RÉGRESSION PAR CARTE TOPOLOGIQUE

F. Badran[1], P Daigremont[1] et S.Thiria[1,2]

1 CEDRIC, Conservatoire National des Arts et Métiers, 292 rue Saint Martin - 75 003 PARIS.(FRANCE)

2 Laboratoire d'Océanographie et de Climatologie (LODYC), Université de PARIS 6. 4 Place Jussieu, T14 - 75005 PARIS (FRANCE)

SUMMARY :

Smoothing is a problem currently addressed through statistical methods. Most recently, non parametric statistical techniques have been used, e.g. in smoothing with kernel functions, with splines, with k-nearest neighbors. The main difficulty with these techniques lies in the huge amount of computations required for their implementation. Today, operational methods allow to approximate the necessary values, so as to maintain a compromise between the obtained smoothing quality and the computation time.

Various Neural Network-based methods have been used for smoothing (multi-layer perceptrons, topological maps). We will show the theoretical links between the kernel method and the topological map method. This paper aims at showing the efficiency of supervised topological maps to process real data. Since this is a non-parametric technique, it is particularly interesting to graphically represent the obtained smoothing: this is why we will focus, in this paper, on functions of two variables.

1. Introduction

La régression constitue un des domaines abordé par la statistique, les solutions proposées utilisent des méthodes non paramétriques : régression par fonctions noyaux, par fonctions splines, par k-plus proches voisins. La principale difficulté rencontrée pour le développement de telles méthodes réside dans la complexité en nombre de calculs que nécessite leur mise en oeuvre. Les méthodes opérationnelles actuelles permettent d'approcher les valeurs cherchées de telle sorte qu'un bon compromis pour la régression soit trouvé entre précision et temps de calcul.
Différents modèles de réseaux de neurones permettent d'effectuer une régression : perceptrons multicouches [Rumelhart,95], fonctions radiales de base [Poggio. and. Giorsi, 1990] [Bishop,95]. Le but de cet article est de montrer l'efficacité de l'algorithme des cartes auto-organisatrice et de sa version supervisée pour traiter du problème de la régression. Il met en évidence les relations existantes entre la régression par fonctions noyaux et celle par cartes topologiques. L'une des propriétés importantes de la régression par carte topologique est d'intégrer tout à la fois les propriétés de la régression par fonctions noyaux et par k-plus-proches voisins.
Dans les premier et second paragraphes nous présentons la méthode de régression par fonctions noyaux et sa version rapide discrète (WARping [Hardle,91]). Le troisième paragraphe décrit l'algorithme des cartes topologiques et sa version supervisée qui permet d'effectuer la régression. Le quatrième paragraphe propose une analyse du fonctionnement de la carte, il établit les liens existants entre la régression par fonctions noyaux et la régression par k-plus-proches-voisins. Le cinquième paragraphe est

consacré aux expériences. Nous présentons, sur des exemples simulés, des comparaisons de performances entre la régression par carte topologique et celle par fonctions noyaux. Nous décrivons une application réelle choisie dans le domaine de l'océanographie et présentons les performances atteintes. La régression par carte topologique est utilisée ici pour produire des cartes de température de surface à partir de mesures effectuées sur l'océan.

2. Régression par fonctions noyaux

Les fonctions noyaux sont des fonctions ayant, en général, la caractéristique d'être symétriques par rapport à 0 et d'avoir une intégrale égale à l'unité. Leur principale propriété est d'être négligeable en dehors d'un domaine compact.
Une fonction noyau de dimension 1 permet de générer une famille de fonctions noyaux par l'intermédiaire d'un paramètre de lissage h :

$$K^h(x) = \frac{1}{h} K\left(\frac{x}{h}\right)$$

Nous utiliserons pour les expériences des fonctions noyaux K() en dimension deux, elles effectuent le produit tensoriel de deux fonctions noyaux de dimension un :

$$K(x, y) = K_1(x) \times K_2(y)$$

Les paramètres de lissage utilisés en dimension 2 peuvent être différents dans chacune des dimensions x ou y. La forme générale d'une fonction noyau de dimension 2 est donc :

$$K^H(x, y) = \frac{1}{h_1}\frac{1}{h_2} K\left(\frac{x}{h_1}\right) K\left(\frac{x}{h_2}\right)$$

$H = (h_1,h_2)$ représente le vecteur de lissage de la fonction noyau considérée.

Bien que le raisonnement soit généralisable, nous limiterons notre étude à la dimension 2, le but de la régression sera donc de trouver une fonction exprimant la dépendance entre les variables explicatives (x,y) d'une part et la variable expliquée r d'autre part.
La fonction recherchée est définie à partir d'un ensemble d'observations : l'ensemble d'apprentissage noté $App = \{(x_i, y_i, r_i); i = 1..n\}$. Les deux ensembles $\{(x_i, y_i) \quad i = 1..n\}$ et $\{r_i; i = 1..n\}$ représentent des mesures différentes d'un même phénomène et l'on cherche à savoir comment les valeurs (x_i,y_i) permettent d'expliquer r_i.
La relation recherchée est classiquement représentée par le modèle :

$$r_i = m(x_i, y_i) + \varepsilon_i \quad i = 1..n$$

Où ε_i est une variable aléatoire de moyenne nulle qui permet de caractériser la variation de la variable aléatoire R autour d'une surface moyenne :

$$m(x, y) = E(R/X = x, Y = y) = \frac{\int r\, f(x, y, r)\, dr}{f(x, y)}$$

Cette surface moyenne est fonction de la densité du triplet (X, Y, R) (notée f(x,y,r)) et de la densité de la loi marginale du couple aléatoire (X,Y) (notée f(x,y)). Le nombre de mesures utilisées étant fini, on supposera par la suite qu'on recherche cette fonction sur un domaine borné : *Dom* =[a, b] x [a', b'].
Le problème est donc d'estimer la fonction moyenne m(x,y) conditionnée par les observations sur le domaine d'étude. La régression par fonctions noyaux estime celle-ci par la fonction $m^H(x,y)$ définie par (1):

