Préparation Quizz
OPT6 : Apprentissage Avancé

Les castors juniors

Décembre 2017

1 Quizz 1

Déjà passé.

2 Quizz 2

2.1 Questions

Dans quel cas le principe de minimisation du risque empirique est-il sain ?

S'il y a des contraintes sur l'espace des hypothèses.

Qu'est-ce qu'une hypothèse de risque empirique nul ?

Pas d'erreur sur l'échantillon d'apprentissage.

Qu'est-ce qu'un cas réalisable ?

$\exists h \in H$ tq $R(h) = 0$ (Risque réel, pas empirique) Avec le perceptron ça veut dire qu'il va converger surtout. Si y'a pas de cas réalisable, le perceptron convergera pas.

Qu'est-ce que iid ?

Indépendemment identiquement distribué.
Each random variable has the same probability distribution as the others and all are mutually independent. Sur un ensemble de test ça permet d'être sûr que l'ensemble est bien représentatif des données réelles (?)

La dimension Vapnik-Chervonenkis ?

La dimension VC (pour dimension de Vapnik-Chervonenkis) est une mesure de la capacité d'un algorithme de classification statistique. Elle est définie comme le cardinal du plus grand ensemble de points que l'algorithme peut pulvériser.

C'est une sorte de mesure de la complexité d'un modèle.

— Séparation par une droite : $D_{VC} = 3$
— Séparation par un rectangle : $D_{VC} = 4$ (je crois)

Pulvérisation ?

Soient C une classe d'ensembles et A un ensemble. On dit que C pulvérise A si et seulement si, pour tout sous-ensemble T de A, il existe un élément U appartenant à C tel que

$U \cap A = T$

Ceci équivaut encore à dire que C pulvérise A si et seulement si l'ensemble des parties de l'ensemble A, P(A), est égal à l'ensemble $U \cap A | U \in C$.

Par exemple, la classe C des disques du plan (lorsqu'on se place dans un espace à deux dimensions) ne peut pas pulvériser tous les ensembles F de quatre points, alors qu'en revanche la classe des ensembles convexes du plan pulvérise tout ensemble fini du cercle unité.

Apprenant faible / apprenant fort ?

— **Apprenant faible :** Qui fait un tout petit peu mieux que la moyenne
— **Apprenant fort :** PAC learning

Y a-t-il un rapport entre les deux ?

Oui, on peut faire un apprenant fort avec plein d'apprenants faibles (== boosting)

PAC Learning ?

Probably Approximately Correct.

Qu'est-ce qu'un decision stump ?

Arbre de décision à un seul noeud. Typiquement un apprenant faible. Découpe l'espace en deux selon un des attributs.

Comment engendrer des apprenants faibles décorrélés ?

En modifiant l'échantillon d'apprentissage à chaque étape : on diminue l'importance des exemples bien classés et on augmente celle des exemples mal classés.

Consistance ?

Qu'est-ce que le principe de consistance universelle ?

Les résultats de la consistance universelle sont des résultats asymptotiques. Ils disent juste que si nous avons suffisamment de données (et un ordinateur suffisamment puissant pour les traiter) alors tout algorithme universellement consistant produira une prédiction très proche de la meilleure prédiction possible.

La loi des grands nombres (Law of Large Numbers) ?

Exprime le fait que les caractéristiques d'un échantillon aléatoire se rapprochent des caractéristiques statistiques de la population lorsque la taille de l'échantillon augmente.

Qu'est-ce que le boosting ?

Une méthode générale pour convertir des règles de prédiction peu performantes en une règle de prédiction très performante.

La sagesse des foules (wisdom of crowd) ?

Le fait d'élire Sarko puis Hollande puis Macron. C'est pas plutôt genre les paniers de bouffe dans les marchés ? Ah oui peut être. Les gens sont cons mais tous ensemble leur moyenne donne l'espérance d'une loi normale avec une bonne précision.

2.2 Formules

Pour l'algorithme du boosting :

Soit e le taux d'erreur d'une hypothèse h :

- $\alpha = \frac{1}{2} ln(\frac{(1-e)}{e_1})$ (c'est le coeff de l'hypothèse dans le vote à la fin)
- $P_b(x) = \frac{1}{(2(1-e))}$ (on multiplie l'ancien coeff des valeurs bien prédites par ça pour avoir leurs nouveaux coeffs)
- $P_m(x) = \frac{1}{2e}$ (on multiplie l'ancien coeff des valeurs mal prédites par ça pour avoir leurs nouveaux coeffs)

3 Quizz 3

3.1 Questions

Structural Risk Minimization ?

Stratification des espaces d'hypothèses : faite a priori (indépendamment des données), par exemple la d_{vc}. Expliqué ici : http ://www.svms.org/srm/srm.pdf

Structural risk minimization (SRM) (Vapnik and Chervonenkis 1974) is an inductive principle for model selection used for learning from finite training data sets. It describes a general model of capacity control and provides a tradeoff between hypothesis space complexity (the VC dimension of approximating functions) and the quality of fitting the training data (empirical error). The procedure is outlined below.

