Chapitre 3

Comités de cartes

Sommaire

1 Introduction aux méthodes ensemblistes 82
1.1 Notion de diversité 84
1.2 Décomposition de l'erreur d'un ensemble 84
2 Typologie des méthodes ensemblistes 88
2.1 Méthodes hétérogènes 88
2.2 Méthodes homogènes 89
2.3 Gestion implicite de la diversité 90
2.4 Gestion explicite de la diversité 92
3 Récapitulatif des méthodes ensemblistes 93
4 Comités de cartes 94
4.1 Gestion de la diversité 95
4.2 Intégration par stacking géographique 99
4.3 Étude de la complexité 101
5 Expérimentations 102
5.1 Apprentissage supervisé 102
5.2 Contextualisation et navigation 104
6 Résistance au bruit 107
6.1 Le bruit en apprentissage 107
6.2 Intérêt du comité de cartes 108
6.3 Expérimentations 109
7 L'apprentissage semi-supervisé 110
7.1 Le problème de l'apprentissage semi-supervisé 110
7.2 Comité de cartes et apprentissage semi-supervisé 113
7.3 Expérimentations 114
8 Discussion 116

Comités de cartes

Publications

Prudhomme, E. et S. Lallich (2008). Maps ensemble for semi-supervised learning of large high dimensional datasets. In *Proc. of the 17th International Symposium on Methodologies for Intelligent Systems*, Toronto, Canada, pp 100–110. LNAI Springer-Verlag.

Prudhomme, E. et S. Lallich (2008). Optimization of self-organizing maps ensemble in prediction. In *Proc. of International Conference on Data Mining (DMIN'08)*, Las Vegas, Nevada.

Lallich S. et E. Prudhomme (2008), Représentation des données et apprentissage supervisé. Journées thématiques *Apprentissage Artificiel & Fouille de Données, AAFD'08*, Université Paris 13, 7 et 8 avril 2008.

Prudhomme, E. et S. Lallich (2007). Ensemble prédicteur fondé sur les cartes auto-organisatrices adaptées aux données volumineuses. In *Extraction des connaissances et apprentissage (Actes de EGC'2007)*, Paris, France, *RNTI-E-3*, pp. 473–484. Cépaduès Editions.

Ce chapitre traite de la mise en œuvre d'un comité de cartes pour la prédiction de données volumineuses. Pour justifier l'emploi en prédiction de plusieurs cartes regroupées en comité, nous avons d'abord étudié les méthodes ensemblistes. Sous ce terme sont généralement regroupées toutes les méthodes dont la prédiction repose sur l'apprentissage de plusieurs classifieurs, comme le boosting, les forêts aléatoires ou encore les méthodes fondées sur les codes correcteurs d'erreurs (dans cette thèse, nous utiliserons les termes de méthodes ensemblistes ou ensembles en référence à cette famille de méthodes et le terme de comité pour nommer les ensembles fondés sur la représentation). Les performances de cette famille de méthodes ont entraîné de nombreuses études tant théoriques que pratiques pour déterminer les règles qui régissent l'obtention d'un ensemble de classifieurs performant.

Dans la section 1, nous présentons les raisons qui ont conduit à la mise en œuvre des ensembles puis nous étudions les principes qui sous-tendent leur succès. Dans la section 2, nous distinguons les différents algorithmes présents dans la littérature qui tirent profit de ces principes. Dans les sections qui suivent, nous introduisons nos contributions. Dans la section 4, nous proposons de bénéficier des avantages des méthodes ensemblistes pour réaliser un apprentissage fondé sur la représentation qui conserve de bonnes qualités prédictives. En pratique, nous instancions les stratégies ensemblistes pour obtenir un comité de cartes.

Nous détaillons deux heuristiques pour l'obtention d'un tel comité. La première de ces heuristiques ne tient jamais compte de l'étiquette de classe lors de l'apprentissage (section 4.1.1) alors que la deuxième exploite cette information pour choisir les représentations qui fondent le comité (section 4.1.2). Une fois le comité obtenu, nous synthétisons l'information portée par les cartes du comité en une carte unique, à l'aide d'un *stacking géographique* (section 4.2). Dans la section 5, nous menons plusieurs expériences sur les comités de cartes. Ces expériences nous permettent de valider les stratégies de sélection des cartes du comité ainsi que leur performance par rapport à d'autres méthodes ensemblistes. Nous envisageons deux applications du comité de cartes. La première, présentée en section 6, évalue la résistance au bruit sur la classe du comité alors que la seconde s'intéresse à la capacité d'un comité de cartes à répondre aux problématiques de l'apprentissage semi-supervisé (section 7).

1 Introduction aux méthodes ensemblistes

L'idée de combiner plusieurs classifieurs en apprentissage supervisé a émergé dans différents contextes. En sciences cognitives, d'abord, les ensembles ont servi à la modélisation de certains processus. Jacobs et al. (1991) ont ainsi mis en place une architecture de deux réseaux de neurones en compétition pour rendre compte de la spécialisation des deux voies neuronales (« quoi » et « où ») impliquées dans le traitement de l'information visuelle. Ils ont dégagé plusieurs avantages à cette architecture modulaire, parmi lesquels la rapidité d'apprentissage, une meilleure généralisation et une complexité plus faible des différents réseaux qui facilite leur interprétation (chaque réseau étant dédié à une tâche particulière).

Dans le domaine de l'apprentissage automatique, l'intérêt de combiner plusieurs classifieurs est d'abord apparu au cours de recherches théoriques dans le cadre du modèle d'apprentissage Probablement Approximativement Correct (PAC, Valiant (1984)). Dans ce modèle, une classe de concepts est dite apprenable au sens fort, si il existe un algorithme qui permet de prédire chaque concept avec une erreur faible et une forte confiance en un temps polynomial. Kearns (1988) ajoute à ce modèle la notion de classes de concepts apprenables au sens faible pour lesquelles il suffit qu'un algorithme prédise chaque concept avec une erreur légèrement inférieure au hasard. L'intérêt de cette notion réside dans la question : est-ce que l'apprenabilité au sens faible est équivalente à l'apprenabilité au sens fort ? Si la réponse est affirmative, il suffit de connaître un algorithme d'apprentissage au sens faible pour réaliser un apprentissage au sens fort. Le travail de Schapire (1990) démontre cette équivalence. Schapire (1990) en tire la méthode du *boosting* qui consiste à apprendre plusieurs algorithmes d'apprentissage faibles sur des distributions différentes des exemples (nous détaillons la version AdaBoost de cet algorithme dans la section 2.3).

Toujours en apprentissage automatique, mais du point de vue des données cette fois,

certains auteurs ont été confrontés à des données naturellement divisées en plusieurs sous-ensembles d'attributs, chacun spécifique d'une série particuliere de capteurs. Pour s'adapter à ces données, plusieurs travaux ont montré que l'apprentissage de chaque sous-ensemble d'attributs par un classifieur spécialisé puis la combinaison des résultats des différents classifieurs améliore la prédiction par rapport à l'apprentissage d'un seul classifieur sur la globalité des attributs (Sharkey et Sharkey, 1997; Ahmad et al., 2003).

Finalement, du point de vue de l'optimisation, Dietterich (2000) distingue trois raisons pour lesquelles il est possible d'améliorer un classifieur seul par une méthode ensembliste. La première, d'ordre statistique, est reliée à la malédiction de la dimension (voir chap. 1 sect. 4.1.1). Dans beaucoup de cas, le nombre d'exemples est insuffisant pour caractériser avec précision l'espace d'apprentissage, ce qui empêche les algorithmes d'apprentissage de trouver la fonction de prédiction optimale. En revanche, plusieurs algorithmes sont capables de trouver des fonctions différentes de performances équivalentes dont la combinaison se rapproche de la fonction optimale. La deuxième raison est d'ordre computationnel. Si de nombreux algorithmes sont en théorie capable d'approximer n'importe quelle fonction, en pratique, ils sont souvent confrontés à la découverte de minima locaux. Là encore, la combinaison de plusieurs classifieurs permet de se rapprocher de l'optimum global. Enfin, la dernière raison est dite représentationnelle. Chaque classifieur effectue une recherche de la solution dans un sous-ensemble fini des fonctions possibles, sous-ensemble qui dépend de la nature du classifieur et des données d'apprentissage. Or, la solution optimale du problème peut se trouver en dehors des frontières de ce sous-ensemble, ce qui ne pourra être exploré que par une combinaison de classifieurs.

Pour comprendre pourquoi une décision prise par un ensemble de classifieurs est de nature à améliorer la décision de chaque classifieur pris individuellement, nous allons prendre l'exemple du jury de Condorcet (Condorcet, 1785, exemple que nous avons détaillé dans Prudhomme et Lallich (2009)). 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 1 indique le risque d'erreur du jury en fonction du nombre de juges et du risque d'erreur p commun aux différents juges. 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 1.

Illustrons ce dernier point. Avec 15 juges indépendants ayant chacun un risque d'erreur de 40%, d'après le tableau 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 les 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 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.

1.1 Notion de diversité

Comme le montre l'expérience du jury de Condorcet, il est nécessaire que les juges ne commettent pas leurs erreurs sur les mêmes cas pour que la décision finale du jury soit améliorée. En apprentissage, cette aptitude d'un ensemble de classifieurs à avoir des points de vue différents a été étudiée sous le terme de diversité. À l'instar de l'indépendance pour le jury de Condorcet, de nombreux travaux ont cherché à relier une mesure de la diversité d'un ensemble à son taux d'erreur en généralisation. Des mesures statistiques sont ainsi apparues, dont on trouve une revue détaillée dans Kuncheva et Whitaker (2001). Malheureusement, même s'il existe un certain lien entre l'erreur de l'ensemble et la diversité des classifieurs qui le composent, aucune corrélation entre ces deux grandeurs n'a pu être clairement établie (Kuncheva et Whitaker, 2003). Afin d'établir plus précisément la place de la diversité dans l'erreur de l'ensemble, d'autres travaux ont décomposé l'erreur d'un ensemble en fonction de ces classifieurs. Nous les présentons dans la section suivante.

