Validation statistique des cartes de Kohonen en apprentissage supervisé

Elie Prudhomme, Stéphane Lallich
Laboratoire E.R.I.C, Université Lumière Lyon 2
5, avenue Pierre Mendès-France, 69676 BRON Cedex – France
elie.prudhomme@etu.univ-lyon2.fr, stephane.lallich@univ-lyon2.fr

Résumé. En apprentissage supervisé, la prédiction de la classe est le but ultime. Plus largement, on attend d'une bonne méthodologie d'apprentissage qu'elle permette une représentation des données susceptible de faciliter la navigation de l'utilisateur dans la base d'exemples et d'aider au choix des exemples et des variables pertinents tout en assurant une prédiction de qualité dont on comprenne les ressorts. Différents travaux ont montré l'aptitude des graphes de voisinage issus des prédicteurs à fonder une telle méthodologie, ainsi le graphe des voisins relatifs de Toussaint. Cependant, la complexité de leur construction, en $O(n^3)$, reste élevée.

Dans le cas de données volumineuses, nous proposons de substituer aux graphes de voisinage les cartes de Kohonen construites sur les prédicteurs. Après un bref rappel du principe des cartes de Kohonen en apprentissage non supervisé, nous montrons comment celles-ci peuvent fonder une stratégie d'apprentissage optimisée. Nous proposons ensuite d'évaluer la qualité de cette stratégie par une statistique originale qui est étroitement corrélée au taux d'erreur en généralisation. Différentes expérimentations montrent la faisabilité de cette approche. On dispose alors d'un critère fiable pour sélectionner les individus et les attributs pertinents.

Mots-clefs : apprentissage supervisé, cartes de Kohonen, validation statistique

1 Position du problème

Les méthodes d'apprentissage supervisé d'une variable catégorielle ont pour objet *in fine* la prédiction de la classe d'appartenance d'un nouvel exemple à partir d'un échantillon d'exemples étiquetés. En fait, la prédiction n'est qu'une étape de la procédure d'apprentissage, qui est enrichie par l'analyse exploratoire des données tout à la fois pour les préparer au mieux et pour leur donner du sens en intégrant d'éventuelles informations contextuelles.

Dans une telle perspective, le recours aux graphes de voisinage apporte une solution efficace. On construit le graphe de voisinage issu des prédicteurs, par exemple le graphe des voisins relatifs de Toussaint (Toussaint et Menard, 1980), puis l'on colorie les sommets du graphe en fonction de leur classe d'appartenance. Pour trouver la classe d'un nouvel exemple, on insère celui-ci dans le graphe de voisinage et on lui attribue la classe majoritaire parmi ses voisins dans le graphe. Divers travaux ont proposé une statistique (le poids des arêtes coupées) qui évalue la capacité prédictive d'un graphe de voisinage et permet la sélection de variables pertinentes ou la détection

des individus atypiques en repérant l'impact d'un individu ou d'une variable sur la capacité prédictive du graphe ((Sebban, 1996), (Zighed *et al.*, 2001), (Lallich, 2002), (Muhlenbach *et al.*, 2003), (Zighed *et al.*, 2004)). Par rapport à la méthode des plus proches voisins (*kNN*), les graphes de voisinage adaptent l'étendue de la prise en compte des plus proches voisins à la topologie locale et la statistique qui évalue leur capacité prédictive est fortement liée à leur taux d'erreur en généralisation. Leurs résultats en généralisation sont au moins aussi bons et ils ont l'avantage de fonder une procédure efficace de navigation dans la base d'exemples.

La difficulté est la complexité des graphes de voisinage qui est en $O(n^3)$ pour la construction du graphe des voisins relatifs de Toussaint, ce qui les rend peu adaptés face à des données très volumineuses en termes d'individus. Plusieurs solutions sont possibles, notamment :

- diminuer le coût algorithmique de l'insertion d'un nouvel exemple, en considérant que dans la vie réelle la construction du graphe de voisinage sur l'ensemble d'apprentissage est acquise définitivement (Clech, 2004).
- construire le graphe de voisinage sur un échantillon de la base qui soit de taille raisonnable (Hacid, 2004).

