État de l'art sur les méthodes statistiques d'apprentissage actif

Alexis Bondu*, Vincent Lemaire*

*France Telecom R&D
2 avenue Pierre Marzin 22300 Lannion, France
(alexis.bondu,vincent.lemaire)@orange-ftgroup.com

Résumé. L'apprentissage statistique désigne un vaste ensemble de méthodes et d'algorithmes qui permettent à un modèle d'apprendre un comportement grâce à des exemples. L'apprentissage actif regroupe un ensemble de méthodes de sélection d'exemples utilisées pour construire l'ensemble d'apprentissage du modèle de manière itérative, en intéraction avec un expert humain. Toutes les stratégies ont en commun de chercher à utiliser le moins d'exemples possible et de selectionner les exemples les plus informatifs. Après avoir formalisé le problème de l'apprentissage actif et après l'avoir situé par rapport aux autres modes d'apprentissage existant dans la littérature, cet article synthétise les principales approches d'apprentissage actif et les illustre grâce à des exemples simples.

1 Introduction

En 1964, Freinet écrit dans ses invariants pédagogiques : *"La voie normale de l'acquisition n'est nullement l'observation, l'explication et la démonstration, processus essentiel de l'École, mais le tâtonnement expérimental, démarche naturelle et universelle"*(Freinet, 1964). Au début du XXè siècle, le pédagogue suisse Adolphe Ferrière (Ferrière, 1922) a été l'un des premiers à employer le terme "d'école active". L'expression "apprentissage actif" désigne en premier lieu une méthode d'enseignement permettant d'améliorer l'apprentissage des élèves en leur donnant un rôle actif.

L'apprentissage actif est une approche qui implique les élèves en les mettant en situation de progresser et en favorisant leurs interactions avec le groupe. Cette méthode d'enseignement amène les élèves à construire leurs propres connaissances en se basant sur les expériences qu'ils ont vécues. Le rôle du professeur est de choisir judicieusement les mises en situation pour atteindre l'objectif pédagogique le plus rapidement possible.

Les méthodes d'apprentissage actif en informatique sont nées d'un parallèle entre la pédagogie active et la théorie de l'apprentissage. L'apprenant est désormais un modèle (statistique) et non plus un élève. Les intéractions de l'étudiant avec son professeur correspondent à la possibilité pour le modèle d'interagir avec un expert humain (aussi appellé "oracle"). Les exemples d'apprentissage sont autant de situations utilisées par le modèle pour générer de la connaissance.

Les méthodes d'apprentissage actif permettent au modèle d'intéragir avec son environnement en sélectionnant les situations les plus "informatives". Le but est d'entraîner un modèle

en utilisant le moins d'exemples possible. La construction de l'ensemble d'apprentissage est réalisée en interaction avec un expert humain de manière à maximiser les progrès du modèle. Le modèle doit être capable de détecter les exemples les plus utiles pour son apprentissage et de demander à l'oracle : *"Que faut-il faire dans ces situations ?"*.

Cet article de synthèse a pour but de présenter les principales approches d'apprentissage actif recensées dans la littérature et de les illustrer grâce à des exemples simples. Les différentes stratégies d'apprentissage actif sont traitées de manière générique, c'est-à-dire indépendament du modèle d'apprentissage envisagé (celui qui apprend à l'aide des exemples délivrés par l'oracle suite à ses demandes). Il existe d'autres approches qui sont, soit des spécifications de ces algorithmes génériques, soit des méthodes très dépendantes du modèle. Elle ne sont pas traitées dans cet article.

La première section de l'article introduit le sujet et formalise l'apprentissage actif de manière générique. Les notations utilisées dans la suite du document sont également détaillées dans cette partie. Le but de cette section est de situer l'apprentissage actif par rapport aux autres méthodes d'apprentissage statistique présentes dans la littérature. La deuxième section est dédiée aux quatre principales approches d'apprentissage actif. Enfin, la dernière section est une discussion sur les questions soulevées par cet état de l'art.

2 Apprentissage Actif

2.1 Généralité

L'apprentissage statistique (non supervisé, semi-supervisé, supervisé[1]... etc) a pour but d'inculquer un comportement à un modèle en se basant sur des observations et sur un algorithme d'apprentissage. Les "observations" sont des instanciations du problème à résoudre et constituent les données d'apprentissage. A l'issue de son entraînement, on espère que le modèle se comportera correctement face à de nouvelles situations, on parle de capacité de généralisation.

Imaginons un modèle de classification binaire qui cherche à distinguer les personnes "heureuses" et "tristes" à partir de leur photo d'identité (voir figure 1). Si le modèle parvient à faire de bonnes prédictions pour des individus qu'il n'a pas vu lors de son entraînement, alors le modèle généralise correctement ce qu'il a appris à de nouveaux cas.

FIG. 1 – *Exemple illustratif*

[1]On ne parle pas ici de l'apprentissage par renforcement, le lecteur interessé pourra se repporter à (Harmon, 1996)

La nature des données utilisées varie selon le mode d'apprentissage (voir figure 2). L'apprentissage non supervisé utilise des données démunies d'étiquette. Dans ces conditions, le modèle ne reçoit aucune information lui indiquant quelles devraient être ses sorties ou même si celles-ci sont correctes. L'apprenant doit donc découvrir par lui-même les corrélations existant entre les exemples d'apprentissage qu'il observe. Dans le cadre de l'exemple illustratif évoqué précédement, le modèle effectue son apprentissage en se basant sur des photos d'identité démunies d'étiquette et n'a aucune indication sur ce qu'on cherche à lui faire apprendre. Parmi les méthodes d'apprentissage non supervisée on peut citer les méthodes de "clustering" (Jain et al., 1999) et les méthodes d'extraction de règles d'association (Jamy et al., 2005).

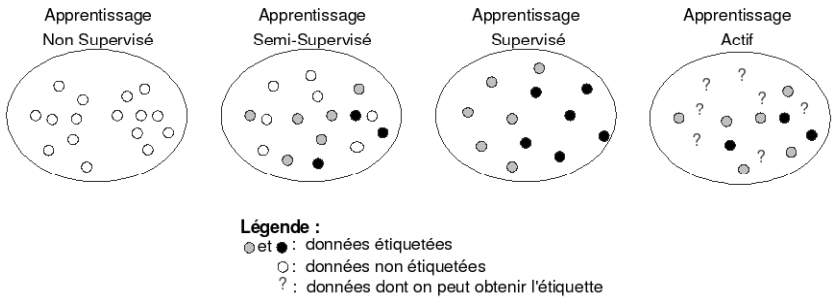

FIG. 2 – *Quelles données pour quel type d'apprentissage ?*

Le mode d'apprentissage semi-supervisé manipule conjointement des données étiquetées et non étiquetées. Parmi les utilisations possibles de ce mode d'apprentissage, on peut distinguer le "clustering" semi-supervisé et la classification semi-supervisée :

- le "clustering" semi-supervisé cherche à regrouper entre elles les instances les plus similaires en utilisant l'information apportée par les données étiquetées. A l'issue de l'apprentissage, deux instances qui ont des étiquettes différentes n'appartiennent pas au même groupe. Il existe des approches de "clustering" semi-supervisé considérant des informations suplémentaires (Cohn et al., 2003). Ces informations permettent notamment spécifier dans l'ensemble d'apprentissage (indépendament des étiquettes) si deux instances doivent ou non appartenir au même groupe.
- la classification semi-supervisée (Chapelle et Zien, 2005) se base dans un premier temps sur les données étiquetées pour séparer les instances en fonction de leurs étiquettes. Ensuite, les données non étiquetées sont utilisées pour affiner le modèle prédictif. On peut par exemple utiliser les données non étiquetées pour estimer les densités de probabilité de chacune des classes (Chappelle, 2005).

