PCA

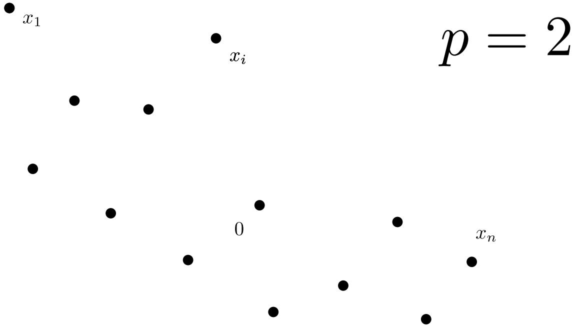
Nicolas Verzelen, Joseph Salmon

INRAE / Université de Montpellier

Plan

ACP

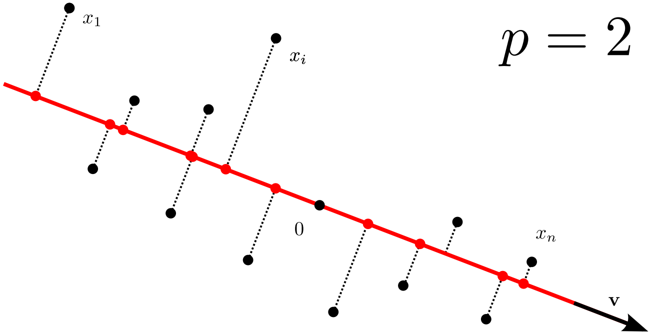
On observe n points $x_1, \ldots, x_n$ dans $\mathbb{R}^p$, ainsi on créé une matrice $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times p}$, n observations (lignes), p *features* (colonnes)



<u>Rem</u>: on doit recentrer les points pour qu'ils aient une moyenne nulle $X \leftarrow [x_1 - \overline{x}_n, \ldots, x_n - \overline{x}_n]^\top = X - \mathbf{1}_n \overline{x}_n^\top$ (on peut aussi mettre à l'échelle pour avoir un écart-type similaire par *feature*)

ACP

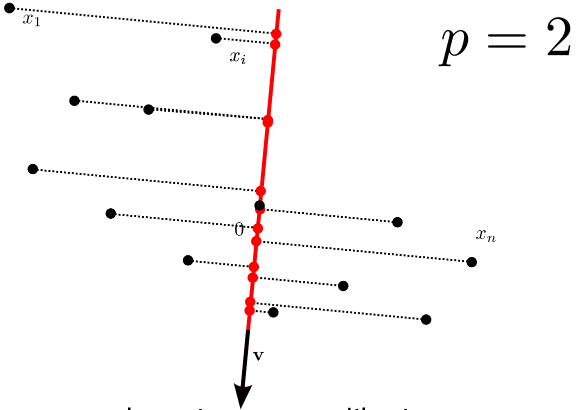
On observe n points $x_1, \ldots, x_n$ dans $\mathbb{R}^p$, ainsi on créé une matrice $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times p}$, n observations (lignes), p *features* (colonnes)



<u>Rem</u>: on doit recentrer les points pour qu'ils aient une moyenne nulle $X \leftarrow [x_1 - \overline{x}_n, \ldots, x_n - \overline{x}_n]^\top = X - \mathbf{1}_n \overline{x}_n^\top$ (on peut aussi mettre à l'échelle pour avoir un écart-type similaire par *feature*)

ACP

On observe n points $x_1, \ldots, x_n$ dans $\mathbb{R}^p$, ainsi on créé une matrice $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times p}$, n observations (lignes), p *features* (colonnes)



Rem: on doit recentrer les points pour qu'ils aient une moyenne nulle $X \leftarrow [x_1 - \overline{x}_n, \ldots, x_n - \overline{x}_n]^\top = X - \mathbf{1}_n \overline{x}_n^\top$ (on peut aussi mettre à l'échelle pour avoir un écart-type similaire par *feature*)

Analyse en Composante Principale, ACP (🇬🇧 : *Principal Component Analysis, PCA*)

Paramètre k : nombre d'axes pour représenter un nuage de n points $(x_1, \ldots, x_n)$, représentés par les lignes de $X \in \mathbb{R}^{n \times p}$.