- Using a priori knowledge of the domain, choose a class of functions, such as polynomials of degree n, neural networks having n hidden layer neurons, a set of splines with n nodes or fuzzy logic models having n rules.
- Divide the class of functions into a hierarchy of nested subsets in order of increasing complexity. For example, polynomials of increasing degree.
- Perform empirical risk minimization on each subset (this is essentially parameter selection).
- Select the model in the series whose sum of empirical risk and VC confidence is minimal

Qu'est-ce que le compromis biais-variance ?

C'est le problème de minimiser simultanément deux sources d'erreurs qui empêchent les algorithmes d'apprentissage supervisé de généraliser au-delà de leur échantillon d'apprentissage :

- Le biais est l'erreur provenant d'hypothèses erronées dans l'algorithme d'apprentissage. Un biais élevé peut être lié à un algorithme qui manque de relations pertinentes entre les données en entrée et les sorties prévues (sous-apprentissage).
- La variance est l'erreur due à la sensibilité aux petites fluctuations de l'échantillon d'apprentissage. Une variance élevée peut entraîner un surapprentissage, c'est-à-dire modéliser le bruit aléatoire des données d'apprentissage plutôt que les sorties prévues.

Luckiness Framework ?

Principe : consiste à définir un ordre sur H qui dépend des données. Si nous avons de la chance alors il n'y aura pas trop d'hypothèses mauvaises aussi compatibles avec la cible que les bonnes.

Que devient l'apprentissage dans ce cas ? (dans le cas théorie de l'apprentissage ou luckiness framework ou je sais pas trop quoi)

- Le choix d'espace d'hypothèse H (nécessairement contraint),
- Choix d'un critère inductif (risque empirique nécessairement régularisé),
- Une stratégie d'exploration de H pour minimiser le risque empirique régularisé (faire ce qu'il faut pour que l'exploration soit efficace : rapide et si possible un seul optimum).

C'est quoi la nouvelle perspective d'un problème d'apprentissage ?

Poser un pb d'apprentissage, c'est :

- L'exprimer sous forme d'un critère inductif à optimiser. Risque empirique (avec une fonction d'erreur adéquate), un terme de régularisation (exprimant les contraintes, et connaissances a priori, si possible conduisant à un problème convexe)
- trouver un algorithme d'optimisation.

Qu'est-ce qu'une hypothèse parcimonieuse ?

les hypothèses suffisantes les plus simples sont les plus vraisemblables (=rasoir d'd'Ockham)

Qu'est-ce qu'une méthode de type LASSO ?

En statistiques, le lasso est une méthode de contraction des coefficients de la régression développée par Robert Tibshirani dans un article publié en 1996 intitulé Regression shrinkage and selection via the lasso. La fonction de coût peut engendrer de l'instabilité dans les résultats de l'estimation car aucune contrainte est imposée au modèle (coefficient dans le cas de la regression). Il est donc possible (si grandes valeurs) d'obtenir des espace de solution très étendues, ce qui peut générer ce qu'on appelle des "vallées" ou "arêtes". Ces vallées sont problématiques car différents vecteurs d'estimation peuvent mener à des solutiosn proches en terme de minimisation de l'erreur, ce qui provoque, en cas de changement de données d'apprentissage (même faible), de possibles variations importantes et donc une instabilité du modèle. Régulariser permet de limiter les grands coefficients en resserant l'espace autour de 0, ce qui supprime els vallées. La fonction de cout est donc de la forme R(h) = l(y,y') + P(lambda, théta) ou P(lambda, théta) gère la pénalité. Cette fonction est définie par la norme l1 (l2 pour ridge) soit $l1_\theta = |\theta_1| + |\theta_2| + .. + |\theta_n|$ (norme de manhattan).Ainsi, une minimisation dans le cas de lasso et de la regression revient à considérer la fonction de cout comme : $J(\theta) = l(y, y') + \lambda \sum_{j=1}^{n} |\theta_j$ où lambda est "l'importance du lasso" et theta l'ensemble des coefficients de la régression. Le lasso permet de simplifier le modèle en considérant des coefficients nuls, il est dnc possible de supprimer des coeff peu "discriminants" . Cette méthode est sensible aux corrélations entre variables alors que ridge(l2 distance) l'est bcp moins. Comment c'est appliqué aux autres méthodes ? don't know

Quels sont les avantages d'un modèle linéaire

Il y a des coefficients et des importances associées à des grandeurs ; ils sont **compréhensibles** et interprétables et ils donnent des problèmes d'optimisation **convexes** (contrairement aux réseaux de neurones par ex).

Qu'est-ce que l'apprentissage multi-tâches ?

Un apprentissage ou l'on apprend plusieurs taches à la fois tout en utilisant les mêmes données. Exemple : prédiction de critères morphologique : âge, genre, émotion, accessoires (lunettes, etc.)