1.2 Décomposition de l'erreur d'un ensemble

Les premiers travaux dans ce domaine ont porté sur la décomposition de l'erreur d'un ensemble en régression. Ueda et Nakano (1996) obtiennent ainsi une décomposition fonction du biais et de la variance. Si l'on note f^m la prédiction du m^{e} apprenant et r la valeur à prédire, ils montrent que l'erreur quadratique moyenne d'un ensemble de M apprenants en

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
33	0,00	0,00	0,00	0,01	0,13	0,50	0,87	0,99	1,00	1,00	1,00
35	0,00	0,00	0,00	0,01	0,12	0,50	0,88	0,99	1,00	1,00	1,00
37	0,00	0,00	0,00	0,01	0,11	0,50	0,89	0,99	1,00	1,00	1,00
39	0,00	0,00	0,00	0,00	0,10	0,50	0,90	1,00	1,00	1,00	1,00
41	0,00	0,00	0,00	0,00	0,10	0,50	0,90	1,00	1,00	1,00	1,00
43	0,00	0,00	0,00	0,00	0,09	0,50	0,91	1,00	1,00	1,00	1,00
45	0,00	0,00	0,00	0,00	0,09	0,50	0,91	1,00	1,00	1,00	1,00

TAB. 3.1: 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)

régression s'écrit :

$$E\left\{\left(\frac{\sum_m f^m}{M} - r\right)^2\right\} = \overline{bias}^2 + \frac{1}{M}\overline{var} + \left(1 - \frac{1}{M}\right)\overline{covar} \ . \tag{3.1}$$

avec $\overline{bias}$, $\overline{var}$ et $\overline{covar}$ respectivement le biais, la variance et la covariance moyenne des apprenants de l'ensemble. À la différence d'une décomposition classique en biais/variance de l'erreur d'un classifieur seul (Geman et al., 1992), cette décomposition introduit un terme de covariance. Ce terme quantifie la corrélation qui existe entre les prédictions des apprenants de l'ensemble. Comme cette corrélation peut être négative, ce terme est de nature à réduire l'erreur finale de l'ensemble sans que le biais ou la variance des classifieurs de l'ensemble ne soit amélioré. En régression, la covariance est donc une expression de la diversité. Outre cette covariance, le nombre de classifieurs de l'ensemble joue également un rôle important dans la réduction de l'erreur. D'une part, il réduit la variance et, d'autre part, il augmente l'importance de la covariance.

Toujours en régression, Krogh et Vedelsby (1995) ont obtenu une décomposition différente de l'erreur d'un ensemble :

$$\left(\frac{\sum_m f^m}{M} - d\right)^2 = \sum_m^M (f^m - r)^2 - \sum_m^M \left(f^m - \frac{\sum_m f^m}{M}\right)^2 \ . \tag{3.2}$$

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 3.1, Brown (2005) montre que ce deuxième terme contient à la fois une part de la variance et de la covariance moyenne de l'erreur de l'ensemble.

Ces deux décompositions obtenues en régression montrent que l'erreur d'un ensemble dépend de 3 paramètres : le nombre de classifieurs, la précision de chaque classifieur et leur covariance. Ces travaux expliquent ainsi pourquoi il n'y a pas de corrélation claire entre la diversité des classifieurs 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 3.1). 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 formel, Tumer et Gosh (1995) ont montré que l'erreur ajoutée par un ensemble s'écrit :

$$E_{add}^{ens} = \left(\frac{1 + \delta \times (M - 1)}{M} \right) E_{add} \ . \tag{3.3}$$

où E_{add} représente l'erreur ajoutée d'un apprenant et δ la corrélation entre les erreurs d'approximation des probabilités *a posteriori* commises par les différents apprenants ($0 \leq \delta \leq M$). De même que pour la covariance, il s'agit là encore d'une mesure de la diversité. En effet, si $\delta = 0$, les erreurs des apprenants sont indépendantes les unes des autres, l'erreur de l'ensemble est $\frac{1}{M}$ fois l'erreur d'un apprenant. À l'inverse, 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.

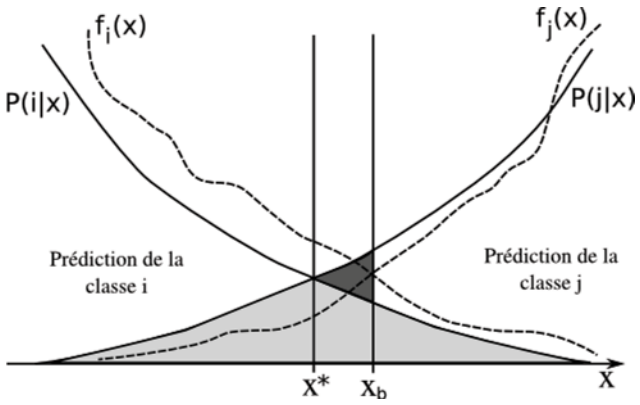


FIG. 3.1: Régions d'erreur associées aux approximations des probabilités *a posteriori* : en gris clair, l'erreur bayésienne irréductible et en gris foncé, l'erreur ajoutée par le classifieur (Tumer et Gosh, 1995).

À partir de cette formalisation, Zanda et al. (2007) remarquent que réduire l'erreur ajoutée revient soit à rapprocher x_b de x^*, soit à estimer l'écart entre les probabilités *a posteriori* (noté $r_{ij} = P(i|x) - P(j|x)$) à partir des estimations du classifieur $f_{ij} = f_i(x) - f_j(x)$. En effet, si on dispose d'un classifieur tel que $f_{ij} = r_{ij}$, 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 et on observe $x_b = x^*$.

Grâce à cette remarque, Zanda et al. (2007) ramènent un problème de prédiction d'une variable catégorielle à un problème de régression où la cible de la régression est r_{ij} et l'estimateur f_{ij}. Or, en regression, la décomposition de l'erreur est connue. Ainsi, en tirant parti des travaux antérieurs de Krogh et Vedelsby (1995) (eq. 3.2), ils expriment l'erreur

d'un ensemble en classification de la manière suivante :

$$E = \sum_{i=1}^{c} \sum_{j>i} (\bar{f}_{ij} - r_{ij})^2 \tag{3.4}$$

$$= \sum_{i=1}^{c} \sum_{j>i} \left[\frac{1}{M} \sum_{m=1}^{M} (f_{ij}^m - r_{ij})^2 \right] - \sum_{i=1}^{c} \sum_{j>i} \left[\frac{1}{M} \sum_{m=1}^{M} (\bar{f}_{ij} - f_{ij}^m)^2 \right] \tag{3.5}$$

$$= \bar{E} - A \; . \tag{3.6}$$

avec $f_i^m(x)$ l'estimation de $P(i|x)$ pour le m^{e} 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 $\bar{f}_{ij} = \bar{f}_i(x) - \bar{f}_j(x)$.

L'erreur est donc décomposée en deux termes. Le premier, $\bar{E}$, correspond à l'erreur moyenne des classifieurs alors que le deuxième, A, est équivalent au terme d'ambiguïté dans les travaux de Krogh et Vedelsby (1995). Ce terme contient la diversité des réponses des classifieurs. Ainsi, pour une erreur moyenne des classifieurs constante et un nombre de classifieurs donné, il est possible de réduire l'erreur de l'ensemble en faisant varier le terme d'ambiguïté.

Cette section a montré que les méthodes ensemblistes, pour être efficaces, doivent tenir compte de plusieurs paramètres qui sont le nombre de classifieurs agrégés, leur erreur moyenne et la diversité des prédictions de ces classifieurs. Dans la section suivante, nous présentons les différentes stratégies qui ont été proposées pour concevoir de tels ensembles.

2 Typologie des méthodes ensemblistes

Suivant la proposition de Suchier (2006), nous marquons une première distinction parmi les méthodes ensemblistes suivant que les classifieurs sont hétérogènes ou non.

2.1 Méthodes hétérogènes

Les méthodes ensemblistes hétérogènes sont composées de classifieurs de nature différente. Une solution immédiate pour construire ce type d'ensemble consiste à apprendre puis à combiner par vote plusieurs algorithmes de classification. Dans ce cas, la diversité de l'ensemble provient des différences de nature entre les hypothèses apprises par chaque algorithme. Cette première approche a été améliorée par deux propositions, le *stacking* et les généralisations en cascade (*cascade generalization*).

Pour améliorer la généralisation obtenue par un ensemble de classifieurs hétérogènes, Wolpert (1992) propose d'opérer à deux niveaux d'apprentissage. L'objectif du premier niveau (niveau 1) est d'obtenir un nouvel espace de représentation des données (ou espace

des généralisations) à l'aide d'un ensemble de classifieurs. Chaque attribut de cet espace des généralisations est déterminé à partir des généralisations d'un des classifieurs de l'ensemble. Pour transcrire un exemple x_i dans ce nouvel espace, l'ensemble des classifieurs est appris sur les exemples d'apprentissage, x_i mis à part. Ce dernier est prédit par les classifieurs de l'ensemble de sorte que ces prédictions constituent la représentation de x_i dans l'espace des généralisations. Pour obtenir les données à apprendre au niveau 2, l'ensemble des classifieurs doit donc être appris autant de fois qu'il y a d'exemples. Une fois les données obtenues dans l'espace des généralisations, elles sont apprises par un dernier classifieur. Pour prédire un nouvel exemple, celui-ci est d'abord présenté aux classifieurs de niveau 1, puis le résultat de leur prédiction est utilisée par le classifieur du niveau 2 pour obtenir la prédiction finale.

Le stacking consiste ainsi à intégrer comme information à l'apprentissage, les généralisations des classifieurs d'un ensemble. Avec la méthode des généralisations en cascade, Gama et Brazdil (2000) proposent de tenir compte différemment de cette information en concevant non plus seulement deux niveaux d'apprentissage mais k niveaux. À chaque niveau, un seul classifieur apprend les exemples (à la différence du stacking – et contrairement à ce qu'indique le nom de la méthode – tous les exemples sont appris, ce qui supprime la notion de généralisation). L'apprentissage au niveau suivant est réalisé dans l'espace d'apprentissage initial augmenté des prédictions du classifieur du niveau précédent. Contrairement au stacking, les prédictions sont exprimées sous la forme des probabilités *a posteriori* de prédire chaque classe, ce qui ajoute autant d'attributs continus qu'il y a de classe à prédire (dans une étude sur le stacking, Ting et Witten (1999) ont montré que les probabilités *a posteriori* remplaçaient avantageusement la prédiction de la classe).

Pour introduire de la diversité au sein d'un ensemble hétérogène, Gama et Brazdil (2000) conseillent d'agréger des classifieurs qui se comportent différemment du point de vue de la décomposition en biais-variance de leur erreur. En particulier, pour les généralisations en cascade, ils conseillent d'utiliser des classifieurs avec un petit biais au niveau 1 et avec une variance faible au niveau 2.

2.2 Méthodes homogènes

Les méthodes ensemblistes homogènes apprennent les différents classifieurs à partir d'une même famille d'hypothèses. Dans ces cas, deux solutions ont été envisagées pour assurer la diversité de leurs prédictions. La première solution consiste à formuler différemment le problème à apprendre en manipulant les exemples, les attributs ou les classes. Dans ce cas, la diversité des classifieurs est gérée de manière implicite. Une autre solution consiste à gérer explicitement la diversité soit lors de l'apprentissage des classifieurs, soit pour sélectionner un ensemble de classifieurs performant.

2.3 Gestion implicite de la diversité

Pour produire de la diversité, le *bagging* (pour *Bootstrap AGGregatING*) (Breiman, 1996) ré-échantillonne les exemples à apprendre pour chaque classifieur de l'ensemble. L'échantillonnage est réalisé par un tirage *bootstrap* (tirage aléatoire avec remise de même taille que l'échantillon de départ) des exemples d'origine. La diversité des prédictions est générée par cet échantillonnage seulement. Pour qu'il soit efficace, il est nécessaire que les classifieurs soient instables, c'est-à-dire que de petites variations dans les données les conduisent à produire des hypothèses différentes (par exemple, les arbres de décisions ou les réseaux de neurones sont des classifieurs instables alors que les k-Plus Proches Voisins (k-PPV) sont au contraire réputés pour être stables). Si tel est le cas, le *bagging* va réduire la variance de l'erreur et assurer à l'ensemble de classifieurs une erreur moins importante que celle du classifieur (instable) seul.

