MASTER : Traitement de l'Information et Exploitation des Données
Fouad Badran, Cécile Mallet, Carlos Mejia,
Awa Niang, Charles Sorror, Sylvie Thiria

TPB04 : Régression : modélisation du diffusiomètre NSCAT

I – Présentation du domaine d'Application

Simulation de la fonction directe d'un diffusiomètre satellitaire à l'aide des Perceptrons Multicouches – PMC et estimation de la variance.

Le diffusiomètre est un radar actif d'observation de la surface océanique (actif car il possède son propre source d'illumination de la surface d'observation, par rapport à d'autres radars dits passifs qui utilisent le rayonnement solaire). Son principe de mesure est basé sur la réflexion diffuse des ondes électromagnétiques qu'il a émises et qui sont diffusées dans toutes les directions par la surface océanique. Le coefficient de rétrodiffusion qu'il mesure est le rapport entre la puissance diffusée dans la direction du radar donc rétrodiffusée et la puissance incidente émise. Ce coefficient de rétrodiffusion (noté sigma0 ou $\sigma 0$) dépend de la rugosité de la surface de l'océan, de la géométrie de mesure (angle de visée ou d'incidence θ) et des caractéristiques du radar (la fréquence du signal utilisé, la polarisation des ondes électromagnétiques). La rugosité de surface dépend au premier ordre d'approximation du vent local soufflant sur la surface. Le vent engendre, par l'intermédiaire d'une force de frottement, des vaguelettes de quelques centimètres de LO (longueur d'onde ou la distance entre 2 crêtes) qui elles interagissent avec les ondes émises par le radar, ondes aussi de quelques centimètres de LO, selon les caractéristiques radars utilisées.

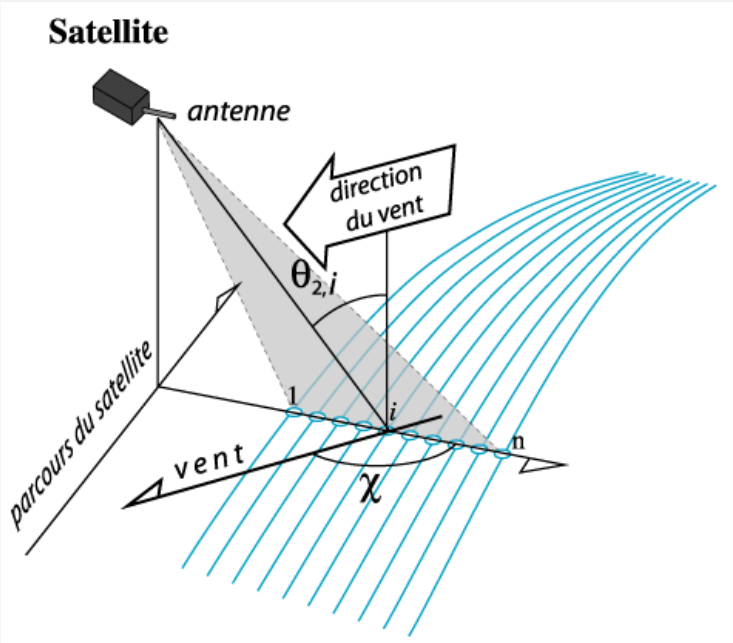
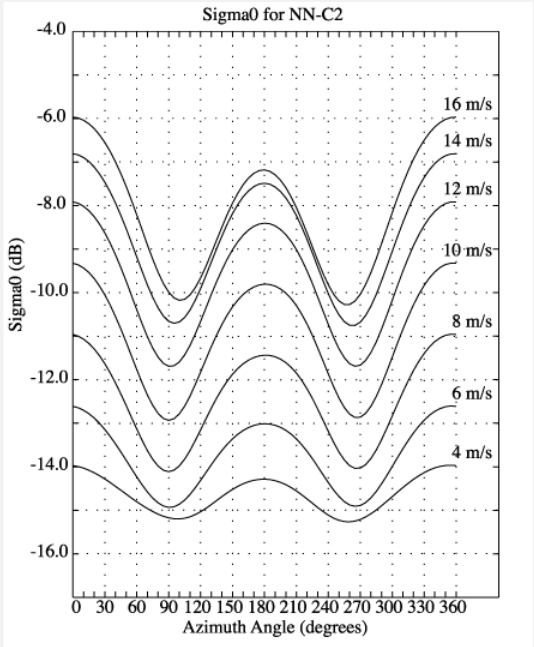


Schéma du capteur NSCAT HH: θ représente l'angle d'incidence du signal et χ l'angle d'azimut ou direction du vent par rapport au signal.