$$m^H(x,y) = \frac{1}{n}\sum_{j=1}^{n} r_j W_j^H$$

Où $W_j^H = \frac{1}{n}\frac{K^H(x-x_j, y-y_j)}{f^H(x,y)}$ et $f^H(x,y)$ représente une estimation de la fonction densité f(x,y). L'estimation de la densité f(x,y) à l'aide des mêmes fonctions noyaux K^H permet d'obtenir l'estimateur de Nadaraya-Watson [Nadaraya, 64] (2):

$$m^H(x,y) = \frac{\sum_{j=1}^{n} K^H(x-x_j, y-y_j) r_j}{\sum_{j=1}^{n} K^H(x-x_j, y-y_j)}$$

qui est un estimateur consistant de la fonction de régression m(x,y) cherchée ($m^H(x,y) \xrightarrow{p} m(x,y)$ quand $H \to (0,0)$ et $nH \to (\infty,\infty)$). Ainsi les poids W_j^H dépendent du vecteur de lissage H = (h_1,h_2) et de la distance entre le point M(x,y) considéré et le point M(x_j,y_j) de l'ensemble d'apprentissage. Les valeurs h_1 et h_2 de H déterminent le degré de lissage que l'on peut obtenir, plus h_p est grand et plus la projection de la courbe obtenue dans la direction p est lisse. Le calcul direct de $m^H(x,y)$ demande un nombre de calculs très important puisqu'il implique pour chaque valeur (x,y) de considérer l'ensemble d'apprentissage dans son ensemble. Quand l'ensemble d'apprentissage est très grand, le problème est donc d'obtenir de bonnes approximations de (2) calculables en un temps raisonnable. Nous présentons au paragraphe 2 des méthodes d'approximation de cet estimateur.

2. L'approche par discrétisation

Une approximation de l'estimateur de Naradaya-Watson consiste à utiliser la méthode des histogrammes. On divise le domaine d'étude *Dom* en N_1xN_2 domaines rectangulaires réguliers F_{ij} de même dimension d=(d_1,d_2) et l'on estime sur chaque domaine : $\overline{m_{ij}} = E\left(R / (X,Y) \in F_{ij}\right)$.
Si on note par c_{ij} le centre du domaine F_{ij}, l'ensemble D des couples : $D = \left\{\left(c_{ij}, \overline{m}_{ij}\right); 1 \le i \le N_1, 1 \le j \le N_2\right\}$ réalisent une discrétisation du nuage de points initial en N_1xN_2 points. La régression s'effectue alors à partir de ces N_1xN_2 couples régulièrement disposés dans l'espace. Cette régression peut être définie de deux manières différentes :
• constante sur chaque rectangle Fij et égale à $\overline{m_{ij}}$.

• par interpolation linéaire effectuée à partir d'une triangulation de *Dom.* On utilise alors l'une des deux triangularisations évidentes définies par les c_{ij} régulièrement répartis.

Le problème est maintenant d'estimer $\overline{m}_{ij}$ au mieux en fonction des observations.

On définit par :

- n_{ij} le nombre d'observations $P_k = (x_k, y_k)$ qui appartiennent à F_{ij}
- S_{ij} = **Erreur!** F_{ij}; ; r_k)

une façon simple d'estimer $\overline{m}_{ij}$ consiste à choisir la moyenne empirique de chaque domaine: $\overline{m}_{ij} = \frac{S_{ij}}{n_{ij}}$. Cette estimation dépend alors beaucoup du découpage: la surface lissée est influencée par l'origine des intervalles et par les dimensions du vecteur de discrétisation d=(d_1,d_2) choisi. D'autre part, quand on utilise des données réelles, beaucoup de domaines peuvent être vides. Il n'est plus possible d'attribuer de valeur à S_{ij} et $\overline{m}_{ij}$ prend une valeur nulle ou n'est pas définie.

La méthode de WARPing plus robuste pour estimer cette valeur moyenne a été proposée en tant qu'alternative [Härdle, 90], cette méthode effectue à l'aide des fonctions noyaux une moyenne mobile sur les points discrétisés. Par exemple, si on choisit une fonction noyau à support sur [-1,+1], l'estimation par WARPing de la valeur $\overline{m}_{ij}$ cherchée est calculée à l'aide de la formule suivante (3):

$$m_{ij}^{M_1 M_2} = \frac{\sum_{p=1-M_1}^{M_1-1} \sum_{q=1-M_2}^{M_2-1} K^H\left(\frac{p}{M_1}, \frac{q}{M_2}\right) S_{i+p,j+q}}{\sum_{p=1-M_1}^{M_1-1} \sum_{q=1-M_2}^{M_2-1} K^H\left(\frac{p}{M_1}, \frac{q}{M_2}\right) n_{i+p,j+q}}$$

Où les nombres M_1 et M_2 considérés correspondent au vecteur de lissage H introduit précédemment à travers les relations $h_1 = M_1 d_1$ et $h_2 = M_2 d_2$. Le nombre $m_{ij}^{M_1 M_2}$ représente une approximation de l'estimateur de Nadaraya-Watson (2) au point c_{ij}. La différence entre un estimateur et la surface théorique décroît avec la finesse de la discrétisation.

Le nombre limité de points de discrétisation permet l'utilisation de moyennes mobiles. Dans la formule (3), le dénominateur est un lissage par fonctions noyaux au point c_{ij} de l'ensemble des effectifs $\{n_{pq}, 1 \le p \le N_1, 1 \le q \le N_2\}$ et le numérateur est un lissage de la suite $\{S_{pq}, 1 \le p \le N_1, 1 \le q \le N_2\}$ en ce même point. L'approche par WARPing permet d'affecter à chaque domaine de dicrétisation F_{ij} et au point c_{ij} une valeur calculée en fonction des domaines avoisinants. En particulier on attribue aux domaines vides ou faiblement représentés des valeurs $\overline{m}_{ij}$ qui prennent en compte des observations plus éloignées. Si l'on compare à l'estimation précédente par la moyenne, la formule (3) donne une valeur à $\overline{m}_{ij}$ qui prend en compte un plus grand nombre d'observations.

Cependant, à cause des domaines réguliers utilisés pour la discrétisation l'approche proposée est plus adaptée aux ensembles d'apprentissage pour lesquels la projection sur *Dom* est uniformément distribuée. En effet, pour calculer la valeur de la régression au point (x,y), les fonctions noyaux réalisent une moyenne mobile des observations dans un voisinage de (x,y) de taille constante. Ce point faible de la méthode est abordé par une autre méthode, celle des k-plus-proches-voisins qui réalise une régression par

fonctions noyaux à l'aide de paramètres de lissage variables en fonction de la densité locale des exemples [Härdel, 91].

