Concept 2*, afterwards.

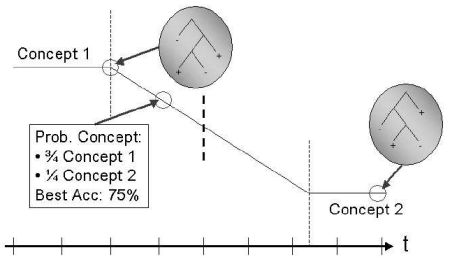
Comment commencer à apprendre concept 2 avant la fin de la transition ?

Dérive de concept

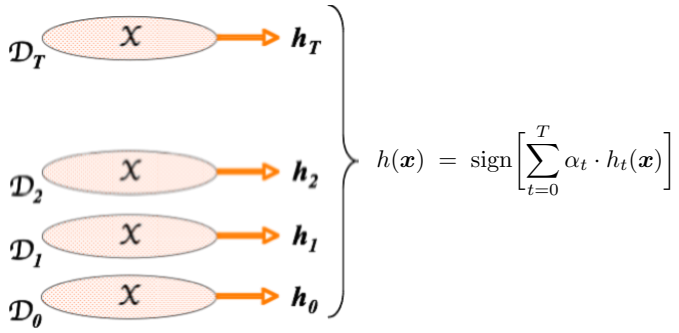
Approche par Boosting

Principe : modifier graduellement la distribution des exemples

en « soustrayant » la distribution correspondant à `concept 1`.



$$\forall \langle \boldsymbol{x}, y \rangle \in \mathcal{X} \times \mathcal{Y} : P_{D'}(\boldsymbol{x}, y) = P_D(\boldsymbol{x}, y) \cdot \frac{P_D[h_{t-1}(\boldsymbol{x}) = \hat{y}] \cdot P_D[y = y*]}{P_D[h_{t-1}(\boldsymbol{x}) = \hat{y}, y = y*]}$$



Dérive de concept

Bilan

Heuristiques

- Efficaces dans leur domaine d'application particulier
- Requièrent un réglage fin
- Non transferables facilement à autres domaines
- Manquent de fondations théoriques

Analyses théoriques

Sous quelles conditions peut-on PAC apprendre avec une erreur de ε ?

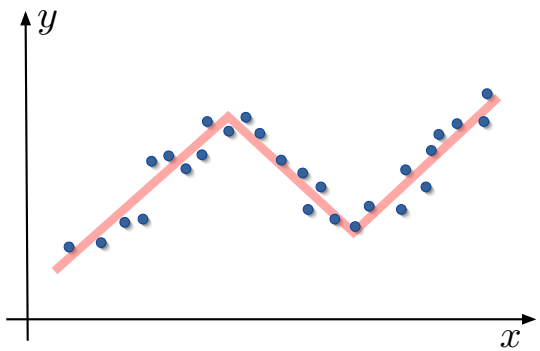
- Dépend de $d_{\mathcal{H}}$ et de la vitesse de dérive v
- Possible si $v = \mathcal{O}(\varepsilon^2 / d_{\mathcal{H}}^2 \ln \frac{1}{\varepsilon}))$
- Rq : protocole adverse

HL94 D. P. Helmbold and P. M. Long (1994) *"Tracking drifting concepts by minimizing disagreements"* Machine Learning 14 (1): 27–45, 1994.

Tracking
Définition

Ressources limitées :

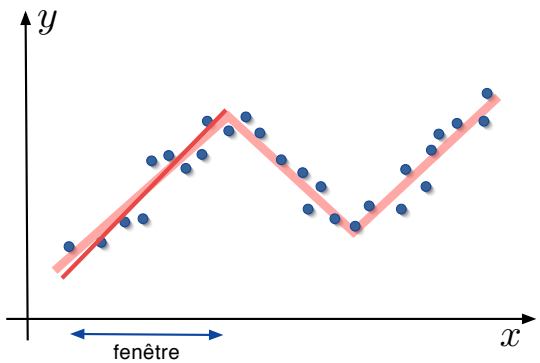
- Impossible de traiter toutes les données
- Flux de données
- Espace des hypothèses $\mathcal{H}$ trop restreint



Apprentissage « local »

et prédiction locale :

$$\begin{aligned} L_t &= \ell(h_t(\boldsymbol{x}_t), y_t) \\ &= \ell(h_t(\boldsymbol{x}_t), f(y_t, \theta_t)) \end{aligned}$$



Tracking
Analyse

Le tracking est-il plus performant ?

