Réglage de la largeur d'une fenêtre de Parzen dans le cadre d'un apprentissage actif : une évaluation

Vincent Lemaire, R&D France Telecom
2 avenue Pierre Marzin, 2300 Lannion – France
email : vincent.lemaire@orange-ftgroup.com

Alexis Bondu R&D France Telecom,
2 avenue Pierre Marzin, 2300 Lannion – France
email : alexis.bondu@orange-ftgroup.com

Maxime Chesnel, INSA Rennes
20, Avenue des Buttes de Coësmes, CS 14315, 35043 Rennes – France
email : maxime.chesnel@insa-rennes.fr

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. La fenêtre de Parzen est un modèle possible pour l'apprentissage actif. Dans cet article seule la fenêtre de Parzen munie du noyau gaussien à norme L^2 est considérée. La variance du noyau gaussien constitue le seul paramètre de ce modèle prédictif. Dans cet article, on se place dans un cas où un apprentissage actif débute avec peu de données étiquetées. En revanche, les données non étiquetées son abondantes. La question centrale de cet article est comment prérégler une fenêtre de Parzen dans de telles conditions. Trois méthodes permettant de faire ce réglage sont testées sur des jeux de données réelles.

Mots Clefs : Apprentissage actif, Fenêtre de Parzen, Réglage largeur du noyau

Abstract: Active learning consists in methods of examples selection that are used to iteratively build a training set of a predictive model. In practice, these methods are carried out in interaction with a human expert. A Parzen window is one possible predictive model for active learning. In this article, only Parzen windows with Gaussian kernel (L^2 norm) are considered. In this case, the variance of the kernel is the only one parameter of the predictive model. In selective sampling framework, active learning starts with few labeled examples and there is a lot of unlabelled data. The main issue of this article is how to adjust the parameter of a Parzen window in such conditions. Three different methods are tested to do this tuning on real data files.

Keywords: Active Learning, Parzen Window, Bandwidth Selection

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 XXe siècle, le pédagogue suisse Adolphe Ferrière (Ferriere, 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 interactions de l'étudiant avec son professeur correspondent à la possibilité pour le modèle d'interagir avec un expert humain (aussi appelé "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'interagir 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 a pour but d'évaluer la largeur du noyau gaussien d'une fenêtre de Parzen dans le cadre de l'apprentissage actif. La première section de l'article introduit le sujet et formalise l'apprentissage actif de manière générique. 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 à une description de la fenêtre de Parzen et des 3 méthodes testées permettant de régler la largeur du noyau gaussien. La troisième section présente une large expérimentation des 3 méthodes retenues dans la section 2. Enfin une discussion termine cet article.

2. Apprentissage actif

2.1. Généralités

L'apprentissage statistique (non supervisé, semi-supervisé), supervisé 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. Si le modèle parvient à faire de bonnes prédictions pour des individus qu'il n'a pas vus lors de son entraînement, alors le modèle généralise correctement ce qu'il a appris à de nouveaux cas.

La nature des données utilisées varie selon le mode d'apprentissage. 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. Parmi les méthodes d'apprentissage non supervisée on peut citer les méthodes de "clustering" (Jain, Murty et al., 1999) et les méthodes d'extraction de règles d'association (Jamy, Tao-Yuan et al., 2005).

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 (Cohn, Caruana et al., 2003) et la classification semi-supervisée (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 (Castro, Willett et al., 2005) 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.

2.2. Deux scénarii 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, Willett et al, 2005) distingue deux scénarii possibles : l'échantillonnage adaptatif et l'échantillonnage sélectif. Il s'agit de deux manières différentes de poser le problème de l'apprentissage actif.

Dans le cas de l'échantillonnage adaptatif (Singh, Nowak 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 œuvre. 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.

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. 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. On 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.

2.3 Un algorithme générique

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

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, qui estime l'intérêt d'une instance *x* pour l'apprentissage du modèle. Grâce à cette fonction, le modèle présente à l'oracle les instances pour lesquelles il espère la plus grande amélioration de ses performances.