Courbe théorique pour un angle d'incidence constant.

•••

Le diffusiomètre effectue des mesures à incidence oblique (θ entre 20° et 50°), plus le vent est fort, plus la diffusion est importante et donc plus la mesure radar est forte. En résumé : à vent fort, diffusion plus importante dans toutes les directions et signal mesuré fort, tandis qu'à vent faible, faible diffusion dans la direction du radar et donc signal mesuré plus faible. Dans le cadre de ce projet, le diffusiomètre est NSCAT de la NASA. Il travaille en bande Ku $(14\ GHz)$ et effectue des mesures en polarisations verticale (VV) et horizontale (HH) de $\sigma 0$.

Ce TP a été inspiré par la thèse sur la « *contribution à l'étude des diffusiomètres NSCAT et ERS-2 par modélisation neuronale Influence de la hauteur des vagues sur le signal diffusiométrique* », soutenue par TRAN Vinh Bao Ngan. Elle est accessible par le lien suivant :
https://www.locean-ipsl.upmc.fr/~mmsa/pages/documents/Tran_These_1999/Tran_These_1999.pdf

II - Les Objectifs

Le 1er but de ce projet est de modéliser la fonction de transfert permettant de déterminer la polarisation horizontale (HH) de $\sigma 0$ et ceci en fonction du vent (direction et module).

Le but suivant sera de mettre en œuvre une méthode d'estimation de la variance du bruit en fonction des données en entrées. Cette méthode pourra être mise en œuvre sur un exemple de démonstration puis aussi et principalement sur la modélisation objet du 1er but. L'estimation de la variance sera utilisée pour attacher des intervalles de confiances aux résultats des régressions.

Incidemment, outre les objectifs applicatifs proprement dit, on aborde dans ce travail des problèmes récurrents au traitement des données réelles comme par exemple les calculs d'erreurs ou la normalisation/dénormalisation qu'il faut savoir bien maitriser. Si la normalisation permet aux systèmes d'apprentissages statistiques d'être plus efficient, seules les valeurs dénormalisées exprimées dans une unité physique peuvent intéresser l'expert du domaine concerné.

==========

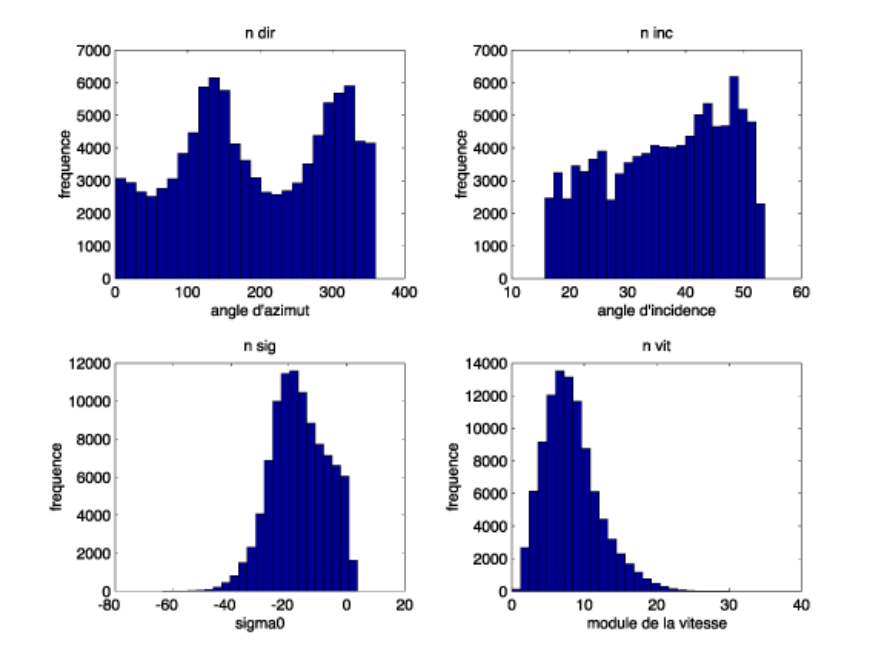
Le rapport de TP devra être synthétique. Il doit montrer la démarche suivie, et ne faire apparaître que les résultats nécessaires. Il s'agit de quantifier les résultats tout en rédigeant un rapport qui les analyse et les commente. Les paramètres utilisés devront être indiqués, Les graphiques des expériences doivent être insérés dans le rapport. Les résultats présentés devront être analysés et commentés.

