# Représentation des données par un comité de cartes auto-organisatrices : une application aux données bruitées.

Elie Prudhomme,Stéphane Lallich

Laboratoire ERIC,
Université Lyon 2,
5, av. Pierre Mendès-France
69676 Bron
eprudhomme@gmail.com,
http://eric.univ-lyon2.fr/∼eprudhomme/
stephane.lallich@univ-lyon2.fr,
http://eric.univ-lyon2.fr/∼slallich/

**Résumé.** Grâce aux approches ensemblistes, les performances en apprentissage supervisé sont devenues excellentes sans pour autant être trop coûteuses en temps. Cependant, ces méthodes ne permettent que la prédiction des données. Or, le couplage entre la prédiction et une méthode de représentation ajoute une valeur qualitative. La représentation permet de redonner la main à l'utilisateur, que ce soit avant la prédiction pour visualiser les données et juger de la qualité du modèle ou après en permettant l'exploration des exemples qui ont conduit à la prédiction. En outre, une représentation des données – obtenue indépendamment de la variable de classe – est robuste au bruit sur la variable de classe puisque celle-ci n'est pas intégrée dans l'apprentissage. Les données en grandes dimensions posent toutefois le problème d'obtenir une représentation de qualité en un temps raisonnable. Dans ce contexte, nous proposons le recours à un comité de cartes auto-organisatrices dont l'apprentissage est synthétisé par une carte supplémentaire, apprise grâce à un *stacking* de la position des neurones. Le comité tire parti du concept de diversité pour assurer une prédiction de qualité alors que le *stacking* géographique offre une représentation synthétique facilement manipulable par l'utilisateur final. Les expérimentations montrent que cette stratégie est compétitive par rapport aux approches spécialisées dans la prédiction tout en permettant une représentation des données. Enfin, elle permet de gérer des niveaux de bruit importants.

## 1 La représentation des données en apprentissage supervisé

Nous allons d'abord rappeler le cadre de l'apprentissage supervisé. Les données d'apprentissage sont constituées de $n$ exemples décrits par $d$ variables, notées $X_1, X_2, \ldots\ X_d$. En outre, on connaît la classe d'appartenance ou l'étiquette de chaque exemple, qui constitue la variable à prédire, notée $Y$. L'objectif est de prédire la classe d'appartenance d'un nouvel exemple dont

on connaît seulement les descripteurs. Les deux familles de méthodes résolvant ce problème consistent soit à raisonner sur l'ensemble des exemples connus à chaque prédiction, soit à construire un modèle de décision. Le but recherché est la prédiction de l'étiquette d'un nouvel exemple dont on connaît les descripteurs, que l'on construise un modèle à partir des exemples étiquetés dont on dispose pour l'apprentissage ou que l'on stocke ces exemples.

Une partie très importante du travail d'apprentissage est l'étape de préparation des données qui précède la prédiction proprement dite. Au cours de cette étape, il faut être capable de sélectionner les variables pertinentes (par exemple Guerif et Bennani (2007)) et de choisir leur spécification, tout en limitant l'impact des exemples atypiques. Il est donc très important de pouvoir explorer efficacement ses données grâce à une navigation intelligente qui favorise la contextualisation. La représentation des données, au sens où nous allons la définir, répond à cette exigence.

Par représentation des données, nous entendons la démarche suivante :

- tout d'abord, on visualise les proximités entre exemples issues des descripteurs, en utilisant une méthode adaptée telle que les graphes de voisinage. On peut aussi utiliser les arbres phylogénétiques, les analyses factorielles ou les cartes de Kohonen ;
- ensuite, on visualise les étiquettes des différents exemples ;
- enfin, après avoir validé statistiquement la capacité prédictive de cette représentation, on s'en sert pour fonder la prédiction d'un nouvel exemple sur la base de son voisinage.

Pour illustrer cette démarche, prenons le cas des graphes de voisinage, largement développé dans différentes thèses effectuées sous la direction de D.A. Zighed (Clech, 2004; Muhlenbach, 2002; Hacid, 2004). Chaque exemple constitue un sommet du graphe. Deux exemples voisins au sens des prédicteurs sont reliés par une arête. Les sommets du graphe sont étiquetés par $Y$. Si on arbitre entre finesse et complexité, la structure de voisinage la plus commode est le graphe des voisins relatifs de Toussaint et Menard (1980) où deux exemples $a$ et $b$ sont voisins s'il n'existe pas d'exemple qui soit à la fois plus proche de $a$ que $b$ et plus proche de $b$ que $a$. Le principal outil utilisé lors de la préparation des données est la statistique des arêtes coupées (Sebban, 1996), une statistique de type *cross-product*, voisine du gamma de Hubert en analyse spatiale, qui admet une version globale et une version locale (Lallich, 2002). On appelle arête coupée, une arête qui joint deux exemples de classe différente. Le nombre total d'arêtes coupées évalue la qualité globale de la représentation et il est prédictif de la qualité de la prédiction en généralisation. Le nombre d'arêtes coupées au voisinage d'un exemple constitue un indice de qualité locale à partir duquel il est possible de repérer les exemples atypiques (Muhlenbach, 2002). Pour utiliser ces statistiques, il faut avoir au préalable établi leur loi sous l'hypothèse d'absence de structure ou d'atypicité, ce qui permet de calculer la p-value de la statistique observée (Lallich, 2002). Enfin, la structuration des exemples par un graphe de voisinage facilite la mise en œuvre de stratégies de navigation et d'interrogation des exemples, en particulier face à des données complexes (voir les thèses de Clech (2004) et Hacid (2004) sur ces thèmes).

Le développement des moyens de stockage et d'acquisition automatique des données a un triple effet sur les données. La réalité des données a changé, elles sont de nature et de source hétérogènes, le plus souvent bruitées. Leur volumétrie devient très importante, tout à la fois en nombre d'exemples, ce qui pose le problème du passage à l'échelle des algorithmes utilisés, et en nombre de prédicteurs, ce qui expose à la malédiction de la dimension (Donoho, 2000). Enfin, il devient plus rare que les exemples soient tous étiquetés, ce qui peut rendre néces-

saire de recourir à l'apprentissage semi-supervisé. Pour répondre à ces nouveaux enjeux, nous proposons une stratégie de représentation fondée sur un comité de cartes auto-organisatrices.

Dans la section suivante, nous commençons par présenter les cartes auto-organisatrices et comment on peut les adapter à la prédiction. La section 3 résume les principaux travaux sur l'erreur d'un ensemble et nous permet, en section 4, de proposer deux heuristiques, l'une non-supervisée (section 4.1) et l'autre supervisée (section 4.2), pour sélectionner les cartes à intégrer au comité en vue de réduire l'erreur en prédiction. Ces deux stratégies sont testées en section 4.3. Dans la section 5, nous présentons le principe du *stacking* géographique qui nous permet de synthétiser les cartes du comité. En section 6, nous montrons l'efficacité de cette approche face à des données bruitées. Dans la section 7, enfin, nous tirons les conclusions de ce travail et nous proposons une extension possible à l'apprentissage semi-supervisé.

# 2 Cartes auto-organisatrices

## 2.1 Introduction

Les cartes auto-organisatrices (ou *Self Organizing Map*, *SOM*) (Kohonen, 1982) permettent à la fois un apprentissage non-supervisé rapide des individus et leur représentation. Pour ce faire, elles se composent d'un réseau de neurones répartis uniformément dans un espace à 2 voire 3 dimensions. Chaque neurone est défini par un vecteur dans l'espace des individus, appelé vecteur de poids. Lors de l'apprentissage, les individus sont présentés successivement au réseau. Pour chaque individu, le neurone le plus proche (dit *Best Matching Unit, bmu*) et son voisinage dans le réseau sont modifiés afin qu'ensemble ils se rapprochent de l'individu.

L'étape classique d'apprentissage du $i$-ème individu à l'instant $t$ peut se résumer par la formule suivante de modification des poids $w$ du neurone $r$ :

$$w_r^{t+1} = w_r^t + \alpha^t \times v_r^t \times (x_i - w_r^t)$$

où $\alpha^t$ est le pas d'apprentissage qui décroît linéairement avec le temps et $v_r^t$ la fonction de voisinage qui définit l'étendue des neurones modifiés autour du $bmu$. En début d'apprentissage, tout à la fois les neurones à modifier se rapprochent fortement des individus ($\alpha^t$ grand) et le nombre de neurones à modifier autour du $bmu$ (le voisinage) est important ($v^t$ grand). Par la suite, l'ampleur de la modification et le nombre de neurones à modifier diminuent. Grâce à cet algorithme, on obtient une conservation de la topologie locale de l'espace des entrées. Pour deux individus, une proximité au sens des neurones correspond à une proximité dans l'espace de départ. La complexité des cartes auto-organisatrices est linéaire en $O(nmd)$ (avec $n$ le nombre d'individus, $m$ le nombre de neurones et $d$ le nombre d'attributs). Ceci en fait l'un des algorithmes d'apprentissage non-supervisé les plus rapides.

## 2.2 Application à l'apprentissage supervisé

Du fait de ces propriétés (rapidité de l'apprentissage et préservation de la topologie), plusieurs auteurs ont adapté les cartes auto-organisatrices à l'apprentissage supervisé. Dans ce cadre, l'approche la plus couramment utilisée est certainement la quantification vectorielle supervisée (notée *LVQ* pour *Learning Vector Quantization*) proposée par Kohonen (1988). Les neurones des cartes auto-organisatrices sont remplacés par des vecteurs de poids associés à

l'une des classes à apprendre. L'apprentissage détermine itérativement le vecteur le plus proche de chaque individu présenté. A la différence des cartes auto-organisatrices, il sera le seul modifié. La règle d'apprentissage utilise alors la classe : si le vecteur et l'individu partagent la même classe, le vecteur sera modifié de manière à se rapprocher de l'individu, dans le cas contraire il s'en éloignera.

Cette approche garde la simplicité et la faible complexité de l'algorithme des cartes auto-organisatrices face à la volumétrie des données. Cependant, elle n'effectue aucune projection de l'espace des entrées. C'est pour pallier ce manque que le modèle *LASSO* a été proposé (Midenet et Grumbach, 1994). Il adapte directement l'algorithme des cartes auto-organisatrices à l'apprentissage supervisé. Pour cela, il étend les vecteurs de poids des neurones à l'étiquette. Ainsi l'espace appris par auto-organisation n'est plus seulement celui des prédicteurs mais celui défini par les prédicteurs et l'étiquette.

