IMN430

Chapitre 3
Analyse de données

Michaël Bernier & Olivier Godin

Université de Sherbrooke

31 janvier 2017

Plan de la présentation

Acquisition des données

Sources de données

Les sources de données sont multiples et produisent, selon le cas, **une quantité variable d'information à représenter**. À mesure que cette quantité augmente, les techniques traditionnelles de visualisation (nuage de points, histogramme, etc.) deviennent insuffisantes.

On verra dans ce chapitre différentes approches rendant possible la **visualisation d'un ensemble de données de grande taille**.

Sources de données

La provenance des données peut être répartie en trois catégories :

- **Monde réel** (données obtenues par observation)
- **Simulations théoriques** (données obtenues par modélisation)
- **Monde artificiel** (données obtenues par élaboration manuelle)

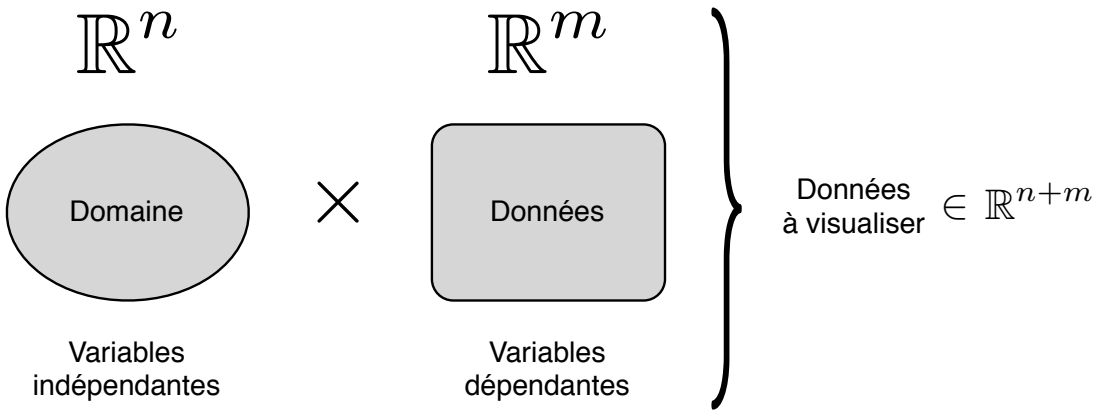
La quantité de données à représenter dépend essentiellement du **nombre de paramètres** d'intérêt et du **nombre de mesures** à considérer.

Sources de données

- **Monde réel**
  - Imagerie médicale, information géographique
  - Météorologie, données sismiques
  - Astronomie
- **Simulations théoriques**
  - Architecture, données financières
  - Dynamique des fluides, design automobile, modèles économiques
- **Monde artificiel**
  - Dessins, mise en page
  - Infographie télévisuelle
  - Films d'animation, effets spéciaux

Représentation des données

Les données à visualiser sont souvent représentées comme un sous-ensemble de l'espace $\mathbb{R}^{n+m}$ où n est le nombre de variables indépendantes et m est le nombre de variables dépendantes.



Si les données sont continues, il sera nécessaire de procéder à un échantillonnage. En effet, les techniques de visualisation considèrent que **l'information n'est disponible qu'à des positions discrètes dans l'espace**.

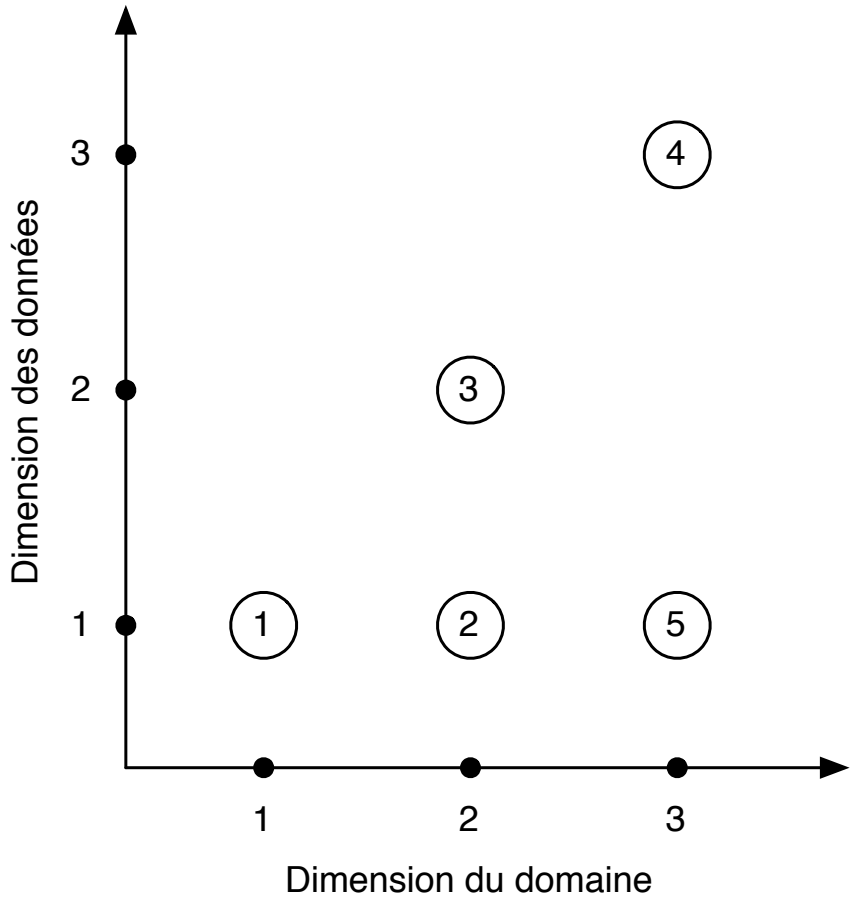
Représentation des données

Les techniques de visualisation à utiliser sont choisies selon **la dimension du domaine** (variable indépendante) et **la dimension des données** (variable dépendante).

Considérons les exemples suivants :

1. La température le long d'une tige en métal
2. L'altitude à la surface d'un continent
3. Un flot d'air 2D
4. Un flot d'air 3D
5. La concentration d'oxygène dans l'air

Représentation des données



Le domaine

L'espace où reposent les données (l'espace des variables indépendantes) est le **domaine** et les points du domaine où des données sont disponibles sont les **échantillons**.

En visualisation, on s'intéressera principalement à trois caractéristiques du domaine :

- sa dimension ;
- l'influence des données sur leur voisinage
- la connectivité entre les échantillons

Le domaine

Les variables indépendantes formant les dimensions du domaine peuvent être continues ou discrètes. Dans le cas des variables continues, il faudra procéder à un **échantillonnage**.

De plus, si le nombre de variables indépendantes est supérieur à deux, il faudra nécessairement effectuer une **projection** pour rendre possible la visualisation des données sur un écran 2D. Une projection pourra causer des problèmes d'**occultation** et ainsi créer des ambiguïtés dans le résultat visuel.

Le domaine

La valeur des échantillons peut aussi servir à **définir des données pour le reste du domaine**. Les échantillons pourront alors avoir trois types d'influence sur leur voisinage :

- **Influence ponctuelle** : seul l'échantillon courant est considéré pour attribuer les nouvelles valeurs
- **Influence locale** : un échantillon a un impact sur les données aux points du domaine situés dans le voisinage de celui-ci.
- **Influence globale** : chaque échantillon influence l'ensemble des autres points dans le domaine, peu importe la distance avec celui-ci.

Le domaine

Considérons dans un premier temps le cas de l'**influence ponctuelle**.

Pour obtenir des données à chaque point du domaine à partir d'un ensemble d'échantillons, on peut construire un **diagramme de Voronoï** à partir des échantillons.

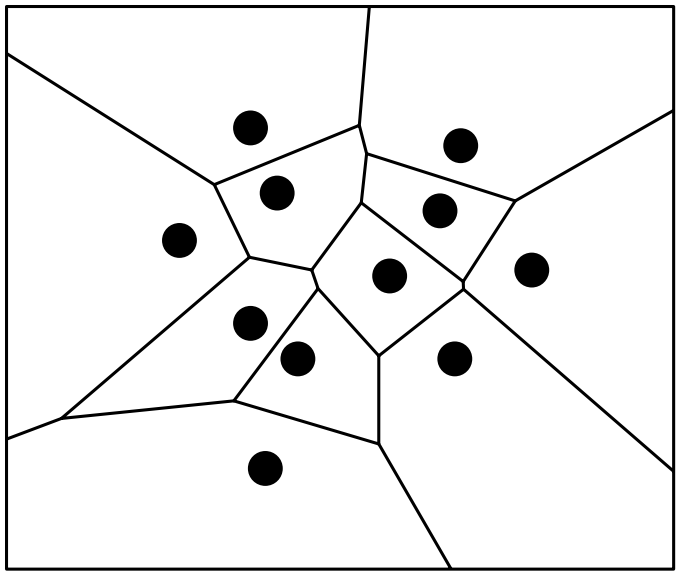
Le domaine

Un diagramme de Voronoï est **une décomposition du domaine en régions construites autour des échantillons**. Chaque région contient un et un seul échantillon et sa région peut être perçue comme étant la zone d'influence de celui-ci dans le domaine.

Une région contient l'ensemble des points du domaine qui sont plus prêts d'un échantillon que de tous les autres.

Le domaine

Il s'agit donc d'une partition de l'espace $\mathbb{R}^n$ en régions R_i où chaque point du domaine dans la région R_i se verra assigner la valeur de la donnée à l'échantillon i.



[2]

Le domaine

Pour les problèmes d'**influences locale et globale**, on utilisera l'interpolation afin d'obtenir des valeurs à chaque point du domaine. On définira donc une fonction ayant les particularités suivantes :

1. **Aux échantillons**, la fonction retournera exactement la valeur de la donnée connue.
2. **Aux autres points du domaine**, la fonction retournera une moyenne pondérée des valeurs des échantillons.

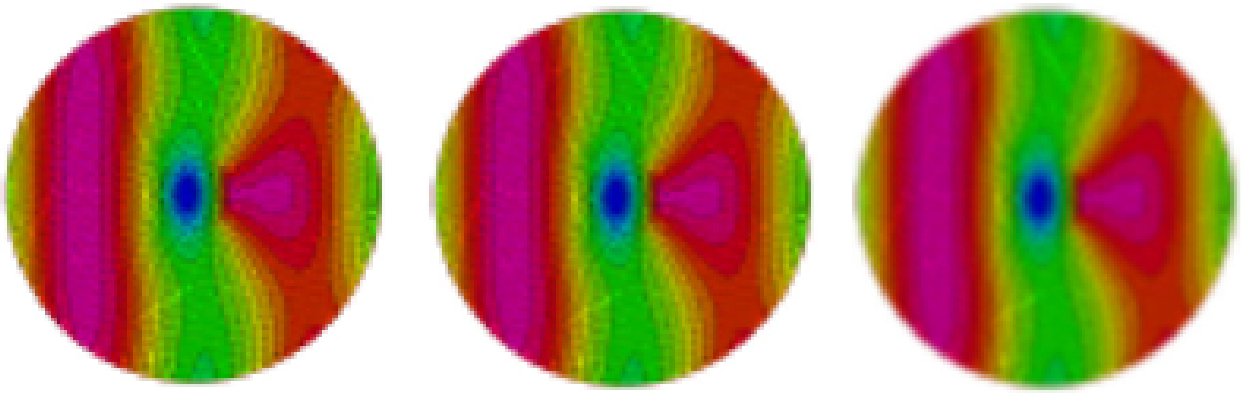
Le domaine

La contribution de chaque échantillon dans le calcul de la moyenne dépendra d'une **fonction de poids**. Habituellement, celle-ci sera **décroissante à mesure que la distance par rapport à un échantillon augmente**.

Dans le cas de l'influence locale, si la distance entre un point quelconque du domaine et un échantillon est trop grande, **la fonction de poids sera nulle**.

Le domaine

Visuellement, une augmentation de la largeur du voisinage à considérer aura **un effet de flou sur les données interpolées**.



[9]

Les données

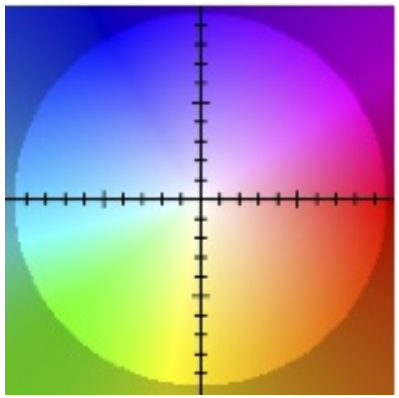
Tout comme pour les points du domaine, on s'intéresse aux caractéristiques des données. Les principales sont

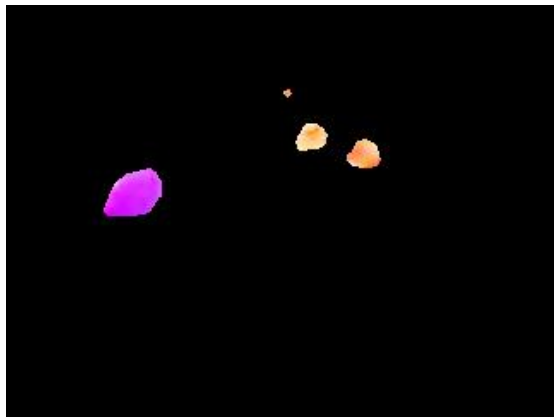
- le type et la dimension des données
- l'étendue des valeurs possibles

Les types de données habituels sont les **scalaires**, les **vecteurs** et les **tenseurs**.