La régression par cartes topologiques que nous présentons maintenant permet d'intégrer les propriétés de la régression par fonction noyau et la rapidité de la méthode de WARping. Pour ces raisons, elle représente une alternative intéressante dans le traitement des applications réelles qui utilisent un très grand nombre de données.

3. Régression par carte topologique

Nous présentons dans un premier temps la version supervisée de l'algorithme de Kohonen (Supervised Organizing Map SOM) [Ritter and Schulten, 92] qui permet de mettre en relation les variables explicatives (x,y) avec la variable dépendante r. Nous discutons par la suite des problèmes rencontrés dans le déroulement de l'algorithme et des améliorations possibles. La présentation générale des cartes topologiques a été faite dans ce même volume [Anouar & all, 96]. Nous renvoyons à ce chapitre pour le déroulement de l'algorithme général et pour les notations. En particulier, Φ représente la fonction d'affectation qui permet le choix du neurone à partir duquel s'effectue l'apprentissage.

3.1 L'algorithme

Nous utilisons pour la régression une carte topologique classique dont :

- La couche d'entrée comporte trois neurones entièrement connectés aux neurones de la carte qui reçoivent les valeurs $\{(x_i,y_i,r_i);\ 1\le i\le n\}$ des observations de l'ensemble d'apprentissage.
- La couche de sortie est composée de N1xN2 neurones

Chaque neurone c de la carte est donc relié par trois coefficients synaptiques à cette couche d'entrée.
On utilise de préférence pour la régression une grille hexagonale où chaque neurone possède six voisins directs. Le graphe G dont les sommets sont les neurones de la carte et les arêtes celles qui relient deux neurones "voisins directs" est composé de triangles. Le système de voisinages ou topologie de la carte permet donc la mémorisation d'une triangularisation du plan (x,y). La distance, sur la carte, entre deux neurones c_i et c_j est définie comme étant la longueur du plus court chemin sur le graphe G de c_i à c_j. Afin de contrôler la taille du voisinage d'un neurone c de la carte, nous utilisons la fonction noyau suivante (4):

$$K^h(c_i,c_j) = \exp - \left(\frac{\delta(c_i,c_j)}{h}\right)^2$$

où $\delta(c_i,c_j)$ représente la distance sur le graphe G et h contrôle la taille du voisinage. Cette fonction qui s'applique sur les neurones de la carte est similaire aux fonctions K^h introduites au chapitre précédent.

On note par la suite C l'ensemble des neurones de la carte et $W_c = (w_c^1, w_c^2, w_c^3)$ le vecteur poids associé au neurone c.

L'algorithme de régression, qui est une variante de l'algorithme de Kohonen, est alors le suivant :

Algorithme SOM-1
Initialisation t = 0;
Étape 1. Présenter un exemple (x_i,y_i,r_i). La fonction d'affectation φ sélectionne le neurone $\varphi(i)$ pour lequel la projection du vecteur poids sur l'espace des données (x,y) est la plus proche du vecteur (x_i,y_i) au sens de la norme euclidienne.
Étape 2. Pour tout neurone c de la carte effectuer une itération d'apprentissage non supervisée :

$$w_c^1(t+1) = w_c^1(t) + \alpha(t)K^{h_1(t)}(c,\varphi(i))(x_i - w_c^1(t))$$
$$w_c^2(t+1) = w_c^2(t) + \alpha(t)K^{h_2(t)}(c,\varphi(i))(y_i - w_c^2(t))$$

Étape 3. pour tout neurone c de la carte effectuer une itération d'apprentissage supervisée:

$$w_c^3(t+1) = w_c^3(t) + \alpha'(t)K^{h(t)}(c,\varphi(i))(r_i - w_c^3(t))$$

Étape 4. t=t+1, l'algorithme s'arrête lorsque t atteint une valeur maximale choisie.

Dans cet algorithme, K^h est la fonction noyau définie dans la relation (4), h_1, h_2 et h des paramètres de lissage qui sont des fonctions décroissantes de t de manière à faire décroître la taille du voisinage, α et α' les pas d'apprentissage.
Les étapes 2 et 3 s'effectuent de manière séquentielle à chaque itération. Il est cependant possible de dissocier ces deux étapes en procédant par blocs, en adaptant tout d'abord les poids (w_c^1,w_c^2), puis le troisième poids w_c^3 . Sous cette forme, l'algorithme de régression devient :

Algorithme SOM-2
Etape 2' (non supervisée):
Rechercher un bon maillage pour la régression qui fait intervenir de manière compétitive différents facteurs : densité des observations, représentation homogène du domaine de la fonction ....
Geler pour toute cellule c les poids (w_c^1 , w_c^2).

Etape 3' (supervisé) :rechercher une régression optimale sur ce maillage.

Les expériences menées au paragraphe 5 montrent que les performances obtenues par SOM-1 et SOM-2 sont similaires. La version SOM-2 sera utilisée au paragraphe 4 afin d'analyser l'algorithme de régression.

En fin d'apprentissage les deux versions de l'algorithme SOM produisent une discrétisation définie à partir de l'ensemble des couples: $D = \left\{\left(\left(w_{c_i}^1,w_{c_i}^2\right),w_{c_i}^3\right) \quad 1 \le i \le N\right\}$.
La troisième composante représente la valeur de la régression au point de discrétisation : $(\overline{m}_c = w_c^3)$. Cependant l'induction de la topologie du graphe G de la carte sur l'ensemble des points $D'=\left\{\left(w_{c_i},w_{c_2}\right), 1 \le i \le N\right\}$ du plan (x,y) déterminé par SOM ne forme pas nécessairement une triangularisation de *Dom*. La régression obtenue aux points de discrétisation ne permet donc pas de reconstruire la fonction

recherchée par interpolation linéaire. La variante proposée par Najafi et Cherkasky [Cherkasky, 91] assure que le graphe G induit sur D' une triangularisation de *Dom.* L'algorithme commence par initialiser les poids de manière à ce qu'ils réalisent une triangularisation et conserve cette propriété au cours des différentes itérations de la phase d'apprentissage. Les cellules c sont initialisées d'une façon régulière en choisissant les couples (w_c^1, w_c^2) aux centres des domaines réguliers et les poids w_c^3 sont choisis nuls. Cette variante de SOM s'obtient en rempaçant simplement l'étape 1 par la nouvelle étape 1', les étapes 2 et 3 (ou 2'et 3') restant identiques:

Algorithme SOM-3
Étape 1'.
Présenter un exemple (x_i, y_i, r_i).
Déterminer le triangle formé des points de D' qui contient sa projection P=(x_i, y_i) :
Sélectionner, à l'aide de la fonction d'affectation φ, la cellule $\varphi(i)$ qui correspond au sommet du triangle le plus proche de P.

