

Apprentissage automatique

Réduction de dimensionnalité - motivation

APPRENTISSAGE AUTOMATIQUE

Sujets: types d'apprentissage

RAPPEL

- Il existe différents types d'apprentissage
 - apprentissage supervisé : il y a une cible à prédire

$$\mathcal{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$$

- apprentissage non-supervisé : cible n'est pas fournie

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

- apprentissage par renforcement (non couvert dans ce cours)

APPRENTISSAGE AUTOMATIQUE

Sujets: types d'apprentissage

RAPPEL

- Il existe différents types d'apprentissage
 - apprentissage supervisé : il y a une cible à prédire

$$\mathcal{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$$

- apprentissage non-supervisé : cible n'est pas fournie

$$\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

- apprentissage par renforcement (non couvert dans ce cours)

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: réduction de dimensionnalité

- Formellement, le problème est d'apprendre une fonction $y(x)$ telle que

$$\mathbf{y} : \mathbb{R}^D \rightarrow \mathbb{R}^M$$

où la dimensionnalité $M < D$ (c'est un hyper-paramètre)

- Applications
 - visualisation de données
 - limiter le sur-apprentissage

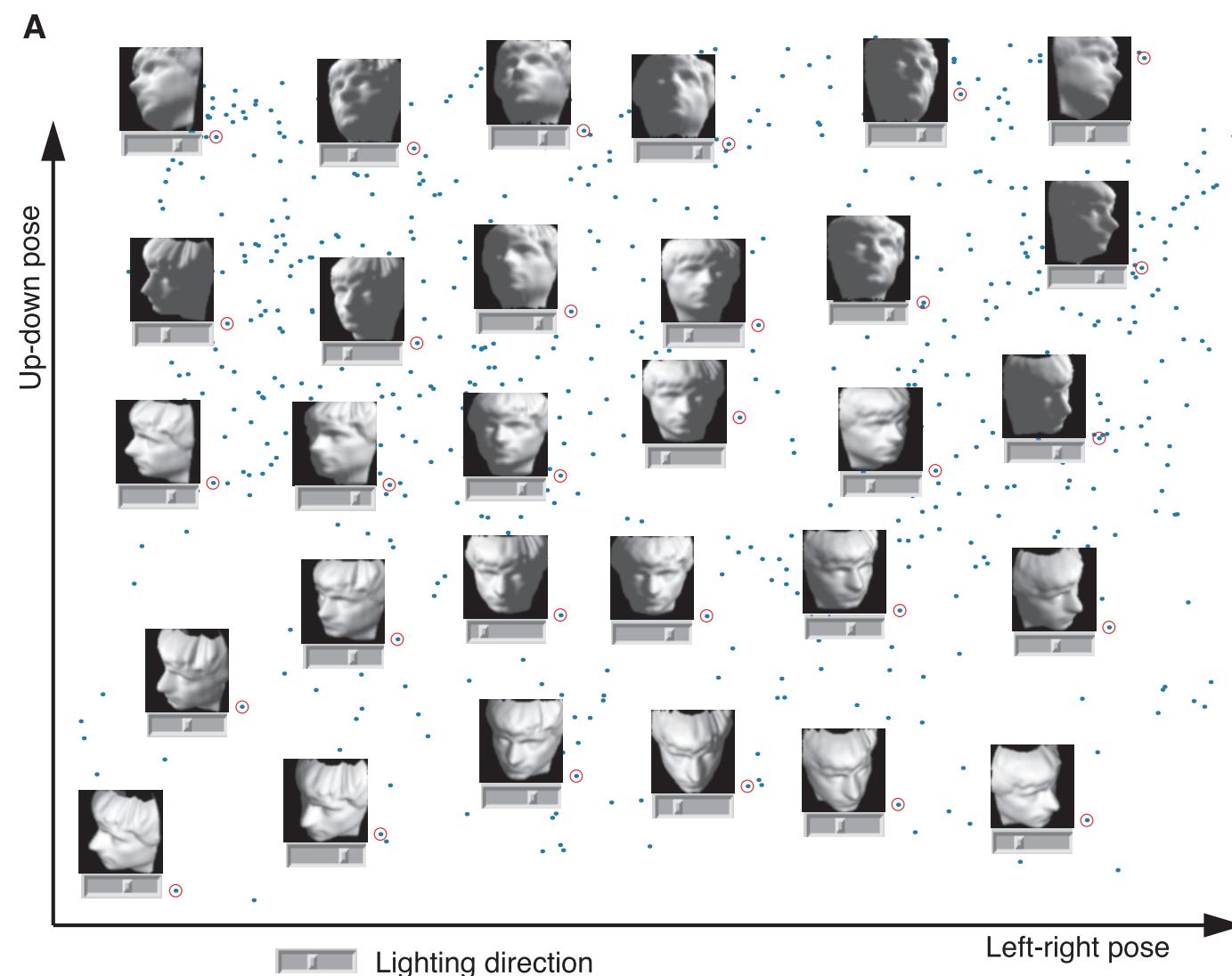
TYPES D'APPRENTISSAGE

Sujets: apprentissage non-supervisé, visualisation

RAPPEL

- L'apprentissage non-supervisé est lorsqu'une cible n'est pas explicitement donnée
 - visualisation de données

