Analyse en Composantes Principales Parcimonieuses (Sparse PCA)

Pablo Hueso, Sam Vallet

November 2024

Résumé

Dans ce document, nous revisitons l'article « Sparse Principal Component Analysis » de Hui Zou et al., en le situant dans le contexte des algorithmes de réduction de dimension dérivés du PCA classique. Nous fournissons des démonstrations détaillées de tous les théorèmes présentés dans l'article, tout en discutant leur motivation. Nous réimplémentons l'algorithme SPCA présenté dans l'article original et reproduisons certains des résultats. Le code des implémentations en R peut être trouvé sur https://github.com/PabloHueso/Sparse-PCA.

1 Introduction

1.1 Contenu du document

Tout d'abord, nous introduisons les notations et le vocabulaire qui seront utilisés tout au long du document. Si le lecteur n'est pas familiarisé avec la théorie de l'analyse en composantes principales ou la théorie de la décomposition en valeurs singulières, nous lui recommandons de lire B en premier lieu. Tout au long du corps, nous discutons de la motivation des théorèmes et situons les idées, en suivant un flux similaire à celui de l'article original, mais en fournissant des démonstrations avec une grande attention aux détails et en nous concentrant uniquement sur les idées principales de l'article, pour des raisons de concision. Après avoir présenté le critère SPCA 3 qui porte le nom de l'article, nous nous concentrerons sur la discussion de l'algorithme utilisé pour le minimiser et sur son application à certains ensembles de données, en reproduisant les résultats originaux.

1.2 Cadre et notations

Tout au long de ce document, nous considérons que nous disposons de n observations de p variables, qu'on note $X^{(1)}, ..., X^{(n)} \in \mathbb{R}^p$. On note avec X la matrice de design, i.e.

$$X = \begin{bmatrix} (X^{(1)})^T \\ \vdots \\ (X^{(n)})^T \end{bmatrix} \in \mathbb{R}^{n \times p}$$

Si A est une matrice, $A_{i,}$ et $A_{,j}$ désignent respectivement la i-ème ligne et la j-ème colonne de cette matrice. Si v est un vecteur, v_i désigne sa i-ème coordonnée (donc en particulier, si v_k est un vecteur, v_{ki} désigne sa i-ème coordonnée). Tout au long du document, r désigne le rang de la matrice X, et nous écrirons la décomposition en valeurs singulières de X sous les formes suivantes :

$$X = UDV^T = \sum_{j=1}^{r} \sigma(X)_j u_j v_j^T, \quad r = rang(X)$$

Ici, $U \in \mathbb{R}^{n \times n}$ et $V \in \mathbb{R}^{p \times p}$ sont matrices orthonormales, et $D \in \mathbb{R}^{n \times p}$ est est définie par $D_{i,i} = \sigma(X)_i$ si $i \leq r$ et égale à 0 sinon. Bien qu'il n'y ait que r valeurs singulières de X, pour simplifier les notations, nous écrirons $\sigma(X)_i = 0 \; \forall i > r$. Tout au long du document, la norme vectorielle utilisée est la norme euclidienne, sauf mention contraire. Dans le cas des normes matricielles, on utilisera la norme de Frobenius ; il s'agit de la norme induite par le produit scalaire de Frobenius, qui est défini par :

$$\langle A, B \rangle_F = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{i,j} B_{i,j}$$

où A et B sont des matrices de dimension $n \times m$. La norme de Frobenius vérifie :

$$\|A\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} A_{i,j}^2 = \mathrm{Tr}(A^T A)$$

Tout au long du document, nous supposons que nos variables sont centrées.

1.3 Notions clés sur l'ACP

Nous introduisons ci-après le vocabulaire de base dans le contexte de l'ACP.

1.3.1 Composantes principales et axes principaux

Dans l'ACP classique, nous appelons axes principaux les vecteurs qui engendrent le sous-espace vectoriel solution de 14, c'est-à-dire, à la famille orthonormale de vecteurs $\{v_1, \ldots, v_r\}$. Nous appelons composantes principales à la famille orthonormale de vecteurs $\{\sigma(X)_1 u_1, \ldots, \sigma(X)_r u_r\}$.
On a que $\sigma(X)_i u_i = X v_i$ pour tout $i = 1, \ldots, r$. En effet,

$$X v_i = (\sum_{j=1}^{r} \sigma(X)_j u_j v_j^T) v_i = \sigma(X)_i u_i$$

1.3.2 Loadings

Une idée clé de l'ACP, et l'une des motivations de l'article original, est que chacune des composantes principales est une combinaison linéaire de toutes les variables (ce qui, dans certains cas, complique l'interprétation des résultats) :

$$\sigma_k u_k = X v_k = \begin{pmatrix} \langle X^{(1)}, v_k \rangle \\ \vdots \\ \langle X^{(n)}, v_k \rangle \end{pmatrix} = \begin{pmatrix} X_1^{(1)} v_{k1} + \cdots + X_p^{(1)} v_{kp} \\ \vdots \\ X_1^{(n)} v_{k1} + \cdots + X_p^{(n)} v_{kp} \end{pmatrix} = \sum_{i=1}^{p} X_{,i} v_{ki}$$

Dans ce contexte, nous appellerons *loadings* les vecteurs v_k, c'est-à-dire les coefficients de la contribution de chaque variable à la composante principale. L'objectif principal de l'article est d'obtenir des *sparse loadings* afin de faciliter l'interprétation.

2 Motivation et détails de la SPCA

2.1 ACP et parcimonie

L'algorithme ACP constitue une pièce fondamentale dans les algorithmes de traitement de données et de réduction de dimension, ayant été appliqué avec succès à de nombreux problèmes (par exemple [4]). Cependant, comme nous l'avons mentionné dans l'introduction, l'ACP présente un inconvénient majeur : chacune des composantes principales est une combinaison linéaire des p variables, et les *loadings* sont généralement différents de zéro, ce qui complique souvent l'interprétation des résultats. Dans la littérature, nous pouvons trouver différentes approches pour résoudre ce problème, comme restreindre les *loadings* à ne prendre que les valeurs 1, 0 ou -1 [7], ou imposer la parcimonie à l'aide d'une technique de *thresholding*, en fixant artificiellement les *loadings* à 0 si leurs valeurs absolues sont inférieures à une certaine constante, une technique parfois utilisée en pratique, mais qui peut conduire à des résultats trompeurs [1].

Les mêmes problèmes d'interprétation surviennent dans la régression linéaire multiple, où la variable réponse est prédite par une combinaison linéaire des variables. Il existe une vaste littérature sur la sélection de variables dans le contexte de la régression. En particulier, le *LASSO* [6] produit des résultats précis tout en étant parcimonieux. Une extension du *LASSO* est l'*Elastic Net* [8], qui présente certains avantages. Dans la section suivante, nous verrons comment formuler l'ACP comme un problème de régression que l'on pénalisera pour assurer des coefficients parcimonieux comme conséquence directe.

2.2 ACP comme problème de régression

Nous allons maintenant montrer que l'ACP peut être reformulée exactement comme un problème de régression ridge, lequel on transformera en *elastic net* en introduisant la pénalité *LASSO*.