Nous avons choisi une solution différente, fondée sur les cartes de Kohonen, qui s'efforce de conserver les avantages des graphes de voisinage (spatialisation de l'information apportée par les prédicteurs, évaluation de la qualité prédictive de cette information, navigation efficace dans la base d'exemples), tout en présentant une bien moindre complexité.

2 Notations

- m : le nombre d'individus, d : le nombre de prédicteurs, p : le nombre de modalités de l'étiquette (ou nombre de classes), n : le nombre de neurones.
- X : matrice (m, d) des individus ; la ligne i correspond à l'individu i, la colonne j au prédicteur j.
- y : vecteur à m composantes qui indique l'étiquette de chaque individu.
- W : matrice (n, d), de terme général w_{ij} qui désigne le poids du neurone i pour le prédicteur j.
- c : vecteur à n composantes qui indique l'étiquette de chaque neurone. $c_i = 0$ si le i^e neurone est ambigu, $c_i = -1$ si le i^e neurone est vide.
- K : matrice (n, p), de terme général k_{ij} qui correspond au nombre d'individus d'étiquette j que représente le neurone i.
- V : matrice symétrique (n, n) de terme général v_{ij} qui vaut 1 si le neurone i est voisin du neurone j sur la carte. v_{i+} représente le nombre de voisins du neurone i.
- $bmu_i = \arg\min_r \|w_r - x_i\|$, indice du neurone *best matching unit,* le plus proche de l'individu i.
- $dist_c(r, q)$: renvoie la distance au sens de la carte entre les neurones r et q ; si chaque neurone est un carré de coté 1, la distance au sens de la carte entre deux neurones r et q est la distance euclidienne entre les centres des carrés correspondant.

- $dist_p(r,q)$: fonction qui renvoie la distance euclidienne entre les poids des neurones r et q.
- $norm(d) = 1 - \frac{d}{max_d}$, fonction qui normalise la distance d ; max_d représente la distance maximum entre deux neurones de la carte.
- PPV : matrice symétrique (n,n), de terme général ppv_{ij} qui vaut 1 si $dist_c(i,j) \leq \max\left(dist_c(i,k), dist_c(j,k)\right), \forall k, k \neq i, k \neq j; c_i, c_j, c_k \neq -1$ (i,j connectés), 0 sinon ; ppv_{r+} représente le nombre de neurones connectés au neurone r.

3 Les *SOM*

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

L'algorithme classique de l'apprentissage du i^{eme} individu à l'instant t peut se résumer par la formule suivante de modification des poids w du neurone r :

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

où $h_r^t = \alpha^t \times v_r^t$, avec α^t le pas d'apprentissage qui décroît linéairement avec le temps et v_r^t la fonction de voisinage qui définit l'étendue des neurones modifiés autour du *bmu*. En début d'apprentissage, tout à la fois les neurones à modifier se rapprochent fortement des individus (α^t grand) et le nombre de neurones à modifier autour du *bmu* (le voisinage) est large (v^t grand). Par la suite, l'ampleur de la modification et le nombre de neurones à modifier décroissent.

Grâce à cet algorithme, on obtient une conservation de la topologie locale de l'espace des entrées. Pour deux individus, une proximité au sens des neurones correspond à une proximité dans l'espace de départ. La complexité des *SOM* est en $O(md)$ si l'on utilise un multiple de $\frac{n}{m}$ fois l'ensemble des individus pour l'apprentissage et que l'on choisit un nombre de neurones équivalent à $\sqrt{m}$ (voir (Vesanto, 2000) pour plus de détails). Ceci en fait l'un des algorithmes d'apprentissage non-supervisé les plus rapides.

4 Les *SOM* en apprentissage supervisé

La simplicité et l'efficacité de l'algorithme des *SOM* a conduit différents auteurs à l'adapter pour l'apprentissage supervisé. On présente ici les principales approches.