Tenenbaum, de Silva,
Langford, (2000)



MALÉDICTION DE LA DIMENSIONNALITÉ

Sujets: malédiction de la dimensionnalité

RAPPEL

- La difficulté à bien généraliser peut donc potentiellement augmenter **exponentiellement** avec la dimensionnalité D des entrées
- Cette observation est appelée la **malédiction de la dimensionnalité**
- Nécessite le design de modèles / algorithmes appropriés pour chaque problème
 - on cherche des modèles / algorithmes qui vont bien exploiter les données à notre disposition

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: dimensionnalité intrinsèque

- Ne perd-on pas de l'information ?
 - pas si la **dimensionnalité intrinsèque** est basse
- Exemple : images (une dimension par pixel)
 - varier individuellement chacun des D pixels ne résulte presque jamais en une image compréhensible

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: dimensionnalité intrinsèque

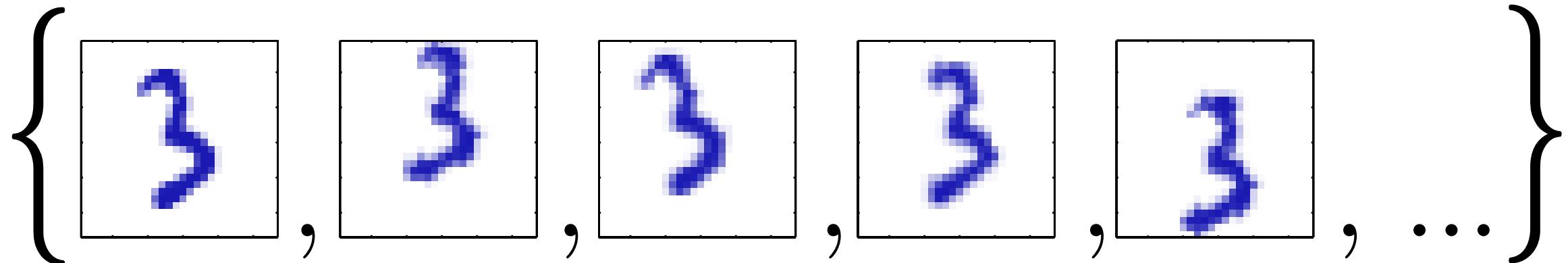
- Ne perd-on pas de l'information ?
 - pas si la **dimensionnalité intrinsèque** est basse
- Exemple : images (une dimension par pixel)
 - varier individuellement chacun des D pixels ne résulte presque jamais en une image compréhensible



RÉDUCTION DE DIMENSIONNALITÉ

Sujets: dimensionnalité intrinsèque

- Soit un jeu de données généré en prenant une seule image du caractère «3» et en appliquant différentes (1) translations verticales, (2) horizontales et (3) rotations