Le *boosting* (algorithme AdaBoost, Freund et Schapire, 1996) utilise également des échantillons *bootstrap* des exemples pour apprendre les classifieurs de l'ensemble. Cependant, à la différence du *bagging*, le tirage des échantillons est pondéré au cours d'un processus itératif, de sorte que plus un exemple est mal prédit par les classifieurs précédents, plus sa probabilité d'être tiré au sort à l'itération courante est grande. En pratique, chaque classifieur appris à une itération t de l'algorithme est associé à un coefficient c_t, inversement proportionnel à l'erreur en apprentissage du classifieur. Le poids d'un exemple à l'itération $t+1$ est alors une fonction du poids de l'exemple à l'itération t et du coefficient c_t, coefficient pondéré négativement si l'exemple est bien classé à l'itération t, positivement sinon. Le classifieur final est la somme pondérée par c_t des classifieurs appris à chaque itération.

En pratique, le *boosting* donne d'excellents résultats. Au fur et à mesure des itérations, les erreurs en apprentissage et en généralisation diminuent, cette dernière pouvant continuer à diminuer alors même que l'erreur en apprentissage est déjà nulle. Autrement dit, dans certains cas, le boosting n'est pas soumis au problème du sur-apprentissage que l'on rencontre lorsqu'un classifieur apprend les données connues avec trop de précision au détriment des données à prédire (cas où la composante variance de l'erreur est élevée alors que le biais est faible). De nombreuses études ont cherché à borner l'erreur du boosting afin d'expliquer ce comportement, parmi lesquels Freund et Schapire (1995) sur l'erreur d'apprentissage et Schapire et al. (1998) sur l'erreur en généralisation. Enfin, notons que le boosting (dans sa version AdaBoost) n'améliore pas toujours l'erreur d'un classifieur seul. En présence de bruit sur la classe, en particulier, le boosting pondère fortement les exemples bruités, ce qui a comme conséquence une dégradation globale de l'apprentissage (voir la thèse de Suchier, 2006, pour des contributions dans ce domaine).

Les deux algorithmes que nous venons de présenter échantillonnent les exemples pour

générer de la diversité au sein de l'ensemble. Dans la méthode des forêts de décision (*decision forests*), Ho (1998) propose au contraire d'utiliser un échantillonnage des variables sur des classifieurs de type arbre de décision. Pour ce faire, Ho construit un ensemble d'arbres (ou encore une forêt) dans lequel l'apprentissage de chaque arbre repose sur un sous-espace aléatoire des attributs (dans ces expériences, Ho construit ce sous-espace avec la moitié des attributs).

Breiman (2001) améliore cette idée en l'associant au *bagging* pour donner la méthode des forêts aléatoires (*random forests*). Dans cette méthode, chaque arbre est appris sur un échantillon bootstrap des exemples. Par ailleurs, à la différence des forêts de décision, un sous-espace des attributs est tiré aléatoirement pour construire chaque nœud de l'arbre. Dans ces expériences, Breiman tire aléatoirement $\log_2 d$ attributs, avec d le nombre d'attributs. Breiman montre qu'avec cette stratégie les forêts aléatoires ne sont pas sujettes au sur-apprentissage lorsqu'on augmente le nombre d'arbres. Au contraire, elles convergent vers la minimisation de l'erreur. Par rapport au *boosting* (et pour une efficacité à peu près similaire), Breiman souligne les avantages suivants : elles sont plus rapides, facilement parallélisables et relativement peu sujettes au bruit et aux exemples atypiques.

Les méthodes à codes correcteurs d'erreurs (*Error-Correcting Output Codes*, ECOC Dietterich et Bakiri, 1991) génèrent la diversité en manipulant la variable de classe. L'idée consiste à diviser un problème à c classes en p problèmes à 2 classes créés artificiellement à partir d'un sous-ensemble disjoint des classes d'origine. Pour chaque nouveau problème sur lequel un classifieur est appris, un vecteur encode la partition des classes correspondantes. Par exemple, pour un problème à 4 classes (a, b, c, d), le vecteur (0, 0, 1, 1) encode un sous-problème qui oppose une première classe artificielle composée des exemples des classes a et b à une deuxième classe artificielle composée des exemples des classes c et d. En combinant ces vecteurs, il est possible d'obtenir le vecteur de chaque classe d'origine qui précise la correspondance avec les classes artificielles à chaque nouveau problème. Lors de la prédiction d'un nouvel exemple, les prédictions de chaque classifieur de l'ensemble constituent un vecteur comparable aux vecteurs des classes d'origine. La classe de l'exemple est ainsi déterminée par la classe dont le vecteur est le plus proche. La qualité de cette approche dépend du choix des sous-problèmes à traiter. Dans une étude sur le sujet, Windeatt et Ghaderi (2003) remarquent que des codes équidistants entre eux donnent souvent de meilleurs résultats. Ils trouvent également que la distance de Hamming est plus adaptée que d'autres mesures de distances (comme la distance euclidienne) pour comparer la prédiction des classifieurs au vecteur de classe.

Du point de vue de la décomposition de l'erreur, Kong et Dietterich (1995) (d'après Suchier, 2006) montrent que ces méthodes améliorent à la fois le biais et la variance. L'amélioration du biais repose sur des classifieurs très divers les uns des autres en termes

d'hypothèses grâce à la formulation de sous-problèmes très distincts. L'amélioration de la variance provient, quant à elle, de la transformation d'un problème à c classes en des sous-problèmes à deux classes.

2.4 Gestion explicite de la diversité

Pour gérer au mieux la diversité des classifieurs, certains travaux exploitent la décomposition de l'erreur d'un ensemble afin de trouver le meilleur compromis possible entre l'erreur moyenne des classifieurs et leur diversité. À ce titre, les travaux sur la corrélation négative sont une réussite. La corrélation négative est une stratégie initiée par Liu et Yao (1999) afin de gérer la diversité d'un ensemble de réseaux de neurones en régression. Dans leur forme standard, les réseaux de neurones apprennent en rétro-propageant l'erreur qu'ils commettent sur chaque exemple. Dans le cadre d'un ensemble de réseaux de neurones, Liu et Yao (1999) proposent d'associer au calcul de l'erreur à rétro-propager sur chaque réseau une pénalité fonction des erreurs commises par les autres réseaux de l'ensemble. Brown (2005) propose un cadre théorique à l'apprentissage par corrélation négative, toujours en régression, en le reliant aux travaux de Krogh et Vedelsby (1995) sur la décomposition de l'erreur d'un ensemble (voir section 1.2). En particulier, il remarque que le terme de pénalité proposé par Liu et Yao et associé au calcul de l'erreur, correspond en fait au terme d'ambiguïté de l'équation 3.2. Cette remarque lui permet de déterminer précisément l'erreur à rétro-propager sur chaque neurone pour optimiser l'erreur globale de l'ensemble. Enfin, les travaux de Zanda et al. (2007) sur la décomposition de l'erreur en classification (équation 3.4 présentée en section 1.2) permettent d'adapter l'apprentissage par corrélation négative à la prédiction d'une variable catégorielle. L'apprentissage par corrélation négative est ainsi une solution élégante et efficace pour optimiser l'erreur d'un ensemble de taille donné au cours de son apprentissage. Cette solution reste cependant très dépendante de l'apprentissage par rétro-propagation spécifique des réseaux de neurones.

Toujours sur des réseaux de neurones en régression, Zhou et al. (2001, 2002) proposent la méthode GASEN qui gère la diversité d'un ensemble par un processus de sélection. Dans un premier temps, ils apprennent un nombre important de réseaux. Ensuite, ils appliquent sur ce premier ensemble un algorithme génétique dont l'objectif est de trouver une pondération des réseaux qui minimise l'erreur en généralisation. Ils expriment cette erreur comme une somme des corrélations deux à deux entre les réseaux, somme pondérée par les poids de chaque réseau. À l'issue de l'algorithme génétique, seuls les réseaux dont le poids est supérieur à un seuil donné sont conservés. Enfin, la prédiction d'un nouvel exemple repose sur l'agrégation par vote des réseaux sélectionnés, sans tenir compte de la pondération (pour ne pas générer de sur-apprentissage). En pratique, sur les 4 jeux de données, Zhou et al. (2002) rapportent que GASEN donne des résultats meilleurs que le *boosting* ou le

bagging dans 3 cas sur 4.

Oliveira et al. (2005) appellent ce type de stratégie « surproduction et sélection » (*overproduce and choose*). Dans GASEN, la phase de surproduction amène peu de diversité puisque les différents réseaux de neurones sont tous appris sur les mêmes données. Par ailleurs, le critère optimisé dans la phase de sélection est orienté vers la sélection de réseaux en régression. La stratégie proposée par Oliveira et al. (2005) apporte une solution sur ces deux points en appliquant successivement deux stratégies de type « surproduction et sélection ». La première vise la sélection de plusieurs sous-espaces d'attributs qui soient à la fois concis et performants. La deuxième vise la sélection, à partir des réseaux appris sur les sous-espaces sélectionnés à l'étape précédente, des ensembles de réseaux qui optimisent à la fois la précision moyenne de leurs classifieurs et leurs ambiguïtés (au sens de l'équation 3.2). Chacune de ces étapes nécessitant l'optimisation de 2 critères, Oliveira et al. (2005) se servent d'algorithmes génétiques multi-objectifs (*Multi-Objective Genetic Algorithme* (MOGA), Fonseca et Fleming, 1993) qui ont la particularité de sélectionner les individus génétiques les plus performants au sens de plusieurs critères. Enfin, à l'issue de la deuxième phase, l'ensemble le plus performant (déterminé par son erreur sur des données de validation) est retenu. Là encore, la stratégie se montre plus performante que le boosting et le bagging (un seul jeu de données a servi à mener la comparaison).

3 Récapitulatif des méthodes ensemblistes

Les travaux de Krogh et Vedelsby (1995) en régression, ainsi que leur adaptation à la classification par Zanda et al. (2007), montrent que deux composantes sont impliquées dans l'erreur d'un ensemble : l'erreur moyenne des classifieurs de l'ensemble et l'ambiguïté, c'est-à-dire la déviation des prédictions des classifieurs par rapport à la prédiction de l'ensemble. L'ambiguïté et la diversité sont deux notions différentes. En définissant la diversité comme la covariance moyenne entre les classifieurs d'un ensemble (eq. 3.1), Brown (2005) exprime l'ambiguïté en fonction à la fois de cette diversité et de la variance moyenne. Ainsi, en maximisant l'ambiguïté des classifieurs pour maximiser leur diversité et minimiser l'erreur de l'ensemble, on modifie également la variance et le biais de ces classifieurs ce qui a pour conséquence de dégrader leur erreur. C'est pourquoi il existe un compromis entre l'ambiguïté et l'erreur moyenne des classifieurs d'un ensemble, tout comme il existe un compromis biais/variance pour l'apprentissage supervisé (Brown, 2005).