2.2.1 Approximation Parcimonieuse Directe

Tout d'abord, nous discuterons d'une approche plus simple, dans laquelle les composantes principales (PCs) sont supposées connues. Étant donné que chacune d'elles est une combinaison linéaire des p variables, les *loadings* peuvent être récupérés à l'aide d'une régression ridge (notez que $Z_i = XV_{,i}$) :

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \|Z_i - X\beta\|^2 + \lambda\|\beta\|^2 \tag{1}$$

Cette idée est formalisée par A.1.
Étant donné qu'après la normalisation les coefficients sont indépendants de λ, la pénalité ridge n'est pas introduite pour pénaliser les coefficients, mais pour garantir l'unicité de la solution, indépendamment de n et p. (Notez en particulier que, lorsque $p > n$, la régression aux moindres carrés ordinaires n'a pas de solution unique, ce qui pourrait empêcher de récupérer les vecteurs $V_{,i}$.)

Une pénalité L_1 est ensuite ajoutée à 1 pour le transformer en un problème d'*elastic net*, ce qui permet d'obtenir des approximations parcimonieuses des composantes principales.

2.2.2 ACP comme régression ridge

Notez que pour appliquer la méthode de la section précédente, il est nécessaire de connaître les composantes principales (PCs) a priori, ce qui n'en fait pas une alternative véritable à l'ACP. Par conséquent, dans cette section, une alternative qui ne nécessite pas la connaissance préalable des PCs est presentée.

Cette idée est formalisée par le théorème A.3, qui montre précisement que nous pouvons récupérer (exactement) l'ACP classique sous forme de régression ridge, en minimisant le critère suivant :

$$(\hat{A}, \hat{B}) \in \underset{(A,B):A^TA=I_{k\times k}}{\operatorname{argmin}} \quad \sum_{i=1}^{n} \|X^{(i)} - AB^TX^{(i)}\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 \tag{2}$$

Notez que l'approche par régression n'est en aucun cas nouvelle dans l'ACP : si, dans 2, nous supprimons la pénalité ridge et établissons $A = B$, nous obtenons un problème strictement équivalent à celui de 14.

2.2.3 Critère SPCA

Dans un esprit similaire à celui de la section 2.2.1, nous ajoutons des pénalités L_1 à 2, en obtenant le critère que nous appellerons désormais SPCA.

$$\underset{(A,B):A^TA=I_{k\times k}}{\operatorname{argmin}} \quad \sum_{i=1}^{n} \|X^{(i)} - AB^TX^{(i)}\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\beta_j\|_1 \tag{3}$$

Dans le reste du document, nous travaillerons avec ce critère, en proposant un algorithme pour le minimiser et en étudiant les résultats de celui-ci sur différents ensembles de données.

3 Solution Numérique

À continuation, nous présenterons l'algorithme proposé par l'article ainsi que certains résultats. L'objectif de l'algorithme est donc de minimiser le critère 3. L'algorithme fonctionne en deux étapes, on résout le critère pour A fixé puis pour B fixé.

Pour A fixé :

$$\hat{\beta}_j = \arg\min \|X\alpha_j - X\beta_j\|_2^2 + \lambda \|\beta_j\|_2^2 + \lambda_{1,j} \|\beta_j\|_1 \tag{4}$$

L'obtention de cette équation est complètement analogue à l'obtention de 8 dans le théorème A.3.

Pour B fixé : On peut ignorer les termes de pénalisation qui ne dépendent pas de A. On cherche donc à minimiser (en fonction de A) :

$$\|X - XBA^T\|_2^2, \quad \text{sous la condition } A^TA = I_{k\times k}.$$

Pour cela on utilise le théorème A.4 qui nous dit que la solution nous est donnée par

$$\hat{A} = UV^T, \quad \text{avec } (X^TX)B = UDV^T \tag{5}$$

3.1 Utilisation de la matrice de corrélation

Une remarque importante est que, pour résoudre 4 et 5, on a simplement besoin de calculer X^TX.

En effet, on peut réécrire :

$$\|X\alpha_j - X\beta_j\|_2^2 = \mathrm{Tr}\left((\alpha_j - \beta_j)^T X^T X(\alpha_j - \beta_j)\right)$$

Si l'on connaît la matrice de corrélation, on peut alors remplacer X^TX par Σ.

Pour se ramener ensuite à un problème d'*elastic net*, on réalise les transformations suivantes :

$$Y^{**} = \Sigma^{1/2}\alpha_j, \quad X^{**} = \Sigma^{1/2},$$

et on obtient finalement :

$$\hat{\beta}_j = \arg\min_{\beta} \|Y^{**} - X^{**}\beta\|_2^2 + \lambda\|\beta\|_2^2 + \lambda_{1,j}\|\beta\|_1.$$

3.2 Algorithme SPCA

Algorithm 1: SPCA (Sparse Principal Component Analysis)

Data: Matrice de données ou matrice de corrélation X, nombre de vecteurs propres k

Result: Coefficients B

Initialiser A avec les k premiers vecteurs propres de X^TX : $A = V[:, 1:k]$;

repeat

  for $j = 1, 2, \ldots, k$ **do**

    Résoudre l'équation suivante :

    $\beta_j = \arg\min \big((\alpha_j - \beta)^T X^T X(\alpha_j - \beta) + \lambda\|\beta\|_2^2 + \lambda_{1,j}\|\beta\|_1\big)$;

  end

  Calculer la SVD : $X^TXB = UDV^T$;

  Mettre à jour $A = UV^T$;

until *convergence*;

Normaliser : $V_j = \frac{\beta_j}{\|\beta_j\|_2}$, $\forall j = 1, 2, \ldots, k$;

Notez que l'algorithme converge nécessairement, car pour A fixé, le critère 3 est convexe en B, et vice-versa. Par conséquent, l'erreur estimée à l'étape $n+1$ est inférieure ou égale à celle de l'étape n, et puisque la fonction objectif est bornée inférieurement par 0, l'algorithme converge.

Une implémentation de cet algorithme qui est disponible sur notre GitHub

Remarque D'après l'article, lorsque le nombre d'individus est supérieur à celui des variables, le paramètre λ a peu d'impact sur les résultats, et il est courant de le fixer à zéro. Pour des paramètres comme λ_1, étant donné que l'algorithme converge rapidement dans ces conditions, il est pertinent de tester plusieurs valeurs de λ afin d'obtenir un bon compromis entre la variance expliquée et la sparsité.

Variance ajusté Dans l'algorithme proposé, les vecteurs B_j renvoyés n'ont pas la garantie d'être orthogonaux entre eux. Cela peut poser un problème lors du calcul de la variance. En effet, pour calculer la variance de la projection (et donc celle des composantes principales), on utilise généralement la formule suivante :

$$\mathrm{Tr}(V^TX^TXV) = \mathrm{Tr}(Z^TZ)$$

Cependant, dans notre cas, les vecteurs de V ne sont pas orthogonaux, ce qui peut entraîner une surestimation de la variance expliquée. Autrement dit, si le vecteur Z_j est colinéaire à d'autres vecteurs $Z_{j-1}, \ldots, Z_1$, la variance calculée $Z_j^TZ_j$ inclura à la fois la variance de Z_j et une part de la variance des composantes précédentes. Il faut donc ajuster ce calcul en soustrayant la variance apportée par ces composantes afin d'obtenir une estimation plus précise de la

variance expliquée par Z_j. Pour ce calcul, l'article propose d'utiliser la décomposition QR de Z, où Q est une matrice orthonormée et R est une matrice triangulaire supérieure. On a alors :

$$\|Z\|^2 = \|QR\|^2 = \mathrm{Tr}((QR)^T QR) = \mathrm{Tr}(R^T R) = \|R\|^2$$

Pour calculer la variance totale ajustée, on utilise donc $\mathrm{Tr}(R^T R)$.

