INSTITUT
POLYTECHNIQUE
DE PARIS

Transfer learning en finance

Rapport de projet de fin d'études

Théo Blanchonnet | Axel Grille

Encadrement assuré par : **Hélène Halconruy**

pendant l'année 2023-2024

Table des matières

1 Introduction **1**

2 Cadre de régression linéaire **2**
2.1 Séparation des régressions pour $\lambda = 0$ 2
2.2 Regroupement des datasets pour $\lambda \to +\infty$ 3

3 Théorèmes importants **3**
3.1 Lemme - expression de $\hat{\beta}$ et $\hat{\gamma}$ 3
3.2 Théorème - degré de liberté de λ 4
3.3 Théorème - erreur quadratique moyenne "prédictive" 4

4 Définition du jeu de données **5**

5 Les signatures **6**

6 Résultats et interprétation **7**
6.1 Application sur données non réelles 7

7 Conclusion **9**

8 Appendice **10**
8.1 Démonstration des théorèmes 10
8.1.1 Lemme 2.1 10
8.1.2 Théorème 2.1 12
8.1.3 Théorème 2.2 13

1 Introduction

Depuis quelques années, les projets d'apprentissage automatique (machine learning) se multiplient. De ce fait, leurs sources d'apprentissage se multiplient elles aussi. On parle de jeu de données ou de dataset. En général, on peut supposer que plus le dataset est grand et plus le modèle qui sera entraîné dessus sera précis et fournira de bonnes prédictions. Cependant, produire des gros jeux de données n'est pas toujours aisé. Par exemple, dans certains secteurs comme la médecine, les jeux de données doivent être annotés par des médecins ce qui est très coûteux. De plus, produire de gros jeux de données pose certains enjeux : économique (lié à la production du dataset et aux équipements), environnemental (lié à la production, au stockage et à l'entraînement du dataset et des modèles) et social (lié aux conditions de travail et aux répercussions sur l'humain). La question est alors : pouvons-nous faire de l'apprentissage frugal ? C'est-à-dire pouvons-nous conserver la performance de nos modèles tout en réduisant les ressources utilisées ? En effet, l'apprentissage par transfert ou le transfer learning semble être dans la littérature une bonne solution à la question précédente. L'objectif de ce rapport est donc de répondre à la question dans un problème de transfer learning et de régression linéaire en finance.

Le principe du transfer learning est de recycler la connaissance d'un domaine source vers un domaine cible. Par exemple, l'humain utilise déjà ce principe dans la vie de tous les jours. Lorsque l'on sait déjà jouer du violon et que l'on veut apprendre le piano, on peut réutiliser nos connaissances comme le solfège par exemple. Ce principe de recycler la connaissance définit le transfer learning. En effet, il faudra différencier différents types de transfer learning répertoriés par [4]

Posons le cadre mathématique derrière le transfer learning. Tout d'abord, un problème de machine learning se pose comme suit.

$$\mathcal{D} = (\mathcal{X}, P(X))$$

$$\mathcal{T} = (\mathcal{Y}, f)$$

Où $\mathcal{D}$ est le domaine et $\mathcal{T}$ la tâche. $\mathcal{X}$ est l'espace des features à disposition et $\mathcal{Y}$ est l'espace des labels à prédire. Enfin, $P(X)$ est la distribution de l'ensemble X (ensemble d'instances de $\mathcal{X}$) et f est la fonction de labellisation que l'on cherche afin de prédire le ou les labels. En pratique, en apprentissage supervisé, on observe les features et les labels que l'on stocke dans un dataset $D^n = \{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i \in \{1, ..., n\}\}$.

Le problème de machine learning ainsi définit facilite la définition du cadre du transfer learning. On définit un problème de transfer learning avec un domaine source $\mathcal{D}_S$ (et sa tâche associée $\mathcal{T}_S$) et un domaine cible $\mathcal{D}_T$ (et $\mathcal{T}_T$). En pratique, on considère les features et labels réellement observés $D_S^n = \{(x_i, y_i) | x_i \in \mathcal{X}^S, y_i \in \mathcal{Y}^S, i \in \{1, ..., n^S\}\}$ et $D_T^n = \{(x_i, y_i) | x_i \in \mathcal{X}^T, y_i \in \mathcal{Y}^T, i \in \{1, ..., n^T\}\}$. Le but du transfer learning est donc d'utiliser le dataset source D_S^n afin d'améliorer les performances de la fonction de labellisation cible f_T. On parle d'amélioration par rapport au cas classique dans lequel on aurait utilisé seulement le dataset cible D_T^n pour calculer f_T et prédire les labels. On parlera donc de transfert positif si les prédictions sont meilleures (selon une fonction de perte donnée) et de transfert négatif dans le cas contraire.

D'autre part, il est également possible de considérer plusieurs domaines sources et plusieurs domaines cibles. Dans la littérature, on parle alors d'adaptation de domaine. Le principe est de réduire la différence/distance entre les domaines pour améliorer les performances. L'article [3] approfondit ce champs de recherche. Cependant, dans notre cas d'étude, nous nous concentrerons sur un cas de transfer learning simple avec un seul domaine source (big) et un seul domaine cible (small). L'objectif est de comprendre l'article [2] et de l'appliquer à des données réelles en finance.

2 Cadre de régression linéaire

Le modèle de régression linéaire proposé par [2] est le suivant.

$$\begin{cases} \boxed{Y_S = X_S\beta + \epsilon_S} \text{ (Small dataset = cible)} \\ \boxed{Y_B = X_B(\beta + \gamma) + \epsilon_B} \text{ (Big dataset = source)} \end{cases}$$

Avec $\gamma \in \mathbb{R}^d$ le paramètre de transfert et $\beta \in \mathbb{R}^d$ le paramètre de régression linéaire classique. On a $Y_S \in \mathbb{R}^n$ et $Y_B \in \mathbb{R}^N$. D'autre part, les bruits ϵ_S et ϵ_B sont gaussiens centrés et de variances respectives σ_S^2 et σ_B^2.

L'objectif est donc de minimiser les moindres carrés en prenant en compte une pénalisation selon le paramètre de transfer $P(\gamma)$.

$$(\hat{\beta}, \hat{\gamma}) \in argmin_{\beta\in\mathbb{R}^d, \gamma\in\mathbb{R}^d}(\|Y_S - X_S\beta\|_2^2 + \|Y_B - X_B(\beta + \gamma)\|_2^2 + \lambda P(\gamma))$$

Désormais, nous considèrerons la fonction de pénalisation suivante : $P(\lambda) = \|X_T\gamma\|_2^2$. D'autres choix peuvent être fait pour la fonction de pénalisation P (voir [2]). Avec $X_T \in \mathbb{R}^{r\times d}$ la matrice de pénalisation. Cette matrice X_T correspond aux colonnes libres de X_S afin qu'elle soit toujours inversible. Par exemple, si X_S est inversible alors $r = n$ et $X_T = X_S$. Mais, si X_S n'est pas inversible, alors $r = d$ et on choisit les colonnes libres de X_S afin que X_T soit encore inversible. Plus simplement, $r = rang(X_S)$.

D'autre part, il est intéressant de pouvoir placer notre problème de transfer learning dans la grille de lecture de [4]. En effet, dans ce problème, nous sommes dans un cas de transfert homogène car les domaines cible (small) et source (big) partagent le même espace des features : $\mathcal{X}_S = \mathcal{X}_B$ and $\mathcal{Y}_S = \mathcal{Y}_B$. De plus, ce problème est de type inductif car les informations sont labellisées dans le domaine source $\mathcal{D}_B$ et aussi dans le domaine cible $\mathcal{D}_S$. Enfin, pour ce qui est du type de solution apporté dans ce rapport, nous nous pencherons exclusivement sur une solution de type instance-based. Ainsi, la solution transfèrera la connaissance au niveau des instances observées D_S^n et D_B^n.