3. La fenêtre de Parzen

3.1. Introduction

La fenêtre de Parzen est une méthode d'apprentissage par voisinage (Chappelle, 2005), proche de la méthode des k plus proches voisins. Elle permet de réaliser une prédiction sur une nouvelle instance, en prenant en compte les instances dont la proximité sera

jugée "suffisante". La différence entre la fenêtre de Parzen et la méthode des k plus proches voisins réside dans la notion de voisinage. Il est constant dans la méthode des k plus proches voisins (valeur de k) alors qu'il est défini par un noyau dans le cas de la fenêtre de parzen.

Notation : *n* représente le nombre d'instances disponibles dans la base de données (les exemples d'apprentissage), *i* représente l'index d'une des instances de la base de données, *x* représente une donnée pour laquelle on souhaite faire une prédiction. $K(x, x_i)$ représente le calcul de la fonction noyau (*K*) entre l'instance *x* et l'instance x_i. L'utilisation d'une fenêtre de Parzen en classification ou en régression est décrite dans le tableau 1 ci-dessous.

Classification	Régression
$\hat{P}(y\|x) = \frac{\sum_{i=1, y_i=y}^{n} K(x, x_i)}{\sum_{\ell=1}^{n} K(x, x_\ell)}$	$\widehat{y} = \sum_{i=1}^{n} \left(\frac{K(x, x_i)}{\sum_{\ell=1}^{n} K(x, x_\ell)} \right) y_i$
La prédiction de la classe *y* pour l'instance *x* est la somme normalisée des fonctions noyau pour la classe considérée entre *x* et l'ensemble des instances de la base d'apprentissage.	La prédiction de la valeur *y* de l'instance *x* est la somme normalisée des fonctions noyau entre cette instance *x* et l'ensemble des instances de la base d'apprentissage pondérée par leur valeur *y*

Tableau 1 - Utilisation de la fenêtre de Parzen

L'utilisation de la fenêtre de Parzen nécessite de faire le choix d'une fonction "noyau" qui définira la notion de proximité. Notre choix a été fait sur le noyau gaussien utilisant la norme euclidienne (Shawe-Taylor et Cristianini, 2004). Ce noyau utilise une matrice de covariance qui définit un noyau gaussien "multi varié". Une écriture de ce noyau gaussien multi varié est alors :

$$K_\Sigma(x - x_i) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - x_i)^T \Sigma^{-1}(x - x_i)\right)$$

Lorsque toutes les variances qui forment la matrice de covariance sont supposées indépendantes et égales (ce qui est rarement vrai mais une approximation très couramment utilisée) alors cette matrice est diagonale et les propriétés du noyau gaussien sont les mêmes selon toutes les directions de l'espace. Le noyau est alors qualifié de sphérique, car ses courbes de niveau forment des hyper-sphères. Ce n'est que dans ce cas très spécifiques que le noyau gaussien ne possède qu'un seul paramètre : σ. Le noyau gaussien de norme L2 peut être défini alors comme suit:

$$k(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\,\sigma^2}\right)$$

Nous adoptons ce noyau gaussien L2 pour toutes nos expériences. Il s'agit alors de régler correctement le paramètre de ce noyau. Ce qui équivaut au bon réglage de la fenêtre de Parzen.

3.2. Les bornes de l'hyper-paramètre σ

Lorsque que σ tend vers 0, les probabilités estimées par la fenêtre de Parzen tendent vers un dirac centré sur l'instance observée. Dans ce cas, les instances d'apprentissage éloignées de l'instance observée auront une influence nulle dans le calcul de la probabilité à prédire. Seules les quelques instances qui se situent dans le voisinage directe de l'instance observée interviennent dans la prédiction du modèle.

Dans nos expériences, on considère que le σ minimal (noté σ_{min}) ne peut pas être plus petit que le plus petit écart entre les données. On obtient σ_{min} en mesurant la distance qui sépare toutes les paires d'exemples d'apprentissage. On retient la plus petite distance.

Lorsque que σ tend vers ∞, on observe un phénomène de lissage. Toutes les gaussiennes deviennent plates, ce qui a pour conséquence d'affecter le même poids à tous les exemples d'apprentissage. Dans ce cas, quel que soit l'instance observée, la prédiction de la fenêtre de Parzen sera toujours la même. Le modèle prédira toujours la classe majoritaire dans un problème de classification, ou encore, la moyenne des valeurs dans un problème de régression. Le noyau gaussien peut être assimilé dans ce cas à un noyau constant.