III - Les Données

Les données mises à disposition sont constituées d'une base de données issue du radar NSCAT du satellite SeaScat. Leur taille et d'un peu plus de 4000 exemples. 4 fichiers sont inclus :

Diffu_Dir.dat :	Angle d'azimut (en degré) ou direction du vent (χ)
Diffu_Inc.dat :	Angle d'incidence (en degré) du signal (θ)
Diffu_Vit.dat :	Module de la vitesse du vent : ν (en m/s).
Diffu_Sig.dat :	Sigma0 (ou $\sigma 0$), composante HH uniquement (en dB)

Cette base n'est pas égalisée (voir histogrammes de fréquences ci-dessous), il y a bien plus d'exemples pour les basses vitesses que pour les hautes. Certains angles d'azimut sont mieux représentés que d'autres.



Attention : la figure montre l'histogramme de valeurs pour la base de données complète (100 000 mesures). Ici nous allons travailler avec un sous-ensemble, à angle d'incidence fixe, autour de 35 degrés (4 098 mesures). De ce fait, l'information relative à l'angle d'incidence peut être écartée des données d'entrée.

IV - Eléments pour le déroulement du TP

Programmes mis à disposition :

pmcinit.m : Permet de définir une architecture de perceptron multicouches et d'initialiser les poids. Les nombres d'entrées et de sorties sont fixés en fonction de la dimension des formes d'apprentissage qui sont utilisées, les biais sont ajoutés automatiquement. Les poids initiaux sont stockés dans les matrices W1 (poids entre couche d'entrée et couche cachée) et W2 (poids entre couche cachée et couche de sortie)

pmctrain.m : Permet d'effectuer l'apprentissage pour un ensemble d'apprentissage (X,Y) donné en utilisant une version de l'algorithme de rétro-propagation du gradient. L'algorithme s'arrête selon 3 critères possibles :

- le nombre maximum d'itération (**nbitemax**) a été effectué.
- les poids ne changent presque plus (variabilité<seuil).
- Si des données de validation sont passées, le programme s'arrête après un certain nombre d'itérations au-delà du minimum de l'erreur en validation (**ntfpamin*tfp**)

Lorsque des données de validation sont passées, les poids (W1, W2) retournés sont ceux du minimum de la validation, et les poids obtenus en fin de procédure sont stockés dans le fichier pmcendwei, sinon ils sont retournés (sans être stockés).
Ce programme comporte pas mal de paramètres dont plusieurs d'entre eux sont passés selon le formalisme Matlab des « arguments variables » (**varargin**). Il vous revient de choisir, pour vos expériences, le nombre de neurones cachés et le nombre d'itérations. Sauf indication, les autres paramètres devraient pouvoir être laissés à leur valeur par défaut.

pmcout.m : Permet de calculer les valeurs de sortie d'un perceptron multicouches de poids [W1,W2] pour un ensemble de points X.

Autre codes pas ou moins directement liés aux PMC :

centred.m : Normalisation par centrage réduction.
decentred : Dénormalisation de données centrées réduites.
split_bases.m : Séparation des matrices de données d'entrée et de sortie en 2 sous-ensembles. Par défaut, le 1er ensemble est constitué des 2/3 des éléments tirés aléatoirement, il est en principe destiné à l'apprentissage le 1/3 restant pouvant servir à la validation.
errors.m : A utiliser pour calculer différentes sortes d'erreurs.
plot_donnees.m : Affichage en 2D de données en dimension 3 en sélectionnant certains points de la 2ème dimension avec une tolérance.
plot_regress.m : Trace la courbe de régression d'un PMC (valeur de sortie Sigma0) en fonction des valeurs d'abscisses et d'une valeur d'abaque fixée.
plot_encadre.m : Trace les courbes d'encadrement à plus ou moins 2 écarts types pour une valeur d'abaque.