Complexité Une observation est que dans le cas où la matrice de données contient plus de lignes que de colonnes, donc on étudie un échantillon avec plus d'individus que de variables la complexité totale est d'ordre $np^2 + \mathcal{O}(p^3)$, ce qui est donc assez léger a computer. Cependant dans le cas ou on a beaucoup plus de variables que de lignes, cela devient beaucoup plus lourd, notamment la complexité de la résolution de l'*elastic net* est de $\mathcal{O}(pnJ + J^3)$ (où J est le nombre de coefficients différents de zéro). Dans ce cas on cherche un autre moyen pour calculer les coefficients de la matrice B.

3.3 SPCA pour $p \gg n$

Pour résoudre le problème lié à la complexité lorsque le nombre de variables p est beaucoup plus grand que le nombre de lignes n, on utilise A.5, qui indique que lorsque $\lambda \to +\infty$, résoudre notre problème revient à minimiser la formule suivante :

$$\hat{\beta}_j = \arg\min_{\beta_j} \left(-2\alpha_j^T (X^T X)\beta_j + \|\beta_j\|_2^2 + \lambda_{1,j}\|\beta_j\|_1\right),$$

La solution pour β_j est donnée par :

$$\beta_j = \max\left(|\alpha_j^T X^T X| - \frac{\lambda_{1,j}}{2}, 0\right) \cdot \mathrm{sign}(\alpha_j^T X^T X).$$

On adapte donc notre algorithme en modifiant le step2 pour le calcul de β_j.

3.4 Résultats

Nous discuterons ici des résultats de l'article et de ceux que nous avons obtenus. On réalisera des études sur deux datasets, un avec plus d'individus que de variables et inversement pour le second.

N $\gg$ P On commence par étudier le cas où on manipule des données avec plus d'individus que de variables, pour cela on utilise le dataset pitprops qui possède 180 individus pour 13 variables, on réalise notre étude sur les six premiers axes.

TABLE 1 – Résultat obtenue avec la version de l'algorithme que nous avons reproduit

Variable	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**
topdiam	-0.56					
length	-0.49					
moist		-0.66				
testsg		-0.75				
ovensg			-0.57			
ringtop			-0.70			
ringbut	-0.144		-0.42			
bowmax	-0.28					
bowdist	-0.40					
whorls						
clear				-1		
knots					-1	
diaknot						-1
variance ajustée	27	13	12	7	6	6

On obtient donc bien des vecteurs sparse avec une variance correcte.

TABLE 2 – Comparaison des algorithmes PCA, SPCA et Scotlass

	PC1	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**
Nombre de non zero SPCA	7	**4**	**4**	**1**	**1**	**1**
Nombre de non zero Scotlass	**6**	6	6	6	10	13
Variance PCA cumulée	32.4	50.7	65.1	73.6	80.6	86.9
Variance ajustée cumulée SPCA	28.0	42.0	55.3	62.7	69.5	75.8
Variance ajustée cumulée Scotlass	19.6	33.4	45.8	53.8	60.9	69.3

Par rapport a la PCA les axes crée par la SPCA perdent un peu en variance,

cela reste cependant plus efficace que Scotlass, l'algorithme SPCA réduit également plus d'éléments a 0 que Scotlass, on peut donc en déduire qu'il est plus efficace. En comparaison avec notre version de l'algorithme, notre variance est plus faible mais nos vecteurs possèdent moins de valeurs non nulles.

P $\gg$ **N** Nous allons maintenant nous placer dans le cas où les données comportent plus de variables que d'individus.

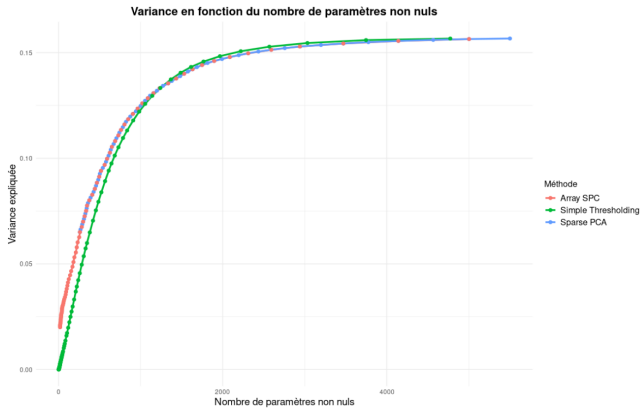

FIGURE 1 – Variance en fonction du nombre de paramètres non nuls(data Breast)

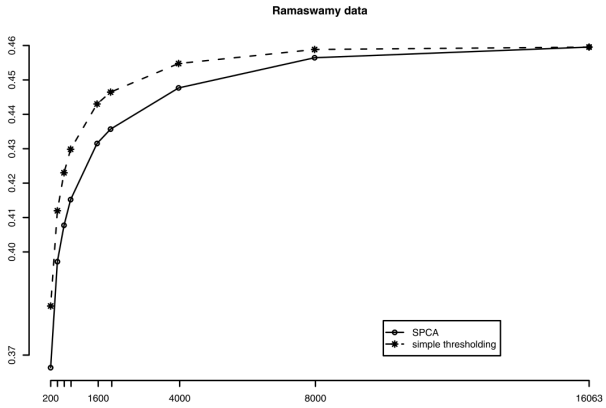

FIGURE 2 – Variance en fonction du nombre de paramètres non nuls(data RAMASWAMY)

Pour la première figure on utilise le dataset `Breast cancer gene expression GSE45827` [3], nous comparons l'algorithme que nous avons implémenté (Sparse

PCA), celui de l'article (Array PCA), et un seuil classique (thresholding). La courbe de notre Sparse PCA se superpose à celle de l'Array PCA, ce qui indique que notre algorithme reproduit bien les résultats de l'algorithme initial. On remarque que la courbe du simple thresholding passe au-dessus de celle de l'Array PCA à partir d'un certain nombre d'éléments non nuls, mais les résultats restent proches.

La figure 2 est tirée de l'article. Elle compare l'algorithme SPCA avec le simple thresholding sur le jeu de données RAMASWAMY, qui comporte 16 000 variables pour 141 individus. Les résultats observés par rapport à notre simulation sont légèrement différents. Dans leur cas, le simple thresholding affiche des résultats plus performants. La différence peut s'expliquer par la différence du datasets utilisé ou par l'implémentation du thresholding, qui peut être différente.

A Démonstrations des théorèmes de l'article

Théorème A.1 (Approche naïf par régression pour la PCA)**.** *Pour chaque i, on dénote par Z_i la i-ème composante principale. On considère un $\lambda \geq 0$ et l'estimateur ridge $\hat{\beta}_{ridge}$ donné par*

$$\hat{\beta}_{ridge} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \|Z_i - X\beta\|^2 + \lambda \|\beta\|^2 \right\},$$

Soit $\hat{v} = \frac{\hat{\beta}_{ridge}}{\|\hat{\beta}_{ridge}\|_2}$. Alors on a $\hat{v} = V_{,i}$

Démonstration. Au cours de la démonstration, nous utiliserons $X^T X = V D^T D V^T$, $V^T V = I_{p\times p}$ et $U^T U = I_{n\times n}$. La première égalité s'obtient en remplaçant X par son écriture matricielle selon la décomposition SVD :

$$X^T X = (UDV^T)^T UDV^T = (DV^T)^T U^T UDV^T = (DV^T)^T DV^T = VD^T DV^T$$

De même, nous utiliserons que l'estimateur ridge s'écrit sous la forme :

