MIAM (ICM1A-ICM2A), UP1, TP2 : Régression

Ce travail pratique numérique sera l'occasion d'implémenter la méthode de régression paramétrique par les moindres carrés, et de l'appliquer à la construction d'un modèle capable de prédire la perméabilité k d'un milieu homogène équivalent en fonction de quatre paramètres de microstructure :

- V_f : le taux volumique de fibre, égal au rapport entre la surface des fibres et la surface totale étudiée
- R_{min} : le rayon minimal des fibres
- R_{max} : le rayon maximal des fibres
- ε : la distance minimale inter-fibres

Ce TP est évalué. Le travail est à faire individuellement. **Si vous choisissez ce TP** : il faudra déposer, au plus tard deux semaines après la fin de la séance « réseaux de neurones » (dernière séance de TP), le code que vous aurez fait, ainsi qu'un compte-rendu sous format `pdf`. Il est également possible de rendre les deux simultanément dans un notebook jupyter. Dans ce cas, il faut rendre à la fois la version `ipynb` et `pdf`. **Si vous ne choisissez pas ce TP** : vous devrez faire apparaître en préambule du compte-rendu du TP que vous choisissez un paragraphe expliquant la démarche de ce TP.

La forme est importante. Dans un contexte « ingénieur », un code mal commenté, mal organisé ou avec des noms de fonctions ou de variables mal choisis a vocation à être une source de stress et d'erreurs pour les contributeurs qui s'y pencheront après vous. Dans le contexte présent, un tel code a simplement vocation à être une source de points en moins pour vous. Il en va de même pour le compte-rendu.

Forme attendue pour le compte-rendu : Il doit pouvoir se lire indépendamment du sujet. Il comprendra une rapide introduction expliquant l'objet d'étude, et surtout présentera les résultats de l'exécution de votre code (images ou matrices si elles sont petites), ainsi qu'une description de ces résultats (paramètres utilisés etc) et un commentaire sur ces résultats. Le compte-rendu est organisé en sections (les mêmes que dans le sujet) et se finit par une conclusion, comme rappelé en fin de document. Si le compte-rendu n'est pas un notebook jupyter, le code ne doit pas y apparaître. Il faut également faire apparaître un court résumé des TPs non-choisis dans le compte-rendu du TP choisi.

Mise en place

En plus de ce sujet, vous aurez besoin du support de cours pour pouvoir mener à bien le travail demandé. Vous avez également à disposition une bibliothèque de fonctions regroupées dans le fichier `regression_ressources`. Cette bibliothèque devra être importée dans le code que vous créerez, de préférence sous un diminutif bien choisi.

Les deux fonctions `readSingleMicrostruct` et `plotMicrostruct` permettent respectivement de lire les microstructure données et de les afficher. Elles ne sont pas directement utiles pour le travail à faire, mais elles vous permettront d'illustrer votre compte-rendu si vous jugez pertinent d'y afficher des images de microstructures.

1 Implémentation de la régression

La fonction `readData` permet d'importer automatiquement les données d'entrée V_f, R_{min}, R_{max} et ε sous la forme d'un multi-vecteur (une matrice à 4 colonnes) et la donnée de sortie k sous la forme d'un unique vecteur.

À faire : Importer les données et tracer les quatre graphes donnant la distribution de k en fonction de chacun des quatre paramètres. Ces graphes permettent-ils de connaître l'influence individuelle de chacun des paramètres sur k ? On se demandera notamment s'il y a des facteurs de confusion.

La fonction `polyPowers` est un outil permettant de construire des polynômes d'ordre voulu. Cette fonction retourne une matrice d'entiers dont chaque ligne donne les puissances impliquées dans un monôme.

La méthode pour construire un monôme à-partir d'une ligne v est la suivante :

- Initialiser le monôme à 1
- Pour chaque terme i du vecteur ligne, multiplier le monôme par $x_i^{v_i}$

À faire : Construire une fonction ayant en argument la matrice de données d'entrée X et des indices au format décrit ci-dessus et sortant une matrice H dont chaque colonne correspond à l'un des monômes de la liste d'indices évalué en chaque point où X est donné.

La matrice H générée est celle du cours : $H_{ns} = h_s(x_n)$. La régression linéaire va demander de construire $A = H^T H$ et $b = H^T y$, et de résoudre le système linéaire correspondant afin de trouver les coefficients $\{a_s\}_{s=1..S}$ à appliquer aux différentes fonctions $\{f_s\}_{s=1..S}$

À faire : Implémenter la méthode de régression polynomiale décrite ci-dessus et tester son bon fonctionnement en effectuant une régression affine. Commenter le résultat. L'influence des paramètres vous paraît-elle être physiquement compréhensible ?