De nombreux travaux ont proposé, implicitement ou explicitement, des solutions à ce compromis. Dans la section précédente, nous les avons divisés en fonction de la typologie détaillée en figure 3.2. Cette typologie montre qu'il existe une grande variété de moyens pour générer de la diversité entre les classifieurs, moyens qu'il est d'ailleurs possible de combiner. Dans la section suivante, nous tirons parti de ces possibilités pour concevoir

deux stratégies adaptées à l'apprentissage d'un comité de cartes.

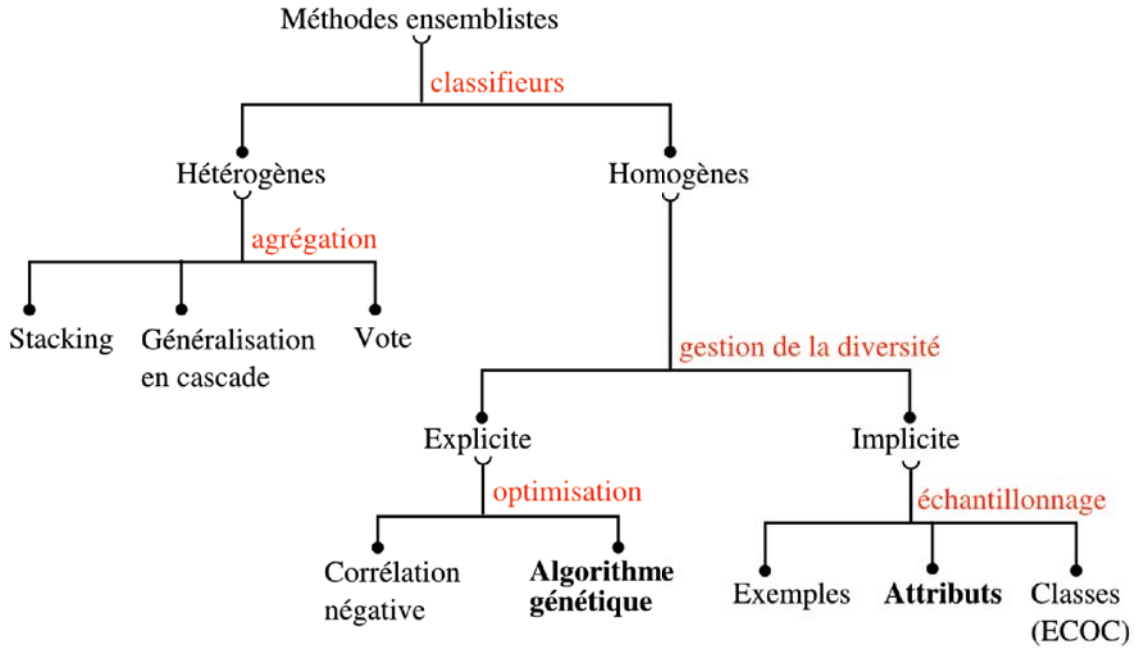


FIG. 3.2: Typologie des méthodes ensemblistes. Les nœuds de l'arbre correspondent aux différentes familles. Le critère qui permet de réaliser la distinction des différentes sous-familles est noté en rouge. Les deux familles dans lesquelles se situent les travaux des sections suivantes sont en gras.

4 Comités de cartes

Comme nous l'avons expérimenté en section 6.3 du chapitre 2, les cartes auto-organisatrices en apprentissage supervisé ne permettent pas une prédiction des plus robustes des données en grandes dimensions. Rassembler ces cartes dans un comité va permettre d'améliorer leur performance. Cependant, l'objectif final demeure la représentation des données. De ce point de vue, le nombre de cartes du comité est un paramètre important. Un grand nombre de cartes de faible qualité permettrait paut-être une erreur faible, mais ces cartes seraient inexploitables du point de vue de leur représentation. Aussi, le comité doit-il limiter au maximum le nombre de cartes qu'il contient afin de pouvoir les synthétiser en une représentation cohérente. En outre, plus le nombre de cartes est important, plus le temps de calcul augmente. Les résultats théoriques décrits en section 1.2, ainsi que les méthodes ensemblistes existantes, nous apportent des informations précises sur les conditions que doivent remplir les classifieurs pour s'agréger avec succès.

Le *bagging* et les forêts aléatoires montrent que l'échantillonnage aléatoire des exemples et des attributs, répété sur un grand nombre de classifieurs, constitue une première solution

à ce problème. Cependant, ces stratégies n'optimisent pas le nombre de cartes membres du comité. Pour réaliser ce contrôle, nous envisageons deux stratégies qui se différencient par la prise en compte ou non de l'étiquette de classe dans le choix des cartes à intégrer au comité. L'avantage de prendre en compte l'étiquette est bien entendu de minimiser efficacement l'erreur. La section 4.1.2 détaille notre proposition pour la construction d'un tel comité (Prudhomme et Lallich, 2008b). Cependant, il est parfois préférable de ne pas utiliser cette étiquette lors de la construction du comité, comme nous l'exposons en section 4.1.1 (Prudhomme et Lallich, 2007). C'est notamment le cas lorsque l'étiquette est bruitée, pour éviter de biaiser le modèle, ou en apprentissage semi-supervisé (Prudhomme et Lallich, 2008a), pour prendre en compte les exemples qui n'ont pas d'étiquette lors de la construction du modèle.

Quelle que soit la stratégie utilisée pour déterminer les cartes à intégrer au comité (qu'elle soit 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. Comme nous l'avons vu en section 2.3, une solution efficace consiste à restreindre l'apprentissage de chaque classifieur à un sous-ensemble des données, que ce soit via un échantillonnage des exemples, des attributs ou une combinaison de ces éléments. Dans le cas de données en grandes dimensions, les attributs engendrent des problèmes liés à la fois à l'apprentissage et à la complexité des algorithmes (voir section 4.1.2 du chapitre 1). Pour contourner ces problèmes dans le cadre d'une méthode ensembliste, il est possible d'apprendre chaque membre du comité 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 d'exploiter l'ensemble des attributs lors de l'apprentissage, contrairement à une stratégie de sélection des attributs durant une phase de pré-traitement.

4.1 Gestion de la diversité

Dans cette section, nous proposons deux stratégies pour concevoir un comité de cartes, qui se servent ou non de l'étiquette pour décider des cartes du comité.

4.1.1 Stratégie non-supervisée

L'intérêt d'un comité construit sans aucune référence à l'étiquette de classe est de pouvoir s'appliquer dans des situations où cette étiquette est bruitée ou indisponible pour une partie des exemples, en conservant une qualité raisonnable de la prédiction (Prudhomme et Lallich, 2007, 2008a).

Comme nous l'avons expliqué dans la section précédente, l'objectif est ici de construire des cartes sur des sous-espaces d'attributs, telles qu'elles aient une bonne prédiction et

qu'elles soient diverses, sans pour autant utiliser l'étiquette. Ainsi, nous devons réfléchir 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 une dimension construites chacune sur un attribut, ces cartes seront d'autant moins diverses que les attributs seront plus corrélés, puisque deux attributs fortement corrélés portent la même information sur la variable de classe. À 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-groupes 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 souhaitons 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 permettrait à coup sûr d'améliorer la prédiction en diminuant les temps de calcul.

Finalement, pour former ces groupes, nous avons procédé en deux étapes. La première étape forme 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. À 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 une 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. À l'issue de cette procédure, on obtient une partition en K grappes d'attributs.

À partir de ces K grappes, 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 constitués 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.

4.1.2 Stratégie guidée par l'étiquette

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 (équation 3.4, section 1.2). Étant 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 d'attributs à apprendre par les différentes cartes du comité (Prudhomme et Lallich, 2005a).

Cependant, le nombre de sous-espaces et de leurs combinaisons a une combinatoire exponentielle avec le nombre de dimensions. 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. 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, nous appliquons une méthode d'optimisation pour trouver la meilleure combinaison de ces cartes au sens de la mesure E. Notons que les cartes ne sont apprises qu'une fois, en amont, 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 Algorithme Génétique (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^{e} bit encode la présence, 1, ou l'absence, 0, de la i^{e} carte dans le comité. 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 phénotype. À 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. 3.3).

Notons que la méthode GASEN de Zhou et al. (2001) (décrite en section 2.4) sélectionne un sous-ensemble de classifieurs en pondérant l'importance des classifieurs dans un ensemble. Cependant, ces auteurs ne se servent pas de cette pondération lors de la prédiction d'un nouvel exemple. Au contraire, ils introduisent une pondération seuil en dessous de laquelle un classifieur est retiré du comité, ce qui à l'inconvénient de rajouter un paramètre supplémentaire. Pour éviter ce paramètre, il serait possible de conserver les pondérations de chaque carte lors de l'agrégation des prédictions. Pour tester l'intérêt des pondérations, nous avions d'abord fixé une pondération pour chaque carte en ayant recours à leur qualité individuelle à l'aide de la statistique J' corrélée avec leur taux d'erreur en généralisation

(Prudhomme et Lallich, 2005b, détaillé en section 4.2.3 du chapitre 2). Néanmoins, les expériences que nous avons menées sur ce type de comité n'ont pas été concluantes. Nous attribuons l'inefficacité de cette pondération au fait qu'elle détermine le poids d'une carte en fonction de sa qualité globale, ce qui risque d'aller à l'encontre de son rôle dans le comité. Par exemple, il est possible d'imaginer une carte dont la qualité globale est faible et qui s'avère pourtant de très bonne qualité lorsqu'il s'agit d'une petite région de l'espace souvent mal représentée par les autres cartes du comité. En pondérant cette carte sur sa qualité globale, la diversité que cette carte apporte sur cette région particulière disparaît. Finalement, nous pensons que la notion d'ensemble se prête mal à une pondération rigide de chaque classifieur. C'est pourquoi nous nous sommes cantonné à la sélection des cartes lors de l'optimisation par l'AG.

