Sparse PCA

Rapport

Malkiel Riveline
M2 Mathématiques et Intelligence Artificielle

30 novembre 2023

Abstract

Ce rapport présente l'article *Sparse Principal Component Analysis* [1], à la lumière des cours d'apprentissage non supervisé avancé dispensés par C.Keribin lors du premier semestre de l'année 2023/2024. Il s'agit d'introduire des conditions de sparsity dans une méthode déjà très populaire de réduction de dimension, la PCA. Le lecteur pourra aussi trouver des analyses mathématiques, numériques, et historiques de cette méthode. Les codes et les références pour l'implémentation numérique sont disponibles sur Github à l'adresse:

`https://github.com/MlRiv/PCA_sparse`

Table des matières

Introduction

L'Analyse en Composantes Principales (dont nous utilisons l'acronyme anglais PCA), est un outil de réduction de dimension particulièrement utilisé mais qui présente un problème: les composantes principales contiennent une combinaison linéaire de toutes les variables, rendant l'interprétation difficile et la complexité plus grande pour un nombre de variables élevé. D'où l'idée naturelle d'une PCA pour variables corrélées en l'interprétant comme un problème de régression, avec un grand nombre de variables nulles (on parle alors de contraintes sparses). Alors, la distinction des variables significatives devrait être plus aisée. C'est l'objectif de l'article *Sparse Principal Component Analysis* ([1]), présenté dans ce rapport. Il s'articule autour des axes suivants: un résumé théorique de l'article, une mise en perspective dans la littérature scientifique, une analyse de l'implémentation numérique, et une discussion sur les améliorations du problème.
En raison de la diversité des traductions (éparse, parcimonieuse), nous nous tiendrons dans le reste du rapport à l'utilisation du terme "sparse". De même, pour ne pas nuire à la précision, certains anglicismes ("loadings" etc.) seront conservés.

Note sur les auteurs

L'article a été publié en 2006 par Hui Zou, son directeur de thèse Trevor Hastie et Robert Tibshirani, tous trois à l'université de Stanford, deux ans après la première soumission. Ces derniers, deux professeurs de statistiques, ont par ailleurs été co-auteurs avec Jerome Friedman du très classique *The Elements of statistical learning* [5]; R.Tibshirani est aussi connu pour la méthode régression LASSO en 1996, dont nous verrons la filiation avec les méthodes de sparse PCA. Hui Zou est entre temps devenu professeur à l'université du Minnesota. Il peut être intéressant de noter que l'article correspond au chapitre 3 de sa thèse de doctorat sur les modèles sparses, [2].

1 Coeur de l'article

L'article est composé de sept parties, en comptant l'introduction, la conclusion et l'annexe (les preuves des cinq théorèmes). Pour cette section, nous nous concentrons sur les résultats théoriques de la partie principale (section 3).

1.1 Résultats antérieurs

1.1.1 Prélude sur la PCA

L'origine de l'analyse en composantes principales n'est pas claire. Certains la font remonter à Pearson en 1901, d'autres l'attribuent à Hotelling (1933). Quoiqu'il en soit, il s'agit d'une des méthodes les plus utilisées avec des applications diverses en reconnaissance faciale ou en génomique (voir [4, 13] pour plus de détails).
D'un point de vue de la réduction de dimension, la PCA peut être décrite comme un ensemble de transformations linéaires orthogonales des variables d'origine, de sorte que les variables transformées conservent autant que possible les informations contenues dans les

variables d'origine.
Plus rigoureusement, soit X une matrice de données $n \times p$, où n et p sont le nombre d'observations et le nombre de variables, respectivement. La première composante principale est définie comme

$$Z_1 = \sum_{j=1}^{p} \alpha_{1j} X_j$$

où $\alpha_1 = (\alpha_{11}, \ldots, \alpha_{1p})^T$ est choisi pour maximiser la variance de Z_1, c'est-à-dire, en introduisant $\hat{\Sigma} = \frac{X^T X}{n}$:

$$\alpha_1 = \arg\max_{\alpha} \alpha^T \hat{\Sigma} \alpha \quad \text{sous la contrainte} \quad \|\alpha_1\| = 1$$

Les composantes principales restantes peuvent être définies par récurrence :

$$\alpha_{k+1} = \arg\max_{\alpha} \alpha^T \hat{\Sigma} \alpha \quad \text{sous les contraintes} \quad \|\alpha_k\| = 1 \quad \text{et} \quad \alpha_k^T \alpha_l = 0, \forall 1 \leq l \leq k$$

Cette définition implique que les k premiers vecteurs ("loading" vectors) sont les premiers vecteurs propres de $\hat{\Sigma}$.
La PCA a une autre interprétation géométrique (voir Fig:1), l'approximation linéaire de variété la plus proche des données observées. Cette définition correspond à la construction considérée par Pearson.
Expliquons: soit x_i la i-ème ligne de X. Considérant les k premières composantes principales conjointement $V^k = [V_1|\ldots|V_k]$, V^k est une matrice orthogonale de dimensions $p \times k$. Projetant chaque observation sur l'espace vectoriel engendré par $\{V_1, \ldots, V_k\}$, l'opérateur de projection est $P^k = V^k (V^k)^T$ et les données projetées sont $P^k X_i$, $1 \leq i \leq n$. Une manière de définir la meilleure projection est de minimiser l'erreur totale d'approximation ℓ_2 :

$$\min_{V^k} \sum_{i=1}^{n} \|x_i - V^k (V^k)^T x_i\|_2^2$$

On peut alors montrer que la solution est exactement les k premières composantes principales.

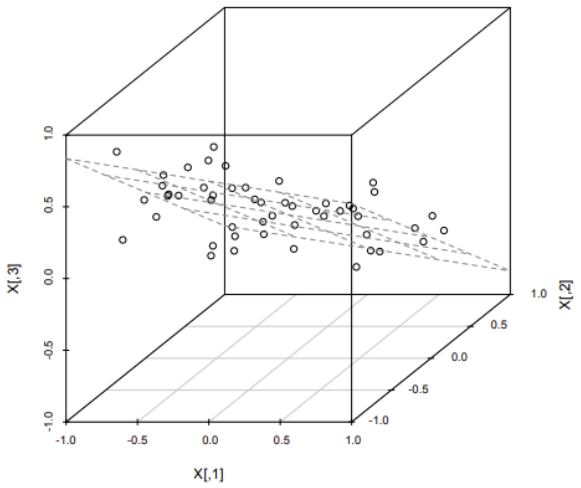

Figure 1: PCA en dimension 3. On cherche le plan V^2 le plus proche des données ([9], p.23)
.