Les données

Idéalement, **chaque échantillon n'est représenté que par une seule valeur**. Cette situation est plus facilement visualisable. Si plusieurs valeurs sont disponibles aux échantillons (données vectorielles ou tensorielles), il devient difficile de trouver une bonne visualisation.





La suite de ce chapitre traitera du **prétraitement** des données à haute dimension pour en permettre l'affichage.

Les données

Les données peuvent être **qualitatives** ou **quantitatives**.

Les **données qualitatives** peuvent être triées (selon un ordre nominal), mais ne permettent pas d'établir une distance entre les mesures. Il est seulement possible de vérifier si deux valeurs sont égales.

Les **données quantitatives** peuvent quant à elles être triées selon leur valeur numérique. Elles permettent aussi de définir une distance entre les mesures de même qu'en déterminer l'étendue. Elles peuvent être continues ou discrètes.

Données à haute dimension

L'augmentation de la puissance de calcul et de l'espace de stockage des ordinateurs a mené à une augmentation de la taille des ensembles de données.

Ainsi, plusieurs champs des sciences reposent maintenant sur notre capacité à analyser et visualiser des données à haute dimension.

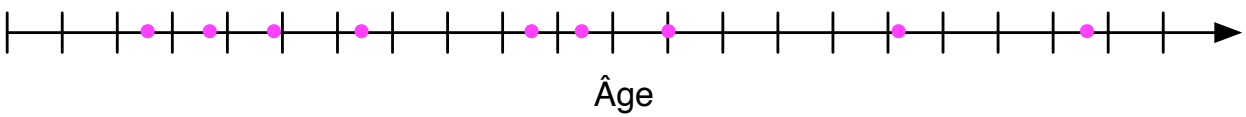
Données à haute dimension

On fait alors face à deux problèmes :

1. **La visualisation des données devient sujette à des ambiguïtés** si le nombre de dimensions dépasse deux.
2. À mesure que le nombre de dimensions augmente, **les données deviennent de plus en plus éparpillées dans l'espace**.

Données à haute dimension

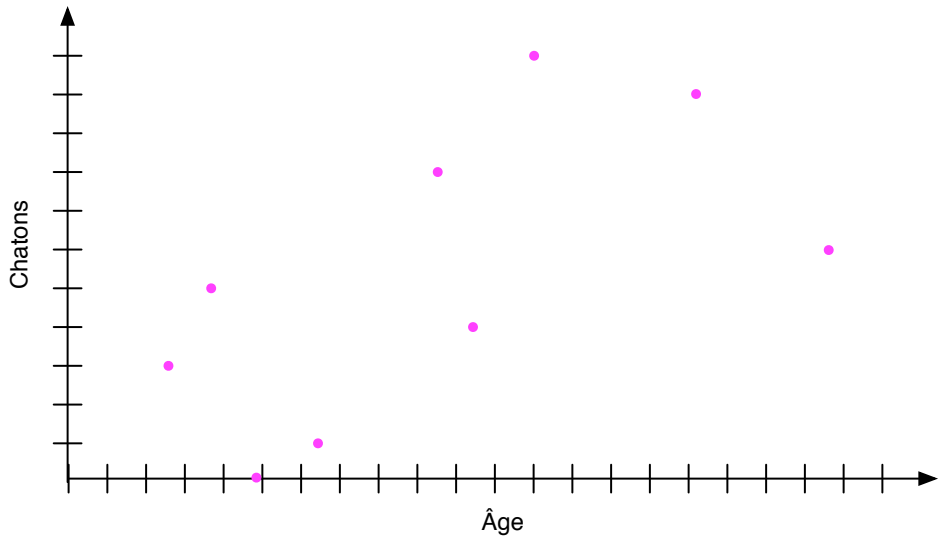
Considérons l'exemple suivant où l'on s'intéresse à **l'âge des gens dans une population**.



La représentation graphique **ne présente aucune ambiguïté** puisqu'il est possible de représenter des données 1D sur une surface 2D.

Données à haute dimension

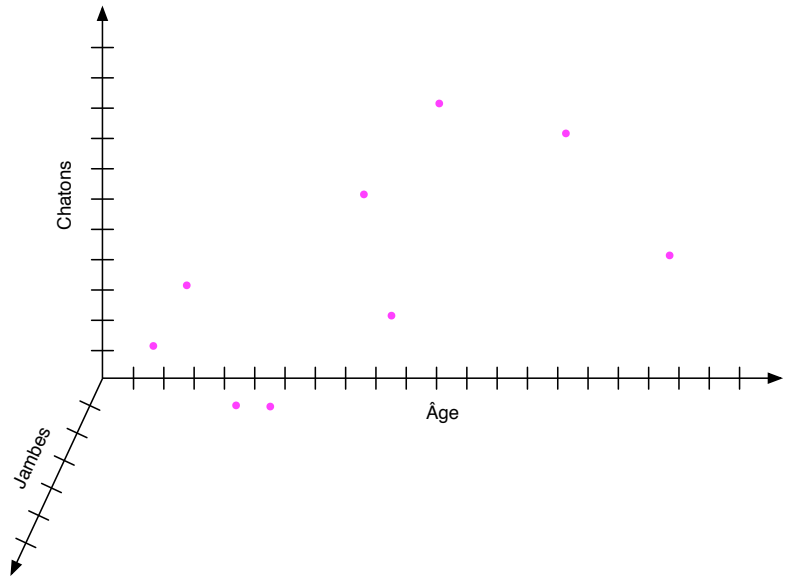
Ajoutons une deuxième variable : **le nombre de chatons des gens dans la population**.



Le graphique **ne laisse à nouveau aucune place au doute**. Les données ont deux dimensions et on les représente sur une surface 2D.

Données à haute dimension

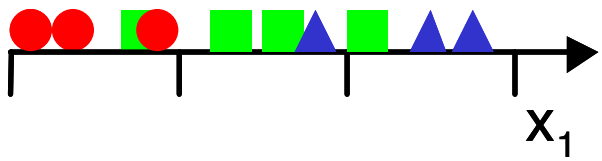
Un problème survient si on ajoute une troisième dimension : **le nombre de jambes des gens dans la population**.



Il devient alors ardu de connaître la position réelle des points dans l'espace 3D en raison d'une mauvaise projection sur la surface de visualisation 2D.

Données à haute dimension

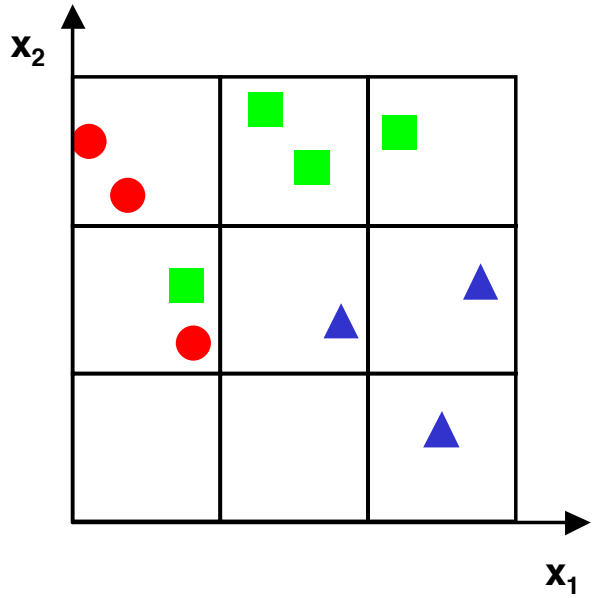
Intéressons-nous maintenant à une autre situation : on dispose de **10 observations 1D** (le long d'un axe).



Si on découpe l'axe en trois régions, on a **une moyenne de 3.3 données par région**.

Données à haute dimension

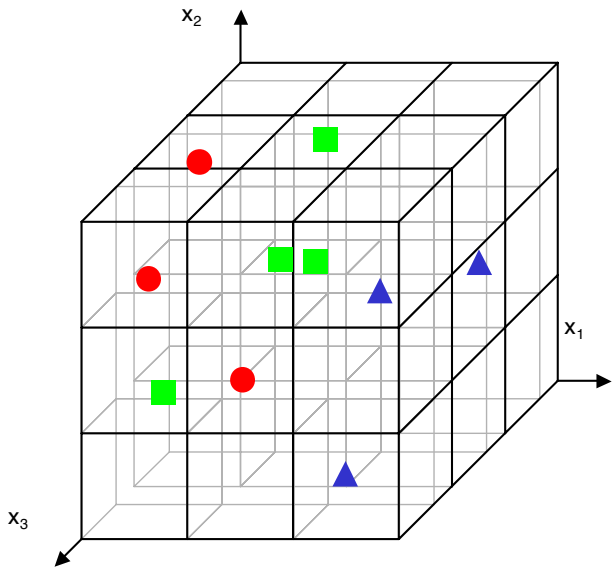
Si on conserve le même nombre d'échantillons, mais cette fois-ci **réparti dans un espace 2D** découpé en 9 régions, on obtient



On constate que **les données sont beaucoup plus éparpillées dans l'espace** et présentent une moyenne de 1.1 données par région.

Données à haute dimension

En conservant le même nombre d'échantillons, si on passe à un espace 3D séparé en 27 régions, **on constate un éparpillement encore plus prononcé** avec une moyenne de 0.37 donnée par région.



Ainsi, moins la densité des données est grande dans le domaine, moins la représentation graphique sera utile et pertinente.

Réduction de la dimension

La solution à ces deux problèmes est alors simple : il faut **trouver une représentation plus compacte** pour les données à haute dimension afin d'en faciliter l'affichage.

Mathématiquement, cela revient à trouver, pour une variable aléatoire de dimension p, $\mathbf{x} = (x_1, \ldots, x_p)$, une représentation de dimension inférieure $\mathbf{s} = (s_1, \ldots, s_k)$ (avec $k < p$) **qui conserve le plus possible l'information des données originales**.

Réduction de la dimension

Deux approches existent pour réduire la dimension d'un espace $\mathbb{R}^p$ à un espace $\mathbb{R}^k$:

- **La sélection des axes** : un sous ensemble de taille k des axes originaux est conservé, les autres sont éliminés :

$$(x_1, x_2, \ldots, x_p) \xrightarrow{\text{sélection}} (x_{i_1}, x_{i_2}, \ldots, x_{i_k})$$

- **L'extraction de caractéristiques** : k nouveaux axes sont créés à partir des axes originaux :

$$(x_1, x_2, \ldots, x_p) \xrightarrow{\text{extraction}} (y_1, y_2, \ldots, y_k)$$

Réduction de la dimension

L'approche par sélection des axes est **simple, mais un peu barbare**. Au prix d'un peu plus de travail, on peut obtenir de bien meilleurs résultats avec l'extraction de caractéristiques.

L'objectif de l'extraction de caractéristiques et de représenter le plus fidèlement possible les échantillons dans un espace de plus faible dimension. On s'intéressera à deux familles de méthodes : les **méthodes linéaires** et les **méthodes non linéaires**.

Réduction de la dimension : méthodes linéaires

Réduction de la dimension : méthodes linéaires

Analyse en composantes principales

L'analyse en composantes principales (ACP) est une méthode d'analyse des données qui consiste à **transformer des variables corrélées en nouvelles variables décorrélées** les unes des autres. Ces nouvelles variables sont nommées **composantes principales**, ou axes principaux.

Elle permet de **réduire l'information en un nombre de composantes plus limité** que le nombre initial de variables.

Rappels mathématiques

Soit $X_1 = \{x_{1_1}, x_{1_2}, \ldots, x_{1_n}\}$ avec $x_{1_i} \in \mathbb{R}$, un ensemble de données.
On définit trois mesures statistiques simples :

- la **moyenne** : $\bar{X_1} = \frac{\sum_{i=1}^{n} x_{1_i}}{n}$;
- l'**écart type** : $s_{X_1} = \sqrt{\frac{\sum_{i=1}^{n}(x_{1_i} - \bar{X_1})^2}{n-1}}$
- la **variance** : $s_{X_1}^2 = \frac{\sum_{i=1}^{n}(x_{1_i} - \bar{X_1})^2}{n-1}$

Rappels mathématiques

Les trois mesures précédentes étaient **calculées à partir d'un seul ensemble de données**. Imaginons qu'on dispose d'une deuxième série de données sur la même population, notée X_2. On se retrouve ainsi avec **des données à deux dimensions**.

Pour voir s'il existe une relation entre X_1 et X_2, on définit le concept de **covariance** :

$$\text{cov}(X_1, X_2) = \frac{\sum_{i=1}^{n}(x_{1_i} - \bar{X}_1)(x_{2_i} - \bar{X}_2)}{n-1}$$

Rappels mathématiques

La covariance se calcule toujours **entre deux dimensions**. Si on dispose de données à n dimensions (avec $n > 2$), on doit généraliser le concept. Celui-ci devient **la matrice de variance-covariance** :

$$C = \begin{bmatrix} \text{var}(X_1) & \text{cov}(X_1, X_2) & \cdots & \text{cov}(X_1, X_n) \\ \text{cov}(X_2, X_1) & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \cdots & \cdots & \text{var}(X_n) \end{bmatrix}$$

Rappels mathématiques

Seules les matrices carrées peuvent posséder des vecteurs propres, mais toutes n'en ont pas nécessairement. Si une matrice $n \times n$ possède des vecteurs propres, alors elle en possèdera exactement n.