Les deux méthodes précédentes se servent de l'étiquette des individus pour établir les poids des vecteurs prototypes. Dans le cas de la quantification vectorielle supervisée, il s'agit de rapprocher ou d'éloigner ces vecteurs les uns des autres afin de les regrouper en fonction de l'étiquette qu'ils doivent représenter. Pour le modèle *LASSO*, l'étiquette sert de la même manière que les prédicteurs à établir la position de ces vecteurs dans l'espace des entrées et sur la carte. Cette utilisation de l'étiquette contribue à une plus grande efficacité des méthodes en phase de prédiction, mais pose le problème de la conservation de la topologie. En effet, dans le cas de la quantification vectorielle supervisée, il n'est plus possible de représenter la topologie de l'espace des entrées puisque la position des prototypes ne consiste plus en une simple projection de cet espace sur celui d'une carte. Pour le modèle *LASSO*, il existe bien encore une carte représentative de l'espace des entrées, mais celle-ci dépend à la fois des prédicteurs et de l'étiquette.

Afin d'obtenir une représentation topologique de l'espace des entrées indépendante de la classe, à la manière de Zighed et al. (2004) pour les graphes de voisinage, d'autres travaux ont séparé l'apprentissage en deux phases. Une première phase réalise la construction de la carte en se fondant uniquement sur les variables prédictives, la deuxième phase se charge d'étiqueter les neurones obtenus. L'étiquette correspond à la classe majoritaire parmi les individus représentés par le neurone. Néanmoins, au terme de l'apprentissage, tous les neurones ne peuvent pas être étiquetés soit parce qu'ils ne représentent aucun exemple (neurone vide) soit parce qu'ils représentent équitablement des exemples de classes différentes (neurone ambigu). Une fois la carte étiquetée, une fonction de prédiction attribue son étiquette à un nouvel individu. Sur ce modèle, différentes méthodes ont été proposées. Elles divergent seulement sur le fonctionnement de la fonction de prédiction face aux neurones vides ou ambigus. C'est le cas de la méthode *Kohonen-KNN* (Zupan et al., 1994), qui sera améliorée par *Kohonen-WI* (Song et Hopke, 1996) puis par *Kohonen-Opt* (Prudhomme et Lallich, 2005). Pour prédire un exemple représenté par un neurone vide ou ambigu, la méthode *Kohonen-KNN* réalise un vote parmi les neurones voisins de ce neurone, alors que *Kohonen-WI* se fie à l'étiquette du neurone le plus proche au sens des poids. *Kohonen-Opt* met en place un indicateur qui tient compte de ces deux informations, poids et représentation de l'étiquette dans le voisinage.

L'avantage majeur est de fournir, outre un modèle de prédiction, une carte qui constitue la projection de l'espace des prédicteurs. Du fait de la construction non supervisée de la carte,

cette projection est indépendante de l'étiquette des données. La construction du modèle n'est donc pas dépendante de la classe, ce qui est particulièrement intéressant dans le contexte de données où le bruit sur la classe induit l'apprentissage en erreur.

# 3 Méthodes ensemblistes

Les cartes auto-organisatrices réalisent une projection de l'espace d'apprentissage dans un espace à 2 dimensions. La qualité de cette projection dépend souvent de la taille de l'espace de description. Dans le cas de données en grandes dimensions, la prédiction se trouve détériorée. Pour garder une prédiction robuste quelle que soit la taille de l'espace de description, nous proposons d'intégrer ces cartes au sein d'un ensemble (appelé également comité). Ces ensembles sont une combinaison de plusieurs modèles de prédiction en un modèle unique dont la qualité dépasse généralement celle de ces membres. Dans cette section, nous décrivons le fonctionnement des ensembles afin de pouvoir ensuite l'adapter à un comité de cartes (section 4).

## 3.1 Introduction : le jury de Condorcet

Pour présenter de façon intuitive l'intérêt de faire prendre une décision par un comité composé de classifieurs suffisamment divers, nous allons prendre l'exemple du jury de Condorcet (Condorcet 1785). Le jury doit choisir entre deux décisions possibles, l'une bonne, l'autre mauvaise. Il est composé d'un nombre impair de juges, noté $J$, et prend sa décision à la majorité de ses membres. Les $J$ juges se prononcent indépendamment avec la même probabilité $p$ de se tromper. On peut donc assimiler le nombre de juges prenant la mauvaise décision à une loi binomiale de paramètres $J$ et $p$, notée $B(J, p)$, ce qui permet de calculer le risque d'erreur du jury, notée $p_{jury} : p_{jury} = \Pr(B(J, p) > \frac{J}{2})$

Le tableau 3.1 indique le risque d'erreur du jury en fonction du nombre de juges (colonne 1) et du risque d'erreur $p$ commun aux différents juges (ligne 1). Par exemple, un jury de $15$ membres ayant chacun $30\%$ de risque d'erreur n'a que $5$ chances sur $100$ de se tromper ! Plusieurs conclusions peuvent être tirées de ce tableau. La première est que le jury diminue considérablement le risque d'erreur, lorsque le risque d'erreur de chaque juge est inférieur à $50\%$. Au contraire, il l'augmente, lorsque le risque d'erreur de chacun dépasse $50\%$. En deuxième lieu, on notera que ce résultat repose sur l'indépendance des juges. Si un jury de 15 membres est divisé en 3 blocs dont les membres votent systématiquement de la même façon, avec 30 chances sur 100 de se tromper, ce qui arrive parfois dans la vie réelle, le risque d'erreur du jury monte à 22% au lieu de 5% . Enfin, si au contraire, les membres du jury se trompent sur des cas différents (corrélation négative), on perçoit que le risque d'erreur du jury sera encore diminué par rapport au résultat du tableau 3.1.

Illustrons ce dernier point. Avec 15 juges indépendants ayant chacun un risque d'erreur de 40%, d'après le tableau 3.1, le risque d'erreur du jury tombe à 21%. Cependant, un arbitre suprême, connaissant dans chaque cas la bonne décision, pourrait organiser les erreurs de telle sorte que celles-ci soient le plus réparties possibles entre les exemples sur lesquels porte la décision. Par exemple, si le jury de 15 juges devait statuer sur 15 cas, il serait possible à l'arbitre par une disposition adéquate des erreurs de limiter à 6 le nombre d'erreurs par cas. C'est ainsi

| $J$ \ $p$ | 0,05 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,95 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0,05 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 | 0,95 |
| 3 | 0,01 | 0,03 | 0,10 | 0,22 | 0,35 | 0,50 | 0,65 | 0,78 | 0,90 | 0,97 | 0,99 |
| 5 | 0,00 | 0,01 | 0,06 | 0,16 | 0,32 | 0,50 | 0,68 | 0,84 | 0,94 | 0,99 | 1,00 |
| 7 | 0,00 | 0,00 | 0,03 | 0,13 | 0,29 | 0,50 | 0,71 | 0,87 | 0,97 | 1,00 | 1,00 |
| 9 | 0,00 | 0,00 | 0,02 | 0,10 | 0,27 | 0,50 | 0,73 | 0,90 | 0,98 | 1,00 | 1,00 |
| 11 | 0,00 | 0,00 | 0,01 | 0,08 | 0,25 | 0,50 | 0,75 | 0,92 | 0,99 | 1,00 | 1,00 |
| 13 | 0,00 | 0,00 | 0,01 | 0,06 | 0,23 | 0,50 | 0,77 | 0,94 | 0,99 | 1,00 | 1,00 |
| 15 | 0,00 | 0,00 | 0,00 | 0,05 | 0,21 | 0,50 | 0,79 | 0,95 | 1,00 | 1,00 | 1,00 |
| 17 | 0,00 | 0,00 | 0,00 | 0,04 | 0,20 | 0,50 | 0,80 | 0,96 | 1,00 | 1,00 | 1,00 |
| 19 | 0,00 | 0,00 | 0,00 | 0,03 | 0,19 | 0,50 | 0,81 | 0,97 | 1,00 | 1,00 | 1,00 |
| 21 | 0,00 | 0,00 | 0,00 | 0,03 | 0,17 | 0,50 | 0,83 | 0,97 | 1,00 | 1,00 | 1,00 |
| 23 | 0,00 | 0,00 | 0,00 | 0,02 | 0,16 | 0,50 | 0,84 | 0,98 | 1,00 | 1,00 | 1,00 |
| 25 | 0,00 | 0,00 | 0,00 | 0,02 | 0,15 | 0,50 | 0,85 | 0,98 | 1,00 | 1,00 | 1,00 |
| 27 | 0,00 | 0,00 | 0,00 | 0,01 | 0,14 | 0,50 | 0,86 | 0,99 | 1,00 | 1,00 | 1,00 |
| 29 | 0,00 | 0,00 | 0,00 | 0,01 | 0,14 | 0,50 | 0,86 | 0,99 | 1,00 | 1,00 | 1,00 |
| 31 | 0,00 | 0,00 | 0,00 | 0,01 | 0,13 | 0,50 | 0,87 | 0,99 | 1,00 | 1,00 | 1,00 |

TAB. 1 – *Risque d'erreur du jury en fonction du nombre de juges $J$ (colonne 1) et du risque d'erreur $p$ commun aux différents juges (ligne 1)*

que dans chaque cas, le vote du jury donnerait toujours 9 voix sur 15 à la bonne décision, ce qui assurerait un risque nul. La stratégie illustrée est bien sûr inapplicable, car elle suppose que l'on connaisse la bonne décision, mais elle donne une première piste, la maximisation de l'entropie de la distribution du nombre d'erreurs par cas.

## 3.2 Décomposition de l'erreur d'un ensemble

Comme le montre l'expérience du jury de Condorcet, la diversité entre les réponses des jurés – au sens où les jurés ne commettent pas leurs erreurs sur les mêmes exemples – joue un rôle déterminant dans l'erreur finale du jury. Pour mesurer et optimiser cette diversité sur un ensemble d'apprenants, plusieurs mesures ont été proposées. Néanmoins, aucune corrélation forte entre ces mesures et le taux d'erreur en généralisation de l'ensemble n'a pu être clairement établie sur des données réelles (voir Kuncheva et Whitaker, 2003). La diversité des prédictions joue pourtant un rôle dans l'erreur d'un ensemble comme le montrent les différentes décompositions de l'erreur d'un ensemble en régression.