Grande nouveauté

Le coût de calcul entre dans le critère inductif

$$\begin{aligned}\varepsilon_{Tot} &= \mathbb{E}[E(h^*_{\mathcal{H}} - E(f))] + \mathbb{E}[E(\hat{h}_m) - E(h^*_{\mathcal{H}})] + \mathbb{E}[E(\tilde{h}_m - E(\hat{h}_m)] \\ &= \varepsilon_{\textbf{approximation}} + \varepsilon_{\textbf{estimation}} + \varepsilon_{\textbf{optimisation}}\end{aligned}$$

Le tracking :

- \- Erreur d'approximation plus forte ($\varepsilon_{\textbf{aproximation}}$)
- \+ $\mathcal{H}$ plus restreint : meilleure $\varepsilon_{\textbf{estimation}}$
- \+ Optimisation plus facile : meilleure $\varepsilon_{\textbf{optimisation}}$

BB08 Léon Bottou and Olivier Bousquet (2008) *"The Tradeoffs of Large Scale Learning"* (NIPS-07) Advances in Neural Information Processing Systems, 20, MIT Press, Cambridge, MA, 2008 (to appear).

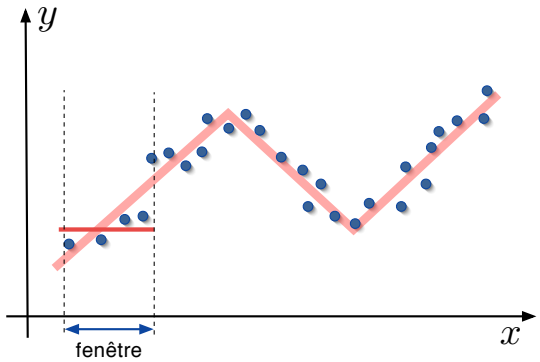
Tracking
Analyse

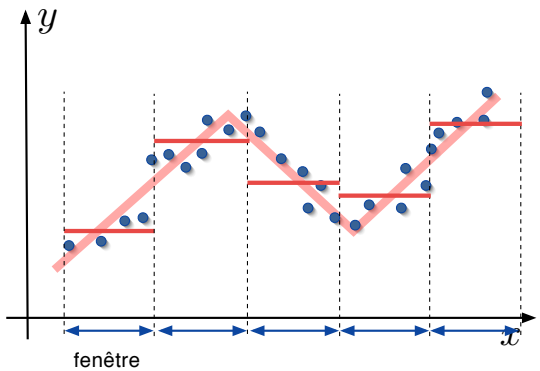
Notion de *cohérence temporelle*

$f(\cdot, \theta_t)$ continue et à variation bornée / θ_t

Nouveau critère inductif

$$L_{\langle 0,T\rangle} = \sum_{t=0}^{T} \ell(h_t(\boldsymbol{x}), y_t) + \text{Capacité}(\mathcal{H}) + \lambda \sum ||h_t - h_{t-1}||^2$$





On ne cherche plus à optimiser le choix de h

mais à **optimiser la règle d'apprentissage** $(h_{t-1}, \boldsymbol{x}_t) \rightarrow h_t$!!

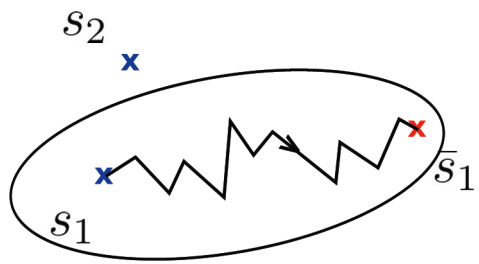
Transfert
Définition

Ré-utilisation de connaissances acquises dans un contexte (pour résoudre une tâche) dans un autre contexte (pour résoudre une autre tâche)

Éventuellement changement de domaine

Exemples :

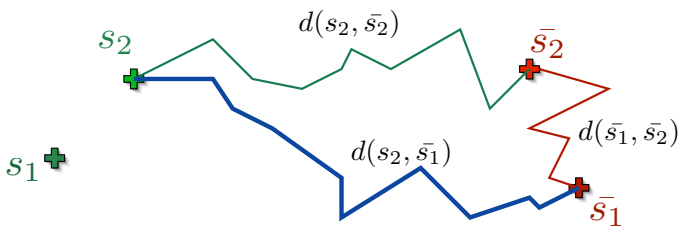
- Adaptive A* / Planification
- Règles apprises pour classer des factures utilisées pour classer des réclamations
- Analogie : $a\ b\ c \rightarrow a\ b\ d\ :\ i\ j\ k \rightarrow ?$

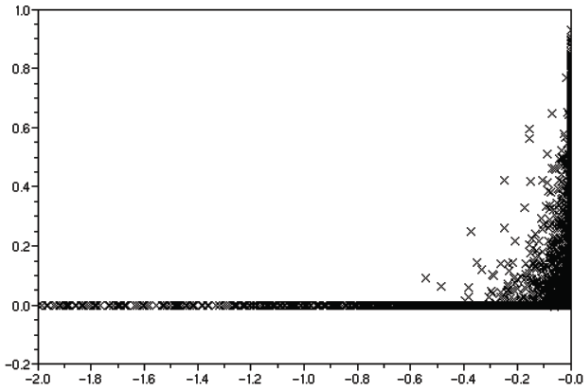


Transfert
Questions

Sous quelles conditions le transfert est avantageux ?