Finalement, une propriété cruciale pour la suite des choses et que **tous les vecteurs propres associés à une matrice sont orthogonaux**.

Il sera possible d'**exprimer les données dans un repère dont les axes seront définis par les vecteurs propres**, plutôt que dans le repère cartésien.

Principe de fonctionnement

Considérons le problème suivant : on souhaite **représenter tous les points** de dimension d d'un ensemble $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ **par un seul point** $\mathbf{x}_0$.

Plus spécifiquement, on cherche le vecteur $\mathbf{x}_0$ qui **minimise la somme du carré des distances**, notée $J_0(\mathbf{x}_0)$, entre $\mathbf{x}_0$ et tous les autres vecteurs $\mathbf{x}_k$, où

$$J_0(\mathbf{x}_0) = \sum_{k=1}^{n} |\mathbf{x}_0 - \mathbf{x}_k|^2 .$$

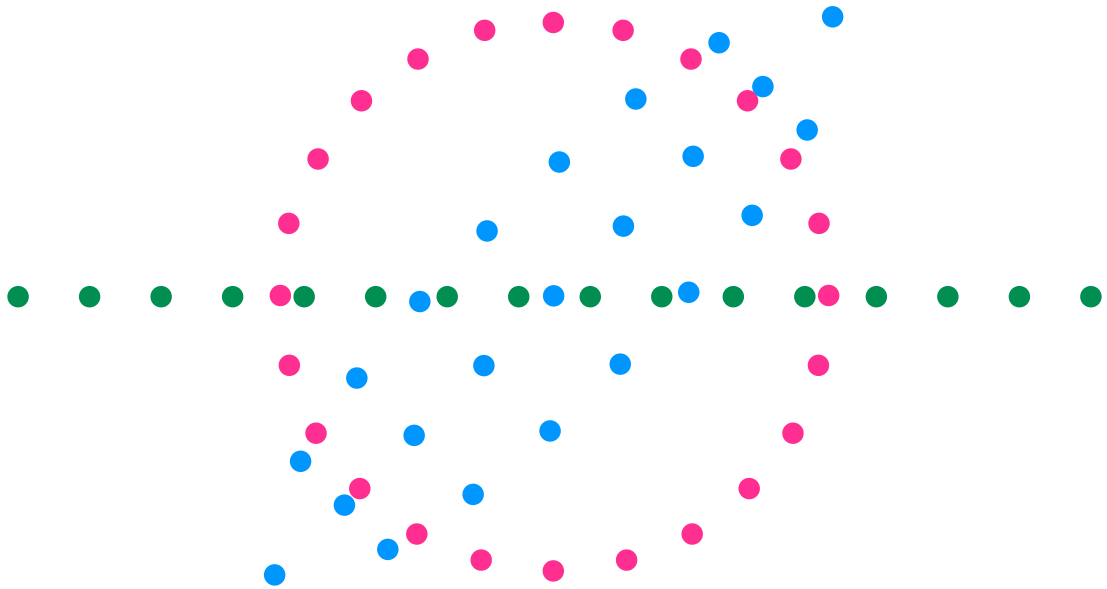
Principe de fonctionnement

On peut facilement vérifier que $J_0(\mathbf{x}_0)$ est minimal lorsque $\mathbf{x}_0 = \mathbf{m}$, où $\mathbf{m}$ est le **vecteur moyen de l'ensemble** $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$:

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k.$$

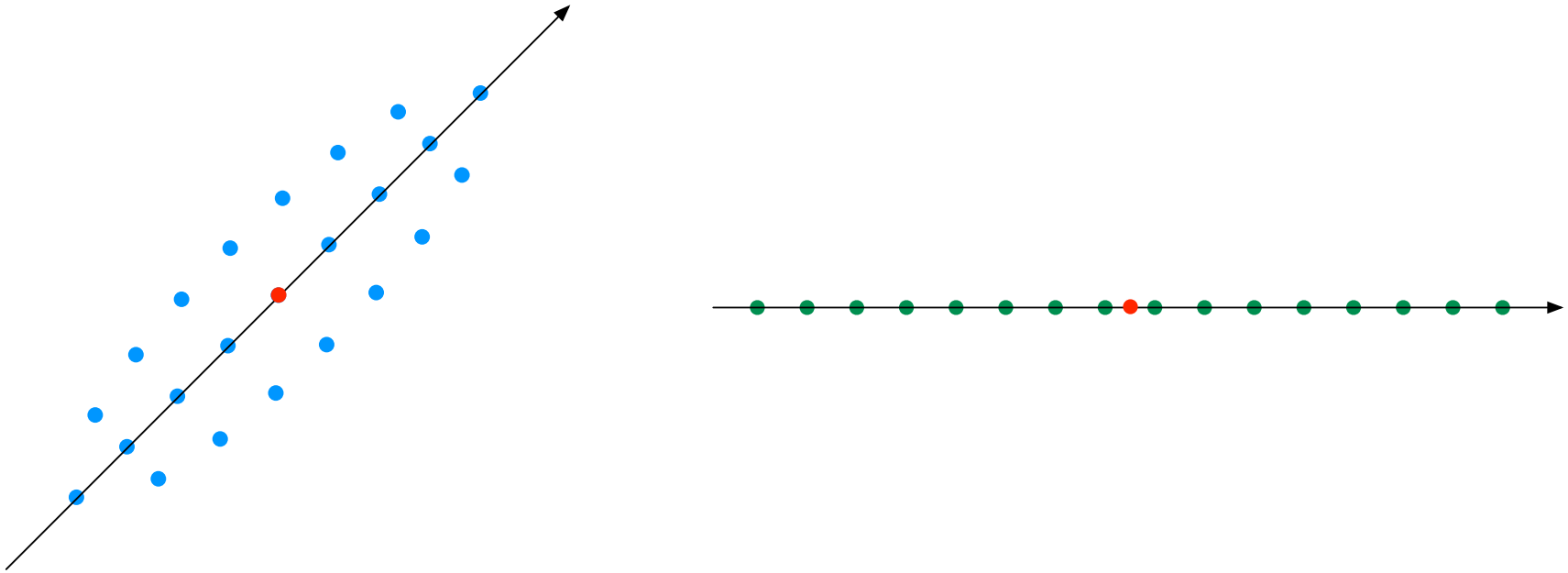
Principe de fonctionnement

Se limiter à la moyenne (un vecteur dans $\mathbb{R}^d$) pour représenter les données **n'illustre toutefois pas la variabilité de celles-ci**.



Principe de fonctionnement

Une représentation beaucoup plus intéressante peut être obtenue en **projetant les données sur une droite** passant par la moyenne **m**.



Principe de fonctionnement

Soit **e** le **vecteur directeur** de cette droite. L'équation de celle-ci est donc $x = \mathbf{m} + a \cdot \mathbf{e}$, avec $a \in \mathbb{R}$.

Pour un vecteur directeur **e** donné, on peut **trouver une valeur** a_k **pour chaque point** $\mathbf{x}_k$ **de l'ensemble** qui minimise la distance entre $\mathbf{x}_k$ et la droite. On note par $\{a_1, \ldots a_n\}$ l'ensemble des valeurs a_k.

Principe de fonctionnement

Cet ensemble optimal de coefficients sera obtenu en minimisant la somme du carré des distances $J_1(a_1, \ldots, a_n, \mathbf{e})$:

$$\begin{aligned} J_1(a_1, \ldots, a_n, \mathbf{e}) &= \sum_{k=1}^{n} |(\mathbf{m} + a_k \cdot \mathbf{e}) - \mathbf{x}_k|^2 \\ &= \sum_{k=1}^{n} |a_k \mathbf{e} - (\mathbf{x_k} - \mathbf{m})|^2 \\ &= \sum_{k=1}^{n} a_k^2 |\mathbf{e}|^2 - 2 \sum_{k=1}^{n} a_k \mathbf{e}^t (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{n} |\mathbf{x}_k - \mathbf{m}|^2 \end{aligned}$$

Principe de fonctionnement

Partant du principe que $|\mathbf{e}| = 1$, on obtient par dérivation partielle que

$$a_k = \mathbf{e}^t(\mathbf{x}_k - \mathbf{m}).$$

Cela nous amène à l'autre problème qui est de **trouver la meilleure direction pour la droite**, et donc le vecteur directeur **e**. Cette direction est obtenue grâce à la **matrice de dispersion S**, qui s'apparente beaucoup à la **matrice de variance-covariance** vue précédemment :

$$\mathbf{S} = \frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t.$$

Principe de fonctionnement

En effet, en substituant les valeurs optimales de a_k dans la mesure J_1, on obtient

$$\begin{aligned} J_1(\mathbf{e}) &= \sum_{k=1}^{n} a_k^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n} |\mathbf{x}_k - \mathbf{m}|^2 \\ &= -\mathbf{e}^t\mathbf{S}\mathbf{e} + |\mathbf{x}_k - \mathbf{m}|^2 \end{aligned}$$

Ainsi, le vecteur $\mathbf{e}$ minimisant J_1 maximise aussi $\mathbf{e}^t\mathbf{S}\mathbf{e}$.

Principe de fonctionnement

En maximisant $\mathbf{e}^t\mathbf{S}\mathbf{e}$, on trouve que $\mathbf{e}$ doit être **un vecteur propre de la matrice de dispersion**. En particulier, on choisira le vecteur propre **correspondant à la plus grande valeur propre de S**.

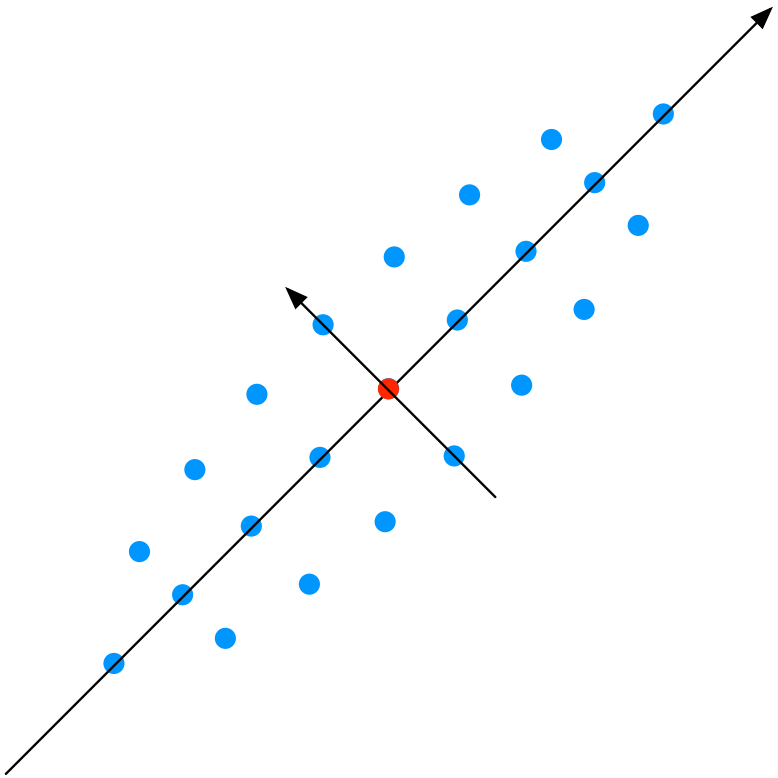
En d'autres mots, pour trouver **la meilleure représentation 1D** d'un ensemble de points dans $\mathbb{R}^d$, il suffit de les projeter sur une droite **passant par le point moyen** et **ayant comme direction le premier vecteur propre de S**.

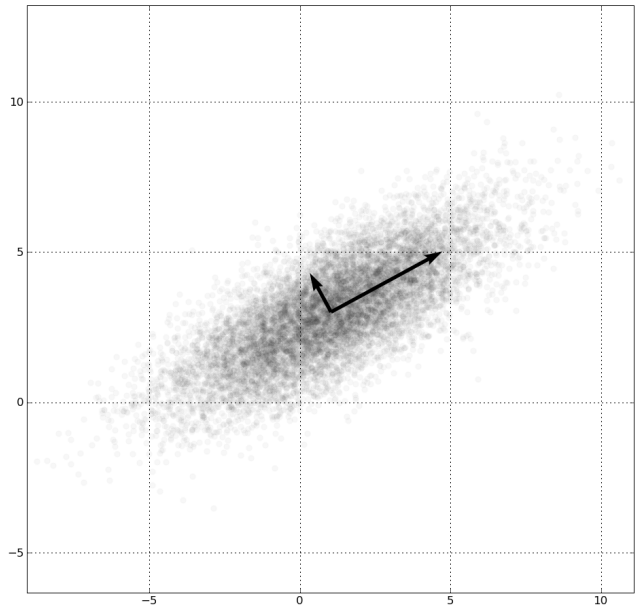
Principe de fonctionnement

Ce résultat peut être **généralisé à une projection à d' dimensions** au lieu d'une seule. La meilleure projection sera obtenue sur d' axes définis par les d' vecteurs propres de **S** associés aux d' plus grandes valeurs propres.

Ce changement de repère est possible en raison de l'**orthogonalité des vecteurs propres**.