Dans le cadre de nos expériences, on considère que le σ maximum (noté σ_{max}) est égale à la plus grande distance séparant un exemple d'apprentissage au barycentre des données

3.3 Les trois méthodes testées pour le réglage de l'hyper-paramètre σ

3.3.1 Le cadre : apprentissage actif et classification

Toutes les stratégies d'apprentissage actif cherchent à utiliser le moins d'exemples possible pour entraîner un modèle prédictif. Pour cela, les exemples les plus informatifs doivent être sélectionnés. Dans le cadre de nos expériences, on cherche à entraîner une fenêtre de Parzen pour des problèmes de classification (voir les bases de données section 4.1). Le seul paramètre de ce modèle est la variance du noyau gaussien muni de la norme euclidienne L^2. On se place du point de vue de l'échantillonnage sélectif : à chaque étape du processus d'apprentissage actif, on possède un ensemble d'apprentissage composé de données étiquetées et de données non étiquetées.

Dans cet article, la performance d'un modèle est mesurée grâce à l'aire sous la courbe de ROC (Fawcett, 2003), aussi appelée AUC. Plus précisément, on utilise l'espérance de l'AUC sur toutes les classes, puisqu'on traite des problèmes de classification qui ont plus de deux classes.

Trois méthodes différentes ont été utilisées pour chercher la valeur optimale de σ. Puisque ces méthodes sont utilisées dans le cadre d'un apprentissage actif, on évalue l'influence de la taille de l'ensemble d'apprentissage en faisant varier le nombre des données considérées. Pour chacune de ces méthodes, 1000 valeurs possibles de σ sont testées dans l'intervalle $[\sigma_{min}, \sigma_{max}]$.

3.3.2. Un réglage optimal

La première méthode utilisée pour le réglage de σ est idéale puisque toutes les étiquettes du jeu de données sont supposées connues. Cette approche peut paraître antagoniste avec l'apprentissage actif et n'est pas réalisable en pratique. Cependant, cette première approche donne une borne supérieure pour la performance de la fenêtre de Parzen.

La recherche du σ optimal se fait grâce à une "k-fold "cross validation" (Bengio et Grandvalet, 2003) sur l'ensemble d'apprentissage. Cette première méthode nous permet d'étalonner les résultats avec les deux autres méthodes décrites ci-dessous. Cette méthode est appelée "méthode 1" dans le tableau de résultats section 4.3. Les résultats y sont indiqués de manière à évaluer les deux autres méthodes.

3.3.3 Un réglage simple

La deuxième méthode est proposée par B. Schölkopf (Schölkopf, Mika et al., 1999) et rencontre un franc succès dans la communauté scientifique. Pour régler une fenêtre de Parzen, B. Schölkopf utilise la moyenne quadratique des écarts-types de chaque composante vectorielle et la dimension des données. Le calcul de l'écart type moyen des données est le suivant (x_u correspond aux variables descriptives, d est la dimension du jeu de données) :

$$\overline{\sigma} = \frac{1}{d}\sum_{u=1}^{d}\left(\mathrm{E}[x_u^2] - \mathrm{E}[x_u]^2\right)$$

On se place dans le cas où la matrice de covariance est diagonale (voir section 3.1). Pour le moment, cette stratégie n'est pas optimale puisque la fenêtre de Parzen n'atteint pas les "bords" de l'hyper-sphère. Cette incapacité augmente d'autant plus que le nombre de dimensions s'élève, le calcul de l'écart type moyen se faisant sur l'ensemble des dimensions. La stratégie adoptée par B. Schölkopf consiste à multiplier l'écart type moyen des données par le nombre de dimensions de manière à corriger cette incapacité. Le σ optimal est définit tel que : $\sigma = \sqrt{d}\overline{\sigma}$

