Paramètres effectifs et modèles non-linéaires

Alexandre CONSTANTIN[1] Rodrigo CABRAL FARIAS[2] Jean-Marc BROSSIER[1] Olivier MICHEL[1]

[1]Univ. Grenoble-Alpes, CNRS, Grenoble-INP, GIPSA Lab, 38000 Grenoble

[2]Univ. Côte d'Azur, CNRS, Laboratoire I3S, Sophia-Antipolis 06900

Résumé – Nous présentons une synthèse de la littérature autour des questions de degrés de liberté et en proposons un estimateur applicable à tout modèle paramétrique. Des résultats sur un perceptron avec une couche cachée montrent des performances équivalentes à l'algorithme de Ye pour une complexité numérique réduite.

Abstract – A summary of the litterature on degrees of freedom is presented and an estimator for non-linear models is proposed. Results on a one hidden layer perceptron show equivalent behavior as Ye's algorithm, with lower numerical complexity.

1 Introduction et notations

Pendant longtemps, notre compréhension d'un phénomène observé s'est trouvée étroitement liée à notre capacité à en donner une représentation mathématique parcimonieuse. Cette dernière doit à la fois rendre compte des caractéristiques (temporelles, spatiales, statistiques, ...) des observations, permettre la prédiction d'observations à venir, et par sa simplicité, en expliquer la nature. Dans ce contexte, les modèles linéaires ont une place de choix. La prise en compte de non-linéarité dans les modèles s'est développée au siècle dernier, le plus souvent en proposant des classes de non linéarité restreintes (*e.g.* polynomiale) et formulées à l'aide de modèles linéaires en leurs paramètres. Les modèles de Volterra en sont un exemple. Ces paradigmes trouvent un intérêt pratique grâce à leur propriété de parcimonie, liée à leur explicabilité, et grâce aux travaux et résultats permettant de caractériser leur fiabilité, leur stabilité, et d'évaluer la pertinence des modèles considérés.

L'émergence des approches d'"Intelligence Artificielle" s'appuie sur un renoncement au principe de parcimonie. Les modèles proposés par l'IA sont beaucoup plus complexes, en général très sur-paramétrés et difficilement explicables ou caractérisables théoriquement. Si cette complexité éloigne la possibilité de développer des modèles explicatifs (au sens physique) des observations, elle permet le développement de modèles prenant en compte des dépendances non linéaires (presque) quelconques entre les variables observées. Les succès de l'IA ne sont pas contestables à cet égard ; les techniques de régularisation permettent de contrôler partiellement le risque de sur-apprentissage, et la complexité des modèles. Nous proposons dans cet article d'appréhender la question suivante : quels sont les degrés de libertés d'un modèle pour représenter un signal ? Nous considérons le cadre d'analyse des séries temporelles[1]. Après avoir revisité la notion de degrés de libertés et leur articulation avec le nombre de paramètres libres du modèle dans les cas linéaires (en leur paramètres), quelques résultats et réflexions sur le cas non linéaire (de type perceptron) sont proposés.

Notations. On observe $T = \{(\mathbf{x}_i, y_i), i \in 1, \dots, N\}$ un ensemble d'entraînement pour l'identification d'un modèle $y = m_{\boldsymbol{\theta}^\star}(\mathbf{x}) + \varepsilon$ où $m_{\boldsymbol{\theta}^\star}$ désigne le modèle inconnu, ε est une erreur de prédiction, $y \in \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^q$.

L'idenfication d'un modèle $m^P_{\boldsymbol{\theta}}$ à P paramètres conduit à estimer $\boldsymbol{\theta}^\star$ par $\hat{\boldsymbol{\theta}} \in \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$. Le risque $J(\boldsymbol{\theta})$ s'écrit :

$$J(\boldsymbol{\theta}) = \mathbb{E}[C(m^P_{\boldsymbol{\theta}}(\mathbf{x}) - y)]$$

où l'espérance est prise par rapport à la densité conjointe $p(\mathbf{x}, y)$. Un choix habituel pour la fonction de coût C est le coût quadratique.

On note $\hat{y} = m^P_{\hat{\boldsymbol{\theta}}}(\mathbf{x})$ la prédiction de y par le modèle $m^P_{\hat{\boldsymbol{\theta}}}$, ou sous forme 'vectorisée' $\hat{\mathbf{y}} = [\hat{y_1}, \dots, \hat{y_N}]^T$ et on note $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{N \times q}$.