2.1 Séparation des régressions pour $\lambda = 0$

Sans commencer les calculs on peut déjà remarquer que si $\lambda = 0$, il n'y a pas de pénalisation et cela revient à faire deux régressions linéaires séparemment :

$$(\hat{\beta}, \hat{\gamma}) \in argmin_{\beta\in\mathbb{R}^d, \gamma\in\mathbb{R}^d}(\|Y_S - X_S\beta\|_2^2 + \|Y_B - X_B(\beta + \gamma)\|_2^2)$$
$$(\hat{\beta}, \hat{\alpha}) \in argmin_{\beta\in\mathbb{R}^d, \alpha\in\mathbb{R}^d}(\|Y_S - X_S\beta\|_2^2 + \|Y_B - X_B\alpha\|_2^2) \text{ avec } \alpha = \beta + \gamma$$
$$\hat{\beta} \in argmin_{\beta\in\mathbb{R}^d}(\|Y_S - X_S\beta\|_2^2) \text{ et } \hat{\alpha} \in argmin_{\alpha\in\mathbb{R}^d}(\|Y_B - X_B\alpha\|_2^2)$$

On peut donc interpréter que plus λ sera petit, plus on tend vers deux régressions linéaires distinctes. Cependant, les estimateurs finaux $\hat{\beta}$ et $\hat{\gamma}$ resteront interdépendants.

2.2 Regroupement des datasets pour $\lambda \to +\infty$

D'autre part, dans le cas limite inverse où $\lambda \to +\infty$ on est dans un cas de regroupement des datasets car si $\lambda \to +\infty$ alors on est contraint d'avoir $P(\gamma) = 0$ pour amortir cet infini et que la régression ait un sens. Dans ce cas, comme la fonction P est une norme, alors, par séparation, on obtient $\gamma = 0$ ce qui revient à regrouper les deux datasets et à faire une seule régression :

$$\hat{\beta} \in argmin_{\beta \in \mathbb{R}^d}(\|Y_S - X_S\beta\|_2^2 + \|Y_B - X_B\beta)\|_2^2)$$

3 Théorèmes importants

Dans leur étude innovante sur la régression linéaire enrichie de données, Chen, Owen et Shi développent plusieurs théorèmes fondamentaux qui renforcent la méthodologie de la fusion de données dans les analyses statistiques.

3.1 Lemme - expression de $\hat{\beta}$ et $\hat{\gamma}$

Le premier résultat important est le lemme 2.1 qui établit une relation entre les estimateurs ($\hat{\beta}$ et $\hat{\gamma}$) et les données dont on dispose. Ce lemme formule la façon dont les estimateurs sont influencés par la combinaison des données de deux ensembles distincts, chacuns ayant des caractéristiques propres. Il s'agit d'un lemme important pour comprendre comment les données de sources différentes peuvent être intégrées de manière efficace, tout en conservant les propriétés statistiques nécessaires pour des estimations fiables. Ce lemme constitue la base sur laquelle repose l'application pratique de la méthode de régression enrichie de données, et il est crucial pour l'interprétation des résultats obtenus par cette approche.

Enoncé du lemme 2.1

Soit X_S, X_B et X_T tous de rang d. Alors pour tout $\lambda \geq 0$, les estimateurs minimisant les moindres carrés (pénalisés) $\hat{\beta}$ et $\hat{\gamma}$ s'expriment comme suit.

$$\boxed{\begin{cases} \hat{\beta} = W_\lambda \hat{\beta_S} + (I - W_\lambda)\hat{\beta_B} \\ \hat{\gamma} = (V_B + \lambda V_T)^{-1} V_B(\hat{\beta_B} - \hat{\beta}) \end{cases}}$$

Avec W_λ, la matrice $W_\lambda = (V_S + \lambda V_T V_B^{-1} V_S + \lambda V_T)^{-1}(V_S + \lambda V_T V_B^{-1} V_S)$
De plus, si $V_T = V_S$, alors l'expression de W_λ devient :

$$W_\lambda = (V_B + \lambda V_S + \lambda V_B)^{-1}(V_B + \lambda V_S)$$

(Démontré en Appendice 8.1.1)

3.2 Théorème - degré de liberté de λ

Parmi les résultats les plus notables, citons le théorème 2.1 qui établit une formule pour calculer le degré de liberté de la pénalisation λ.

Enoncé du théorème 2.1

Pour les régressions enrichies de données, le degré de liberté donnné par :

$$df(\lambda) = \frac{1}{\sigma_s^2} \sum_{i \in S} cov(\hat{Y}_i, Y_i) \text{ avec } \hat{Y}_S = X_S \hat{\beta}_S$$

s'expriment $df(\lambda) = tr(W_\lambda)$ avec W_λ donné dans le lemme 2.1.
Si de plus, $V_T = V_S$, alors on a :

$$\boxed{df(\lambda) = \sum_{j=1}^{d} \frac{1 + \lambda\nu_j}{1 + \lambda + \lambda\nu_j}}$$

où $\nu_1, \ldots, \nu_d$ sont les valeurs propres de

$$M = V_S^{1/2} V_B^{-1} V_S^{1/2}$$

où $V_S^{1/2}$ est la racine carrée symétrique de V_S.
(Démontré en Appendice 8.1.2)

3.3 Théorème - erreur quadratique moyenne "prédictive"

Le théorème 2.2 quant à lui, présente l'erreur quadratique moyenne "prédictive" sur $\hat{\beta}$. Ces théorèmes sont cruciaux pour comprendre comment leur méthode permettent d'optimiser l'utilisation de deux ensembles de données de tailles différentes, en particulier en régularisant l'influence des données du plus grand ensemble (B) sur le plus petit (S).

Enoncé du théorème 2.2

L'erreur quadratique moyenne "prédictive" de l'estimateur $\hat{\beta}$ est

$$\mathbb{E}(\|X_S(\hat{\beta} - \beta)\|^2) = \sigma_S^2 \sum_{j=1}^{d} \frac{(1 + \lambda\nu_j)^2}{(1 + \lambda + \lambda\nu_j)^2} + \sum_{j=1}^{d} \frac{\lambda^2 \kappa_j^2}{(1 + \lambda + \lambda\nu_j)^2}$$

où $\kappa_j^2 = u_j^T V_S^{1/2} \Theta V_S^{1/2} u_j$
et $\Theta = \gamma\gamma^T + \sigma_B^2 V_B^{-1}$

(Démontré en Appendice 8.1.3)

En outre, cette approche intégrant la pénalité L2 dans la régression enrichie est une contribution notable, permettant une analyse plus flexible et robuste en présence de données hétérogènes. Ces résultats théoriques offrent une base solide pour l'application pratique de cette méthode dans divers contextes de régression linéaire.