Multi-task learning (MTL) is a subfield of machine learning in which multiple learning tasks are solved at the same time, while exploiting commonalities and differences across tasks. This can result in improved learning efficiency and prediction accuracy for the task-specific models, when compared to training the models separately.[1][2][3] Early versions of MTL were called "hints"[4][5]. It aims

to improve the performance of multiple classification tasks by learning them jointly. One example is a spam-filter, which can be treated as distinct but related classification tasks across different users. To make this more concrete, consider that different people have different distributions of features which distinguish spam emails from legitimate ones, for example an English speaker may find that all emails in Russian are spam, not so for Russian speakers. Yet there is a definite commonality in this classification task across users, for example one common feature might be text related to money transfer. Solving each user's spam classification problem jointly via MTL can let the solutions inform each other and improve performance.[6] Further examples of settings for MTL include multiclass classification and multi-label classification.[7]

Quelles sont les garanties de l'apprentissage statistique et quelles sont les conditions ?

Garantie qu'il y a un lien entre le risque empirique et le risque réel, mais seulement si on est dans un monde stationnaire, les données sont iid et les questions sont iid. Ne dit rien sur l'intelligibilité, la fécondité et l'insertion dans une théorie du domaine.

Quelles sont les limites de l'apprentissage statistique ?

- Apprentissage passif et données et questions iid (agents situés : le monde n'est pas iid.)
- requiert beaucoup d'exemples (nous sommes beaucoup plus efficaces ; nous on est "producteurs de théories", théories que nous testions ensuite. On est tout le temps en train d'essayer d'expliquer).
- Pas adapté à la recherche de causalités
- pas intégré avec un raisonnement.

C'est quoi l'apprentissage semi-supervisé ?

L'apprentissage semi-supervisé est une classe de techniques d'apprentissage automatique qui utilise un ensemble de données étiquetées et non-étiquetés. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non-supervisé qui n'utilise que des données non-étiquetées. Un exemple d'apprentissage semi-supervisé est le co-apprentissage, dans lequel deux classifieurs apprennent un ensemble de données, mais en utilisant chacun un ensemble de caractéristiques différentes, idéalement indépendantes. Si les données sont des individus à classer en hommes et femmes, l'un pourra utiliser la taille et l'autre la pilosité par exemple. TODO peut etre expliquer plus concrètement.

Ca marche parce que : Let's assume that it is reasonable that the fronierer between two classes does not cut through high density regions of the input space X. Then the unlabeled data points bring constraints on the possible decision functions -¿ gain of information.

Qu'est-ce que le co-apprentissage ?

On apprend 2 classifieurs en même temps indépendants l'un de l'autre. Par exemple voir question d'avant.

Qu'est-ce qu'une compatibility function ?

Could be an increasing function of the distance of x to the decision function (separator) h.

Comment avoir des garanties sur l'apprentissage semi supervisé ?

TODO reprendre les slides du cours 109 à 114.

Qu'est-ce que le Tracking ?

The learning agent receives inputs that are driven by a time dependent process. It therefore encounters different parts of the environment at different times. Event though the world involves a piecewise linear law, the learning agent may perform well by maintaining a very simple model, a constant, over its local environment.

Temporal consistency (small memory, simpleH) ¡-¿ iid data (large memory, "complex" H).

Qu'est-ce que le théorème du lampadaire ?

SI le signal présent les propriétés supposées a priori, alors la méthode assure que le signal cible (d'origine) sera (quasi) reconstruit.

Qu'est-ce que l'Explanation-Based Learning ?

An Explanation-based Learning (EBL) system accepts an example (i.e. a training example) and explains what it learns from the example. The EBL system takes only the relevant aspects of the training. This explanation is translated into particular form that a problem solving program can understand. The explanation is generalized so that it can be used to solve other problems. Un exemple d'application à l'EBL est un programme de jeu d'échecs qui apprendrait à partir d'exemples de jeu. Une position spécifique du jeu d'échecs qui contient une combinaison importante, par exemple, "perdre la reine noire en deux coups", contient aussi des informations inutiles, par exemple la disposition des pièces qui n'interviennent pas dans ladite combinaison. EBL peut, à partir d'un seul exemple d'application, déterminer les informations importantes pour induire des règles de généralisation

Fonction de perte surrogée (surrogate loss function) ?

The 0-1 loss function has nice properties that we would like to take advantage of for many problems. However, because it is not convex, it is difficult to optimize using the 0-1 loss function, so we often turn to convex surrogate loss functions, which are convex and easy to optimize (for example hinge loss). Fonction de l forme $l(x, y) = \phi(-xy)$

Qu'est-ce que la hinge loss ?

Souvent pour les SVM. $l(y) = max(0, 1 - t * y)$, for an intended output t=0 or t=1 and a classifier score y.

Qu'est-ce que le bagging ?

- Généralisation de k échantillons "indépendants" par tirage avec remise dans l'échantillon S_m, (63% d'éléments uniques en moyenne).
- Pour chaque échantillon, apprentissage d'un classifieur en utilisant le même algorithme d'apprentissage.
- La prédiction finale pour un nouvel exemple est obtenue par vote (simple) des classifieurs.