Principe de fonctionnement





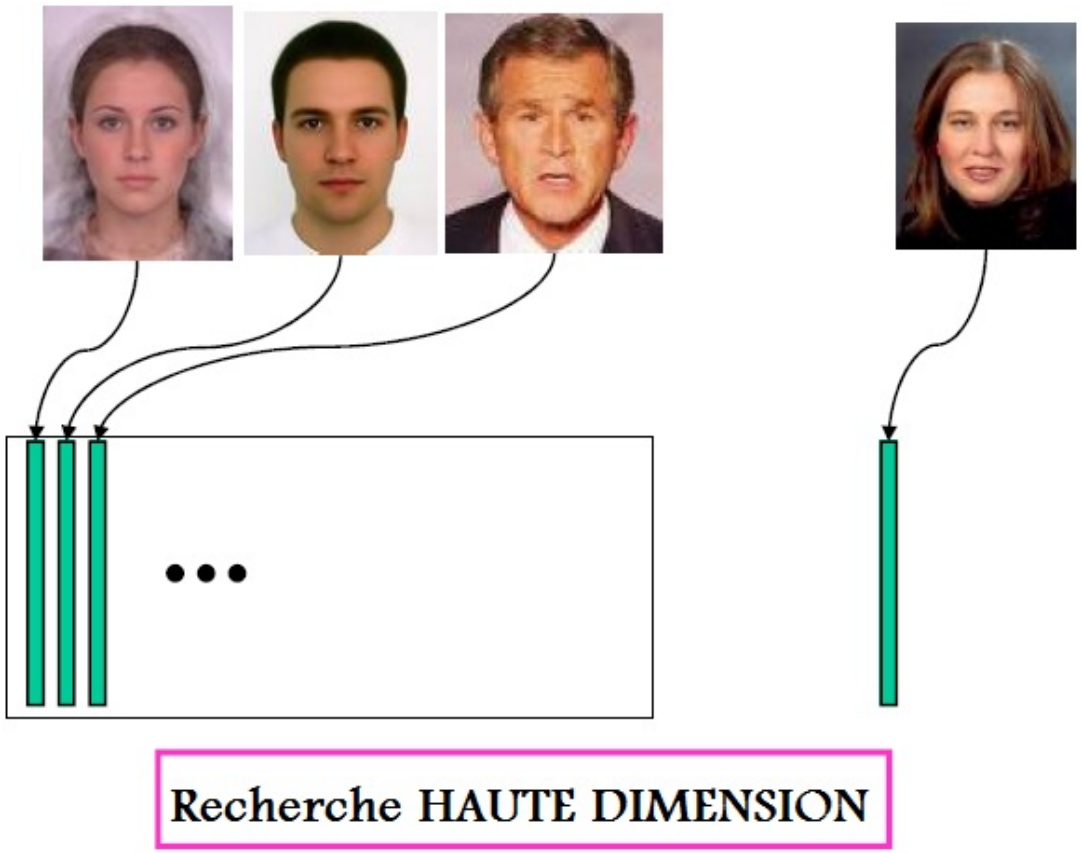
Principe de fonctionnement

L'analyse en composantes principales réduit la dimension d'un espace en trouvant **les directions selon lesquelles les données sont les plus dispersées**.

Les vecteurs formant la base du nouveau repère sont appelés les **composantes principales**.

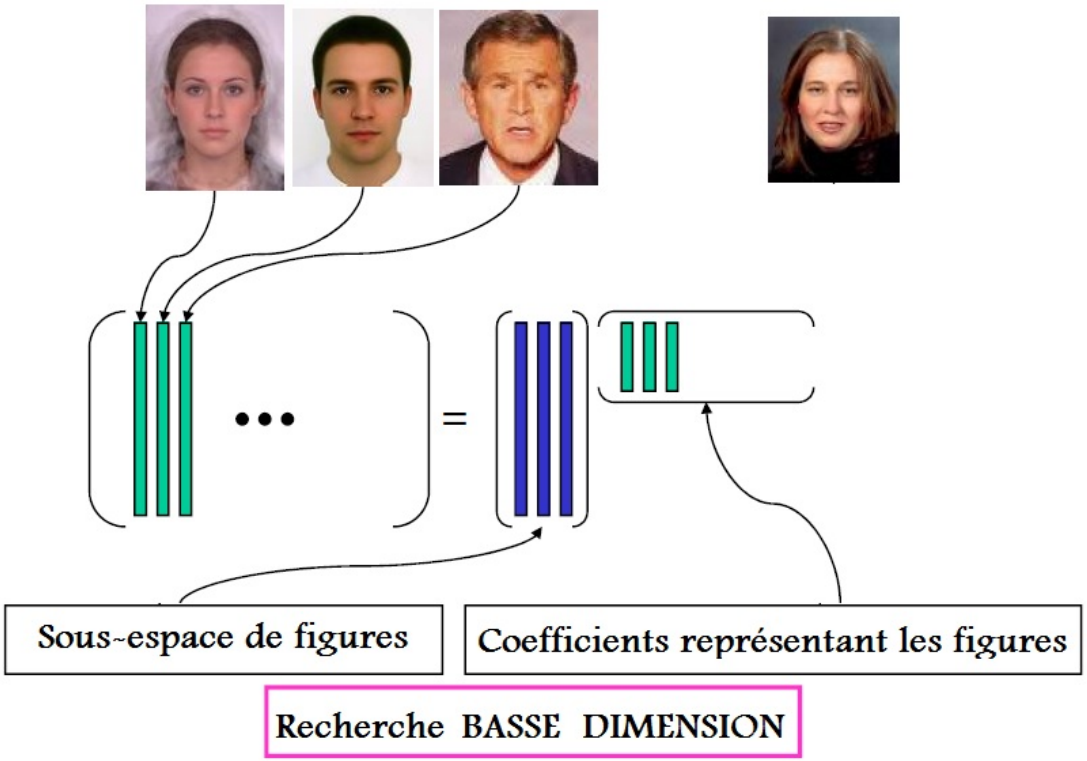
Exemple : Figures propres

Supposons que nous avons une base de donnée d'images (ex : des images de visage). On cherche à classifier un objets **en se basant sur cette base d'images** :



Exemple : Figures propres

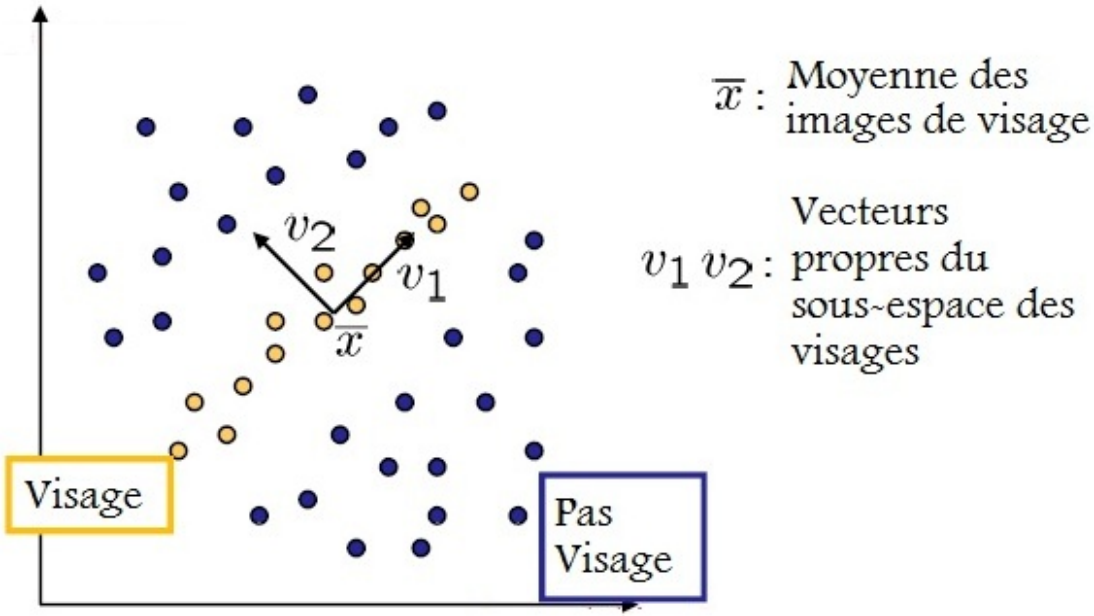
Une recherche exhaustive parmi cette base de donnée serait trop longue. On voudrait **réduire l'espace de recherche** :



Nous **exprimerons notre requête** dans une **sous-espace** de la base d'image.

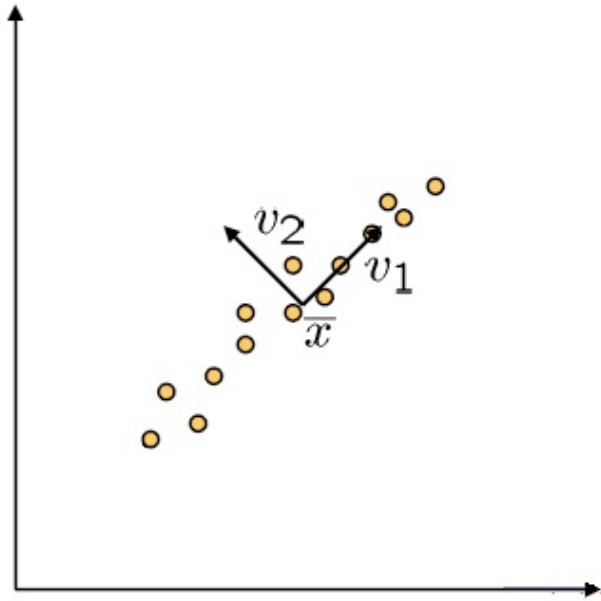
Exemple : Figures propres

Imaginons un très grand ensemble d'images de *deux pixels (dimensions)* **sans lien sémantique** (bleu). Supposons que nous voulons effectuer une recherche de visage. Nous ne choisirons que les images de visage (fort lien sémantique) :



Exemple : Figures propres

L'**ACP** nous permet d'effectuer un changement de base représentatif des données en utilisant les **valeurs et vecteurs propres** de l'ensemble de donnée.



La **force du lien** entre les images nous permet d'**abandonner** la composante (dimension) faible (v_2).

- Moins de données à traiter ;
- Très efficace dans un cas à N dimensions.

Exemple : Figures propres

On cherche la base orthonormée **v** représentant le mieux les données :

$$\begin{aligned} \mathbf{S} &= \frac{1}{n}\sum_{k=1}^{n}((x_0 - x_k)\cdot(x_0 - x_k)) \\ &= \frac{AA^T}{N} \end{aligned} \tag{1}$$

On construit une matrice contenant toutes les images
$x_k \leftarrow$ Les n images x_k, $k \in 1, N$, (vecteurs colonne de p pixels);

On calcule l'image moyenne de toutes les images
$x_0 \leftarrow$ Moyenne des n images;

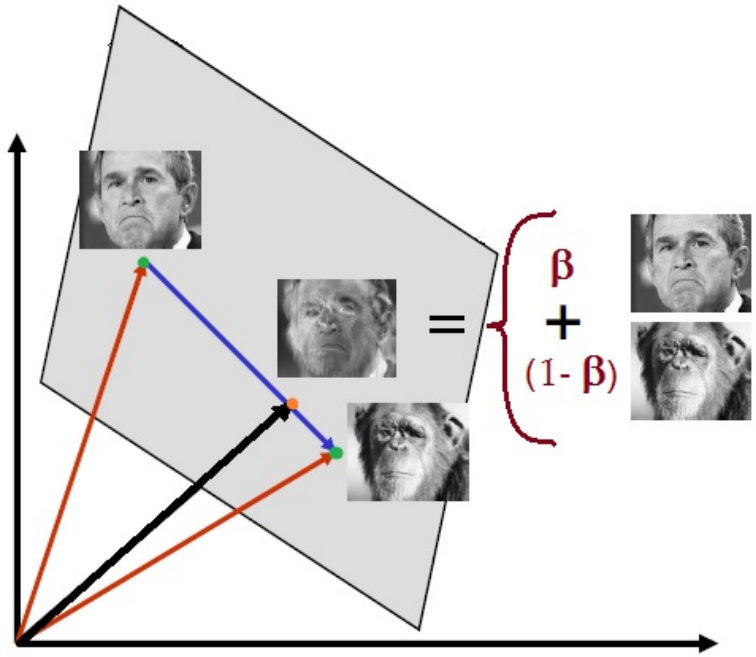
On soustrait l'image moyenne de toutes les images
$A \leftarrow$ Les n images $x - \bar{x}$;

On calcule la matrice de covariance AA^T
$$S \leftarrow \frac{1}{n}\sum_{k=1}^{n}((x_0 - x_k)\cdot(x_0 - x_k)) \leftarrow \frac{AA^T}{n-1};$$

Extraction des vecteurs (v) et valeurs (λ) propres par SVD
$\mathbf{v}, \lambda \leftarrow \mathrm{SVD}(S)$;

En seuillant les λ, on peut ne conserver **que les v principaux** représentant l'ensemble de donnée.