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
1980-12-12	0.100178	0.100614	0.100178	0.100178	469033600	0.0	0.0
1980-12-15	0.095388	0.095388	0.094952	0.094952	175884800	0.0	0.0
1980-12-16	0.088418	0.088418	0.087983	0.087983	105728000	0.0	0.0
1980-12-17	0.090160	0.090596	0.090160	0.090160	86441600	0.0	0.0
1980-12-18	0.092774	0.093210	0.092774	0.092774	73449600	0.0	0.0
...	...	...	...	...	...	...	...
2022-07-06	141.350006	144.119995	141.080002	142.919998	74064300	0.0	0.0
2022-07-07	143.289993	146.550003	143.279999	146.350006	66253700	0.0	0.0
2022-07-08	145.259995	147.550003	145.000000	147.039993	64493200	0.0	0.0
2022-07-11	145.669998	146.639999	143.779999	144.869995	63141600	0.0	0.0
2022-07-12	145.759995	148.449997	145.050003	145.860001	77523400	0.0	0.0

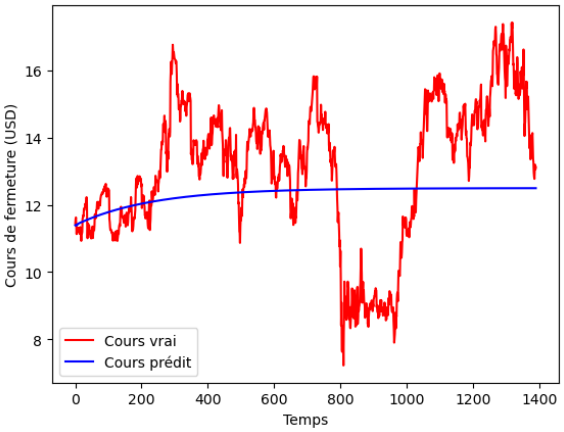
TABLE 1 – 10 lignes (5 premières et 5 dernières) extraites du fichier AAPL.csv (entreprise Apple). Les features sont : 'Date' (date du jour AAAA-MM-JJ), 'Open' (cours d'ouverture en $), 'Close' (de fermeture), 'High' et 'Low' (le minimum et le maximum journalier), 'Volume' (le volume d'action échangées), 'Dividends' (les dividendes versés) et 'Stock Splits' (les divisions d'action).

4 Définition du jeu de données

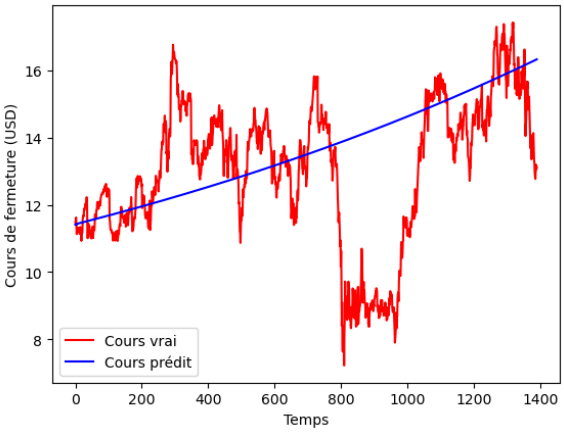
Les méthodes que nous avons présentées sont particulièrement efficaces sur de grands ensembles de données. Par "efficaces" nous entendons des performances meilleures que pour une régression linéaire classique. De ce fait, nous essayerons d'appliquer ces méthodes à un problème de prédiction financier et plus précisemment aux cours boursiers.

Dans cette section, nous allons présenter les jeux de données que nous utiliserons. Premièrement, nous avons à disposition un jeu de données regroupant les entreprises qui figurent au Standard & Poor's 500 (S&P500) depuis 1962. Ce jeu de données est disponible publiquement sur la plateforme Kaggle via ce lien : `https://www.kaggle.com/datasets/rprkh15/sp500-stock-prices/data`. Ce dossier à téléchargé, contient des fichiers .csv sur chaque entreprise figurant au S&P500 jusqu'en juillet 2022. Par exemple, le tableau 1 donne un aperçu du fichier AAPL.csv.

Ainsi, à l'aide de ce dataset contenant plus de 500 entreprises américaines, nous pensons avoir assez de données pour entraîner et tester notre modèle de transfer learning par régression linéaire enrichie. Pour s'accomoder avec ce dataset, nous avons essayé d'appliquer deux méthodes : la régression linéaire classique et la régression linéaire enrichie. Pour cela, nous avons choisit deux entreprises du même secteur (technologique) : IBM et HP. Avant tout, il est clair qu'une régression linéaire est une méthode trop simpliste pour faire de la prédiction sur des séries temporelles aussi complexes que l'évolution du prix d'une action au cours du temps. On ne s'attend donc pas à une bonne prédiction mais on cherche plutôt à s'approprier les données. La figure 1 donne une idée de la capacité prédictive des deux modèles. Comme prévu, les modèles sont trop simples pour pouvoir prédire avec précision le cours de fermeture HP. Ces modèles donnent tout de même la tendance mais ne sont pas exploitables directement dans le domaine financier dans lequel une erreur de quelques dollars est significative. Finalement, au vu de la précision de notre modèle de régression linéaire enrichie, nous



(a) Classique | RMSE = 2.21



(b) Enrichie | RMSE = 2.30

FIGURE 1 – Deux graphes correspondant aux tests des modèles de regréssion linéaire classique et enrichie. Ces deux modèles ont été entraînés sur le cours de fermeture de HP entre 2015-10-19 et 2017-01-01 ($n = 304$ données). Le modèle enrichi a été entraîné en plus sur les données de l'entreprise IBM (de 1962-01-02 à 2017-01-01 : $N = 13846$ données). Les données ont été prétraitées avec un décalage $L = 10$ dans le temps qui correspond donc au nombre de features ($d = 10$). Le graphe (a) correspond à la régression linéaire classique. Le graphe (b) correspond à la régression linéaire enrichie.

avons décidé d'y incorporer une étape de prétraitement des données avec l'objet mathématique que sont les signatures.

5 Les signatures

Dans le domaine financier, le concept de "signature" est souvent utilisé pour extraire des ensembles de caractéristiques essentielles des données. En effet, cet objet mathématique (défini ci-dessous) permet de capturer les informations les plus pertinentes et les plus discriminantes. De ce fait, en prétraitant nos données de cours de fermeture avec les signatures tronquées nous espérons améliorer la qualité de prédiction du modèle de régression linéaire enrichie comme dans l'article [1].

Définition signature

Pour un chemin continu et lisse par morceaux $x : [0, T] \to \mathbb{R}^d$, sa signature est donnée par

$$S(x) = \left(1, S(x)^{(1)}, \dots, S(x)^{(d)}, S(x)^{(1,1)}, \dots, S(x)^{(i,j)}, \dots, S(x)^{(d,d)}, \dots\right)$$

où pour $m = 1, 2, \dots$ et $i_1, \dots, i_m \in [d]$,

$$S(x)^{(i_1,\dots,i_m)} = \int_{0 \le t_1 < \cdots < t_m} dx_{t_1}^{i_1} \cdots dx_{t_m}^{i_m}.$$

La signature tronquée jusqu'au degré M est donnée par

$$S_M(x) = \left(1, S(x)^{(1)}, \dots, S(x)^{(n\dots n)}\right).$$

Il est désormais bien connu que la signature d'un chemin généré par une séquence de données détermine essentiellement le chemin d'une manière efficace sur le plan informatique.Pour calculer les signatures, il faudra donc suivre la procédure suivante :

1. Fixer un degré de troncature M (par exemple $M = 3$).
2. Pour chaque unité de temps $t = 0, 1, \ldots$, considérer le prix et le volume de l'action avec un décalage L : $\{\log(s_\tau)\}_{t-L+1\leq\tau\leq t}$ et $\{\log(v_\tau)\}_{t-L+1\leq\tau\leq t}$.
3. Considérer le chemin tri-dimensionnel $\{z_\tau\}_{t-L+1\leq\tau\leq t} = \{(\tau, \log(s_\tau), \log(v_\tau))\}_{t-L+1\leq\tau\leq t}$ et calculer sa M-ième signature tronquée $x_t = S_M(z)$, définie précédemment.
4. Reproduire les étapes précédentes pour toutes les données à disposition.