La loi $p(\mathbf{x}, y)$ étant inconnue, le risque est estimé empiriquement par :

$$J_T(\boldsymbol{\theta}) = N^{-1} \sum_{(\mathbf{x}_i, y_i) \in T} [C(m^P_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)].$$

Le choix de P ne peut pas être fait par minimisation de J_T pour des modèles imbriqués (les modèles d'ordre $P_1 > P_0$ contiennent les modèles d'ordre P_0). D'autres critères sont nécessaire pour choisir P.

2 Sélection de modèles

L'estimation de P s'appuie classiquement sur deux approches : d'une part les critères informationnels [6] (BIC, AIC) d'autre part les méthodes de régularisation [4]. AIC et BIC supposent des observations i.i.d.. La forme usuelle de BIC suppose aussi un *a priori* uniforme sur $\boldsymbol{\theta}$ et approche la vraisemblance à l'ordre deux autour de $\hat{\boldsymbol{\theta}}$. AIC repose sur la minimisation d'un développement au second ordre de la divergence de Kullback-Leibler entre la loi des observations et la loi paramétrée par $\hat{\boldsymbol{\theta}}$. Ces approximations font que ni BIC ni AIC ne donnent de bons résultats pour P (très) grand. Les approches par régularisation sont une réponse possible à ce problème. [8] établit les liens entre le choix d'*a priori* particuliers pour les critères Bayésiens (BIC) et les méthodes de régularisation 'Ridge' et *'LASSO'* rappelées dans la section suivante. Pour AIC, quelques auteurs [3] ont proposé empiriquement de remplacer la variable P dans l'expression du critère par la notion de *degrés de liberté* (DoF, Degrees of Fredom), discutée dans la section suivante.

[1]Ce travail bénéficie du soutient de la chaire MIAI "Environmental issues underground" de l'Institut MIAI@Grenoble Alpes (Programme "Investissements d'avenir" ANR–19–P3IA–0003, France).

2.1 DoF et nombre de paramètres

2.1.1 Cas des modèles linéaires en $\boldsymbol{\theta}$

Pour un modèle linéaire en ses paramètres, $q = P$, et $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta}$. En l'absence de connaissance sur la densité de probabilité $p(y|\mathbf{x})$, on considérera une fonction de coût quadratique ; la minimisation du risque empirique conduit à minimiser

$$\mathcal{L}(\boldsymbol{\theta}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2, \tag{1}$$

$\mathcal{L}(\boldsymbol{\theta})$ est l'opposé de la log-vraisemblance de $\boldsymbol{\theta}$ pour T si $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ où $\boldsymbol{\mu} = m_{\boldsymbol{\theta}}(\mathbf{X})$. L' approche est optimale (au sens bayésien) sous cette hypothèse.

Le problème de sélection de variables utiles conduit à ajouter à (1) un terme de pénalisation fonction de $\boldsymbol{\theta}$, conduisant à minimiser le coût régularisé *ridge* :

$$\mathcal{L}(\boldsymbol{\theta}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 + \eta\, \mathrm{pen}(\boldsymbol{\theta}) \tag{2}$$

où η permet de régler le compromis entre le terme d'attache aux données et le terme de régularisation. Le choix de la régularisation 'Ridge', $\mathrm{pen}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$ favorise le choix de solutions calculables analytiquement, 'simples', *i.e.* contraintes à avoir des composantes de valeurs proches de zéro. Des solutions parcimonieuses (avec peu de composantes non nulles) sont obtenues avec $\mathrm{pen}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_\alpha$, $\alpha \in [0, 1]$. Le choix $\alpha = 1$ ('LASSO') conduit à un problème d'optimisation convexe (résolu numériquement) [1]. Une première approche des DoF est introduite dans le cas de régression 'Ridge' pour les modèles linéaires.