TODO : expliquer mieux. Bagging : on rajoute juste une règle de décision, alors que si on rajoute par exemple un svm à la fin c'est du boosting.

- **Bagging :** tous les classifieurs sont entraînés en parallèle, contrairement au boosting où un classifieur dépend du précédent.

Random forest ?

Principe : réduire la corrélation entre apprenants faibles. Créer B nouveaux ensembles d'apprentissage par un double processus d'échantillonage (sur les observations, en utilisant un tirage avec remise d'un nombre N d'observations identique à celui des données d'origine (technique connue sous le nom de bootstrap), et sur les p prédicteurs, en n'en retenant qu'un échantillon de cardinal $m < \sqrt{p}$ (la limite n'est qu'indicative). Puis, sur chaque échantillon, on entraîne un arbre de décision selon une des techniques connues en limitant sa croissance par validation croisée. On stocke les B prédictions de la variable d'intérêt pour chaque observation d'origine. La prédiction de la forêt aléatoire est alors un simple vote majoritaire (Ensemble learning).

Le risque empirique régularisé ?

C'est un critère inductif qui consiste à d'abord Satisfaire les contraintes posées par les exemples, puis à Choisir le meilleur espace d'hypothèses (capacité de H)

Qu'est-ce que la marge ?

poids des classifieurs ayant voté correctement - poids des classifieurs en erreur. Sert à estimer la confiance d'une prédiction.

Adaboost ?

On choisit un bon classifieur sur les données, puis on choisit son poids avec une formule qui dépend du taux d'erreur, puis on met à jour la pondération des exemples d'apprentissage, etc.

Complexité de Kolmogorov

The Kolmogorov complexity of an object, such as a piece of text, is the length of the shortest computer program that produces the object as output.

Qu'est-ce que l'apprentissage actif ?

L'algorithme demande des informations pour des données précises. Intervention d'un oracle.

4 Quizz 4

4.1 Notes générales

Les flux de données

Une séquence a une structure interne dont on ne peut pas permuter les éléments sans modifier le problème.

On peut avoir deux types de séquences de mesures :
- A : $S = X_1, X_2, ...X_t$
- B : $S =< (X_1, Y_1), (X_2, Y_2), ...(X_t, Y_t) >$ (prédit x, puis on donne y la réponse).

Si je suis dans le cas A, une tâche classique est la prédiction :
- Pour une séquence. Les outils : la régression linéaire, chaînes de markhov, HMM s'il y a des états cachés, les grammaires (avec automates à états finis). Limite de ces outils : ça s'applique sur des symboles, alors qu'on a des mesures réelles qu'il faut **discrétiser** et il faut savoir comment le faire. On peut discrétiser avec des RNN.
- Dans le cas de plusieurs séquences : clustering (groupes d'évolution similaires), classification supervisée (pour une nouvelle séquence, à quel groupe l'attribuer ?). Il faut définir une **distance** appropriée ! Par exemple, si on veut comparer des génomes, distance d'édition, puis programmation dynamique pour aligner les séquences.

Le cas B : l'apprentissage en ligne.

Scénario

$x_1 \to y_1 = h_1(x_1) \to y_1$ $x_2 \to y_2 = h_2(x_2) \to y_2$

x1 arrive, je fais ma prédiction h1, et on me fournit y1.

Pourquoi ce scénario ?

Motivations :
- *Cadre Anytime* : on doit pouvoir donner n'importe quand une réponse. On calcule tout le temps une hypothèse pour qu'elle soit prête le jour où on nous demande une réponse.
- *Environnement non stationnaire.* On ne veut pas garder la même hypothèse tout le temps. La distribution des exemples $P_{XY} = P_{Y|X}P_X$.. On fait la distinction entre deux types de changements :
  - Le changement peut opérer sur P_X : Covariate shift. le concept ne change pas mais l'environnement change. Par exemple, distribution des patients pour un médecin en hiver ou en été, mais pas de changement des symptômes. : quand mon concept cible ne bouge pas mais en moyenne je ne rencontre pas les mêmes événements.
  - Si le changement opère sur $P_Y|X$: dérive de concept (concept drift), le concept cible bouge. Pour l'exemple du médecin, ça serait les symptômes qui ne sont plus associées aux mêmes maladies.
- *Données très volumineuses* : Si j'ai pas assez de mémoire pour faire de l'apprentissage batch.

Domain adaptation : Par exemple, on apprend à reconnaître des images sous un certain type d'éclairage, et on nous demande d'être capable de reconnaître ces objets avec une luminosité différents. Plutôt lié au covariate shift.

Apprentissage par transfert : Notion plus forte : j'apprends à jouer aux dames et on me demande de jouer aux échecs. Vrai changement de tâche.

Incremental learning : En général, c'est en environnement stationnaire, mais on se demande comment on peut apprendre au mieux un concept compliqué. On ne peut pas l'apprendre d'un coup. Quel est le meilleur ordre des exemples (curriculum), la meilleure séquence pédagogique pour que le système apprenne au mieux ?