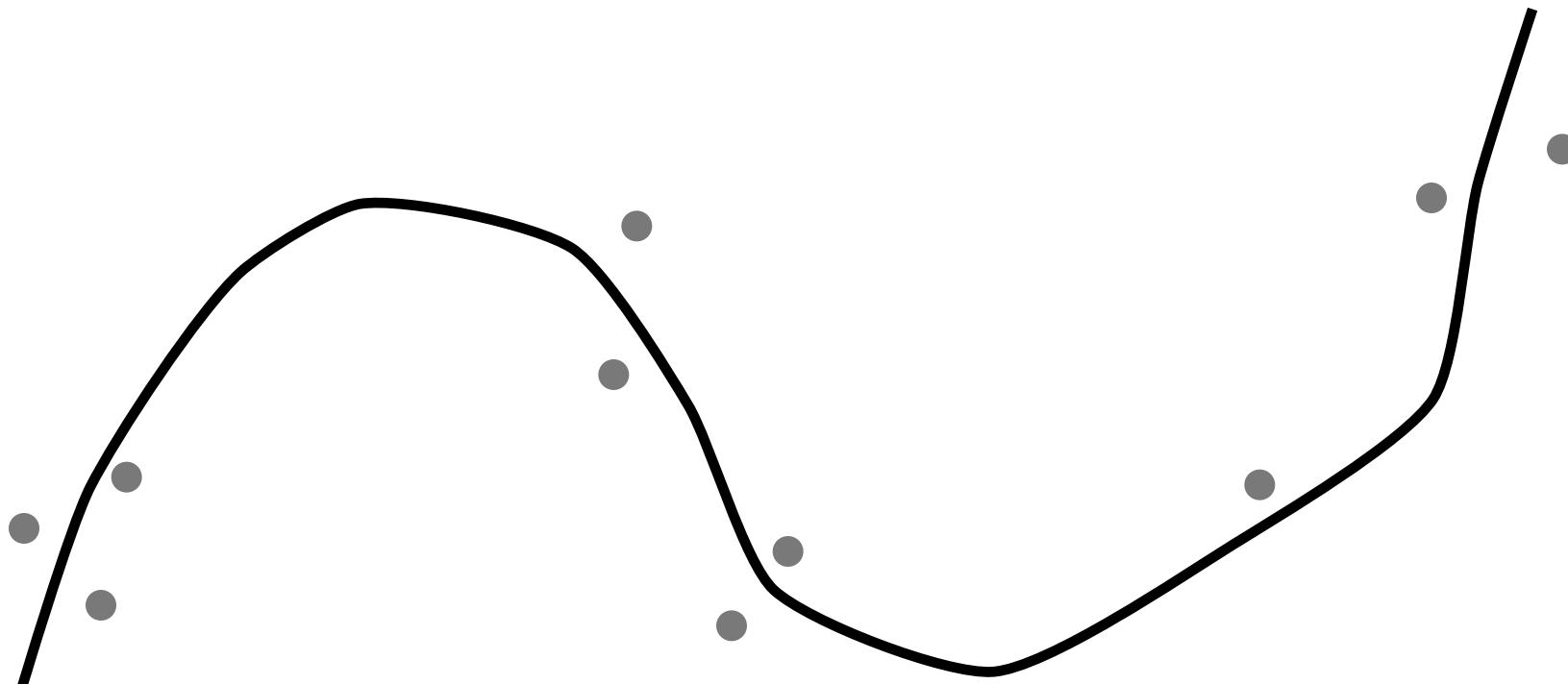


- Même si les images sont 100×100 ($D=10\,000$), les images ne peuvent varier que selon 3 degrés de liberté
 - la dimensionnalité intrinsèque (M) est donc de 3

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété, *manifold*

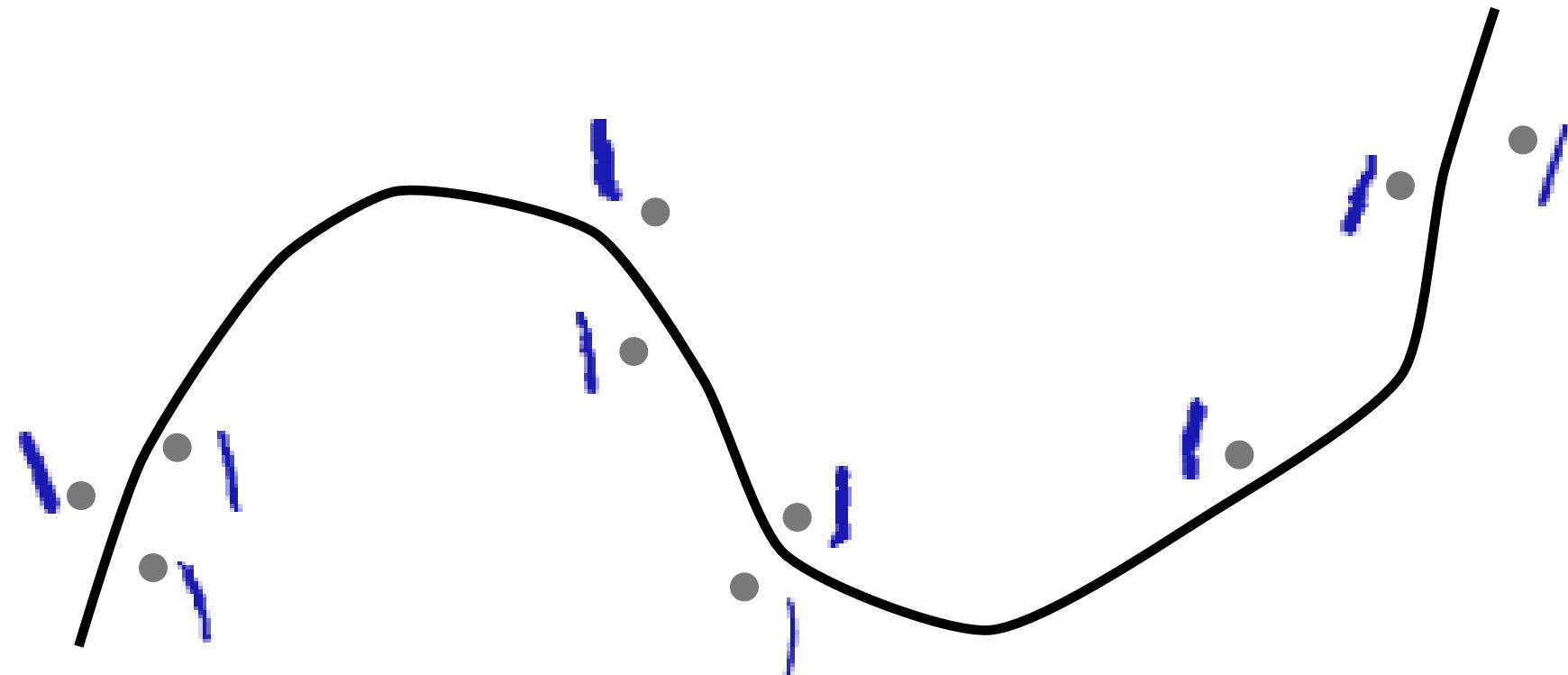
- De façon générale, lorsque D est grand, on s'attend à ce que les données se trouvent surtout autour d'une **variété (*manifold*)** de dimensionnalité $M < D$



RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété, *manifold*

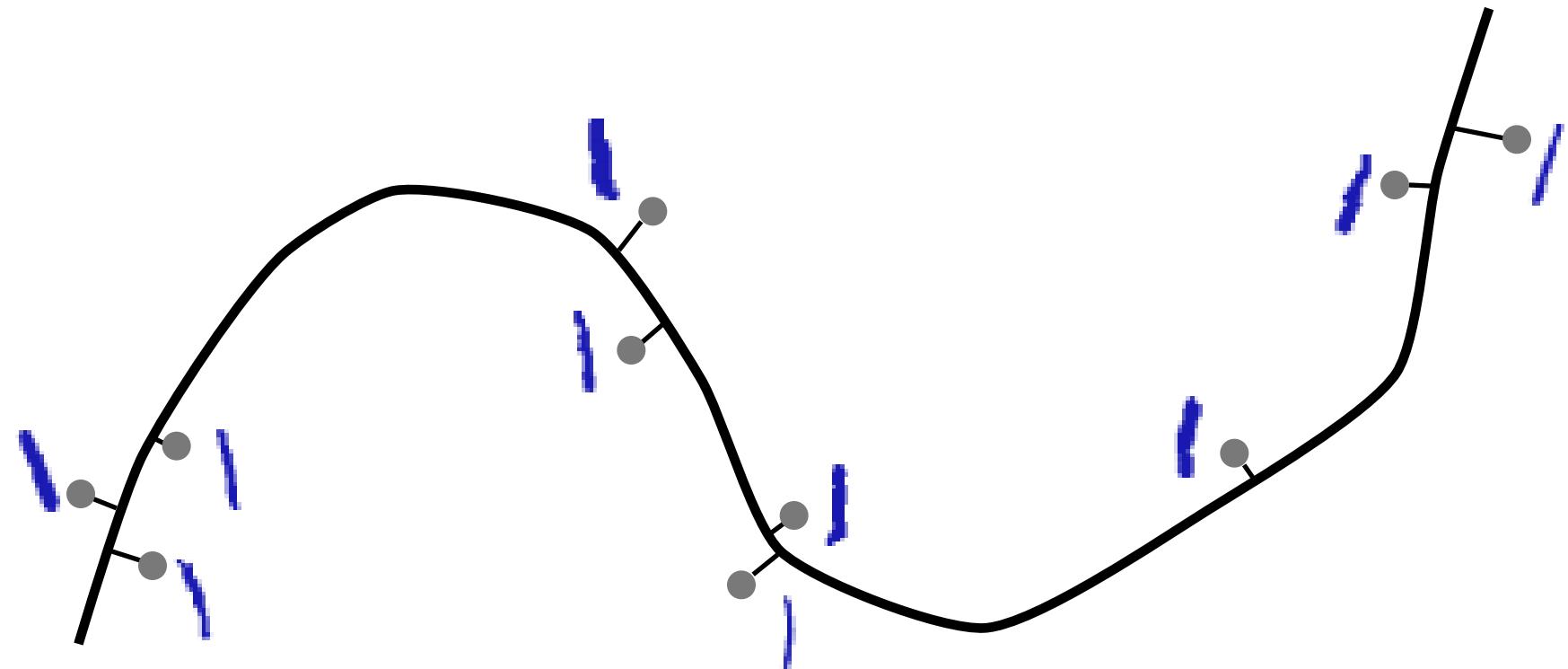
- De façon générale, lorsque D est grand, on s'attend à ce que les données se trouvent surtout autour d'une **variété (*manifold*)** de dimensionnalité $M < D$



RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété, *manifold*

- De façon générale, lorsque D est grand, on s'attend à ce que les données se trouvent surtout autour d'une **variété (*manifold*)** de dimensionnalité $M < D$