Notre but est donc d'incorporer les signatures au prétraitement des données. Et ce, dans le but d'améliorer drastiquement notre modèle de régression linéaire enrichie lorsqu'il est confronté à des processus issus de la finance. Malheureusement, notre étude des signatures n'a pas pu aboutir car lors de la mise en oeuvre des algorithmes nécessaires, nous avons été confrontés à des difficultées liées à la présence de matrices non inversibles. Ces matrices, cruciales pour le calcul des paramètres du modèle de transfer learning, se sont avérées singulières, c'est-à-dire dépourvues d'inverse. Cette singularité peut être attribuée à plusieurs facteurs, notamment la multicollinéarité des variables, une dimensionnalité trop élevée par rapport au nombre d'observations, ou encore la présence de valeurs propres trop faibles. Nous avons exploré plusieurs pistes de solution telles que des techniques de régularisation ou l'utilisation de pseudo-inverse. Cependant, ces solutions n'ont pas permis de résoudre entièrement le problème, ce qui a limité l'application du transfer learning.

6 Résultats et interprétation

6.1 Application sur données non réelles

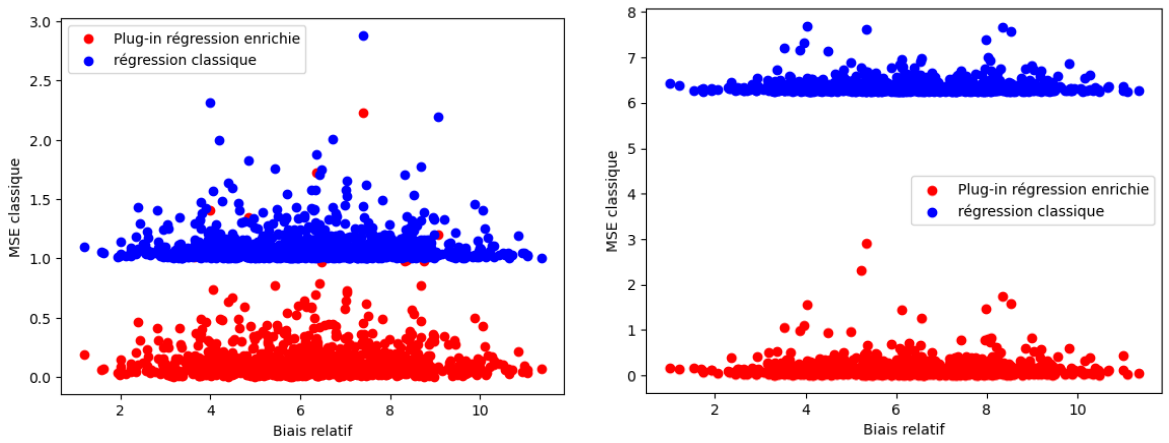
Premièrement, avant de tester le modèle complet avec la méthode de régression linéaire enrichie et le prétraitement par signature, on se propose de tester la méthode de régression enrichie seule sur des données non réelles. En effet, en se basant sur l'article [2], on a généré des données aléatoirement en partant d'une valeur de β d'origine. L'objectif est donc de vérifier si le $\hat{\beta}$ est proche de ce β d'origine en moyenne quadratique.

Nous nous plaçons dans le cadre du tableau 2. Premièrement, on remarque que les modèles enrichis et classique sont assez proches. Curieusement, le modèle classique affiche toujours un zéro pour la dernière composante de $\hat{\beta}_{classic}$. Ceci peut être du au fait que la dernière colonne de X_S est toujours une colonne de 1 quelque soit la génération. Ainsi, il est possible que la méthode LinearRegression de scikit-learn considère qu'il n'est pas nécessaire d'estimer le paramètre de la dernière composante. Deuxièmement, le modèle enrichi est légèrement plus précis que le modèle classique : nous le voyons sur la valeur du MSE qui est calculé entre la vraie valeur de β et celle de $\hat{\beta}$. De plus, la figure 2 est en accord avec cette remarque. En effet, nous avons testé pour 1000 simulations et un β constant. Le résultat montre bien que le modèle enrichi se trouve en moyenne plus performant (MSE plus faible) que le modèle classique.

Cependant, cet écart de MSE semble être environ de 1 pour la figure 2.(a) et d'environ 6 pour la figure 2.(b). On se demande donc si cet écart ne provient du fait que le modèle LinearRegression place automatiquement la dernière colonne de $\hat{\beta}_{classic}$

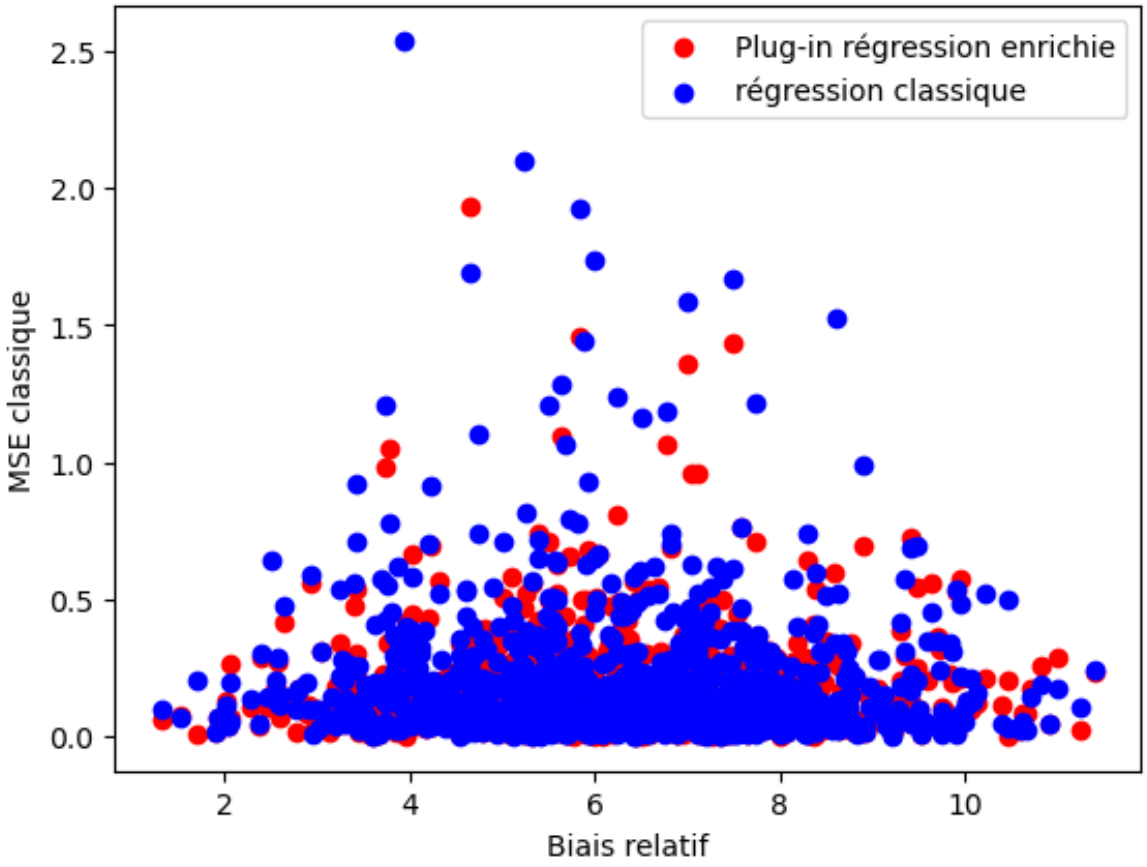
	$\beta = [0, 0, 0, 0]$	$\beta = [2, 2, 2, 2, 2]$	$\beta = [1, 2, 3, 4]$
$\hat{\beta}_{\text{enriched}}$	$[0.2, 0.04, 0.4, 0.5]$	$[2.8, 2.4, 2.1, 1.7, 1.7]$	$[-2.6, 3.4, 2.1, 4.4]$
$\hat{\beta}_{\text{classic}}$	$[0.2, 0.04, 0.4, 0.]$	$[2.8, 2.4, 2.1, 1.7, 0.]$	$[-2.6, 3.4, 2.1, 0.]$
γ	$[0.6, -1.0, -0., 0.2]$	$[-0.7, -0.5, 0.7, 0.9, -0.5]$	$[0., -0.7, 0.6, -0.3]$
$\hat{\gamma}$	$[0.4, -1.0, -0.5, -0.3]$	$[-1.5, -0.9, 0.6, 1.2, -0.2]$	$[3.6, -2.1, 1.5, -0.7]$
$\text{MSE}_{\text{enriched}}$	0.1	0.2	3.9
$\text{MSE}_{\text{classic}}$	0.05	1.0	7.9