Ueda et Nakano (1996) obtiennent une décomposition de l'erreur d'un ensemble en régression qui est fonction du biais et de la variance. Si l'on note $f^m$ la prédiction du $m$-ième apprenant et $d$ la valeur à prédire, ils montrent que l'erreur quadratique moyenne d'un ensemble de $M$ apprenants en régression s'écrit :

$$E\left\{\left(\frac{\sum_m f^m}{M} - d\right)^2\right\} = \overline{bias}^2 + \frac{1}{M}\overline{var} + \left(1 - \frac{1}{M}\right)\overline{covar}\,. \qquad (1)$$

avec $\overline{bias}$, $\overline{var}$ et $\overline{covar}$ respectivement le biais, la variance et la covariance moyenne des apprenants de l'ensemble. Par rapport à la décomposition classique en biais/variance de l'erreur d'un classifieur seul (Geman et al., 1992), cette décomposition introduit un terme de covariance. Celle-ci quantifie la corrélation qui existe entre les prédictions des apprenants de l'ensemble et peut, en prenant des valeurs négatives, réduire l'erreur finale. En régression, la covariance est donc une expression de la diversité. Par ailleurs, le nombre de classifieurs de l'ensemble joue également un rôle important. D'une part, il réduit la variance et, d'autre part, il augmente l'importance de la covariance.

Un autre résultat a été obtenu par Krogh et Vedelsby (1995a). Ils ont décomposé l'erreur d'un ensemble de la manière suivante :

$$\left(\frac{\sum_m f^m}{M} - d\right)^2 = \sum_m^M (f^m - d)^2 - \sum_m^M \left(f^m - \frac{\sum_m f^m}{M}\right)^2 . \tag{2}$$

Dans cette décomposition, le premier terme correspond à la somme des erreurs commises par les apprenants alors que le deuxième terme (appelé terme d'ambiguïté) mesure les différences entre les prédictions d'un apprenant et celles de l'ensemble. À partir de l'équation 1, Brown (2005) montre que ce deuxième terme contient à la fois une part de variance et de covariance moyenne de l'erreur de l'ensemble.

Ces deux décompositions obtenues en régression, montrent que, outre le nombre d'apprenants, l'erreur d'un ensemble dépend fortement de la précision de chaque apprenant et de leur diversité. Elles expliquent ainsi qu'il n'y ait pas de corrélation claire entre la diversité et le taux d'erreur sur des données réelles.

Dans le cas de la classification, Tumer et Gosh (1995) formalisent le problème différemment. Pour une variable prédictive $x$ et deux classes $i$ et $j$, ils s'intéressent aux probabilités *a posteriori* réelles de prédire correctement $i$ et $j$, notées $P(i|x)$ et $P(j|x)$. Lorsque les valeurs de $x$ ne suffisent pas à séparer les classes $i$ et $j$, il existe une erreur bayésienne irréductible (en gris clair sur la figure 1). Dans ce cas, $x^*$ représente la valeur de $x$ pour laquelle il n'est pas possible de décider entre les classes $i$ et $j$ (c'est à dire $P(i|x) = P(j|x)$). De plus, à cause des paramètres du modèle et du nombre fini d'individus, $P(i|x)$ et $P(j|x)$ ne sont qu'approximés respectivement par les fonctions $f_i(x)$ et $f_j(x)$ produites par les classifieurs. Outre l'erreur irréductible, il existe alors une erreur ajoutée (en gris foncé sur la figure). La frontière de décision entre les classes $i$ et $j$ devient alors $x_b$. Dans ce cadre théorique, Tumer et Gosh (1995) ont montré que l'erreur ajoutée par un ensemble s'écrit :

$$E_{add}^{ens} = \left(\frac{1 + \delta \times (M-1)}{M}\right) E_{add} . \tag{3}$$

où $E_{add}$ représente l'erreur ajoutée d'un apprenant et $\delta$ la corrélation entre les erreurs d'approximation des probabilités *a posteriori* commises par les différents apprenants. De même que pour la covariance, il s'agit là encore d'une mesure de la diversité. En effet, dans le cas où les erreurs sont commises sur les mêmes exemples ($\delta = 1$), les apprenants sont dépendants, l'erreur de l'ensemble est alors celle d'un apprenant seul. En revanche, lorsque les erreurs sont commises indépendamment ($\delta = 0$), l'erreur de l'Ensemble est $\frac{1}{M}$ fois l'erreur d'un apprenant.

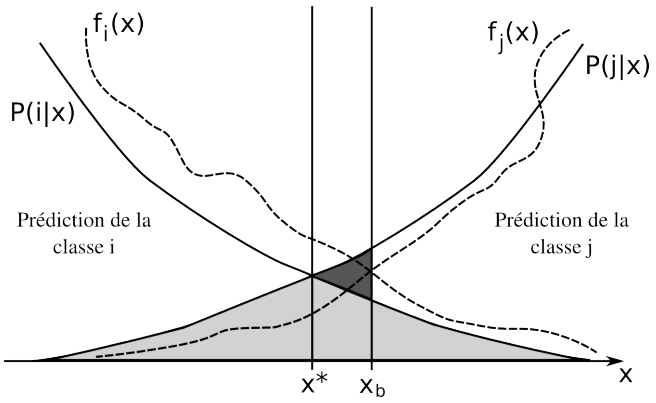


FIG. 1 – *Régions d'erreur associées aux approximations des probabilités a posteriori : en gris clair, l'erreur bayésienne irréductible et en gris foncé, l'erreur ajoutée par le classifieur (Tumer et Gosh, 1995).*

Dans ce cadre théorique, Zanda et al. (2007) remarquent que réduire l'erreur ajoutée revient soit à rapprocher $x_b$ de $x^*$, soit à faire en sorte que l'écart entre les probabilités *a posteriori* (noté $d_{ij} = P(i|x) - P(j|x)$) soit retrouvé grâce aux estimations du classifieur $f_{ij} = f_i(x) - f_j(x)$. En effet, si on dispose d'un classifieur tel que $d_{ij} = f_{ij}$, alors les décisions prises grâce aux prédictions des classifieurs sont toujours conformes aux décisions qui auraient été prises si les probabilités *a posteriori* réelles étaient connues. Dans ce cas, l'erreur ajoutée par le classifieur est nulle. Comme $d_{ij}$ et $f_{ij}$ sont des grandeurs réelles et non plus catégorielles comme l'étiquette à prédire, Zanda et al. (2007) se ramènent à un problème de régression pour lequel l'estimateur est $f_{ij}$ et la cible $d_{ij}$. En tirant parti des travaux antérieurs de Krogh et Vedelsby (1995a) (eq. 2), ils expriment l'erreur d'un ensemble en régression de la manière suivante :

$$E = \sum_{i=1}^{c} \sum_{j>i} (\bar{f}_{ij} - d_{ij})^2 \quad (4)$$

$$= \sum_{i=1}^{c} \sum_{j>i} \left[ \frac{1}{M} \sum_{m=1}^{M} (f_{ij}^m - d_{ij})^2 \right] - \sum_{i=1}^{c} \sum_{j>i} \left[ \frac{1}{M} \sum_{m=1}^{M} (\bar{f}_{ij} - f_{ij}^m)^2 \right] \quad (5)$$

$$= \bar{E} - A \, . \quad (6)$$

avec $f_i^m(x)$ l'estimation de $P(i|x)$ pour le $m$-ième classifieur, $M$ le nombre de classifieurs, $c$ le nombre de classes, $\bar{f}_i(x) = \sum_m f_i^m(x)$ l'estimation de $P(i|x)$ par l'Ensemble, $f_{ij}^m = f_i^m(x) - f_j^m(x)$ et $\bar{f}_{ij} = \bar{f}_i(x) - \bar{f}_j(x)$.

Là encore, l'erreur peut être décomposée en deux termes. Le premier, $\bar{E}$, correspond à l'erreur moyenne des classifieurs de base alors que le deuxième, $A$, correspond à une mesure de la diversité, équivalente au terme d'ambiguïté dans les travaux de Krogh et Vedelsby (1995b). Il est donc possible, en augmentant la diversité et en gardant l'erreur moyenne des classifieurs constante, de diminuer l'erreur de l'ensemble.

# 4 Comité de cartes auto-organisatrices

Les cartes auto-organisatrices en apprentissage supervisé ne permettent pas une prédiction très robuste de données en grandes dimensions. Rassembler ces cartes dans un comité va permettre d'améliorer leur performance. Cependant, l'objectif demeure de pouvoir visualiser les données. Aussi, le comité doit-il limiter au maximum le nombre de cartes qu'il contient afin de pouvoir les synthétiser en vue d'une visualisation. En outre, plus le nombre de cartes est important, plus le temps de calcul augmente. Les résultats théoriques de la section précédente, nous apportent un certain nombre d'informations sur les conditions que doivent remplir les classifieurs pour pouvoir être agrégés avec succès. Si l'on considère l'équation 2 proposée par Krogh et Vedelsby (1995a), l'erreur en généralisation d'un ensemble est la différence entre deux termes, l'erreur quadratique des classifieurs par rapport à la valeur à prédire et l'erreur quadratique de ces mêmes classifieurs par rapport à la prédiction de l'ensemble (le terme d'ambiguïté). L'objectif, en régression, est donc de diminuer l'erreur des classifieurs tout en augmentant l'écart entre leurs prédictions. Cependant, comme l'ensemble est une approximation de la valeur à prédire, plus l'erreur des classifieurs est faible, plus il est probable que leurs écarts à la prédiction de l'ensemble soit faible. Il existe ainsi un compromis entre la qualité des classifieurs et la diversité de leur prédiction qui semble être également valable en classification (Zanda et al., 2007).

Une première solution pour contourner ce problème consiste à utiliser le hasard pour apprendre un grand nombre de classifieurs différents. C'est notamment la stratégie utilisée avec succès par le bagging et les forêts aléatoires. Cependant, pour optimiser le nombre de cartes du comité, il est nécessaire de mettre en place des heuristiques qui contrôlent les cartes qui sont membre du comité. Pour réaliser ce contrôle, nous envisageons deux stratégies qui se différencient d'abord par l'utilisation ou non de l'étiquette de classe des exemples dans le choix des cartes du comité. L'avantage de prendre en compte l'étiquette est bien entendu de minimiser efficacement l'erreur. La section 4.2 détaille la construction d'un tel comité. Cependant, il est parfois préférable de ne pas utiliser cette étiquette lors de la construction du comité, comme nous l'exposons en section 4.1. C'est notamment le cas lorsque l'étiquette est bruitée, pour éviter de biaiser le modèle, ou en apprentissage semi-supervisé, pour prendre en compte les exemples qui n'ont pas d'étiquette lors de la construction du modèle. La section 6 revient sur l'utilisation d'un comité de cartes dans le cas de données bruitées.