À faire : Observer visuellement l'écart entre les valeurs d'entraînement et celles prédites par le modèle en utilisant la fonction `scatter` de `matplotlib`, et vérifier que les points dans l'espace $(f(x), y)$ ne s'éloignent pas trop de la droite $f(x) = y$.

Le résidu est la norme du vecteur $r = Ha - y$. C'est cette quantité qui est minimisée lors de l'inversion du système linéaire.

À faire : Tracer l'évolution du résidu avec l'ordre des polynômes utilisés. On fera démarrer cet ordre à 0 pour augmenter le nombre de données à afficher. Interpréter le résultat.

Le conditionnement d'une matrice est le rapport entre sa plus grande et sa plus petite valeur propre. Il est obtenu avec la fonction `np.linalg.cond`. Quand ce conditionnement dépasse l'inverse de la précision numérique ($\simeq 10^{16}$), les algorithmes d'inversion de systèmes linéaires (sauf cas particuliers) se comportent comme s'ils « ne voyaient pas » les plus petites valeurs propres. La matrice devient alors numériquement singulière et l'inversion du système linéaire est impossible (ou fausse).

À faire : Tracer l'évolution du conditionnement avec l'ordre des polynômes utilisés. En observant les données (en ouvrant `VfRminRmaxepsi.csv` dans un éditeur de texte), cette évolution était-elle prévisible ? En déduire l'ordre maximal utilisable en pratique.

2 Ordre polynomial et validation croisée

À faire : Construire une fonction ayant comme argument un multi-vecteur d'entrée X, un ordre maximal et un vecteur de coefficients, et sortant dans un vecteur la valeur de y associé à chaque vecteur de X. Cette fonction est le modèle de régression. **Attention :** pour que la suite fonctionne correctement, X doit être le premier argument de la fonction.

Dans les ressources, on met à disposition une fonction qui s'appelle `modelPlotter`. Elle permet de comparer le résultat donné par le modèle de régression et les données sur un segment de la base de donnée à R_{min}, R_{max} et ε fixés.

À faire : Utiliser la fonction `modelPlotter` afin d'évaluer visuellement le modèle. Faire cette opération pour un ordre faible et un ordre élevé et commenter les résultats.

À faire : Sans utiliser `modelPlotter`, mais en s'inspirant de la façon dont cette fonction est construite, afficher les résultats du modèle pour R_{min}, R_{max} ou ε variables. Pourquoi ne peut-on pas effectuer la comparaison avec les données dans ces cas-là ?

Une méthode pour vérifier la pertinence d'une solution d'identification s'appelle la validation croisée. Il s'agit de séparer la base de données en deux sous-ensembles. L'un des sous-ensembles (habituellement le plus gros) sert à construire le modèle tandis que l'autre sert à l'évaluer.

$$\begin{aligned} \{x_n\}_{n=1..N} &\twoheadrightarrow \begin{cases} \{x_n\}_{n=1..N_1} \twoheadrightarrow \hat{H}_{ns} = h_s(x_n) \\ \{x_n\}_{n=N_1+1..N} \twoheadrightarrow \check{H}_{ns} = h_s(x_n) \end{cases} \\ \{y_n\}_{n=1..N} &\twoheadrightarrow \begin{cases} \{y_n\}_{n=1..N_1} \twoheadrightarrow \hat{y} \\ \{y_n\}_{n=N_1+1..N} \twoheadrightarrow \check{y} \end{cases} \end{aligned} \tag{1}$$

Le vecteur de coefficients a est déterminé en résolvant $\hat{H}^T\hat{H}a = \hat{H}^T\hat{y}$, mais le résidu de validation croisé est $\check{r} = \check{H}a - \check{y}$. La fonction `crossValSplit` permet de séparer les deux sous-ensembles précédemment évoqués.

À faire : Tracer le résidu de validation croisée en fonction de l'ordre polynomial. Quel ordre faut-il choisir pour minimiser ce résidu ? Faire le lien avec les notions de sur et sous-échantillonnage. Le résultat est-il cohérent avec les observations faites précédemment sur le conditionnement ?

3 Normalisation

À faire : Revenir au cas de la régression affine (ordre 1). Peut-on savoir quels sont les paramètres micro-structuraux influant le plus sur la sortie y ? Pourquoi ?