Deux remarques pour conclure: la formulation de la décomposition en vecteurs propres de la PCA relie également la PCA à la décomposition en valeurs singulières (SVD) de X (voir Annexe A pour plus de détails). D'autre part, on a centré les variables (i.e la somme des colonnes de X est nulle: voir l'annexe pour une explication mathématique). Citons maintenant les résultats introduits par les statisticiens de Stanford dans les années 2000 avant la rédaction de [1]. En somme, les nouveautés que Zou avait en tête au moment de se lancer dans ses recherches.

1.1.2 LASSO et elastic net

L'idée de Tibshirani en introduisant le LASSO est de convexifier les problèmes et d'obtenir un "shrinkage" moins brutal que la regression ridge. S'il est abusif de dire que la méthode Lasso est toujours sparse, elle permet d'obtenir d'excellent résultats en grande dimension (voir [5] chapitre 5 et surtout 18.3 et 18.4 pour une présentation des deux méthodes).

LASSO (Least Absolute Shrinkage and Selection Operator) Le LASSO est une méthode de régularisation utilisée dans la régression linéaire. Il introduit une pénalité ℓ_1 sur les coefficients du modèle pour favoriser la sparsity, c'est-à-dire pour encourager certains coefficients à devenir exactement zéro (c'est le "shrinkage"). La formulation du problème d'optimisation du LASSO est donnée par:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}$$

où y_i est la variable dépendante, x_i est le vecteur de caractéristiques de la i-ème observation, β_0 est l'ordonnée à l'origine, β est le vecteur des coefficients de régression, et λ est le paramètre de régularisation.

Elastic Net Elastic Net est une extension du LASSO qui combine à la fois les termes de régularisation ℓ_1 et ℓ_2, et qui fut introduite par les mêmes auteurs en 2004 (chapitre 2 de la thèse de Zou, [2]). Cela permet de surmonter certains des inconvénients du LASSO, notamment lorsque le nombre de caractéristiques est élevé et plusieurs caractéristiques sont fortement corrélées. La formulation du problème d'optimisation de l'Elastic Net est donnée par:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2 \right\}$$

où les notations sont les mêmes que celles du LASSO, et λ_1 et λ_2 sont les paramètres de régularisation.

L'algorithme LARS (Least Angle Regression) L'algorithme LARS a été introduit par Bradley Efron en 2004. Il est utilisé pour résoudre le problème de régression lorsque le nombre de prédicteurs (p) est supérieur au nombre d'observations (n). Contrairement aux méthodes classiques qui ajoutent un prédicteur à la fois, LARS ajoute plusieurs prédicteurs simultanément, ce qui peut être plus efficace dans le contexte de données de grande dimension.

a) **Initialisation** : Commence avec tous les coefficients à zéro ($\beta_1 = \beta_2 = \ldots = \beta_p = 0$) et le résidu $r = y$.

b) **Étape a** : Trouve le prédicteur x_j le plus corrélé avec le résidu r (c'est-à-dire, celui dont le produit scalaire avec r a la plus grande valeur absolue).

c) **Étape b** : Ajuste le coefficient β_j dans la direction de x_j jusqu'à ce qu'un autre prédicteur x_k atteigne son coefficient dans le modèle ($\beta_k = \pm\gamma$, où γ est un paramètre déterminé par la condition LARS).

d) **Étape c** : Lorsque plusieurs prédicteurs ont atteint leurs coefficients, ajuste simultanément dans la direction jointe jusqu'à ce qu'un autre prédicteur atteigne son coefficient.

e) **Répétition** : Répète les étapes b et c jusqu'à ce que tous les prédicteurs aient atteint leurs coefficients.

L'algorithme LARS est particulièrement utile dans le contexte de la régression avec pénalisation (comme LASSO), car il permet de suivre le chemin de régularisation et d'obtenir des solutions pour différents niveaux de pénalité sans pour autant avoir à résoudre le problème d'optimisation à chaque étape. Cela en fait un outil efficace pour la sélection de modèles dans des espaces de grande dimension. Zou propose un algorithme LARS-EN similaire en considérant l'Elastic Net (voir [2]).

1.2 Apport de l'article

L'article apporte plusieurs résultats théoriques dont cinq théorèmes. Il semble que le résultat principal soit les théorèmes 2 et 3, qui reprend les résultats du régularisateur

elasticnet introduit par les mêmes auteurs et est régulièrement cité (par exemple: [2, 5]). Néanmoins, il semble intéressant de reprendre le cheminement de [1] pour y arriver.
L'idée est la suivante pour tous les théorèmes: reformuler la PCA avec une forme de relaxation.

$$\min_{\alpha,\beta} \sum_{i=1}^{n} \|x_i - \alpha\beta^T x_i\|_2^2$$

sous contrainte que $\|\alpha\|_2^2 = 1$ et $\alpha = \beta$.

L'idée des résultats de Zou consiste à éliminer la contrainte d'égalité ci-dessus et toujours récupérer exactement le premier *loading vector* V_1 (d'où la longue introduction essentielle sur la PCA).

Théorème: 1

Pour tout $\lambda_0 > 0$, soit $(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \|\mathbf{x}_i - \alpha\beta^T\mathbf{x}_i\|_2^2 + \lambda_0\|\beta\|_2^2$ sous contrainte $\|\alpha\|_2^2 = 1$. Alors $\hat{\beta} \propto V_1$.

Avec α fixé, le problème d'optimisation sur β est un problème de régression.

En se basant sur le Théorème 1, nous pouvons imposer une pénalité sparse sur β pour obtenir une charge nulle, car l'étape de normalisation ne change pas le support de β. Le critère pour la première composante principale est défini comme suit :

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \|\mathbf{x}_i - \alpha\beta^T\mathbf{x}_i\|_2^2 + \lambda_0\|\beta\|_2^2 + \lambda_1\|\beta\|_1$$

sous contrainte $\|\alpha\|_2^2 = 1$, et le *loading vector* de sortie est $V_1 = \hat{\beta}/\|\hat{\beta}\|_2$. Zou propose deux solutions efficaces: pour $n > p$, laisser $\lambda_0 = 0$ et résoudre β avec un α fixé comme un problème de régression lasso, ce qui peut être fait efficacement. Lorsque $n < p$, utiliser une λ_0 positif (par exemple, $\lambda_0 = 10^{-3}$), résoudre β avec un α fixé comme un problème de régression elasticnet.