Quelle que soit la stratégie utilisée pour déterminer les cartes à intégrer au comité (supervisée ou non), ces cartes doivent se différencier les unes des autres afin de présenter la diversité nécessaire au succès du comité, tout en assurant à chacune une précision suffisante. La solution la plus généralement usitée consiste à restreindre l'apprentissage de chaque classifieur à un sous ensemble des données, que ce soit via les exemples, les attributs, la classe ou une combinaison de ces éléments. Dans le cas de données en grandes dimensions, la multitude d'attributs engendre des problèmes d'apprentissage et de complexité d'apprentissage. Une solution à ces problèmes dans le cadre d'un ensemble est donc d'apprendre chaque membre de l'ensemble sur un sous-espace d'attributs. L'avantage est double. D'une part, cette stratégie permet d'éviter à chaque membre d'être confronté aux problèmes des espaces de grandes dimensions et d'autre part, elle permet au comité de prendre en compte l'ensemble des attributs lors de

l'apprentissage, contrairement à une stratégie de sélection des attributs durant une phase de prétraitement.

## 4.1 Comité non supervisé

L'intérêt d'un comité construit sans référence à l'étiquette de classe est de pouvoir l'appliquer dans des situations où cette étiquette est soit bruitée soit disponible pour une partie seulement des exemples, sans pour autant que la qualité finale du comité n'en soit diminuée.

Comme nous l'avons expliqué dans la section précédente, l'objectif est de constituer des groupes d'attributs dont chacun engendre une carte à la fois de qualité et diverse par rapport à une carte apprise sur un autre groupe d'attributs. Il est donc nécessaire d'accéder à la diversité et à la qualité des cartes, non pas au travers de leur prédiction, mais au travers des attributs sur lesquels elles sont apprises. Pour la diversité, en considérant une paire de cartes à 1 dimension construites chacune sur un attribut, ces cartes seront d'autant moins diverses que les attributs seront corrélés, puisque deux attributs fortement corrélés portent la même information sur la variable de classe. A l'inverse, pour la qualité, en considérant une carte construite sur une paire d'attributs, cette carte aura une précision d'autant plus grande que les attributs seront orthogonaux, puisque deux attributs orthogonaux apportent une information supplémentaire sur la variable de classe. Ces deux hypothèses placent la corrélation entre les attributs comme un facteur déterminant de la diversité et de la qualité des cartes du comité. Elles réinterprètent également le compromis soulevé dans la section précédente entre la qualité des cartes et leur diversité, qui nécessite respectivement une corrélation intra-groupe et inter-groupe faible. Pour construire un ensemble divers, une solution consisterait donc à partitionner l'ensemble des attributs en grappes composées d'attributs homogènes puis à apprendre chaque carte sur une grappe afin d'obtenir les cartes les plus diverses possibles. Cependant, ces cartes auraient une qualité individuelle trop faible pour permettre à la prédiction d'être améliorée lors de l'agrégation (ce que nous avons vérifié par l'expérimentation). Nous avons donc choisi une approche différente qui consiste à apprendre des cartes sur des groupes d'attributs hétérogènes en prenant un attribut dans chaque grappe.

Par ailleurs, bien que nous souhaitions fonder la diversité de notre comité sur les attributs, il est possible d'augmenter cette diversité en apprenant chaque carte sur des individus différents. Pour se rapprocher de l'indépendance entre les cartes, une solution consiste à construire des groupes d'individus mutuellement exclusifs et d'apprendre chaque carte sur un groupe. En plus d'améliorer la diversité, cette solution réduit les temps de calcul nécessaires à la construction de l'ensemble. Cependant, dans le cas des espaces de grandes dimensions, un des problèmes concerne justement le nombre d'exemples qui est souvent trop faible pour décrire correctement l'espace d'apprentissage. En particulier, dans les expériences que nous avons menées, le nombre d'exemples des bases de test, comparativement à leur dimensionnalité, ne permet pas de mettre en œuvre une stratégie de ce type. En revanche, dans le cas de données très volumineuses, cette stratégie peut permettre d'améliorer la prédiction en diminuant les temps de calcul.

Finalement, pour former ces groupes, nous avons procédé en deux étapes. La première étape constitue des grappes d'attributs corrélés par le biais d'une classification non supervisée. Parmi différentes méthodes possibles (classification hiérarchique (Ward, 1963) ou k-moyennes (McQueen, 1967) sur la matrice attributs/valeurs transposée, par exemple), nous avons choisi la méthode Varclus (SAS, 1989). Elle est conçue spécifiquement pour la classification d'attributs

et présente l'avantage de déterminer automatiquement le nombre de grappes. Il s'agit d'une méthode divisive. A chaque itération, l'algorithme calcule les deux premiers vecteurs propres de la matrice attributs/valeurs autour desquels se formeront les groupes. Pour cela, chaque attribut est affecté au vecteur propre avec lequel il est le plus corrélé. Cependant, pour éviter que le premier vecteur propre ne soit corrélé avec la plus grande majorité des variables, une rotation orthogonale des vecteurs propres est effectuée par la méthode varimax (Kaiser, 1958). Chacune des 2 grappes formées conduit à l'obtention d'une nouvelle matrice attributs/valeurs ayant un nombre réduit d'attributs. On itère alors sur chacune de ces matrices jusqu'à ce que la deuxième valeur propre obtenue soit inférieure à 1. A l'issue de cette procédure, on obtient une partition en $K$ grappes d'attributs. En termes de complexité, Varclus exige d'abord le calcul de la matrice des corrélations entre attributs, soit une complexité en $d^2$. Ensuite, pour chacune des $\log_2 K$ itérations, les corrélations entre chaque attribut d'une part et les deux premiers vecteurs propres d'autre part sont comparées, soit $2 \times d \times \log_2 K$ comparaisons. Au final, la complexité de Varclus est donc en $d^2 + 2 \times d \times \log_2 K$.

A l'issue de Varclus, $k$ nouveaux groupes de $d'$ attributs ($d' < d$, avec $d$ le nombre d'attributs initial du problème) sont formés en piochant pour chaque groupe un attribut dans chaque grappe, et en prenant soin de ne sélectionner un attribut une deuxième fois au sein d'une grappe que lorsque tous les attributs de cette grappe ont été selectionnés (de façon à distribuer au mieux les attributs entre les groupes). De cette manière, les groupes obtenus sont formés d'attributs peu corrélés entre eux, afin de garantir la qualité des cartes, tout en étant constitué chacun d'un jeu d'attributs différent pour assurer la diversité des cartes. Enfin, chaque carte est apprise sur $d'$ attributs pour contourner les problèmes liés aux espaces de grandes dimensions.

## 4.2 Comité supervisé

L'utilisation de l'heuristique précédente n'optimise pas de manière explicite l'erreur du comité. Elle permet toutefois de mettre en place un comité de qualité sans recourir à la variable de classe. Cette dernière peut cependant être utilisée pour estimer l'erreur du comité, en particulier au travers de la mesure $E$ (Eq. 4, Sect. 3.2). Etant donnée cette mesure, il est donc possible d'estimer *a posteriori* (i.e. une fois que le comité a été appris), l'erreur en généralisation. Nous l'avons donc utilisée pour déterminer les sous-espaces de l'espace de description initial à apprendre par les différentes cartes du comité.

Cependant, le nombre de sous-espaces et de leurs combinaisons a une combinatoire exponentielle avec le nombre de dimension. Comme nous considérons le problème des espaces de grandes dimensions, nous proposons deux heuristiques pour limiter l'espace de recherche. Premièrement, nous choisissons de ne nous intéresser qu'aux sous-espaces de $d' = \sqrt{d}$ attributs. Ce petit nombre d'attributs permet d'augmenter les chances que deux sous-espaces conduisent à des classifieurs divers tout en restant suffisamment important pour obtenir des classifieurs de bonne qualité. Deuxièmement, afin de contrôler le nombre de cartes à apprendre, nous choisissons de ne nous intéresser qu'à un nombre fini $k$ de sous-espaces. Ces $k$ sous-espaces sont déterminés aléatoirement avant l'optimisation de l'erreur par un algorithme génétique (AG). Nous nous posons donc le problème de trouver, étant donné $k$ cartes apprises sur $d'$ attributs, le sous-ensemble de ces cartes qui optimise la mesure $E$. Par exemple, si $d = 100$ et $k = 12$, 12 cartes sont d'abord apprises sur des sous-espaces de taille 10. Ensuite, l'AG cherche la meilleure combinaison de ces cartes au sens de la mesure $E$. Les cartes sont apprises en amont

de l'AG et ne sont pas réapprises au cours de l'optimisation. Seule la mesure $E$ est estimée durant ce processus, en positionnant chaque exemple de validation sur chaque carte du comité.

L'optimisation de $E$ est conduite à l'aide d'un AG dans sa forme standard (Goldberg, 1989) dont la fonction *fitness* est la mesure $E$. Les génotypes manipulés par l'AG sont des vecteurs binaires de taille $k$. Leur $i$-ème bit encode la présence, 1, ou l'absence, 0, de la $i^{\text{e}}$ carte dans l'ensemble final. Ainsi un phénotype de la forme 1 0 0 1 1 0 1 1 0 0 0 1 représente un ensemble composé des 6 cartes numérotées 1, 4, 5, 7, 8, 12. Après une population initiale générée aléatoirement, chaque nouvelle population est sélectionnée à partir de la population précédente par le mécanisme de la roue de la fortune. Pour ce faire, chaque phénotype est associé à une probabilité $p$, proportionnelle à son *fitness*, d'être reproduit. De cette manière, les meilleurs phénotypes au sens du *fitness* (donc de $E$) auront plus de chance d'être reproduits que ceux ayant un score faible avec cette mesure. Les opérateurs génétiques de mutation et de *crossing-over* sont ensuite appliqués à cette population afin d'obtenir une nouvelle génération. Pour la mutation, chaque bit du phénotype a une probabilité $p_{mut}$ de changer de valeur. En ce qui concerne le *crossing-over*, 2 phénotypes parents tirés au sort ont une probabilité $p_{co}$ d'être croisés. S'ils sont croisés, un point de croisement est déterminé au hasard entre les deux extrémités du phénotypes. A partir de ce point, deux nouveaux phénotypes différents sont instanciés, possédant la partie du phénotype du premier parent à gauche (resp. à droite) du point de croisement associée à la partie du phénotype du deuxième parent à droite (resp. à gauche) du point de croisement. Enfin, le processus d'évolution est répété jusqu'à ce qu'un phénotype de la population ait un fitness de 1 (on a alors convergence) ou que le nombre d'itérations atteigne un seuil fixé par avance (voir la Fig. 2).

Notons que la sélection d'un sous-ensemble de cartes à partir d'un ensemble initial est un cas particulier de la pondération des cartes de cet ensemble. Cette pondération particulière affecte un poids de 1 aux cartes sélectionnées et un poids de 0 aux cartes éliminées. Une alternative à la sélection consiste donc à pondérer chaque carte lors de l'agrégation des prédictions. Nous avons d'abord fixé cette pondération en ayant recours à la qualité individuelle à l'aide d'une statistique corrélée avec leur taux d'erreur en généralisation (Prudhomme et Lallich, 2005). Néanmoins, les expériences que nous avons menées sur ce type de comité n'ont pas été concluantes. En effet, contrairement à la mesure $E$, la pondération obtenue ne prend pas en compte la diversité des cartes mais seulement leur qualité.