Apprentissage automatique

Réduction de dimensionnalité - analyse en composantes principales

ANALYSE EN COMPOSANTES PRINCIPALES

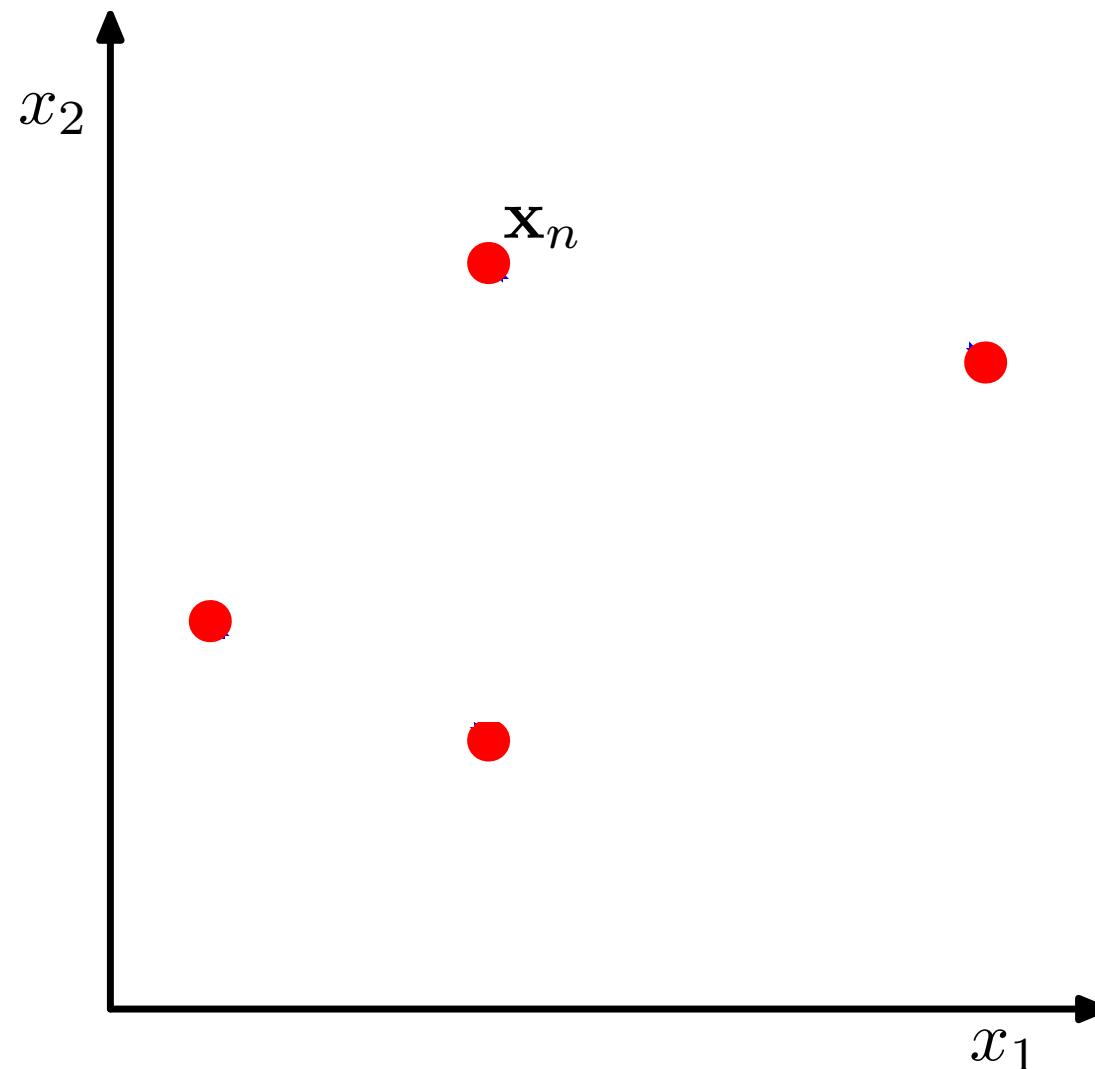
Sujets: analyse en composantes principales

- L'analyse en composantes principales (ACP) est un des algorithmes de réduction de dimensionnalité les plus simples
 - en anglais : *principal component analysis* (PCA)
- **Idée :** projeter les données de façon à maximiser la variance des données projetées