Batch learning : Cadre classique.

Comment évaluer l'apprentissage dans ces contextes là ?

On ne peut plus faire comme avant : on n'a pas d'environnement stationnaire. Avant, on se disait que si on minimise critère inductif, ça va donner une performance. Or, maintenant on ne sait pas comment ça va se passer dans le futur. On ne peut plus utiliser le risque empirique.

On va donc bêtement calculer le nombre d'erreur : c'est le **Prequential criterion**.

Comment réaliser l'apprentissage ?

Ca va être une fonction qui prend une hypothèse, un exemple, et qui ressort une nouvelle hypothèse.

$A : H \times X \times Y$

$(h_t, x_t, y_t) \rightarrow h_{t+1}$

Un scénario maximiliste : la théorie de l'apprentissage en ligne (online learning theory)

Attention : c'est une théorie pour un cas très particulier, c'est pas LA théorie.

Les chercheurs se sont posé la question de si on pouvait dire quelque chose même si on n'a aucune idée des changements qui vont arriver.

Est-ce que je peux encore dire quelque chose contre toutes séquences, y compris les séquences adversarial ? On ne peut pas compter sur la structure interne de la séquence. C'est comme si on avait un adversaire qui manipulait les données, les concepts etc.

Notion de "comités d'experts". On a un ensemble de N experts. Un expert, c'est quelqu'un qui, quand je donne x, répond y. Je n'ai aucune théorie sur la manière dont fonctionnent ces experts.

Soient 6 experts. On fait un méta contrôleur qui a n experts, il va utiliser les prédictions de ces experts. Comme dans le boosting. 0 = l'expert a eu raison, 1 = l'expert s'est trompé.

Exemple	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6
ex1	1	0	0	1	0	1
ex2	?	?	?	?	?	?

On sait lesquels des experts ont eu bon et lesquels non sur l'ex1. Qu'allons-nous faire pour l'exemple 2 ?

Bandit learning : compromis exploration-exploitation. On essaie d'optimiser le regret.

Si j'avais su à l'avance le meilleur expert pour une séquence donnée, à combien je serais de ce meilleur expert ?

Bandit $\neq$ notre cas : Dans le cas des bandits, on tire un bras, on a notre récompense. Là, on tire tous les bras (tous les experts). On connaît la réponse pour tous et comparer à la vraie réponse.

Algorithme glouton déterministe

On peut celui qui s'est pas trompé de gauche à droite en partant de celui qu'on avait choisi juste avant. Donc le 2

Exemple	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6
ex1	1	0*	0	1	0	1
ex2	?	?	?	?	?	?

Étape suivante :

Exemple	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Expert 6
ex1	1	0*	0	1	0	1
ex2	1	1	1	0	0*	0
ex3	1	0*	0	0	1	1

Avec cet algorithme, quel est dans le pire cas le nb d'erreurs que je vais faire ? Est-ce que je peux le borner par rapport au nombre d'erreurs que ferait le nb d'erreurs que fait les experts après coup ?

Le pire cas : je fais des erreurs partout. C'est pas bon. L'adversaire qui sabote sait quel expert on va choisir à l'étape suivante et peut s'arranger pour qu'il se foire.

Glouton non déterministe

On cherche l'ensemble des experts ayant fait le moins d'erreurs jusque là et on en prend un au hasard. Alors on a dans le pire cas :

$$L_{RG} \leq (ln(N) + 1)(L*)$$

Avec L* le meilleur.

On va donc faire un algo de vote

Algorithme de vote

N experts, initialement poids de 1 pour chaque expert.

W = Somme des experts a l'instant 0 = N

"Cas réalisable" = un expert aura 0 erreur sur la séquence.

Dans le cas réalisable :

Si je fais une erreur à l'instant t, cad si la majorité des experts ont tort, $W_{t+1} < W_t$. On élimine au moins la moitié des experts.

$$L_{CR} \leq \lfloor log_2(N) \rfloor$$

Dans le cas non réalisable :

On n'est pas sûr qu'il y a un expert qui fasse 0 erreur. On va donc pondérer au lieu d'éliminer ; on diminue le poids de ceux qui ont fait une erreur.

Pour chaque expert i,

$$w_i(t+1) = \begin{cases} w_i(t) \\ \beta w_i(t), \beta \in [0, 1[\end{cases}$$

Supposons qu'une erreur soit commise à l'instant t,

$$W_t \leq \frac{W(t)}{2} + \beta \frac{W(t)}{2}$$

Après m erreur :

$$W_m \leq W_0 \frac{(1+\beta)^m}{2^m}$$

Poids du meilleur expert : β^{m^*}

$$W_m \geq \beta^{m^*}$$

$$\beta^{m*} \leq W_0 \frac{(1+\beta)^m}{2^m}$$

$$m \leq \frac{log_2 N + m^* log_2(\frac{1}{\beta})}{log_2 \frac{2}{1+\beta}}$$

On a donc des bornes sur le comportement de l'algorithme par rapport au meilleur expert a posteriori.