## 4.3 Pertinence des comités : expérimentations

Pour tester la pertinence de ces deux approches, nous les avons comparées à une carte auto-organisatrice en apprentissage supervisé sur l'ensemble des attributs, aux forêts aléatoires (Breiman, 2001), au boosting d'arbres C4.5 (Freund et Schapire, 1995) et aux 3-plus proches voisins (3-PPV) sur 7 jeux de données provenant de l'UCI (Newman et al., 1998) présentés dans le tableau 2. Pour les forêts aléatoires et le boosting, les implémentations utilisées sont celles de Tanagra (Rakotomalala, 2005). Les taux d'erreurs ont été obtenu par des 2-cross-validations répétées 5 fois, comme recommandé par Dietterich (1998). Ils sont reportés en figure 3.

L'apprentissage de chaque carte a été réalisé en deux étapes. Au cours de la première étape, rapide, les poids ont été organisés grossièrement : chaque exemple a été présenté une fois et les

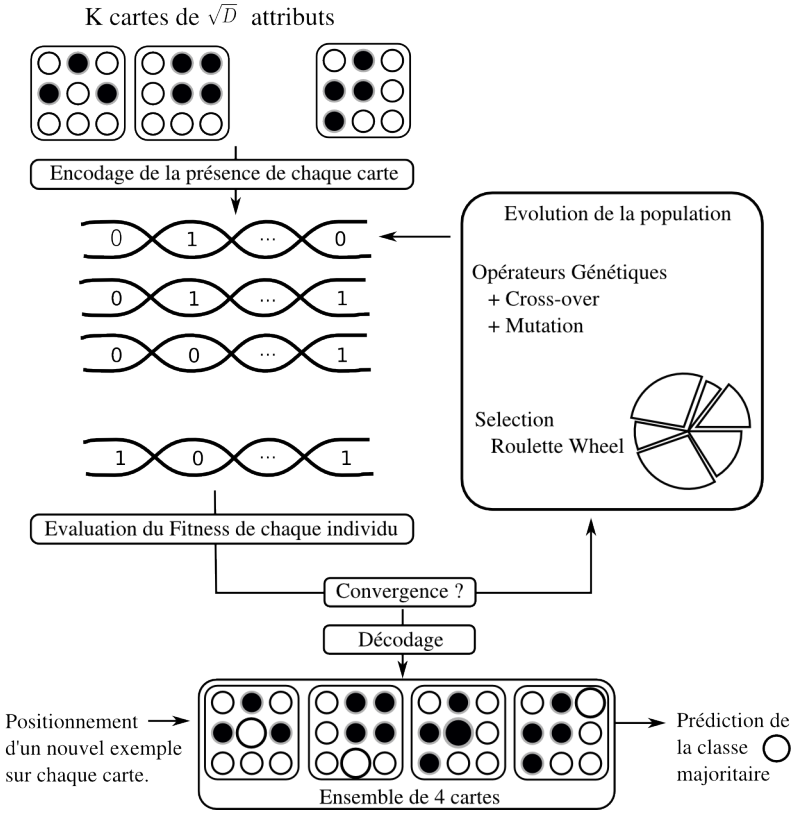


FIG. 2 – *Les différentes étapes de l'optimisation d'un ensemble par un AG.*

poids ont été modifiés à partir d'un pas d'apprentissage initial de $0,9$ et d'un rayon de voisinage initial égal au quart de la largeur de la carte. Durant la deuxième étape ($15000$ itérations), les poids ont été ajustés plus finement à partir d'un pas d'apprentissage initial de $0,1$ et d'un rayon de voisinage initial de 2. Enfin, quelle que soit l'étape, nous avons fait décroître le pas d'apprentissage et le rayon de voisinage linéairement avec le temps. En ce qui concerne la topologie de chaque carte, nous avons opté pour des neurones carrés disposés sur des cartes rectangulaires, suivant les expérimentations menées par Ultsch et Herrmann (2005). D'après Kohonen (2001), la taille des cartes doit être proportionnelle à la dimensionnalité des données afin de permettre une meilleure projection. Par ailleurs, en apprentissage supervisé, l'objectif n'est pas de décrire chaque classe par un neurone mais plutôt d'obtenir une description précise de la répartition des étiquettes dans l'espace. Suivant ces constatations, nous avons fixé la taille de nos cartes en fonction de la dimension des données à représenter, en prenant pour largeur $l = 2\sqrt{d}$ et pour hauteur $h = 3l/2$. Pour les données *wbdc*, *ionosphère*, *spectrometer*, *multi-features (profile)* qui ont peu d'exemples, ce paramétrage conduit à des cartes qui ont plus de neurones qu'il n'y a d'exemples à apprendre. Dans ces cas, nous fixons le nombre de neurones au $\frac{2}{3}$ du nombre d'exemples d'apprentissage, en gardant un rapport de $\frac{3}{2}$ entre la hauteur et la largeur de la carte. Le tableau 2 indique les tailles de cartes utilisées pour chaque jeu de

données.

Le paramétrage du comité non supervisé requiert de déterminer $k$, le nombre de groupes d'attributs à apprendre à partir des $K$ grappes et $d'$, le nombre d'attributs de ces groupes. Nous avons choisis $d' = K$ afin d'avoir un attribut de chaque grappe dans les groupes à apprendre et $k = K$ afin d'apprendre un nombre raisonnable de cartes.

Enfin, le comité supervisé demande le paramétrage de l'algorithme génétique. Nous avons choisi une population initiale de $40$ phénotypes reproduits sur $1000$ générations avec un taux de mutation de $0,025$ et un taux de *cross-over* de $0,85$. La recherche du meilleur comité a été effectuée sur une base de $100$ cartes auto-organisatrices apprises sur $70\%$ de l'ensemble d'apprentissage, les $30\%$ restant ayant servi à l'estimation de la fonction *fitness*.

| **Données** | **Attr.** | **Cl.** | **Ex.** | **Répartition** | **Taille** |
|---|---|---|---|---|---|
| (1) Wbdc | 30 | 2 | 569 | 212/357 | $9 \times 14$ |
| (2) Ionosphère | 34 | 2 | 351 | 126/225 | $7 \times 11$ |
| (3) Spambase | 57 | 2 | 4601 | 1813/2788 | $14 \times 21$ |
| (4) Optdigits | 64 | 10 | 5620 | - | $14 \times 21$ |
| (5) Multi-features (Fourier) | 76 | 10 | 2000 | - | $9 \times 13$ |
| (6) Spectrometer lrs | 100 | 10 | 531 | - | $16 \times 24$ |
| (7) Multi-features (Profile) | 216 | 10 | 2000 | - | $17 \times 26$ |

TAB. 2 – *Caractéristiques des données en termes de nombre d'attributs (Attr.), nombre de classes (Cl.) et nombre d'exemples (Ex.). La colonne « répartition » indique la proportion d'exemples de chaque classe lorsqu'il existe un déséquilibre dans la répartition des exemples entre les classes. La colonne Taille précise la taille des cartes utilisées lors de l'apprentissage par les comités.*

Les résultats montrent d'abord que le comité améliore l'apprentissage d'une carte seule sur l'ensemble des attributs, que ce soit pour la méthode supervisée ou non-supervisée. Ils valident ainsi les stratégies d'optimisation de l'erreur utilisées pour déterminer les cartes du comité. Ces stratégies ne donnent cependant pas des résultats équivalents. La stratégie supervisée est plus performante sur la plupart des données grâce notamment à la prise en compte explicite de la diversité dans le choix des cartes. En effet, la moyenne des taux d'erreur des cartes associées au comité supervisé est supérieure à la moyenne de ces taux pour le comité non-supervisé (voir figure 4). Puisque le gain de performance obtenu par l'agrégation ne provient pas de la performance individuelle des classifieurs, c'est qu'il provient de leur diversité. Enfin, les comités de cartes supervisés permettent d'obtenir des performances similaires aux forêts aléatoires (le boosting demeurant dans la plupart des cas la meilleure stratégie). Dans la section suivante, nous montrons comment nous souhaitons profiter de ces comités pour accéder à une représentation des données, ce qui n'est possible ni avec le boosting ni avec les forêts aléatoires.

Pour les jeux de données déséquilibrées, nous avons reporté le taux d'erreur par classe pour les comités de cartes dans le tableau 3. Pour chaque jeu de données étudié, la classe majoritaire bénéficie d'une meilleure qualité de prédiction. Cet avantage se construit à plusieurs niveaux. Lors de l'apprentissage, d'abord, les régions de l'espace d'apprentissage les plus fournies en exemples deviennent les mieux représentées par chaque carte de l'ensemble. Lors de la prédiction, ensuite, le vote à la majorité – mis en place pour chaque neurone vain-

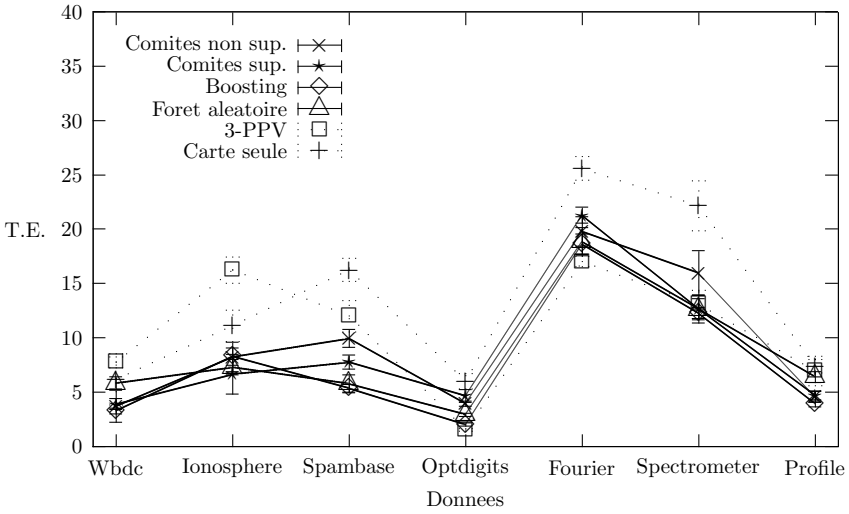


FIG. 3 – *Taux d'erreur des comités de carte supervisé et non-supervisé comparé à ceux d'une carte seule apprise sur la totalité des attributs, du boosting et des forêts aléatoires.*

queur des cartes du comité – favorise mécaniquement la classe majoritaire. Pour tenir compte de ce deuxième biais, Guérif et al. (2006) proposent de remplacer le vote à la majorité par un test du $\chi_2$ pour déterminer si la proportion d'exemples de chaque classe observée dans le neurone est significativement différente de la proportion observée dans les données. Ce problème n'est pas spécifique aux cartes auto-organisatrices. Il a été étudié pour d'autres méthodes de prédiction et plus particulièrement pour les arbres de décision. Pour cette méthode, Ritschard et al. (2007) proposent d'étiqueter chaque feuille de l'arbre en retenant la règle de décision qui a la plus forte intensité d'implication. Cette stratégie est transposable des feuilles de l'arbre aux neurones d'une carte auto-organisatrice.