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

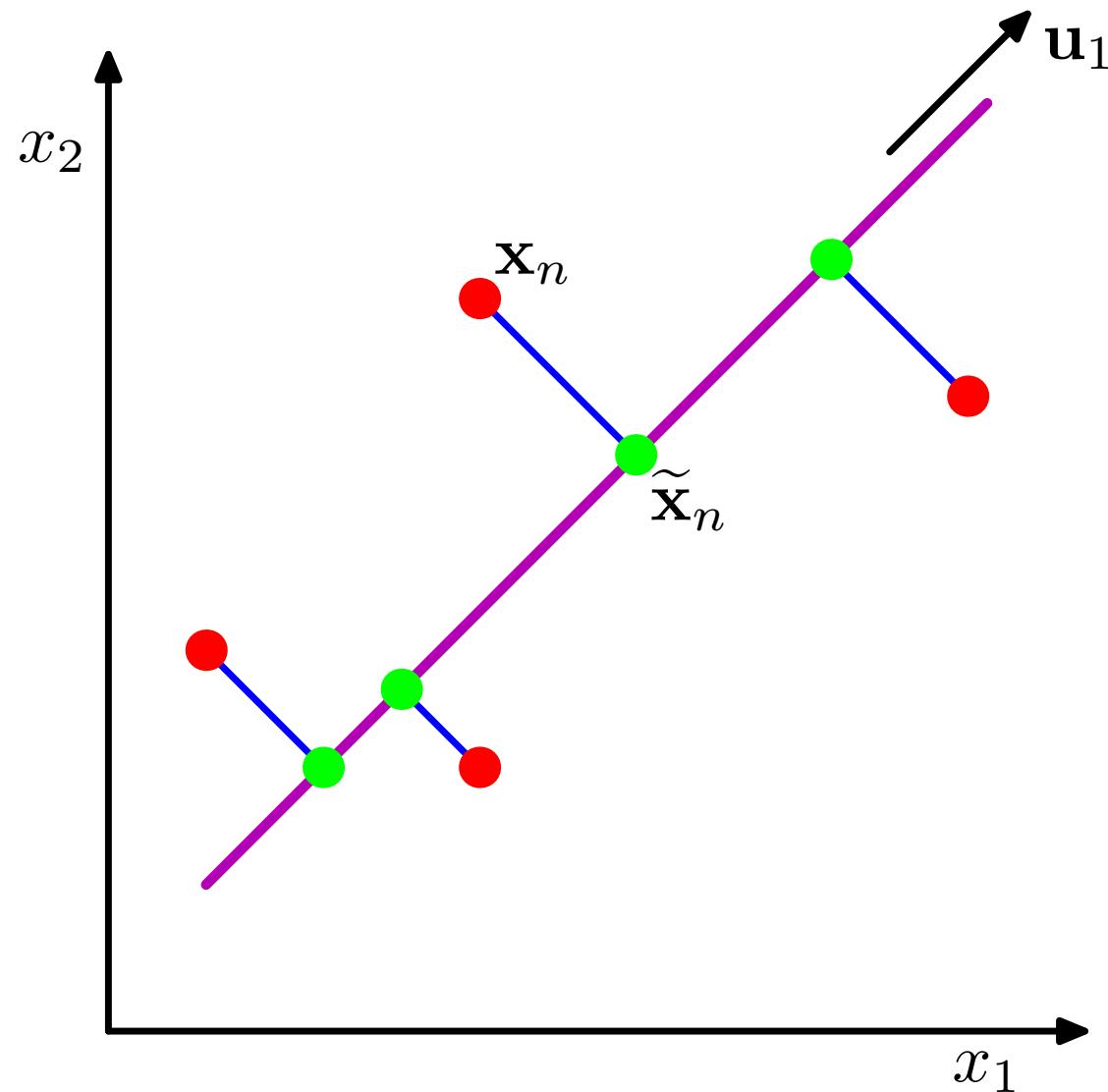
- Exemple : de $D=2, M=1$



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- Exemple : de $D=2, M=1$



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- La variance des données projetées en $M=1$ dimension est

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \{\mathbf{u}_1^T \mathbf{x}_n - \mathbf{u}_1^T \bar{\mathbf{x}}\}^2 &= \frac{1}{N} \sum_{n=1}^N \{\mathbf{u}_1^T (\mathbf{x}_n - \bar{\mathbf{x}})\}^2 \\ &= \frac{1}{N} \sum_{n=1}^N (\mathbf{u}_1^T (\mathbf{x}_n - \bar{\mathbf{x}})) ((\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{u}_1) \\ &= \mathbf{u}_1^T \left(\frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T \right) \mathbf{u}_1 \\ &= \mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 \end{aligned}$$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- On constraint \mathbf{u}_1 à être de norme 1 et on maximise

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 + \lambda_1 (1 - \mathbf{u}_1^T \mathbf{u}_1)$$

- En fixant le gradient par rapport à \mathbf{u}_1 à 0, on obtient que la solution doit satisfaire

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

- C'est donc un des vecteurs propres de \mathbf{S}

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- Si \mathbf{u}_1 est un vecteur propre, alors la variance sera égale à

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \mathbf{u}_1^T (\lambda_1 \mathbf{u}_1) = \lambda_1$$

