

Construction de métamodèles sur sorties fonctionnelles de codes de calcul

Benjamin Auder

CEA Cadarache. DEN/DER/SESI/LCFR - UPMC paris 6

Résumé

Dans l'industrie nucléaire, les codes de simulation utilisés sont souvent très lourds en temps de calcul. Or il est nécessaire de réaliser des analyses de sensibilité sur ces codes. Ces analyses demandent beaucoup d'évaluations du code pour être précises. C'est pourquoi nous cherchons à réduire le temps d'exécution d'un code de calcul, en le modélisant par une fonction mathématique appelée "métamodèle", de coût négligeable.

Supposons connus un certain nombre de résultats de code $(x_i, y_i = f(x_i))$, avec y_i une variable fonctionnelle, x_i étant une entrée vectorielle du code et f la relation entre x_i et y_i pour une simulation. Il faut d'abord classer les fonctions y_i en groupes de similarités pour mieux préciser le problème, puis déterminer un métamodèle pour chaque groupe en se ramenant à la dimension finie (en général en choisissant une base de fonctions B). En effet les métamodèles en dimension finie ont été largement étudiés. Nous illustrerons la méthode sur un exemple réel : l'évolution de la température dans l'espace annulaire d'une cuve de réacteur nucléaire lors d'un transitoire accidentel.

Introduction

Nous observons que les sorties fonctionnelles d'un code de calcul se divisent assez naturellement en plusieurs sous-groupes. Nous commençons donc par une étape de séparation des données y_i en classes ("clustering" ou classification non supervisée). Ensuite, il est nécessaire de répercuter cette classification sur les entrées. En effet, notre but étant la prédiction de nouvelles courbes, il faut pouvoir déterminer à quel groupe d'entrées appartient une nouvelle donnée x . Il s'agit cette fois de classification supervisée. La seconde partie portera sur la réduction de la dimension des fonctions en sortie : informatiquement on ne peut traiter que des quantités discrètes, donc nous chercherons ici la meilleure représentation possible des fonctions $f(x)$. Finalement, nous évoquerons les premiers résultats obtenus sur la simulation du choc thermique pressurisé.

Classification des courbes et entrées

On utilise la méthode du clustering spectral (Ng et al. 2001 [3]) pour classer les courbes. Celle-ci part de la matrice A de taille $n \times n$ où n est le nombre de résultats de code, représentant les distances entre les courbes y_i en sortie : $A_{i,j} = d(y_i, y_j)$, d étant une fonction distance sur les courbes. Toute l'information sur les distances entre les courbes est contenue dans la matrice A : cette distance sera notre critère pour construire des groupes de fonctions. Techniquement, nous utilisons ensuite le laplacien L de A , dont nous calculons les k plus grandes valeurs propres ainsi que les vecteurs propres associés, regroupés en colonnes dans la matrice V . L'algorithme des k -means alors appliqué aux lignes de V fournit une classification des données. Le passage dans l'espace des k premiers vecteurs propres permet en général de mieux distinguer les clusters, dont les centres se situent à la surface d'une sphère réelle de dimension k .

Nous connaissons maintenant les classes de fonctions en sortie du code. La méthode la plus simple pour faire correspondre les entrées aux sorties est alors celle des k plus proches voisins (Cover & Hart 1967 [1]) : on choisit un entier $k > 0$, et on classe toute nouvelle entrée x dans le groupe tel que les entrées de ce groupe soient majoritaires parmi les k plus proches voisins de x dans l'espace d'entrée \mathcal{X} . Cette méthode est en fait un estimateur de Bayes empirique, qui converge vers le classificateur optimal quand k et n tendent vers $+\infty$, avec $k/n \rightarrow 0$.

Représentation des fonctions en sortie

Les fonctions que nous manipulons sont discrétisées, donc de dimension finie mais relativement importante : il faut la réduire. Pour cela nous avons mis au point une méthode de construction empirique d'une base de fonctions, adaptée à la topologie des courbes pour un code de calcul donné. L'idée est d'optimiser pas à pas certains critères sur une nouvelle fonction ajoutée au système libre courant. Celle-ci doit en effet :

- être au moins aussi lisse que les courbes y_i ;
- être orthogonale aux autres fonctions de base précédemment construites ;
- avoir de bonnes propriétés de regroupement des coefficients de décomposition, en vue de l'apprentissage ultérieur ;
- bien représenter les données y_i , et donc maximiser les valeurs absolues des coefficients de projection.

Pour réaliser cette optimisation nous avons utilisé un algorithme génétique, en définissant des opérateurs de croisement et mutation adéquats. La figure 1 montre quelques résultats obtenus avec ce dernier algorithme pour le choc thermique pressurisé.

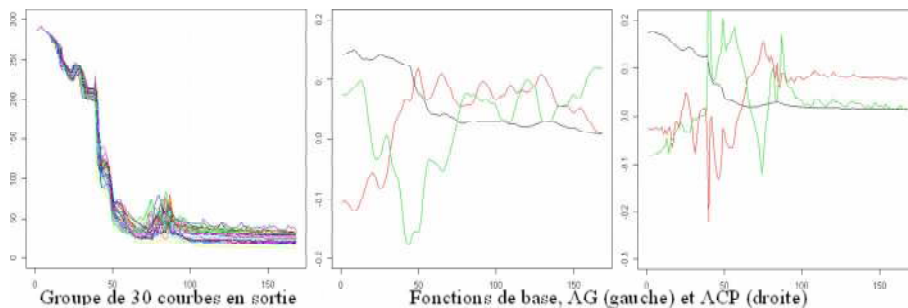


FIG. 1 – 30 courbes issues du code de calcul (à gauche), 3 premières fonctions de base obtenues par l'algorithme génétique (au milieu), puis par ACP fonctionnelle (à droite)

Conclusion

Nous avons testé quelques méthodes existantes et constaté qu'elles se comportaient assez bien sur les données à disposition. Dans la suite du travail il faudra définir exactement la qualité de ces méthodes d'une manière théorique, et si celles-ci ne sont pas satisfaisantes en proposer d'autres, par exemple en choisissant une ACP fonctionnelle non linéaire comme point de départ. Nous nous intéresserons également aux problématiques de prétraitement des courbes (lissage, interpolation, alignement), étapes indispensables en pratique.

Le but ultime de ce travail, dans son contexte industriel, est d'appliquer une méthodologie pour construire des métamodèles permettant de prédire les courbes thermo-hydrauliques en sortie de simulations du choc thermique pressurisé (Marques & Auder 2008 [2]).

Références

- [1] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13 :21–27, 1967.
- [2] M. Marques and B. Auder. Construction de métamodèles pour la propagation d'incertitudes dans les calculs thermo-hydrauliques de chocs thermiques pressurisés sur la cuve d'un REP de 900 MWe. Technical report, Note Technique CEA/DEN/CAD/DER/SESI/LCFR/NT DR 9 22/10/08, 2008.
- [3] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering : Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.