Les solutions d'un problème de regression linéaire en paramètres sont de la forme $\hat{\mathbf{y}} = \hat{\mathbf{S}}_\eta \mathbf{y}$ où $\hat{\mathbf{S}}_\eta$ est la "*hat*" matrix. Une définition du nombre de DoF (noté temporairement dP) est proposée dans [4, Section 5.4], $\mathrm{dP} = \mathrm{Tr}(\mathbf{S}_\eta)$. Dans le cas 'Ridge', $\hat{\mathbf{S}}_\eta = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X} + \eta\mathbf{I}\right)^{-1}\mathbf{X}^\top$ et

$$\mathrm{dP} = \sum_{i=1}^{P} \frac{\lambda_i}{\lambda_i + \eta}, \tag{3}$$

où λ_i est la i-ème valeur propre de la matrice hessienne de $\mathcal{L}(\boldsymbol{\theta})$ pour le problème non régularisé (1). Pour une régression 'LASSO', dP est le nombre de composantes non nulles de $\hat{\boldsymbol{\theta}}$ [7]. Quelle que soit la régularisation, dans le cas linéaire $\mathrm{dP} \leq P$.

2.1.2 Cas des modèles non-linéaire en $\boldsymbol{\theta}$

Dans le cas non-linéaire (*e.g.* les perceptrons muticouches), P est (très) grand, conduisant à ce que les critères informationnels tels BIC et AIC ne donnent pas de résultats satisfaisants. Soit $\mathcal{H}$ l'hypothèse $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbb{I})$: les observations y_i sont i.i.d., de moyenne $\boldsymbol{\mu} = [m_{\boldsymbol{\theta}}(\mathbf{x}_1), \ldots, m_{\boldsymbol{\theta}}(\mathbf{x}_N)]^\top$ et le bruit d'observation est à distribution gaussienne, centré, blanc, de variance σ^2. Suivant Ye [10], l'*optimisme* (évaluant l'erreur à la moyenne supposée des prédictions) du modèle défini par $\mathbb{E}(\|\hat{\mathbf{y}} - \boldsymbol{\mu}\|_2^2) = \mathbb{E}(\|\hat{\mathbf{y}} - \mathbf{y}\|_2^2) + 2\sum_{i=1}^N \mathrm{cov}(y_i, \hat{y}_i) - N\sigma^2$, conduit à proposer la définition des degrés de liberté suivante :

Définition 1 (DoF [10]) *Le nombre de degrés de liberté, notés* df*, est défini pour des échantillons gaussiens i.i.d., par :*

$$\mathrm{df} \triangleq \frac{1}{\sigma^2}\sum_{i=1}^{N} \mathrm{cov}(y_i, \hat{y}_i). \tag{4}$$

Dans le cas particulier des modèles linéaires précédents, sous l'hypothèse $\mathcal{H}$,

$$\begin{aligned}\mathrm{df} &= \frac{1}{\sigma^2}\sum_{i=1}^{N} \mathrm{cov}(y_i, \sum_{k=1}^{N} (\mathbf{S}_\eta)_{i,k} y_k) \\ &= \frac{1}{\sigma^2}\sum_{i=1}^{N} (\mathbf{S}_\eta)_{i,i}\mathrm{cov}(y_i, y_i) \\ &= \mathrm{Tr}(\mathbf{S}_\eta) = \mathrm{dP}\end{aligned}$$

Par la suite, le nombre de DoF (aussi appelé 'nombre effectif de paramètres') sera toujours noté df. Dans le cas des modèles non linéaires, l'eq. 4 ne permet pas de développer une méthode opérationnelle d'estimation de df (en particulier du fait que σ est inconnu). Il est cependant possible d'exploiter le lemme de Stein [5]. Les données étant indépendantes sous H (indispensable), ce lemme énonce l'égalité

$$\mathrm{df} = \sum_{i=1}^{N} \mathbb{E}\left(\frac{\partial \hat{y}_i}{\partial y_i}\right). \tag{5}$$

Les quantités sous la somme sont cette fois des statistiques observables sur lesquelles Ye bâtit l'algorithme d'estimation suivant :

Algorithme 1 : (Ye [10]) Estimation de df (y_i i.i.d.)