Dans le cadre de l'exemple illustratif, les exemples d'apprentissage pour le mode semi-supervisé seraient un mélange de photos démunies d'étiquette et de photos associées à une étiquette.

Lors d'un apprentissage supervisé, le modèle s'entraîne sur des données étiquetées. Ces exemples d'apprentissage sont autant d'instanciations du problème à résoudre pour lesquelles le modèle connaît la réponse attendue. Un algorithme d'apprentissage est utilisé pour régler les paramètres du modèle en se basant sur l'ensemble d'apprentissage. Dans le cadre de l'exemple illustratif évoqué ci-dessus, les exemples d'apprentissage pour le mode supervisé seraient des photos d'identité associées à une étiquette ayant pour valeur "heureux" ou "triste".

Enfin, l'apprentissage actif permet au modèle de construire son ensemble d'apprentissage au cours de son entraînement, en interaction avec un expert (humain). L'apprentissage débute avec peu de données étiquetées. Ensuite, le modèle sélectionne les exemples (non étiquetés) qu'il juge les plus "instructifs" et interroge l'expert à propos de leurs étiquettes. Dans notre exemple illustratif, le modèle présente des photos à l'oracle pour obtenir les étiquettes associées. Les stratégies d'apprentissage actif et les exemples jouets présentés dans cet article sont utilisés dans le cadre de la classification. Il est évident que ces approches peuvent être transposée au cas de la régression.

La particularité de l'apprentissage actif réside dans l'interaction du modèle avec son environnement. Contrairement à la stratégie "passive" où les exemples sont choisis avant l'apprentissage, de manière aléatoire, les stratégies "actives" permettent d'accélérer l'apprentissage en considérant d'abord les exemples les plus informatifs. Cette approche est particulièrement avantageuse lorsque les données sont coûteuses à acquérir et à étiqueter.

2.2 Deux sénarii possibles

L'apprentissage actif a pour but de détecter les exemples les plus *"instructifs"* pour les étiqueter, puis de les incorporer à l'ensemble d'apprentissage. Rui Castro (Castro et Nowak, 2005) distingue deux scénarii possibles : l'échantillonnage adaptatif et l'échantillonnage sélectif (voir figure 3). Il s'agit de deux manières différentes de poser le problème de l'apprentissage actif.

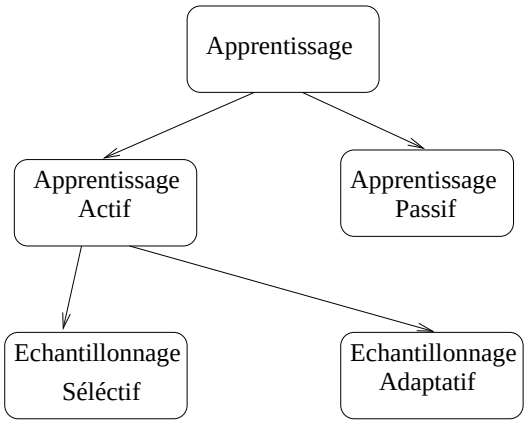

FIG. 3 – *Apprentissage : vue générale*

Il est important de distinguer les données brutes et les descripteurs qui leur sont associés (voir figure 4). Dans notre exemple illustratif, les données brutes sont les photos d'identité à proprement parlé et les descripteurs sont des attributs décrivant les photos (pixel, luminosité, contraste... etc). Le modèle fait correspondre la prédiction de la classe "heureux" ou "triste" à chaque vecteur de descripteurs placé en entrée. Il est important de noter que le calcul des descripteurs à partir des données brutes n'est pas forcement "bijectif". Il est parfois impossible de reconstituer une donnée brute en se basant sur des descripteurs.

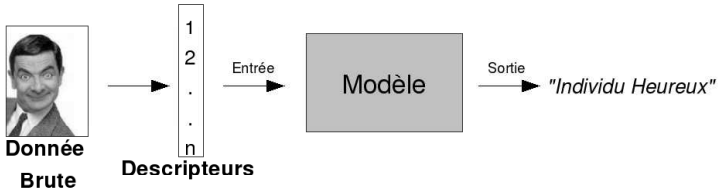

FIG. 4 – *Chaîne de traitement d'un modèle prédictif*

Dans le cas de **l'échantillonnage adaptatif** (Singh et al., 2006), le modèle demande à l'oracle des étiquettes correspondant à des vecteurs de descripteurs. Le modèle n'est pas restreint et peut explorer tout l'espace de variation des descripteurs, à la recherche de zones à échantillonner plus finement. Dans certains cas l'échantillonnage adaptatif peut poser problème lors de sa mise en oeuvre. En effet, il est difficile de savoir si les vecteurs de descripteurs générés par le modèle ont toujours un sens vis à vis du problème initial. Pour illustrer ceci, on peut se reporter à la figure 4. Supposons que le modèle demande l'étiquette associée au vecteur $[10, 4, 5.., 12]$, on ne sait pas si cet ensemble de descripteurs correspond à une photo de visage humain ou bien à une fleur ou encore à un animal.

Dans le cas de **l'échantillonnage sélectif** (Roy et McCallum, 2001), le modèle n'observe qu'une partie restreinte de l'univers matérialisée par des exemples d'apprentissage démunis d'étiquette. Par conséquent, les vecteurs d'entrées sélectionnés par le modèle correspondent toujours à une donnée brute. On emploie généralement l'image d'un *"sac"* d'instances pour lesquelles le modèle peut demander les labels associés. Dans l'exemple illustratif précédent, le modèle demande l'étiquette associée au vecteur $[10, 4, 5.., 12]$ qui correspond à une photo d'identité dont on dispose. L'oracle aura beaucoup plus de facilité à étiqueter une photo qu'un ensemble de descripteurs.

Dans la suite de cet article, on se place du point de vue de l'échantillonnage sélectif. Cet état de l'art s'intéresse aux problèmes d'apprentissage pour lesquels il est facile d'obtenir un grand nombre d'instances non étiquetées et pour lesquels l'étiquetage est coûteux. Dans la pratique, le choix de l'échantillonnage sélectif ou adaptatif dépend essentiellement du domaine d'application. Selon les cas le modèle est autorisé (ou non) à "générer" de nouvelles instances.

2.3 Notations

Cette section définit l'ensemble des notations utilisées dans cet article, ces notations sont également illustrées grâce à un problème jouet.

Soit $\mathcal{M} \in \mathbb{M}$ le modèle prédictif dont on cherche à faire l'apprentissage grâce à un algorithme $\mathcal{L}$. La figure 5 représente les différents ensembles mis en jeux. L'ensemble $\mathbb{X} \subseteq \mathbb{R}^n$ représente toutes les entrées possibles du modèle et $x \in \mathbb{X}$ en est une instance particulière. On définit également $\mathbb{Y}$ l'ensemble des réponses potentielles du modèle. Soit $y \in \mathbb{Y}$ un label[2] particulier associé à une entrée $x \in \mathbb{X}$.