| | Wbdc | Ionosphère | Spambase |
|---|---|---|---|
| Comité non sup. | 1.68/7.17 | 2.75/18.1 | 4.57/17.12 |
| Comité sup. | 2.18/6.79 | 5.68/8.41 | 2.37/17.03 |

TAB. 3 – *Taux d'erreur par classe pour les jeux de données déséquilibrés obtenus par les comités de cartes supervisés et non supervisés. La première valeur donne le taux d'erreur obtenu sur la classe majoritaire.*

# 5 Visualisation

Un ensemble de cartes améliore fortement les capacités prédictives d'une carte seule. Cependant, il rend plus difficile la visualisation et la navigation à travers les données. Pour y par-

venir, il est nécessaire de synthétiser la connaissance du comité en une structure plus simple, ici une carte synthétique, que l'utilisateur pourra manipuler, que ce soit pour visualiser les données ou naviguer à travers elles.

## 5.1 *Stacking* géographique

Pour construire une carte synthétique qui tire parti des informations issues des cartes de l'ensemble, nous proposons d'adapter le *stacking* utilisé en apprentissage supervisé à l'apprentissage non-supervisé. En apprentissage supervisé, le *stacking* a été introduit par Wolpert (1992). À la suite de l'apprentissage d'un ensemble de classifieurs, il consiste à réaliser un second apprentissage pour lequel chaque exemple est représenté non plus par ses attributs, mais par les prédictions des classifieurs de l'ensemble. Le *stacking* a pour objectif d'améliorer la prédiction de l'ensemble. Cependant, en cherchant à construire une carte synthétique, nous ne cherchons pas à améliorer la prédiction du comité mais à assurer sa représentation.

Chaque carte du comité est une projection de l'espace d'apprentissage qui associe à chaque exemple un neurone positionné dans l'espace de la carte. Ces associations entre un exemple et le neurone vainqueur de chaque carte constituent la représentation de cet exemple par le comité. En effet, deux exemples proches dans l'espace d'apprentissage vont avoir tendance à être représentés par des neurones proches sur chaque carte du comité. Ainsi, plus les cartes du comité auront appris cette proximité entre exemples avec précision, et seront nombreuses à la représenter, plus les exemples seront proches au sens des neurones qui les représentent. De cette manière, la proximité entre les exemples est considérée au sens des neurones qui les représentent sur chaque carte plutôt qu'au sens des prédicteurs. Pour obtenir une carte synthétique du comité de cartes, une stratégie consiste alors à apprendre les exemples en considérant une définition de la proximité issue de la position des neurones vainqueurs. L'espace d'apprentissage de cette carte synthétique est ainsi construit à partir des coordonnées des neurones vainqueurs d'un exemple sur chaque carte. Par exemple, pour un comité de 3 cartes, si un exemple est représenté sur la $1^{\text{re}}$ carte par le neurone de coordonnées $(3, 2)$, sur la $2^{\text{e}}$ par celui de coordonnées $(1, 5)$ et sur la $3^{\text{e}}$ par celui de coordonnées $(4, 7)$, il aura comme nouvelles coordonnées $(3, 2, 1, 5, 4, 7)$ dans l'espace d'apprentissage de la carte synthétique. Dans le domaine de la modélisation cognitive, une telle stratégie a été envisagée par Newman et Polk (2007) pour rendre compte de l'émergence et de l'organisation de concepts de haut niveau à partir des représentations topologiques issues des différents cortex.

A la manière du *stacking*, le résultat de l'apprentissage des classifieurs de l'ensemble est appris par un nouveau classifieur. Cependant, à la différence du *stacking*, ce n'est pas la prédiction des classifieurs qui est apprise mais la position des exemples dans les différents espaces de projection. C'est pourquoi nous parlons de *stacking* géographique. Il poursuit également un but différent du *stacking* original qui vise l'amélioration de la prédiction. Ici, le *stacking* géographique améliore la projection obtenue par chaque carte du comité afin de pouvoir proposer une projection unique de meilleure qualité. En outre, la carte synthétique offre une indexation des autres cartes. En effet, chaque neurone de cette carte possède des coordonnées dans son espace d'apprentissage, c'est à dire celui des coordonnées des neurones vainqueurs de chaque carte. Il est donc possible d'associer, à chaque neurone de la carte, un neurone sur chaque carte du comité. De cette manière, pour chaque exemple positionné sur la carte, on peut trouver

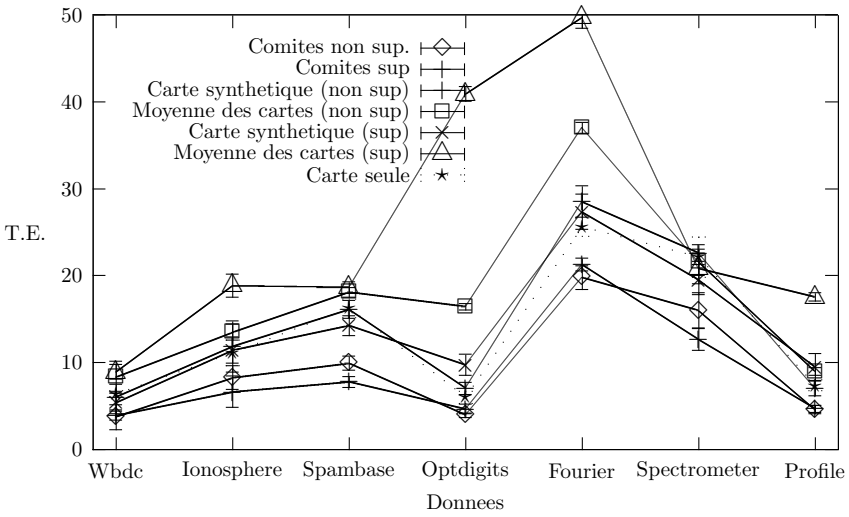


FIG. 4 – *Taux d'erreur d'une carte synthétique comparé à celui d'une carte seule apprise sur la totalité des attributs, d'un comité de cartes et à la moyenne des taux d'erreur des cartes du comité.*

une approximation des neurones vainqueurs sur les cartes de l'ensemble, et par la suite, des exemples associés à ces neurones.

## 5.2 Expérimentations

Pour ces expérimentations, les données présentées dans la table 2 de la section 4.3 ont été reprises afin d'estimer les taux d'erreur des cartes synthétiques issues d'un *stacking* géographique sur les comités supervisé et non supervisé. Les résultats obtenus sont reportés sur la figure 4. Ils sont comparés, pour chaque jeu de données, aux taux d'erreur du comité non supervisé, à la moyenne des taux d'erreur des cartes des comités supervisé et non supervisé et à une carte auto-organisatrice apprise sur l'ensemble des attributs (carte seule).

Les cartes synthétiques présentent un taux d'erreur intermédiaire entre la moyenne des taux d'erreur obtenus sur les cartes des comités et les taux d'erreur obtenu par les comités eux-mêmes. En ce sens, le stacking géographique tire parti de la diversité des cartes du comité pour construire une meilleure représentation, même si cette stratégie demeure moins efficace en prédiction que le vote à la majorité des cartes. La qualité de cette représentation (évaluée ici par sa capacité prédictive) est équivalente à celle d'une carte seule construite sur l'ensemble des attributs, à l'exception de celle obtenue sur les données *Fourier*.

Le stacking géographique permet ainsi d'obtenir une représentation qui bénéficie, pour la prédiction et l'indexation des exemples, du comité de cartes sous-jacent. Cette représentation sert d'abord à visualiser les données à travers les méthodes spécifiques développées pour les cartes auto-organisatrices (Vesanto, 1999), mais pas seulement. Elle peut également jouer un rôle dans le processus d'aide à la décision.

### 5.3 Utilisation en aide à la décision

La carte synthétique du comité sert également en situation d'aide à la décision pour expliquer la prédiction. Ainsi, lorsque l'utilisateur soumet un nouvel exemple, l'étiquette renvoyée par l'ensemble est facilement associée aux exemples qui lui sont proches au sens de la carte. Cette proximité se détermine en considérant les exemples d'apprentissage qui ont le même neurone vainqueur que l'exemple soumis. Lorsque le neurone vainqueur ne représente aucun exemple ou que l'utilisateur veut une plus grande précision, il est possible de considérer la proximité au sens de l'ensemble en utilisant les exemples qui ont le même neurone vainqueur sur chaque carte de l'ensemble. Dans ce deuxième cas, les exemples correspondant sont pondérés en fonction du nombre de neurones vainqueurs qui les représentent dans l'ensemble. De cette manière, l'utilisateur peut mieux juger de la prédiction, par analogie avec les exemples qui ont servi à la réaliser, et ainsi détecter les éventuelles erreurs. Enfin, cette carte peut servir à la navigation pure, pour rechercher des cas similaires à un exemple, en proposant à l'utilisateur d'explorer le voisinage de l'exemple à l'aide des neurones de la carte.

## 6 Application au contrôle des données bruitées

### 6.1 Le bruit

Sur les données réelles, il n'est pas rare que du bruit soit présent sur la variable de classe. Contrairement au bruit sur les attributs, le bruit sur la classe dégrade l'apprentissage supervisé (Quinlan, 1990).

Une première stratégie filtre les données avant l'apprentissage en comparant l'étiquette prédite avec un algorithme donné à l'étiquette réelle. Les exemples pour lesquels il existe un désaccord sont retirés avant l'apprentissage par un autre algorithme. Le travail fondateur est celui de Wilson (1972). Il filtre les exemples avec un algorithme 3-NN avant de réaliser la prédiction avec un algorithme 1-NN. Plus récemment, plusieurs auteurs ont suggéré le recours aux ensembles pour réaliser ce filtrage (Brodley et Friedl, 1996; Berthelsen et Megyesi, 2000; Verbaeten et Van Assche, 2003). Ces travaux exploitent la diversité des classifieurs qui composent l'ensemble en les faisant voter (à la majorité ou au consensus) pour les exemples à considérer comme bruités. Ces exemples sont ensuite retirés de l'apprentissage ou réétiquetés.