Approches heuristiques
dérive de concept

Dérive : Continue, ruptures rares.

Comment faire pour que l'algorithme s'adapte à un environnement changeant ?

Deux défis :
- la *détection de dérive*,
- le compromis *plasticité / stabilité*. Stabilité : j'ai intérêt à avoir un maximum de données. Mais si le concept change, une grande partie de ces données sont pas bonnes : il faut savoir oublier pour s'adapter. On doit à la fois pas oublier et oublier si l'environnement change ; il faut faire un compromis.

Plasticité / stabilité

Supposons qu'on veuille régler l'oubli. Deux approches :
- soit on joue sur les exemples par fenêtre glissante (on garde une fenêtre sur les exemples passés, par exemple on garde les 1000 derniers, ou on peut adapter la taille de la fenêtre en fonction de la dérive) ou par pondération, ou un troisième TODO aller voir slide 45.
- soit on diminue le poids des experts dont la performance est mauvaise.

4.1.1 Détection de dérive

Problème : dans un environnement, on a du bruit. C'est pas parce qu'on se trompe un peu à un moment qu'il y a un changement. Comment bien détecter qu'on est dans un vrai changement ?

Il existe 3 grandes techniques de détection :
- ADWIN : on regarde le nombre moyen d'erreur une fenêtre et à partir d'un certain seuil, on a du changement
- DDM : Pareil avec le nombre d'erreur au lieu de la moyenne
- EDDM : On surveille la distance entre les erreurs.

Le DACC system

Papier par Richard Sutton et David Silver.

Ils disent que même quand l'environnement est stationnaire, on peut avoir intérêt à faire comme s'il ne l'était pas. Normalement on apprend une fonction globale au début (par exemple 50,000 parties d'un jeu), soit on apprend dans chaque contexte et ça marche mieux (5,000 parties dans un contexte, puis 5,000 dans un autre).

Ca marche parce que dans la vie, c'est pas iid. On peut avoir une fonction moins précise mais adaptée localement

4.2 Questions

Qu'est-ce qu'une distance d'édition ?

= distance de Levenshtein. C'est une distance mathématique donnant une mesure de la similarité entre deux chaînes de caractères. Elle est égale au nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre.

Qu'est-ce que l'apprentissage en ligne ?

il se fait sur des données qui arrivent de manière successive.

Qu'est-ce que le cadre Anytime ?

on doit pouvoir donner n'importe quand une réponse. On calcule tout le temps une hypothèse pour qu'elle soit prête le jour où on nous demande une réponse.

Dérive sur l'environnement (covariate shift) ?

La distribution des exemples $P_{XY} = P_{Y|X}P_X$. Le changement peut opérer sur P_X : Covariate shift. le concept ne change pas mais l'environnement change. Par exemple, distribution des patients pour un médecin en hiver ou en été, mais pas de changement des symptomes. : quand mon concept cible ne bouge pas mais en moyenne je ne rencontre pas les mêmes événements.

(Environnement non-stationnaire)

Dérive sur le concept (concept drift) ?

Si le changement opère sur $P_Y|X$: dérive de concept (concept drift), le concept cible bouge. Pour l'exemple du médecin, ça serait les symptomes qui ne sont plus associées aux mêmes maladies.

Domain adaptation ?

Par exemple, on apprend à reconnaître des images sous un certain type d'éclairage, et on nous demande d'être capable de reconnaître ces objets avec une luminosité différents. Plutôt lié au covariate shift. L'objectif est d'effectuer une tâche d'adaptation d'un système d'apprentissage d'un domaine (une distribution de probabilité) source vers un domaine (une distribution de probabilité) cible.

Apprentissage par transfert ?

Notion plus forte : j'apprends à jouer aux dames et on me demande de jouer aux échecs. Vrai changement de tâche. vise à transférer des connaissances d'une ou plusieurs tâches sources vers une ou plusieurs tâches cibles. Il peut être vu comme la capacité d'un système à reconnaître et appliquer des connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes.

Incremental learning ?

En général, c'est en environnement stationnaire, mais on se demande comment on peut apprendre au mieux un concept compliqué. On ne peut pas l'apprendre d'un coup. Quel est le meilleur ordre des exemples (curriculum), la meilleure séquence pédagogique pour que le système apprenne au mieux ?

Curriculum learning ?

When training ML models, the idea of starting with easier subtasks and gradually increase the difficulty level of the tasks, is called curriculum learning (CL). The motivation comes from the observation that humans and animals seem to learn better when trained with a curriculum or strategy.

Batch learning ?

La méthode classique d'apprentissage supervisé passif est appelé *apprentissage batch*. C'est la technique la plus utilisée aujourd'hui.

Prequential criterion ?

On ne peut plus faire comme avant : on n'a pas d'environnement stationnaire. Avant, on se disait que si on minimise critere inductif, ça va donner une performance. Or, maintenant on ne sait pas comment ça va se passer dans le futur. On ne peut plus utiliser le risque empirique. On va donc betement calculer le nombre d'erreur : c'est le Prequential criterion

Bandit learning ?