3.2 Mise en oeuvre de SOM-3

Clairement les étapes 2 et 3 de l'algorithme précédent ont des rôles différents :
• A l'étape 2 les couples (w_c^1, w_c^2) se répartissent selon la densité des exemples dans le plan (x,y). Il s'agit d'une itération de l'algorithme classique de Kohonen. Le critère de choix utilisé à l'étape 1' permet simplement de reconstruire un maillage en utilisant la topologie induite par le graphe G.
• L'étape 3 permet d'adapter la troisième composante w_c^3 du neurone c en fonction de la variable dépendante r et du maillage en cours obtenu à l'étape 2.

L'utilisation directe de l'algorithme SOM, sans précautions préalables, peut concentrer une trop grande proportion de neurones dans les régions de fortes densité du plan (x,y) et donc entraîner une dégénérescence de la carte C. Un tel découpage réalise une mauvaise représentation de la fonction dans les régions de faible concentration ainsi que sur les bords où le voisinage est difficile à contrôler. Dans ces cas un WARPing utilisant le même nombre de points de discrétisation répartis de façon régulière permet en moyenne d'obtenir une meilleure régression. Le phénomène de dégénérescence dépend en bonne partie de la façon dont H varie : Une valeur initiale trop grande pour H amène la carte à se concentrer trop rapidement et détériore irrémédiablement la régression. Une valeur de H trop faible ne permet aucune concentration et produit une carte dont le découpage est identique à celui du WARPing (découpage régulier), les performance sont alors identiques pour les deux méthodes. Une bonne régression sera donc obtenue en contrôlant la valeur initiale de H et sa décroissance. La dégénérescence sera de plus évitée en forçant à chaque itération les neurones du bord de la carte C à recouvrir *Dom.* On introduit alors à l'étape 2 une contrainte qui force les neurones de la périphérie à rester sur le bord initial de *Dom.* Une coordonnée, w_c^1 ou w_c^2 selon le segment du bord, reste fixe durant l'apprentissage.

4. Analyse de la régression par carte topologique:

Nous nous plaçons maintenant dans le cadre de SOM-2 et au début de l'étape 3'. Si l'on ne varie pas le voisinage en fonction du temps, soit $K^{h(t)} = K^h$. L'Étape 3' peut être interprétée comme une optimisation par une méthode de gradient stochastique de la fonction coût suivante (5) :

$$E(\varphi, W) = \frac{1}{2} \sum_{i \ App} \sum_{c \ C} K^h(c, \varphi(i)) \left(w_c^3 - r_i \right)^2$$

Un minimum local est atteint lorsque $\frac{\partial E}{\partial w^3} = 0$ pour tout neurone c. Ainsi pour un neurone particulier c cette condition devient :

$$\frac{\partial E}{\partial w_c^3} = \sum_{i \ App} K^h(c, \varphi(i)) \left(w_c^3 - r_i \right) = 0$$

Cette expression peut être décomposée par rapport à l'ensemble des Voronoï $\{F_b / b \quad C\}$ où $F_b = \{P = (x,y) / \|P - W_b\| \leq \|P - W_r\| \ r \quad C \text{ et } r \neq b\}$, $W_b = \left(w_b^1, w_b^2 \right)$ et $\| \ \|$ est la distance euclidienne.

$$\frac{\partial E}{\partial w_c^3} = \sum_{b \ C} \sum_{i \ F_b} K^h(c, b) \left(w_c^3 - r_i \right) = 0$$

$$\sum_{b \ C} \sum_{i \ F_b} K^h(c, b) \ w_c^3 = \sum_{b \ C} \sum_{i \ F_b} K^h(c, b) \ r_i$$

$$\sum_{b \ C} n_b K^h(c,b) w_c^3 = \sum_{b \ C} K^h(c,b) \sum_{i \ F_b} r_i$$

Où n_b représente le nombre d'exemples de F_b.
Si nous notons $S_b = \sum_{i \ F_b} r_i$ nous obtenons alors la solution (6):

$$w_c^3 = \frac{\sum_{b \ C} K^h(c, b) \ S_b}{\sum_{b \ C} K^h(c,b) \ n_b}$$

Qui montre que (6) est une solution unique de (5).
Les itérations de l'étape 3' effectuées en gardant constant le paramètre h de la fonction $K^h(c, b)$ font évoluer les valeurs de w_c^3 vers la valeur définie à la relation (6). Cette

formule intègre, pour le calcul de w_c^3, tous les exemples des domaines de Voronoï associés aux cellules de la carte se trouvant au voisinage de la cellule c.
L'étape 2' correspond à l'algorithme non supervisé de Kohonen qui place les référents en tenant compte de la densité du nuage de points [Kohonen, 95]. Dans les régions de forte densité l'algorithme concentre les référents et engendre des domaines de Voronoï de taille réduite, alors que dans les régions de faible densité il place peu de référents et engendre des domaines de Voronoï de grande taille. Dans l'espace des données, l'expression de w_c^3 s'apparente est une régression par fonctions noyaux sur des domaines de tailles variables adaptés à la concentration des données. Chaque domaine sera donc estimé avec un nombre suffisant d'exemples, ce qui doit garantir une meilleure estimation. Le lissage est réduit dans les régions de forte densité et se trouve important dans les régions sous représentées. De ce point de vue, SOM ressemble à la méthode de régression par fonctions noyaux avec des paramètres de lissage dans l'espace des données variables dépendant de la densité locale des exemples.
La pondération $K^h(c, b)$ dépend de la distance entre les deux cellules c et b sur la carte. Lorsque le voisinage défini par la fonction $K^h(c, b)$ se réduit au seul neurone c la formule (6) devient alors $w_c^3 = \frac{S_c}{n_c}$, dans ce cas elle fournit la moyenne empirique sur chaque domaine.