Cette méthode **compresse** le nuage de points de dimension p en un nuage de dimension k

L'ACP (de niveau k) consiste à effectuer la SVD de X, et à ne garder que les k axes principaux pour représenter le nuage.

$$X = \sum_{i=1}^{r} s_i \mathbf{u}_i \mathbf{v}_i^\top \longrightarrow \sum_{i=1}^{k} s_i \mathbf{u}_i \mathbf{v}_i^\top$$

On appelle **axes principaux** les k vecteurs $\mathbf{v}_1, \ldots, \mathbf{v}_k$, et en général $k \ll p$ (*e.g.*, $k = 2$, pour une visualisation planaire)

Nouvelle représentation des données

- Les axes (de direction) $\mathbf{v}_1, \ldots, \mathbf{v}_p \in \mathbb{R}^p$ sont appelés **axes principaux** ou **axes factoriels**, les nouvelles variables $\mathbf{c}_j = X\mathbf{v}_j, j = 1, \ldots, p$ sont appelées **composantes principales**

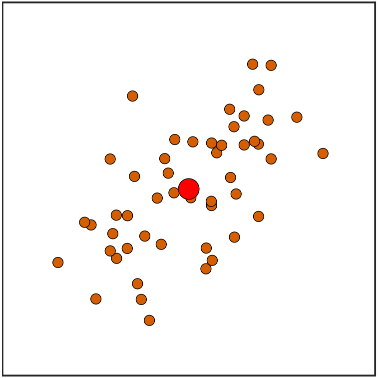
Nouvelle représentation (ordre k) :

- La matrice XV_k (avec $V_k = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$) est la matrice représentant les données dans la base des k premiers vecteurs propres

Reconstruction dans l'espace original (débruiter) :

- Reconstruction "parfaite" pour $\mathbf{x} \in \mathbb{R}^p$: $\mathbf{x} = \sum_{j=1}^{p} (\mathbf{x}^\top \mathbf{v}_j)\mathbf{v}_j$
- Reconstruction avec perte d'information : $\hat{\mathbf{x}} = \sum_{j=1}^{k} (\mathbf{x}^\top \mathbf{v}_j)\mathbf{v}_j$

Axe principal : maximisation de la variance

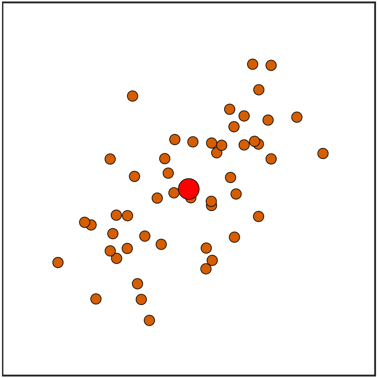


Data and mean

Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance

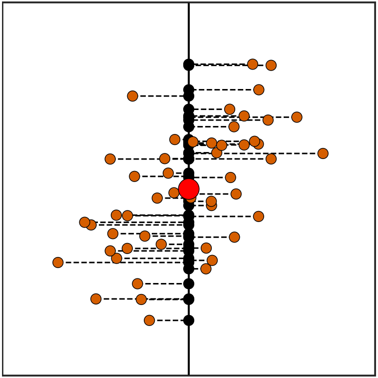


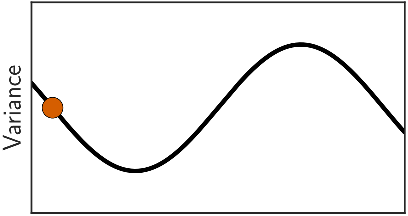
Data and mean

Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance

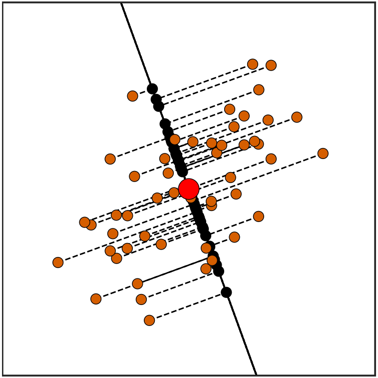




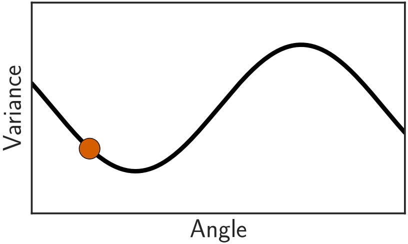
Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance



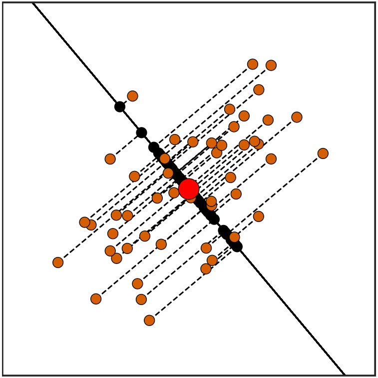
Data, mean and projection

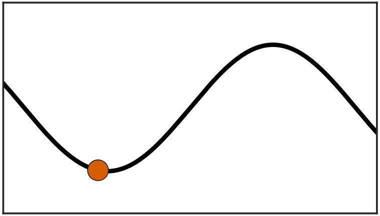


Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance

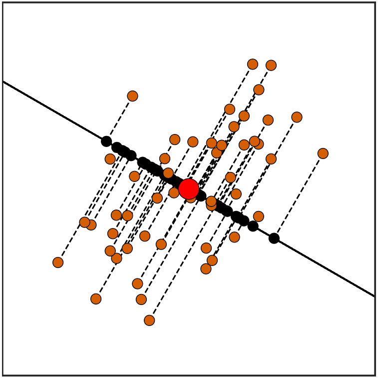

Data, mean and projection



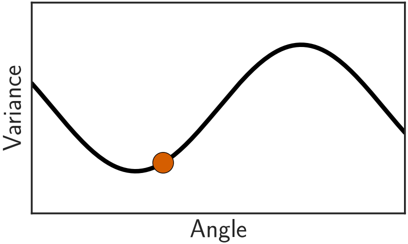
Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance



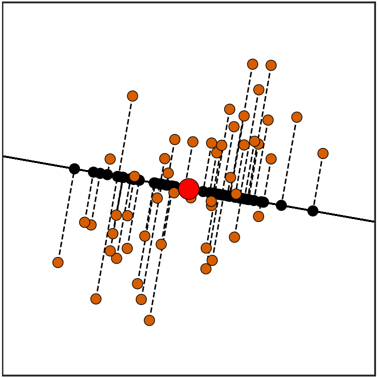
Data, mean and projection



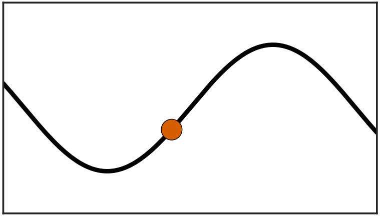
Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance



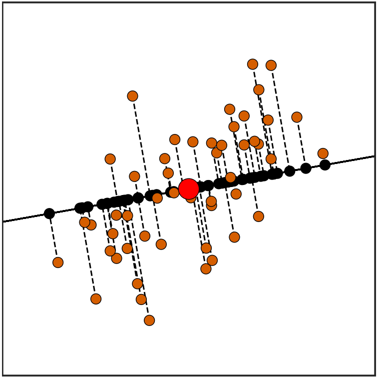
Data, mean and projection

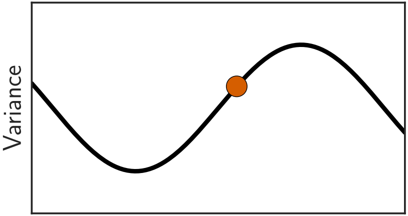


Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance

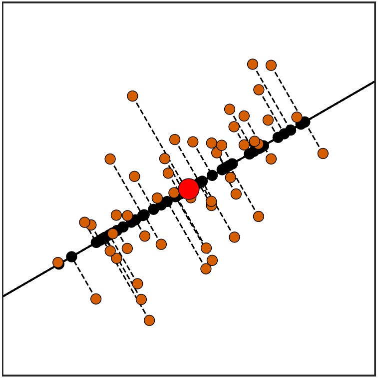




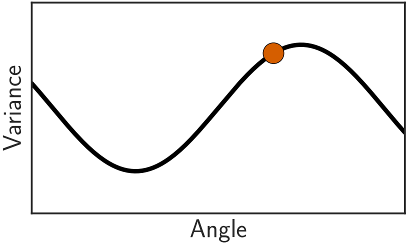
Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance



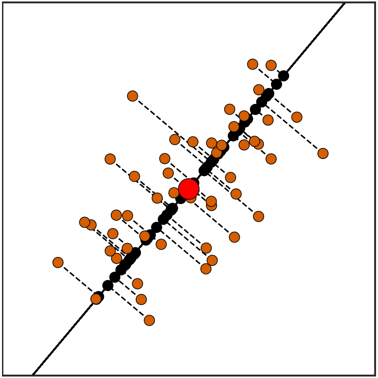
Data, mean and projection



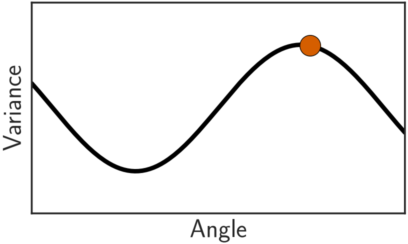
Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance



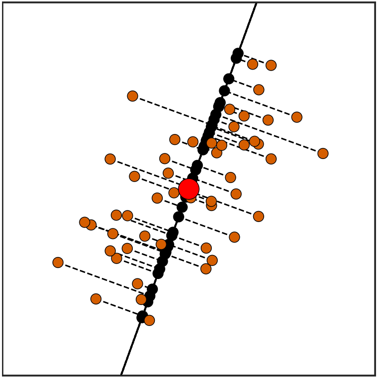
Data, mean and projection

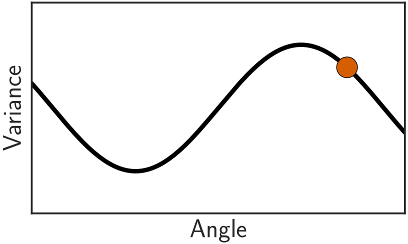


Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance

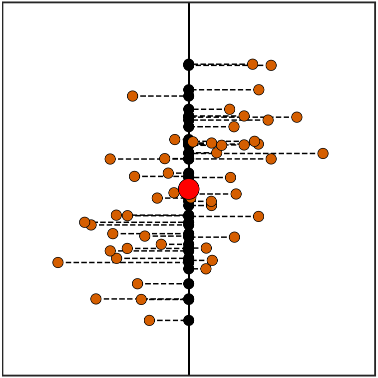


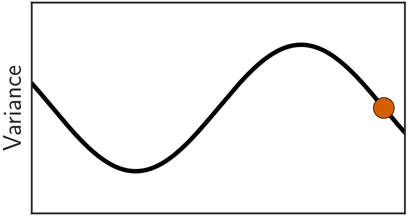


Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance

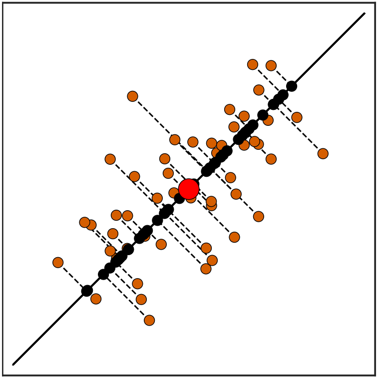




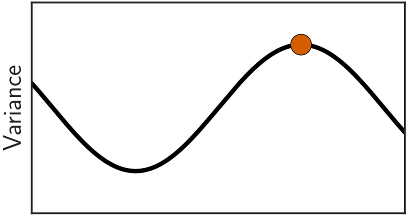
Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

Axe principal : maximisation de la variance



Principal direction (main axis)



Voir aussi la vidéo :

https://twitter.com/salmonjsph/status/1177564363261194240

ACP : axe principal

L'axe principal (normalisé) $\mathbf{v}_1$ est la solution du problème :

$$\mathbf{v}_1 \in \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max}\ \mathbf{v}^\top X^\top X \mathbf{v} = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \|X\mathbf{v}\|^2 = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \sum_{i=1}^n (x_i^\top \mathbf{v})^2$$

Rem: après recentrage le dernier terme est la variance du nuage de points projeté sur l'axe $\mathbf{v}$