Le Théorème 1 peut être généralisé pour traiter simultanément les k premiers composantes principales, comme indiqué dans le théorème suivant.

Théorème: 2

Supposons que nous considérons les k premières composantes principales. Soit $A_{p\times k} = [\alpha_1, \ldots, \alpha_k]$ et $B_{p\times k} = [\beta_1, \ldots, \beta_k]$. Pour tout $\lambda_0 > 0$, posons

$$(\hat{A}, \hat{B}) = \arg\min_{A,B} \sum_{i=1}^{n} \left\|x_i - AB^T x_i\right\|^2 + \lambda_0 \sum_{j=1}^{k} \|\beta_j\|^2$$

sous contrainte $A^\top A = I_{k\times k}$.

Alors, $\hat{\beta}_j \propto V_j$ pour $j = 1, 2, \ldots, k$.

Ces théorèmes permettent d'introduire la fonction objectif (ou le critère) pour les premières k composantes principales (SPCA) est définie comme

$$(\hat{A}, \hat{B}) = \arg\min_{A,B} \sum_{i=1}^{n} \left\| x_i - AB^T x_i \right\|^2 + \lambda_0 \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j} \|\beta_{j,1}\|$$

sous contrainte $A^\top A = I_{k\times k}$, où différents $\lambda_{1,j}$ sont autorisés pour pénaliser les chargements des différentes composantes principales.

1.3 Avantage théorique de la méthode

Une question évidente se pose: quel intérêt d'estimer une composante que l'on peut calculer explicitement? En réalité, la grande dimension implique une multitude d'avantages. Cette approche conduit à une amélioration de l'interprétabilité du modèle, car les composantes principales sont formées comme une combinaison linéaire de seulement quelques-unes des variables d'origine. De plus, les méthodes sparses évitent l'overfitting dans un contexte de données de grande dimension où le nombre de variables p est supérieur au nombre d'observations n.

1.4 Résumé à mi-parcours

On a donc affaire à une variante moderne de la PCA. Plus précisément, elle tente de trouver des vecteurs de poids avec seulement quelques valeurs "actives" non nulles.

Un tel modèle sparse est obtenu en introduisant des régularisateurs favorisant le shrinkage. Plus concrètement, étant donné une matrice de données (n, p) X, on tente de minimiser la fonction suivante :

$$f(A, B) = \frac{1}{2}\|X - XBA^T\|_F^2 + \psi(B)$$

où B est la matrice de poids sparse ("loadings"), et A est une matrice orthonormale. ψ désigne un régularisateur induisant la sparsity, en l'occurence elastic net (une combinaison des normes ℓ^1 et ℓ^2). L'idée "computationelle" est d'alterner suivant que l'on ait sous la main A (un problème de régression pour estimer B selon elastic net) ou B (un problème pour estimer A, qui se résoud par une SVD).

2 Exemples numériques

L'article présente des résultats sur le dataset `pritprops`. Cette partie est basée sur un repository Github, dont l'adresse est rappelée:

`https://github.com/MlRiv/PCA_sparse/`

2.1 Schéma des algorithmes

L'algorithme final de Zou est décrit dans l'article:

Algorithm 1: Algorithme général SPCA

a) Soit A initialisée à $V[:, 1:k]$, les loadings des k premières composantes principales ordinaires.

b) Étant donné un A fixé $= [\alpha_1, \ldots, \alpha_k]$, résoudre le problème de régularisation suivant pour $j = 1, 2, \ldots, k$:

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

c) Pour un B fixé $= [\beta_1, \ldots, \beta_k]$, calculer la SVD de $X^T X = UDV^T$, puis mettre à jour $A = UV^T$.

d) Répéter les étapes b-c jusqu'à convergence.

e) Normalisation : $\hat{V}_j = \frac{\beta_j}{\|\beta_j\|},\ j = 1, \ldots, k.$

La convergence b) est assurée par des résultats du théorème 2 et c) par un résultat d'analyse matricielle d'approximation d'une matrice de rang k par la SVD (notons qu'on a changé les numéros de l'article, puisque nous n'avons conservé que les résultats originaux). On peut, par l'alternance et la simultanéité, trouver une filiation certaine avec l'algorithme LARS introduit plus haut (voir de l'aveu même de Zou [4]: "motivated by LARS"). Notons aussi qu'il introduit une méthode d'accélération basée sur le soft-thresholding (présenté dans l'annexe A.3).

2.2 Packages existants

Un rapide tour d'horizon permet de s'apercevoir de l'existence de bon nombre d'implantations de méthode de PCA sparse en `R`; on peut encore citer des implémentations en matlab (voir par exemple la toolbox SpaSM de à l'adresse: `http://www2.imm.dtu.dk/projects/spasm/`) ou en python via la version 1.2 de scikit-learn (`https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.SparsePCA.html`). Le dataset utilisé par Zou est très classique en PCA: `pitprops`. Tirées d'un article de Jeffers (1967), les données pitprops comprennent des informations sur 13 variables mesurées pour 180 rondins de bois découpés à partir de bois de pin noir (voir le code pour plus de détails).

2.2.1 Reproduction des résultats grâce au package `elasticnet`

Le package `elasticnet` fut publié en même temps que la thèse de Zou, qui le maintient depuis. Fait notable: la documentation a été mise à jour par Zou en 2022, ce qui montre que la méthode est toujours utilisée. La fonction spca applique directement l'algorithme décrit. La reproduction des résultats afférents peut poser de problèmes: nous prenons en annexe un exemple de la reproduction des cinq tableaux et trois figures de l'article (pour les autres figures, voir le repository).

2.2.2 Amélioration en R

On peut noter deux packages qui généralisent la PCA sparse: un par Daniella Witten et Robert Tibshirani ([6], voir la section Méthodes alternatives), et un autre par Erichson ([8] qui introduit des méthodes non convexes: un algorithme de descente de gradient proximal par coordonnées et des méthodes aléatoires)

Enfin, une meilleure appréciation de la méthode serait son utilisation sur d'autres datasets avec un setting $p > n$, voire p très grand devant n. Pour un article comparant onze datasets (dont `pitprops` et aussi `micromass` avec $p = 1300$), on peut citer la récente comparaison de Cory Wright et J.Pauphilet (2023: `https://arxiv.org/pdf/2209.14790.pdf`). L'article étant néanmoins assez touffu, il est assez compliqué d'extraire des résultats clairs.