Une autre solution efficace au problème des données bruitées consiste à utiliser des algorithmes tolérants au bruit sur la classe. D'un point de vue théorique, de tels algorithmes existent comme le montre le modèle des requêtes statistiques proposé par (Kearns, 1993). Entre autres, il a été montré que tout algorithme capable d'apprendre à partir de ces requêtes est capable d'apprendre en présence de bruit dans le modèle PAC (Valiant, 1984). En pratique, certains algorithmes ont été modifiés pour tenir compte de la présence de bruit lors de l'apprentissage. C'est le cas du perceptron dont la variante fondée sur le vote (*voting perceptron*, Freund et Schapire (1998)) a été montrée plus robuste au bruit que sa forme classique par Khardon et Wachman (2007), des arbres de décision (Sakakibara, 1993) ou encore du boosting (BrownBoost, Freund (1999)) dont la version originale est particulièrement sensible au bruit.

Il est intéressant de remarquer le niveau de bruit que ces méthodes sont capables de tolérer. Sur des données réelles, il est rare que le bruit dans les données dépasse un seuil raisonnable. Néanmoins, ces stratégies sont testées pour des niveaux de bruit importants. Les expérimenta-

tions menées par Brodley et Friedl (1999) montrent par exemple que les ensembles permettent de filtrer entre $20\%$ et $30\%$ du bruit suivant les données utilisées.

## 6.2 Proposition

En pratique, les méthodes filtres sont les plus souvent appliquées pour leur facilité d'implémentation. Elles soulèvent néanmoins deux problèmes. D'abord, elles réduisent l'ensemble d'apprentissage en retirant les données qu'elles considèrent comme bruitées. Ensuite, leur précision ne leur permet généralement pas de faire la différence entre des exceptions (c'est à dire des exemples dont l'étiquette est surprenante compte-tenu de la région de l'espace où ils se trouvent) et des exemples réellement bruités (Brodley et Friedl, 1999). Dans ce même travail, Brodley et Friedl comparent les résultats obtenus par le vote à la majorité de trois classifieurs (1-NN, machine linéaire, arbre de décision) à ceux obtenus par ce même vote mais précédé d'un filtre (constitué d'un vote à la majorité ou au consensus sur le même ensemble) et concluent que le vote à la majorité ne peut pas se substituer au filtrage des données pour gérer le bruit, en particulier pour les niveaux les plus élevés. Dans cette expérience, l'ensemble est appris deux fois. La deuxième fois, les données difficiles ont été retirées de l'apprentissage par le biais du filtrage. Comme les classifieurs utilisent la variable de classe pour construire leur modèle de prédiction, celui-ci s'améliore grâce au filtrage des données. Le nombre d'exemples ayant servi à leur apprentissage est néanmoins fortement réduit par ce filtrage.

Dans ce contexte, les cartes auto-organisatrices ne souffrent pas des étiquettes bruitées lors de la construction du modèle. Tous les exemples – bruités ou non – servent à réaliser la projection des cartes sans induire une erreur supplémentaire. Les étiquettes des exemples n'interviennent qu'à l'étiquetage des neurones dont la position a été déterminée indépendamment de l'étiquette. De cette manière, un comité de cartes – en particulier dans sa forme non-supervisée, décrite en section 4.1 – gère la présence d'étiquettes erronées à deux niveaux. Au niveau local de chaque carte, les neurones sont étiquetés avec la classe majoritaire parmi les exemples que le neurone représente. Un exemple bruité n'aura un impact sur la prédiction d'une carte que s'il entraîne le changement de la classe majoritaire. Au niveau global du comité, l'étiquette d'un nouvel exemple est encore déterminée par un vote majoritaire, celui des cartes qui composent le comité. Là encore, une carte dont l'étiquette est biaisée par le bruit présent dans les données ne modifie la prédiction que dans le cas où cette carte fait basculer la majorité d'une classe vers l'autre. Ces deux mécanismes de vote à la majorité contrôlent l'impact du bruit sur l'apprentissage en limitant son effet aux frontières de décision entre les classes où un exemple bruité peut plus facilement emporter la décision lors de l'étiquetage des neurones.

Contrairement aux méthodes filtres, toutes les données sont prises en compte lors de l'apprentissage du comité de cartes, ce qui permet d'une part d'assurer un meilleur apprentissage et d'autre part de ne pas considérer systématiquement des exceptions comme des données bruitées. Qui plus est, la pertinence du vote à la majorité dépend également de la diversité des votants (si les votants sont biaisés de la même manière par les données bruitées, leur vote n'apportera aucune amélioration). Or, à la différence de l'approche filtre proposée par Brodley et Friedl (1999) qui utilise simplement 3 classifieurs sans se soucier de leur diversité, le comité de cartes veille à optimiser cette diversité en sélectionnant les sous-espaces d'apprentissage de chaque carte.

### 6.3 Application au bruit

Pour tester la robustesse de notre approche face aux données bruitées, nous avons utilisé les 4 premiers jeux de données présentés dans le tableau 2 en section 4.3. Sur chaque jeu, un bruit uniforme sur la classe a été ajouté avec une probabilité variant entre $5$ et $35\%$ ($60\%$ pour les données optdigits). Nous avons ainsi obtenu des données contenant deux variables de classes une pour la classe originale et l'autre pour la classe bruitée. Les données ont été apprises sur la classe bruitée avec un comité de cartes (dans sa forme non-supervisée), des forêts aléatoires et du boosting. Les tests ont été réalisés en comparant la prédiction de ces algorithmes avec la classe originale. De cette manière, l'impact du bruit sur la prédiction a pu être apprécié. Les résultats sont reportés dans la figure 5.

Comme le montrent les expérimentations, le comité de cartes est la stratégie qui supporte le mieux l'ajout de bruit sur la classe. Sur les 4 jeux de données, le bruit est toléré par les comités de cartes au minimum jusqu'à un seuil de $20\%$. Pour les taux de bruit inférieurs à ce seuil, la prédiction du comité varie très faiblement, contrairement aux forêts aléatoires dont la prédiction se dégrade plus rapidement ou au boosting qui réalise un sur-apprentissage dès lors que les données sont bruitées.

Ces différences s'expliquent par le degré d'implication de la classe au cours de l'apprentissage. Pour les comités, la classe n'intervient pas lors de la construction du modèle mais seulement lors de la prédiction. En revanche, pour les forêts aléatoires, la classe intervient lors de l'apprentissage de chaque arbre sans pour autant intervenir lors du choix de ces arbres. Pour le boosting, enfin, la classe intervient à chaque étape de l'apprentissage.

## 7 Conclusion

Cet article présente une méthode de représentation des données pour l'apprentissage supervisé de données volumineuses en grandes dimensions. L'intérêt d'une représentation en apprentissage supervisé concerne l'exploration intelligente des données, l'aide à la décision mais également la robustesse face à des données bruitées sur la variable de classe. Néanmoins, et particulièrement dans le contexte de données en grandes dimensions, il est difficile d'obtenir une représentation de qualité.

Pour faire face à ce problème, nous avons employé une stratégie en deux étapes. La première étape consiste à regrouper des cartes auto-organisatrices en un comité afin d'améliorer leurs capacités prédictives. Deux stratégies, non-supervisée et supervisée, sont détaillées dans l'article pour sélectionner les cartes à intégrer au comité. La stratégie supervisée donne de meilleurs résultats en prédiction grâce à une optimisation plus fine de l'erreur du comité. Néanmoins, la stratégie non supervisée présente l'intérêt d'être indépendante de la classe, ce qui s'avère particulièrement avantageux dans le cas de données bruitées. Le comité engendre une multitude de représentations de qualité diverse. Pour fournir à l'utilisateur une représentation unique dont la qualité est contrôlée, le comité est synthétisé à l'aide d'une carte auto-organisatrice par un *stacking* géographique. Cette carte a une qualité égale à ce que l'on aurait pu obtenir sur l'ensemble des attributs mais elle a l'avantage de représenter l'apprentissage réalisé par le comité sur lequel se fonde la prédiction. En outre, elle peut utiliser ce comité pour améliorer la navigation au travers des exemples. Enfin, face à des données bruitées, les comités de cartes se comportent de manière très robuste en tirant profit de leur représentation des

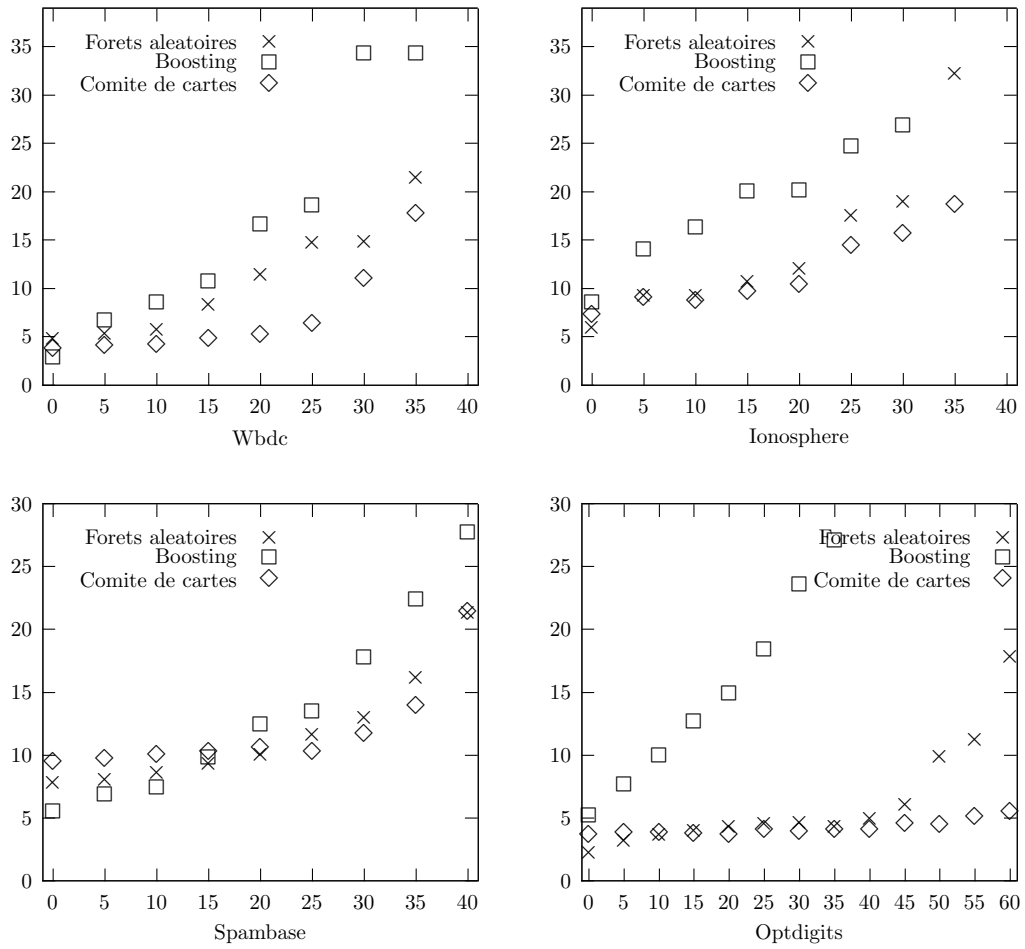


FIG. 5 – *De gauche à droite et de haut en bas les jeux wdbc, ionosphère, spambase et optdigits. Sur chaque figure est reporté en abscisse le niveau de bruit (en %) et en ordonnée le taux d'erreur (en % également)*

données. Dans le contexte de données très bruitées, leur prédiction s'avère plus performante que les forêts aléatoires.