Algorithme : Méthode de la puissance itérée

Entrées : $X \in \mathbb{R}^{n\times p}$, itérations K

Rem: on résout une maximisation sous contrainte convexe

ACP : axe principal

L'axe principal (normalisé) $\mathbf{v}_1$ est la solution du problème :

$$\mathbf{v}_1 \in \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \ \mathbf{v}^\top X^\top X \mathbf{v} = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \ \|X\mathbf{v}\|^2 = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \sum_{i=1}^n (x_i^\top \mathbf{v})^2$$

Rem: après recentrage le dernier terme est la variance du nuage de points projeté sur l'axe $\mathbf{v}$

Algorithme : Méthode de la puissance itérée

Entrées : $X \in \mathbb{R}^{n\times p}$, itérations K

$\mathbf{v}$ tiré aléatoirement dans $\mathbb{R}^{n\times p}$ (*e.g.*, $\varepsilon/\|\varepsilon\|$ avec ε gaussien)

Rem: on résout une maximisation sous contrainte convexe

ACP : axe principal

L'axe principal (normalisé) $\mathbf{v}_1$ est la solution du problème :

$$\mathbf{v}_1 \in \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max}\ \mathbf{v}^\top X^\top X \mathbf{v} = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \|X\mathbf{v}\|^2 = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \sum_{i=1}^{n} (x_i^\top \mathbf{v})^2$$

Rem: après recentrage le dernier terme est la variance du nuage de points projeté sur l'axe $\mathbf{v}$

Algorithme : Méthode de la puissance itérée

Entrées : $X \in \mathbb{R}^{n\times p}$, itérations K
$\mathbf{v}$ tiré aléatoirement dans $\mathbb{R}^{n\times p}$ (*e.g.*, $\varepsilon/\|\varepsilon\|$ avec ε gaussien)
pour $k = 1, \ldots, K$ **faire**

Rem: on résout une maximisation sous contrainte convexe

ACP : axe principal

L'axe principal (normalisé) $\mathbf{v}_1$ est la solution du problème :

$$\mathbf{v}_1 \in \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \ \mathbf{v}^\top X^\top X\mathbf{v} = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \|X\mathbf{v}\|^2 = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \sum_{i=1}^n (x_i^\top \mathbf{v})^2$$

Rem: après recentrage le dernier terme est la variance du nuage de points projeté sur l'axe $\mathbf{v}$

Algorithme : Méthode de la puissance itérée

Entrées : $X \in \mathbb{R}^{n\times p}$, itérations K

$\mathbf{v}$ tiré aléatoirement dans $\mathbb{R}^{n\times p}$ (*e.g.*, $\varepsilon/\|\varepsilon\|$ avec ε gaussien)

pour $k = 1, \dots, K$ **faire**

$\quad \mathbf{u} \leftarrow X\mathbf{v}$

Rem: on résout une maximisation sous contrainte convexe

ACP : axe principal

L'axe principal (normalisé) $\mathbf{v}_1$ est la solution du problème :

$$\mathbf{v}_1 \in \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \; \mathbf{v}^\top X^\top X \mathbf{v} = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \; \|X\mathbf{v}\|^2 = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \sum_{i=1}^n (x_i^\top \mathbf{v})^2$$

Rem: après recentrage le dernier terme est la variance du nuage de points projeté sur l'axe $\mathbf{v}$

Algorithme : Méthode de la puissance itérée

Entrées : $X \in \mathbb{R}^{n\times p}$, itérations K

$\mathbf{v}$ tiré aléatoirement dans $\mathbb{R}^{n\times p}$ (*e.g.*, $\varepsilon/\|\varepsilon\|$ avec ε gaussien)

pour $k = 1, \ldots, K$ **faire**

  $\mathbf{u} \leftarrow X\mathbf{v}$

  $\mathbf{v} \leftarrow X^\top \mathbf{u}$

Rem: on résout une maximisation sous contrainte convexe

ACP : axe principal

L'axe principal (normalisé) $\mathbf{v}_1$ est la solution du problème :

$$\mathbf{v}_1 \in \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \; \mathbf{v}^\top X^\top X \mathbf{v} = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \; \|X\mathbf{v}\|^2 = \underset{\mathbf{v}\in\mathbb{R}^p, \|\mathbf{v}\|=1}{\arg\max} \sum_{i=1}^{n} (x_i^\top \mathbf{v})^2$$