3 Mise en perspective de la méthode et appréciation historique

Nous avons donc vu la méthode de PCA sparse introduite par Zou. Cet article s'inscrit non seulement dans la continuité des méthodes d'Hastie et Tibshirani, mais aussi dans celle des méthodes de PCA.

3.1 Histoire de la PCA sparse

Il serait faux de dire que Zou a introduit la PCA sparse. D'une part, parce que l'idée d'imposer une condition de sparsity est bien antérieure à 2004; d'autre part, parce qu'elle semble particulièrement naturelle après la PCA.

3.1.1 SCoTLASS et ses défauts

Ian Joliffe [7] introduit en 2003 un résultat qui consiste en une remarque simple: on veut que la caractérisation de la variance maximale de la PCA soit modifiée pour incorporer la sparsity. La modification la plus naturelle consisterait à imposer une contrainte ℓ_0 sur le critère, conduisant au problème

$$\max_{\|v\|_2=1} v^T X^T X v$$

sous contrainte que $\|v\|_0 \leq t$ où $\|v\|_0$ compte le nombre de valeurs non nulles dans le vecteur v. Cependant, ce problème est doublement non convexe, car il implique la maximisation (plutôt que la minimisation) d'une fonction convexe avec une contrainte combinatoire. La procédure SCoTLASS est une relaxation naturelle de cet objectif, en remplaçant la norme ℓ_0 par la norme ℓ_1, conduisant à

$$\max_{\|v\|_2=1} v^T X^T X v$$

sous contrainte que $\|v\|_1 \leq t$.

La contrainte ℓ_1 encourage certaines des charges à être nulles et donc v à être sparse. Bien que la norme ℓ_1 soit convexe, le problème global reste non convexe puisqu'on maximise une fonction convexe (ajout du 30 novembre après l'oral) et coûte cher numériquement. C'est après ces difficultés que Zou intervient.

3.1.2 Portée et applications

Au delà, on peut lister bon nombre d'applications directes et probantes des méthodes de PCA sparse (voir [4] section IV pour plus de détails): en neuroscience, en image processing, ou encore pour un modèle écologique (pour déterminer les raisons de l'invasion d'une espèce en Nouvelle Zélande).

Continuons cette mise en perspective par une analyse bibliométrique succinte.

3.2 Bibliométrie

Même si le nombre de citation est souvent un indicateur imparfait, l'indice de citation Google Scholar indique 3786 à la date de ce rapport (moitié moins pour l'outil Crossref). Néanmoins, ces indications semblent biaisées, puisque la méthode proposée est implémentée dans le package `elasticnet`, dont la référence [3] est citée plus de 19000 fois. Plusieurs nuances contradictoires sont alors à introduire: d'une part, comme nous l'avons vu, l'article s'inscrit dans la lignée des méthodes de PCA et peut n'être cité qu'en introduction historique. D'autre part, il faudrait analyser les articles utilisant [3] pour tirer des conclusions précises quant à l'utilisation de la fonction PCA sparse (spca). Il n'en reste pas moins que ces indices, si lacunaires soient-ils, indiquent un intérêt certain de la communauté scientifique pour la méthode de Zou.

Il s'agit du troisième article le plus cité de Zou, et parmi les dix les plus cités de Hastie. Une autre indication intéressante de la portée de l'article est sa récente utilisation, stable depuis plus de dix ans (voir le graphe à l'adresse: https://scholar.google.com/citations). Aussi la méthode est-elle citée 249 fois en 2023, près de vingt ans après sa mise au point, pour des sujets allant du Deep learning au trafic ferroviaire, en passant par la médecine du poumon. Ce qui confirme la popularité de la méthode.

Ensuite, *The Elements of Statistical Learning* reprend l'article dans sa deuxième édition (section 14.5.5 sur Sparse Principal Component, voir [5] p.550). Vu l'extrême popularité du livre, il y a fort à parier de l'influence de cet article sur toute une génération de statisticiens pour les méthodes de PCA sparse.

Enfin, détail sémantique qui a son importance, l'article [1] est souvent décrit en des termes élogieux. Pour prendre un échantillon non exhaustif, citons certains qualificatifs particulièrement flatteurs: "iconique" [12] ou encore "des plus populaires" [13].

4 Ouvertures et améliorations

La méthode a été acclamée et très utilisée. Des résultats théoriques ont depuis trouvé une meilleure consistance de l'estimateur sparse en grande dimension ([4], section III). Seulement, la tendance récente n'est plus seulement à se ramener à un problème d'optimisation convexe, mais parfois d'utiliser des algorithmes d'optimisation non convexe (voir [9] chapitre 5 pour une justification en grande dimension). Un papier récent de Zou revient sur la PCA sparse [4] et liste plusieurs améliorations ou plutôt alternatives à sa méthode elastic net. Les méthodes ne seront pas développées et reposent sur les résultats des sections A.2 et A.3.1.

4.1 Méthodes alternatives

Programmation semi-définie: cette approche fut introduite par le professeur français D'Aspremont en 2007, il s'agit de relaxer encore plus le SCoTLASS dans un espace bien choisi. En fait, il montre l'équivalence du problème de SCoTLASS avec le problème suivant, sous une autre contrainte (M de rang 1). L'aspect est donc radicalement différent de celui de Zou.

$$\text{maximiser}_{\text{ M est semi définie positive}} \text{tr}(X^T X M)$$
$$\text{sous réserve que } \text{tr}(M) = 1, \text{tr}(|M|E) \leq t^2.$$

où $E \in \mathbb{R}^{p \times p}$ est une matrice de uns, et $|M|$ est la matrice obtenue en prenant les valeurs absolues élément par élément.

Méthodes itératives de thresholding Il s'agit de lier la PCA à la SVD et au résultat présenté et prouvé dans l'annexe A.2, qu'il est fortement conseillé de lire avant de considérer la méthode. Shen & Huang [11] ont donc proposé le problème d'optimisation suivant:

$$(U, \hat{V}) = \arg\min_{U,V} \|X - UV^T\|_F + \lambda\|V\|_1 \quad \text{sous la contrainte} \quad \|U\| = 1,$$

et le vecteur normalisé $\hat{V}$. Notons que, étant donné V, le U optimal est $U = XV/\|XV\|$. Étant donné U, le V optimal est

$$\arg\min_V -2\text{Tr}(X^T U V^T) + \|V\|_2^2 + \lambda\|V\|_1,$$

et la solution est donnée par l'opérateur de Soft-thresholding (voir Annexe):

$$V = S(X^T U, \lambda/2).$$

Ainsi, la méthode de Shen et Huang est un algorithme itératif. La procédure est similaire à l'algorithme SPCA spécial de Zou. La grande différence est que le SPCA résout simultanément k composantes, tandis que la méthode de Shen et Huang ne traite qu'une composante à la fois.