Remarque : Ces fonctions et leurs paramètres sont en principe explicités dans chacune d'elle. A défaut, un examen des codes devrait permettre de déduire des informations supplémentaires qui n'ont pas été exposées ici. **demoNSCAT.m** est un exemple de script qui montre l'utilisation de certaines des fonctions utiles pour le TP.

V – 1ère Partie : Modélisation du diffusiomètre NSCAT

Nous vous demandons de réaliser une régression par PMC pour modéliser la fonction du diffusiomètre NSCAT. Le rôle du PMC sera donc de donner la valeur de sortie Sigma0 en fonction de la direction et de la vitesse du vent. Les fichiers de données à utiliser sont donc Diffu_Dir.dat et Diffu_Vit.dat, pour les données d'entrées et Diffu_Sig.dat pour celles de sortie. Cependant, Le PMC ne devra pas travailler directement avec les données brutes mais avec des données normalisées dont on va préciser les méthodes. Après normalisation, les matrices de données devront être séparées en ensemble d'apprentissage et ensemble de validation à concurrence de 2/3 pour l'un et 1/3 pour l'autre (voir le code **split_bases.m**)

Travail à faire

1) Préparation des données :

► Après lecture des fichiers de données, il faudra procéder à leur normalisation de la façon suivante :

- Pour la direction du vent, la normalisation va consister à prendre les sinus et les cosinus de la direction du vent qu'il faudra donc bien penser à exprimer en radian préalablement (*La relation de la variable de sortie, sigma0, par rapport à la direction du vent a une périodicité de 360°. Un codage linéaire pour cette variable où l'intervalle de 0 à 360 serait codé entre –1 et 1, ne permettrait pas de préserver la relation de périodicité. Un codage par une fonction sinusoïdale est conseillé*).

- Pour la vitesse on appliquera une normalisation par centrage-réduction (avec un coefficient de 2/3). (*Le module de la vitesse du vent varie dans l'intervalle de 0 à 30 m/s approximativement. La relation avec la variable de sortie est proche du linéaire. Un codage linéaire est donc approprié. Un codage classique est la normalisation par* ***min/max****. Les données sont codes entre –1 et 1. Nous proposons cependant une normalisation par* ***centre-réduction*** *des données. Cette méthode permet de ramener la moyenne des données ainsi traitées à 0 et son écart type à 1. Empiriquement, dans ces conditions, l'apprentissage se fait dans des « bonnes conditions » pour éviter de tomber en saturation dans l'état des cellules internes, et en évitant l'écrasement des valeurs produit lors d'une normalisation par min/max d'une série de données à distribution gaussienne*).

- Pour le coefficient de rétrodiffusion sigma0 (*qui varie entre -45 et 5 dB approximativement*) on appliquera là aussi une normalisation par centrage-réduction avec un coefficient de 2/3.

► Constituer les matrices d'entrée et de sortie du PMC pour l'ensemble d'apprentissage et pour celui de validation (utilisation **de split_bases.m**)

► Produire les nuages de points des variables normalisée. On présentera 3 figures pour lesquelles on mettra en abscisse la variable d'entrée et en ordonnée la sortie Sigma0.

•••

2) Réalisation de la fonction neuronale

Pour l'initialisation puis l'apprentissage du PMC (procédures **pmcinit.m** et **pmctrain.m**) vous devez choisir certains paramètres. Pour les fonctions d'activation, nous vous conseillons l'usage de la fonction tangente hyperbolique pour les neurones cachés (F1='tah') et linéaire pour les neurones de sortie (F2='lin'). Le nombre de neurones sur la couche cachée ne devrait pas être trop élevé (de l'ordre de 5). Un nombre d'itérations entre quelques centaines ou quelques milliers devrait convenir. Les autres paramètres peuvent être laissés à leur valeur par défaut.
On précise ici, en accord avec ce qui a été présenté, que la matrice d'entrée du PMC doit être constituée de 3 vecteurs colonne correspondant :

- au sinus de la direction du vent exprimée en radian
- au cosinus de la direction du vent exprimée en radian
- à la vitesse du vent centrée-réduite

La matrice de sortie est pour sa part, constituée du vecteur sigma0 en valeur normalisée par centrage-réduction.