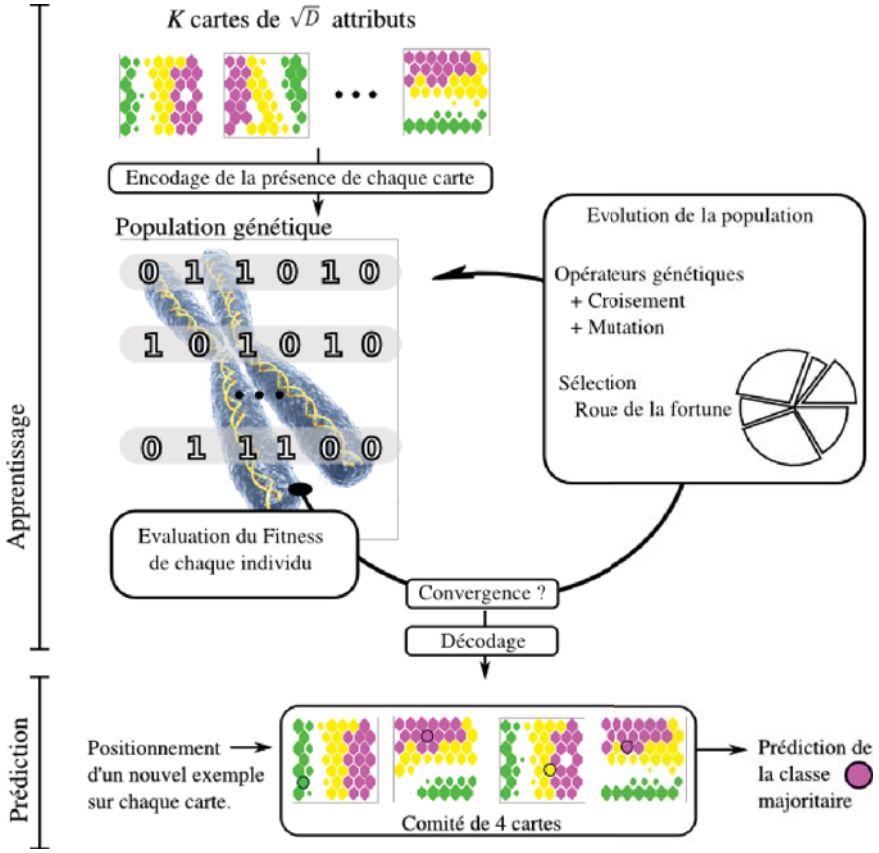


FIG. 3.3: Les différentes étapes de l'optimisation d'un comité par un AG.

4.2 Intégration par stacking géographique

Comme nous le verrons dans la section 5 sur les expérimentations, un comité 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 parvenir, nous proposons

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

Pour constituer cette structure, nous avons choisi de mettre en place une carte synthétique qui tire parti des informations issues des cartes de l'ensemble, en adaptant à l'apprentissage non-supervisé le *stacking* (Wolpert, 1992, voir section 2.1) utilisé en apprentissage supervisé. En apprentissage supervisé, le *stacking* consiste à réaliser un second apprentissage, à la suite de l'apprentissage d'un ensemble de classifieurs, pour lequel chaque exemple est représenté non plus par ses attributs, mais par les prédictions des classifieurs de l'ensemble. Le *stacking* peut se voir comme une méthode d'agrégation qui vise l'amélioration de la prédiction de l'ensemble. Cependant, en cherchant à construire une carte synthétique, nous ne cherchons pas à améliorer la prédiction du comité, mais plutôt à assurer sa représentation.

Chaque carte du comité est une projection de l'espace d'apprentissage qui associe à chaque exemple un modèle positionné dans l'espace de la carte. Ces associations entre les exemples et leur modèle vainqueur sur chaque carte constituent la représentation de cet exemple par le comité. En effet, deux exemples proches dans l'espace d'apprentissage auront tendance à être représentés par des modèles 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 modèles qui les représentent. De cette manière, la proximité entre les exemples est considérée au sens des modèles 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 modèles vainqueurs. L'espace d'apprentissage de cette carte synthétique est ainsi construit à partir des coordonnées des modèles vainqueurs d'un exemple sur chaque carte. Par exemple, pour un comité de 3 cartes, si un exemple est représenté sur la 1^{re} carte par le modèle de coordonnées $(3, 2)$, sur la 2^{e} par celui de coordonnées $(1, 5)$ et sur la 3^{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.

À 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.

Celui-ci 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 modèle de cette carte possède des coordonnées dans son espace d'apprentissage, c'est à dire celui des coordonnées des modèles vainqueurs de chaque carte. Il est donc possible d'associer, à chaque modèle de la carte, un modèle sur chaque carte du comité. De cette manière, pour chaque exemple positionné sur la carte, on peut trouver une approximation des modèles vainqueurs sur les cartes de l'ensemble, et par la suite, des exemples associés à ces modèles.

4.3 Étude de la complexité

La stratégie non-supervisée de construction d'un comité de cartes se divise en deux étapes, la première relative à Varclus et la seconde à l'apprentissage des cartes du comité. 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 $d^2 + 2 \times d \times \log_2 K$. En considérant $k = K$ groupes composés chacun de $d' = K$ attributs (ce qui correspond aux paramètres utilisés dans les expérimentations), K cartes sont à apprendre, chacune sur K attributs, soit une complexité en $O(K^2 \times n \times m)$, avec $K < d/2$, puisqu'on a au moins 2 attributs par groupes.

En ce qui concerne la stratégie guidée par l'étiquette, la recherche du meilleur ensemble de sous-espaces se limite à l'exploration de k sous-espaces de taille $\sqrt{d}$. Une carte est apprise pour chaque sous-espace avant l'optimisation par l'algorithme génétique, soit une complexité en $O(k \times m \times n \times \sqrt{d})$. À l'issue de cette phase, la recherche du meilleur comité requiert l'estimation de la mesure E sur chaque phénotype et à chaque génération (par la suite, on note I le nombre de phénotypes et G le nombre de générations). Cette estimation nécessite de connaître la différence entre les probabilités *a posteriori* de prédire chaque classe pour chaque exemple de validation sur les différentes cartes du phénotype, soit une complexité en $O(I \times G \times n' \times k \times c^2)$ avec n' le nombre d'exemples en validation ($n' < n$) et c le nombre de classe.

5 Expérimentations

5.1 Apprentissage supervisé

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 (notée carte seule dans les graphiques et tableaux), 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 (Blake et Merz, 1998) présentés dans le tableau 3.2. Pour les forêts aléatoires, le boosting et les 3-PPV, les expérimentations ont été menées sur la plateforme Tanagra (Rakotomalala, 2005). Les taux d'erreurs ont été obtenus par des 2-validations-croisées répétées 5 fois, comme recommandé par Dietterich (1998). Ils sont reportés en figure 3.4 avec les écarts-types correspondants.

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

TAB. 3.2: 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.

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 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 modèles 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 d'obtenir un modèle pour chaque classe des données mais plutôt de décrire la répartition des étiquettes dans l'espace d'apprentissage. 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 modèles qu'il n'y a d'exemples à apprendre. Dans ces cas, nous fixons le nombre de modèles au $^2/_3$ du nombre d'exemples d'apprentissage, en gardant un rapport de $^3/_2$ entre la hauteur et la largeur de la carte. Le tableau 3.2 récapitule les tailles de cartes 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 choisi $d' = K$ 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*.

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 mises en place 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 l'ambiguïté 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 3.5). 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 au travers de l'optimisation de l'ambiguïté. 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 3.3. 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 modèle vainqueur 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 χ_2 pour déterminer si la proportion d'exemples de chaque classe observée dans le modèle est significativement différente de la proportion observée dans les données. Notons également que 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 modèles 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. 3.3: 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.

5.2 Contextualisation et navigation

Pour ces expérimentations, les données présentées dans la table 3.2 de la section 5.1 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 3.5. 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 obtenus 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

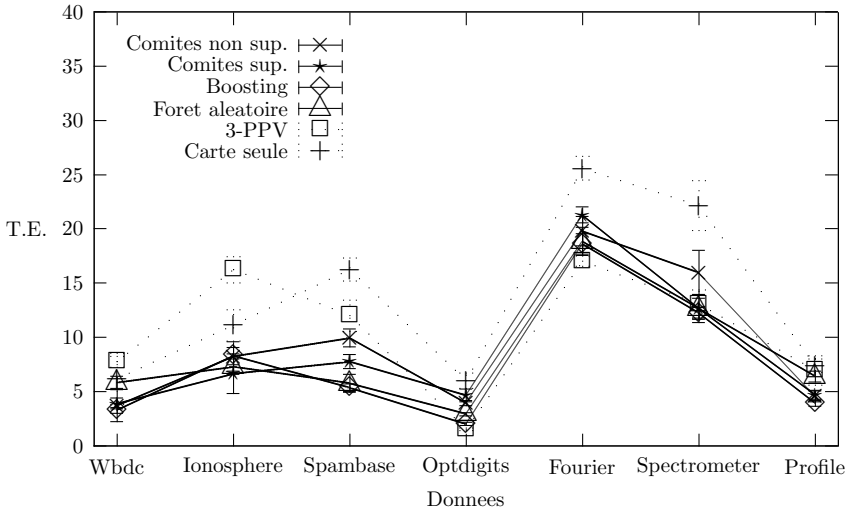


FIG. 3.4: Taux d'erreur des comités de cartes supervisés et non-supervisés comparés à ceux d'une carte seule apprise sur la totalité des attributs, du boosting et des forêts aléatoires.

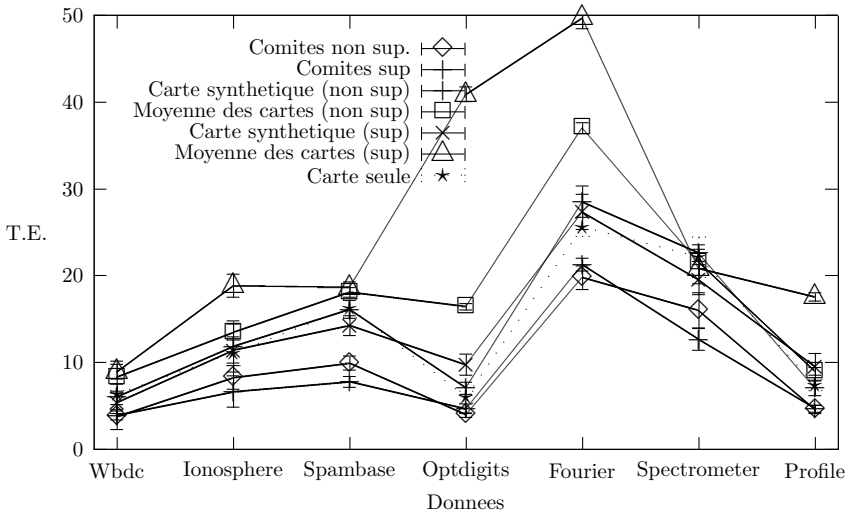


FIG. 3.5: Taux d'erreur d'une carte synthétique comparé à celui d'une carte seule apprise sur la totalité des attributs, d'un comité de cartes et à la moyenne des taux d'erreur des cartes du comité.