4.1 Les *LVQ*

Les *Learning Vector Quantization* (*LVQ*) proposés par (Kohonen, 1988) sont les plus couramment utilisés. Il en existe trois variantes dont on peut trouver une revue

par (Kohonen, 1998). Le principe est proche de celui des *SOM* mais les neurones sont étiquetés au départ par exemple avec l'algorithme des *k-Means*. Ils garderont cette étiquette tout au long de l'apprentissage. Une fois le *bmu* calculé, la modification des poids à la particularité de faire intervenir la comparaison entre la classe de l'individu et celle du *bmu*. Si ces classes sont identiques, le vecteur poids est modifié de telle sorte que le *bmu* se rapproche de l'individu, dans le cas contraire, de telle sorte qu'il s'en éloigne. Une fois l'apprentissage effectué, la classe d'un nouvel individu est égale à celle de son *bmu*.

4.2 Le modèle *LASSO*

Le modèle *LASSO* est proposé par (Midenent et Grumbach, 1994) a la particularité d'ajouter les indicatrices de classes aux prédicteurs et de calculer les *bmu* sur cette base. L'apprentissage de la *SOM* est ensuite réalisé de manière classique. Dans la phase de prédiction, un nouvel individu est présenté avec seulement la partie vectorielle correspondant aux prédicteurs. A partir de celle-ci, le modèle recherche le *bmu* et complète la partie manquante du nouvel individu par le vecteur des indicatrices de classe du *bmu*, réalisant ainsi la prédiction.

4.3 Les méthodes *Kohonen-KNN* et *Kohonen-WI*

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

Pour contourner ce problème d'autres travaux ont séparé l'apprentissage en deux phases. Une première phase réalise la construction de la *SOM* en se fondant uniquement sur les variables prédictives, la deuxième phase se charge d'étiqueter les neurones obtenus. L'étiquetage se fait en fonction de la classe la plus représentée dans le neurone et/ou son voisinage. C'est le cas de la méthode *Kohonen-KNN* (Zupan *et al.*, 1994), qui sera améliorée par *Kohonen-WI* (Song et Hopke, 1996). La différence entre ces deux méthodes réside dans la fonction de prédiction qui attribue l'étiquette à un nouvel individu. Au terme de l'apprentissage, on distingue les neurones qui représentent des individus (ayant donc une étiquette attribuée dans la deuxième phase) des neurones qui sont vides (n'ayant donc pas d'étiquette). Dans la phase de prédiction, la méthode *Kohonen-KNN* attribue son étiquette à un nouvel individu représenté par un neurone vide en fonction des neurones voisins de son *bmu* : l'étiquette majoritaire dans le voisinage est retenue. Deux problèmes en découlent. D'abord, les neurones voisins du *bmu* ne sont pas forcément ses plus proches voisins dans l'espace de départ (vecteur poids),

ce qui provoque une perte d'information due à la projection sur la carte. Ensuite il existe des cas indéterminés, lorsque les neurones voisins sont divisés équitablement entre les différentes classes. De son coté, *Kohonen-WI* se fonde sur l'interprétation des poids pour attribuer sa classe à un nouvel individu qui serait représenté par un neurone vide ou ambigu. L'étiquette retenue est celle du neurone voisin le plus proche du *bmu* retenu, en termes de distance euclidienne dans l'espace des entrées. La comparaison expérimentale de ces méthodes entre elles et avec les *LVQ* (Song et Hopke, 1996) fait apparaître que les méthodes *Kohonen-WI* et *LVQ* donnent les mêmes résultats. Par ailleurs, ceux ci sont clairement supérieurs à ceux de la méthode *Kohonen-KNN*. Ces résultats montrent donc qu'il est possible d'effectuer un apprentissage supervisé avec les cartes de Kohonen, apprentissage qui garde les propriétés de préservation de la topologie. On peut donc dissocier la projection des prédicteurs sur la carte, de la prédiction. Cependant, la méthode *Kohonen-KNN* n'a été validée que sur un jeu de données.