Dans le même esprit, Witten et al. [6] ont proposé un critère de décomposition matricielle pénalisée (PMD) comme suit:

$$(U, \hat{V}, \hat{d}) = \arg\min_{U,V,d} \|X - dUV^T\|_F$$

sous la contrainte $\|U\| = 1, \|U\|_1 \leq c_1; \|V\| = 1, \|V\|_1 \leq c_2$.

On peut montrer l'équivalence avec le problème:

$$(\hat{U}, \hat{V}) = \arg\max_{U,V} U^T X V$$

Sous la contrainte $\|U\| = 1$, $\|U\|_1 \leq c_1$, $\|V\| = 1$, $\|V\|_1 \leq c_2$, et $\hat{d} = \hat{U}^T X V$.

On pourrait aussi citer la *Generalized power method* qui approfondit encore l'idée de Witten.

D'où la remarque importante sur la spécificité de la méthode de [1]: la plupart des formulations pour "sparse PCA" reposent sur différentes formulations de la PCA, de sorte que les problèmes d'optimisation correspondants sont différents. Contrairement à une PCA ordinaire, elles ne donnent pas des solutions équivalentes puisque les différentes méthodes donnent des estimations "sparse" pour des structures de modèle différentes. Ainsi, la méthode choisie doit dépendre de l'objectif de l'analyse et de la structure pour laquelle la sparsity est souhaitée.

4.2 Ouvertures

L'étude depuis plus de vingt ans des problèmes de PCA sparse s'est beaucoup développé comme en atteste la partie précédente. Plusieurs problèmes ouverts restent néanmoins à étudier:

a) Une procédure générique pour choisir les setting sparses (i.e, les contraintes les plus efficaces). La spca serait alors en quelque sorte "automatique".

b) L'application au Deep Learning.

c) D'avantage d'analyses empiriques pour permettre une méta-analyse rigoureuse des différents types de PCA sparse pour mieux cerner quelle méthode utiliser.

Conclusion

L'article [1] semble encore plus difficile à catégoriser qu'en introduction. D'un coté, il développe les méthodes de PCA et s'inscrit dans la lignée de Jollife (PCA sparse) et Tibshirani (convexification). D'un autre, il est profondément novateur dans la mesure où il combine plusieurs techniques nouvelles pour l'époque (cette faculté à combiner des méthodes s'observe encore plus clairement pour elastic net, [3]). Il s'inscrit dans la continuité de l'étude de la PCA et de sa constante reformulation pour construire des passerelles avec d'autres domaines: la SVD repose sur des mathématiques du XIX^{e} siècle, et des passerelles entre algèbre linéaire, statistiques et optimisation. Ces équivalences et ces liaisons m'ont semblé passionnantes à étudier, d'autant que les problèmes en grande dimension promettent de se multiplier dans un futur proche.

A Annexe

L'annexe consiste en des développements mathématiques liés à la PCA et de discussions autour des angles morts laissés par ce rapport, qui nuiraient à la fluidité du corpus principal.

A.1 Reproduction des résultats d'un tableau

	X	PC1	PC2	PC3	PC4	PC5	PC6
1	topdiam	-0.40	0.22	-0.21	0.09	-0.08	0.12
2	length	-0.41	0.19	-0.24	0.10	-0.11	0.16
3	moist	-0.12	0.54	0.14	-0.08	0.35	-0.28
4	testsg	-0.17	0.46	0.35	-0.05	0.36	-0.05
5	ovensg	-0.06	-0.17	0.48	-0.05	0.18	0.63
6	ringtop	-0.28	-0.01	0.48	0.06	-0.32	0.05
7	ringbut	-0.40	-0.19	0.25	0.06	-0.22	0.00
8	bowmax	-0.29	-0.19	-0.24	-0.29	0.19	-0.06
9	bowdist	-0.36	0.02	-0.21	-0.10	-0.11	0.03
10	whorls	-0.38	-0.25	-0.12	0.21	0.16	-0.17
11	clear	0.01	0.21	-0.07	-0.80	-0.34	0.18
12	knots	0.12	0.34	0.09	0.30	-0.60	-0.17
13	diaknot	0.11	0.31	-0.33	0.30	0.08	0.63
14	Variance (%)	0.32	0.18	0.14	0.09	0.07	0.06
15	Cumulative variance (%)	0.32	0.51	0.65	0.74	0.81	0.87

Figure 2: Reproduction de la Table 1, avec les mêmes valeurs (PCA)

Sa principale observation est une bonne performance sur la variance observée (la variance cumulée "transporte" l'information), pour un setting sparse (voir codes et présentation). Aussi, notons ses expérience sur deux autres dataset, l'un synthétique et l'autre réel (identification de gènes).

- Partout, il compare les PCA et SPCA avec du soft-thresholding dont l'accélération est discutée en A.3.1 (avec un léger angle mort, puisqu'il le rend équivalent au simple thresholding, intégré dans `elasticnet`) et SCoTLASS.
- Il étudie un modèle synthétique pour retrouver un modèle caché à partir de données observées. La SPCA identifie correctement les variables importantes. A l'inverse, le seuillage inclut à tort des variables moins importantes en raison d'une corrélation élevée.
- Il compare sur des données médicales de soft-thresholding et SPCA pour un problème d'identification de gènes où le soft-thresholding performe bien.

A.2 Lien entre PCA et SVD

Nous donnons ici la preuve rigoureuse du lien entre PCA et SVD, qui prolonge l'idée vue en cours et est essentielle pour certaines méthodes ([11], notamment). Cela permet aussi de comprendre pourquoi ce lien n'est plus évident dans le cadre d'autres PCA sparse.