(et quand vaut-il mieux repartir de 0 ?)





FC08 L. Fedon et A. Cornuéjols (2008) *"Comment optimiser A* adaptatif"* Proc. of RFIA-08, 2008.

Apprentissage guidé

Choix des exemples effectué par un professeur

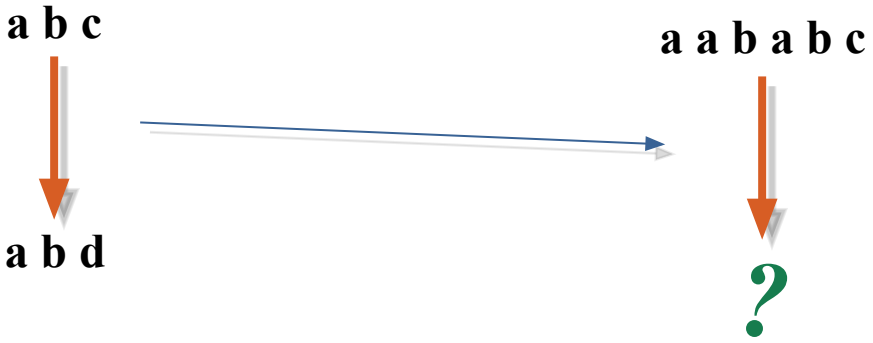
Questions

- Comment fournir **la séquence** d'exemples la plus efficace
  - pour un apprenant
  - dans un but donné

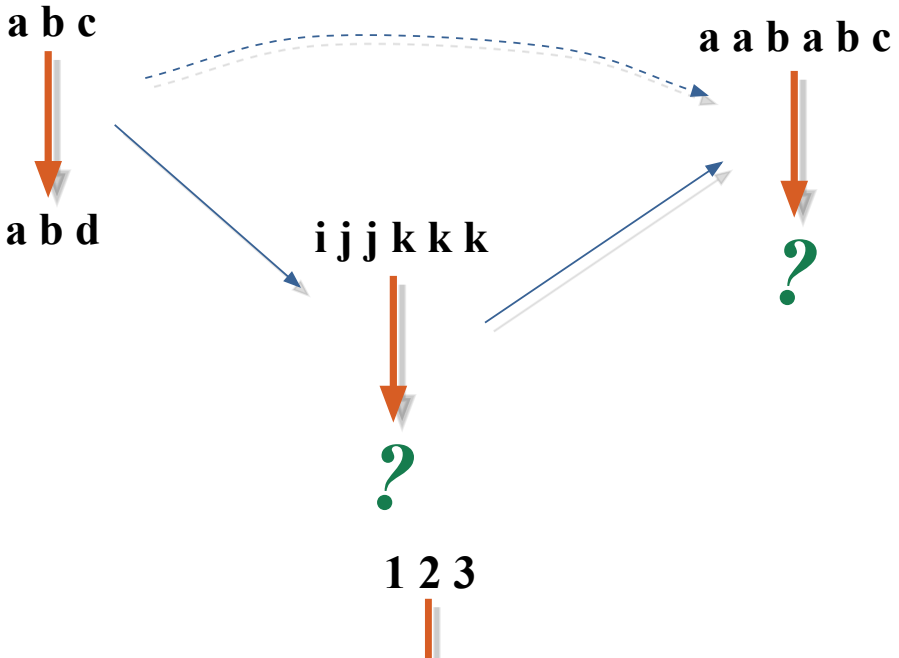
Les effets de séquences

Exemple

[Sur 24 étudiants de DEA, 1996]



- Long et difficile
- Grande variété de réponses



- Beaucoup plus rapide
- Spectre de réponses beaucoup plus serré

Questions

Effets de l'ordre

- Qu'est-ce qui les caractérise ?
  - Optimalité
  - Oubli du passé
- Comment les éliminer ?
- Comment les utiliser ?
- Que révèlent-ils sur la notion d'information transmise d'un état au suivant ?

Quelle **mémoire du passé** ? Que peut-on réutiliser ? Doit-on réutiliser ?

Nouveau critère inductif ?

Conclusions

1. **Apprentissage artificiel : science bien établie**
   - Nombreuses méthodes
   - Bien fondées
   - Mais la théorie repose sur le cadre stationnaire et i.i.d.
2. **Des méthodes d'apprentissage en-ligne existent**
   - de nature heuristique
   - avec peu de fondements
3. **On sort progressivement du cadre i.i.d.**
   - Covariate shift, transduction, dérive de concept, ...
   - Nouveaux critères inductifs
4. **L'avenir est à l'apprentissage continu**
   - Optimisation de la règle d'apprentissage
   - Science des transferts d'information
   - *oubli, changement de représentation, ...*
5. **Une nouvelle science de l'apprentissage en-ligne**
   - ... doit naître et se développer

L'avenir ...

... commence ici !

MERCI !