	Comité		3-PPV	Boosting	Forêts
	Non sup.	Sup.			aléa.
Wbdc	3,7 ±1,4	3,9 ±0,5	7,8 ±0,5	3,3 ±0,3	5,8 ±0,6
Ionosphère	8,2 ±1,3	6,6 ±1,8	16,2 ±1,2	8,3 ±0,8	7,3 ±0,7
Spambase	9.9 ±0,8	7,7 ±0,6	12 ±1,5	5,4 ±0,4	5,8 ±0,9
Optdigits	3,9 ±0,2	4,7 ±0,6	1,6 ±0,1	2,0 ±0,1	3,0 ±0,2
Multi-feat. (Fourier)	19,8 ±1,3	21,3 ±0,7	17,0 ±0,5	19,6 ±0,9	18,8 ±1,3
Spectrometer lrs	15,9 ±2,0	12,6 ±1,3	13,1 ±1,3	12,2 ±0,6	12,7 ±0,9
Multi-feat. (Profile)	4,6 ±0,5	4,7 ±0,4	7,0 ±1,3	3,9 ±0,3	6,4 ±0,5

TAB. 3.4: Taux d'erreur des comités de cartes comparés à ceux des 3-PPV, des forêts aléatoires et du boosting. Pour chaque taux d'erreur est reporté l'écart-type associé, estimé à partir de la validation croisée.

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 (voir section 5 du chapitre 2), mais pas seulement. La carte synthétique du comité sert également en situation d'aide à la décision où elle vient expliquer la prédiction. Ainsi, lorsque l'utilisateur soumet un nouvel exemple, l'étiquette renvoyée par le comité 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 sont représentés par le même modèle que l'exemple soumis. Lorsque ce modèle ne représente aucun exemple, ou que l'utilisateur veut une plus grande précision, il est possible de considérer la proximité au sens du comité en considérant les exemples qui sont représentés par le même modèle sur chaque carte du comité. Dans ce deuxième cas, les exemples correspondant sont pondérés en fonction du nombre de modèles qui les représentent dans l'ensemble. De cette manière, l'utilisateur peut mieux juger de la prédiction, en visualisant les exemples sur laquelle elle repose, et ainsi détecter les éventuelles erreurs. Enfin, cette carte sert à la navigation pure, pour rechercher des cas similaires à un exemple, en proposant à l'utilisateur d'explorer le voisinage de l'exemple au travers du voisinage défini par les modèles de la carte.

6 Résistance au bruit

Nous étudions dans cette section la résistance d'un comité non-supervisé au bruit sur la classe. Avant de présenter les raisons de la résistance au bruit d'un comité de cartes, nous introduisons succinctement quelques travaux du domaine.

6.1 Le bruit en apprentissage

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 est de nature à dégrader l'apprentissage supervisé (Quinlan, 1990) ce qui demande un traitement particulier.

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-PPV avant de réaliser la prédiction avec un algorithme 1-PPV. 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 du 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), mais aussi des arbres de décision (Sakakibara, 1993) ou encore du boosting Freund (BrownBoost, 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 expérimentations menées par Brodley et Friedl (1999) montrent par exemple que les ensembles permettent de filtrer entre 20% et 30% du bruit suivant les données.

6.2 Intérêt du comité de cartes

En pratique, les méthodes filtres sont les plus souvent appliquées pour leur facilité d'implémentation. Elles soulèvent néanmoins deux problèmes. D'abord, elles réduisent les données d'apprentissage en retirant les exemples qu'elles considèrent comme bruités. 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, ces auteurs comparent les résultats obtenus par le vote à la majorité d'un ensemble de trois classifieurs hétérogènes (1-PPV, 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). À l'issue de cette comparaison, ils concluent que le vote à la majorité ne peut se substituer au filtrage des données pour gérer le bruit, en particulier pour les niveaux les plus élevés. Notons que 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 bruitées. Le nombre d'exemples ayant servi à leur apprentissage est néanmoins fortement réduit lors du filtrage.

Les cartes auto-organisatrices ne souffrent pas des étiquettes bruitées lors de l'apprentissage. 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 modèles 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 4.1.1 – gère la présence d'étiquettes erronées à deux niveaux. Au niveau local de chaque carte, les modèles sont étiquetés avec la classe majoritaire des exemples qu'ils représentent. 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 modèles.

Contrairement aux méthodes filtres, toutes les données sont prises en compte lors de l'apprentissage du comité de cartes. Par rapport aux méthodes filtres, cela apporte deux avantages. Le premier est de garantir un meilleur apprentissage, en particulier dans le cas des données en grandes dimensions où le nombre d'exemples est critique. Le deuxième

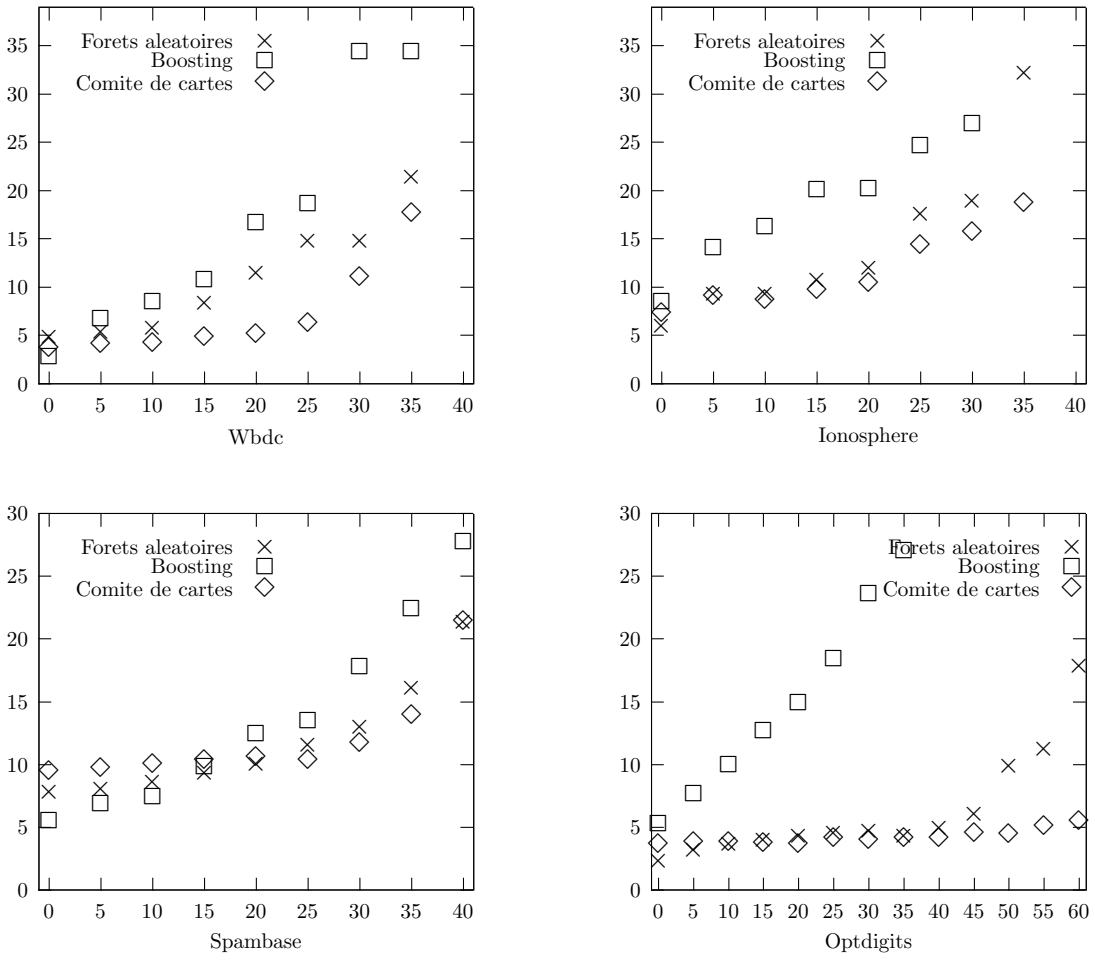


FIG. 3.6: De gauche à droite et de haut en bas les jeux *wdbc*, *ionosphère*, *spambase* et *optdigits*. Sur chaque figure est reporté en abscisse le niveau de bruit (en %) et en ordonnée le taux d'erreur (en % également)

avantage est d'éviter de considérer les exceptions comme des données bruitées.

6.3 Expérimentations

Pour évaluer la robustesse de notre approche face aux données bruitées, nous avons utilisé les 4 premiers jeux de données présentés dans le tableau 3.2 en section 5.1. 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 classe, 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), une forêt aléatoire et un *boosting* d'arbre. 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 L'apprentissage semi-supervisé

Dans cette section, nous nous sommes intéressés à la capacité d'un comité non-supervisé à répondre au problème de l'apprentissage supervisé. Dans la suite, nous présentons ce contexte particulier et quelques stratégies mises en place. Afin de rester concis, nous avons choisi les stratégies qui sont le plus en lien avec notre travail. Ensuite, nous présentons l'intérêt du comité de cartes pour l'apprentissage supervisé et les expérimentations que nous avons menées dans ce domaine (Prudhomme et Lallich, 2008a).

7.1 Le problème de l'apprentissage semi-supervisé

En apprentissage supervisé, les algorithmes infèrent un modèle de prédiction à partir de données préalablement étiquetées. Cependant, l'étiquetage est un processus long et coûteux qui nécessite souvent l'intervention d'un expert. Cette phase contraste avec une acquisition automatique des données, libérée par ailleurs des contraintes de stockage. Il n'est alors pas rare de se retrouver avec un volume important de données dont seulement une petite partie à pu être étiquetée. Par exemple, en recherche d'images par le contenu, l'utilisateur souhaite étiqueter le minimum d'images pour fouiller une base aussi grande que possible. Dans ce contexte, l'apprentissage semi-supervisé (SSL) intègre les données non-étiquetées dans la mise en place du modèle de prédiction. En ce sens, l'apprentissage semi-supervisé est à mi-chemin entre les apprentissages supervisé et non-supervisé : il cherche à exploiter les données non-étiquetées pour apprendre la relation entre les exemples et leur étiquette.

L'apprentissage semi-supervisé fait donc l'hypothèse que les données non étiquetées vont

améliorer l'apprentissage. Cependant, cela n'est pas toujours vrai (Zhu, 2005). Chapelle et al. (2006) distinguent trois hypothèses que les données doivent satisfaire pour pouvoir profiter des exemples non-étiquetés.

- l'hypothèse de régularité : « *if two points* $x1, x2$ *in a high-density region are closed, then should be the corresponding outputs* $y1, y2$ »,
- l'hypothèse de classe : « *if points are on the same cluster, they are likely to be of the same class* »,
- l'hypothèse de sous-espace : « *the (high-dimensional) data lie (roughly) on a low-dimensional manifold* ».

La première hypothèse est une condition que les données doivent remplir pour permettre une généralisation à partir d'un petit nombre d'étiquettes. En effet, des données pour lesquelles deux exemples très proches, situés dans des régions denses de l'espace d'apprentissage, n'auraient pas plus de chance de posséder des étiquettes communes, seraient des données dont la nature empêche un apprentissage semi-supervisé. Les deux autres hypothèses spécifient des conditions que les données peuvent remplir et sous lesquelles il est possible d'établir une généralisation à partir d'un petit nombre d'exemples étiquetés : soit les données sont regroupées en classes homogènes soit les données sont regroupées dans des sous-espaces denses et homogènes. La différence entre ces deux hypothèses concerne essentiellement l'espace dans lequel est mesurée la distance entre deux exemples.