[2]On entend par label, soit une valeur discrète dans un problème de classification, soit une valeur continue dans un problème de régression

Lors de son appentissage, le modèle n'observe qu'une partie $\Phi \subseteq \mathbb{X}$ de l'univers. On dispose d'un ensemble fini de points d'observation et on ne connaît pas nécessairement les labels associés à ces points. Soient U_x la partie des instances observables pour lesquelles on ne connaît pas les labels et L_x la partie des instances observables pour lesquels on dispose des labels associés. On a : $\Phi \equiv U_x \cup L_x$ et $U_x \cap L_x \equiv \emptyset$.

Le concept cible que l'on veut apprendre au modèle peut être vu comme une fonction $f : \mathbb{X} \rightarrow \mathbb{Y}$, avec $f(x_1)$ la réponse arrendue du modèle pour l'entrée x_1. On définit également $\widehat{f} : \mathbb{X} \rightarrow \mathbb{Y}$ la réponse effective du modèle, il s'agit d'une estimation du concept cible. Les éléments de L_x et les labels qui leur sont associés constituent un ensemble d'apprentissage T. Les exemples d'appentissage sont des couples $(x, f(x)) : \forall x \in L_x, \exists!(x, f(x)) \in T$.

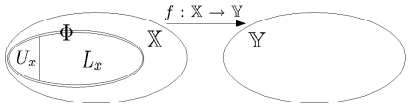

FIG. 5 – *Ensembles mise en jeux*

Illustration des notations :
Pour bien comprendre ces notations, considérons le problème jouet suivant (Guestrin et al., 2005) (voir figure 6). On dispose d'une pièce équipée de capteurs de température fixés au plafond. On cherche à estimer la température en tout point de la pièce, en se basant sur quelques points de mesure. Pour ce problème le vecteur d'entrée du modèle est de dimension deux ($\mathbb{R}^2$), puisqu'on considère le plan sur lequel sont fixés les capteurs. On se restreint à la pièce, l'ensemble $\mathbb{X} \subseteq \mathbb{R}^2$ est donc délimité par les murs. $\mathbb{Y} \equiv [-50C, 100C]$ est l'ensemble des températures envisageables dans une pièce.

Ici, l'ensemble des points observables $\Phi \subseteq \mathbb{X}$ correspond à l'ensemble des capteurs. Il s'agit bien d'un problème d'échantillonnage sélectif puisqu'on ne peut obtenir la température que pour l'ensemble (le "sac") des capteurs. Supposons maintenant qu'une partie de ces capteurs soit en panne. L_x est l'ensemble des capteurs en état de fonctionner et U_x est l'ensemble des capteurs hors service. Le concept qu'on cherche à apprendre peut être vu comme une fonction $f : \mathbb{X} \rightarrow \mathbb{Y}$ qui à chaque point de la pièce associe sa température. Dans le cadre de cet exemple jouet, l'apprentissage actif est une approche capable de désigner les capteurs à réparer prioritairement pour améliorer la prédiction du modèle.

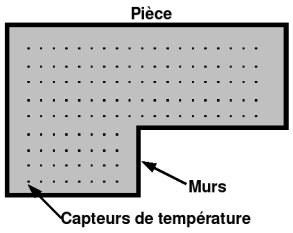

FIG. 6 – *Estimation de la température dans une pièce*

3 Méthodes d'apprentissage actif

3.1 Introduction

Le problème de l'échantillonnage sélectif a été posé formellement par Muslea (Muslea, 2002) (voir Algorithme 1). Celui-ci met en jeu une fonction d'utilité, $\mathcal{U}tile(u, \mathcal{M})$, qui estime l'intérêt d'une instance u pour l'apprentissage du modèle $\mathcal{M}$. Grâce à cette fonction, le modèle présente à l'oracle les instances pour lesquelles il espère la plus grande amélioration de ses performances.

L'Algorithme 1 est générique dans la mesure où seule la fonction $\mathcal{U}tile(u, \mathcal{M})$ doit être modifiée pour exprimer une stratégie d'apprentissage actif particulière. Comment peut-on "préjuger" efficacement de l'intérêt d'un exemple pour l'apprentissage d'un modèle, avant même que celui-ci n'ait été étiqueté ? C'est ce que nous allons voir dans la suite de cet état de l'art.

Étant donnés :

- $\mathcal{M}$ un modèle prédictif muni d'un algorithme d'apprentissage $\mathcal{L}$
- Les ensembles U_x et L_x d'exemples non étiquetés et étiquetés
- n le nombre d'exemples d'apprentissage souhaité.
- L'ensemble d'apprentissage T avec $\|T\| < n$
- La fonction $\mathcal{U}tile : \mathbb{X} \times \mathbb{M} \rightarrow \Re$ qui estime l'utilité d'une instance pour l'apprentissage d'un modèle.

Répéter
(A) Entraîner le modèle $\mathcal{M}$ grâce à $\mathcal{L}$ et T (et éventuellement U_x).
(B) Rechercher l'instance $q = argmax_{u \in U_x}\, \mathcal{U}tile(u, \mathcal{M})$
(C) Retirer q de U_x et demander l'étiquette $f(q)$ à l'oracle.
(D) Ajouter q à L_x et ajouter $(q, f(q))$ à T

Tant que $\|T\| < n$

Algorithme 1: échantillonnage sélectif, Muslea 2002

3.2 Echantillonnage par incertitude

Cette stratégie d'apprentissage actif (Lewis et Gale, 1994) (Thrun et Möller, 1992) est basée sur la confiance que le modèle a en ses prédictions. Le modèle d'apprentissage utilisé doit être capable de fournir une réponse au problème traité et d'estimer la fiabilité de ses réponses. Pour illustrer ceci, prenons l'exemple des fenêtres de Parsen (Chappelle, 2005) dont la "sortie" est une estimation de probabilité conditionnelle, définie de la manière suivante :

$$\hat{P}(y_j|u_n) = \frac{\sum_{i=1\ \ f(u_i)=y_j}^{N} K(u_n, u_i)}{\sum_{i=1}^{N} K(u_n, u_i)}$$

où

$$K(u_n, u_i) = e^{\frac{||u_n - u_i||^2}{2\sigma^2}}$$

Dans le cadre de cet exemple, on choisit d'utiliser un noyau gaussien de norme L2 et on se place dans le cas d'une classification. Finalement, on dispose des probabilités d'observer chacune des classes y_j pour une instance u_n. Cela permet au modèle de faire une prédiction en choisissant la classe la plus probable pour l'instance u_n. La probabilité d'observer la classe choisie peut être vue comme la confiance que le modèle a en sa prédiction. Le choix des nouveaux exemples à étiqueter se déroule en deux étapes :

- on utilise le modèle dont on dispose à l'itération t et on prédit les étiquettes des exemples non-étiquetés ;
- on sélectionne les exemples pour lesquels la prédiction est la plus incertaine.

L'incertitude d'une prédiction peut également être définie par rapport à un seuil de décision. Prenons l'exemple d'un réseau de neurones qui ne possède qu'un seul neurone de sortie et qui est utilisé dans le cadre d'une classification binaire. Supposons que la sortie du réseau de neurones soit une valeur continue comprise entre 0 et 1. Un seuil de décision est défini pour faire correspondre une sortie à une des deux classes. Plus la sortie du réseau de neurones est proche du seuil de décision, plus la décision sera considérée comme incertaine.