Rem: après recentrage le dernier terme est la variance du nuage de points projeté sur l'axe $\mathbf{v}$

Algorithme : Méthode de la puissance itérée

Entrées : $X \in \mathbb{R}^{n\times p}$, itérations K

$\mathbf{v}$ tiré aléatoirement dans $\mathbb{R}^{n\times p}$ (*e.g.*, $\varepsilon/\|\varepsilon\|$ avec ε gaussien)

pour $k = 1, \dots, K$ **faire**

$\quad \mathbf{u} \leftarrow X\mathbf{v}$

$\quad \mathbf{v} \leftarrow X^\top \mathbf{u}$

$\quad \mathbf{v} \leftarrow \frac{\mathbf{v}}{\|\mathbf{v}\|}$

Sorties : Axe principale (approché) $\mathbf{v}_1 = \mathbf{v}$

Rem: on résout une maximisation sous contrainte convexe

Premier axe principal

Maximiser la fonction objectif suivante en $\mathbf{v}$:
$\mathcal{L}(\mathbf{v}, \lambda) = (X\mathbf{v})^\top (X\mathbf{v}) - \lambda(\mathbf{v}^\top \mathbf{v} - 1) = \mathbf{v}^\top X^\top X\mathbf{v} - \lambda(\mathbf{v}^\top \mathbf{v} - 1)$

λ : multiplicateur de Lagrange
Conditions d'optimalité du premier ordre en un extremum

$$\frac{\partial \mathcal{L}(\mathbf{v}_1, \lambda)}{\partial \mathbf{v}} = 0 \Leftrightarrow X^\top X\mathbf{v}_1 = \lambda \mathbf{v}_1$$

La matrice de Gram $X^\top X$ est diagonalisable (symétrique) donc si $\mathbf{v}_1$ est un extremum alors c'est un vecteur propre.

Rem: on normalise $\mathbf{v}_1$ pour que $\|\mathbf{v}_1\| = 1$, ainsi $\lambda = \mathbf{v}_1^\top X^\top X\mathbf{v}_1$ et $\mathbf{v}_1$ est un vecteur propre, de valeur propre λ maximale

Aspect récursif de l'ACP/SVD - Déflation

Construction récursive : définir les axes principaux en partant du plus important et en descendant
Par récurrence, on définit le k^{e} axe pour qu'il soit orthogonal aux axes principaux précédents :

$$\mathbf{v}_k = \underset{\mathbf{v}\in\mathbb{R}^p,\, \mathbf{v}^\top\mathbf{v}_1=\cdots=\mathbf{v}^\top\mathbf{v}_{k-1}=0, \|\mathbf{v}\|=1}{\arg\max} \|X\mathbf{v}\|^2$$

- le premier axe maximise la variance des données projetées sur l'axe porté par ce vecteur
- le deuxième axe est celui orthogonal au premier, de variance projetée maximale
- etc.

Rem: numériquement il y a d'autres alternatives à la déflation

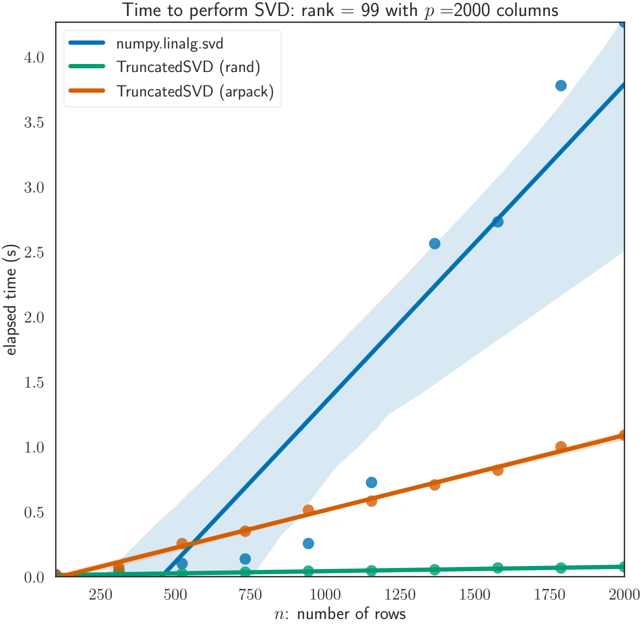
Autres méthodes numériques

- Algorithme de Lánczos / Espace de Krylov : utile quand plusieurs composantes / valeurs propres sont requises
- Itérations d'Arnoldi

cf. Golub et VanLoan (2013)