TABLE 2 – Résultats comparatifs pour les méthodes de régression linéaire enrichie et classique pour trois valeurs différentes de β. Les valeurs sont arrondies au dixième (pour des raisons de place). γ a été généré uniformement entre -1 et 1. Les observations X_S et X_B ont été générées aléatoirement selon des lois gaussiennes centrées et de matrice de covariances respectives C_S et C_B. Les matrices C_S et C_B ont été générées à l'aide d'une loi de Wishart à $d-1$ degrés de liberté (car la dernière dimension de X_S et X_B est une colonne de 1). On en déduit ainsi les Y_S et Y_B correspondant pour faire les entraînements. $n = 10$ et $N = 10000$.



(a) $\beta = [2, 2, 2, 2]$ (b) $\beta = [5, 5, 5, 5]$

FIGURE 2 – MSE de β par rapport à $\hat{\beta}$ en fonction du biais relatif ($rb = \sqrt{n}|\gamma|_1$). Chaque point correspond à une simulation et 1000 simulations ont été faites. Les points bleus correspondent à la méthode de régression linéaire classique. Les points rouges réfèrent à la méthode de régression linéaire enrichie. Les β d'origine choisis ici sont $\beta = [2, 2, 2, 2]$ (a) et $\beta = [5, 5, 5, 5]$ (b). La méthode de génération des données est la même que dans le tableau 2 à la seule différence que C_S et C_B sont générés à l'aide de la méthode Spiked de l'article [2]. $n = 10$ et $N = 10000$.



(a) $\text{MSE}_{\text{avg,enriched}} = 0.15$ et $\text{MSE}_{\text{avg,classic}} = 0.19$

FIGURE 3 – MSE de β par rapport à $\hat{\beta}$ en fonction du biais relatif ($rb = \sqrt{n}|\gamma|_1$) pour 1000 simulations. Les points bleus correspondent à la méthode de régression linéaire classique. Les points rouges réfèrent à la méthode de régression linéaire enrichie. Le β d'origine choisi ici est toujours le même : $\beta = [2, 2, 2, 2]$ et la méthode de génération des données est la même que dans le tableau 2 à la seule différence que C_S et C_B sont générés à l'aide de la méthode Spiked mais cette fois ci de dimension d (et non $d-1$). Les valeurs de $\text{MSE}_{\text{avg,enriched}}$ et $\text{MSE}_{\text{avg,classic}}$ sont des moyennes de MSE sur tous les points présents pour les deux modèles. $n = 10$ et $N = 10000$.

à 0. Ainsi, la figure 3 reproduit le même déroulé que la figure 2 avec C_S et C_B générés à l'aide de la méthode Spiked mais cette fois ci de dimension d et non $d-1$. Le résultat montre effectivement que les modèles se chevauchent en termes de MSE. Malgré tout, en moyenne $\text{MSE}_{\text{avg,enriched}}$ est inférieur à $\text{MSE}_{\text{avg,classic}}$. La méthode de régression linéaire enrichie reste donc (mais avec une différence moins significative) la meilleure des deux.

7 Conclusion

Pour conclure sur la méthode de régression linéaire enrichie, nous n'avons pas pu exploité tout son potentiel sur des données financières. En effet, les signatures semblaient constituer un outil mathématique efficace pour une régression linéaire en finance. D'autre part, en générant des données de test aléatoirement, on remarque que la méthode de régression linéaire enrichie fournit tout de même une meilleure estimation que la méthode de régression linéaire classique.

8 Appendice

8.1 Démonstration des théorèmes

8.1.1 Lemme 2.1

Soit la fonction de coût issue des des données prédites et les vraies valeurs :

$$\begin{aligned}C(\gamma,\beta) &= \|Y_S - X_S\beta\|^2 + \|Y_B - X_B(\beta+\gamma)\|^2 + \lambda\|X_T\gamma\|^2 \\ &= \|Y_S\|^2 - 2Y_S^T X_S\beta + \beta^T X_S^T X_S\beta + \|Y_B\|^2 - 2Y_B^T X_B(\beta+\gamma) + (\beta+\gamma)^T X_B^T X_B(\beta+\gamma) + \lambda\gamma^T X_T^T X_T\gamma \\ &= \|Y_S\|^2 + \|Y_B\|^2 - 2Y_S^T X_S\beta + \beta^T X_S^T X_S\beta - 2Y_B^T X_B(\beta+\gamma) + (\beta+\gamma)^T X_B^T X_B(\beta+\gamma) + \lambda\gamma^T X_T^T X_T\gamma\end{aligned}$$

Calculons maintenant le gradient de cette fonction par rapport à β :

$$\nabla_\beta C(\beta,\gamma) = -2Y_S^T X_S - 2X_B^T Y_B + 2X_S^T X_S\beta + \nabla_\beta[\beta^T X_B^T X_B(\beta+\gamma) + \gamma^T X_B^T X_B(\beta+\gamma)]$$

avec

$$\begin{aligned}&\nabla_\beta[\beta^T X_B^T X_B(\beta+\gamma) + \gamma^T X_B^T X_B(\beta+\gamma)] \\ &= \nabla_\beta[\beta^T X_B^T X_B\beta + \beta^T X_B^T X_B\gamma + \gamma^T X_B^T X_B\beta + \gamma^T X_B^T X_B\gamma] \\ &= 2X_B^T X_B\beta + 2X_B^T X_B\gamma\end{aligned}$$

donc

$$\nabla_\beta C(\beta,\gamma) = -2Y_S^T X_S - 2X_B^T Y_B + 2X_S^T X_S\beta + 2X_B^T X_B\beta + 2X_B^T X_B\gamma$$

Finalement,

$$\nabla_\beta C(\beta,\gamma) = 0 \Leftrightarrow [X_S^T X_S + X_B^T X_B]\hat{\beta} + X_B^T X_B\gamma = X_S^T Y_S + X_B^T Y_B$$

De même par rapport a γ,

$$\nabla_\gamma C(\beta,\gamma) = -2X_B^T Y_B + 2\lambda X_T^T X_T\gamma + 2X_B^T X_B\beta + 2X_B^T X_B\gamma$$

Puis,

$$\begin{aligned}&\nabla_\gamma C(\beta,\gamma) = 0 \Leftrightarrow \lambda V_T\hat{\gamma} + V_B\hat{\beta} + V_B\hat{\gamma} = X_B^T Y_B \\ &\Leftrightarrow V_B\hat{\beta} + (V_B + \lambda V_T)\hat{\gamma} = X_B^T Y_B\end{aligned}$$