A.2.1 Résultats d'analyse matricielle

Rappelons d'abord la définition de la SVD:

Théorème: SVD (Singular Value Decomposition)

Toute matrice A de taille $n \times p$ de rang r peut être décomposée comme

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

où

- $r = \text{rang}(A)$,
- $\sigma_1 \geq \ldots \geq \sigma_r > 0$,
- $\sigma_1^2, \ldots, \sigma_r^2$ sont les valeurs propres non nulles de $A^T A$ (qui sont également les valeurs propres non nulles de AA^T) que l'on note parfois $\sigma_1^2(A), \ldots, \sigma_r^2(A)$,
- $\{u_1, \ldots, u_r\}$ et $\{v_1, \ldots, v_r\}$ sont deux familles orthonormales de $\mathbb{R}^n$ et $\mathbb{R}^p$ telles que
$$AA^T u_j = \sigma_j^2 u_j \quad \text{et} \quad A^T A v_j = \sigma_j^2 v_j.$$

Rappelons également l'expression de la norme de Frobenius:

Définition: Norme de Frobenius et norme de Ky-Fan (2,q)

Le produit scalaire standard sur les matrices est donné par $\langle A, B \rangle_F = \sum_{i,j} A_{i,j} B_{i,j}$. Il induit la norme de Frobenius :

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2} = \sqrt{\text{Tr}(A^T A)} = \sqrt{\sum_k \sigma_k(A)^2}.$$

On peut considérer la norme Ky–Fan (2,q) avec $q = \text{rang}(A) \wedge \text{rang}(B)$:

$$\|A\|_{(2,q)}^2 = \sum_{k=1}^{q} \sigma_k(A)^2$$

On observe que $\|A\|_{(2,q)} \leq \|A\|_F$, avec une stricte inégalité si $q < \text{rang}(A)$.

Enfin, introduisons un théorème caractérisant la "projection" sur l'ensemble des ma-

trices de rang r. Il fournit également une amélioration de l'inégalité de Cauchy–Schwartz $\langle A, B\rangle_F \leq \|A\|_F \|B\|_F$ en termes de la norme Ky–Fan (2,q) :

$$\|A\|_F \|B\|_F \leq \|A\|_{(2,q)} \|B\|_{(2,q)}$$

Théorème: Approximation de bas rang

Pour toute matrice $A, B \in \mathbb{R}^{n\times p}$, on pose $q = \text{rang}(A) \wedge \text{rang}(B)$. Alors, on a

$$\langle A, B\rangle_F \leq \|A\|_{(2,q)} \|B\|_{(2,q)},$$

Par conséquent, pour $A = \sum_{k=1}^{r} \sigma_k(A) u_k v_k^T$ et $q < r$, on a

$$\min_{B:\text{rang}(B)\leq q} \|A - B\|_F = \sum_{k=q+1}^{r} \sigma_k(A)^2.$$

De plus, le minimum est atteint pour

$$B = \sum_{k=1}^{q} \sigma_k(A) u_k v_k^T.$$

Démonstration. On pourra trouver une preuve [9], p.315 qui consiste à projeter sur $\text{Im}(B)$. □

A.2.2 Reformulation de la PCA

Pour tout ensemble de points de données $\mathbf{X}(1), \ldots, \mathbf{X}(n) \in \mathbb{R}^p$ et toute dimension $d \leq p$, la PCA calcule le sous-espace vectoriel dans $\mathbb{R}^p$:

$$V_d \in \arg\min_{\dim(V)\leq d} \sum_{i=1}^{n} \|\mathbf{X}(i) - \text{Proj}_V \mathbf{X}(i)\|_2^2,$$

où Proj_V est la matrice de projection orthogonale sur V.
Soit $\mathbf{X} = \sum_{k=1}^{r} \sigma_k u_k v_k^T$ une SVD de la matrice $n \times p$:

$$\mathbf{X} = \begin{bmatrix} (\mathbf{X}(1))^T \\ \vdots \\ (\mathbf{X}(n))^T \end{bmatrix},$$

avec $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$.
Terminologie : les vecteurs propres à droite $v_1, \ldots, v_r$ sont appelés les axes principaux. Les vecteurs $c_k = \mathbf{X} v_k = \sigma_k u_k$ pour $k = 1, \ldots, r$ sont appelés les composantes principales. La composante principale c_k regroupe les coordonnées de $\mathbf{X}(1), \ldots, \mathbf{X}(n)$ sur v_k.
Remarque : puisque V_d est un espace vectoriel et non un espace affine, on comprend l'usage

de centrer d'abord les données :

$$\mathbf{X}_e(i) = \mathbf{X}(i) - \frac{1}{n}\sum_{i=1}^{n}\mathbf{X}(i)$$

et ensuite procéder à la PCA sur $\mathbf{X}_e(1), \ldots, \mathbf{X}_e(n)$.

A.2.3 Démonstration du lien

La démonstration se fait en trois étapes:

1. Relier le problème de minimisation de la PCA à la SVD.

On a pour tout $d \leq r$, où l'on rappelle que r est le rang de X:

$$\sum_{i=1}^{n}\|\mathbf{X}(i) - \mathrm{Proj}_V\mathbf{X}(i)\|_2^2 = \|\mathbf{X} - \mathbf{X}\mathrm{Proj}_V\|_F^2 \geq \sum_{k=d+1}^{r}\sigma_k^2,$$

Démonstration: On se base sur la symétrie de la projection orthogonale pour la première égalité.

$$(\mathrm{Proj}_V\mathbf{X}(i))_j = \sum_{k=1}^{p}(\mathrm{Proj}_V)_{j,k}X(i)_k = \sum_{k=1}^{p}X(i)_k(\mathrm{Proj}_V)_{k,j} = (X\mathrm{Proj}_V)_{i,j}$$

On a alors :

$$\begin{aligned}
&\sum_{i=1}^{n}\|\mathbf{X}(i) - \mathrm{Proj}_V\mathbf{X}(i)\|_2^2 \\
&= \sum_{i=1}^{n}\sum_{j=1}^{p}(X(i)_j - (X\mathrm{Proj}_V)_{i,j})^2 \\
&= \|\mathbf{X} - X\mathrm{Proj}_V\|_F^2
\end{aligned}$$

Nous allons maintenant appliquer le théorème d'approximation de bas rang à $X = \sum_{k=1}^{r}\sigma_k u_k v_k^T$. Pour tout $d < r$, on a :

$$\min_{B:\mathrm{rang}(B)\leq d}\|\mathbf{X} - B\|_F = \sum_{k=d+1}^{r}\sigma_k^2$$

Par conséquent, on a **une borne inférieure**:

$$\|\mathbf{X} - X\mathrm{Proj}_V\|_F \geq \sum_{k=d+1}^{r}\sigma_k^2$$

puisque $\mathrm{rang}(X\mathrm{Proj}_V) \leq \dim V \leq d$.