L'apprentissage à partir de représentations trouve également une application dans le contexte des données semi-supervisées. Dans ce contexte, l'étiquette de classe est seulement connue pour une petite partie des exemples. Le problème consiste alors à prendre en compte les exemples non-étiquetés pour améliorer la prédiction. Comme la représentation n'a pas besoin de l'étiquette, elle tire mécaniquement parti de tous les exemples, étiquetés ou non (Chapelle et al., 2006; Zhu, 2005). Nous envisageons donc d'adapter notre approche à la problématique de l'apprentissage semi-supervisé en testant sa capacité à transmettre l'étiquette de classe aux exemples non-étiquetés. Par ailleurs, nous souhaitons modifier le comité de cartes pour qu'il exploite sa capacité de représentation dans le but de sélectionner les exemples à faire étiqueter par un expert et qu'il profite de sa résistance au bruit pour filtrer les erreurs lors du transfert des étiquettes aux exemples non-étiquetés.

# Références


Berthelsen, H. et B. Megyesi (2000). Ensemble of classifiers for noise detection in PoS tagged corpora. In *TSD*, pp. 27–32.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Brodley, C. E. et M. A. Friedl (1996). Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pp. 799–805.

Brodley, C. E. et M. A. Friedl (1999). Identifying mislabeled training data. In *Journal of Artificial Intelligence Research*, Volume 11, pp. 131–167.

Brown, G. (2005). Managing diversity in regression ensembles. *Journal of Machine Learning Research 6*, 1621–1650.

Chapelle, O., B. Schölkopf, et A. Zien (Eds.) (2006). *Semi-Supervised Learning*. MIT Press.

Clech, J. (2004). *Contribution méthodologique à la fouille de données complexes*. Thèse de doctorat en informatique, Université Lumière Lyon 2, Lyon : France.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation 10*(7), 1895–1924.

Donoho, D. (2000). High-dimensional data analysis : The curses and blessings of dimensionality.

Freund, Y. (1999). An adaptive version of the boost by majority algorithm. In *Computational Learing Theory*, New York, NY, USA, pp. 102–113. ACM Press.

Freund, Y. et R. Schapire (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pp. 23–37.

Freund, Y. et R. Schapire (1998). Large margin classification using the perceptron algorithm. In *Computational Learing Theory*, pp. 209–217.

Geman, S., E. Bienenstock, et R. Doursat (1992). Neural networks and the bias/variance dilemma. *Neural Computing 4*(1), 1–58.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.

Guerif, S. et Y. Bennani (2007). Sélection de variables en apprentissage numérique non supervisé. In *Cap'07 : conférence francophone sur l'apprentissage automatique*, Grenoble : France.

Guérif, S., Y. Bennani, et C. Baudoin (2006). Connectionist and Ethological Approaches for Discovering Salient Facial Movement Features in Human Gender Recognition. In *Proceeding of the 28th International Conference Information Technology Interfaces*, Cavtat, Croatia, pp. 189–194.

Hacid, H. (2004). *Un environnement informatique pour l'interrogation et l'accès intelligent aux bases de données complexes.* Thèse de doctorat en informatique, Université Lumière Lyon 2, Lyon : France.

Kaiser, H. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika 23*, 187–200.

Kearns, M. (1993). Efficient noise-tolerant learning from statistical queries. In *Proc. of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pp. 392–401.

Khardon, R. et G. Wachman (2007). Noise tolerant variants of the perceptron algorithm. *J. Mach. Learn. Res. 8*, 227–248.

Kohonen, T. (1982). Self-organization of topogically correct feature maps. *Biological Cybernetics 43*, 59–69.

Kohonen, T. (1988). Learning vector quantization. *Neural Network 1*, 303.

Kohonen, T. (2001). *Self-Organizing Maps*, Volume 30 of *Springer Series in Information Sciences*. Berlin.

Krogh, A. et J. Vedelsby (1995a). Neural network ensembles, cross validation, and active learning. In *Advances in NIPS*, Volume 7, pp. 231–238.

Krogh, A. et J. Vedelsby (1995b). Neural network ensembles, cross validation, and active learning. In G. Tesauro, D. Touretzky, et T. Leen (Eds.), *Advances in Neural Information Processing Systems*, Volume 7, pp. 231–238. The MIT Press.

Kuncheva, L. et C. Whitaker (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning 51*(2), 181–207.

Lallich, S. (2002). *Mesure et validation en extraction des connaissances à partir des données*. Habilitation à diriger les recherches, Université Lumière Lyon 2, Lyon : France.

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 1, pp. 281–297.

Midenet, S. et A. Grumbach (1994). Learning associations by self-organisation : the lasso model. *Neurocomputing 6*, 343–361.

Muhlenbach, F. (2002). *Évaluation de la qualité de la représentation en fouille de données*. Thèse de doctorat en informatique, Université Lumière Lyon 2, Lyon : France.

Newman, D., S. Hettich, C. Blake, et C. Merz (1998). UCI repository of machine learning databases, http ://www.ics.uci.edu/∼mlearn/mlrepository.html.

Newman, L. et T. Polk (2007). The emergence of semantic topography in a neurally-inspired computational model. In *Proceedings of 8th International Conference on Cognitive Modeling*.

Prudhomme, E. et S. Lallich (2005). Validation statistique des cartes de Kohonen en apprentissage supervisé. In *Actes de EGC'2005*, Volume 1 of *RNTI-E-3*, pp. 79–90.

Quinlan, J. R. (1990). Induction of decision trees. In J. W. Shavlik et T. G. Dietterich (Eds.), *Readings in Machine Learning*. Morgan Kaufmann. Originally published in *Machine Learning* 1 :81–106, 1986.

Rakotomalala, R. (2005). Tanagra : un logiciel gratuit pour l'enseignement et la recherche. In *Actes de EGC'2005*, Volume 2 of *RNTI-E-3*, pp. 697–702.

Ritschard, G., D.-A. Zighed, et S. Marcellin (2007). Données déséquilibrées, entropie décentrée et indice d'implication. In *Nouveaux apports théoriques à l'analyse statistique implicative et applications*, pp. 315–327. Departament de Matematiques, Universitat Jaume I.

Sakakibara, Y. (1993). Noise-tolerant Occam algorithms and their applications to learning decision trees. *Machine Learning 11*(1), 37–62.

SAS (1989). *SAS/STAT user's guide, Version 6, Fourth Edition*, Volume 2. SAS Institute Inc.

Sebban, M. (1996). *Modèles théoriques en reconnaissance des formes et architecture hybride pour machine perceptive*. Thèse de doctorat en informatique, Université Lumière Lyon 2, Lyon : France.

Song, X.-H. et P. K. Hopke (1996). Kohonen neural network as a pattern recognition method based on the weight interpretation. *Analytica Chimica Acta 334*, 57–66.

Toussaint, G. T. et R. Menard (1980). Fast algorithms for computing the planar relative neighborhood graph. In *Methods of Operations Research, Proceedings of the Fifth Symposium on Operations Research*, pp. 425–428.

Tumer, K. et J. Gosh (1995). Theoretical foundations of linear and order statistics combiners for neural pattern classifiers. Technical report, Computer and Vision Research Center, University of Texas, Austin.

Ueda, N. et R. Nakano (1996). Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN96)*, Volume 1, pp. 90–95.

Ultsch, A. et L. Herrmann (2005). The architecture of emergent self-organizing maps to reduce projection errors. In *European Symposium on Artificial Neural Network*, pp. 1–6.

Valiant, L. G. (1984). A theory of the learnable. *Commun. ACM 27*(11), 1134–1142.

Verbaeten, S. et A. Van Assche (2003). Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems*, pp. 317–325. Springer.

Vesanto, J. (1999). Som-based data visualization methods. *Intelligent Data Analysis 3*, 111–126.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association 58*(301), 236–244.

Wilson, D. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. on Systems, Man and Cybernetics 2*, 408–421.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks 5*(2), 241–259.

Zanda, M., G. Brown, G. Fumera, et F. Roli (2007). Ensemble learning in linearly combined classifiers via negative correlation. In *International Workshop on Multiple Classifier Systems*.

Zhu, X. (2005). Semi-supervised learning literature survey. Technical report.

Zighed, D. A., S. Lallich, et F. Muhlenbach (2004). A statistical approach of classes separability. In T. Elomaa, H. Mannila, et H. Toivonen (Eds.), *Applied Stochastic Models in Business and Industry*, pp. 475–487. Springer-Verlag.

Zupan, J., M. Novic, X. Li, et J. Gasteiger (1994). Classification of multicomponent analytical data of olive oils using different neural networks. *Analytica Chimica Acta 292*, 219–234.

## Summary

In supervised learning, ensemble methods, like boosting or random forests, allow a very efficient prediction while keeping a low algorithmic complexity. However, that kind of algorithms are able to predict new examples but not to give any information on that prediction. Another solution is to associate the prediction with a data representation to allow the end-user to take part of this prediction. As a visualization tool, the representation could be used to assess the model validity and as a navigation tool, the representation help the user to understand the prediction of an example thanks to the examples used for it. Moreover, because the representation is learned independantly from the class label, it is robust to class noise. Nevertheless, computing the representation of high-dimensional data is time-consuming, often for a poor result in prediction. In that context, we propose an ensemble of self-organizing maps which is synthesized by another map, learned on a stacking of the neurons coordinates. Maps ensemble uses the diversity concept to ensure an efficient prediction while the geographical stacking give a representation for the end-user. Experimentations show the effectiveness of this strategy for the prediction and the representation of the data, even if these data are noisy.