Parmi les méthodes qui profitent de ces hypothèses, Zhu (2005) distingue plusieurs familles telles que les modèles génératifs (*generative models*), les méthodes fondées sur les graphes, les méthodes fondées sur la recherche des séparations maximales (de type TSVM) ou encore le *co-training* (Blum et Mitchell, 1998). Cette dernière catégorie de méthodes a un lien direct avec les méthodes ensemblistes puisqu'elle se sert de plusieurs classifieurs pour gérer les exemples non-étiquetés. Dans leur travail initial, Blum et Mitchell (1998) s'intéressent à la classification semi-supervisée de pages web. Suivant les données qu'ils ont récoltées, chaque page est décrite par deux ensembles d'attributs : le premier correspond aux mots présents dans la page, alors que le second contient les mots présents dans les liens hypertextes qui font référence à la page en question. Ils entraînent alors deux classifieurs sur ces deux espaces différents au cours d'un processus itératif. Chaque classifieur est entrainé une première fois sur les exemples étiquetés pour classifier un petit sous-ensemble d'exemples non-étiquetés. L'obtention de ces nouveaux exemples étiquetés permet de répéter l'apprentissage des classifieurs sur un nombre croissant d'exemples. La procédure s'arrête lorsque tous les exemples non-étiquetés ont servi lors de l'apprentissage. À l'issue de cette procédure, les auteurs mettent en place un classifeur combiné qui apprend les prédictions des deux classifieurs initiaux et sert à la prédiction finale d'une nouvelle page web. Les auteurs ont également montré que l'indépendance des deux sous-ensembles d'attributs était une condition nécessaire pour que cet algorithme du *co-training* améliore la prédic-

tion. En pratique, cependant, il n'est pas toujours possible d'obtenir deux sous-ensembles d'attributs indépendants par rapport à l'étiquette, ce qui rend le *co-training* difficilement généralisable. Pour contourner cette difficulté, Zhou et Goldman (2004) proposent d'adapter cette stratégie à un ensemble de classifieurs hétérogènes. Leur méthode suit la procédure mise en place par Blum et Mitchell (1998), même s'ils sont obligés de contrôler la qualité des exemples nouvellement étiquetés avant de les ajouter définitivement à l'apprentissage. Enfin, comme ils disposent d'un ensemble de 3 classifieurs, ils réalisent la prédiction par un vote à la majorité plutôt que par un classifieur combiné ou un *stacking*.

Le *co-training* étend la connaissance portée par les exemples étiquetés grâce à l'apprentissage supervisé. Les méthodes d'apprentissage semi-supervisé fondées sur les graphes reposent au contraire sur la propagation des étiquettes au sein d'une structure non-supervisée (le graphe). À ce titre, la méthode des champs gaussiens et des fonctions harmoniques constitue un bon exemple de ces stratégies (Zhu et al., 2003). Qui plus est, cette stratégie a été adaptée à l'apprentissage semi-supervisé d'une carte auto-organisatrice par Herrmann et Ultsch (2007). Dans ce cas, les nœuds du graphe sont les modèles de la carte et les arêtes sont définies suivant les relations de voisinage sur la carte entre ces modèles. À partir de cette structure, Zhu et al. (2003) (suivis par Herrmann et Ultsch (2007)) affectent à chaque paire de nœuds un poids inversement proportionnel à la distance entre ces nœuds. Dans le cas d'une carte, la distance entre deux modèles voisins est calculée dans l'espace d'apprentissage et la distance entre deux modèles quelconques correspond au plus court chemin entre ces modèles, en se déplaçant de voisins en voisins sur la carte. De cette manière, la distance tient compte de la topologie des données au travers de la projection de la carte. Pour Zhu et al. (2003), la propagation des étiquettes dans le graphe revient alors à trouver une fonction d'étiquetage telle que la somme des poids des paires de nœuds d'étiquettes différentes soit la plus petite possible. Dans leur article, Zhu et al. (2003) montrent que cette fonction est harmonique et qu'elle correspond à la moyenne pondérée des étiquettes des noeuds voisins. Pour les cartes, Herrmann et Ultsch (2007) mettent en place une procédure itérative de mise à jour des étiquettes des modèles non-étiquetés à partir de cette fonction.

Les approches proposées par Blum et Mitchell (1998) et Zhou et Goldman (2004) (ainsi que d'autres telles que Becker (1995); Leskes et Torenvliet (2008)) montrent l'avantage qu'il existe à utiliser plusieurs classifieurs. Cependant, apprendre ces classifieurs implique de prédire les exemples non-étiquetés avant d'en profiter. À l'inverse, si la prédiction repose sur l'apprentissage non-supervisé (comme c'est le cas pour l'apprentissage semi-supervisé à partir des cartes auto-organisatrices), les exemples non-étiquetés jouent un rôle avant d'être prédits. C'est pourquoi nous pensons que les comités de cartes peuvent s'appliquer à l'apprentissage semi-supervisé. Dans la section suivante, nous montrons leur intérêt dans

ce domaine.

7.2 Comité de cartes et apprentissage semi-supervisé

Dans le contexte de l'apprentissage semi-supervisé, la représentation des données apporte plusieurs avantages. Tout d'abord, l'apprentissage semi-supervisé partage avec l'apprentissage actif le problème du choix des exemples à apprendre pour garantir un apprentissage de qualité. Or, la représentation peut aisément servir de structure pour effectuer ce choix en déterminant un sous-ensemble d'exemples approprié. Ces exemples peuvent être sélectionnés de telle sorte qu'ils soient convenablement répartis dans l'espace d'apprentissage ou encore au cœur des régions les plus denses de l'espace. La représentation peut également servir lors de la prédiction d'un nouvel exemple. Ainsi, en montrant à l'utilisateur les exemples proches de sa requête qui ont été étiquetés automatiquement, celui-ci pourra juger de l'efficacité du processus d'étiquetage automatique dans la région d'intérêt. Enfin, comme nous l'avons montré en section 4 du chapitre 2, la qualité de la représentation est accessible statistiquement, que ce soit dans un contexte supervisé ou non-supervisé. Or, l'un des problèmes majeurs de l'apprentissage semi-supervisé concerne la validation de l'apprentissage, puisqu'il n'y a pas suffisamment d'exemples étiquetés pour constituer des données de tests. La validation statistique de la représentation doit pouvoir apporter des réponses à ce problème.

L'apprentissage à partir de la représentation est donc approprié au contexte semi-supervisé. Néanmoins, en pratique, les données concernées par l'apprentissage semi-supervisé sont souvent volumineuses. Comme nous l'avons évoqué précédemment, le problème d'apprentissage supervisé se produit souvent à la suite d'une collecte automatique des données. Ce type de collecte engendre à la fois un problème d'étiquetage mais également un grand nombre d'exemples (c'est le cas, entre autres, pour la classification de pages web, Blum et Mitchell, 1998). En outre, le problème de l'apprentissage semi-supervisé se pose sur des données ayant une sémantique complexe, comme le texte ou l'image, que seul un expert humain est capable d'étiqueter. Or, ces données requièrent souvent un grand nombre d'attributs pour leur description. Cette double volumétrie attributs/exemples rajoute à l'apprentissage semi-supervisé les problèmes des données volumineuses décrits en section 4 du chapitre 1. C'est pourquoi, nous nous proposons d'adapter les comités de cartes à ce contexte.

Les comités de cartes tels que décrits dans la section 4.1.1 conviennent à l'apprentissage semi-supervisé (Prudhomme et Lallich, 2008a). Néanmoins, dans cette thèse, nous proposons de remplacer la fonction Kohonen-Opt de prédiction (voir section 3.3, chapitre 1), par la procédure de propagation des étiquettes de Herrmann et Ultsch (2007), plus adaptée à l'apprentissage semi-supervisé. Cependant, dans leur version, Herrmann et Ultsch (2007)

ne semblent pas tenir compte de la procédure de normalisation qui vise à intégrer les connaissances sur les probabilités *a priori* de chaque classe dans le processus de prédiction (Zhu et al., 2003). Dans nos tests, nous avons remarqué que cette normalisation améliore notablement la prédiction, c'est pourquoi nous l'avons intégrée dans notre implémentation.

7.3 Expérimentations

Dans leur livre sur l'apprentissage semi-supervisé, Chapelle et al. (2006) ont mis en place un cadre de validation. Ils mettent ainsi à disposition plusieurs jeux de données[1]. Le tableau 3.5 décrit ces jeux en termes de nombre d'exemples, d'attributs et de classes. Pour estimer l'erreur sur chaque jeu, les auteurs ont prédéterminé 12 ensembles d'exemples étiquetés (ces ensembles sont disponibles conjointement avec les jeux). Pour chaque ensemble, l'algorithme est évalué sur sa capacité à prédire correctement l'étiquette des exemples non-étiquetés[2]. Les exemples étiquetés ont été choisis aléatoirement par les auteurs, avec pour seule contrainte qu'au moins un exemple de chaque classe soit présent dans chaque ensemble. Enfin, pour chaque jeu de données, les 12 ensembles d'exemples étiquetés ont été définis à la fois avec 100 exemples et 10 exemples, ce qui permet d'évaluer la robustesse des méthodes lorsque le nombre d'exemples étiquetés disponibles diminue.

TAB. 3.5: Descriptions des données

	g241c	**g241d**	**Digits1**	**USPS**	**BCI**
Nb. d'attributs	241	241	241	241	117
Nb d'exemples	1500	1500	1500	1500	400
Nombre de classes	2	2	2	2	2

Les auteurs rapportent dans leur livre les taux d'erreurs de 2 méthodes d'apprentissage supervisé (1-PPV et SVM) et de 11 méthodes d'apprentissage semi-supervisé. La plupart des grandes familles d'apprentissage semi-supervisé sont représentées à l'exception du *co-training*. Pour chaque méthode, les auteurs donnent uniquement la moyenne des taux d'erreur sur les 12 ensembles d'exemples étiquetés.

Nous avons donc appliqué les comités de cartes sur les données de Chapelle et al. (2006) en respectant les exemples étiquetés qui leur sont définis. De cette manière, les taux d'erreur que nous obtenons peuvent se comparer à ceux des autres méthodes. En termes de paramétrage des comités, nous avons employé les mêmes que ceux décrits en section 5, à l'exception du choix de la taille des cartes. Pour l'apprentissage semi-supervisé, nous

[1] disponibles à l'adresse suivante http ://www.kyb.tuebingen.mpg.de/ssl-book/

[2] en ce sens, l'évaluation porte sur un problème de transduction

avons eu recours à des cartes plus petites, de taille 10×10 pour 100 exemples étiquetés et 6×6 pour 10 exemples étiquetés. Les résultats sont reportés dans les tableaux 3.6 pour 100 exemples étiquetés et 3.7 pour 10 exemples étiquetés. Dans ces tableaux, nous avons également reporté les taux d'erreur des SVM et des 1-PPV, le rang des comités de cartes par rapport aux autres stratégies semi-supervisées et enfin le taux d'erreur minimum et maximum de ces stratégies pour chaque jeu de données.