Le cas $d = r$ est trivial car la somme est égale à 0. □

2. Caractériser $\mathbf{Proj}_{V_d}$ où V_d l'espace linéaire engendré par $\{v_1, \ldots, v_d\}$ pour $d \leq r$ et montrer qu'il atteint la borne inférieur précédente.

Démonstration. Définissons $Q = \sum_{k=1}^{d} v_k v_k^T$. Q est clairement un projecteur orthogonal par orthogonalité des $\{v_1, \ldots, v_d\}$. Prouvons qu'il est égal Proj_{V_d}.

Soit $\mathbf{X} \in \mathbb{R}^p$, on a la décomposition: $\mathbb{R}^p = V_d \oplus V_d^{\perp}$. D'où:

$$\mathbf{X} = \mathbf{Y} + \mathbf{Z}$$

avec $\mathbf{Y} = a_1 v_1 + \ldots + a_d v_d$ et $\mathbf{Z} \in V_d^{\perp}$. Par orthogonalité:

$$Q\mathbf{Z} = \left(\sum_{k=1}^{d} v_k v_k^T \right) \mathbf{Z} = \sum_{k=1}^{d} v_k (v_k^T \mathbf{Z}) = 0$$

Ainsi, $Q\mathbf{X} = Q\mathbf{Y} + Q\mathbf{Z} = Q\mathbf{Y}$.

$$Q\mathbf{X} = \sum_{k=1}^{d} \sum_{i=1}^{d} a_i v_k (v_k^T v_i)$$

Or $v_k^T v_i = \delta_{k,i}$:

$$Q\mathbf{X} = \sum_{k=1}^{d} a_k v_k = \mathbf{Y} = \mathrm{Proj}_{V_d} \mathbf{X}$$

et donc $Q = \mathrm{Proj}_{V_d}$.

Soit $d < r$, avec $\mathbf{X} = \sum_{k=1}^{r} \sigma_k u_k v_k^T$, nous avons :

$$\mathbf{X}\mathrm{Proj}_{V_d} = \left(\sum_{k=1}^{r} \sigma_k u_k v_k^T \right) \left(\sum_{i=1}^{d} v_i v_i^T \right) = \sum_{k=1}^{r} \sum_{i=1}^{d} \sigma_k u_k (v_k^T v_i) v_i^T$$

Par orthogonalité, on a:

$$\mathbf{X}\mathrm{Proj}_{V_d} = \sum_{k=1}^{d} \sigma_k u_k v_k^T$$

Ainsi, on peut appliquer la dernière partie du théorème d'approximation:

$$\|\mathbf{X} - \mathbf{X}\mathbf{Proj}_{V_d}\|_F = \sum_{k=d+1}^{r} \sigma_k^2$$

Si $d = r$, on a encore une somme nulle. □

Ce qui permet de conclure en donnant une expression des coordonnées de la projection: le calcul de la SVD donne donc les composantes principales.

3. Les coordonnées de $\mathbf{Proj}_{V_d}\mathbf{X}(i)$ dans la base orthonormée $\{v_1, \ldots, v_d\}$ de V_d sont données par $(\sigma_1 \langle e_1, u_1 \rangle, \ldots, \sigma_d \langle e_d, u_d \rangle)$ avec e_i le i-ieme vecteur de la base canonique.

Démonstration. Soit $(a(i)_1, \ldots, a(i)_d)$ les coordonnées de $\mathrm{Proj}_{V_d}\mathbf{X}(i)$ dans la base orthonormée $(v_1, \ldots, v_d)$.

On a $a(i)_j = \langle \mathrm{Proj}_{V_d}\mathbf{X}(i), v_j \rangle = \langle \mathbf{X}(i), v_j \rangle$, puisque $\mathrm{Proj}_{V_d} = (\mathrm{Proj}_{V_d})^T$ et $\mathrm{Proj}_{V_d} v_j = v_j$.

Or, $(\mathbf{X}(i))^T$ est la i-ème ligne de $\mathbf{X}$. On a:

$$(\mathbf{X}(i))^T = \sum_{k=1}^{r} \sigma_k (e_i^T u_k) v_k^T$$

D'où, en transposant:

$$\mathbf{X}(i) = \sum_{k=1}^{r} \sigma_k \langle e_i, u_k \rangle v_k$$

Ce qui donne, par linéarité:

$$a(i)_j = \langle \mathbf{X}(i), v_j \rangle = \sum_{k=1}^{r} \sigma_k \langle e_i, u_k \rangle \langle v_k, v_j \rangle$$

Puisque $\langle v_k, v_j \rangle = \delta_{k,j}$, on peut conclure que :

$$a(i)_j = \sigma_j \langle e_i, u_j \rangle$$

□

A.3 Explications autour des démonstrations et des remarques secondaires de l'article

A.3.1 Accélération et Soft-Thresholding

Zou et Hastie ont dérivé un autre critère pour accélérer davantage l'efficacité numérique. La dérivation est basée sur l'observation que le Théorème 2 est valide pour tous les $\lambda_0 > 0$. Il s'avère qu'une solution économique émerge si λ_0 est choisi comme une constante élevée. Introduisons avant:

Définition: Soft-thresholding

$S(Z, \gamma)$ est l'opérateur soft-thresholding sur un vecteur $Z = (z_1, ..., z_p)$ avec le paramètre de seuillage γ et défini par:

$$S(Z, \gamma)_j = (\|z_j\| - \gamma)_+ \text{sgn}(z_j)$$

Théorème

Soit $(\widehat{V_j}(\lambda_0))_j$ les k premiers loadings définis par le critère SPCA. Soit $(\widehat{A}, \widehat{B})$, la solution du problème d'optimisation

$$\arg\min_{A,B} -2\text{Tr}\left(A^T X^T X B\right) + \sum_{j=1}^{k} \|\beta_j^k\|_2^2 + \sum_{j=1}^{k} \lambda_{1,jk} \|\beta_j^k\|_1$$

soumis à $A^T A = I_{k \times k}$. Lorsque $\lambda_0 \to \infty$, $\widehat{V_j}(\lambda_0) \to \hat{\beta}_j$(normé).