La figure 7 représente un problème de classification binaire. La séparatrice correspondant au seuil de décision du modèle est tracée ainsi que des lignes de niveaux relatifs aux valeurs de sortie du modèle. Les données non-étiquetées qui se situent à proximité de la séparatrice (au sens des lignes de niveaux) sont considérées comme étant les plus incertaines, elles seront donc choisies pour être étiquetées par l'oracle.

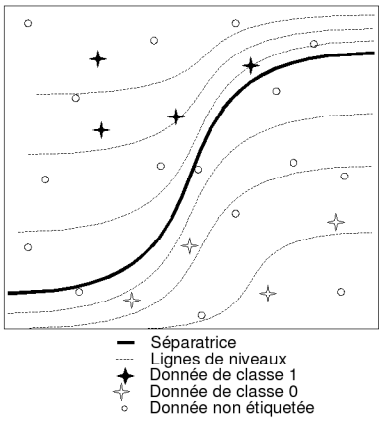

FIG. 7 – *échantillonnage par incertitude : Classification binaire*

Cette première approche a l'avantage d'être intuitive, facile à mettre en oeuvre et rapide. L'échantilonnage par incertitude montre cependant ses limites lorsque le problème à résoudre n'est pas séparable par le modèle. En effet, cette stratégie aura tendance à sélectionner les exemples à étiqueter dans des zones de mélange, où il n'y a peut être plus rien à apprendre.

3.3 Echantillonnage par comité de modèles

L'échantillonnage par comité de modèles (ou *"Query-by-Committee"*) est une stratégie qui vise à réduire l'espace des versions[3] d'un problème d'apprentissage (Dima et Hebert, 2005). Rappelons que l'espace des versions $\mathbb{S} \subseteq \mathbb{H}$ est l'ensemble des hypothèses consistantes avec les données d'apprentissage étiquetées, cet espace représente les informations contenues dans l'ensemble d'apprentissage.

Réduction de l'espace des versions :
Seung (Seung et al., 1992) est le précurseur de l'échantillonnage par comité. Cette approche génère une multitude de modèles d'apprentissage qui sont entraînés en parallèle sur les mêmes données. En considérant que chacun de ces modèles trouvent une hypothèse consistante $h \in \mathbb{S}$, on dispose alors d'un "échantillon" d'hypothèses qu'on espère représentatif de l'espace des versions. On cherche à mesurer le désaccord au sein du comité de modèles lors de la prédiction du label des exemples non étiquetés. Les exemples qui suscitent le plus grand désaccord sont ceux qui ont la plus forte probabilité de réduire $\mathbb{S}$ une fois étiquetés.

Dans la pratique, on n'a pas la garantie que le comité de modèles échantillonne correctement l'espace de versions. En effet, il peut être impossible pour les modèles de trouver des hypothèses consistantes avec les données d'apprentissage. Cela peut être simplement dû à du bruit présent dans les données. Ou encore, le problème traité peut être trop complexe par rapport au type de modèles utilisés. Pour pallier à cette difficulté, on peut utiliser des modèles "stochastiques"[4], ce qui favorise la diversité des hypothèses.

L'échantillonnage par comité peut être utilisé avec des modèles génératifs[5](McCallum et Nigam, 1998). Cela permet d'échantillonner l'espace des versions en définissant une distribution de probabilité sur les paramètres des modèles. Il s'agit d'une solution élégante mais qui n'est pas applicable dans le cas général, où on ne possède pas de distribution de probabilité sur les paramètres des modèles.

Il existe également un moyen d'utiliser des modèles dont l'apprentissage est "déterministe" dans le cadre d'un echantillonnage par comité de modèles. Abe et Mamitsuka (Abe et Mamitsuka, 1998) proposent le "Query-by-Bagging" qui permet d'entraîner les modèles sur des sous-ensembles d'exemples disjoints. Les sous-ensembles d'exemples ont une distribution de probabilité similaire à celle de la totalité des exemples. Dans ce cas, on peut utiliser des modèles "déterministes" (tels que des régressions linéaires) et obtenir des hypothèses différentes. La diversité des hypothèses est due aux sous-ensembles d'exemples utilisés pour l'apprentissage de chaque modèle.

Mesure de désacord :
Yoav Freund (Freund et al., 1997) propose une approche qui se base sur la théorie de l'information et estime le désaccord du comité de modèles par une mesure d'entropie sur les prédictions. Dans le cadre de cette approche le gain d'information est un bon estimateur de la réduction de $\mathbb{S}$ (c'est-à-dire de l'amélioration du comité de modèles après l'ajout d'un nouvel exemple). Plus précisément, considérons un exemple non étiqueté noté $u \in U_x$ qui peut potentiellement être

[3] Pour plus de détails sur l'espace des versions, se reporter à la section 3.4.

[4] on entend par modèles "stochastiques" des modèles dont l'apprentissage n'est pas déterministe

[5] un modèle "génératif" est un modèle capable de générer des donnés aléatoires et de modéliser la distribution de probabilité qui régit ces données.

associé à $\|\mathbb{Y}\|$ étiquettes. On peut estimer l'entropie des prédictions des modèles $\mathcal{M}_1, ..., \mathcal{M}_m$ de la manière suivante :

$$\widehat{\mathcal{H}}(u|\mathcal{M}_1, ..., \mathcal{M}_m) = \sum_{j=1}^{\|\mathbb{Y}\|} -\widehat{P}(y_i|u, \mathcal{M}_1, ..., \mathcal{M}_m) log \widehat{P}(y_i|u, \mathcal{M}_1, ..., \mathcal{M}_m)$$

Notons que la probabilité d'observer la classe y_i conditionnellement à l'instance u est estimée grâce à l'ensemble des prédictions du comité de modèles. Cette expression estime l'entropie des étiquettes associées à une instance u, sur l'espace de versions.

Dans la littérature, il existe d'autres métriques pouvant quantifier le désaccord au sein d'un comité de modèles. Dans le cas d'une classification binaire, on peut se baser sur le nombre de modèles qui prédisent l'une ou l'autre des classes (Abe et Mamitsuka, 1998). Dans ce cas, le score représentant le désaccord peut être donné par l'effectif des modèles qui prédisent la classe minoritaire. Cela peut s'écrire de la manière suivante (avec $\widehat{f}(u)$ la prédiction du comité de modèle et $\widehat{f}_{\mathcal{M}_k}(u)$ la prédiction du modèle $\mathcal{M}_k$) :

$$\mathcal{D}esaccord(u) = \sum_{k=1}^{m} \mathbb{1}_{\{\widehat{f}_{\mathcal{M}_k}(u) \neq \widehat{f}(u)\}}$$