- Pour maximiser la variance on doit donc prendre le vecteur propre \mathbf{u}_1 dont la valeur propre λ_1 est la plus grande
- La fonction $y(\mathbf{x})$ pour l'ACP avec $M=1$ est alors

$$y(\mathbf{x}) = \mathbf{u}_1^T \mathbf{x}$$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- Pour $M > 1$, on y va itérativement
 - à chaque fois, on cherche une autre projection qui maximise la variance, mais qui est orthogonale aux projections précédentes
- On peut montrer que le résultat correspond à garder les M vecteurs propres $\mathbf{u}_1, \dots, \mathbf{u}_M$ ayant les plus grandes valeurs propres $\lambda_1, \dots, \lambda_M$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- Pour $M > 1$, on y va itérativement
 - à chaque fois, on cherche une autre projection qui maximise la variance, mais qui est orthogonale aux projections précédentes
- Soit \mathbf{U} la matrice des vecteurs propres de \mathbf{S} , ordonnés par valeurs propres décroissantes, $\mathbf{y}(\mathbf{x})$ est alors

$$\mathbf{y}(\mathbf{x}) = (\mathbf{U}_{:,1:M})^T \mathbf{x}$$

Apprentissage automatique

Réduction de dimensionnalité - ACP en pratique

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

RAPPEL

- Pour $M > 1$, on y va itérativement
 - à chaque fois, on cherche une autre projection qui maximise la variance, mais qui est orthogonale aux projections précédentes
- Soit \mathbf{U} la matrice des vecteurs propres de \mathbf{S} , ordonnés par valeurs propres décroissantes, $\mathbf{y}(\mathbf{x})$ est alors

$$\mathbf{y}(\mathbf{x}) = (\mathbf{U}_{:,1:M})^T \mathbf{x}$$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

- En pratique, on commence par soustraire la moyenne des données :

$$\mathbf{y}(\mathbf{x}) = (\mathbf{U}_{:,1:M})^T (\mathbf{x} - \bar{\mathbf{x}})$$

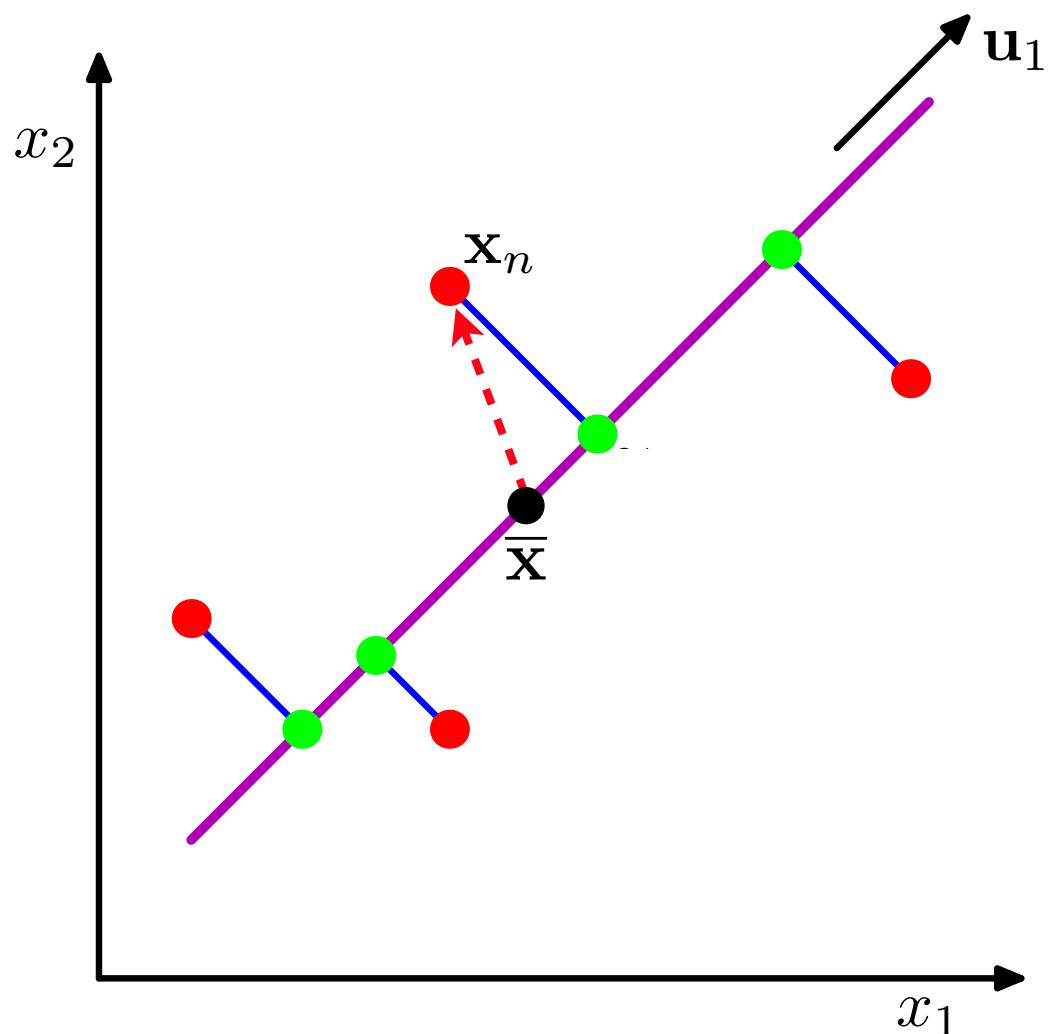
- Permet de centrer les données projetées