3.3.4 Un réglage basé sur une régression

Pour σ variant de σ_{min} à σ_{max} (avec un pas σ_{pas}), **faire :**
  Pour toutes les d dimensions des instances
    Enlever une dimension des instances qui devient valeur cible
      Évaluer la fenêtre de Parzen de régression
      - réglée avec le sigma courant:
        - sur les instances de l'ensemble d'apprentissage
        - pour les instances de l'ensemble de test
        - avec la dimension choisie comme valeur à estimer
      Remplir un tableau de valeur prédite par instance de test
      Construire la courbe de REC
      Calculer l'aire sous la courbe de REC (AOC)
    Fin Pour
    Calculer la moyenne des AOC sur les dimensions
Fin Pour

Algorithme 2

La troisième stratégie consiste à transformer le problème de classification en *d* problèmes de régression (avec *d* le nombre de dimensions des données). On commence par faire abstraction des classes des données étiquetées. On retire une des *d* composantes des vecteurs constituant les données d'apprentissage. Cette composante devient la variable à prédire. On se ramène donc à un problème de régression supervisé à *d-1* dimensions. L'algorithme 2 présente cette stratégie de manière détaillée, et retient la valeur du σ qui se comporte au mieux sur toutes les dimensions.

On utilise l'aire au dessus de la courbe de REC (Bi et Bennett, 1998) pour évaluer la performance de la fenêtre de Parzen en régression, pour chacune des valeurs de σ testées. A la sortie de cet algorithme la valeur du σ optimal est testée en classification, sur un l'ensemble de test.

4. Experimentations

4.1 Les bases de données utilisées

Nous avons utilisé 7 jeux de données issus de "l'UCI machine learning repository" (http://www.ics.uci.edu/mlearn/MLSummary.html), 1 jeu de données réelles portant sur la détection d'émotions dans la parole 'emovoc' (Bondu, Lemaire et al., 2007) et un jeu de données synthétique $\mathrm{Sin}(x^3)$.

Données		d	Taille ensemble d'apprentissage	Taille ensemble de test
1	Iris	4	90	60
2	Glass	9	107	107
3	Wine	13	119	59
4	Pima	8	354	354
5	Segment	19	310	1998
6	Ionosphere	34	240	111
7	USPS2	256	317	1998
8	Emovoc	20	3783	1622
9	Sinx3	2	2000	30000

Tableau 2 : Jeux de données utilisés

4.2 Protocole Expérimental et évaluation

- Choix d'une des 9 bases de données décrite dans le tableau 2 .
- Normalisation des données d'apprentissage et de test par la moyenne et la variance.
- Chaque expérience est réalisée 10 fois de manière à obtenir une performance moyenne munie d'une mesure de variance.
- On fait varier la quantité d'exemples d'apprentissage en utilisant 25%, 50%, 75% et 100% du jeu de donnée
- Recherche de la valeur optimale de σ en utilisant une des trois méthodes précédemment décrite.
- Teste de la valeur optimale de σ en faisant les prédictions des étiquettes sur l'ensemble de test et en mesurant les performances à l'aide de l'AUC

Les résultats donnés à la section 4.3 sont des résultats moyennés sur les 10 expériences. On rappelle également que la recherche de la valeur optimale de σ a été réalisée dans un intervalle tel que décrit section 3.3.1.

4.3 Résultats

Pour des raisons de place on ne peut pas présenter ici tous les résultats (selon les 3 axes d'études considérés). Le tableau 3 présente[1] un résumé des résultats obtenus. Pour chaque méthode explorée on présente l'AUC obtenue en utilisant 100% et 25% de la base d'apprentissage, ceci en utilisant la valeur optimale de σ trouvée.

