Methode de PCA parcimonieuse

Sofia HARROUCH

30 mars 2018

SPCA: Sparse Principal Component Analysis

Dans cette partie, on montre une deuxième approche pour l'obtention d'une ACP modifié et "spase". Cette méthode consiste à faire une analogie entre l'ACP et la regression ridge. Tout d'abord, on montrera que les composantes principales d'une ACP s'écrivent comme un problème d'optimisation d'une regression ridge. Pour ensuite rajouter la pénalité du Lasso afin d'obtenir un problème d'optimisation d'une regression sous la pénalité d'Elastic net ce qui va nous permettre d'avoir des composantes principales parcimonieuses.

Approche simple de regression pour l'ACP

On considère $Y_k = \alpha_k' X$ la kième composante principale obtenu à partir d'une ACP, qui est une combinaison linéaire des p variables intiales. En effet Ces coefficients de saturation α_k peuvent aussi être obtenu en faisant une regression multiple de la CP sur les p variables initiales ($Y_k = X\beta + \epsilon$). Ensuite, on étend cette regression à une regression ridge en ajoutant la pénalité Ridge dans le but de manipuler toute sorte de données (quand le nombre de variable est supérieur aux nombres de donnée $p > n$ et $\lambda = 0$ la regression multiple n'a pas une solution unique, quand $n > p$ et X n'est pas une matrice complète) et d'obtenir une solution unique comme l'ACP. Pour résumer, cette pénalité nous permet juste de reconstruire les composantes principales et non pas de les penaliser. Enfin, on rajoute la pénalité du Lasso L_1 pour penaliser les composantes et de les rendre parcimonieuses. Le problème d'optimisation devient: $\hat{\beta} = argmin_\beta ||Y_k - X\beta|| + \lambda||\beta||^2 + \lambda_1||\beta||_1 \quad (1.1)$, et la kième composante principale approximée est $X\hat{V}_k = X\frac{\hat{\beta}}{||\hat{\beta}||}$. Cette technique dépend essentiellement des résulats de l'ACP (car on applique l'ACP et on utilise l'équation (1.1) pour trouver une autre apprixmation approprié des CPs pour qu'elles soient parcimonieuses). On peut dire que cette approche n'est pas une alternative et par la suite pas véritable. Dans la sous partie qui suit, on montrera la même chose que précédement mais cette fois-ci sans l'utilisation du résultat de l'ACP.

Approche complexe de regression pour l'ACP

Soient X la matrice de données, et x_i la ième ligne de cette matrice, $A_{p*k} = [\alpha_1, \alpha_2, ..., \alpha_k]$ les k premières composantes principales, et $B_{p,k} = [\beta_1, \beta_2, ..., \beta_k]$ les k premières composantes principales estimées de tel sorte que les CPs soient parcimonieuses. Le théorème 1 suivant montre que le problème de l'ACP peut se transformer en un problème de regression.

Théorème 1 Pour tout $\lambda > 0$:

$$(\hat{A}, \hat{B}) = argmin_{A,B} \sum_{i=1}^{n} ||x_i - AB^T x_i||^2 + \lambda \sum_{j=1}^{k} ||\beta_j||^2$$

Sous la contrainte

$$A^T A = I_{k*k}$$

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

Ce théorème nous permet de bien voir la transformation du problème d'optimisation.

En fait,

$$\sum_{i=1}^{n} ||x_i - AB^T x_i||^2 = ||X - XAB^T||^2$$

$$= ||XA_\perp||^2 + ||XA - XB^T||^2$$

Comme A est orthonormal, $A_\perp$ est une matrice qualeconque ortonormal tel que $[A; A_\perp]$ est $p * p$ orthonormal. Soit A fixé, le problème de minimisation devient:

$$(A, \hat{B}) = argmin_B C_\lambda(A, \hat{B}) = argmin_{A,B} \sum_{j=1}^{k} ||X\alpha_j - X\beta_j||^2 + \lambda \sum_{j=1}^{k} ||\beta_j||^2$$

Ce qui est équivalent à resoudre k regressions ridges indépendantes. Ce qui donne $\hat{\beta} = (X^T X + \lambda I)^{-1} X^T XA$. En fait, si A correspond aux CPs ordinaire alors on sait effectivement que B est proportionnelle à V (ce qu'on a montré dans la première sous partie).

Aprés avoir montré l'analogie entre les deux méthodes, on rajoute la pénalité de lasso pour avoir la "sparsité" des coefficients de regression (de saturation). Le problème devient:

$$(\hat{A}, \hat{B}) = argmin_{A,B} \sum_{i=1}^{n} ||x_i - AB^T x_i||^2 + \lambda \sum_{j=1}^{k} ||\beta_j||^2 + \sum_{j=1}^{k} \lambda_{1,j} ||\beta_j||_1$$

Sous la contrainte:

$$A^T A = I_{k*k}$$

Tel que λ est utilisé pour les k composantes alors que λ_1, j utilisée pour la "spasité" est différente d'une composante à l'autre.

L'algorithme pour la résolution du problème d'optimisation

- On suppose que A est connu et égale au CPs de l'ACP, et $Y_j^* = X\alpha_j$ donc:

$$\hat{\beta}_j = argmin_{\beta_j} ||Y_j - X\beta_j||^2 + \lambda ||\beta_j||^2 + \lambda_{1,j} ||\beta_j||_1$$

$$\hat{\beta}_j = argmin_{\beta_j} (\alpha_j - \beta_j)^T X^T X (\alpha_j - \beta_j) + \lambda ||\beta_j||^2 + \lambda_1, j ||\beta_j||_1$$

-On suppose que B est connu et fixé, on ignore la pénalité qui est lié aux termes de B, et le problème de minimisation devient:

$$min_\beta \sum_{i=1}^{k} ||x_i - AB^T x_i||^2 = ||X - XBA^T||$$

Sous la contrainte:

$$A^T A = I_{k*k}$$

La solution est obtenu en utilisant le théorème 2; en calculant la décomposition en valeurs singilieures (SVD) $(X^T X)B = UDV^T$ et posant $\hat{A} = UV^T$.

Théorème 2 Soit M_{n*p} et N_{n*k} deux matrices, On considère le problème de minimisation suivant:

$$\hat{A} = argmin_A ||M - NA^T||$$

Sous la contrainte

$$A^T A = I_{k*k}$$

Soit la décomposition en valeurs singulière de $M^T N$ est UDV^T donc $\hat{A} = UV^T$.

Ceci étant expliqué, l'algorithme peut être résumer comme suit:

1. On commence par $A = V[1:k]$ les coefficients de saturation des K premières composantes principales.
2. Sachant $A = [\alpha_1, \alpha_2, ..., \alpha_k]$, On résout le problème pour $j = 1, 2, ..., k$

$$\beta_j = argmin_\beta (\alpha_j - \beta)^T X^T X (\alpha_j - \beta) + \lambda ||\beta||^2 + \lambda_1, j||\beta||_1$$

3. Pour une matrice $B = [\beta_1, \beta_2, ..., \beta_k]$ fixée, on calcule la décomposition de la matrice en valeur singulière (SVD) de $X^T XB = UDV^T$, par la suite on met A à jour, ce qui donne $A = UV^T$.
4. On répète les étapes 2-3 jusqu'à la convergence.
5. On normalise V, $\hat{V}_j = \frac{\beta_j}{||\beta_j||}$

Et les V_j sont les nouvelles composantes principales paricimonieuses.