Dans la réalité ces fonctions sont mises au point sur un ensemble de données très important (plusieurs centaines de millier). Afin d'avoir une meilleure appréciation des résultats, nous les présenterons en utilisant l'ensemble de toutes les données (réunion de l'ensemble d'apprentissage et de validation) et en utilisant les poids au minimum de l'erreur en validation.

Après apprentissage, vous devrez produire (et commenter) :

► Deux tableaux d'erreurs qui permettront d'apprécier les performances du PMC, l'un en fonction de la vitesse et l'autre de la direction. On travaillera pour estimer les résultats sur des intervalles de vitesse et de direction. On prendra des intervalles de **2m/s** pour la vitesse et de **60°** pour la direction. Les erreurs à produire sont la RMS, la RMS relative et le BIAIS. Le nombre de points et la moyenne par intervalle peuvent être des indications intéressantes. Que peut-on conclure de ces informations ?

► Le diagramme de dispersion de sigma0 en valeur dénormalisée. C'est le nuage de points des valeurs estimées par le PMC par rapport aux valeurs de références (sorties désirées) après dénormalisation. On devra de plus produire 2 deux autres diagrammes où les points devront être associés à une couleur en fonction de la vitesse pour l'un et en fonction de la direction pour l'autre.

•••

► Une figure présentant en valeur dénormalisée et pour chacune des vitesses [**4 8 12 16 20**] :

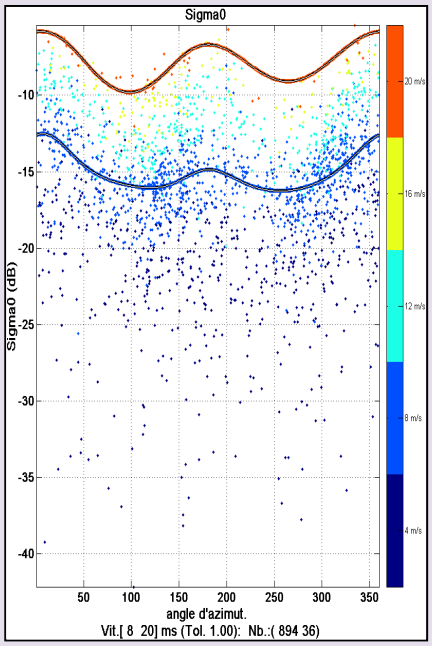
- les données proches, avec une tolérance de **1m/s**, autour de la vitesse considérée
- « la » courbe de régression du PMC. On en représentera les abaques en fonctions des vitesses.

(Utilisation de **plot_donnees.m** et **Plot_regress.m)**. Sur la figure, on aura :

- en abscisse : l'angle d'azimut en degré en abscisse. Pour cela on échantillonnera l'axe de 1° à 360° par pas de 1°.
- en ordonnée : sigma0
- les courbes d'abaque de la régression (PMC) associées à chaque vitesse.

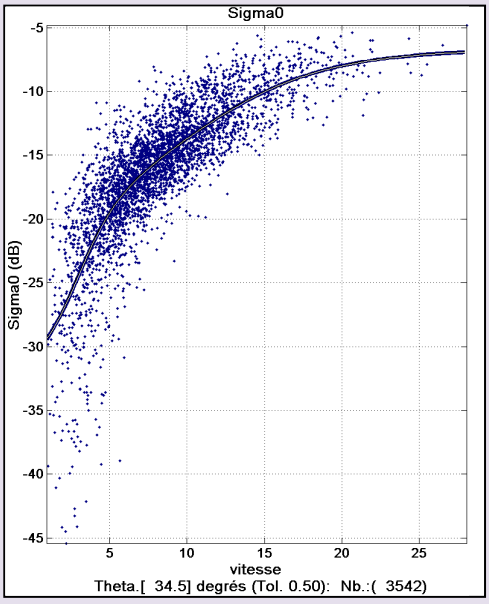
Le programme **plot_donnees.m** permet d'afficher les points avec une couleur différente selon la vitesse. Le programme **plot_regress.m** permet l'affichage d'une courbe.

La figure ci-contre est un exemple, limité aux vitesses [8 20] de ce à quoi vous devriez à peu près aboutir. L'exemple **demoNSCAT.m** évoqué ci-dessous devrait pouvoir vous aider en partie.