À faire : À l'aide d'uns transformation affine, ramener toutes les données d'entrée entre 0 et 1. Refaire la régression affine. Peut-on maintenant connaître l'influence relative des paramètres micro-structuraux sur la sortie y ?

À faire : Tracer le conditionnement en fonction de l'ordre polynomial. Interpréter le résultat.

4 Régularisation

On travaille sur les données normalisées. On choisit un paramètre de régularisation $\mu = 10^{-5}$. On a vu dans le cours qu'il était possible d'effectuer une régression polynomiale d'ordre aussi élevé que voulu (tant que la matrice résultante n'est pas trop grosse pour les capacités CPU de votre machine) en régularisant le problème d'optimisation. Il s'agit de remplacer la matrice A par la matrice suivante :

$$A_\mu = H^T H + \mu I_S \tag{2}$$

À faire : Quel est l'avantage de cette méthode par-rapport à simplement limiter l'ordre polynomial ?

À faire : Observer l'évolution du conditionnement avec l'ordre polynomial lorsque μ est fixé.

À faire : À l'aide de la méthode de validation croisée, trouver à la main une valeur optimale de μ.

Dans le contexte de la régression à-partir de grandes quantités de données, la validation croisée est probablement la méthode la plus pertinente pour déterminer la valeur d'un paramètre de régularisation. Cependant, dans les cas où les données sont peu nombreuses, ou les cas où le problème est naturellement mal posé (très mauvais conditionnement), cette méthode présente l'inconvénient d'ignorer des données qui pourraient avoir une importance cruciale dans l'identification. C'est pourquoi d'autres méthodes existent pour déterminer le paramètre μ optimal.

La méthode de la courbe en L *(L-curve)* consiste à tracer la courbe $(\|r_\mu\|, \|a_\mu\|)$, le plus souvent en coordonnées logarithmiques, c'est à dire la norme de a en face du résidu pour différentes valeurs de μ. Une telle courbe est censée présenter un coude, qui indique la « bonne » valeur de μ.

Bonus : Appliquer la méthode de la L-curve au problème présent afin de trouver la meilleure valeur à donner à μ. Ce résultat est-il compatible avec celui obtenu précédemment ?

5 Bonus : Encore plus de régularisation

On travaille toujours sur les données normalisées. On va tenter une autre méthode de régularisation que l'on espère être plus pertinente : on va pénaliser les gradients du modèle. En reprenant les notations du cours, on se retrouve à résoudre le problème suivant :

$$\min_a \frac{1}{2}\sum_{n=1}^{N}\left(\sum_{s=1}^{S} a_s h_s(x_n) - y_n\right)^2 + \frac{\mu}{2}\sum_{n=1}^{N}\sum_{i=1}^{I}\left(\sum_{s=1}^{S} a_s \frac{\partial h_s}{\partial [x]_i}(x_n)\right)^2 \tag{3}$$

où I est la dimension des données d'entrée (4 dans le cas présent).

Bonus : Montrer que cette régularisation revient à remplacer A par A_μ suivant :

$$A_\mu = H^T H + \mu \sum_{i=1}^{I} H_{d,i}^T H_{d,i} \tag{4}$$

Où les termes de $H_{d,i}$ s'écrivent :

$$[H_{d,i}]_{ns} = \frac{\partial h_s}{\partial [x]_i}(x_n) \tag{5}$$

Bonus : Sur le modèle de la fonction qui a été précédemment implémentée pour construire la matrice H, implémenter la fonction pour construire les matrices $H_{d,i}$ pour i allant de 0 à 3 (ou 1 à 4 en notation mathématique).

Bonus : Comme dans la section précédente, déterminer le paramètre μ optimal par validation croisée (et/ou L-curve). Comparer à la méthode précédente. Quel est le défaut de la régularisation que l'on vient d'implémenter ?

Bonus : Corriger le problème précédemment mentionné en calculant le terme régularisant sur une autre famille de vecteurs $\{\hat{x}_m\}_{m=1..M}$ choisie judicieusement.

Conclusion

À faire : Proposer une conclusion synthétique, claire et pertinente à ce travail. On s'interrogera notamment (mais pas nécessairement exclusivement) sur les questions suivantes :

- La notion de facteur de confusion, et la manière de minimiser le risque d'être victime de corrélations fallacieuses
- La notion de sur et sous-échantillonnage, et les méthodes permettant de s'affranchir de ces problèmes
- La notion d'extraoplation et les risques associés