Exemple : Figures propres

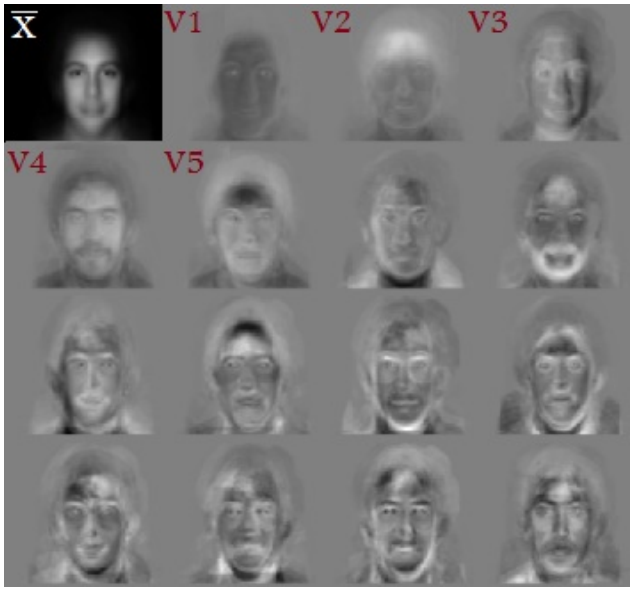


1. L'ACP nous permet d'exprimer les images **par une combinaison d'images clés**.
2. L'ACP nous donne **des images représentatives de l'espace**
3. Filtrer les vecteurs propres nous permet d'obtenir **un sous-espace réduit**.

Exemple : Figures propres

L'ACP extrait les vecteurs propres ($\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_p}$) de S (où p est le nombre de dimensions (pixels)).

Chacun de ces vecteurs est **une direction dans le sous-espace de figures**.



Exemple : Figures propres

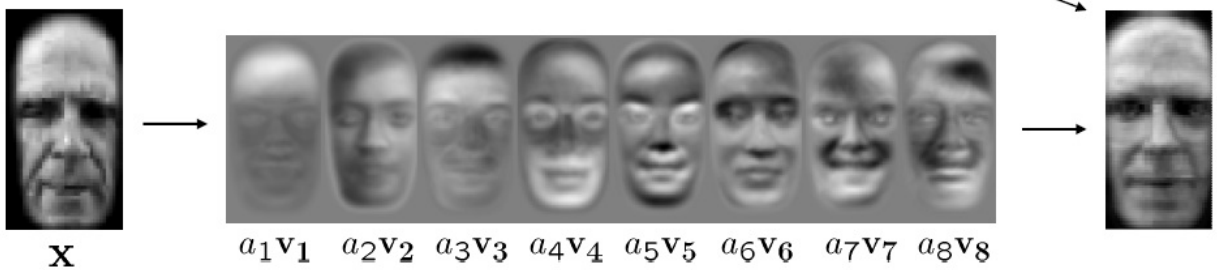
Soit x une image d'entrée.

On **projette** alors l'image centrée-réduite $(x - \bar{x})$ sur **chacun des vecteurs propres** ($\mathbf{v}$).

On forme alors une **combinaison linéaire** de **figures propres** nous permettant de calculer la **projection** de x sur le **sous-espace de figures propres**.

$$\mathbf{x} \to (\underbrace{(\mathbf{x}-\bar{\mathbf{x}})\cdot\mathbf{v_1}}_{a_1}, \underbrace{(\mathbf{x}-\bar{\mathbf{x}})\cdot\mathbf{v_2}}_{a_2}, \ldots, \underbrace{(\mathbf{x}-\bar{\mathbf{x}})\cdot\mathbf{v_K}}_{a_K})$$

$$\mathbf{x} \approx \bar{\mathbf{x}} + a_1\mathbf{v_1} + a_2\mathbf{v_2} + \ldots + a_K\mathbf{v_K}$$



Exemple d'application

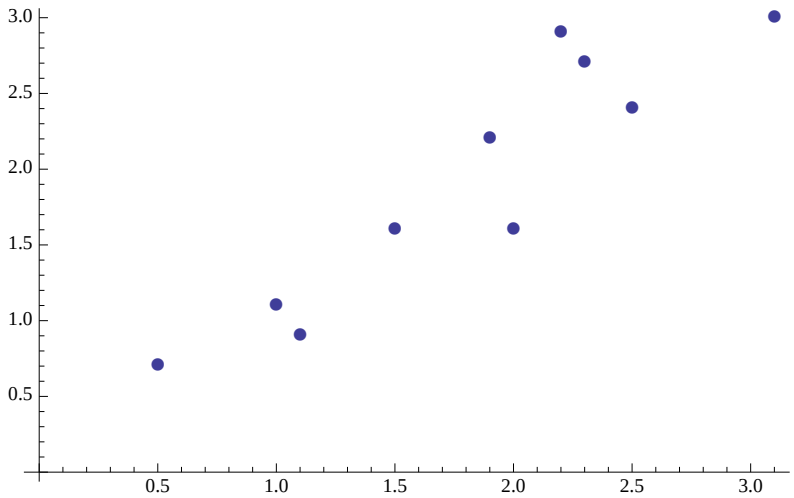
On s'intéresse maintenant aux étapes de **la réalisation d'une analyse en composantes principales** sur un ensemble de données.

Étape 1 : Obtenir des données

Dans cet exemple, on utilisera des données à deux dimensions seulement afin d'être en mesure de bien les visualiser dans leur espace original.

Exemple d'application

x_1	x_2
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2.0	1.6
1.0	1.1
1.5	1.6
1.1	0.9



Exemple d'application

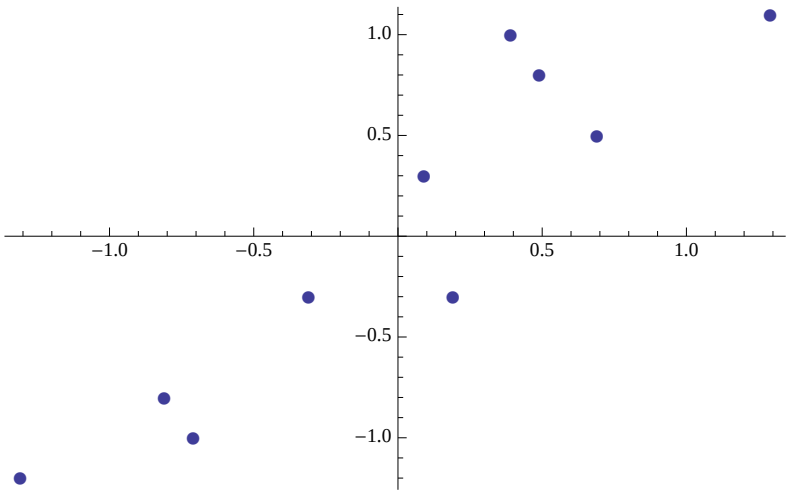
Étape 2 : Centrer les données

Cette étape consiste à **soustraire le vecteur moyen m de chaque donnée**. Le résultat est un nouvel ensemble de données dont la moyenne est **0**.

Dans le cas présent, on trouve que $\mathbf{m} = (m_1, m_2) = (1.81, 1.91)$.

Exemple d'application

$x_1 - m_1$	$x_2 - m_2$
0.69	0.49
-1.31	-1.21
0.39	0.99
0.09	0.29
1.29	1.09
0.49	0.79
0.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01



Exemple d'application

Étape 3 : Calculer la matrice de dispersion

On évalue la matrice de dispersion S (ou la matrice de variance-covariance) à l'aide de la formule donnée précédemment.

Dans le cas présent, on trouve

$$\mathbf{S} = \begin{bmatrix} 0.616556 & 0.615444 \\ 0.615444 & 0.716556 \end{bmatrix}$$

Exemple d'application

Étape 4 : Calculer les vecteurs et valeurs propres de S

En évaluant les valeurs propres et vecteurs propres de la matrice de dispersion, on trouve **les axes principaux du nuage de points**.

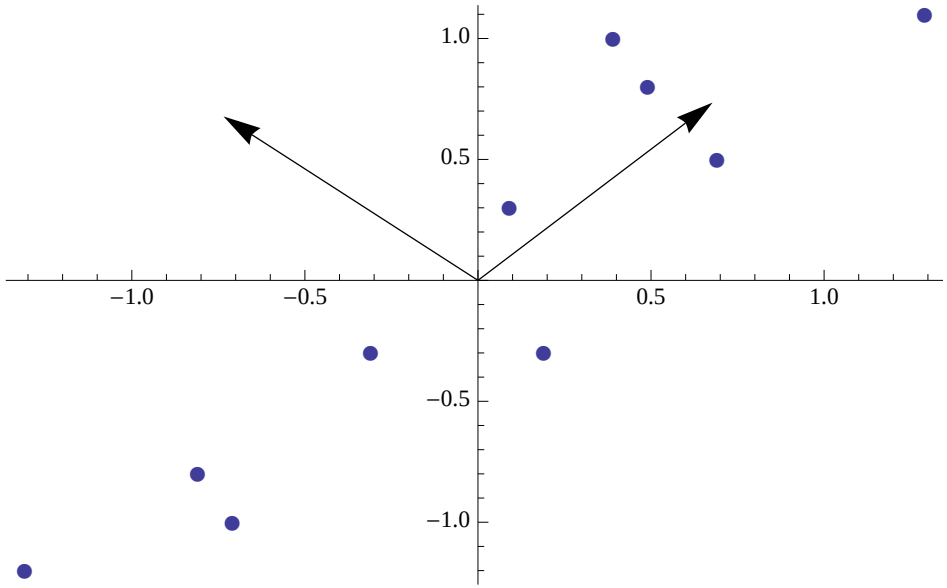
Dans le cas présent, on trouve

$$\mathbf{v}_1 = \begin{bmatrix} 0.677873 \\ 0.735179 \end{bmatrix} \quad \text{et} \quad \mathbf{v}_2 = \begin{bmatrix} -0.735179 \\ 0.677873 \end{bmatrix}$$

qui sont associés aux valeurs propres $\lambda_1 = 1.28403$ et $\lambda_2 = 0.04908$.

Exemple d'application

Pour la suite du processus, il est important que **les vecteurs propres soient normalisés**.



Le vecteur propre avec **la plus grande valeur propre** est la première composante principale associée aux données. Celui-ci illustre **la tendance principale entre les différentes dimensions**.

Exemple d'application

L'utilité de cette méthode repose sur la possibilité d'**ignorer les composantes de moindre importance** et ainsi exprimer les données dans un espace de plus petite dimension.

Pour obtenir cette nouvelle représentation, on forme une matrice **F** dont les colonnes sont les d' premiers vecteurs propres.

Ici, on conservera seulement un vecteur propre :

$$\mathbf{F} = \begin{bmatrix} 0.677873 \\ 0.735179 \end{bmatrix}$$

Exemple d'application

Étape 5 : Changement de repère

C'est ici que l'on obtient le nouvel ensemble de données $\mathbf{D}'$ exprimé dans un espace de dimension réduite.

Soit $\mathbf{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, l'ensemble des données originales centrées et $\mathbf{F}$ la matrice des vecteurs propres. On définit $\mathbf{D}'$ comme

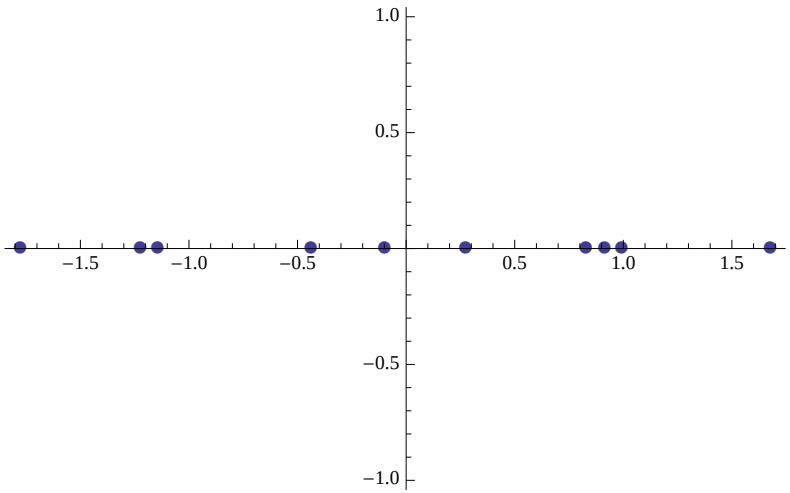
$$\mathbf{D}' = \mathbf{F}^t \times \mathbf{D}^t.$$

On obtiendra alors les données originales **exprimées selon les vecteurs propres choisis**.

Exemple d'application

En conservant seulement un vecteur propre, **on passe de données 2D à des données 1D** :

x
0.82797
-1.77758
0.992198
0.27421
1.6758
0.912949
-0.0991096
-1.14457
-0.438046
-1.22382

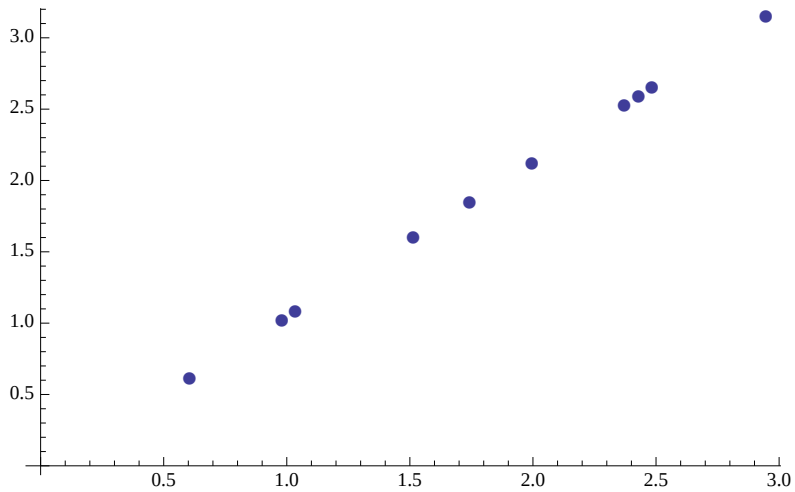


Exemple d'application

Étape 6 : Retour au repère original

On ramène ici les données de dimension réduite **dans leur domaine original** afin de pouvoir les visualiser dans le contexte prévu. Soient $\mathbf{D}''$ ces données, on les obtient par l'équation

$$\mathbf{D}'' = (\mathbf{S}^t \times \mathbf{D}'^t) + \mathbf{m}$$



Positionnement multidimensionnel

Principe de fonctionnement

À la base de la technique de positionnement multidimensionnel, il y a la notion de **distance**. On doit disposer d'une matrice dont les éléments sont **les distances entre chaque échantillon**.