TAB. 3.6: Taux d'erreur du comité de cartes, des 1-PPV et des SVM lors de l'apprentissage semi-supervisé des données avec 100 exemples étiquetés. La ligne rang donne le classement des comités comparativement aux 11 autres méthodes d'apprentissage semi-supervisé dont le taux d'erreur est donné dans l'article de Chapelle et al. (2006)

	g241c	**g241d**	**Digits1**	**USPS**	**BCI**
1-PPV	43.9	42.4	3.9	5.8	48.7
SVM	23.1	24.6	5.5	9.7	34.3
Comité	23.1	17.6	3.2	6.8	46.8
Rang (/12)	7	3	7	7	9
[min ;max]	[13.5 ;43.6]	[4.9 ;41.6]	[2.8 ;6.1]	[4.7 ;9.8]	[31.4 ;47.9]

TAB. 3.7: Taux d'erreur du comité de cartes, des 1-PPV et des SVM lors de l'apprentissage semi-supervisé des données avec 10 exemples étiquetés. La ligne rang donne le classement des comités comparativement aux 11 autres méthodes d'apprentissage semi-supervisé dont le taux d'erreur est donné dans l'article de Chapelle et al. (2006)

	g241c	**g241d**	**Digits1**	**USPS**	**BCI**
1-PPV	47.9	46.7	13.6	16.7	49
SVM	47.3	46.7	30.6	20	49.8
Comité	31.1	28.0	12.0	17.3	48.9
Rang (/12)	4	2	4	3	5
[min ;max]	[22.8 ;47.1]	[18.6 ;42]	[5.4 ;23.5]	[13.6 ;25.4]	[46.9 ;50.4]

En premier lieu, notre approche améliore à la fois la prédiction des SVM et des 1-PPV dans la très grande majorité des cas, la seule exception concernant les données USPS où les 1-PPV font partie des méthodes les plus performantes. Ce premier élément montre que les comités de cartes bénéficient des exemples non-étiquetés pour la prédiction. Comparativement aux autres approches, les comités de cartes se placent dans la moyenne lorsque 100

étiquettes sont disponibles, mais font partie des quatres méthodes les plus performantes lorsque 10 étiquettes seulement sont à disposition. Il s'agit là d'un point important qui montre que les comités de cartes réalisent un réel apprentissage semi-supervisé, capable d'extraire l'information relative à l'étiquette même dans des cas où celle-ci est peu présente.

Ces expérimentations montrent également que les méthodes ensemblistes ont un rôle à jouer dans le cadre de l'apprentissage semi-supervisé. Jusque là, seuls des petits ensembles de classifieurs supervisés ont été adaptés à cet apprentissage dans le cadre du *co-training*. Nous pensons que des ensembles de classifieurs fondés sur la représentation ou sur un apprentissage non-supervisé constituent une voie prometteuse pour l'apprentissage semi-supervisé.

8 Discussion

Pour contourner les difficultés d'obtention d'une représentation de qualité en un temps raisonnable, nous avons eu recours aux méthodes ensemblistes. Notre proposition est un processus en deux temps. Le premier temps concerne l'apprentissage d'un comité de cartes, suivi dans un deuxième temps de l'émergence, à partir du comité, d'une représentation unique des données qui prend la forme d'une carte synthétique.

Le comité est orienté vers l'amélioration de l'erreur en prédiction. Dans notre proposition, ce comité joue un rôle central puisqu'il vise à doter un apprentissage fondé sur la représentation de capacité prédictive d'un niveau sensiblement équivalent à celui des méthodes les plus robustes en apprentissage supervisé ou semi-supervisé. En particulier, lorsque la dimensionnalité des données augmente, le comité reste en mesure d'offrir une prédiction fiable. L'avantage du comité réside dans la diversité de ses membres, ici des cartes auto-organisatrices, qui sont appris chacun sur un sous-espace différent. Alors que l'apprentissage d'une carte sur l'espace des attributs se dégrade lorsque le nombre d'attributs augmente, l'apprentissage de plusieurs cartes sur des sous-espaces différents tire parti de la diversité pour bénéficier de l'augmentation des attributs. Nous avons proposé deux stratégies pour déterminer les sous-espaces sur lesquels apprendre chaque carte du comité. La première stratégie est non-supervisée. Elle repose sur une heuristique qui exprime le compromis erreur moyenne/ambiguïté en terme de corrélation entre attributs. La deuxième stratégie est supervisée. Elle optimise une mesure d'erreur pour guider une sélection des meilleurs sous-espaces. Cette stratégie offre à la fois les meilleurs résultats en prédiction et les comités les plus concis. Elle montre clairement qu'il est possible de concevoir un petit comité de cartes dont l'erreur en prédiction est faible. Les résultats de la stratégie non-supervisée en apprentissage supervisé sont mécaniquement moins bons. En revanche, la nature non-supervisée du comité le rend robuste au bruit sur la variable de

classe et lui permet de réaliser un apprentissage semi-supervisé. Dans ce cadre, il obtient de bonnes performances, en particulier dans le cas où le nombre d'étiquettes disponibles est faible.

La carte synthétique a pour vocation, quant à elle, d'offrir une représentation des données. La stratégie mise en place pour obtenir cette représentation établit un lien avec le comité de cartes. Grâce à ce lien, la navigation à travers les données est possible à deux niveaux : celui de la carte synthétique et celui du comité. Pour un modèle donné sur la carte synthétique (identifiable à partir d'un exemple-requête ou d'une région d'intérêt), il est possible de naviguer à travers les modèles voisins (navigation au niveau de la carte) ou à travers les modèles de chaque carte du comité (navigation au niveau du comité). Enfin, la carte synthétique permet la visualisation des données.

Les travaux présentés dans ce chapitre montrent qu'un apprentissage fondé sur la représentation des données est applicable au cas des données volumineuses. Ces travaux esquissent également l'intérêt d'un tel apprentissage, que ce soit lorsque les étiquettes sont bruitées ou lorsque le processus d'étiquetage ne peut être mis en place pour tous les exemples. Néanmoins, les stratégies supervisées et semi-supervisées (par exemple, celles auxquelles nous avons eu recours dans nos expérimentations) peuvent être opposées à une stratégie représentationnelle. Dans le cas général – où le taux d'erreur est la seule finalité – la stratégie représentationnelle a peu de chances d'être la plus performante. C'est pourquoi il nous paraît important d'orienter la suite de la discussion sur la place de l'apprentissage à partir de la représentation dans la fouille de données.

Ces dernières années ont émergé sur internet différents acteurs qui proposent à la fois l'hébergement de données individuelles et leur mise à disposition à une communauté d'utilisateurs. Tous les types de données sont visés : textes (*del.icio.us*), photos (*flickr*), images (*deviant-art*), musiques (*jamendo*), vidéos (*youtube*), informations personnelles (*facebook*). Dans ce contexte, un challenge important est de faciliter l'accès à ces données pour un utilisateur. Pourtant, la mise en place d'un processus efficace de recherche d'information se heurte à plusieurs problèmes :

- le volume de données à organiser est important, notamment parce que l'enrichissement de ces bases de données est ouvert à tous,
- la sémantique de ces données est difficilement accessible au travers d'une description en matrice attributs/valeurs,
- le sens d'une recherche varie suivant les utilisateurs.

Ce dernier point s'illustre facilement par les recherches dans les bases d'images : à partir d'une même image-requête, un utilisateur peut vouloir rechercher des images similaires sur le contenu (les formes) ou des images similaires sur les couleurs. À l'extrême, il est possible d'imaginer que l'utilisateur ne s'intéresse à la similarité que sur une partie de l'image

seulement. Ce type de problématique tranche avec un usage plus traditionnel de la fouille de données où des experts du domaine s'attachent à décrire leurs données en vue d'un problème d'apprentissage particulier. Ici, pour une même base de données, le problème varie. Face à ces difficultés, l'étiquetage des données – ou leur indexation – à partir d'informations sémantiques est de nature à aider fortement la recherche. Les différents hébergeurs de contenu public ont donc rendu accessible à leurs utilisateurs, en plus de la consultation des données, un système de *tags* qui permet de donner un ou plusieurs « sens » aux données et sur lesquels il est possible d'effectuer des recherches.

Le système de *tags* conduit à un étiquetage multiple, partiel et bruité :
- multiple : des utilisateurs différents ont de fortes chances d'associer des *tags* différents ;
- partiel : toutes les données ne sont pas étiquetées ;
- bruité : les utilisateurs à l'origine des *tags* ne sont pas experts des données qu'ils manipulent.

Nous pensons que la place de la représentation dans la fouille de données se trouve dans ce type de problématique. Le premier intérêt de la représentation dans ce domaine porte évidemment sur la propagation des tags. Comme nous l'avons vu au cours de ce chapitre, un apprentissage fondé sur la représentation, et en particulier un comité de cartes, garantit un apprentissage robuste au bruit mais permet également un apprentissage semi-supervisé des données. Par ailleurs, d'autres travaux (en particulier au sein du laboratoire ERIC) ont montré qu'une structure de graphe de voisinage permettait la gestion de plusieurs étiquettes associées aux exemples (Hacid, 2008). Or, les graphes de voisinage constituent une représentation des données à condition d'être associée à une stratégie de projection.

En outre, le problème de ces données n'est pas seulement un problème d'optimisation de l'apprentissage. Ou plutôt, pour assurer le meilleur apprentissage des données, il est nécessaire de connaître avec précision le point de vue de l'utilisateur. Dans cette optique, la représentation devient un outil très intéressant à trois niveaux.

Lors de l'étiquetage, la représentation est une structure qui facilite la sélection des exemples à étiqueter pour assurer un apprentissage semi-supervisé de qualité. De plus, en interagissant avec une visualisation, l'utilisateur peut définir des portions de l'espace (en terme d'exemples) dont il ne souhaite pas s'occuper afin de concentrer les efforts de l'apprentissage sur les régions d'intérêt.

Lors de l'apprentissage, la structure la plus adaptée n'est pas nécessairement celle qui repose sur une mesure de distance calculée sur tous les attributs. En fournissant une visualisation des données et en autorisant l'utilisateur à interagir avec cette visualisation, il est possible de le laisser exprimer sa conception de la similarité en lui permettant de rapprocher ou d'éloigner certains exemples. Nous pensons qu'il est possible de récupérer

cette connaissance pour améliorer l'apprentissage.

Enfin, à la suite de ces deux processus d'étiquetage et d'apprentissage, la représentation possède un sens qui émerge de l'interaction d'un utilisateur avec elle. Il peut donc être intéressant de mémoriser cette représentation en vue d'une utilisation ultérieure.

Ce chapitre a montré que l'apprentissage à partir de représentations était suffisant pour assurer une prédiction de qualité, que ce soit en apprentissage supervisé ou semi-supervisé, grâce notamment aux approches ensemblistes. Pour rendre un tel apprentissage réellement compétitif vis-à-vis des autres approches, nous pensons qu'il reste maintenant à développer des interactions qu'un utilisateur peut avoir avec une telle représentation.