pour $k = 1, \ldots, K$ **faire**
  Générer $\Delta_k \sim \mathcal{N}(0, \tau^2)$
  Evaluer $\hat{\mathbf{y}}_k$ pour le modèle appris par $\mathbf{y} + \Delta_k$
fin
Pour chaque échantillon y^i, $i = 1, \ldots, N$
Calculer $\hat{h}_i$ le vecteur des pentes de régression tel que $[\hat{\mathbf{y}}^i_{k=1}, \ldots, \hat{\mathbf{y}}^i_{k=K}]^\top = h_i[\Delta^i_{k=1}, \ldots, \Delta^i_{k=K}]^\top$
Renvoyer $\hat{\mathrm{df}} = \sum \hat{h}_i$

Ye reporte l'observation de la convergence de df pour $K \simeq 100$ et $\tau^2 \simeq .6\sigma^2$ (nécessite de connaître un ordre de grandeur de σ^2). Pour chacune des K perturbations réalisées, $m_{\boldsymbol{\theta}}$ doit être appris. Le coût de cet algorithme peut devenir exorbitant. Dans la section suivante, un estimateur de df ne nécessitant pas l'estimation de σ est introduit, puis étendu au cas d'observations non i.i.d.

2.2 Séries temporelles, modèles non linéaires, DoF

On s'intéresse au problème de régression non linéaire d'ordre q suivant : identifier $m_{\boldsymbol{\theta}}$ dans le cas $\mathbf{x}_n = [x_1(n), \ldots, x_q(n)]^T$; l'ordre de la régression q est différent de P, nombre de paramètres du modèle, et différent de df nombre de paramètres effectifs ou nombre de DoF. Notons que la valeur $\hat{\boldsymbol{\theta}}$ obtenue par minimisation du risque empirique $\mathcal{L}$ (satisfaisant $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$) est une fonction implicite des données d'apprentissage : $\hat{\boldsymbol{\theta}} = f(\mathbf{y})$. On note dans la suite $\hat{\mathbf{y}} = m \circ f(\mathbf{y})$ afin de souligner la composition de deux fonctions, le modèle et l'apprentissage.

2.2.1 Estimation de df, cas i.i.d.

On suppose dans cette section que les y_i dans l'ensemble d'apprentissage sont i.i.d. ; Les calculs détaillés de cette section sont disponibles ici [2].

Proposition 1 *Soient $\mathbf{H}_{\boldsymbol{\theta}}\mathcal{L}$ et $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ respectivement la matrice Hessienne et le gradient de $\mathcal{L}$ par rapport au vecteur de paramètres $\boldsymbol{\theta}$. Soit $\mathbf{J}_{\mathbf{x}}\mathbf{y} \in \mathbb{R}^{m\times n}$ la matrice jacobienne telle que, pour $\mathbf{y} \in \mathbb{R}^m$ et $\mathbf{x} \in \mathbb{R}^n$, $(\mathbf{J}_{\mathbf{x}}[\mathbf{y}])_{i,j} = \partial y_i/\partial x_j$; alors* $\hat{\mathrm{df}}$ *introduit ci-dessous, est un estimateur du nombre de degrés de liberté :*

$$\hat{\mathrm{df}} = \mathrm{Tr}\left(-\mathbf{J}_{\boldsymbol{\theta}}[m_{\boldsymbol{\theta}}(\mathbf{y})]\left[\mathbf{H}_{\boldsymbol{\theta}}\mathcal{L}\right]^{-1}\mathbf{J}_{\mathbf{y}}[\nabla_{\boldsymbol{\theta}}\mathcal{L}]\right), \qquad (6)$$

évalué pour $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ (satisfaisant $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\hat{\boldsymbol{\theta}}) = \mathbf{0}$).

Remarque : L'expression de df (eq. 6) utilise l'égalité $\hat{\mathbf{y}} = m \circ f(\mathbf{y})$ pour obtenir $\mathbf{J}_{\mathbf{y}}[\hat{\mathbf{y}}] = \mathbf{J}_{f(\mathbf{y})}[\hat{\mathbf{y}}]\mathbf{J}_{\mathbf{y}}[f(\mathbf{y})]$. Le terme $\mathbf{J}_{\mathbf{y}}[f(\mathbf{y})]$ n'est pas évaluable car les paramètres ($\boldsymbol{\theta} = f(\mathbf{y})$) sont fonctions implicites des sorties ($\mathbf{y}$). Le gradient s'annulant au point de convergence, le théorème des fonctions implicites nous donne la décomposition suivante :