4.4 Kohonen-Opt

Définition La méthode *Kohonen-WI* perd deux types d'informations : celles relatives aux neurones ambigus et celles du voisinage d'un neurone sur la carte. Nous proposons d'améliorer cette méthode à l'aide d'une fonction de prédiction qui prend en compte ces informations, tout en gardant l'idée de l'interprétation des poids (*méthode Kohonen-opt*). Dans le cas où l'entrée est représentée par un neurone vide ou ambigu, on calcule un indicateur qui quantifie la présence de chaque classe en la pondérant par les distances des poids des neurones à l'entrée et ce pour la région qui s'étend à tous les plus proches neurones non vide du *bmu*. L'entrée présentée prend la valeur de la classe qui obtient l'indicateur maximum. Lorsque l'on cherche à prédire l'étiquette du vecteur d'entrée x_i, on peut résumer ce principe par la fonction suivante :

$$\begin{cases} \hat{Y} = \arg\max_{h \epsilon K}(P(h)), \\ P(h) = \begin{cases} c_{bmu_i}, & \text{si } c_{bmu_i} > 0 \\ norm(dist_p(x_i, w_{bmu_i})) * k_{bmu_i,h} & \\ + \sum_{j=1}^{ppv_{i+}} norm(dist_p(x_i, ppv_{bmu_i,j})) * k_{bmu_i,h}, & sinon \end{cases} \end{cases}$$

Données et paramétrage des *SOM* Les jeux de données proviennent de l'université de Californie à Irvine (Blake et Merz, 1998) sauf *Italian Olive Oil* qui est tiré de (Hopke et Massart, 1993). On trouve dans le tableau 1 le détail de ces jeux en termes de nombre de prédicteurs, de classes et d'instances.

Pour chaque jeu, le tableau 1 indique le nombre de cycle d'apprentissage *nb_appr* et les dimensions des *SOM* utilisés, toutes composées de neurones "carrés". L'apprentissage est quand à lui réalisé par l'algorithme classique défini en section 3.

Expérimentations Pour tester la validité de la méthode *Kohonen-Opt*, nous l'avons comparée à la méthode *Kohonen-WI* sur les 10 jeux de données présentés par le tableau 1. La comparaison s'effectue sur les résultats d'une 10-validation croisée, repris par le

Base	Prédicteurs	Classes	Instances	Dimensions	nb_appr
(1) Abalone	8	29	4177	25×25	90000
(2) Balance Scale	4	3	625	15×15	60000
(3) Breast Cancer	9	2	699	20×20	90000
(4) Glass Indent	9	6	214	10×10	10000
(5) Haberman	3	2	306	10×10	20000
(6) Ionosphère	34	2	351	10×10	20000
(7) Iris	4	3	150	10×10	2000
(8) Italian Olive Oil	9	9	572	15×15	45000
(9) Liver	6	2	345	10×10	35000
(10) Yeast	8	10	1484	25×25	90000

TAB. 1 – Description des bases et des hyperparamètres associés

Base	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
T.E. Opt	73,86	17,3	3,21	34,21	24,06	11,60	4,67	7,69	37,53	47.53
T.E. WI	74,37	19,5	3,5	34,46	24,73	12,27	5,34	8,04	43,19	48.88
T.E. ID3	74,17	23,39	4,25	28,96	26,47	8,55	4,00	5,94	42,03	39,82

TAB. 2 – Taux d'erreurs pour les méthodes Kohonen-Opt, Kohonen-WI, ID3

tableau 2, à partir d'une même carte étiquetée. On présente également les taux d'erreurs obtenus sur ces bases avec les arbres de décisions (algorithme *ID3* implémenté dans *Tanagra* par (Rakotomalala, 2003)).

Ces résultats montrent une plus grande efficacité de la méthode *Kohonen-Opt* par rapport à *Kohonen-WI* sur l'ensemble des bases de test. L'utilisation des informations de voisinage associées à celles des distances dans l'espace de représentation des voisins permet donc une meilleur interprétation de la *SOM* étiquetée. En outre les résultats de cette méthode sont comparables à ceux produit par les arbres de décisions (26.17 en moyenne pour *Kohonen-Opt* contre 25.76 pour *ID3*) avec autant de cas favorables que défavorables. *Kohonen-Opt* est donc une méthode de prédiction fiable qui montre que les *SOM* peuvent être utilisées en apprentissage supervisé dans de bonnes conditions.