La *"Kullback-Leibler-divergence-to-the-mean"* (McCallum et Nigam, 1998) est une autre métrique qui a l'avantage de prendre en compte la confiance des prédictions faites par les modèles pour mesurer leur désaccord. Cette métrique est définie comme étant la moyenne (sur les m modèles) de la "KL-divergence" entre la distribution des classes estimée par un modèle $\mathcal{M}_k$ du comité et la distribution moyenne estimée grâce à la totalité du comité. Cela peut s'écrire de la manière suivante :

$$\mathcal{D}esaccord(u) = \frac{1}{m} \sum_{k=1}^{m} \mathcal{D}iv(P(\mathbb{Y}|u, \mathcal{M}_k) \| \overline{P}(\mathbb{Y}|u, \mathcal{M}_1 ... \mathcal{M}_m))$$

$$\mathcal{D}esaccord(u) = \frac{1}{m} \sum_{k=1}^{m} \sum_{j=1}^{\|\mathbb{Y}\|} P(y_j|u, \mathcal{M}_k) Log \frac{P(y_j|u, \mathcal{M}_k)}{\overline{P}(y_j|u, \mathcal{M}_1 ... \mathcal{M}_m)}$$

3.4 Arbre sur l'espace des versions

Contrairement à l'échantillonnage par comité (voir section 3.3), certains modèles d'apprentissage permettent de manipuler directement l'espace des versions (sans avoir recourt à un comité de modèles). On citera à titre d'exemple la stratégie qui consiste à choisir les exemples à étiqueter dans la marge d'un SVM (Tong et Koller, 2000) ou celle des SG-net (Cohn et al., 1994) qui est basée sur les réseaux de neurones. Il existe de nombreuses autres approches d'apprentissage actif basées sur la réduction de l'espace des versions, notamment celles qui sont dépendantes du modèle. On ne les sitera pas toutes, le but de cette section étant de présenter cette approche d'un point de vue générique.

L'apprentissage actif peut être vu comme le parcours d'un arbre de décision construit sur l'espace des versions du modèle (voir figure 8). Le nœud principal de l'arbre correspond au

premier exemple qu'on soumet à l'oracle (partie A de la figure 8). Selon l'étiquette attribuée à cet exemple (partie B de la figure 8), on se place sur une des branches issues de ce nœud. Cela à pour effet de désigner le prochain nœud à parcourir et donc le prochain exemple à étiqueter. Dans le cadre de cette approche, toutes les décisions sont prises lors de la construction de l'arbre de décision.

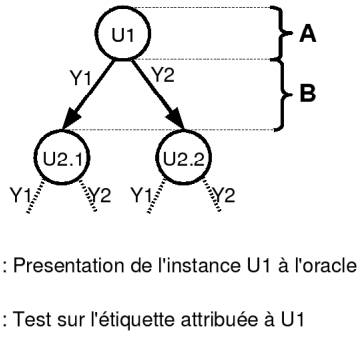

FIG. 8 – *Arbre sur l'espace des versions : cas d'un problème binaire*

Soient les ensembles d'apprentissage U_x et L_x précédemment décrits. On définit $\mathbb{H}$ l'ensemble des hypothèses que le modèle $\mathcal{M} \in \mathbb{M}$ peut atteindre. Rappelons qu'une hypothèse $\widehat{f}$ représente le concept à apprendre. L'espace des versions $\mathbb{S} \subseteq \mathbb{H}$ est l'ensemble des hypothèses consistantes avec L_x. On définit également $s \subseteq U_x$ l'ensemble des données *"non étiquetées"* classées différemment par les hypothèses de $\mathbb{S}$.

On cherche à déterminer l'hypothèse $h \in \mathbb{H}$ consistante avec toutes les données d'apprentissage[6], dont la construction requiert un minimum d'étiquetages. L'étiquetage d'une instance $x \in s$ réduit l'espace des versions. Ce nouvel espace $\mathbb{S}'$ dépend de l'étiquette qu'on attribue à l'instance x.

On définit π la distribution de probabilité des hypothèses de $\mathbb{H}$. La stratégie gloutonne proposée par Sanjoy Dasgupta (Dasgupta, 2005) consiste à demander le label de l'instance $x_i \in s$ qui conduit "potentiellement"[7] à des espaces des versions de même poids, au sens de π.

Exemple illustratif :
Pour illustrer ceci, décrivons cette stratégie dans le cas d'un problème de classification à deux classes. Pour chaque instance $x_i \in s$, on définit $S_i^+ \subseteq \mathbb{S}$ (*resp* $S_i^- \subseteq \mathbb{S}$) l'ensemble des hypothèses consistantes qui classent x_i positivement (*resp* négativement). On cherche à étiqueter l'instance qui sépare $\mathbb{S}$ en deux sous-ensembles S_i^+ et S_i^- les plus équiprobables possibles au sens de π.

Comme on peut le voir à travers l'algorithme 2, cette stratégie peut être représentée par un arbre de décision, dans lequel un nœud N_i est un "test" sur le label de l'instance x_i. Les

[6] On suppose dans cette approche qu'il existe au moins une hypothèse $h \in \mathbb{H}$ consistante avec toutes les données d'apprentissage

[7] Selon l'étiquette attribuée à x_i

branches connectées à ce nœud correspondent aux différentes valeurs possibles du label de x_i. Les feuilles de l'arbre sont des hypothèses $h \in \mathbb{H}$. La hauteur de l'arbre représente le nombre maximal de questions posées à l'oracle.

Étant donnés :

- $\mathcal{N} \equiv \{N_1[x_1], N_2[x_2], ..., N_n[x_n]\}$ les nœuds de l'arbre et les instances testées
- $\mathbb{S} \equiv \{\mathbb{S}_1, \mathbb{S}_2, ..., \mathbb{S}_n\}$ les espaces des versions du modèle sur les nœuds
- $s \in U_x$ les instances classées différemment par les hypothèse de $\mathbb{S}$
- $\mathbb{Y} \equiv \{y_1, y_2, ..., y_m\}$ les labels qui correspondent aux m branches connectées à un nœud.
- $\{S^1_{h,i}, .., S^m_{h,i}\}$ les sous-ensembles de $\mathbb{S}_h$ dus au test de x_i au nœud h
- $\mathcal{E}nt : \mathbb{H}^N \rightarrow \Re$ l'entropie d'un ensemble d'hypothèses
- $\mathcal{G}ain : \mathbb{H}^N \times \mathbb{X} \rightarrow \Re$ le gain d'information

$\mathcal{N} \leftarrow \{N_1\}$
$\mathbb{S} \leftarrow \{\mathbb{S}_1\}$
Répéter
  Pour chaque nœud $N_h \in \mathcal{N}$ **faire**
    Pour chaque instance candidate $x_i \in s$ **faire**
      Calculer $\mathcal{E}nt(\mathbb{S}_h)$
      Pour chaque label $y_j \in \mathbb{Y}$ **faire**
        Calculer $\mathcal{E}nt(S^j_{h,i})$
      Fin Pour
      Calculer $\mathcal{G}ain(\mathbb{S}_h, x_i)$
    Fin Pour
    Sélectionner $q = argmax_{x \in s}\, \mathcal{G}ain(\mathbb{S}_h, x)$ pour le nœud courant
    On affecte q au nœud courant $N_h \leftarrow N_h[q]$
    On retire q des instances candidates $s \leftarrow s \setminus \{q\}$
    Pour chaque label $y_j \in \mathbb{Y}$ **faire**
      création d'un nouveau nœud $\mathcal{N} \leftarrow \mathcal{N} \cup \{N_{[h \times \|\mathbb{Y}\| + j]}\}$
      création de l'espace des versions associé $\mathbb{S} \leftarrow \mathbb{S} \cup \{\mathbb{S}_{[h \times \|\mathbb{Y}\| + j]}\}$
    Fin Pour
  Fin Pour
Tant qu'il y a plusieurs hypothèses dans les feuilles de l'arbre