$$J_{\mathbf{y}}[f(\mathbf{y})] = -\left[J_{f(\mathbf{y})}[\nabla_{\boldsymbol{\theta}}\mathcal{L}]\big|_{(\mathbf{y},\hat{\boldsymbol{\theta}})}\right]^{-1} J_{\mathbf{y}}[\nabla_{\boldsymbol{\theta}}\mathcal{L}]|_{(\mathbf{y},\hat{\boldsymbol{\theta}})},$$

où la quantité $J_{\boldsymbol{\theta}}[\nabla_{\boldsymbol{\theta}}\mathcal{L}]|_{(\mathbf{y},\hat{\boldsymbol{\theta}})} = \mathbf{H}_{\boldsymbol{\theta}}\mathcal{L}$ est évaluée au point de convergence de l'algorithme de minimisation de la fonction de coût $\mathcal{L}$. Finalement, la matrice jacobienne $\mathbf{J}_{f(\mathbf{y})}[\hat{\mathbf{y}}] = \mathbf{J}_{\boldsymbol{\theta}}[\hat{\mathbf{y}}]$ peut être directement évaluée.

Cette proposition permet, sous réserve d'existence de $[\mathbf{H}_{\boldsymbol{\theta}}\mathcal{L}]^{-1}$, d'évaluer un estimateur de df pour tout modèle $m_{\boldsymbol{\theta}}$ sans utiliser l'algorithme 1, donc sans avoir à estimer σ.

Propriété : Si le modèle $m_{\boldsymbol{\theta}}$ est linéaire en $\boldsymbol{\theta}$, l'expression de $\hat{\mathrm{df}}$ dans l'éq. 6 est équivalente à $\mathrm{Tr}(\mathbf{S}_\eta)$, voir [2].

Remarque : L'équation 6 est à rapprocher de l'expression du critère NIC introduit dans [9]. Cependant, le critère s'appuie sur l'espérance de $\mathbf{H}_{\boldsymbol{\theta}}\mathcal{L}$ difficile à estimer.

2.2.2 Estimation de df, cas non indépendant

Les observations sont supposées identiquement distribuées mais non indépendantes. C'est par exemple le cas pour la recherche de modèles auto-régressifs (AR) où $x_i(n) = y(n-i)$. On a alors $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. La matrice $\boldsymbol{\Sigma}$ est supposée inversible. On établit alors

$$\mathbb{E}(d_{\boldsymbol{\Sigma}}(\hat{\mathbf{y}}, \boldsymbol{\mu})) = \mathbb{E}(d_{\boldsymbol{\Sigma}}(\mathbf{y}, \hat{\mathbf{y}})) + 2\mathrm{df} - N, \qquad (7)$$

et

$$\mathrm{df} = \mathrm{Tr}(\boldsymbol{\Sigma}^{-1}\mathrm{cov}(\mathbf{y}, \hat{\mathbf{y}}))$$

où $d_{\boldsymbol{\Sigma}}(\mathbf{y}, \hat{\mathbf{y}})$ est $\boldsymbol{\Sigma}$-distance de Mahanalobis ; les équations précédentes généralisent les expressions de l'*optimisme* et de df obtenues dans le cas i.i.d., via la prise en compte d'une distance adaptée à la structure de corrélation du bruit. De manière équivalente, ces résultats peuvent se retrouver par blanchiment des données : $\mathbf{y} \leftarrow \mathbf{y}_b = \mathbf{Q}^{-1}\mathbf{y}$ où $\mathbf{Q}$ est le facteur de Choleski, $\boldsymbol{\Sigma} = \mathbf{Q}\mathbf{Q}^T$, voir [2].

3 Simulations

Pour les simulations, nous générons des observations i.i.d. obtenues par des modèles polynômiaux composées de 5 monômes : $y_i = 0.1x_{i,1}^5 - 0.4x_{i,1}^2x_{i,2}^2 + 0.8x_{i,1}^2x_{i,2} - 0.01x_{i,2}^3 + 0.4x_{i,2} + \epsilon$, avec $i = 1, \ldots, N$ et $\epsilon \sim \mathcal{N}(0, \sigma^2)$, où $\sigma^2 = 0.1$. Le vecteur d'entrée $\mathbf{x}_i = [x_{i,1}, x_{i,2}]^\top \in \mathbb{R}^2$ est généré uniformément dans le compact $[-1, 1]$ et le nombre de réalisations dans l'ensemble d'apprentissage est $N = 10000$.