5 Mesures de qualité pour les SOM en supervisé

Nous suggérons donc une stratégie d'apprentissage qui repose sur la construction de la carte de Kohonen issue des prédicteurs. La fiabilité de la carte obtenue, abstraction faite de la classe, peut être évaluée à l'aide de différents outils statistiques proposés notamment par (Bodt *et al.*, 2002). Nous proposons ici une évaluation de la capacité prédictive d'une telle carte par différentes statistiques dont nous montrerons expérimentalement (section 6) la forte corrélation avec la précision en généralisation. A l'image du poids des arêtes coupées (Lallich, 2002) utilisé pour les graphes de voisi-

nage, ces différentes statistiques sont fondées sur la notion de statistique *cross-product* (Mantel, 1967) qui est construite comme le produit scalaire de deux mesures de proximité, l'une liée aux prédicteurs, l'autre liée à la classe.

5.1 Définition des statistiques du type J

Pour évaluer la force du lien entre la proximité au sens des prédicteurs résumée par la carte et la proximité au sens de la classe, on peut raisonner sur les exemples ou sur les neurones, ce qui permet de s'abstraire de la volumétrie des exemples.

S'agissant des exemples, la proximité au sens des prédicteurs entre deux exemples est évaluée par la matrice T' de terme général t'_{ij} qui vaut 1 si les exemples i et j sont représentés par le même neurone, 0 sinon. Pour mieux prendre en compte les propriétés topologiques de la carte, on peut aussi utiliser la matrice T'' dont le terme général t''_{ij} vaut 1 si i et j sont représentés par le même neurone, $norm(dist_p(w_{bmu_i}, w_{bmu_j}))$ si i et j sont représentés par des neurones adjacents (soit $dist_c(bmu_i, bmu_j) = 1$), 0 sinon. La proximité au sens de la classe entre deux exemples est évaluée par la matrice U dont le terme général U_{ij} vaut 1 si les exemples i et j ne sont pas de la même classe, 0 sinon.

Dans le cas des neurones, on mesure la proximité au sens des prédicteurs entre deux neurones i et j par la matrice T''' de terme général t'''_{ij} qui vaut $norm(dist_p(w_i, w_j))$ si $ppv_{ij} = 1$, 0 sinon. La proximité au sens de la classe entre les neurones i et j est évaluée par la matrice R dont le terme général R_{ij} vaut 1 si i et j n'ont pas la même étiquette (soit $c_i \neq c_j$), 0 sinon.

On distinguera ainsi trois statistiques, J', J'' et J''' dont la définition est indiquée ci-dessous.

J'	**J"**	**J"'**
$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T'_{ij}U_{ij}$	$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T''_{ij}U_{ij}$	$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} T'''_{ij}R_{ij}$

Par ailleurs on utilise les notations simplificatrices suivantes, où T peut prendre la valeur de T',T'' où T''', les sommes se terminant alors respectivement en m pour les deux premiers cas, en n pour le dernier :

S_0	S_1	S_2
$\sum_{i=1}\sum_{j=1} t_{ij}$	$\frac{1}{2}\sum_{i=1}\sum_{j=1}(t_{ij} + t_{ji})^2$	$\sum_{i=1}(t_{i+} + t_{+i})^2$

Les statistiques du type J varient entre 0 et $\frac{1}{2}S_0$ et sont d'autant plus faibles que la liaison entre la proximité suivant la classe et la proximité suivant les prédicteurs portée par la carte est fortement positive. On pourra les standardiser en formant $2J/S_0$ qui varie entre 0 et 1.

5.2 Signification des statistiques du type J

Pour savoir dans quelle mesure l'évaluation donnée par J n'est pas liée au hasard, on définit un schéma aléatoire multinomial où l'hypothèse nulle (H_0) est que les exemples

(resp. les neurones) sont étiquetés indépendamment suivant une même distribution de probabilités $(\pi_r)_r$ où π_r désigne la proportion d'exemples de classe $y_r, r = 1, 2, \ldots, p$.

