Rappel de Cours n°2

Principes qui sous-tendent l'Apprentissage Supervisé

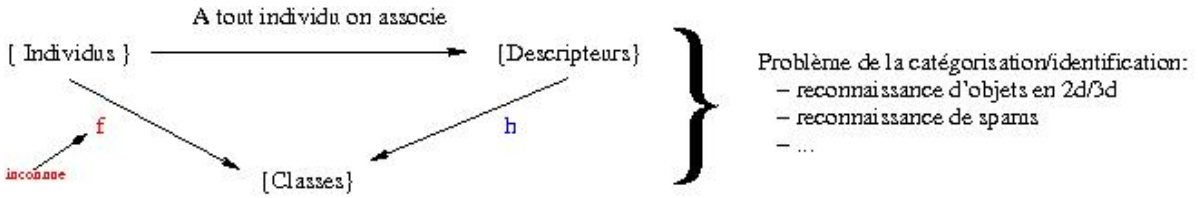

FIG. 1 – Idée générale

Le problème de la catégorisation vu sous l'angle de l'apprentissage peut être illustré comme suit :
On dispose d'un ensemble d'emails (les individus sur la figure) que l'on veut trier automatiquement en 2 catégories : spam/non-spam. On suppose qu'il existe un moyen (une fonction f) qui permet de dire avec certitude pour chaque email à quelle catégorie il appartient. Malheureusement, cette fonction f nous est inconnue. On choisi donc de représenter les emails par un ensemble de descripteurs. Par exemple :
– la fréquence d'apparition des lettres (permet notamment de retrouver la langue utilisée)
– certains mots clés (sex,viagra,money,$, ...)
– la longueur du message
– la source du message
– l'objet
– etc...

Ce changement de représentation a pour but de mettre en évidence certains traits des emails non explicités dans leur forme originale[1]. L'idée est alors d'arriver à détecter, par l'utilisation d'une fonction h sur ces descripteurs, d'éventuels points communs entre les individus d'une même catégorie. Dans le cadre de l'apprentissage supervisé, et pour notre exemple, cela veut dire que l'on dispose d'un certain nombre d'emails dont on connait la classe.

La fonction h est une hypothèse. La tâche de l'inférence inductive est, étant donné cette collection d'exemples de f connus, de déterminer une hypothèse h qui approxime f. Le problème de l'induction est de trouver une fonction h qui généralise bien, c'est à dire que h doit être capable de prédire correctement la classe à laquelle appartient un individu (un email) non encore vu.

L'évaluation des perfomances de h necessite par conséquent de ne pas utiliser toutes les exemples que nous connaissons durant la phase d'apprentissage. Une partie des emails dont nous connaissons la classe sera utilisée pour apprendre h (c'est l'ensemble d'apprentisssage). On évaluera ensuite la qualité de cette hypothese sur le reste des emails connus (c'est l'ensemble de test).

[1] le choix de ces descripteurs est donc une étape clé

On cherche donc à minimiser la différence entre h et f : $P(h \triangle f) \leq \epsilon$
et on souhaite de plus que la probabilité que h soit proche de f soit très forte, quelles que soient les données : $P(P(h \triangle f) \leq \epsilon) \leq 1 - \gamma$

Apprentissage PAC : On a alors l'intuition que toute hypothese h vraiment mauvaise a une forte probabilité de se tromper après un petit nombre de tests sur des individus dont on connais la classe. Par conséquent, on va considérer que toute hypothèse qui donne de bons résultats sur un ensemble d'exemple *suffisament* grand a peu de chances d'être loin de la vérité. Une telle hypothèse sera considérée comme **Probablement Approximativement Correcte**.

Que ce soit pour l'apprentissage ou pour l'évaluation de h, la question de la taille et de la forme des ensembles utilisés se pose. On va alors chercher :
- à définir quelle quantité et quels types d'exemples sont nécessaires pour pouvoir qualifier un ensemble de test représentatif.
- à quantifier le nombre d'exemples nécessaires pour que h soit dans l'ϵ-boule autour de f, c'est le critère de Vapnik-Chervonenkis.

Le problème du surapprentissage (overfitting) : Afin d'étudier l'évolution de la qualité de la solution h lors de l'apprentissage, on représente l'évolution de l'erreur de classification pour les ensembles de test (E_{test}) et d'apprentissage (E_{app}) sur la figure .

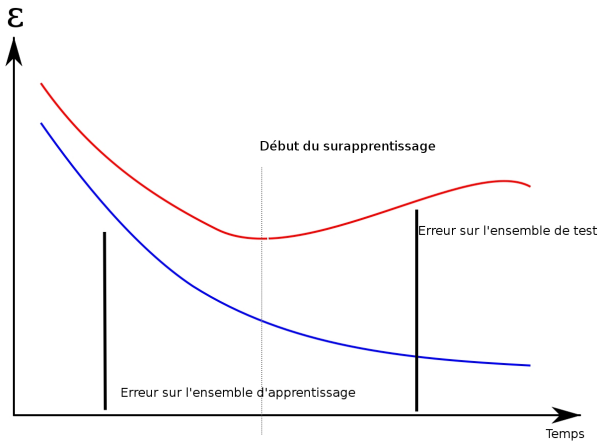

FIG. 2 – Ilustration du surapprentissage, taux d'erreur sur E_{app} et E_{test} [wikipedia]

La diminution du taux d'erreur E_{app} se traduit bien par une diminution de E_{test} lors de la première partie de l'apprentissage. Cependant, bien que le taux d'erreur en apprentissage continue de décroitre tout au long de la phase d'apprentissage, l'erreur sur l'ensemble de test (inconnu de l'algorithme d'apprentissage) se stabilise puis se met à augmenter. Ce phénomène s'explique par un surapprentissage de l'algorithme qui gènère l'hypothèse ; h "colle" aux données de l'ensemble d'apprentissage te ne permet plus de généraliser. En d'autre termes, si on lui donne suffisament (trop) de temps relativement à ces capacités et au nombre d'exemples considérés, l'algorithme d'apprentissage apprend "par coeur" les exemples de l'ensemble d'apprentissage et devient incapable de généraliser.
Le calcul de la dimension de Vapnik-Chervonenkis[2] doit donc prendre en compte ce critère pour la détermination de la taille de l'ensemble à considérer.

[2]calcul qui sort de cette introduction