Bandit learning : compromis exploration-exploitation. On essaie d'optimiser le regret.

Le compromis plasticité / stabilité ?

- **Stabilité :** Intérêt à considérer un maximum de données.
- **Plasticité :** Pouvoir s'adapter si le concept change, savoir oublier.

Apprendre le modèle le plus précis possible (Longue mémoire), Être réactif mais en résistant au bruit (Oublier rapidement) J'ai intérêt à avoir un maximum de données. Mais si le concept change, une grande partie de ces données sont pas bonnes : il faut savoir oublier pour s'adapter. On doit à la fois pas oublier et oublier si l'environnement change ; il faut faire un compromis.

Le regret ?

Minimizing (or, alternatively, optimizing for) "regret" is simply reducing the number of actions taken which, in hindsight, it is apparent that there was a better choice.

5 Quizz 5

5.1 Notes générales

Very fast decision tree : des arbres de décision en ligne et regarde estce que les exemples qu'il a justifient de changer l'arbre ?

L'apprentissage par transfert et adaptatin de domaine

J'apprends sur une tache et j'en profite pour résoudre une autre tache. Pour l'apprentissage par transfert, c'est plus le même domaine de définition contrairement à l'adaptation de domaine où on change un peu la tâche mais elle est définie sur le même ensemble de départ. On va considérer que c'est le même problème pour la suite.

Notations

Domaine source : S Domaine cible : T

On notera D_S (resp D_T pour la cible) la distribution des exemples dans la source.

H_S l'hyp. apprise sur la source, resp. H_T

S_S : échantillon de données sources
S_T : échantillon de données cibles
Souvent, on suppose $|S_T| << |S_C|$

Scénario d'apprentissage

On a des données source et des données cibles non étiquetées. (c'est deux tourbillons avec les cibles translatées)

Comment avoir une théorie qui permet de dire j'ai raison de prendre la transformation qui coûte le moins ?

Stephane mallat il est dar.

Sur quoi établit on des bornes dans le game classique ? Le PAC learning, etc. sur quoi c'est basé ? SUr le fait que les échantillons sont iid et le monde est stationnaire.

Comment caractériser des changements de distributions ? Divergence de Kullback-Leibler.

On peut avoir de l'apprentissage par transfert avec uniquement l'hyothèse apprise sans les données sources. On n'a donc plus de différence entre les données sources et les données cibles. On n'a même pas de théorie comme en apprentissage en ligne.

Illustration

Pour le réseau neuronal : On va garder les premières couches du réseau et on va adapter les dernières. On suppose que les premières couches sont correctes. C'est ce qu'on fait pour les tâches classiques (genre analyse d'image).

Exemples : images bonne qualité vs mauvaise, changement des conditions d'éclairage, pour du text on apprend sur un type de journal et on transfert à un autres, amazon reconnaissance de sentiments sur des livres dans des commentaires et transfert aux commentaires des cuisinières

Différent du réseau de neurones profond : on change les premières couches et on garde les dernières.

L'analogie

Qu'est-ce qu'on peut faire dans ce cas là ?

Douglas Hofstadter : Godel Escher Bach livre.

Exemple analogie :

a b c -¿ a b d i j k -¿ ?

Ces problèmes suffisent à traiter tous les pb profonds de l'analogie.

Avant lui, comment on rendait compte de l'analogie ? Ex : atome et système solaire. Avant, les gens supposaient qu'il y avait deux représentations données et cherchaient en généralisant et supposaient que la représentation était déjà donnée. Quel est le chemin minimal pour passer de l'un à l'autre ? Mais c'est pas vrai, selon Hof, ce qui est fondamental c'est la perception. Il faut construire la bonne perception.

Pourquoi certaines réponses paraissent meilleures qe d'autres, par ex i j l ? Comment faire un système qui résoud ça ? COPYCAT. Qu'est-ce qu'on va faire qu'on fait un transfert plutôt qu'un autre ? On peut vouloir les plus simples, mais comment définir la simplicité ? Avec la complexité de Kolmogorov sur une machine de Turing.

Avant Kolmo, on savait caractériser la complexité d'un ensemble d'objet avec l'entropie de tous ces ojets alors que Kolmo sait faire avec un seul objet.

Dans l'analogie, on va avoir la source avec les dictionnaire et on va faire en sorte que la complexité de kolmo de tout soit minimale.

Un exemple de système de transfert

Classification précoce de série temporelles.

On va utiliser des continuateurs faibles. Même principe que pour les jeux avec les arbres. On va apprendre des fonctions de $\pi : X_t \to X_S$ (d'une partie des mesures vers toutes les mesures possibles). On apprend des fonctions π_i par le boosting. A la fin, étant donné une séquence mesurée jusquà l'instant t $H(x_t) = sigmo\{\sum_{i=1}^{N} \alpha_i h_s o \pi_i etc flemme decrire\}$

Comment avoir des garanties ?