Distance entre 2 villes en km	Arras	Berck-sur-Mer	Béthune	Boulogne-sur-Mer	Calais	Cambrai	Douai	Dunkerque	Hardelot	Hazebrouck	Le Touquet	Lille	Maubeuge	Saint-Omer	Valenciennes	Wimereux
Arras	00	90	31	118	111	35	30	97	117	59	100	51	99	74	66	124
Berck-sur-Mer	90	00	90	39	74	130	110	109	35	93	18	127	201	66	166	45
Béthune	31	90	00	87	82	64	38	71	100	26	95	41	111	44	75	93
Boulogne-sur-Mer	118	39	87	00	36	157	143	72	17	73	31	120	215	53	180	6
Calais	111	74	82	36	00	148	122	38	53	62	67	109	195	40	168	30
Cambrai	35	130	64	157	148	00	27	139	153	92	136	63	66	108	34	163
Douai	30	110	38	143	122	127	00	112	150	64	133	40	73	82	38	141
Dunkerque	97	109	71	72	38	139	112	00	90	41	103	81	176	37	137	68
Hardelot	117	35	100	17	53	153	150	90	00	88	27	127	224	70	188	23
Hazebrouck	59	93	26	73	62	92	64	41	88	00	104	44	132	24	100	79
Le Touquet	100	18	95	31	67	136	133	103	27	104	00	132	208	84	172	37
Lille	51	127	41	120	109	63	40	81	127	44	132	00	90	67	56	119
Maubeuge	99	201	111	215	195	66	73	176	224	132	208	90	00	155	37	210
Saint-Omer	74	66	44	53	40	108	82	37	70	24	84	67	155	00	115	59
Valenciennes	66	166	75	180	168	34	38	137	188	100	172	56	37	115	00	178
Wimereux	124	45	93	6	30	163	141	68	23	79	37	119	211	59	178	00

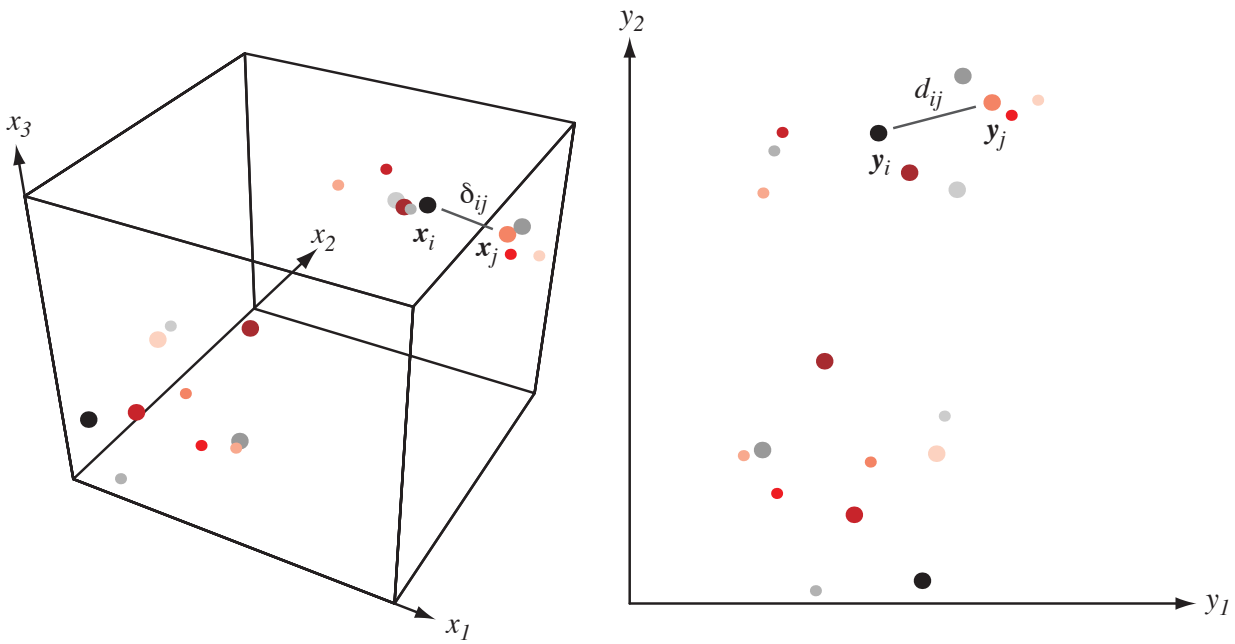
Principe de fonctionnement

Soit $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ un ensemble de données de dimension d et $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ une représentation de X **dans un espace de dimension inférieure** d'.

On note par δ_{ij} la distance entre les échantillons $\mathbf{x}_i$ et $\mathbf{x}_j$ et d_{ij} la distance entre $\mathbf{y}_i$ et $\mathbf{y}_j$.

Principe de fonctionnement

On cherche une configuration de points image $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ pour laquelle les valeurs d_{ij} sont **les plus près possibles** des distances δ_{ij}.



[3]

Principe de fonctionnement

Il sera généralement impossible de trouver un ensemble $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ tel que $d_{ij} = \delta_{ij}$ pour tout i et j. En ce sens, il est nécessaire d'**établir un critère de comparaison entre les configurations**.

La somme du carré des différences s'avère être, encore une fois, un choix approprié.

- $J_{ee} = \dfrac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}$
- $J_{ff} = \displaystyle\sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$
- $J_{ef} = \dfrac{1}{\sum_{i<j} \delta_{ij}} \displaystyle\sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$

Principe de fonctionnement

Il faut noter que ces critères ne sont basés que sur les mesures de distance, ils sont ainsi **invariants aux transformations affines sur les configurations** $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$.

Une fois qu'un critère a été choisi, **on cherche une configuration qui le minimise**. Cette minimisation peut être obtenue à l'aide d'une descente de gradient, par exemple.

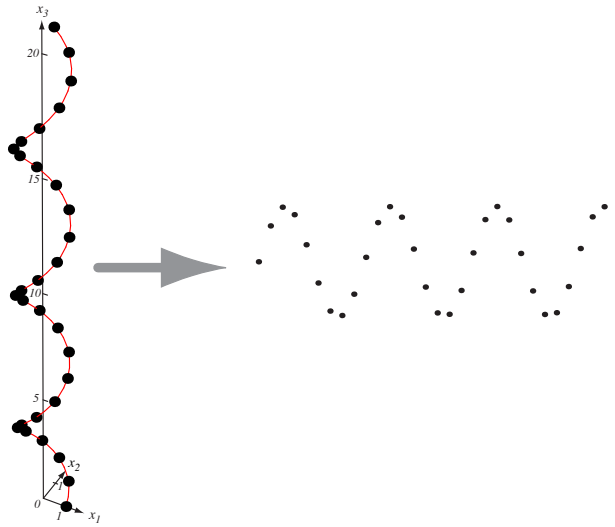
Principe de fonctionnement

À titre d'exemple, considérons 30 échantillons répartis uniformément le long d'une spirale 3D :

$$x_1(k) = \cos\frac{k}{\sqrt{2}} \qquad x_2(k) = \sin\frac{k}{\sqrt{2}} \qquad x_3(k) = \frac{k}{\sqrt{2}}$$

avec $k = 0, \ldots, 29$.

On cherche **une représentation pertinente en 2D**.



[3]

Exemple d'application

Considérons le problème suivant : en observant une carte géographique, **on s'intéresse aux distances séparant certaines villes**. Si on a accès à une règle, on peut facilement mesurer celles-ci.

Une solution mathématique est aussi disponible : connaissant les coordonnées (x_a, y_a) et (x_b, y_b) de deux villes a et b, on peut évaluer la distance euclidienne entre elles :

$$d_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}.$$

Exemple d'application

Attardons-nous maintenant au problème inverse : en connaissant seulement les distances δ_{ij} entre les villes, est-il possible de **retrouver la carte** ?

C'est ce que suggère de faire le positionnement multidimensionnel. Au lieu d'utiliser des algorithmes itératifs d'optimisation, voyons une méthode faisant appel aux outils de l'**algèbre linéaire**...

Exemple d'application

Étape 1 : Obtenir la matrice des proximités au carré $\mathbf{P}^{(2)}$

L'algorithme de positionnement multidimensionnel repose sur notre capacité à retrouver les coordonnées des points $\mathbf{y}_i$ en ne connaissant que les distances δ_{ij}.

Dans l'exemple suivant, on dispose des distances entre quatre villes du Danemark sur une carte : Copenhague (cph), Aarhus (aar), Odense (ode) et Aalborg (aal). Comme il s'agit de distance sur une carte, **on sera à la recherche d'une représentation 2D** pour les points.

Exemple d'application

La carte des distances étant

	cph	aar	ode	aal
cph	0	93	82	133
aar	93	0	52	60
ode	82	52	0	111
aal	133	60	111	0

On trouve que la **matrice des proximités au carré** est

$$\mathbf{P}^{(2)} = \begin{bmatrix} 0 & 8649 & 6724 & 17689 \\ 8649 & 0 & 2704 & 3600 \\ 6724 & 2704 & 0 & 12321 \\ 17689 & 3600 & 12321 & 0 \end{bmatrix}$$

Exemple d'application

Étape 2 : Centrer les données de distance

On définit la **matrice de centrage J** $= I - \frac{1}{n} 1_{n \times n}$.

$$\mathbf{J} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.75 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0.75 & -0.25 & -0.25 \\ -0.25 & -0.25 & 0.75 & -0.25 \\ -0.25 & -0.25 & -0.25 & 0.75 \end{bmatrix}$$

Exemple d'application

On l'applique ensuite sur $\mathbf{P}^{(2)}$ pour obtenir la **matrice des distance centrées B** avec la formule $\mathbf{B} = -\frac{1}{2} J \mathbf{P}^{(2)} J$.

$$
\begin{aligned}
\mathbf{B} &= -\frac{1}{2}\mathbf{J}\mathbf{P}^{(2)}\mathbf{J} \\
&= \begin{bmatrix}
5035.06 & -1553.06 & 258.938 & -3740.94 \\
-1553.06 & 507.813 & 5.3125 & 1039.94 \\
258.938 & 5.3125 & 2206.81 & -2471.06 \\
-3740.94 & 1039.94 & -2471.06 & 5172.06
\end{bmatrix}
\end{aligned}
$$

Exemple d'application

Étape 3 : Extraire les valeurs propres et vecteurs propres de B

Pour obtenir une représentation à d' dimensions pour l'ensemble de points $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, on trouve **les d' plus grandes valeurs propres** λ_i ainsi que les vecteurs propres $\mathbf{v}_i$ qui leur sont associés.

Exemple d'application

Dans le cas présent, comme on cherche **une représentation 2D**, on se contente de trouver λ_1 et λ_2 accompagnés de $\mathbf{v}_1$ et $\mathbf{v}_2$:

$$\mathbf{v}_1 = \begin{bmatrix} -0.63716 \\ 0.186621 \\ -0.253117 \\ 0.703656 \end{bmatrix} \quad \text{et} \quad \mathbf{v}_2 = \begin{bmatrix} 0.586498 \\ -0.213917 \\ -0.706315 \\ 0.333734 \end{bmatrix}$$

qui sont associés aux valeurs propres $\lambda_1 = 9724.17$ et $\lambda_2 = 3160.99$.

Exemple d'application

Étape 4 : Obtenir les coordonnées de l'ensemble $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$

Soit **D** la matrice des coordonnées des points de dimension d' de l'ensemble $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. On l'obtient avec la formule

$$\mathbf{D} = \mathbf{V}_{d'} \mathbf{L}_{d'}^{\frac{1}{2}},$$

où $\mathbf{V}_{d'}$ est la matrice des d' vecteurs propres, tandis que $\mathbf{L}_{d'}^{\frac{1}{2}}$ est la matrice diagonale des racines carrées des d' plus grandes valeurs propres de **B**.