Le paragraphe suivant présente une série d'expériences sur des exemples simulés et sur des données réelles qui permettent de valider l'utilisation de l'algorithme SOM modifié pour la régression de fonctions.

5. EXPERIENCES :

Nous présentons deux séries d'expériences :

• La première sur des données simulées de manière à pouvoir calculer l'erreur vraie effectuée par le lissage proposé. Nous montrerons à partir des performances comparées l'intérêt de la méthode.

• La seconde sur des données réelles fortement bruitées. Les données à traiter représentent les mesures océanographiques de la surface des mers au mois de Juillet depuis le début du siècle dans la zone atlantique nord.

1° données simulées

Nous avons simulé trois fonctions F(x,y), la figures 1 représente les trois fonctions.. Elles sont définies sur [0,1]x[0,1] par les équations suivantes :

$F^1(x,y) = 1/9\ (\tanh(9y-9x)+1$

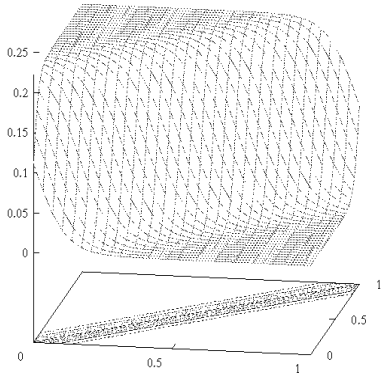
$F^2(x,y)=1/3 \exp\{81/4\ [(x-0.5)^2 + (y-0.5)^2]\}$

$F^3(x,y)= \tanh\{-3g(x,y)\}+1$ avec $g(x,y) = 0.595576*(y + 3.79762)^2 - x - 10$

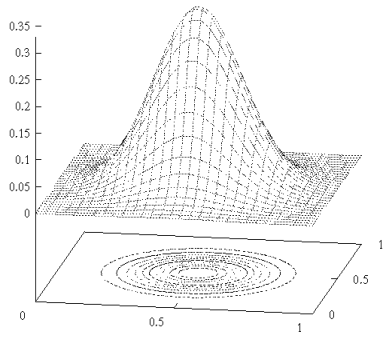
Nous avons généré les ensembles d'apprentissage $App = \{(x_i,y_i,r_i)\}_{i=1}^{n}$ en tirant, dans l'ordre, les abscisses et les ordonnées des points selon deux lois différentes non

indépendantes : x suit une loi uniforme sur [0,1] et y la loi normale N(x, 1/3). Trois jeux de données ont été générés à partir des trois fonctions précédentes en ajoutant un bruit gaussien N(0; 0.02). Les différents ensembles d'apprentissage relatifs à chaque fonction sont constitués de 2000 points. Nous avons considéré deux ensembles de test distincts sur lesquels nous avons calculé l'erreur quadratique moyenne entre l'estimation produite par la régression et la valeur de la fonction initiale. Le premier ensemble TEST-1 est constitué de 900 points répartis sur une grille régulière (30x30) recouvrant le domaine de la carte [0,1]x[0,1]. Cet ensemble permet de juger de la qualité de la courbe reconstruite. Le second ensemble TEST-2 est constitué de 600 points tirés selon la même distribution que celle de l'ensemble d'apprentissage. La figure 2 donne une représentation de l'ensemble d'apprentissage dans le plan (x,y). La valeur de la régression de la carte a été obtenue par extrapolation linéaire sur la triangularisation obtenue.

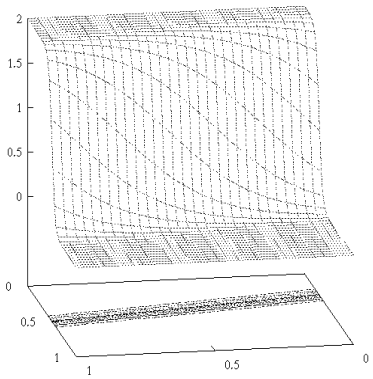
Dans tous les cas nous avons effectué la comparaison des performances en effectuant la méthode de WARPing : régression par fonction noyaux et détermination des valeurs optimales des paramètres h_1 et h_2 par validation croisée sur un ensemble indépendant. Nous avons utilisé pour les deux méthodes (régression par carte topologique et WARPing) le même nombre d'intervalles. La carte topologique utilisée est carrée et contient 10x10 neurones, Le WARPing est effectué sur un découpage régulier de 10x10 secteurs.



(a)



(b)



(c)

Figure 1: (a), (b),(c) représentent respectivement les fonctions F_1, F_2, F_3 ainsi que les courbes isobares dans le plan (x,y)

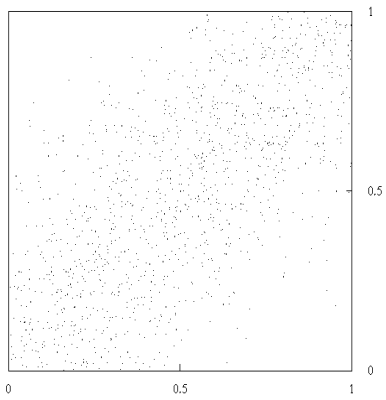


Figure 2 : Distribution des points de l'ensemble d'apprentissage sur le plan (x,y).

Les performances des deux méthodes sont comparées à l'aide d'un critère d'erreur qui calcule l'écart quadratique par rapport à la courbe théorique. Nous avons utilisé le critère RMS calculé sur un ensemble de test de taille p défini par la formule :

$$\mathrm{RMS(Re)} = \sqrt{\frac{1}{p}\sum_{i=1}^{p}\left(F^{j}(x_i, y_i) - \mathrm{Re}(x_i, y_i)\right)^2}$$

Où Re représente la fonction de régression considérée (SOM, WARPing). La table 1 donne les performances comparées des deux méthodes SOM et WARPing pour les trois fonctions F^1, F^2, F^3 sur les deux ensembles de test TEST-1,TEST-2. On peut constater que les performances des deux méthodes sont comparables sur TEST-2. TEST-2 suit la même distribution que l'ensemble d'apprentissage, les deux ensembles possèdent beaucoup d'exemples dans les zones fortement représentées. Dans ces zones, la méthode de WARPing est bien échantillonnée et permet une bonne régression, les zones faiblement représentées n'influent presque pas sur le critère RMS. L'efficacité de la régression sur ces zones peut être comparée à partir des performances obtenues sur TEST-1 qui contient des exemples uniformément répartis et dont une bonne partie se trouvent situés dans les zones qui ont faiblement participé à l'apprentissage. Ces performances montrent que la fonction obtenue par régression par carte topologique est meilleure pour les trois fonctions. L'extrapolation de la fonction proposée est donc meilleure pour la méthode neuronale qui tire un meilleur parti de la densité des exemples d'apprentissage.