La résolution de ce nouveau problème peut également être effectuée avec l'algorithme

SPCA légèrement modifié. Pour A fixé, nous avons que pour chaque j,

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

et la solution est donnée par

$$\hat{\beta}_j = S(X^T X \alpha_j, \frac{\lambda_{1,j}}{2}),$$

où $S(Z, \gamma)$ est l'opérateur de Soft-thresholding. Étant donné B, la solution de A est à nouveau $\hat{A} = UV^T$ où U, V proviennent de la SVD de $(X^T X)B$: $(X^T X)B = UDV^T$.

A.3.2 La variance ajustée

La section 3.5 est une section plutôt complexe de l'article et a pour principal résultat une variance ajustée de la matrice des composantes $\widehat{U}$ qui aboutit à la norme de la matrice R découlant de la décomposition QR de $\widehat{U}$. Zou est gêné du caractère potentiellement corrélé des composantes principales estimées. Il calcule donc une autre variance ajustée sur les résidus des composantes d'ordre inférieurs. Or, la décomposition QR peut être obtenue grâce au procédé de Gram-Schmidt des colonnes de U, qui peut s'interpréter comme un ajustement des résidus d'une base de $\mathbb{R}^p$; la covariance étant un produit scalaire, son idée consisterait à "orthonormaliser" sa base de composantes. Cette analogie m'a paru intéressante pour comprendre son point.

A.3.3 Ambiguïté des notations

Les différentes lectures d'articles liés à la PCA conduit à remarquer une utilisation parfois erratique de certaines notations. Ces difficultés renferment parfois quelques subtilités, qu'il n'est peut-être pas inutile d'inclure dans le rapport.

Une certaine ambiguïté est notée par [10, 13] quant à l'utilisation du terme "loadings" et "variables", qui ne sont pas égales dans un setting sparse. Les loadings proviennent des loading vectors de la PCA, qui sont les coefficients de la matrice de covariance uniquement dans le cadre de la PCA classique. De même, la notion de sparsity est floue, dans la mesure où l'objectif n'est pas toujours précisé. Au risque de répéter, les estimations sparses endommagent les équivalences de la PCA, et la formulation du problème et de l'objectif se révèle déterminante.
Notons enfin une autre petite difficulté: avant 2004, on pouvait trouver l'acronyme "spca" pour "special pca", qui décrit une autre méthode datant de 1984. Depuis l'article de Zou, il semble néanmoins que "spca" se réfère uniquement aux situations sparses.

A.3.4 Avantage au niveau de la complexité?

La PCA usuelle implique une recherche de directions de variance maximale de l'ordre de $O(\min(p^3, n^3))$. Cependant, une méthode a priori pour sélectionner $k \leq \min(n, p)$ coordonnées permettrait de réduire à $O(k^3)$.

Zou discute de la complexité de son algorithme en distinguant deux cas (en réalité, on gagne surtout en vitesse de convergence):

a) $n > p$. Les données multivariées traditionnelles entrent dans cette catégorie. Le critère SPCA dépend uniquement de X. Une astuce consiste à calculer d'abord la matrice de covariance, ce qui nécessite np^2 opérations. Ensuite, la même matrice est utilisée à chaque étape dans la boucle. Les coûts de calcul restants sont de l'ordre de $O(pk^2)$ avec $k < p$. Chaque solution d'elastic net nécessite au plus $O(p^3)$ opérations. Le coût total de calcul est au plus $np^2 + mO(p^3)$, où m est le nombre d'itérations avant convergence.

b) Si p est très grand devant n, le calcul de la matrice de covariance est trop cher. Dès lors, on a au pire une complexité en $mO(pJ^2k)$, avec J le cardinal des composantes non nulles (le coût d'elastic net ne peut faire l'économie d'un ordre p). Ce qui peut poser un problème si on n'utilise pas l'accélération A.3.1.

Bibliographie

[1] Hui Zou, Trevor Hastie, & Robert Tibshirani (2006). *Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics*, 15:2, 265-286. DOI: 10.1198/106186006X113430.

[2] Hui Zou (2006). *Some Perspectives of Sparse Statistical Modeling*. PhD thesis. Disponible à l'adresse: https://hastie.su.domains/public/students/THESES/hui_zou.pdf.

[3] Hui Zou & Trevor Hastie (2005). *Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society Series B: Statistical Methodology*, **67**(2):301–320, April. Oxford University Press.

[4] Hui Zou & Lingzhou Xue, (2018). *A Selective Overview of Sparse Principal Component Analysis*, *Proceedings of the IEEE*. Disponible à l'adresse: https://ieeexplore.ieee.org/document/8412518.

[5] Trevor Hastie, Robert Tibshirani, & Jerome Friedman (2009). *The Elements of Statistical Learning*. Springer.

[6] D. M. Witten, R. Tibshirani, & T. Hastie (2009). *A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis*, *Biostatistics*, vol. 10, no. 3, pp. 515-534, Jul. doi: 10.1093/biostatistics/kxp008.

[7] Ian T. Jolliffe, N. T. Trendafilov, & M. Uddin, (2003). *A modified principal component technique based on the lasso*, *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pages 531–547.

[8] N. Benjamin Erichson & al., (2020). *Sparse Principal Component Analysis via Variable Projection*, *SIAM Journal on Applied Mathematics*, 80:2, 977-1002, DOI: 10.1137/18M1211350. Disponible à l'adresse: https://arxiv.org/abs/1804.00341.

[9] Christophe Giraud (2021). *Statistics in High Dimension: 2nd Edition*. Springer. La partie utilisée est mise en ligne par l'auteur: https://www.imo.universite-paris-saclay.fr/~christophe.giraud/Orsay/Bookv3.pdf.

[10] Wikipedia. (vu la dernière fois le 28/11/2023). *Sparse PCA*. Disponible à l'adresse: https://en.wikipedia.org/?curid=18566488.

[11] Shen, H., & Huang, J. Z. (2008). *Sparse principal component analysis via regularized low rank matrix approximation. Journal of Multivariate Analysis*, 99(6), 1015–1034.

[12] Fan Chen & Karl Rohe (2023). *A New Basis for Sparse Principal Component Analysis. Journal of Computational and Graphical Statistics*. DOI: 10.1080/10618600.2023.2256502.

[13] R. Guerra-Urzola, & al., (2021). *A Guide for Sparse PCA: Model Comparison and Applications*, *Psychometrika*, vol. 86, pages 893–919. Disponible à l'adresse: https://doi.org/10.1007/s11336-021-09773-2.