Exemple d'application

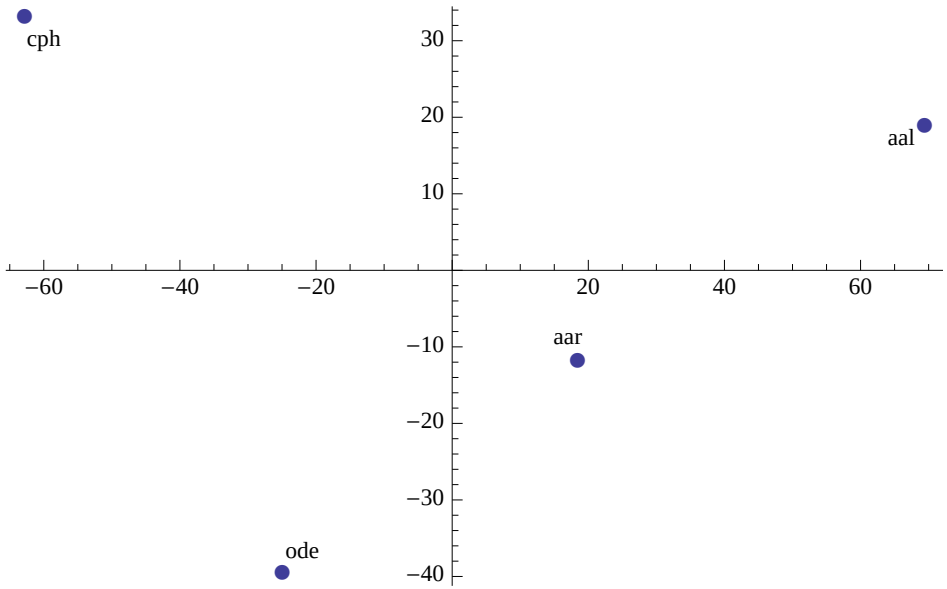
Dans le cas présent, on a donc

$$\mathbf{D} = \begin{bmatrix} -0.63716 & 0.586498 \\ 0.186621 & -0.213917 \\ -0.253117 & -0.706315 \\ 0.703656 & 0.333734 \end{bmatrix} \times \begin{bmatrix} 98.6112 & 0 \\ 0 & 56.2226 \end{bmatrix}$$

$$= \begin{bmatrix} -62.8311 & 32.9745 \\ 18.4029 & -12.027 \\ -24.9602 & -39.7109 \\ 69.3884 & 18.7634 \end{bmatrix}$$

Exemple d'application

Visuellement, ce résultat correspond à



Rappelons que si on avait souhaité un résultat de dimension supérieure, **il aurait suffit de considérer plus que deux vecteurs propres**.

Réduction de la dimension : méthodes non linéaires

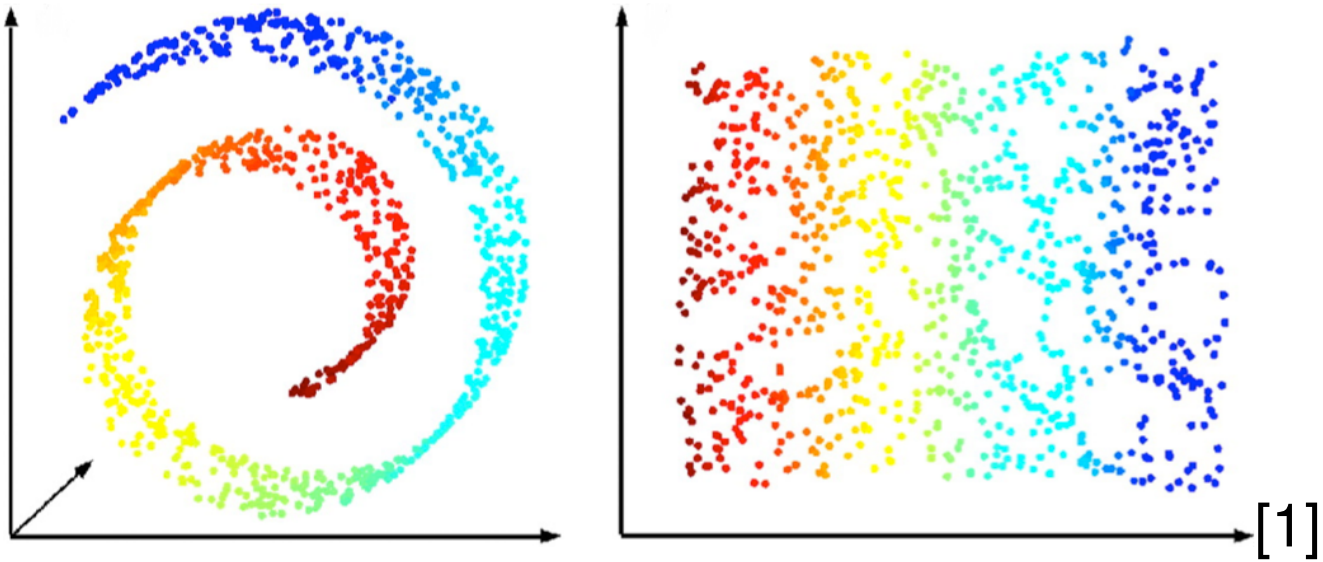
Isomap

Principe de fonctionnement

L'objectif des approches non linéaires de réduction de la dimension est toujours le même : **trouver une représentation significative à basse dimension** pour un ensemble de points de dimension élevée.

Toutefois, à l'opposé des méthodes précédentes, les deux techniques suivantes, **Isomap** et **Insertion linéaire locale**, proposent de **mieux approximer la structure géométrique réelle** de l'ensemble des données.

Principe de fonctionnement

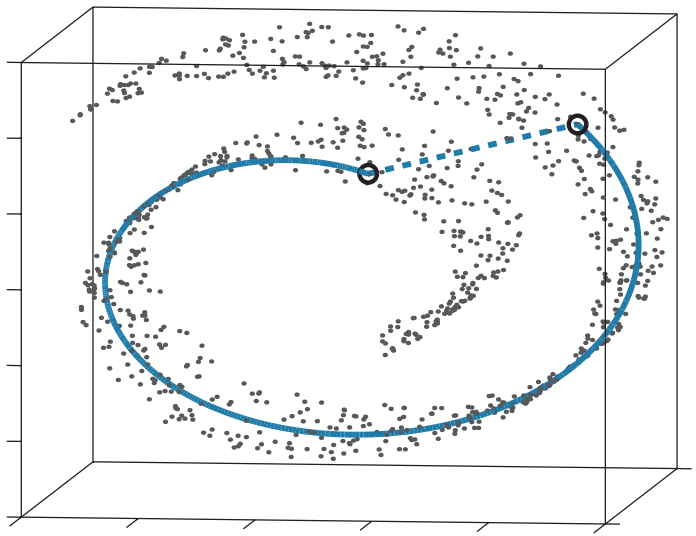


Principe de fonctionnement

La méthode Isomap combine **les avantages algorithmiques du positionnement multidimensionnel** (basse complexité, optimisation globale, garantie de convergence) tout en ayant **la flexibilité nécessaire** pour s'adapter aux structures non linéaires.

Principe de fonctionnement

Pour illustrer les problématiques associées aux structures non linéaires, considérons l'exemple du rouleau suisse :



[10]

Le problème vient du fait que deux points peuvent être **à proximité selon la distance euclidienne**, mais très éloignés **si on mesure la distance sur la surface définie par les points**.

Principe de fonctionnement

La solution à ce problème est de **ne plus considérer les distances euclidiennes**, mais plutôt d'**utiliser les distances géodésiques**. Or, c'est là que le bât blesse pour les méthodes linéaires, car celles-ci reposent uniquement sur la distance euclidienne.

Isomap reprend l'idée du positionnement multidimensionnel, mais en utilisant **une table de distances géodésiques**.

Principe de fonctionnement

Une question demeure sans réponse : si on ne dispose que d'un ensemble discret de points, **on ne peut connaître la forme réelle de la surface sur laquelle reposent ceux-ci**.

Si on ne connaît pas la surface, **comment peut-on évaluer les distances géodésiques** ?

Principe de fonctionnement

Partant du principe que, pour des points voisins, la distance euclidienne fournit une approximation juste de la distance géodésique, on peut **estimer celle-ci en faisant une série de petits sauts** entre des points voisins.

Ces approximations peuvent être calculées efficacement en **représentant les points sous forme de graphe**.

L'algorithme Isomap s'effectue en **trois étapes**.

Principe de fonctionnement

Étape 1 : Déterminer quels points sont voisins

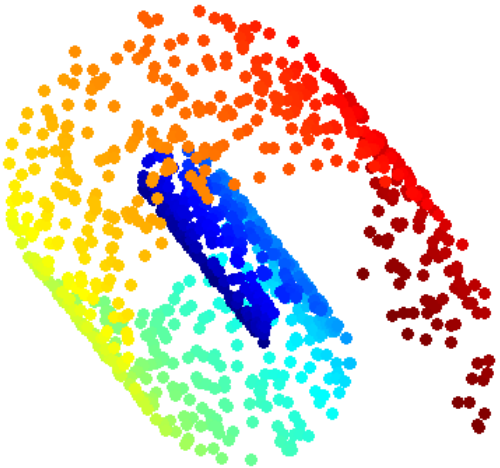
Dans l'ensemble original X de dimension d, on utilise la distance euclidienne pour trouver, pour chaque point $\mathbf{x}_i$, **l'ensemble des points $\mathbf{x}_j$ situés à l'intérieur d'un certain rayon ϵ.**

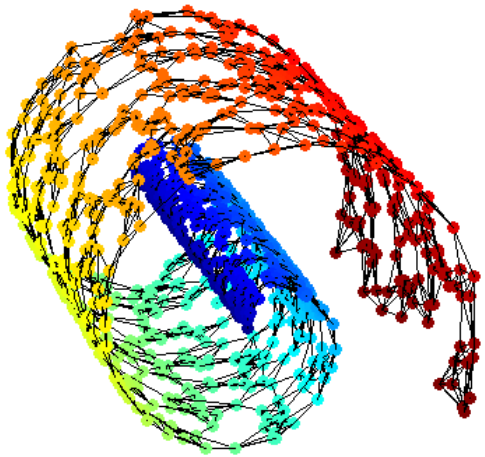
Principe de fonctionnement

Ces relations de voisinage sont représentées par un graphe G dont **les noeuds sont les points $\mathbf{x}_i$** et où **le poids de l'arête reliant $\mathbf{x}_i$ à $\mathbf{x}_j$ correspond à la distance euclidienne** entre $\mathbf{x}_i$ et $\mathbf{x}_j$.

Notons que si deux points $\mathbf{x}_k$ et $\mathbf{x}_\ell$ ne sont pas voisins (c'est-à-dire si $d_X(k, \ell) > \epsilon$), alors le poids de l'arête est fixé à ∞.

Principe de fonctionnement





[5]

Principe de fonctionnement

Étape 2 : Calcul des distances géodésiques

Pour toutes les paires de points $(\mathbf{x}_i, \mathbf{x}_j)$, on calcule la distance géodésique $d_G(i,j)$ en se basant sur le graphe G et en appliquant un algorithme de **recherche du plus court chemin**.

On construit ainsi une **matrice de distances géodésiques $\mathbf{D}_G$**.

Principe de fonctionnement

Étape 3 : Réduction de la dimension

On applique finalement l'algorithme du **positionnement multidimensionnel** sur la matrice des distances géodésiques pour ramener les points dans un ensemble Y de dimension inférieure d' qui préserve le plus possible la géométrie du nuage de points original.

Exemple d'application

Un problème classique en réduction de la dimension est celui de la **perception du visage** selon différents angles et sous plusieurs conditions d'éclairage.

[10]

Chaque image peut être considérée comme **un point dans un espace de haute dimension**.

Exemple d'application

En effet, si les images sont de dimensions 64×64, alors elles seront toutes représentées par un vecteur de dimension 4096 où **chaque composante sera l'intensité d'un pixel**.

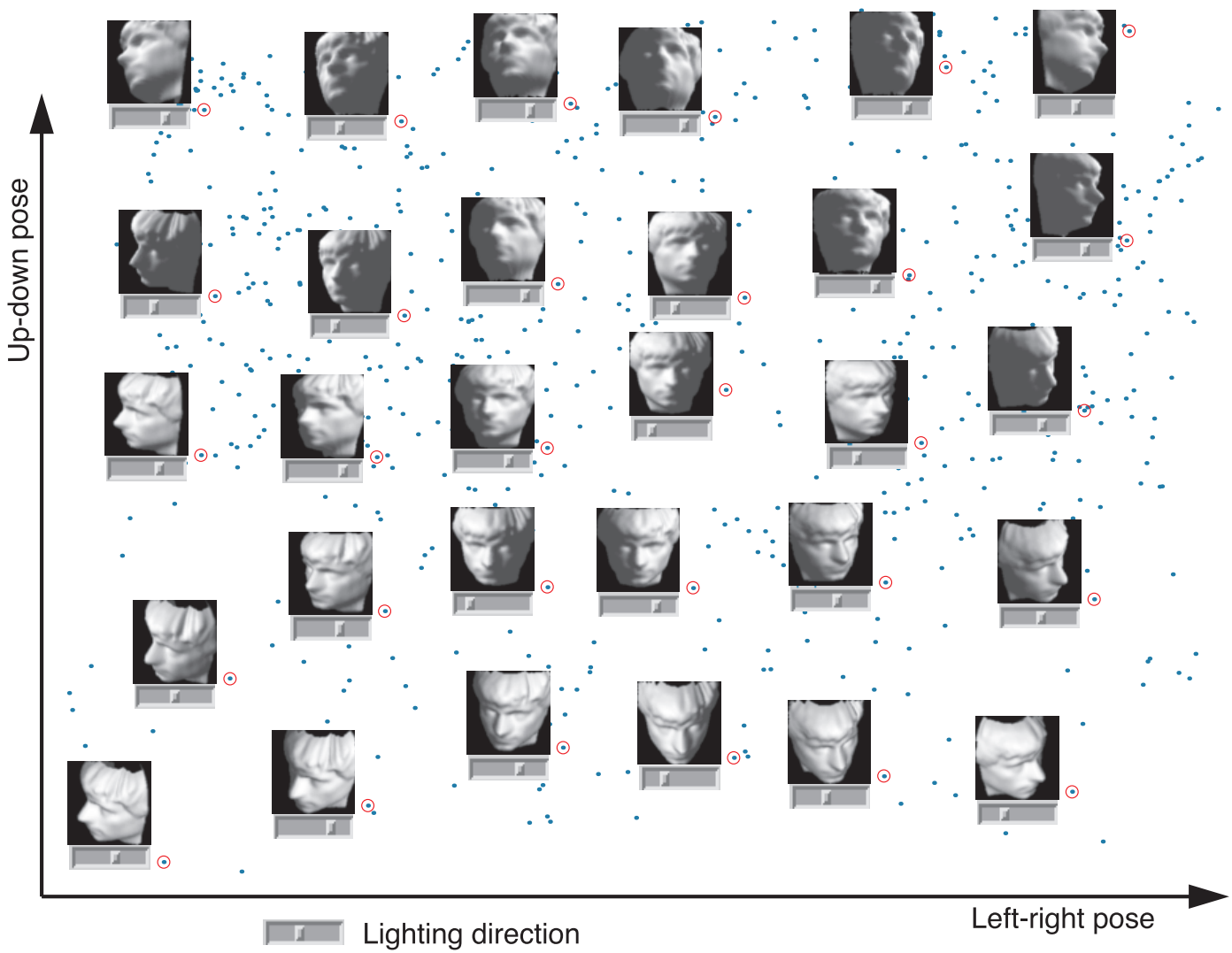
Perceptuellement, ces 4096 dimensions **ne sont pas toutes utiles**.