Il est alors possible de calculer la *p-value* unilatérale à gauche de J qui indique la probabilité d'obtenir sous H_0 une valeur de J aussi petite que celle observée. Ce calcul peut être fait par simulation ou plus rapidement par approximation normale (Cliff et Ord, 1981), ce qui impose de calculer $\mu = E(J/H_0)$ et $\sigma^2 = Var(J/H_0)$. Il est facile de calculer $\mu = S_0 \sum_{r=1}^{p-1} \sum_{s=r+1}^{p} \pi_r \pi_s$. On trouve dans (Lallich, 2002), suivant (Cliff et Ord, 1981), le calcul de la variance σ^2, dont la formule plus compliquée fait intervenir S_0, S_1 et S_2.

6 Expérimentations

Ces différentes statistiques ont d'abord été testées sur les 10 bases présentées en section 4.4 par le tableau 1. Le tableau 3 donne leurs valeurs, ainsi que celles des *p-value* associées et du taux d'erreur obtenu par 10-validation croisée sur la carte avec la méthode *Kohonen-Opt*.

L'analyse de ces résultats montre d'abord que les *p-value* sont significatives ($p < 0,05$) pour la statistique J'', de même pour J' (sauf dans le cas de la base *Haberman* où $p = 0,08$). Ces statistiques sont donc suffisamment robustes pour rendre compte de la qualité de la représentation faite par la *SOM* au terme de l'apprentissage. Dans le cas de J''' néanmoins, deux *p-value* sont proches de 1. Cette statistique considère toute arête qui part d'un sommet étiqueté ambigu comme une arête coupée. Dans les deux cas où la *p-value* est proche de 1, les graphes de voisinage qui résultent des cartes de Kohonen ont un grand nombre de sommets ambigus, donc un grand nombre d'arêtes coupées. On se rapproche alors d'un cas où les neurones auraient été étiquetés indépendamment de leur topologie, ce qui induit une valeur élevée de la *p-value*. En tout état de cause, cette sensibilité de J''' aux neurones ambigus ne se produit que pour des bases où le taux d'erreur en généralisation est très élevé relativement au nombre de classes.

On observe ensuite de fortes corrélations entre les différentes statistiques et le taux d'erreur. Parmi elles, la plus faible est celle issue de la statistique J' ($r^2 = 0.78$). Le détail des corrélations issues de J'' ($r^2 = 0,98$) et de J''' ($r^2 = 0,88$) est repris par la figure 1. Comme J' ne prend en compte que les individus d'étiquettes différentes représentés par un même neurone, elle ne tient pas compte l'information de topologie locale de la *SOM* étiquetée. Cette information est utilisée par J'', qui prend en compte les individus d'étiquettes différentes représentés par des neurones voisins suivant leur distance, ce qui explique sa meilleure corrélation avec le taux d'erreur. La statistique J''' présente un coefficient de corrélation intermédiaire. Elle prend en compte la topologie locale au travers d'un graphe de voisinage construit à partir des neurones de la carte. En revanche, les individus n'entrent plus dans son calcul, c'est le poids des arêtes coupées du graphe qui établit la statistique. Une certaine quantité d'information se perd donc lors de la projection, information relative à la distribution des individus dans leur espace de représentation. Cependant, la complexité du calcul de la statistique J''' est très avantageuse, surtout dans le cas de base d'exemples de grande taille, celui-ci se faisant uniquement à partir de la *SOM*. Ce n'est pas le cas de la statistique J'' qui a

Base	$2J'/S_0$	**p-value**	$2J''/S_0$	**p-value**	$2J'''/S_0$	**p-value**	**T.E.**
(1)	79,96	0	79,88	0	81,97	1	73,86
(2)	21,66	0	23,68	0	22,28	0	17,3
(3)	0,40	0	1,10	0	8,30	0	3,21
(4)	43,29	0	44,64	0	61,65	0,02	34,21
(5)	34,57	0,078	32,51	0,005	28,20	0,022	24,06
(6)	10,92	0	14,76	0	36,11	0,022	11,6
(7)	3,60	0	4,70	0	8,50	0	4,67
(8)	41,40	0	8,05	0	20,06	0	7,69
(9)	60,58	0,0031	0,3750	0	58,28	0,9989	37,53
(10)	50,00	0	52,40	0	64,52	0	47,53
Moyenne	31,64	0,0081	29,92	0,0005	27,99	0,2045	

TAB. 3 – Résultats sur 10 bases des statistiques de test, de leur *p-value* et du taux d'erreur en généralisation avec Kohonen-Opt

besoin de la base d'exemples pour être calculée. De même pour J', qui a de toute façon un coefficient de corrélation trop inférieur aux autres statistiques proposées.