5.2 Questions

Apprentissage par transfert ?

J'apprends sur une tache et j'en profite pour résoudre une autre tache. Ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel domains/tasks

Adaptation de domaine ?

Pour l'apprentissage par transfert, c'est plus le même domaine de définition contrairement à l'adaptation de domaine où on change un peu la tâche mais elle est définie sur le même ensemble de départ.

Comment caractériser des changements de distributions ?

Divergence de Kullback-Leibler. Qu'est-ce que c'est ?

Comment on adapte un réseau neuronal pour l'apprentissage par transfert ?

On va garder les premières couches du réseau et on va adapter les dernières. On suppose que les premières couches sont correctes. C'est ce qu'on fait pour les tâches classiques (genre analyse d'image).

Quelles sont les grandes conférences en ML ?

NIPS, CVPR, ICML, ECML, IJCAI, AAAI, ICLR

Exemples d'apprentissage par transfert ?

Exemples : images bonne qualité vs mauvaise, changement des conditions d'éclairage, pour du text on apprend sur un type de journal et on transfert à un autres, amazon reconnaissance de sentiments sur des livres dans des commentaires et transfert aux commentaires des cuisinières

Pourquoi l'apprentissage par transfert ?

Hofstadter et analogie ?

Avant, les gens supposaient qu'il y avait deux représentations données et cherchaient en généralisant et supposaient que la représentation était déjà donnée. Quel est le chemin minimal pour passer de l'un à l'autre ? Mais c'est pas vrai, selon Hofstadter, ce qui est fondamental c'est la perception.

Complexité de Kolmogorov d'un objet

La taille du plus petit programme capable de produire cet objet.

Kolmogorov vs shannon ?

Avant Kolmo, on savait caractériser la complexité d'un *ensemble* d'objet avec l'entropie de tous ces ojets alors que Kolmo sait faire avec un seul objet.

Continuateur faible ?

Quelles sont les 3 grandes approches ?

Slide 87

- Repondération des données : Repondérer les données sources. Les instances "proches" des instances cibles sont privilégiées.
- Auto-étiquetage : On étiquette X_t à l'aide d'un modèle entrainé sur X_s, puis on entraîne un modèle sur X_t nouvellement étiqueté
- Recherche d'un espace de représentation (projection) commun (dans lequel source et cible sont proches).

Donnez des idées sur la manière d'évaluer la distance entre une hypothèse source et une hypothèse cible

Dans la partie conclusion (slide 51) il est dit qu'un transfert est facile (donc au pifomètre qu'une hypothèse source est proche d'une hypothèse test) lorsque l'ensemble des projections pour passer de l'une à l'autre est de faible capacité.

J'aurais du coup tendance à répondre que pour mesurer une distance entre deux hypothèses, on peut s'intéresser au cardinal des projections faibles nécessaires pour obtenir un résultat similaire entre Hs(Ss) et Ht(St) (a epsilon arbitraire près).

De plus en voyant la définition de H_T en fonction de h_S, je vois mal comment on pourrait jouer sur autre chose que les projections.

Dans l'algorithme transboost, il y a une autre étape qui peut être intéressante (mais je doute de la robustesse de ce raisonnement), c'est la recherche d'une projection faible à une étape $i \in [|1, N|]$. Si on cherche par monte carlo sur un espace de projection prédéfini, on peut avoir deux nouveaux critères :

— La proportion des projections faibles tirées qui vérifient le critère $\epsilon_n < 0.5$, ie qui permettent un meilleur score qu'une stratégie random
— La taille de l'ensemble des projections à considérer

Quels sont les critères / aspects qui selon vous jouent un rôle dans le succès d'un apprentissage par transfert, et dites pourquoi

La similarité des domaines, le fait d'avoir suffisamment de données sources,

There must exist an ideal joint hypothesis with small error. there must exist a very good hypothesis on the target and the best hypothesis on source must be close to the best on target w.r.t to DT `https://epat2014.sciencesconf.org/conference/epat2014/pages/slides_DA_epat_17.pdf`

La qualité d'un apprentissage par transfert
— ne dépend pas de la qualité de l'hypothèse source
— mais dépend de **l'ensemble des projections**
Idée : On veux construire un espace de représentation/projection commun aux deux domaines. L'apprentissage par transfert est de qualité si l'espace de projection dans lequel les domaines sont similaires permet de garder de bonnes performances sur la tâche d'étiquetage du domaine source.

D'autre part, si l'ensemble des données cibles est trop petit (devant la taille de l'ensemble des projections), on a un gros risque d'overfitting qui nuira à la qualité du transfert.

Par rapport à l'algorithme transboost

— $\alpha_i > 0$ ssi $\epsilon_i < 0.5$
— Projection faible : $\pi_i : X_t \to X_s$ telle que l'erreur $\epsilon_n = P[hs(\pi_n(x_i)) \neq y_i] < 0.5$, c'est-à-dire l'hypothèse source est meilleure que l'aléatoire avec la projection π_n.