Exemple d'application

Dans le cas présent, on peut même supposer que **trois axes seront suffisants** pour exprimer convenablement les images :

- Orientation gauche-droite de la tête ;
- Orientation haut-bas de la tête ;
- Direction de l'éclairage

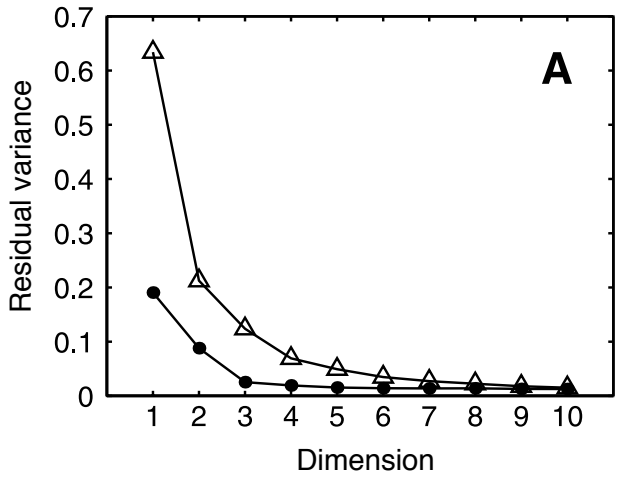
Exemple d'application



[10]

Exemple d'application

L'algorithme Isomap permettra d'obtenir **une représentation de qualité à basse dimension** à partir des vecteurs de taille 4096.



[10]

Le graphique précédent illustre la **variance résiduelle** pour le positionnement multidimensionnel (△) et pour Isomap (●).

Locally Linear Embedding

Principe de fonctionnement

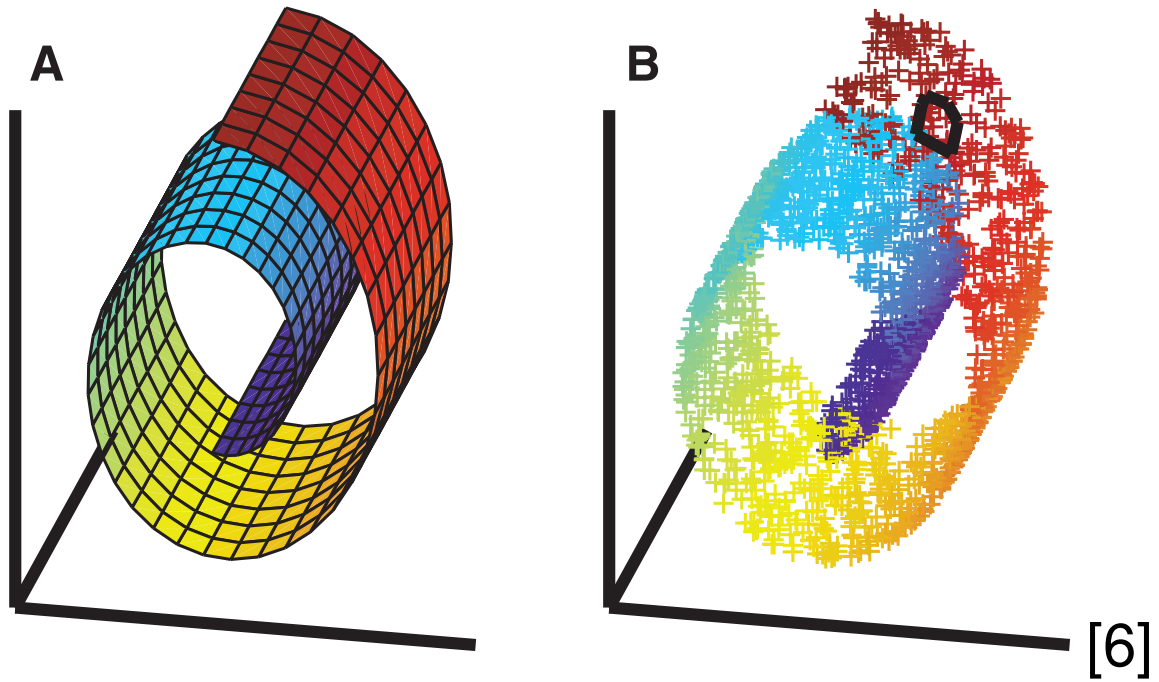
Comme c'était le cas pour la technique Isomap, la méthode du *Locally Linear Embedding* (LLE) a pour objectif de réduire la dimension de l'espace dans lequel les données sont exprimées **tout en préservant leur strucure globale non linéaire**.

Principe de fonctionnement

Supposons que l'on dispose de n vecteurs $\mathbf{x}_i$ de dimension d **reposant approximativement sur une variété** de dimension d'.

Si la densité des échantillons est assez grande, on peut supposer que chaque point, de même que son voisinage, repose sur une région de la variété qui soit **localement approximativement linéaire**.

Principe de fonctionnement



Principe de fonctionnement

L'algorithme LLE s'effectue en trois étapes.

Étape 1 : Trouver les k plus proches voisins de chaque point $\mathbf{x}_i$

Notons que si les points sont trop éloignés les uns des autres, on peut toutefois se limiter à moins de k voisins.

Principe de fonctionnement

Étape 2 : Évaluer la matrice des poids W

Partant de la supposition de linéarité énoncée précédemment, on est en mesure d'**exprimer chaque points à partir de ses voisins** à l'aide de coefficients linéaires.

Les **erreurs de reconstruction** sont données par

$$E(\mathbf{W}) = \sum_{i=1}^{n} \left| \mathbf{x}_i - \sum_{j \neq i} w_{ij} \mathbf{x}_j \right|^2 ,$$

où le coefficient w_{ij} de la matrice $\mathbf{W}$ est le poids de la contribution du j-ième point à la i-ième reconstruction.

Principe de fonctionnement

On cherche donc la matrice **W** dont les éléments minimisent les erreurs de reconstruction.

On posera $w_{ij} = 0$ si $\mathbf{x}_i$ n'est pas dans le voisinage de $\mathbf{x}_j$, et on exigera que

$$\sum_{j=1}^{n} w_{ij} = 1,$$

de manière à assurer l'**invariance à la translation** de la solution.

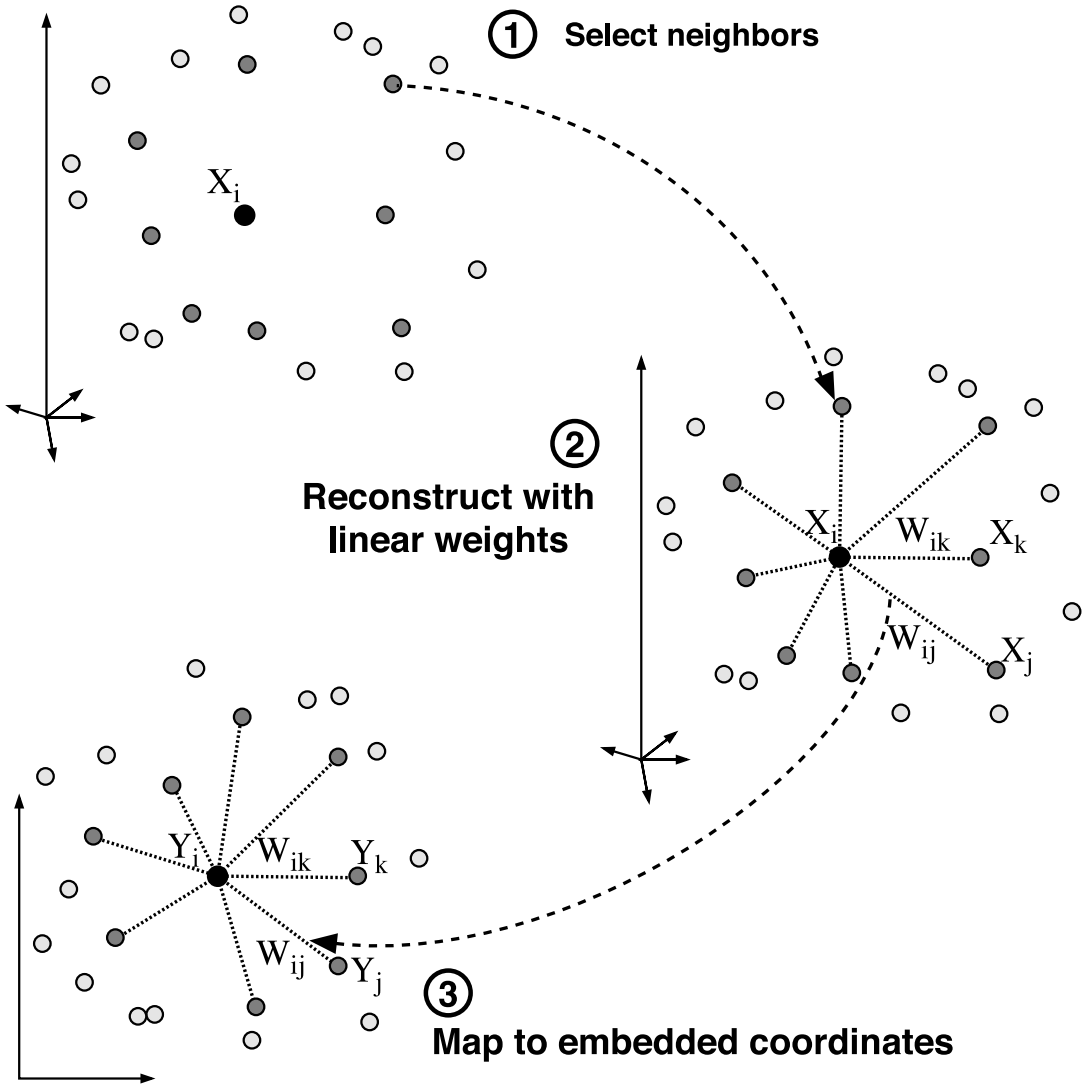
Principe de fonctionnement

Étape 3 : Réduction de la dimension

À partir des poids trouvés à l'étape précédente, on cherche l'ensemble de points $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ de dimension inférieure d' qui minimise l'erreur de reconstruction

$$E(Y) = \sum_{i=1}^{n} \left| \mathbf{y}_i - \sum_{j \neq i} w_{ij} \mathbf{y}_j \right|^2 .$$

Principe de fonctionnement



[6]

Principe de fonctionnement

Comme pour les méthodes précédentes, **la solution optimale peut être obtenue à l'aide des vecteurs propres et valeurs propres**.

Le texte de Cosma Shalizi dans les références propose les détails de cette démarche.

Références

Références I


M. Balasubramanian and E. L. Schwartz.
The isomap algorithm and topological stability, 2002.


M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars.
Computational geometry, 2010.


R. O. Duda, P. E. Hart, and D. G. Stork.
Pattern classification, 2001.


R. Gutierrez-Osuna.
Intelligent sensor systems, 2001.


G. Peyre.
Manifold learning with isomap, 2008.


S. T. Roweis and L. K. Saul.
Nonlinear dimensionality reduction by locally linear embedding, 2000.


C. Shalizi.
Nonlinear dimensionality reduction 1 : Local linear embedding, 2009.


L. I. Smith.
A tutorial on principal components analysis, 2002.


C. Taras, T. Ertl, R. Botchen, and I. Entina.
Online course for scientific visualization, 2007.

Références II


J. B. Tenenbaum, V. de Silva, and J. C. Langford.
A global geometric framework for nonlinear dimensionality reduction, 2000.


F. Wickelmaier.
An introduction to mds, 2003.