SOM	F^1	F^2	F^3
TEST-1	0.006	0.011	0.061
TEST-2	0.019	0.022	0.069
WARP	F^1	F^2	F^3
TEST-1	0.021	0.019	0.158
TEST-2	0.02	0.019	0.07

Table 1 : performances comparées (critère RMS) des deux méthodes SOM et WARPing pour les trois fonctions F1, F2, F3 sur les deux ensembles de test

2° données réelles

Nous présentons maintenant une application de SOM à des données réelles. Les données à traiter représentent les mesures océanographiques de la température de la surface des mers au mois de juillet depuis le début du siècle dans la zone atlantique nord.

L'ensemble des données est composé de 54853 échantillons. Pour chaque mesure de température de la surface de l'eau, on dispose de la longitude et de la latitude du point de mesure ainsi que de l'année à laquelle elle a été effectuée.

Comme le montre la répartition spatiale de l'ensemble des mesures présentées figure 2.a, les observations ne sont pas spatialement réparties de façon homogène. On remarque particulièrement des points de forte concentration le long des côtes européennes et dans certaines zones du nord est de l'atlantique. Des campagnes de mesure intensives ont été effectuées certaines années en des régions très précises, ces années sont alors sur représentées dans l'ensemble des données. 0n remarque, en particulier, les années 1962 à1964 qui possèdent plus de 4000 points de mesure alors que la moyenne par an est proche de 600. La figure 3.a donne l'histogramme de l'ensemble des mesures en fonction de l'année.

Afin d'obtenir une fonction de régression donnant la température de surface en fonction de la position géographique qui dépende le moins possible des campagnes de mesures, l'ensemble d'apprentissage Z_{app} a été constitué à partir d'un quadrillage régulier de l'espace (ici 40×40, soit 1600 secteurs). On a choisi au hasard deux mesures par année dans chaque secteur, Z_{app} possède alors 16464 échantillons. On peut remarquer que la répartition spatiale de Z_{app} est maintenant plus homogène (figure 2.b) mais respecte cependant les statistiques de base des températures (moyenne, écart type des températures par années). La figure 3.a montre l'histogramme des répartition des mesures par année pour l'ensemble global et la figure 3.b donne cette répartition pour Z_{app}, les figures 4.a et 4.b montrent la moyenne et l'écart type par année pour ces deux ensembles. Clairement les deux ensembles ont des statistiques proches.

Afin de pouvoir juger de l'approximation obtenue nous avons effectué une régression à l'aide de la méthode de WARPing mise en oeuvre dans des conditions optimales (détermination du rayon de lissage par validation croisée). Nous avons effectué l'apprentissage de Z_{app} sur une carte de 40x40 neurones et le WARPing dans les mêmes conditions que précédemment en utilisant 40x40 intervalles réguliers.

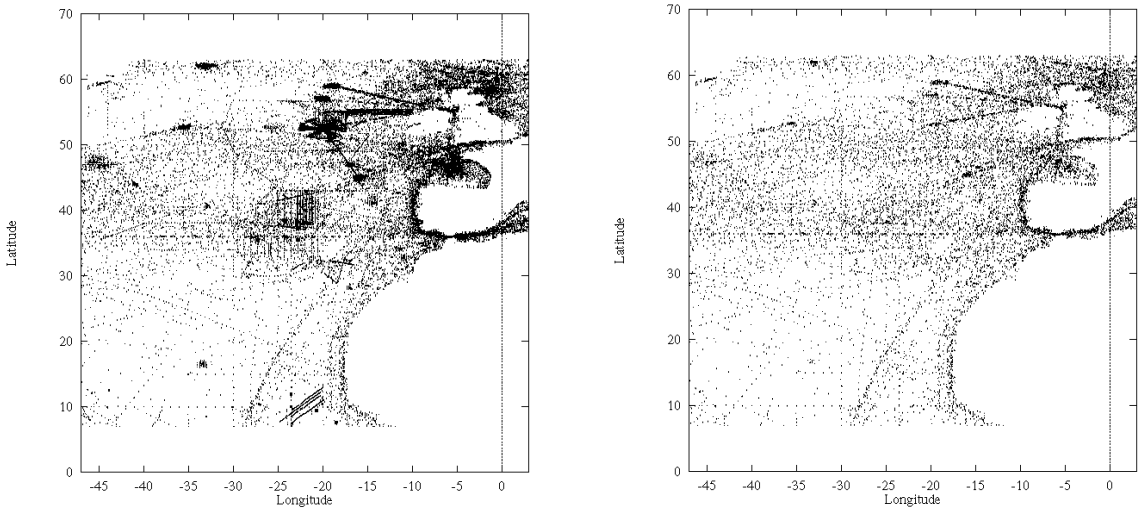


figure 2 : répartition spatiale des données. 2(a) à gauche répartition pour toutes les mesures 2(b) à droite répartition sur l'ensemble d'apprentissage Z_{app}.