On obtient alors le système suivant :

$$\begin{cases}(V_S + V_B)\hat{\beta} = V_S\hat{\beta_S} + V_B\hat{\beta_B} - V_B\hat{\gamma}\ (1) \\ (\lambda V_T + V_B)\hat{\gamma} = V_B\hat{\beta_B} - V_B\hat{\beta}\ (2)\end{cases}$$

La matrice $(\lambda V_T + V_B)$ est inversible car la matrice X_T est inversible (donc valeurs propres non nulles) donc la matrice $V_T = X_T^T X_T$ est symétrique définie positive (et $\lambda > 0$) et la matrice V_B est symétrique. Ainsi,

$$(2) \Leftrightarrow \hat{\gamma} = (\lambda V_T + V_B)^{-1} V_B (\hat{\beta_B} - \hat{\beta})$$

Puis en injectant, (2) dans (1) on obtient :

$$\begin{aligned}
&(V_S + V_B)\hat{\beta} = V_S\hat{\beta_S} + V_B\hat{\beta_B} - V_B(\lambda V_T + V_B)^{-1} V_B(\hat{\beta_B} - \hat{\beta}) \\
\Leftrightarrow &(V_S + V_B - V_B(\lambda V_T + V_B)^{-1} V_B)\hat{\beta} = V_S\hat{\beta_S} + V_B\hat{\beta_B} - V_B(\lambda V_T + V_B)^{-1} V_B\hat{\beta_B}
\end{aligned}$$

D'autre part, on remarque que :

$$\begin{aligned}
V_B(\lambda V_T + V_B)^{-1} &= (V_B^{-1})^{-1}(\lambda V_T + V_B)^{-1} \\
&= ((\lambda V_T + V_B)V_B^{-1})^{-1} \\
&= (\lambda V_T V_B^{-1} + I_d)^{-1}
\end{aligned}$$

En remplaçant dans l'équation précédente on obtient :

$$\begin{aligned}
(2) \Leftrightarrow \hat{\beta} = &\boxed{(V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1} V_S}\hat{\beta_S} + \\
&\boxed{(V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1}(V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)}\hat{\beta_B}
\end{aligned}$$

Ainsi, pour démontrer le Lemme 2.1, on calcule les termes séparemment. On note le premier terme $A_1 = (V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1} V_S$ et $A_2 = (V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1}(V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)$.

$$\begin{aligned}
A_1 &= (V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1} V_S \\
&= [(\lambda V_T V_B^{-1} + I_d)^{-1}(\lambda V_T V_B^{-1} + I_d)(V_S + V_B) - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B]^{-1} V_S \\
&= [(\lambda V_T V_B^{-1} + I_d)^{-1}((\lambda V_T V_B^{-1} + I_d)(V_S + V_B) - V_B)]^{-1} V_S \\
&= ((\lambda V_T V_B^{-1} + I_d)(V_S + V_B) - V_B)^{-1}(\lambda V_T V_B^{-1} + I_d) V_S \\
&= (V_S + \lambda V_T V_B^{-1} V_S + \lambda V_T)^{-1}(V_S + \lambda V_T V_B^{-1} V_S) \\
&= W_\lambda
\end{aligned}$$

$$\begin{aligned}
A_2 &= (V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1}(V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B) \\
&= (V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1}(V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B - V_S) \\
&= I_d - (V_S + V_B - (\lambda V_T V_B^{-1} + I_d)^{-1} V_B)^{-1} V_S \\
&= I_d - A_1 \\
&= I_d - W_\lambda
\end{aligned}$$

On a donc bien montré que :

$$\hat{\beta} = W_\lambda\hat{\beta_S} + (I_d - W_\lambda)\hat{\beta_B}$$

Avec $W_\lambda = (V_S + \lambda V_T V_B^{-1} V_S + \lambda V_T)^{-1}(V_S + \lambda V_T V_B^{-1} V_S)$

□

8.1.2 Théorème 2.1

$$
\begin{aligned}
df(\lambda) &= \frac{1}{\sigma_S^2}\sum_{i\in S} cov(\hat{Y_i^S}, Y_i^S) \\
&= \frac{1}{\sigma_S^2} Tr\left(\begin{bmatrix} cov(\hat{Y_1^S}, Y_1^S) & . & . & cov(\hat{Y_1^S}, Y_n^S) \\ & . & & . \\ & . & & . \\ cov(\hat{Y_n^S}, Y_1^S) & . & . & cov(\hat{Y_n^S}, Y_n^S) \end{bmatrix}\right) \\
&= \frac{1}{\sigma_S^2} Tr(cov(\hat{Y_S}, Y_S)) \\
&= \frac{1}{\sigma_S^2} Tr(cov[X_S\hat{\beta}, X_S\beta + \epsilon_S])
\end{aligned}
$$

(On applique le lemme 2.1)

$$
= \frac{1}{\sigma_S^2} Tr(cov[X_S(W_\lambda\hat{\beta_S} + (I_d - W_\lambda)\hat{\beta_B}), X_S\beta + \epsilon_S])
$$

(On remplace $\hat{\beta_S} = V_S^{-1}X_S^T Y_S$ et $\hat{\beta_B} = V_B^{-1}X_B^T Y_B$)

$$
= \frac{1}{\sigma_S^2} Tr(cov[X_S W_\lambda V_S^{-1}X_S^T Y_S + X_S(I_d - W_\lambda)V_B^{-1}X_B^T Y_B), X_S\beta + \epsilon_S])
$$

(On remplace $Y_S = X_S\beta + \epsilon_S$ et $Y_B = X_B(\beta + \gamma) + \epsilon_B$)

$$
\begin{aligned}
&= \frac{1}{\sigma_S^2} Tr(cov[X_S W_\lambda\beta + X_S W_\lambda V_S^{-1}X_S^T\epsilon_S + X_S(I_d - W_\lambda)(\beta + \gamma) + X_S(I_d - W_\lambda)V_B^{-1}X_B^T\epsilon_B) \\
&\qquad , X_S\beta + \epsilon_S]) \\
&= \frac{1}{\sigma_S^2} Tr(cov[X_S W_\lambda V_S^{-1}X_S^T\epsilon_S + X_S(\beta + \gamma) - X_S W_\lambda\gamma + X_S(I_d - W_\lambda)V_B^{-1}X_B^T\epsilon_B), X_S\beta + \epsilon_S])
\end{aligned}
$$

(Billinéarité de la *cov*)

$$
= \frac{1}{\sigma_S^2} Tr(cov[X_S W_\lambda V_S^{-1}X_S^T\epsilon_S, \epsilon_S] + cov[X_S W_\lambda V_S^{-1}X_S^T\epsilon_S, X_S\beta]) + cov[R, X_S\beta + \epsilon_S]
$$

(Les bruits sont centrés et indépendants et aussi entre datasets donc :
$cov[X_S W_\lambda V_S^{-1}X_S^T\epsilon_S, X_S\beta] = 0 = cov[R, X_S\beta + \epsilon_S]$)

$$
\begin{aligned}
&= \frac{1}{\sigma_S^2} Tr(X_S W_\lambda V_S^{-1}X_S^T cov[\epsilon_S, \epsilon_S]) \\
&= \frac{1}{\sigma_S^2} Tr(X_S W_\lambda V_S^{-1}X_S^T\sigma_S^2 I_n) \\
&= Tr(X_S W_\lambda (X_S^T X_S)^{-1}X_S^T) \\
&= Tr(X_S W_\lambda (X_S)^{-1})
\end{aligned}
$$

(Car $Tr(AB) = Tr(BA)$)

$$
= Tr(W_\lambda)
$$

Or, par hypothèse, $V_T = V_S$ donc on peut simplifier l'expression de W_λ.