Rem: des techniques récentes ont permis des gains en rapidité en utilisant des méthodes aléatoires (*cf.* **sketching**), Halko *et al.* (2011)

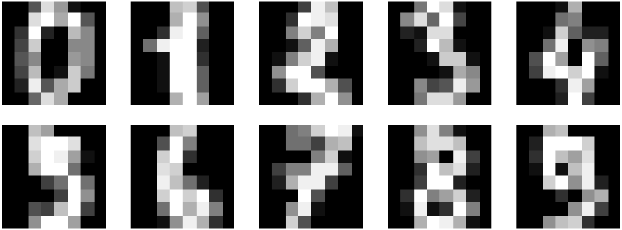
Temps de calcul pour quelques solveurs SVD



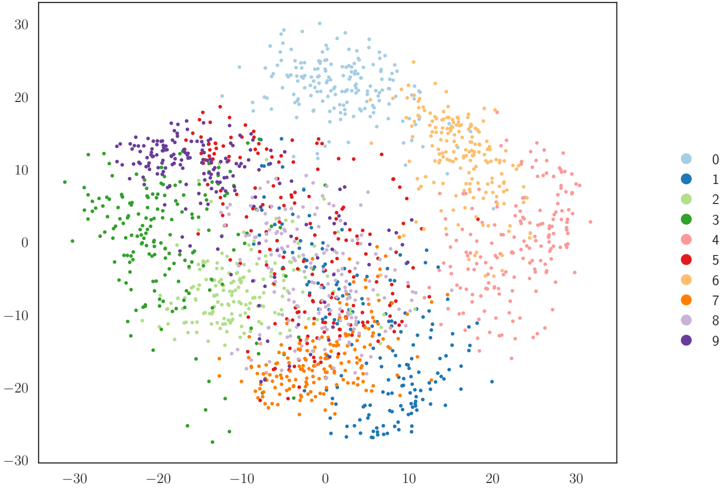
Alternatives à l'ACP

D'autres méthodes de réduction de dimension peuvent existent, *e.g.*, **t-SNE** (t-distributed Stochastic Neighbor Embedding)

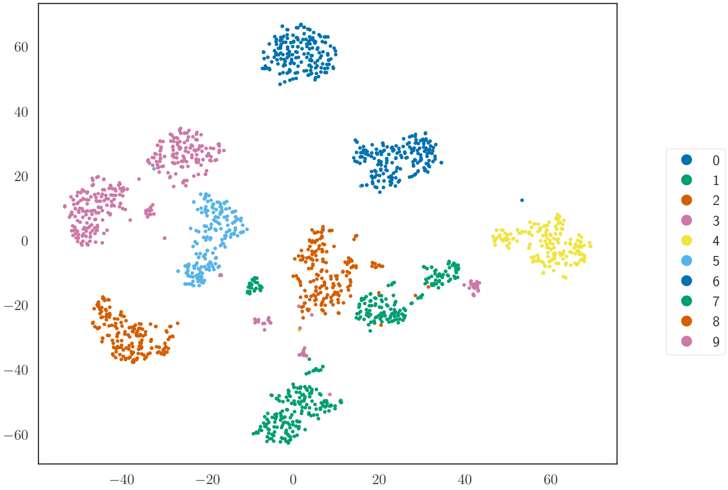
Exemple : dataset "digits" $X \in \mathbb{R}^{n \times p}$ avec $(n, p) = (1797, 64)$ (1797 chiffres numérisées d'image 8×8)



Exemple sur “digits” : PCA (2 axes)



Exemple sur "digits" : t-SNE (2 axes)



Références

- GOLUB, G. H. et C. F. VAN LOAN. *Matrix computations*. Fourth. Johns Hopkins University Press, Baltimore, MD, 2013, p. xiv+756.
- HALKO, N., P. MARTINSSON et J. A. TROPP. "Finding structure with randomness : Probabilistic algorithms for constructing approximate matrix decompositions". In : *SIAM Review* 53 (2011), p. 217.