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T Z_i$$

La dérivation de l'estimateur ridge est élémentaire [5], mais nous ne l'écrirons pas ici. On a donc :

$$\begin{aligned}
\hat{\beta}_{\text{ridge}} &= (X^T X + \lambda I)^{-1} X^T Z_i \\
&= (X^T X + \lambda I)^{-1} X^T X V_{,i} \\
&= (X^T X + \lambda I)^{-1} V D^T D V^T V_{,i} \\
&= (X^T X + \lambda I)^{-1} V D^T D e_i \\
&= (X^T X + \lambda I)^{-1} V \sigma(X)_i^2 e_i \\
&= (V D^T D V^T + \lambda V^T V)^{-1} V \sigma(X)_i^2 e_i \\
&= (V(D^T D + \lambda I)V^T)^{-1} V \sigma(X)_i^2 e_i \\
&= V(D^T D + \lambda I)^{-1} V^T V \sigma(X)_i^2 e_i \\
&= V(D^T D + \lambda I)^{-1} \sigma(X)_i^2 e_i \\
&= V(\text{diag}(\lambda + \sigma(X)_1^2, \lambda + \sigma(X)_2^2, \ldots, \lambda + \sigma(X)_p^2))^{-1} \sigma(X)_i^2 e_i \\
&= V(\text{diag}(\frac{1}{\lambda + \sigma(X)_1^2}, \frac{1}{\lambda + \sigma(X)_2^2}, \ldots, \frac{1}{\lambda + \sigma(X)_p^2})) \sigma(X)_i^2 e_i \\
&= V \frac{\sigma(X)_i^2}{\lambda + \sigma(X)_i^2} e_i \\
&= V_{,i} \frac{\sigma(X)_i^2}{\lambda + \sigma(X)_i^2}
\end{aligned}$$

On rappelle que les $V_{,i}$ sont vecteurs orthonormaux i.e. $\|V_{,i}\|_2 = 1$, donc $\hat{v} = \frac{\hat{\beta}_{\text{ridge}}}{\|\hat{\beta}_{\text{ridge}}\|} = V_{,i}$

□

Lemme A.2. *Considérons le critère de régression ridge :*

$$C_\lambda(\beta) = \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2,$$

Si $\hat{\beta} = \arg\min_\beta C_\lambda(\beta)$*, alors :*

$$C_\lambda(\hat{\beta}) = y^T(I - S_\lambda)y,$$

où S_λ *est l'opérateur ridge :*

$$S_\lambda = X(X^TX + \lambda I)^{-1}X^T.$$

Démonstration. On sait que $\hat{\beta}_{\text{ridge}} = (X^TX + \lambda I)^{-1}X^Ty$ est l'argmin de $C_\lambda(\beta)$. Par conséquent, $\hat{\beta}_{\text{ridge}}$ vérifie en particulier :

$$\nabla_\beta(C_\lambda(\hat{\beta}_{\text{ridge}})) = -2X^T(y - X\hat{\beta}_{\text{ridge}}) + 2\lambda\hat{\beta}_{\text{ridge}} = 0,$$

ou, de manière équivalente :

$$\begin{aligned}
& 2\lambda\hat{\beta}_{\text{ridge}} = 2X^T(y - X\hat{\beta}_{\text{ridge}}) \\
\iff \quad & \lambda\hat{\beta}_{\text{ridge}}^T\hat{\beta}_{\text{ridge}} = \hat{\beta}_{\text{ridge}}^TX^T(y - X\hat{\beta}_{\text{ridge}}) \\
\iff \quad & \lambda\|\hat{\beta}_{\text{ridge}}\|_2^2 = (X\hat{\beta}_{\text{ridge}})^T(y - X\hat{\beta}_{\text{ridge}}) \\
\iff \quad & \lambda\|\hat{\beta}_{\text{ridge}}\|_2^2 = (y - X\hat{\beta}_{\text{ridge}})^T(X\hat{\beta}_{\text{ridge}}).
\end{aligned}$$

D'autre part, on peut écrire l'autre terme de $C_\lambda(\hat{\beta}_{\text{ridge}})$ sous la forme :

$$\begin{aligned}
\|y - X\hat{\beta}_{\text{ridge}}\|_2^2 &= (y - X\hat{\beta}_{\text{ridge}})^T(y - X\hat{\beta}_{\text{ridge}}) \\
&= (y - X\hat{\beta}_{\text{ridge}})^Ty - (y - X\hat{\beta}_{\text{ridge}})^TX\hat{\beta}_{\text{ridge}}.
\end{aligned}$$

En sommant les deux expressions et en substituant $X\hat{\beta}_{\text{ridge}} = S_\lambda y$, on obtient :

$$\begin{aligned}
C_\lambda(\hat{\beta}_{\text{ridge}}) &= (y - X\hat{\beta}_{\text{ridge}})^Ty \\
&= (y - S_\lambda y)^Ty \\
&= ((I - S_\lambda)y)^Ty \\
&= y^T(I - S_\lambda)^Ty \\
&= y^T(I - S_\lambda)y.
\end{aligned}$$

Pour la dernière étape, notez que $(I - S_\lambda)$ est symétrique, car S_λ l'est. $\square$

Le théorème 2 de l'article est un cas particulier du théorème 3 pour $k = 1$. Par conséquent, il suffit de démontrer le théorème 3.

Théorème A.3 (Approche par régression pour la PCA)**.** *Soient $A, B \in \mathbb{R}^{p\times k}$ définies par $A = [\alpha_1 \dots \alpha_k]$ et $B = [\beta_1 \dots \beta_k]$. Pour tout $\lambda > 0$, soit*

$$(\hat{A}, \hat{B}) \in \underset{(A,B):A^TA=I_{k\times k}}{\operatorname{argmin}} \sum_{i=1}^{n} \|X^{(i)} - AB^TX^{(i)}\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2$$

Alors, $\hat{\beta}_j \propto V_{,j}$ pour $j = 1, \dots, k$.

Démonstration. D'abord, on pose :

$$C_\lambda(A, B) = \sum_{i=1}^{n} \|X^{(i)} - AB^TX^{(i)}\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2$$

Ensuite, on remarque que :

$$\sum_{i=1}^{n} \|X^{(i)} - AB^TX^{(i)}\|^2 = \|X - XBA^T\|_F^2$$

où $\|\cdot\|_F$ est la norme de Frobenius. En effet, on a :

$$\begin{aligned}
\sum_{i=1}^{n} \|X^{(i)} - AB^TX^{(i)}\|^2 &= \sum_{i=1}^{n}\sum_{j=1}^{p} (X_j^{(i)} - \langle (AB^T)_{j,}, X^{(i)}\rangle)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p} (X_j^{(i)} - \langle (AB^T)_{,j}^T, X^{(i)}\rangle)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p} (X_j^{(i)} - (XBA^T)_{i,j})^2 \\
&= \|X - XBA^T\|_F^2
\end{aligned}$$

A est orthonormée, i.e. ses colonnes $\alpha_1, \dots, \alpha_k$ forment une famille orthonormée de vecteurs de $\mathbb{R}^p$. On considère une complétion de cette famille en une base orthonormée $\{\alpha_1, \dots, \alpha_p\}$ de $\mathbb{R}^p$. On définit $A_\perp = [\alpha_{k+1} \dots \alpha_p]$. Par construction, nous avons que $[A : A_\perp]$ est $p \times p$ orthonormée, et on a $A^TA = I_{k\times k}$, $A_\perp^TA = I_{(p-k)\times(p-k)}$ et $A^TA_\perp = 0$. Maintenant, montrons que

$$\|X - XBA^T\|_F^2 = \|XA_\perp\|_F^2 + \|XA - XB\|_F^2 \tag{6}$$

Nous verrons qu'il s'agit d'une application du théorème de Pythagore pour la norme de Frobenius. D'abord, remarquons que

$$I_{p\times p} = [A; A_\perp][A; A_\perp]^T = AA^T + A_\perp A_\perp^T \tag{7}$$