Algorithme 2: Construction d'un arbre sur l'espace des versions

Le problème jouet présenté par la figure 9 illustre la construction d'un arbre de décision sur l'espace des versions d'un modèle d'apprentissage. Le problème traité dans cet exemple est une classification binaire dans le plan. On suppose que le concept cible est une séparatrice linéaire contrainte de passer par un certain point. L'espace des versions du modèle est l'ensemble des hypothèses consistantes avec les données étiquetées. Sur la partie gauche de la figure 9, il s'agit de l'ensemble des droites qui passent par le point noir et qui séparent correctement les données étiquetées. Dans le cadre de cet exemple jouet on suppose que la distribution de probabilité des hypothèses est uniforme. Dans la partie centrale de la figure 9,

on trouve l'instance non étiquetée qui sépare l'espace des versions en deux sous-espaces les plus équiprobables possibles. Le test de l'étiquette de cette instance correspond à un nœud de l'arbre. La partie droite de la figure 9 montre que l'étiquetage de cette instance réduit environ de moitié l'espace des versions (au sens de π).

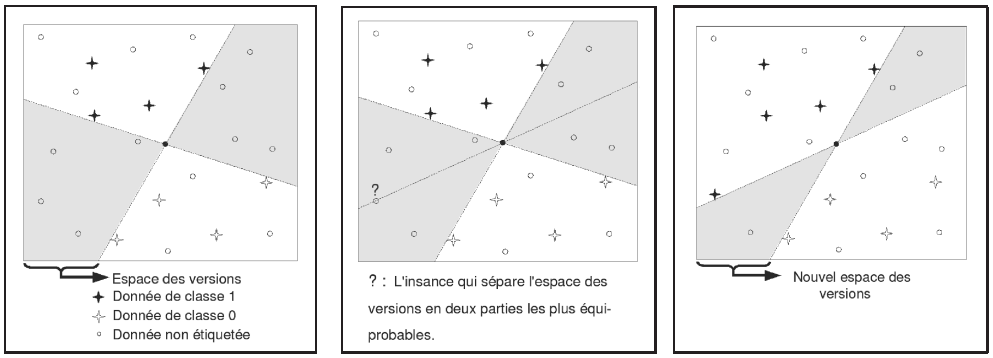

FIG. 9 – *Arbre sur l'espace des versions : problème jouet*

Lors de la construction de cet arbre de décision, on cherche les meilleures instances à tester pour chacun des nœuds. Pour cela, on mesure pour chaque $x_i \in s$ le gain d'information. Ce dernier est calculé grâce à l'entropie de l'espace des versions *"courant"* et des n sous-ensembles[8] induits par le test du label de x_i. On sélectionne l'instance dont le "test" induit le plus grand gain d'information pour la construction du nœud courant. L'entropie "$\mathcal{E}nt$" et le gain d'information "$\mathcal{G}ain$" sont définis de la manière suivante (les S_i^j étant les n sous-ensembles induits par le test de l'instance x_i) :

$$\mathcal{E}nt(\mathbb{S}) = \sum_{\mathbb{S}} -log\pi(h).\pi(h)$$

$$\mathcal{G}ain(\mathbb{S}, x_i) = \mathcal{E}nt(\mathbb{S}) - \sum_{j=1}^{n} \frac{\|S_i^j\|}{\|\mathbb{S}\|} \mathcal{E}nt(S_i^j)$$

Dans le cadre de l'apprentissage actif, la qualité Q de la stratégie utilisée peut être définie comme étant la hauteur moyenne de l'arbre $\mathcal{T}$ résultant de cette stratégie, puisqu'on cherche à poser le moins de questions possible à l'oracle. Cela s'écrit de la manière suivante (avec $\mathcal{H}eight(h)$ la hauteur de la feuille correspondant à l'hypothèse h) :

$$Q(\mathcal{T}, \pi) = \textstyle\sum_{h \in \mathbb{H}} \pi(h).\mathcal{H}eight(h)$$

De manière générale, l'efficacité d'une stratégie d'apprentissage actif dépend de la typologie des données d'apprentissage. Il y a des problèmes triviaux qui nécessitent tous les labels manquants pour trouver la bonne hypothèse. Sanjoy Dasgupta (Dasgupta, 2005) établit des bornes théoriques pour cette méthode sur le nombre de labels demandés à l'oracle.

[8] Pour la classification à deux classes, les deux sous-ensembles en question sont S_i^+ et S_i^-

Pour conclure, la construction d'un arbre de décision sur l'espace des versions est difficile à mettre en œuvre dans la pratique car on dispose rarement de l'espace des versions d'un modèle et de la distribution de probabilité des hypothèses. Néanmoins, cette approche constitue une vue théorique intéressante de l'apprentissage actif.

3.5 Réduction de l'erreur de généralisation du modèle

L'approche basée sur la réduction de l'erreur de généralisation du modèle (Cohn et al., 1995) choisit les exemples à étiqueter de manière à minimiser cette erreur notée $E(\mathcal{M}^t)$. Dans la pratique cette erreur ne peut pas être calculée. Pour cela il faudrait disposer de l'ensemble des entrées du modèle noté $\mathbb{X}$. On peut cependant exprimer l'erreur de généralisation du modèle $\mathcal{M}$ à l'instant t en utilisant une fonction de "coût" ($\mathcal{L}oss(\mathcal{M}^t, x)$) qui évalue l'erreur du modèle pour une entrée particulière $x \in \mathbb{X}$.

$$E(\mathcal{M}^t) = \int_{\mathbb{X}} \mathcal{L}oss(\mathcal{M}^t, x)P(x)dx$$

On définit $\mathcal{M}^{t+1}_{(x^\diamond, y^\diamond)}$, le même modèle à l'itération $t+1$. Ce modèle prend en compte un nouvel exemple d'apprentissage noté $(x^\diamond, y^\diamond)$. Dans la pratique, la sortie du modèle $y^\diamond$ n'est pas connue puisque $x^\diamond$ est une donnée non étiquetée. Pour estimer l'erreur de généralisation à l'itération $t+1$, il faut envisager toutes les possibilités de l'ensemble $\mathbb{Y}$ et les pondérer par leur probabilité. L'erreur de généralisation "attendue" s'écrit alors :

$$E(\mathcal{M}^{t+1}_{x^\diamond}) = \int_{\mathbb{X}} \int_{\mathbb{Y}} P(y|x^\diamond)\mathcal{L}oss(\mathcal{M}^{t+1}_{(x^\diamond,y)}, x)P(x)dxdy$$

Cette stratégie sélectionne l'instance q qui minimise $E(\mathcal{M}^{t+1}_{x^\diamond})$. Une fois étiquetée, cette instance est incorporée à l'ensemble d'apprentissage. On espère ainsi construire itérativement un ensemble d'apprentissage optimal.