$$
\begin{aligned}
df(\lambda) &= Tr((V_S + \lambda V_S V_B^{-1} V_S + \lambda V_S)^{-1}(V_S^{\frac{1}{2}} V_S^{-\frac{1}{2}})(V_S + \lambda V_S V_B^{-1} V_S)) \\
&= Tr((V_S^{-\frac{1}{2}} V_S^{\frac{1}{2}})(V_S^{\frac{1}{2}} + \lambda V_S^{\frac{1}{2}} V_B^{-1} V_S + \lambda V_S^{\frac{1}{2}})^{-1}(V_S^{\frac{1}{2}} + \lambda V_S^{\frac{1}{2}} V_B^{-1} V_S)(V_S^{-\frac{1}{2}} V_S^{\frac{1}{2}})) \\
&= Tr(V_S^{-\frac{1}{2}}(I_d + \lambda V_S^{\frac{1}{2}} V_B^{-1} V_S^{-\frac{1}{2}} + \lambda I_d)^{-1}(I_d + \lambda V_S^{\frac{1}{2}} V_B^{-1} V_S^{-\frac{1}{2}}) V_S^{\frac{1}{2}}) \\
&= Tr((I_d + \lambda M + \lambda I_d)^{-1}(I_d + \lambda M)) \\
&= \sum_{i=1}^{d} \frac{1 + \lambda \nu_i}{1 + \lambda + \lambda \nu_i}
\end{aligned}
$$

Avec les $(\nu_i)_{i \in [|1,d|]}$ les valeurs propres de la matrice $M = V_S^{\frac{1}{2}} V_B^{-1} V_S^{-\frac{1}{2}}$. D'où le résultat du théorème.

□

8.1.3 Théorème 2.2

$$
\begin{aligned}
&\mathbb{E}[\|X_S(\hat{\beta} - \beta))\|^2] = \mathbb{E}[(X_S(\hat{\beta} - \beta))^T X_s(\hat{\beta} - \beta)] \\
&\text{(Or } X^T X = \|X\|^2 = Tr(XX^T) \text{)} \\
&\qquad = \mathbb{E}[Tr(X_S(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T X_s^T)] \\
&\text{(Car } Tr(AB) = Tr(BA) \text{)} \\
&\qquad = \mathbb{E}[Tr(V_S(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T)] \\
&\text{(La fonction } X \mapsto Tr(X) \text{ est linéaire)} \\
&\qquad = Tr(V_S \mathbb{E}[(\hat{\beta} - \beta)(\hat{\beta} - \beta)^T])
\end{aligned}
$$

D'autre part, on a (en utilisant le lemme 2.1) :

$$
\begin{aligned}
\mathbb{E}[(\hat{\beta} - \beta)] &= \mathbb{E}[W\hat{\beta}_S + (I - W)\hat{\beta_B} - \beta] \\
&= \mathbb{E}[W\hat{\beta}_S] + \mathbb{E}[\hat{\beta_B} - \beta] - \mathbb{E}[W\hat{\beta_B}]
\end{aligned}
$$

Or, $\mathbb{E}[W\hat{\beta}_S] = \mathbb{E}[WV_S^{-1}X_S^T Y_S] = WV_S^{-1}X_S^T \mathbb{E}[Y_S] = WV_S^{-1}X_S^T \mathbb{E}[X_S\beta + \epsilon_S] = W\beta$. De même, $\mathbb{E}[\hat{\beta_B}] = \beta + \gamma$. Ainsi, on trouve $\mathbb{E}[\hat{\beta_B} - \beta] = \gamma$ et $\mathbb{E}[W\hat{\beta_B}] = W(\beta + \gamma)$. Finalement, on a :

$$
\mathbb{E}[(\hat{\beta} - \beta)] = W\beta + \gamma - W(\beta + \gamma) = (I - W)\gamma
$$

Puis, par décomposition biais-variance on trouve :

$\mathbb{E}[(\hat{\beta}-\beta)(\hat{\beta}-\beta)^T] = biais(\hat{\beta})biais(\hat{\beta})^T + Var[\hat{\beta}]$

(Or les datasets S et B sont supposés indépendants)

$$= \mathbb{E}[(\hat{\beta}-\beta)]\mathbb{E}[(\hat{\beta}-\beta)]^T + Var[W\hat{\beta_S}] + Var[(I-W)\hat{\beta_B}]$$
$$= (I-W)\gamma\gamma^T(I-W)^T + Var[W\hat{\beta_S}] + Var[(I-W)\hat{\beta_B}]$$
$$= (I-W)(\gamma\gamma^T + \sigma_B^2 V_B^{-1} - \sigma_B^2 V_B^{-1})(I-W)^T + Var[W\hat{\beta_S}] + Var[(I-W)\hat{\beta_B}]$$

(On pose $\Theta = \gamma\gamma^T + \sigma_B^2 V_B^{-1}$)

$$= (I-W)\Theta(I-W)^T - (I-W)\sigma_B^2 V_B^{-1}(I-W)^T + Var[W\hat{\beta_S}] + Var[(I-W)\hat{\beta_B}]$$

D'autre part,

$$Var[W\hat{\beta_S}] = \mathbb{E}[(W\hat{\beta_S} - \mathbb{E}[W\hat{\beta_S}])(W\hat{\beta_S} - \mathbb{E}[W\hat{\beta_S}])^T]$$

(Or $\mathbb{E}[W\hat{\beta_S}] = W\beta$)

$$= \mathbb{E}[(W(\hat{\beta_S}-\beta))(W(\hat{\beta_S}-\beta))^T]$$
$$= W\mathbb{E}[(\hat{\beta_S}-\beta)(\hat{\beta_S}-\beta)^T]W^T$$
$$= W\mathbb{E}[\hat{\beta_S}\hat{\beta_S}^T - \hat{\beta_S}\beta^T - \beta\hat{\beta_S}^T + \beta\beta^T]W^T$$
$$= W(\mathbb{E}[\hat{\beta_S}\hat{\beta_S}^T] - \mathbb{E}[\hat{\beta_S}\beta^T] - \mathbb{E}[\beta\hat{\beta_S}^T] + \mathbb{E}[\beta\beta^T])W^T$$
$$= W(\mathbb{E}[(V_S^{-1}X_S^T Y_S)(V_S^{-1}X_S^T Y_S)^T] - \beta\beta^T - \beta\beta^T + \beta\beta^T)W^T$$
$$= W(\mathbb{E}[(\beta + V_S^{-1}X_S^T\epsilon_S)(\beta + V_S^{-1}X_S^T\epsilon_S)^T] - \beta\beta^T - \beta\beta^T + \beta\beta^T)W^T$$
$$= W(\mathbb{E}[(\beta + V_S^{-1}X_S^T\epsilon_S)(\beta + V_S^{-1}X_S^T\epsilon_S)^T] - \beta\beta^T)W^T$$
$$= W(\mathbb{E}[V_S^{-1}X_S^T\epsilon_S)(V_S^{-1}X_S^T\epsilon_S)^T])W^T$$
$$= WV_S^{-1}X_S^T\mathbb{E}[\epsilon_S\epsilon_S^T]X_S V_S^{-1T}W^T$$
$$= \sigma_S^2 WV_S^{-1}W^T$$