Donc on a :

$$\begin{aligned} X - XBA^T &= X(I_p - BA^T) \\ &= X(AA^T + A_\perp A_\perp^T - BA^T) \\ &= X(A_\perp A_\perp^T + (A-B)A^T) \\ &= XA_\perp A_\perp^T + X(A-B)A^T \end{aligned}$$

Nous vérifions que cette décomposition est orthogonale pour le produit scalaire de Frobenius, en effet, $\langle XA_\perp A_\perp^T\,, X(A-B)A^T\rangle_F = \langle XA_\perp\,, X(A-B)A^TA_\perp\rangle_F = 0$, car $A^TA_\perp = 0$. En conséquence, par le théorème de Pythagore, nous avons :

$$\|X - XBA^T\|_F^2 = \|XA_\perp A_\perp^T\|_F^2 + \|X(A-B)A^T\|_F^2$$

Maintenant, la seule chose qui reste à prouver pour 6 est de montrer $\|XA_\perp\|_F^2 = \|XA_\perp A_\perp^T\|_F^2$ et $\|XA - XB\|_F^2 = \|(XA-XB)A^T\|_F^2$. On a :

$$\begin{aligned} \left\|(XA-XB)A^T\right\|_F^2 &= \mathrm{Tr}\left(\left((XA-XB)A^T\right)^T\left((XA-XB)A^T\right)\right) \\ &= \mathrm{Tr}\left(A(XA-XB)^T(XA-XB)A^T\right) \\ &= \mathrm{Tr}\left((XA-XB)^T(XA-XB)(A^TA)\right) \\ &= \mathrm{Tr}\left((XA-XB)^T(XA-XB)\right) \quad (\text{car } A^TA = I_k) \\ &= \|XA-XB\|_F^2 \end{aligned}$$

et aussi :

$$\begin{aligned} \left\|XA_\perp A_\perp^T\right\|_F^2 &= \mathrm{Tr}\left(\left(XA_\perp A_\perp^T\right)^T\left(XA_\perp A_\perp^T\right)\right) \\ &= \mathrm{Tr}\left(A_\perp A_\perp^T X^T X A_\perp A_\perp^T\right) \\ &= \mathrm{Tr}\left(A_\perp^T X^T X A_\perp A_\perp^T A_\perp\right) \\ &= \mathrm{Tr}\left(A_\perp^T X^T X A_\perp\right) \quad (\text{car } A_\perp^T A_\perp = I_{p-k}) \\ &= \mathrm{Tr}\left((XA_\perp)^T X A_\perp\right) \\ &= \|XA_\perp\|_F^2 \end{aligned}$$

Donc on a bien 6.
Maintenant, notons que

$$\|XA - XB\|_F^2 = \sum_{j=1}^{k} \|X\alpha_j - X\beta_j\|^2$$

En effet,

$$\begin{aligned}
\|XA - XB\|_F^2 &= \sum_{i=1}^{n}\sum_{j=1}^{k}(\langle X^{(i)}, A_{,j}\rangle - \langle X^{(i)}, B_{,j}\rangle)^2 \\
&= \sum_{j=1}^{k}\sum_{i=1}^{n}(\langle X^{(i)}, A_{,j}\rangle - \langle X^{(i)}, B_{,j}\rangle)^2 \\
&= \sum_{j=1}^{k}\|X\alpha_j - X\beta_j\|^2 \quad \text{(car ici } A_{,j} = \alpha_j \text{ et } B_{,j} = \beta_j)
\end{aligned}$$

Par conséquent, si nous fixons A, minimiser $C_\lambda(A, B)$ est est équivalent à minimiser

$$\underset{B}{\operatorname{argmin}} \sum_{j=1}^{k}\{\|X\alpha_j - X\beta_j\|^2 + \lambda\|\beta_j\|^2\} \tag{8}$$

Ce qui est équivalent à résoudre k régressions ridge indépendantes. Notez en particulier que, avec $A = V$, nous avons que $XA_{,j} = Z_j$ (la j-ième composante principale) et, par le théorème A.1, β_j sera proportionnel à $V_{,j}$.
La solution à 8 est donnée par $\hat{B} = [\hat{\beta}_1 \dots \hat{\beta}_k]$, avec $\hat{\beta}_i = (X^TX + \lambda I)^{-1}X^TX\alpha_i$ pour tout $i = 1, \dots, k$ (chaque colonne est un estimateur ridge). En notant $T = (X^TX + \lambda I)^{-1}X^TX$ on a $\hat{\beta}_i = T\alpha_i$ pour tout $i = 1, \dots, k$. Donc :

$$\hat{B} = [T\alpha_1, \dots, T\alpha_k] = TA = (X^TX + \lambda I)^{-1}X^TXA \tag{9}$$

Maintenant, on injecte $\hat{B}$ dans $C_\lambda(A, B)$. On rapelle la définition de l'opérateur ridge : $S_\lambda = X(X^TX + \lambda I)^{-1}X^T$. On a :

$$\begin{aligned}
C_\lambda(A, \hat{B}) &= \|XA_\perp\|_F^2 + \sum_{j=1}^{k}\{\|X\alpha_j - X\beta_j\|^2 + \lambda\|\beta_j\|^2\} \\
&= \|XA_\perp\|_F^2 + \sum_{j=1}^{k}(X\alpha_j)^T(I - S_\lambda)X\alpha_j \quad \text{(D'après le lemme } A.2) \\
&= \|XA_\perp\|_F^2 + \sum_{j=1}^{k}((XA)^T(I - S_\lambda)XA)_{j,j} \\
&= \|XA_\perp\|_F^2 + \operatorname{Tr}((XA)^T(I - S_\lambda)XA)) \\
&= \operatorname{Tr}((XA_\perp)^TXA_\perp) + \operatorname{Tr}((XA)^TXA) - \operatorname{Tr}((XA)^TS_\lambda XA) \\
&= \operatorname{Tr}((XA_\perp)^TXA_\perp) + \operatorname{Tr}(A^TX^TXA) - \operatorname{Tr}(A^TX^TS_\lambda XA) \\
&= \operatorname{Tr}(A_\perp^TX^TXA_\perp) + \operatorname{Tr}(A^TX^TXA) - \operatorname{Tr}(A^TX^TS_\lambda XA) \\
&= \operatorname{Tr}(X^TX) - \operatorname{Tr}(A^TX^TS_\lambda XA)
\end{aligned}$$

Où, pour la dernière égalité, nous avons utilisé que $Tr(X^TX) = \text{Tr}(A_\perp^T X^T X A_\perp) + \text{Tr}(A^T X^T X A)$, ce qui est vrai d'après 7.