	Méthode 1		Méthode 2		Méthode 3	
	Σ (100% - 25%)	AUC (100% - 25%)	σ	AUC	σ (100% - 25%)	AUC (100% - 25%)
Iris	0.50 - 0.25	1.00 - 1.00	2	0.97	0.35 - 0.40	1.00 - 1.00
Glass	0.40 - 1.50	0.70 - 0.72	3	0.77	0.33 - 0.40	0.71 - 0.71
Wine	0.95 - 0.14	0.99 - 0.98	3.61	1.00	1.10 - 2.00	0.99 - 0.99
Pima	2.30 - 4.00	0.83 - 0.83	2.83	0.83	0.75 – 1.00	0.81 - 0.83
Segment	0.70 - 1.20	0.96 - 0.96	4.36	0.94	0.40 - 0.70	0.96 - 0.96
Ionosphere	1.30 - 1.50	0.99 - 0.95	5.83	0.89	0.75 – 1.00	0.99 - 0.99
SinX3	0.05 - 0.09	1.00 – 1.00	1.41	0.96	2.38 – 2.00	0.96 - 0.96
USPS2	3.10 – 3.00	0.93 - 0.93	16	0.83	2.90 - 2.90	0.93 - 0.93
Emovoc	5.20 – 8.00	0.91 - 0.92	4.47	0.91	0.80 - 0.01	0.89 - 0.80

Tableau 3 : Résultats

Comme prévu, la méthode 1 rassemble globalement les meilleures performances. Rappelons que cette méthode est idéale puisque les classes sont toutes supposées connues, ce qui n'est pas le cas dans la pratique.

On s'attendait d'abord à une plus grande pertinence de la méthode 3 qui réalise un apprentissage approfondi sur les données. En réalité, elle n'est pas beaucoup plus performante que la méthode 2. On constate malgré tout que les méthodes 2 et 3 offrent des résultats quasiment équivalents. Les résultats sont d'une manière générale assez bons par rapport à ceux observés grâce à la méthode 1. Puisqu'elles offrent des performances similaires, les méthodes 2 et 3 se départagent sur le critère de la rapidité d'exécution. De ce point de vue, la méthode 2 est la plus avantageuse.

5. Conclusion

L'apprentissage statistique à noyaux offre de bonnes possibilités d'apprentissage en particulier avec la fenêtre de Parzen. Cet outil permet, quand il est bien réglé, de fournir des performances intéressantes d'apprentissage en classification et en régression.

[1] La méthode 2 ne dépendant que du nombre de dimension du problème ses résultats ne varient pas en fonction de la taille de la base d'apprentissage utilisée pour régler le sigma (suivant notre protocole expérimental voir section …)

Cependant le réglage du noyau n'est pas aisé. La recherche de méthodes capables de régler le paramètre σ aussi finement que pour la classification supervisée est maintenant bien entamé. La méthode d'obtention de σ par les variances et la dimension des données paraît satisfaisante. Cependant, cette solution peut être incomplète, car elle ne convient pas à tous les jeux de données. Néanmoins, c'est celle que nous conseillerions au regard d'un critère alliant performances, rapidité de calcul, simplicité et spectre de validité.

Bibliographie

Bengio, Y., Grandvalet Y. (2003) No unbiased Estimator of the Variance of K-Fold Cross-Validation. Technical report 2003s-22, CIRANO, 2003.

Bi J., Bennett K. (1998) Regression Error Characteristic Curves, *International Conference on Machine Learning*, 1998.

Bondu, A., Lemaire, V., Poulain, B. (2007). Active learning Strategies: a case study for the detection of emotions in speech, *Industrial Conference on Data Mining,* 2007.

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

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

Chesnel, M., Lemaire, V., Bondu, A. (2007) Apprentissage Actif et Fenêtre de Parzen. Note Technique FT/RD/TECH/07/11/136

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

Fawcett, T. (2003) Roc graphs : Notes and practical considerations for data mining researchers. Technical Report HPL-2003-4, HP Labs, 2003.

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

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

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

Jamy, I., Tao-Yuan J, Laurent, D., Loizou, G., Oumar, S.. Extraction de règles d'association pour la prediction de valeurs manquantes. *Revue Africaine de la Recherche en Informatique et Mathématique Appliquée ARIMA*, 2005.

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

Roy, N., McCallum, 1. (2001). Toward optimal active learning through sampling estimation of error reduction. *In Proc. 18th International Conf. on Machine Learning,* pages 441-448. Morgan Kaufmann, San Francisco, CA, 2001.

J. Shawe-Taylor, J., Cristianini, N. (2004) *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P. Müller, Gunnar, K-R., Rätsch, Smola, A. J. (1999) Input Space Versus Feature Space in Kernel-Based Methods. *IEEE Transactions on Neural Networks*, 10(5), 1000 1017, 1999.

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