

Modèles hiérarchiques ou pas ?

- L'essentiel des études a porté sur des modèles de co-krigeage autoregressifs avec un coefficient d'ajustement constant ou simple.

Ce sont des modèles hiérarchiques avec structure hiérarchique simple.

Essentiellement OK pour des codes à convergence numérique dégradée, pour des codes Monte Carlo.

Pour ces modèles : on retrouve les mêmes difficultés que le krigeage (parallélisation de plans séquentiels, variables d'entrée discrètes/catégorielles, ...)

Structure hiérarchique complexe

Il existe des situations où la relation hiérarchique ne se laisse pas décrire comme un simple coefficient d'ajustement (exposés de Mathias De Lozzo et Federico Zertuche).

- Autre exemple : calcul de la croissance d'instabilités (hydrodynamiques par exemple, type Richtmyer-Meshkov ou Rayleigh-Taylor, pour la description d'une interface entre deux fluides $z = d(t) + a(x, t)$).
 - code complet et/ou expérience (lourd),
 - formule analytique approchée pour le taux de croissance des modes en régime linéaire et un écoulement de base simplifié (typiquement $\hat{a}(k, t) = \hat{a}_0(k) \exp(\gamma_k t)$),
 - code linéarisé (encore assez simple, code 1d pour l'écoulement de base, code 2d-3d linéarisé pour la perturbation).
 - métamodèle (du type modèle de Haan, ...), qui prédit l'amplitude des modes en régime non-linéaire à partir de leur amplitude en régime linéaire via une formule de saturation du type $\hat{a}_{non-lin}(k, t) = F(\hat{a}_{lin}(k, t))$, avec $F(a) \sim \ln(1 + a)$.
- Peut-on faire mieux pour le métamodèle ?

Structure non-hiérarchique

Il existe des situations où la relation entre les métamodèles n'est pas hiérarchique (exposé de Bertrand Iooss, sur les simulateurs concurrentiels).

- Autre exemple : neutronique ou photonique.
 - code complet : équation de transfert radiatif,
 - code Monte Carlo (bruité), avec plusieurs niveaux (hiérarchique),
 - code de diffusion (approché).

Pour les deux codes simples, on a une idée du type d'erreur commise.

Comment utiliser les deux métamodèles ?

Apprentissage séquentiel robuste sur une famille de métamodèles

- Mélanges de modèles.

Si on a plusieurs métamodèles $M_i(x)$, $i = 1, \dots, N_m$,
si on a observé (y_1, \dots, y_{n-1}) en (x_1, \dots, x_{n-1}) ,
alors la prédiction d'un nouveau point x_n est

$$\hat{y}_n = \sum_{i=1}^{N_m} \hat{\alpha}_{i,n} M_i(x_n)$$

↔ Le statisticien cherche une stratégie \mathcal{S} permettant de déterminer $\hat{\alpha}_n = (\hat{\alpha}_{i,n})_{i=1}^{N_m}$ à partir de (x_1, \dots, x_{n-1}) , (y_1, \dots, y_{n-1}) .

- Philosophie assez différente (on ne cherche pas -a priori- à améliorer les modèles; on peut avoir beaucoup de modèles).
- Domaine de recherche actif en apprentissage statistique (méthode de bandits, etc).

- Apprentissage.

On se donne une fonction de perte $\ell(y, y')$ ($= |y - y'|^2$, ou autre chose),
on regarde la perte cumulée obtenue par un statisticien utilisant la stratégie \mathcal{S}

$$\hat{L}_n(\mathcal{S}) = \sum_{t=1}^n \ell\left(\sum_{i=1}^{N_m} \hat{\alpha}_{i,t} M_i(x_t), y_t\right)$$

on regarde la perte qu'on aurait eue avec le choix $\hat{\alpha}_t = \alpha \forall t \leq n$,

$$L_n(\alpha) = \sum_{t=1}^n \ell\left(\sum_{i=1}^{N_m} \alpha_i M_i(x_t), y_t\right)$$

on considère le regret

$$R_n(\mathcal{S}) = \hat{L}_n(\mathcal{S}) - \min_{\alpha} L_n(\alpha)$$

\Leftrightarrow Le statisticien cherche une stratégie \mathcal{S} permettant de déterminer le $\hat{\alpha}_n = (\hat{\alpha}_{i,n})_{i=1}^{N_m}$
minimisant le regret.