Ensuite, nous cherchons à minimiser la dernière expression par rapport à A sous la contrainte $A^TA = I$, ce qui est strictement équivalent à maximiser $Tr(A^TX^TS_\lambda XA)$ (sous la même contrainte). Transformons cette expression de la trace :

$$\begin{aligned} Tr(A^TX^TS_\lambda XA) &= \sum_{i=1}^{k} (A^TX^TS_\lambda XA)_{i,i} \\ &= \sum_{i=1}^{k} \langle \alpha_i, X^TS_\lambda X\alpha_i \rangle \end{aligned}$$

Nous remarquons que la matrice $X^TS_\lambda X$ est symétrique, donc ses vecteurs propres sont orthogonaux. Par conséquent, l'expression ci-dessus est maximisée quand on prend $\alpha_i = w_i$ pour tout $i = 1, \dots k$, où w_i est le vecteur unitaire associé à la i-ème plus grande valeur propre de cette matrice. Par ailleurs, en remplaçant X par sa décomposition SVD dans $X^TS_\lambda X$, nous obtenons :

$$\begin{aligned} X^TS_\lambda X &= X^TX(X^TX + \lambda I)^{-1}X^TX \\ &= VD^TDV^T(VD^TDV^T + \lambda I)^{-1}VD^TDV^T \\ &= VD^TDV^T(V(D^TD + \lambda I)V^T)^{-1}VD^TDV^T \\ &= VD^TDV^T(V(D^TD + \lambda I)^{-1}V^T)VD^TDV^T \\ &= VD^TD(D^TD + \lambda I)^{-1}D^TDV^T \end{aligned}$$

Notez que $D^TD(D^TD + \lambda I)^{-1}D^TD$ est diagonale, donc $\hat{A} = [V_{,1}, \dots, V_{,k}]$ (les $V_{,1}, \dots, V_{,k}$ sont les vecteurs propres unitaires associées aux k plus grandes valeurs propres de $X^TS_\lambda X$). Finalement, en injectant la SVD de X et $\hat{A}$ dans 9, on remarque que chacun des $\hat{\beta}_i$ est proportionnel à $V_{,i}$. En effet,

$$\begin{aligned} \hat{\beta}_i &= (X^TX + \lambda I)^{-1}X^TX\hat{\alpha}_i \\ &= (VD^TDV^T + \lambda I)^{-1}VD^TDV^T\hat{\alpha}_i \\ &= V(D^TD + \lambda I)^{-1}V^TVD^TDV^T\hat{\alpha}_i \\ &= V(D^TD + \lambda I)^{-1}D^TDV^T\hat{\alpha}_i \\ &= V(D^TD + \lambda I)^{-1}D^TDe_i \\ &= \gamma_i Ve_i \\ &= \gamma_i V_{,i} \end{aligned}$$

Où γ_i est le i-ème élément de la diagonale de la matrice $(D^TD + \lambda I)^{-1}D^TD$, qui est diagonale. Le théorème est démontré. □

Théorème A.4 (Rotation de Procrustes à rang réduit)**.** *Soient $M_{n\times p}$ et $N_{n\times k}$ deux matrices réelles. On considère le problème de minimisation suivant :*

$$\hat{A} \in \underset{A:A^TA=I_{k\times k}}{\operatorname{argmin}} \quad \|M - NA^T\|_F^2 \tag{10}$$

Supposons que la décomposition en valeurs singulières (SVD) de M^TN soit $M^TN = UDV^T$. Alors, $\hat{A} = UV^T$.

Démonstration. On commence par développer la norme matricielle :

$$\begin{aligned}\|M - NA^T\|^2 &= \mathrm{Tr}((M - NA^T)^T(M - NA^T)) \\ &= \mathrm{Tr}(M^TM) - 2\mathrm{Tr}(M^TNA^T) + \mathrm{Tr}(AN^TNA^T)\end{aligned}$$

Comme $A^TA = I_{k\times k}$, on peut simplifier le terme $\mathrm{Tr}(AN^TNA^T)$:

$$\mathrm{Tr}(AN^TNA^T) = \mathrm{Tr}(N^TA^TAN) = \mathrm{Tr}(N^TN).$$

L'objectif devient alors de maximiser le terme $\mathrm{Tr}(M^TNA^T)$, car c'est le seul qui dépend de A.

Avec la SVD $M^TN = UDV^T$, on peut réécrire $\mathrm{Tr}(M^TNA^T)$ sous la forme :

$$\mathrm{Tr}(M^TNA^T) = \mathrm{Tr}(UDV^TA^T).$$

Posons $A^* = AV$. On obtient alors :

$$\mathrm{Tr}(UDV^TA^T) = \mathrm{Tr}(UDA^{*T}) = \mathrm{Tr}(A^{*T}UD).$$

Comme D est une matrice diagonale dont les éléments sont positifs ou nuls, le terme $\mathrm{Tr}(A^{*T}UD)$ est maximisé lorsque la diagonale de $A^{*T}U$ est positive et maximale.

Notez que nous pouvons appliquer l'inégalité de Cauchy-Schwarz, puisque le produit scalaire de Frobenius peut s'écrire comme $\mathrm{Tr}(A^TB)$. On a donc :

$$\mathrm{Tr}(A^{*T}U) \leq \sqrt{\mathrm{Tr}(A^{*T}A^*)}\sqrt{\mathrm{Tr}(U^TU)}$$

Donc $\mathrm{Tr}(A^{*T}U)$ est maximal lorsque $A^{*T}U = I$, ce qui revient à dire que $A^* = U$. Par conséquent, $\hat{A} = UV^T$. □

Théorème A.5. *Soit* $(\hat{A}, \hat{B})$ *solution de* (3)*, avec* $\hat{B} = [\hat{\beta}_1, \dots, \hat{\beta}_k]$*. On pose* $\hat{V}_j(\lambda) = \frac{\hat{\beta}_j}{\|\hat{\beta}_j\|}$ *pour tout* $j = 1, \dots, k$*. Soit* $(\hat{\mathfrak{A}}, \hat{\mathfrak{B}})$ *solution de :*

$$(\hat{\mathfrak{A}}, \hat{\mathfrak{B}}) \in \underset{(A,B):A^TA=I_{k\times k}}{\operatorname{argmin}} \; -2\,Tr(A^TX^TXB) + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\beta_j\|_1, \tag{11}$$

avec $\hat{\mathfrak{B}} = [\hat{\mathfrak{b}}_1, \dots, \hat{\mathfrak{b}}_k]$*. Alors, pour tout* $j = 1, \dots, k$ *on a :*

$$\hat{V}_j(\lambda) \xrightarrow{\lambda\to\infty} \frac{\hat{\mathfrak{b}}_j}{\|\hat{\mathfrak{b}}_j\|}.$$

Démonstration. On définit $\hat{B}^* = [\hat{\beta}_1^*, \dots, \hat{\beta}_k^*]$ avec $\hat{\beta}_i^* = (1+\lambda)\hat{\beta}_i$ pour tout $i = 1, \dots, k$. Si on définit le critère

$$C_{\lambda,\lambda_1}(A,B) = \sum_{i=1}^{n} \|X^{(i)} - A\frac{B^T}{1+\lambda}X^{(i)}\|^2 + \lambda \sum_{j=1}^{k} \|\frac{\beta_j}{1+\lambda}\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\frac{\beta_j}{1+\lambda}\|_1 \tag{12}$$

Alors $(\hat{A}, \hat{B}^*)$ appartient par définition à $\operatorname{argmin}_{(A,B):A^TA=I_{k\times k}} C_{\lambda,\lambda_1}(A,B)$.