Dans un second temps, nous avons testé la capacité de notre approche à faire face à une forte volumétrie des données. La base *Wave* de l'université d'Irwine en Californie (Blake et Merz, 1998) permet de générer aléatoirement des jeux de données dont le nombre d'individus évolue mais pour lesquels l'erreur en généralisation est connue. On a appliqué *Kohonen-Opt* et les statistiques de test à différentes bases *Waves* avec un nombre d'individus variant de 5 000 à 1 280 000. On a également chronométré le temps nécessaire à l'apprentissage. Les *SOM* utilisées sont identiques, seul change entre les bases le nombre de cycle d'apprentissage qui est égale à 4 fois le nombre d'individus. La validation est effectuée à partir d'un même fichier de 100 000 waves généré sur le même modèle.

Le tableau 4 présente les résultats obtenus. Il montre d'abord que le taux d'erreur en généralisation est stable avec l'augmentation du nombre d'individus (environ 15%) et que le temps nécessaire à l'apprentissage croit quasi linéairement d'un facteur 2 (comme le nombre d'individus). D'autre part, les statistiques ($2J/S0$) sont elles aussi stables avec une légère diminution de leurs valeurs au fur et à mesure que le nombre d'individus croît. Les *p-value* de ces statistiques sont quand à elles toujours nulles.

La qualité de l'apprentissage issu des *SOM* ne se détériore pas avec l'accroissement du nombre d'individus, ce qui justifie leur utilisation face à de grand jeux de données. Les statistiques proposées sont elles aussi adaptées : leur relative stabilité au regard de l'accroissement du nombre d'exemples montre qu'elles sont robustes suivant la taille de l'échantillon et permettent de valider un échantillon d'apprentissage de grande taille.

Taille	$2J'/S_0$	**p-value**	$2J''/S_0$	**p-value**	$2J'''/S_0$	**p-value**	**T.E.**	**Tps (s)**
1250	26,67	0	26,27	0	17,80	0	16,03	2
2500	26,08	0	26,09	0	13,39	0	15,70	5
5000	24,57	0	24,92	0	9,87	0	15,46	10
10000	24,45	0	24,36	0	9,44	0	15,57	20
20000	23,25	0	23,68	0	7,42	0	14,92	41
40000	22,65	0	23,04	0	7,78	0	14,84	78
80000	22,64	0	23,22	0	7,40	0	15,25	127
160000	22,53	0	23,03	0	7,45	0	14,93	245
320000	22,94	0	23,37	0	7,92	0	15,04	500
640000	22,54	0	23,09	0	7,18	0	15,17	1073
1280000	22,32	0	22,94	0	7,98	0	15,22	2014

TAB. 4 – Résultats sur les bases Waves des statistiques de test, de leur *p-value* et du taux d'erreur en généralisation avec Kohonen-Opt

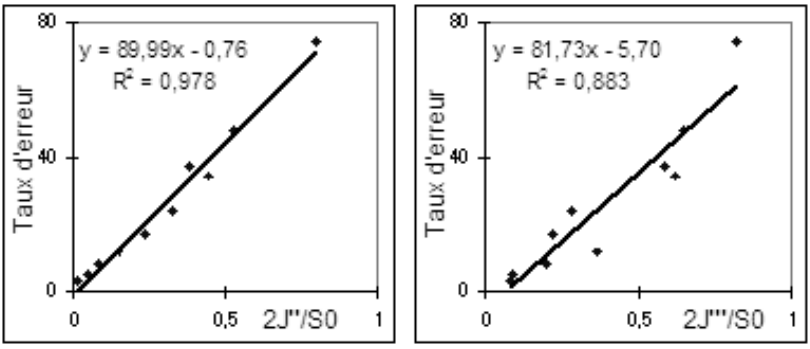

FIG. 1 – Corrélation du taux d'erreur avec J" (à gauche) et J"' (à droite)