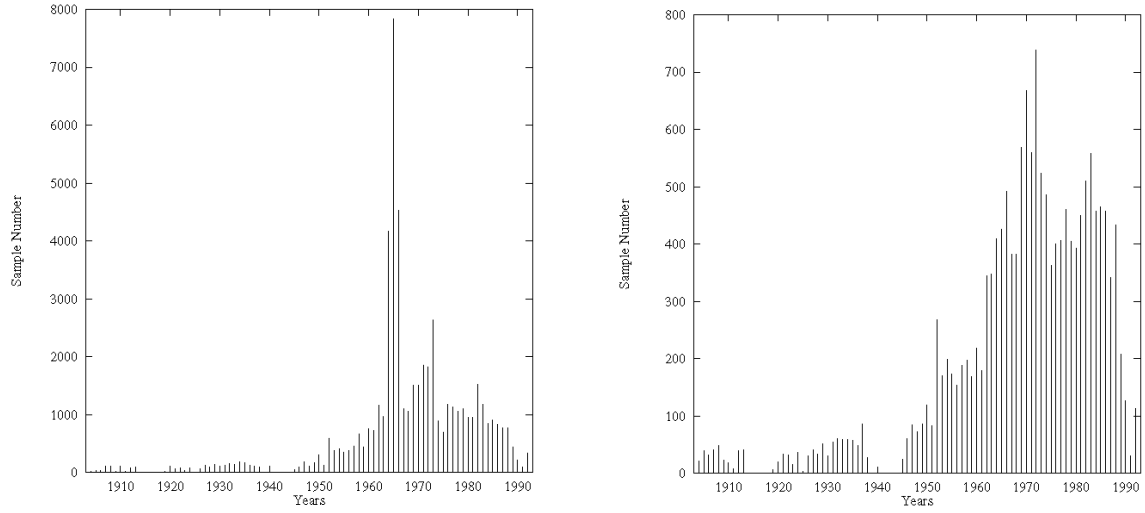


figure 3 : histogramme des données par années. 3(a) à gauche toutes les mesures disponibles. 3(b) à droite ensemble d'apprentissage Z_{app}.

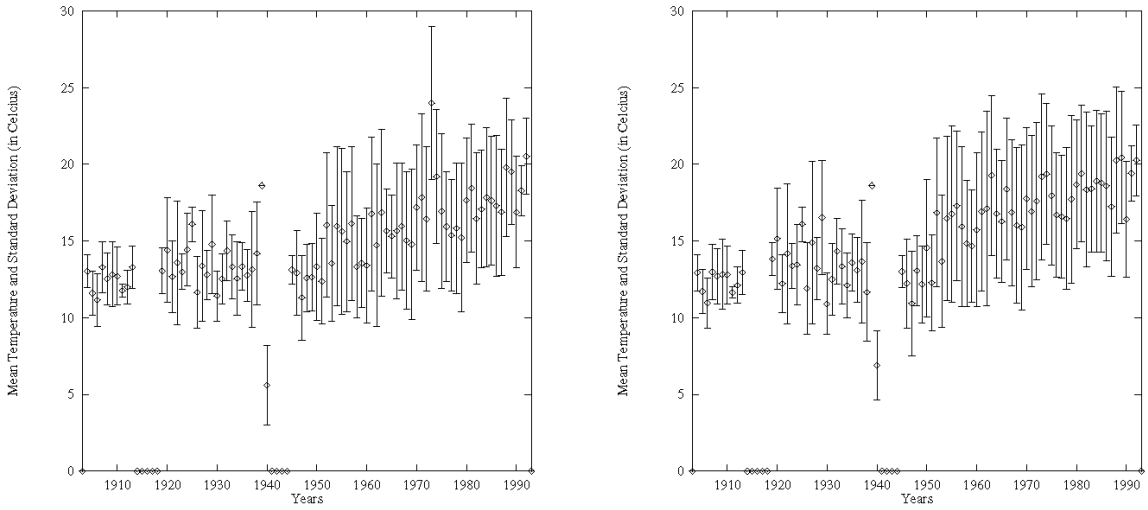


figure 4 : Moyenne et écart type par année. 3(a) à gauche représente la moyenne et l'écart type pour toutes les mesures disponibles. 3(b) à droite représente la moyenne et l'écart type sur l'ensemble d'apprentissage Z_{app}.

La figure 5.a présente la carte topologique obtenue en projection dans l'espace (x,y), pour plus de clarté la représentation du maillage a été simplifiée (maille carrée). Le maillage obtenu représente bien la projection sur le plan (x,y) d'une fonction et se répartit bien en fonction de la densité de probabilité de Z_{app}. La figure 5.b montre les isobares obtenues par SOM, celles-ci sont réalistes et sont similaires à celles obtenues à l'aide des méthodes classiques de Krigage utilisées par les océanographes. Comme la régression par fonctions noyaux ces méthodes nécessitent un temps de calcul très important.

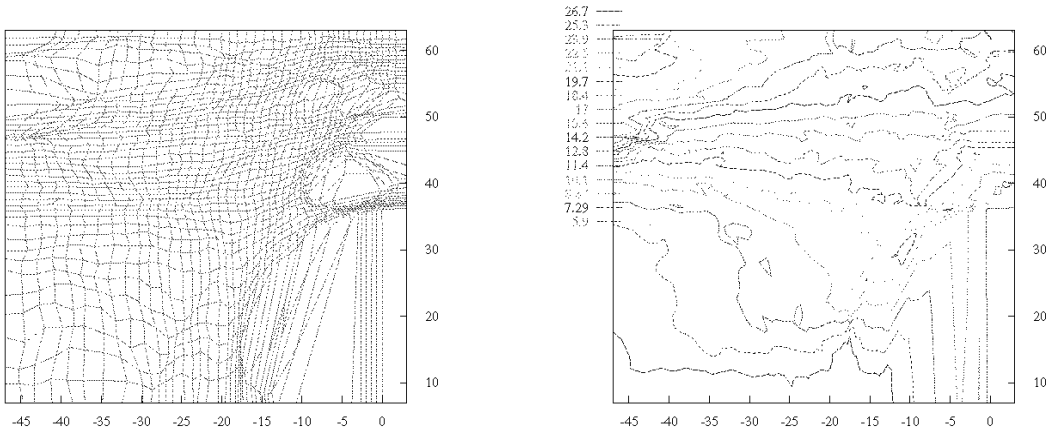


figure 5: 5.a à gauche représente la projection dans le plan (x,y) du maillage obtenu. 5.b à droite représente les isobares de la régression dans le plan (x,y).

Nous ne possédons pas dans cette expérience la fonction théorique de référence, les performances ont donc évaluées selon deux critères l'erreur quadratique moyenne

(RMS) et le contraste (CNT). Dans les deux cas nous effectuons des comparaisons avec la méthode du WARPing.