Maintenant, on remarque que $\sum_{j=1}^{k} \|\frac{\beta_j}{1+\lambda}\|^2 = \frac{1}{(1+\lambda)^2}\mathrm{Tr}(B^TB)$. En effet,

$$\begin{aligned}
\sum_{j=1}^{k} \|\frac{\beta_j}{1+\lambda}\|^2 &= \frac{1}{(1+\lambda)^2} \sum_{j=1}^{k} \|\beta_j\|^2 \\
&= \frac{1}{(1+\lambda)^2}\|B\|_F^2 \\
&= \frac{1}{(1+\lambda)^2}\mathrm{Tr}(B^TB)
\end{aligned}$$

Nous cherchons également une expression alternative pour $\sum_{i=1}^{n} \|X^{(i)} - A\frac{B^T}{1+\lambda}X^{(i)}\|^2$:

$$\begin{aligned}
\sum_{i=1}^{n} \|X^{(i)} - A\frac{B^T}{1+\lambda}X^{(i)}\|^2 &= \sum_{i=1}^{n}\sum_{j=1}^{p}(X_j^{(i)} - \frac{1}{1+\lambda}\langle (AB^T)_{j,}, X^{(i)}\rangle)^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p}(X_j^{(i)} - \frac{1}{1+\lambda}\langle (BA^T)_{,j}, X^{(i)}\rangle)^2 \\
&= \|X - X\frac{B}{1+\lambda}A^T\|_F^2 \\
&= \mathrm{Tr}((X - X\frac{B}{1+\lambda}A^T)^T(X - X\frac{B}{1+\lambda}A^T)) \\
&= \mathrm{Tr}((X^T - A\frac{B^T}{1+\lambda}X^T)(X - X\frac{B}{1+\lambda}A)) \\
&= \mathrm{Tr}(X^TX - X^TX\frac{B}{1+\lambda}A^T - A\frac{B^T}{1+\lambda}X^TX + A\frac{B^T}{1+\lambda}X^TX\frac{B}{1+\lambda}A^T) \\
&= \mathrm{Tr}(X^TX) + \frac{1}{(1+\lambda)^2}\mathrm{Tr}(AB^TX^TXBA^T) \\
&\quad - \mathrm{Tr}(X^TX\frac{B}{1+\lambda}A^T) - \mathrm{Tr}(A\frac{B^T}{1+\lambda}X^TX) \\
&= \mathrm{Tr}(X^TX) + \frac{1}{(1+\lambda)^2}\mathrm{Tr}(B^TX^TXB) - \frac{1}{(1+\lambda)^2}\mathrm{Tr}(A^TX^TXB)
\end{aligned}$$

Maintenant, si on remplace ces deux expressions dans 12, on obtient :

$$C_{\lambda,\lambda_1}(A,B) = \mathrm{Tr}(X^TX) + \frac{1}{1+\lambda}((\mathrm{Tr}(B^T\frac{X^TX+\lambda I}{1+\lambda}B) - 2\mathrm{Tr}(A^TX^TXB) + \lambda_{1,j}\|\frac{\beta_j}{1+\lambda}\|_1)$$

Ce qui implique que

$$(\hat{A},\hat{B}^*) \in \underset{(A,B):A^TA=I_{k\times k}}{\mathrm{argmin}} (\mathrm{Tr}(B^T\frac{X^TX+\lambda I}{1+\lambda}B) - 2\mathrm{Tr}(A^TX^TXB) + \lambda_{1,j}\|\frac{\beta_j}{1+\lambda}\|_1 \tag{13}$$

Finalement, on se rend compte que $\mathrm{Tr}(B^T\frac{X^TX+\lambda I}{1+\lambda}B) \xrightarrow{\lambda\to\infty} \mathrm{Tr}(B^TB) = \sum_{j=1}^{k}\|\beta_j\|^2$.
Ainsi, 13 se rapproche de 11, et la conclusion du théorème s'ensuit. □

Lemme A.6. *La matrice B, solution du problème d'optimisation suivant :*

$$B = \underset{B}{\operatorname{argmin}} \Big(-2\, Tr\big(A^T X^T X B\big) + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\beta_j\|_1 \Big),$$

est donnée par :

$$\beta_j = \max\left(\big|\alpha_j^T X^T X\big| - \frac{\lambda_{1,j}}{2},\ 0 \right) \operatorname{sign}\big(\alpha_j^T X^T X\big), \quad \forall j \in \{1, \ldots, k\}.$$

où β_j représente la j-ème colonne de B et α_j la j-ème colonne de A.

Démonstration. Premier cas : Si $|\alpha_j^T X^T X| \leq \frac{\lambda_{1,j}}{2}$, alors :

$$-2\alpha_j^T (X^T X)\beta_j + \lambda_{1,j} \|\beta_j\|_1 \geq 0.$$

Ainsi, la valeur qui minimise est $\beta_j = 0$.

Si $|\alpha_j^T X^T X| > \frac{\lambda_{1,j}}{2}$, alors on dérive par rapport à β_j et on cherche la valeur qui annule la dérivée.

La dérivée s'écrit :

$$\frac{\partial}{\partial \beta_j} = -2\alpha_j^T (X^T X) + 2\beta_j + \lambda_{1,j} \operatorname{sign}(\beta_j)$$

Cette dérivée s'annule pour :

$$\beta_j = \left(|\alpha_j^T X^T X| - \frac{\lambda_{1,j}}{2} \right) \operatorname{sign}(\alpha_j^T X^T X)$$

En effet, on a $|\alpha_j^T X^T X| - \frac{\lambda_{1,j}}{2} > 0$, ce qui implique que :

$$\operatorname{sign}(\alpha_j^T X^T X) = \operatorname{sign}(\beta_j)$$

Vérifions l'annulation de la dérivée :

$$-2\alpha_j^T (X^T X) + 2 \left(|\alpha_j^T X^T X| - \frac{\lambda_{1,j}}{2} \right) \operatorname{sign}(\alpha_j^T X^T X) + \lambda_{1,j} \operatorname{sign}(\beta_j)$$

En développant, nous avons :

$$-2\alpha_j^T (X^T X) + 2\alpha_j^T (X^T X) - \lambda_{1,j} \operatorname{sign}(\alpha_j^T X^T X) + \lambda_{1,j} \operatorname{sign}(\beta_j) = 0$$

Ainsi, la solution pour β_j est donnée par :

$$\beta_j = \max\left(|\alpha_j^T X^T X| - \frac{\lambda_{1,j}}{2}, 0 \right) \operatorname{sign}(\alpha_j^T X^T X)$$

□

B Théorie supplémentaire

B.1 Décomposition en valeurs singulières (SVD)

Lemme B.1. *Pour toute matrice $A \in \mathbb{R}^{n\times p}$, nous avons la décomposition orthogonale*

$$\mathbb{R}^p = \ker(A) \oplus^{\perp} \textit{range}(A^T).$$

Démonstration. Tout d'abord, nous observons que $\ker(A) \perp \text{range}(A^T)$. En effet, pour tout $y = A^T x \in \text{range}(A^T)$ et $x_0 \in \ker(A)$, nous avons

$$\langle x_0, y\rangle = \langle x_0, A^T x\rangle = \langle Ax_0, x\rangle = 0,$$

car $Ax_0 = 0$.

Puisque $\dim(\text{range}(A^T)) = \text{rang}(A^T) = \text{rang}(A) = p - \dim(\ker(A))$, et puisque $\ker(A) \perp \text{range}(A^T)$, la conclusion s'ensuit. □ □

Théorème B.2 (Décomposition en valeurs singulières (SVD))**.** *Pour toute matrice $A \in \mathbb{R}^{n\times p}$ de rang r, il existe une décomposition*

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^T,$$

où :

- *$r = \textit{rang}(A)$,*
- *$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ sont les valeurs singulières de A,*
- *$\{\sigma_1^2, \ldots, \sigma_r^2\}$ sont les valeurs propres non nulles de A^TA (ce sont aussi les valeurs propres non nulles de AA^T),*
- *$\{u_1, \ldots, u_r\}$ et $\{v_1, \ldots, v_r\}$ sont deux familles orthonormées dans $\mathbb{R}^n$ et $\mathbb{R}^p$, respectivement, satisfaisant :*

$$AA^T u_j = \sigma_j^2 u_j \quad \textit{et} \quad A^T A v_j = \sigma_j^2 v_j.$$

Démonstration. D'après le Lemme B.1, l'image de A et celle de AA^T coïncident, donc $\text{rang}(AA^T) = \text{rang}(A) = r$. Puisque AA^T est semi-définie positive et de rang r, elle admet une décomposition spectrale :

$$AA^T = \sum_{j=1}^{r} \lambda_j u_j u_j^T,$$

où $\lambda_1 \geq \cdots \geq \lambda_r > 0$ et $\{u_1, \ldots, u_r\}$ est une famille orthonormée dans $\mathbb{R}^n$.