Le calcul du nombre de degrés de liberté, pour ces données et relatif à un perceptron totalement connecté à une couche cachée, obtenus par l'algorithme de Ye (σ^2 connu) est présenté en Section 3.1 et l'estimateur correpondant est décrit en Section 3.2.

3.1 Algorithme de Ye et convergence

Cette section présente les résultats obtenus par l'algorithme de Ye si on suppose connue la variance des données.

Le modèle est un perceptron totalement connecté à une couche cachée composée de d neurones avec une fonction d'activation $x \mapsto \tanh(x)$. Il y a deux neurones en entrée et un en sortie. Ainsi $P = (2d + d) + (d + 1)$ paramètres.

En pratique, le calcul de la matrice Hessienne a été faite de manière analytique. Le code, incluant les détails d'implémentation, et le calcul de la matrice Hessienne sont disponibles en ligne [2].

Pour terminer, les paramètres à l'initialisation, notés $\boldsymbol{\theta}_0$, sont générés selon la loi Normale tronquée (*He-Normal*, du package *tensorflow*). L'apprentissage se fait sur 600 epochs par descente de gradient stochastique afin d'assurer l'annulation du gradient à convergence. La fonction objectif correspond au risque empirique régularisé ('Ridge').

Nous appliquons l'Algorithme 1 en supposant la variance $\sigma^2 = 0.1$ connue. Pour chaque itération k nous réinitialisons les paramètres à $\boldsymbol{\theta}_0$.

La Figure 1 présente la convergence de l'algorithme pour deux perceptrons de complexité différentes ($d = 2$, $d = 5$ respectivement) correspondant à $P = 9$ et $P = 21$ respectivement. On remarquera que, comme attendu, le nombre de degrés de liberté dépasse le nombre d'entrées contrairement au cas linéaire où l'on attendrait de ne pas dépasser le nombre de monômes.

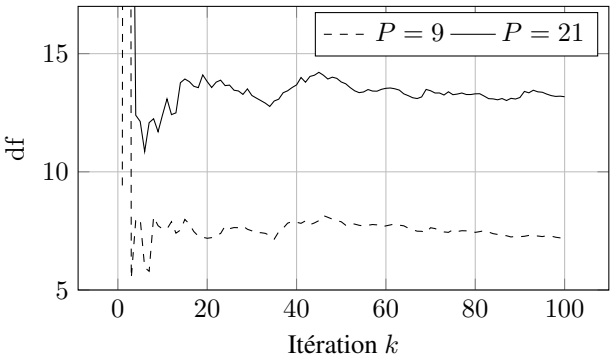

FIGURE 1 : Convergence de l'algorithme de Ye pour 100 itérations sans régularisation, pour respectivement $d = 2$ et $d = 5$ neurones.

Quelques résultats, avec régularisation 'Ridge' ($\eta = 10^{-2}$) pour un nombre de paramètres croissant, sont reportés dans le Tableau 1. D'autres résultats sont reportés en Figure 2.

3.2 Estimation des degrés de liberté

Nous testons deux configuration, sans régularisation ($\eta = 0$) et avec régularisation 'Ridge' ($\eta = 10^{-2}$) et reportons les résultats en Figure 2. Nous remarquons qu'une forte régularisation

TABLE 1 : Degrés de liberté avec régularisation ($\eta = 10^{-2}$) en fonction du nombre de paramètres P. Notre estimateur $\hat{\mathrm{df}}$ (6) est reporté en deuxième ligne.

P	9	13	21	25	29	37	41
Ye [10]	3.35	3.22	3.27	3.22	3.24	3.17	3.21
$\hat{\mathrm{df}}$	1.65	2.72	2.91	2.97	2.96	3.00	2.99

empêche le modèle d'avoir trop de degrés de libertés. Sans régularisation, les DoF croissent en fonction du nombre de paramètres avant d'osciller autour de $\mathrm{df} = 12$.