• le critère RMS défini par RMS (Re)=$\sqrt{\frac{1}{p}\sum_{i=1}^{p}\left(r_i - Re(x_i, y_i)\right)^2}$

où Re représente comme dans l'exemple précédent la régression linéaire obtenue à partir de la triangularisation de la carte C.

• Une discrétisation est idéale si elle permet d'associer un même nombre d'exemples à chaque point de discrétisation. De ce point de vue pour évaluer une discrétisation donnée nous calculons le contraste CNT(Re) = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{n_c}{n} - 1\right)^2}$

où n_C représente le nombre d'éléments de l'ensemble d'apprentissage appartenant au Voronoï associé au point de discrétisation w_C.

La table 2 donne les performances obtenues pour les deux algorithmes sur un ensemble de test indépendant Z_{test} dont les caractéristiques sont données par les figures 6.a et 6.b, il comporte 6308 exemples. Clairement les performances obtenues sont meilleures pour la méthode neuronale.

	SOM	WARPing
RMS-2(R)	1.19°	1.25°
CNT(C)	1.22	9

Table 2 : performances comparées des méthodes SOM etWARPing

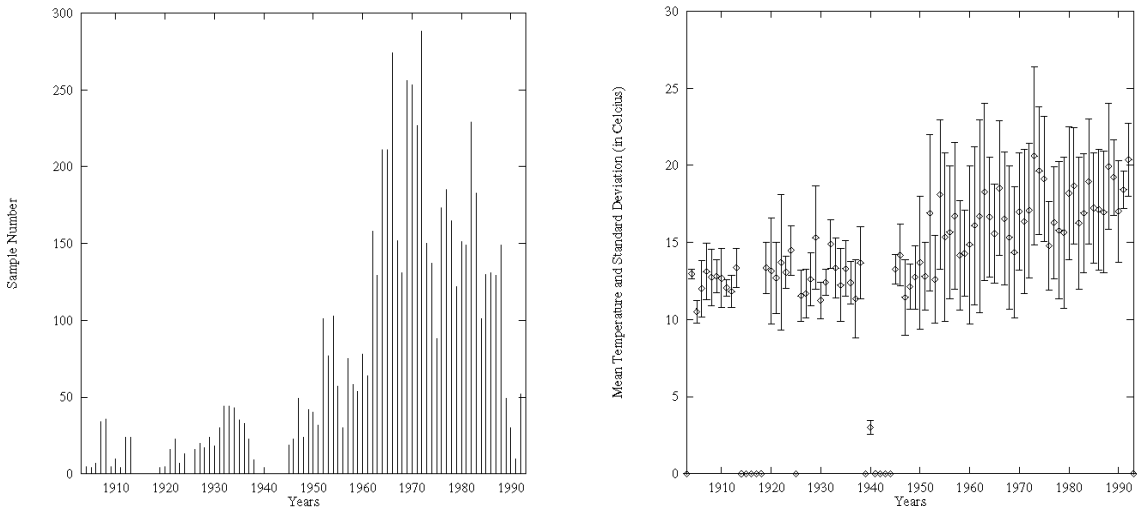


figure 6 : 6(a) à gauche représente l'histogramme sur l'ensemble de test Z_{test} . 6(b) à droite représente la moyenne et l'écart type sur l'ensemble de test Z_{test}.

5. Conclusion

Nous avons présenté dans cet article une adaptation de l'algorithme des cartes topologiques de Kohonen au problème de la régression *Dom*. L'algorithme SOM opère en deux phases : la première phase se comporte comme une carte topologique classique et réalise une triangularisation de l'ensemble *Dom*. Les triangles obtenus sont de taille variable, ils dépendent de la concentration locale des points de l'ensemble d'apprentissage. Chaque neurone de la carte est associé à un domaine de Voronoï, la topologie de la carte induit une notion de voisinage entre ces Voronoï. Cette première phase est non supervisée et s'oppose à la seconde phase qui affecte à chaque sommet du triangle (point de discrétisation) une valeur numérique en fonction des observations. Cette régression correspond à une approximation par WARPing d'un estimateur de type Naradaya-Watson. La valeur régressée est calculée en faisant une moyenne pondérée qui utilise les moyennes empiriques sur le Voronoï au point de discrétisation et celles des Voronoï voisins. La triangularisation obtenue ainsi que les régressions associées aux sommets permettent, par extrapolation linéaire, d'obtenir une régression pour tout point de *Dom*.
Nous avons comparé cette méthode de régression avec la méthode de WARPing classique opérant sur un découpage régulier de *Dom*. Les résultats obtenues sur des données simulées montrent l'avantage de la méthode des cartes topologiques supervisées. D'autre part, nous avons appliqué cette méthode à un problème de données réelles mesurant la température sur l'océan atlantique. Les résultats obtenus montrent que l'algorithme supervisé des cartes topologiques s'adapte mieux à la distribution et à la concentration des nuages de points. Les performances obtenues sont meilleures que celle obtenu par le WARPing classique.

D'autre part cette méthode peut être facilement généralisée à la régression de plusieurs variables $\left\{ r^i ; i = 1..p \right\}$ en fonction des deux mêmes variables (x, y) et également à plus de deux variables explicatives .

Références

BISHOP C.M (1995) : Neural networks for pattern recognition. *Oxford University Press*

CHERKASSKY V., LARI-NAJAFI H. (1991). Constrained topological mapping for nonparametric regression analysis. *Neural Network*, vol 4 pp 27-40.

HARDLE W. (1990): Applied non parametric regression . *Econometric society monographs n°19*. CambridgeUniversity Press

HARDLE W. (1991) . Smoothing techniques with implementation in S. *Springer series in statistics. Springer Verlag*

KOHONEN T. (1987). Self-organisation and associative memory. *Springer Verlag. 3rd edition*

NADARAYA E.A. (1964). On estimation regression. Theory of Probability and its applications, 10, 186-190.

POGGIO, T. and F. GIORSI (1990a). Networks for approximation and learning. Proceeding of the IEEE 78 (9), 1481-1497.

RITTER H, MARTINETZ T., SCHULTEN K.(1992). Neural computation and self-organizing Maps. *CNS Addison Wesley*

Y CHAUVIN, D.E RUMELHART (1995): Backpropagation : Theory, Architecture, and Applications. In Y Chauvin and D.E Rumelhart (Eds),Hillsdale, NJ : Lawrence Erlbaum.