Définissons les vecteurs $v_1, \ldots, v_r$ par :

$$v_j = \lambda_j^{-1/2} A^T u_j, \quad j = 1, \ldots, r.$$

Nous avons :

$$\langle v_i, v_j \rangle = \lambda_i^{-1/2} \lambda_j^{-1/2} u_i^T A A^T u_j = \lambda_i^{-1/2} \lambda_j^{1/2} u_i^T u_j = \delta_{i,j},$$

et

$$A^T A v_j = \lambda_j^{-1/2} A^T (A A^T) u_j = \lambda_j^{1/2} A^T u_j = \lambda_j v_j.$$

Ainsi, $\{v_1, \ldots, v_r\}$ est une famille orthonormée de vecteurs propres de $A^T A$. En posant $\sigma_j = \lambda_j^{1/2}$, on obtient :

$$\sum_{j=1}^{r} \sigma_j u_j v_j^T = \sum_{j=1}^{r} \sigma_j u_j (\lambda^{-1/2} A^T u_j)^T = \sum_{j=1}^{r} \lambda_J^{-1/2} \lambda_J^{-1/2} T u_j u_j^T A = (\sum_{j=1}^{r} u_j u_j^T) A$$

Pour vérifier cette décomposition, écrivons $P = \sum_{j=1}^{r} u_j u_j^T$. Nous remarquons que P est la projection orthogonale sur l'espace image de AA^T. D'après le Lemme B.1, l'espace image de AA^T coïncide avec celui de A. Ainsi, P est également la projection orthogonale sur l'espace image de A.

Nous avons alors :

$$PA = A,$$

et donc

$$\sum_{j=1}^{r} \sigma_j u_j v_j^T = \left(\sum_{j=1}^{r} u_j u_j^T \right) A = PA = A.$$

La preuve du Théorème B.2 est ainsi complète.

□

B.2 Analyse en Composantes Principales (PCA)

Théorème B.3 (Algorithme PCA)**.** *La solution de*

$$\hat{V_d} \in \underset{V : \dim(V) \leq d}{\operatorname{argmin}} \sum_{i=1}^{n} \| X^{(i)} - Proj_V X^{(i)} \|^2 \tag{14}$$

est le sous espace $V_d = span\{v_1, \ldots, v_d\}$. De plus, les coordonnées de $Proj_{V_d} X^{(i)}$ dans la base orthonormée $(v_1, \ldots, v_d)$ de V_d sont données par $(c_1^{(i)}, \ldots, c_d^{(i)})$, où $c_k^{(i)}$ désigne la i-ième entrée du vecteur $c_k := \sigma_k u_k \in \mathbb{R}^n$.

Démonstration. Pour commencer, nous observons que

$$\sum_{i=1}^{n} \|X^{(i)} - \mathrm{Proj}_V X^{(i)}\|^2 = \|X - X\mathrm{Proj}_V\|_F^2.$$

Pour tout sous-espace linéaire V de dimension d, le rang de la matrice $X\mathrm{Proj}_V$ n'est pas supérieur à d, donc, on peut montrer [2] que,

$$\sum_{i=1}^{n} \|X^{(i)} - \mathrm{Proj}_V X^{(i)}\|^2 = \|X - X\mathrm{Proj}_V\|_F^2 \geq \min_{\mathrm{rang}(B)\leq d} \|X - B\|_F^2 = \sum_{k=d+1}^{r} \sigma_k^2.$$

De plus, pour $V_d = \mathrm{span}\{v_1, \ldots, v_d\}$, nous avons

$$X\mathrm{Proj}_{V_d} = \sum_{k=1}^{r} \sigma_k u_k v_k^T \sum_{j=1}^{d} v_j v_j^T = \sum_{k=1}^{d} \sigma_k u_k v_k^T.$$

Ainsi,

$$\|X - X\mathrm{Proj}_{V_d}\|_F^2 = \| \sum_{k=d+1}^{r} \sigma_k u_k v_k^T \|_F^2 = \sum_{k=d+1}^{r} \sigma_k^2.$$

En comparant les deux équations, nous trouvons que $V_d = \mathrm{span}\{v_1, \ldots, v_d\}$ est la solution de 14.

De plus, la coordonnée de $\mathrm{Proj}_{V_d} X^{(i)}$ sur v_k est obtenue en prenant le produit scalaire entre $X^{(i)}$ et v_k,

$$\langle X^{(i)}, v_k \rangle = \langle X^T e_i, v_k \rangle = \langle e_i, X v_k \rangle = \sigma_k \langle e_i, u_k \rangle,$$

où nous avons utilisé pour la dernière égalité que $X v_k = \sigma_k u_k$. Ainsi, les coordonnées de $\mathrm{Proj}_{V_d} X^{(i)}$ dans la base orthonormée $(v_1, \ldots, v_d)$ de V_d sont données par $(c_1^{(i)}, \ldots, c_d^{(i)})$, où $c_k := \sigma_k u_k$. □ □

Références

[1] Jorge Cadima et Ian T. Jolliffe. « Loading and Correlations in the Interpretation of Principal Components ». In : *Journal of Applied Statistics* 22.2 (1995), p. 203-214. DOI : 10.1080/757584614. URL : https://doi.org/10.1080/757584614.

[2] Christophe Giraud. *Introduction to High Dimensional Statistics*. Accessed : [December 2024]. Boca Raton, FL : CRC Press, 2014. URL : https://www.crcpress.com/Introduction-to-High-Dimensional-Statistics/Giraud/p/book/9780367736144.

[3] Bruno Grisci. *GSE45827*. 2019. URL : https://www.kaggle.com/datasets/brunogrisci/breast-cancer-gene-expression-cumida.

[4] Peter J. B. Hancock, A. Mike Burton et Vicki Bruce. « Face Processing : Human Perception and Principal Components Analysis ». In : *Memory & Cognition* 24.1 (1996), p. 21-40. URL : https://pubmed.ncbi.nlm.nih.gov/8822156/.

[5] Christine Keribin. *ENSTA STA203 - Apprentissage statistique supervisé et non supervisé, Section 11.1 : Ridge Regression*. Accessed : [november 2024]. 2024. URL : https://www.imo.universite-paris-saclay.fr/~christine.keribin/STA203/ENSTA-STA203-Poly-2024.pdf.

[6] Robert Tibshirani. « Regression Shrinkage and Selection via the Lasso ». In : *Journal of the Royal Statistical Society : Series B (Methodological)* 58.1 (1996), p. 267-288. URL : http://www.jstor.org/stable/2346178.

[7] S. K. Vines. « Simple Principal Components ». In : *Journal of the Royal Statistical Society : Series C (Applied Statistics)* 49.4 (2000), p. 441-451. DOI : 10.1111/1467-9868.00204. URL : https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00204.

[8] Hui Zou et Trevor Hastie. « Regularization and Variable Selection Via the Elastic Net ». In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 67.2 (avr. 2005), p. 301-320. DOI : 10.1111/j.1467-9868.2005.00503.x. URL : https://doi.org/10.1111/j.1467-9868.2005.00503.x.