Remarque :

La perte cumulée d'un statisticien utilisant la stratégie \mathcal{S} se décompose en

$$\hat{L}_n(\mathcal{S}) = \min_{\boldsymbol{\alpha}} L_n(\boldsymbol{\alpha}) + R_n(\mathcal{S})$$

- le premier terme est la performance de la meilleure combinaison des métamodèles (erreur d'approximation),
- le second terme est le regret (erreur d'estimation).

Note : Plus la famille de métamodèles est grande, plus l'erreur d'approximation est faible, plus l'erreur d'estimation est forte.

On souhaite trouver le $\hat{\alpha}_n$ qui minimise le regret

$$\hat{L}_n - \min_{\alpha} L_n(\alpha)$$

- Stratégie \mathcal{S}_η de pondération par poids exponentiels des pertes cumulées :

$$\hat{\alpha}_{i,n} = \frac{\exp\left(-\eta \sum_{t=1}^{n-1} l_{i,t}\right)}{\sum_{j=1}^{N_m} \exp\left(-\eta \sum_{t=1}^{n-1} l_{j,t}\right)}$$

avec $l_{i,t} = \ell(M_i(x_t), y_t)$.

Majoration d'un pseudo-regret (uniformément en les observations)

$$\hat{L}_n(\mathcal{S}_\eta) - \inf_{i=1,\dots,N_m} L_n(\mathbf{e}_i) \leq \frac{\ln N_m}{\eta} + \eta n$$

avec $L_n(\mathbf{e}_i) = \sum_{t=1}^n l_{i,t}$.

Calcul d'une bonne vitesse d'apprentissage η (de η_n , de $\eta_{t,n}$), raffinement pratique (adaptatif).

Compétitif pour apprendre vite le meilleur métamodèle.

- Stratégie $\tilde{\mathcal{S}}_\eta$ de pondération par poids exponentiels des gradients des pertes :

$$\hat{\alpha}_{i,n} = \frac{\exp\left(-\eta \sum_{t=1}^{n-1} \tilde{l}_{i,t}\right)}{\sum_{j=1}^{N_m} \exp\left(-\eta \sum_{t=1}^{n-1} \tilde{l}_{j,t}\right)}$$

avec $\tilde{l}_{j,t} = \nabla \ell\left(\sum_{i=1}^{N_m} \hat{\alpha}_{i,t} M_i(x_t), y_t\right) \cdot M_j(x_t)$.

Majoration du regret (uniformément en les observations)

$$R_n(\tilde{\mathcal{S}}_\eta) = \hat{L}_n(\tilde{\mathcal{S}}_\eta) - \inf_{\alpha} L_n(\alpha) \leq \frac{\ln N_m}{\eta} + \eta n$$

Compétitif pour apprendre vite la meilleure combinaison de métamodèles.

- Construction d'un ensemble de métamodèles (quelle taille N_m en fonction de n pour optimiser le compromis erreur d'approximation/erreur d'estimation, ...) ?
- Versions en situations non-stationnaires $\alpha(x)$?
- Versions avec $M_i(x; x_1, \dots, x_{n-1}, y_1, \dots, y_{n-1})$?
- Quantification de l'incertitude sur le modèle agrégé ?
- Retour sur le krigage/co-krigage : on peut utiliser ce type de méthodes (d'agrégation) pour considérer différentes structures de fonctions de regression et de fonction de covariance (réduction de la dimension, etc).