7 Conclusion et perspectives

On connaît les qualités des cartes de Kohonen en apprentissage non supervisé, notamment leur complexité linéaire suivant le nombre d'exemples, l'existence d'outils mesurant leur fiabilité et leur aptitude à fonder une exploration des données (Lechevallier, 2002). Nous suggérons de les utiliser en apprentissage supervisé pour synthétiser l'information apportée par les prédicteurs lorsque l'échantillon d'apprentissage est de grande taille, assurant ainsi une bonne navigation dans la base d'exemples à l'image des graphes de voisinage. Nous leur associons *Kohonen-opt*, une procédure de prédiction efficace, $2J/S_0$, une statistique ayant une forte liaison linéaire avec le taux d'erreur en généralisation (analogue au complément à 1 d'un coefficient de détermination) et le test de signification correspondant. Nous avons entrepris l'utilisation des variations de cette statistique pour fonder une procédure de détection des exemples atypiques et de sélection de variables.

Références

Blake C.L. et Merz C.J., (1998). UCI repository of machine learning databases, 1998.

Bodt E., Cottrell M., et Verleysen M., (2002), Statistical tools to access the reliability of the self organizing maps. *Neural Network*, 15 :967–978, 2002.

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

Cliff A. D. et Ord J. K., (1981). *Spatial processes, models & applications*. London, 1981.

Hacid H., (2004), *Fouille de données dans les bases de données complexes*. Mémoire dea ecd, Université Lumière Lyon 2, Lyon : France, 2004.

Hopke P. K. et Massart D. L., (1993), Reference data sets for chemometrical methods testing. *Chemometrics and Intelligent Laboratory Systems*, 19 :35–41, 1993.

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

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

Kohonen T., (1998), The self-organizing map. *Neurocomputing*, 21 :1–6, 1998.

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

Lechevallier Y., (2002). Construction de super-classes à partir de la carte de kohonen et indicateurs de qualité de cette carte, séminaire laboratoire eric, http ://www-sop.inria.fr/axis/talks/∼eric/, 2002.

Mantel N., (1967), The detection of disease clustering and a general regression approach. *Cancer Res.*, 27 :209–220, 1967.

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

Muhlenbach F., Lallich S., et Zighed D.A., (2003), Identifying and handling mislabelled instances. *Journal of Information Intelligent Systems*, 22 :89–109, 2003.

Rakotomalala R., (2003). Tanagra [logiciel], http ://eric.univ-lyon2.fr/∼ricco/tanagra/index.html, 2003.

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

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

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

Vesanto J., (2000), *Using SOM in data mining*. Licentiate's thesis, Helsinki University of Technology, Helsinki : Finland, 2000.

Zighed D. A., Lallich S., et Muhlenbach F. Séparabilité des classes dans r^p. In *Actes du VIIIème Congrès de la Société Francophone de Classification (SFC'01)*, pages 356–363.

Zighed D. A., Lallich S., et Muhlenbach F. A statistical approach of classes separability. In T. Elomaa H. Mannila et H. Toivonen, editor, *Revue Applied Stochastic Models in Business and Industry*, pages 475–487.

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

8 Summary

In supervised learning, the prediction of the class is the ultimate goal. On a broader basis, a good learning methodology is expected to (1) enable a representation of the data in order to facilitate user's navigation within the data set and (2) contribute to the choice of examples and attributes, while ensuring a structured, understandable prediction. Various studies have shown how the so-called neighborhood graph, from the predictors, gives ground to such a methodology (e.g. : the relative neighborhood graph of Toussaint). However, the construction of such a graph ($O(n^3)$) remains complex.

In the case of high dimensionality data, we propose to substitute a self-organized map built on the predictors to the neighborhood graph. After a short reminder on the principles of the SOM for unsupervised learning, we analyze how it can found an optimized strategy of learning. Then we propose to use original statistics (narrowly correlated with the error in generalization) in order to assess the level of quality of this strategy. Diverse experiments highlight the feasibility of this approach, therefore reliable criterion are available for us to select relevant examples and attributes.

Keywords : supervised learning, Kohonen maps, statistical validation.