Nous soulignons que les résultats obtenus ne nécessitent d'apprendre qu'un seul modèle par architecture, réduisant significativement la complexité numérique. Les résultats semblent cohérents avec ceux de l'algorithme de Ye au tableau 1. Ajoutons, par ailleurs, que les résultats de l'estimateur n'utilisent pas la variance des données.

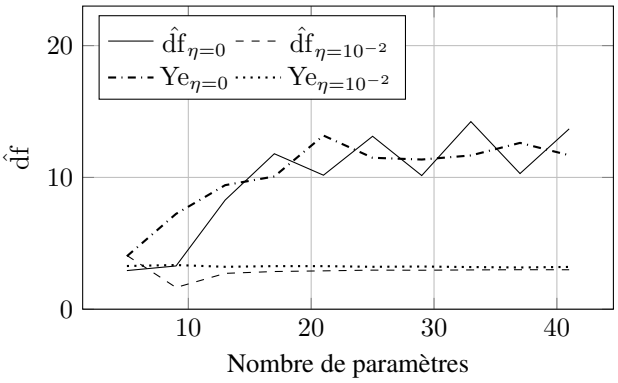

FIGURE 2 : Estimation des degrés de liberté par (6) en fonction du nombre de paramètres du modèle (augmentation de d). Avec ou sans régularisation dans le problème (2).

4 Conclusion

Nous avons présenté ici la notion d'optimisme et son lien avec le nombre de degrés de liberté. Ces relations s'appuient sur l'hypothèse d'un bruit d'observation, additif gaussien, en sortie. Dans le cas de modèles linéaires en ses paramètres, nous avons souligné que le nombre de degrés de liberté est équivalent au nombre de paramètres effectifs ; les deux définitions concordent et évaluent la même quantité.

Dans le cas non-linéaire, nous avons présenté une méthode permettant de s'affranchir de la nécessité de connaître la variance du bruit d'observation comme c'est le cas dans l'algorithme de Ye. Une discussion permettant d'évaluer le nombre de degrés de liberté dans le cas d'échantillons identiquement distribués mais non indépendants est proposée, permettant d'appréhender le cas de regression non-linéaire sur des séries temporelles.

Ce que signifie un nombre de degré de liberté reste complexe ; nous pensons qu'il permet de décrire la dimension de l'espace tangent à la fonction de coût, autour du point de convergence : cette notion est relative à la structure des données d'apprentissage et à la famille considérée de modèles paramétriques.

Nous avons illustré les approches proposées par des simulations et montré que cette proposition conduit à des résultats équivalents à ceux obtenus par l'algorithme de Ye.

Dans les perpectives, nous souhaitons étendre ces résultats à la sélection d'une architecture de modèle (au sein d'une famille de modèle), ayant le plus petit nombre de neurones avec un nombre de degrés de liberté maximal. Par ailleurs, l'estimateur requiert le calcul et l'inversion de la matrice Hessienne, en paramètres, au point de convergence. Des travaux restent à développer pour son application à des réseaux de neurones de très grande taille. Une dernière piste de travail consiste à développer une forme explicite ou estimée ("*double-backward*") de la Hessienne dans le cas de réseaux récurrents, nécessaire pour la modélisation de séries temporelles.

Références

[1] Francis BACH, Rodolphe JENATTON, Julien MAIRAL, Guillaume OBOZINSKI *et al.* : Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, 5:19–53, 2011.

[2] Alexandre CONSTANTIN : Lien vers le code et les détails de calculs. `https://alexandre-constantin.github.io/softwares.html`. Consulté le 7 Avril 2023.

[3] Bradley EFRON : The estimation of prediction error. *Journal of the American Statistical Association*, 99(467): 619–632, 2004.

[4] Trevor HASTIE, Robert TIBSHIRANI et Jerome FRIEDMAN : *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer New York, 2009.

[5] Charles M. STEIN : Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, 1981.

[6] Petre STOICA et Yngve SELEN : Model-order selection : a review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004.

[7] Robert TIBSHIRANI : Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society : Series B (Methodological)*, 58(1):267–288, 1996.

[8] Wessel N. van WIERINGEN : Lecture notes on ridge regression, 5 2021.

[9] Sumio WATANABE : Information criteria and cross validation for bayesian inference in regular and singular cases. *Japanese Journal of Statistics and Data Science*, 4(1):1–19, 2021.

[10] Jianming YE : On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93(441):120–131, 1998.