Sans disposer de tous les éléments de $\mathbb{X}$, Nicholas Roy (Roy et McCallum, 2001) montre comment cette stratégie peut être mise en œuvre en utilisant uniquement les données d'apprentissage. On estime l'erreur de généralisation en considérant seulement les exemples dont on dispose à l'instant t et en adoptant un à priori uniforme pour $P(x)$:

$$\widehat{E}(\mathcal{M}^t) = \frac{1}{\|L_x\|} \sum_{i=1}^{\|L_x\|} \mathcal{L}oss(\mathcal{M}^t, x_i)$$

Pour choisir l'instance à étiqueter, le modèle est ré-entraîné plusieurs fois en considérant un exemple supplémentaire. Chaque instance $x_i \in U_x$ et chaque label $y_j \in \mathbb{Y}$ peuvent s'associer pour former cet exemple supplémentaire. Pour chaque exemple candidat x_i, on entraîne le modèle plusieurs fois en fixant la valeur de l'étiquette puis en mesurant l'erreur de généralisation $\widehat{E}(\mathcal{M}^{t+1}_{(x_i,y_j)})$. Lorsque toutes les étiquettes ont été envisagées pour une instance x_i, on estime l'erreur de généralisation attendue après l'étiquetage de cette instance notée $\widehat{E}(\mathcal{M}^{t+1}_{x_i})$. Pour se faire, on utilise un modèle capable d'estimer $P(y_j|x_i)$, les probabilités des labels y_i sachant l'instance x_i.

Après avoir traité tous les exemples candidats, il ne reste plus qu'à choisir ceux qui minimisent l'erreur de généralisation attendue, et les faire étiqueter par l'oracle. Ce procédé est répété de manière itérative pour enrichir l'ensemble d'apprentissage. Une vue synthétique de

cette approche est présentée par l'algorithme (3). Il existe autant de variantes de cette stratégie qu'on peut imaginer de fonction de coût. Considérons maintenant un cas d'utilisation de ce type d'approche à titre illustratif.

Étant donnés :

- $\mathcal{M}$ un modèle prédictif muni d'un algorithme d'apprentissage $\mathcal{L}$
- Les ensembles U_x et L_x d'exemples non étiquetés et étiquetés
- n le nombre d'exemples d'apprentissage souhaité.
- L'ensemble d'apprentissage T avec $\|T\| < n$
- $\mathbb{Y}$ l'ensemble des labels qui peuvent être attribués aux instances de U_x
- $\mathcal{L}oss : \mathbb{M} \rightarrow \Re$ l'erreur de généralisation du modèle
- $\mathcal{E}rr : U_x \times \mathbb{M} \rightarrow \Re$ l'erreur de généralisation attendue pour le modèle $\mathcal{M}$ entraîné avec une instance supplémentaire, $T \cup (x_i, f(x_i))$

Répéter
- (A) Entraîner le modèle $\mathcal{M}$ grâce à $\mathcal{L}$ et T
- **Pour** chaque instance $x_i \in U_x$ **faire**
  - **Pour** chaque label $y_j \in \mathbb{Y}$ **faire**
    - i) Entraîner le modèle $\mathcal{M}_{i,j}$ grâce à $\mathcal{L}$ et $(T \cup (x_i, y_j))$
    - ii) Calculer l'erreur de généralisation $\widehat{E}(\mathcal{M}^{t+1}_{(x_i,y_j)})$
  - **Fin Pour**
  - Calculer l'erreur de généralisation attendue $\widehat{E}(\mathcal{M}^{t+1}_{x_i}) = \sum_{y_j \in \mathbb{Y}} \widehat{E}(\mathcal{M}^{t+1}_{(x_i,y_j*)}).P(y_j|x_i)$
- **Fin Pour**
- (B) Rechercher l'instance $q = argmin_{u \in U_x} \widehat{E}(\mathcal{M}^{t+1}_{x_i})$
- (C) Retirer q de U_x et demander l'étiquette $f(q)$ à l'oracle.
- (D) Ajouter q à L_x et ajouter $(q, f(q))$ à T

Tant que $\|T\| < n$

Algorithme 3: Apprentissage actif *"optimal"*, de Nicholas Roy 2000

X. Zhu (Zhu et al., 2003) propose d'approcher l'erreur de généralisation par le risque empirique et d'utiliser une fenêtre de Parzen à noyau gaussien (Parzen, 1962) comme modèle d'apprentissage. Ici, le risque $R(\mathcal{M})$ est défini comme étant la somme des probabilités que le modèle prenne une mauvaise décision sur l'ensemble d'apprentissage. On note $P(y_i|l_n)$ la probabilité réelle d'observer la classe y_i pour l'instance $l_n \in L_x$. Le risque empirique s'écrit alors selon l'équation 1, avec $\mathbb{1}$ la fonction indicatrice égale à 1 si $f(l_n) \neq y_i$ et égale à 0 sinon.

Le modèle que l'on utilise doit être un estimateur de densité dont la sortie est la probabilité $P(y_i|l_n)$ d'observer la classe y_i conditionnellement à l'instance l_n. Sous cette condition, on peut approximer le risque empirique en adoptant un a priori uniforme sur les $P(l_n)$ (voir équation 2). Le but de cette stratégie est de sélectionner l'instance non étiquetée $u_i \in U_x$ qui minimisera le risque à l'itération $t+1$. On estime $R(\mathcal{M}^{+u_n})$ le risque *"attendu"* après l'étiquetage de l'instance u_n, on se base sur les données étiquetées dont on dispose. On suppose (dans le cas d'une classification binaire) que $f(u_n) = y_1$ ***[resp*** $f(u_n) = y_0$***]*** pour estimer

$\hat{R}(\mathcal{M}^{+(u_n,y_1)})$ ***[resp*** $\hat{R}(\mathcal{M}^{+(u_n,y_0)})$***]***. L'équation 3 montre comment agréger les estimations de risque selon les probabilités d'observer chacune des classes.

Pour exprimer la stratégie de réduction du risque sous forme algorithmique, il suffit de remplacer l'étape (B) de l'Algorithme 1 par : *"Rechercher l'instance* $q = argmin_{u \in U_x}\ \hat{R}(\mathcal{M}^{+u_n})$*"*.

$$R(\mathcal{M}) = \sum_{n=1}^{N} \sum_{y_i=0,1} \mathbb{1}_{\{f(l_n) \neq y_i\}}\, P(y_i|l_n) P(l_n) \qquad avec\, l_n \in L_x \tag{1}$$

$$\hat{R}(\mathcal{M}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{y_i=0,1} \mathbb{1}_{\{f(l_n) \neq y_i\}}\, \hat{P}(y_i|l_n) \tag{2}$$

$$\hat{R}(\mathcal{M}^{+u_n}) = \hat{P}(y_1|u_n)\hat{R}(\mathcal{M}^{+(u_n,y_1)}) + \hat{P}(y_0|u_n)\hat{R}(\mathcal{M}^{+(u_n,y_0)}) \qquad avec\, u_n \in U_x \tag{3}$$

Les approches d'apprentissage actif par réduction d'erreur sont des stratégies efficaces puisqu'elles ont un caractère exhaustif. Elles examinent tous les exemples candidats et toutes les valeurs d'étiquette possibles. Ces stratégies se différencient entres elles par la fonction de coût utilisée pour rendre compte de la qualité du modèle. Il est important de noter que ces stratégies sont fortement combinatoires. En effet, le choix de n exemples engendre $n.\|\mathbb{Y}\|.\|Ux\|$ entraînements du modèle.