Programme de démonstration demoNSCAT

Le programme de démonstration NSCAT peut vous aider dans une certaine mesure, mais attention, il peut aussi vous induire en erreur car il ne travaille pas avec les mêmes variables et il ne les utilise pas non plus dans le même ordre.



Lorsqu'on lance l'exécution, on obtient, après apprentissage une figure comme celle ci-contre sur laquelle on peut remarquer plusieurs choses par rapport au travail qui vous est demandé :

- Le fichier des directions (Diffu_Dir.dat) n'est pas utilisé ici, par contre, on se sert de celui des angles d'incidence Théta (fichier Diffu_Inc.dat).
- Ce sont les vitesses qui sont en abscisse, et non pas les angles d'azimut.
- Une seule courbe est tracée pour **Théta**=[34.5] avec une tolérance de 0.5, alors que pour votre TP, les courbes doivent être tracées par vitesses et pour plusieurs d'entre elles ([4 8 12 16 20]) avec une tolérance de 1.0.

VI – 2ème Partie : Encadrement par approximation de la variance

Afin d'attacher des intervalles de confiance aux résultats trouvés, on propose d'estimer la variance du bruit en fonction des données. La méthode proposée a déjà été mise en œuvre dans un TP précédant (TPB01) à la normalisation près :
Un premier réseau (PMC1) a été optimisé pour modéliser une fonction $y=f(x)$, à partir d'une base d'apprentissage $\{(x_i,d_i)\ i=1,N\}$.
Après apprentissage, la sortie de PMC1 donne une estimation de l'espérance $E(d/x)$. On note $Eapp$ l'erreur commise par ce 1er réseau : $Eapp = (y - E(d/x))$.

Il s'agit maintenant d'optimiser un second réseau (PMC2) pour modéliser la fonction $Var(d/x) \approx E[Eapp^2/x]$ à partir d'une autre base d'apprentissage où la sortie désirée est l'erreur quadratique : $\{(x_i,\ Eapp^2{}_i)\ i=1,N\}$. Après apprentissage, la sortie de ce second réseau donnera une estimation de l'espérance $E[(d/x - E[d/x])^2]$, c'est à dire, une estimation de la variance de d/x pour chaque valeur de d.
Dans le cas de ce TP, les PMC travaillant en valeur normalisée, il faudra penser à dénormaliser la sortie de PMC1 avant de calculer l'erreur quadratique $Eapp^2$, puis re-normaliser cette dernière pour construire l'ensemble d'apprentissage de PMC2. Au final les présentations de résultat devront être faites en valeur dénormalisée.

Le travail à réaliser pour cette partie devra porter sur 2 cas :

- le cas de l'exemple de démonstration (**demoNSCAT.m**) où l'on a :
  - y = la polarisation horizontale (HH) de σ0,
  - x = angle d'incidence (θ) et module du vent (ν).
- le cas de la 1ère partie de ce TP avec :
  - y = la polarisation horizontale (HH) de σ0,
  - x = direction du vent (χ), et module du vent (ν).

Etant donné que la variance est un estimateur du second ordre, il est important d'avoir un maximum de données pour estimer les paramètres : pour estimer PMC2 on prendra l'ensemble des données pour l'apprentissage (on travaillera donc sans ensemble de validation ni de test). On rappelle par ailleurs que pour ce type de modèle c'est une fonction d'activation exponentielle qui est appropriée pour les neurones de la couche de sortie (F2='exp').

•••

Travail à faire

Les étapes nécessaires sont les suivantes :

1. Générer les bases d'apprentissage nécessaires à ce second réseau
2. Choisir l'architecture du réseau, puis appliquer l'algorithme de rétro-propagation du gradient.
3. Présenter des figures :
   - qui montrent les erreurs quadratiques et la régression qui en est faite par PMC2. (faire soit-même l'affichage des données, la régression peut être faite avec demoplot_regres, ou plot_regress)
   - qui donne une représentation simultanée des données, de la régression (de PMC1) et de l'encadrement à plus ou moins 2 écarts types (utilisation des programmes plot_donnees, demoplot_regres, plot_regress et plot_encadre). On indiquera le pourcentage de données compris dans l'encadrement.

   Pour éviter d'avoir des figures trop confuses, il est demandé de faire une figure pour théta et pour chaque vitesse définie précédemment avec leur tolérance respective.