De même, on obtient :

$$Var[(I-W)\hat{\beta_B}] = \sigma_B^2(I-W)V_B^{-1}(I-W)^T$$

En revenant à la décomposition précédente et en remplaçant on a :

$$\mathbb{E}[(\hat{\beta}-\beta)(\hat{\beta}-\beta)^T] = (I-W)\Theta(I-W)^T - (I-W)\sigma_B^2 V_B^{-1}(I-W)^T +$$
$$Var[W\hat{\beta_S}] + Var[(I-W)\hat{\beta_B}]$$
$$= (I-W)\Theta(I-W)^T - (I-W)\sigma_B^2 V_B^{-1}(I-W)^T + \sigma_S^2 WV_S^{-1}W^T + \sigma_B^2(I-W)V_B^{-1}$$
$$= (I-W)\Theta(I-W)^T + \sigma_S^2 WV_S^{-1}W^T$$

Enfin, on revient à l'écart quadratique moyen principal :

$$\mathbb{E}[\|X_S(\hat{\beta}-\beta))\|^2] = Tr(V_S\mathbb{E}[(\hat{\beta}-\beta)(\hat{\beta}-\beta)^T])$$
$$= Tr(V_S(I-W)\Theta(I-W)^T + \sigma_S^2 V_S WV_S^{-1}W^T)$$
$$= \boxed{\sigma_S^2 Tr(V_S WV_S^{-1}W^T)} + \boxed{Tr(V_S(I-W)\Theta(I-W)^T)}$$

On calcule séparemment les deux termes encadrés :

$$
\begin{aligned}
B_1 &= \sigma_S^2 Tr(V_S W V_S^{-1} W^T) \\
&= \sigma_S^2 Tr(V_S^{\frac{1}{2}} V_S^{\frac{1}{2}} W V_S^{-\frac{1}{2}} V_S^{-\frac{1}{2}} W^T) \\
&= \sigma_S^2 Tr(V_S^{\frac{1}{2}} W V_S^{-\frac{1}{2}} \times V_S^{\frac{1}{2}} W^T V_S^{-\frac{1}{2}}) \\
&\text{(On pose } \tilde{W} = V_S^{\frac{1}{2}} W V_S^{-\frac{1}{2}} = (I + \lambda M + \lambda I)^{-1}(I + \lambda M) = U \tilde{D} U^T) \\
&= \sigma_S^2 Tr(\tilde{W} \tilde{W}^T) \\
&\text{(Avec } \tilde{D} = diag((1 + \lambda\nu_j)/(1 + \lambda + \lambda\nu_j))_{j \in [|1,d|]} \text{)} \\
&= \sigma_S^2 Tr(U \tilde{D} U^T U \tilde{D} U^T) \\
&= \sigma_S^2 Tr(U \tilde{D}^2 U^T) \\
&= \sigma_S^2 Tr(\tilde{D}^2) \\
&= \sigma_S^2 \sum_{i=1}^{d} \frac{(1 + \lambda\nu_i)^2}{(1 + \lambda + \lambda\nu_i)^2}
\end{aligned}
$$

$$
\begin{aligned}
B_2 &= Tr(V_S (I - W) \Theta (I - W)^T) \\
&= Tr(\Theta (I - W)^T V_S (I - W)) \\
&= Tr(\Theta V_S^{\frac{1}{2}} V_S^{-\frac{1}{2}} (I - W)^T V_S^{\frac{1}{2}} V_S^{\frac{1}{2}} (I - W) V_S^{\frac{1}{2}} V_S^{-\frac{1}{2}}) \\
&= Tr(V_S^{\frac{1}{2}} \Theta V_S^{\frac{1}{2}} \times (V_S^{\frac{1}{2}} (I - W) V_S^{-\frac{1}{2}})^T \times V_S^{\frac{1}{2}} (I - W) V_S^{-\frac{1}{2}}) \\
&= Tr(\tilde{\Theta} \times (I - \tilde{W})^T \times (I - \tilde{W})) \\
&= Tr(\tilde{\Theta} \times (U (I - \tilde{D}) U^T)^T \times U (I - \tilde{D}) U^T) \\
&= Tr(\tilde{\Theta} \times U (I - \tilde{D})^2 U^T) \\
&= Tr(U^T \tilde{\Theta} U \times (I - \tilde{D})^2)
\end{aligned}
$$

Or les valeurs propres de la matrice $(I - \tilde{D})^2$ sont les $((1 - \frac{1+\lambda\nu_i}{1+\lambda+\lambda\nu_i})^2)_{i \in [|1,d]}$ et on a même que $(I - \tilde{D})^2 = diag(\frac{\lambda^2}{(1+\lambda+\lambda\nu_i)^2})_{i \in [|1,d]}$

Pour la matrice $U^T \tilde{\Theta} U$, on peut la réécrire : $U = (u_1, ..., u_d)$ et $\tilde{\Theta} = (\tilde{\Theta}_1, ..., \tilde{\Theta}_d)$

$$
\begin{aligned}
U^T\tilde{\Theta}U &= \begin{bmatrix} u_1^T \\ . \\ . \\ u_d^T \end{bmatrix} \begin{bmatrix} \tilde{\Theta}_1 & . & . & \tilde{\Theta}_d \end{bmatrix} \begin{bmatrix} u_1 & . & . & u_d \end{bmatrix} \\
&= \begin{bmatrix} u_1^T\tilde{\Theta}_1 & . & . & u_1^T\tilde{\Theta}_d \\ . & & & . \\ . & & & . \\ u_d^T\tilde{\Theta}_1 & . & . & u_d^T\tilde{\Theta}_d \end{bmatrix} \begin{bmatrix} u_1 & . & . & u_d \end{bmatrix} \\
&= \begin{bmatrix} L_1 \\ . \\ . \\ L_d \end{bmatrix} \begin{bmatrix} u_1 & . & . & u_d \end{bmatrix} \\
&= \begin{bmatrix} L_1u_1 & . & . & L_1u_d \\ . & & & . \\ . & & & . \\ L_du_1 & . & . & L_du_d \end{bmatrix}
\end{aligned}
$$

On a alors que : $Tr(U^T\tilde{\Theta}U) = \sum_{k=1}^{d} L_k u_k$. Et comme $L_k = (u_k^T\tilde{\Theta}_1, ..., u_k^T\tilde{\Theta}_d) = u_k^T\tilde{\Theta}$, alors :

$$Tr(U^T\tilde{\Theta}U) = \sum_{k=1}^{d} u_k^T\tilde{\Theta}u_k$$

De ce fait, on peux conclure sur la valeur de B_2 :

$$B_2 = Tr(U^T\tilde{\Theta}U \times (I - \tilde{D})^2) = \lambda^2 \sum_{k=1}^{d} \frac{u_k^T\tilde{\Theta}u_k}{(1 + \lambda + \lambda\nu_k)^2}$$

D'où le résultat final :

$$\mathbb{E}[\|X_S(\hat{\beta} - \beta))\|^2] = \sigma_S^2 \sum_{i=1}^{d} \frac{(1 + \lambda\nu_i)^2}{(1 + \lambda + \lambda\nu_i)^2} + \sum_{i=1}^{d} \frac{\lambda^2\kappa_i^2}{(1 + \lambda + \lambda\nu_i)^2}$$

Avec $\forall i \in [|1, d|], \kappa_i^2 = u_i^T\tilde{\Theta}u_i$.

□

Références

[1] Haoyang Cao, Haotian Gu, Xin Guo, and Mathieu Rosenbaum. *Risk of Transfer Learning and its Applications in Finance.* SSRN, Novembre 2023.

[2] Aiyou Chen, Art B. Owen, and Minghui Shi. *Data enriched linear regression.* Electronic Journal of Statistics, 2015.

[3] Guillaume Richard, Antoine de Mathelin, Georges Hébrail, Mathilde Mougeot, and Nicolas Vayatis. *Unsupervised Multi-Source Domain Adaptation for Regression.* European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, September 2020.

[4] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. *A Comprehensive Survey on Transfer Learning.* IEEE, Janvier 2021.