4 Discussion et perspectives

A l'issue de cet état de l'art plusieurs questions peuvent être soulevées.

l'évaluation des stratégies d'apprentissage actif
La qualité d'une stratégie active est généralement représentée par une courbe mesurant la performance du modèle en fonction du nombre d'exemples étiquetés (voir figure 10). Le critère de performance (axes des ordonnées de la figure 10) utilisé peut prendre plusieurs formes selon le problème traité. Ce type de courbe permet uniquement de comparer les stratégies entre elles de manière ponctuelle, c'est-à-dire pour un nombre d'exemples fixé. Si on observe des courbes qui se croisent, il est impossible de déterminer si une stratégie est meilleure qu'une autre (sur la totalité du jeu de données). Une piste pour résoudre ce problème serait de mesurer l'apport d'une stratégie active par rapport à la stratégie alléatoire et d'intégrer ce nouveau critère sur la totalité du jeu de données. Nous nous intéressons actuellement à ce sujet.

L'ensemble de test
les méthodes d'apprentissage actif sont généralement utilisées dans le cas où l'acquisition des données est coûteuse. Dans la pratique, on ne dispose pas d'ensemble de test et il est difficile d'évaluer la qualité du modèle au cours de son apprentissage.

les critères d'arrêt d'un apprentissage actif
On peut soit définir un nombre maximal d'exemples à étiqueter, soit chercher un critère plus fin qui n'ajoute des exemples que si les progrès du modèle sont avérés. Des travaux futurs seront réalisés dans ce sens.

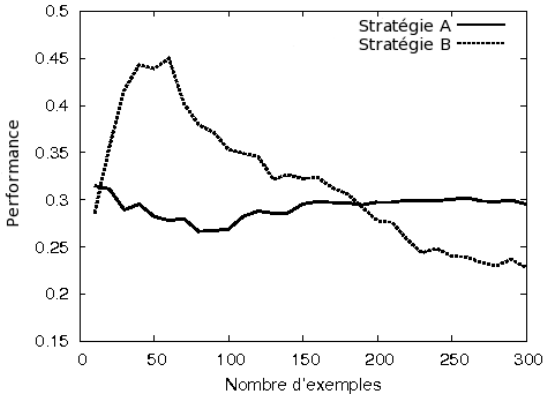

FIG. 10 – *Performance vs nombre d'exemples utilisés*

D'une manière générale, les stratégies d'apprentissage actif permettent d'estimer l'utilité des exemples d'apprentissage. Ces mêmes critères pourraient être utilisés dans le cadre d'un apprentissage en ligne. L'ensemble d'apprentissage serait constitué des N exemples les plus "utiles" vus jusqu'à présent (avec N fixé). Cette approche permettrait de considérer des problèmes d'apprentissage non stationnaires et donc d'effectuer un apprentissage qui s'adapte aux variations du système observé.

Références

Abe, N. et H. Mamitsuka (1998). Query learning strategies using boosting and bagging. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, San Francisco, CA, USA, pp. 1–9. Morgan Kaufmann Publishers Inc.

Castro, R. andWillett, R. et R. Nowak (2005). Faster rate in regression via active learning. In *NIPS (Neural Information Processing Systems)*, Vancouver.

Chapelle, O. et A. Zien (2005). Semi-supervised classification by low density separation. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics.

Chappelle, O. (2005). Active learning for parzen windows classifier. In *AI & Statistics*, Barbados, pp. 49–56.

Cohn, D., R. Caruana, et A. McCallum (2003). Semi-supervised clustering with user feedback. Technical Report TR2003-1892, Cornell University.

Cohn, D. A., L. Atlas, et R. E. Ladner (1994). Improving generalization with active learning. *Machine Learning 15*(2), 201–221.

Cohn, D. A., Z. Ghahramani, et M. I. Jordan (1995). Active learning with statistical models. In G. Tesauro, D. Touretzky, et T. Leen (Eds.), *Advances in Neural Information Processing Systems*, Volume 7, pp. 705–712. The MIT Press.

Dasgupta, S. (2005). Analysis of greedy active learning strategy. In *NIPS (Neural Information Processing Systems)*, San Diego.

Dima, C. et M. Hebert (2005). Active learning for outdoor obstacle detection. In *Proceedings of Robotics: Science and Systems*, Cambridge.

Ferrière, A. (1922). *L'école active*. Editions Forums.

Freinet, C. (1964). *Les invariants pédagogiques*. Bibliothèque de l'école moderne.

Freund, Y., H. S. Seung, E. Shamir, et N. Tishby (1997). Selective sampling using the query by committee algorithm. *Machine Learning 28*(2-3), 133–168.

Guestrin, c., A. Krause, et P. Singh (2005). Near-optimal sensor placements in gaussian processes. In *ICML (International Conference on Machine Learning)*, Bonn.

Harmon, M. (1996). Reinforcement learning: a tutorial. http://eureka1.aa.wpafb.af.mil/rltutorial/.

Jain, A. K., M. N. Murty, et P. J. Flynn (1999). Data clustering: a review. *ACM Computing Surveys 31*(3), 264–323.

Jamy, I., T.-Y. Jen, D. Laurent, G. Loizou, et O. Sy (2005). Extraction de règles d'association pour la prédiction de valeurs manquantes. *Revue Africaine de la Recherche en Informatique et Mathématique Appliquée ARIMA Spécial CARI04*, 103–124.

Lewis, D. et A. Gale (1994). A sequential algorithm for training text classifiers. In W. B. Croft et C. J. van Rijsbergen (Eds.), *Proceedings of SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, Dublin, pp. 3–12. Springer Verlag, Heidelberg.

McCallum, A. K. et K. Nigam (1998). Employing EM in pool-based active learning for text classification. In J. W. Shavlik (Ed.), *Proceedings of ICML-98, 15th International Conference on Machine Learning*, Madison, US, pp. 350–358. Morgan Kaufmann Publishers, San Francisco, US.

Muslea, I. (2002). *Active Learning With Multiple View*. Phd thesis, University of southern california.

Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics 33*, 1065–1076.

Roy, N. et A. McCallum (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, pp. 441–448. Morgan Kaufmann, San Francisco, CA.

Seung, H. S., M. Opper, et H. Sompolinsky (1992). Query by committee. In *Computational Learning Theory*, pp. 287–294.

Singh, A., R. Nowak, et P. Ramanathan (2006). Active learning for adaptive mobile sensing networks. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, New York, NY, USA, pp. 60–68. ACM Press.

Thrun, S. B. et K. Möller (1992). Active exploration in dynamic environments. In J. E. Moody, S. J. Hanson, et R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems*, Volume 4, pp. 531–538. Morgan Kaufmann Publishers, Inc.

Tong, S. et D. Koller (2000). Support vector machine active learning with applications to text classification. In P. Langley (Ed.), *Proceedings of ICML-00, 17th International Conference on Machine Learning*, Stanford, US, pp. 999–1006. Morgan Kaufmann Publishers, San

Francisco, US.

Zhu, X., J. Lafferty, et Z. Ghahramani (2003). Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML (International Conference on Machine Learning)*, Washington.

Summary

Machine learning indicates a vast whole of methods and algorithms which allow a model to learn a behavior thanks to examples. Active learning gathers a whole of methods of examples selection used to build training set for the predictive model. All the strategies aim to use the less examples as possible and to select the most informative examples. After having formalized the active learning problem and after having located it in the literature, this article synthesizes the main approaches of active learning and illustrates them thanks to toyes examples.