$$\frac{1}{N} \sum_n (\mathbf{U}_{:,1:M})^T (\mathbf{x}_n - \bar{\mathbf{x}}) = (\mathbf{U}_{:,1:M})^T \left(\frac{1}{N} \sum_n \mathbf{x}_n - \bar{\mathbf{x}} \right) = 0$$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: compression avec ACP

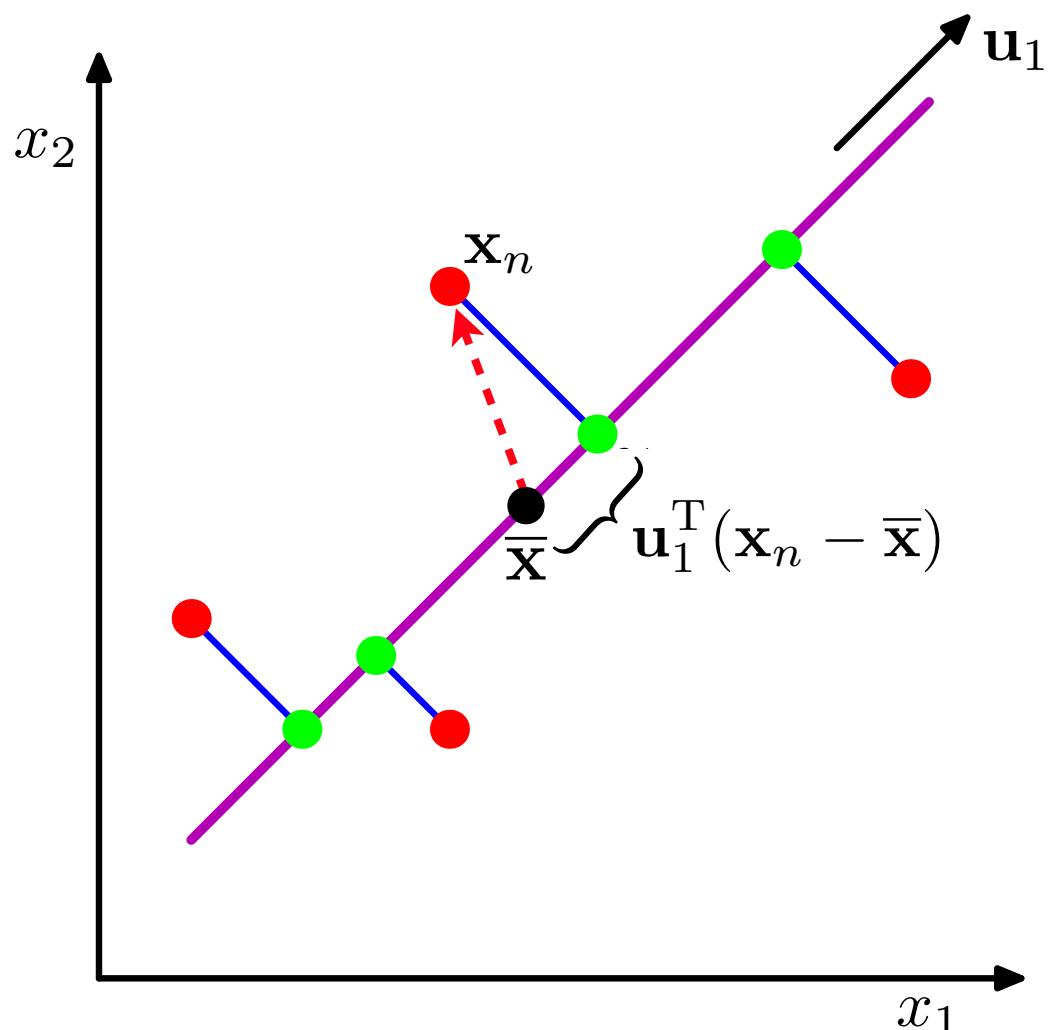
- On peut utiliser l'ACP pour compresser les données
 - on peut décompresser en multipliant chaque dimension par son vecteur u_i



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: compression avec ACP

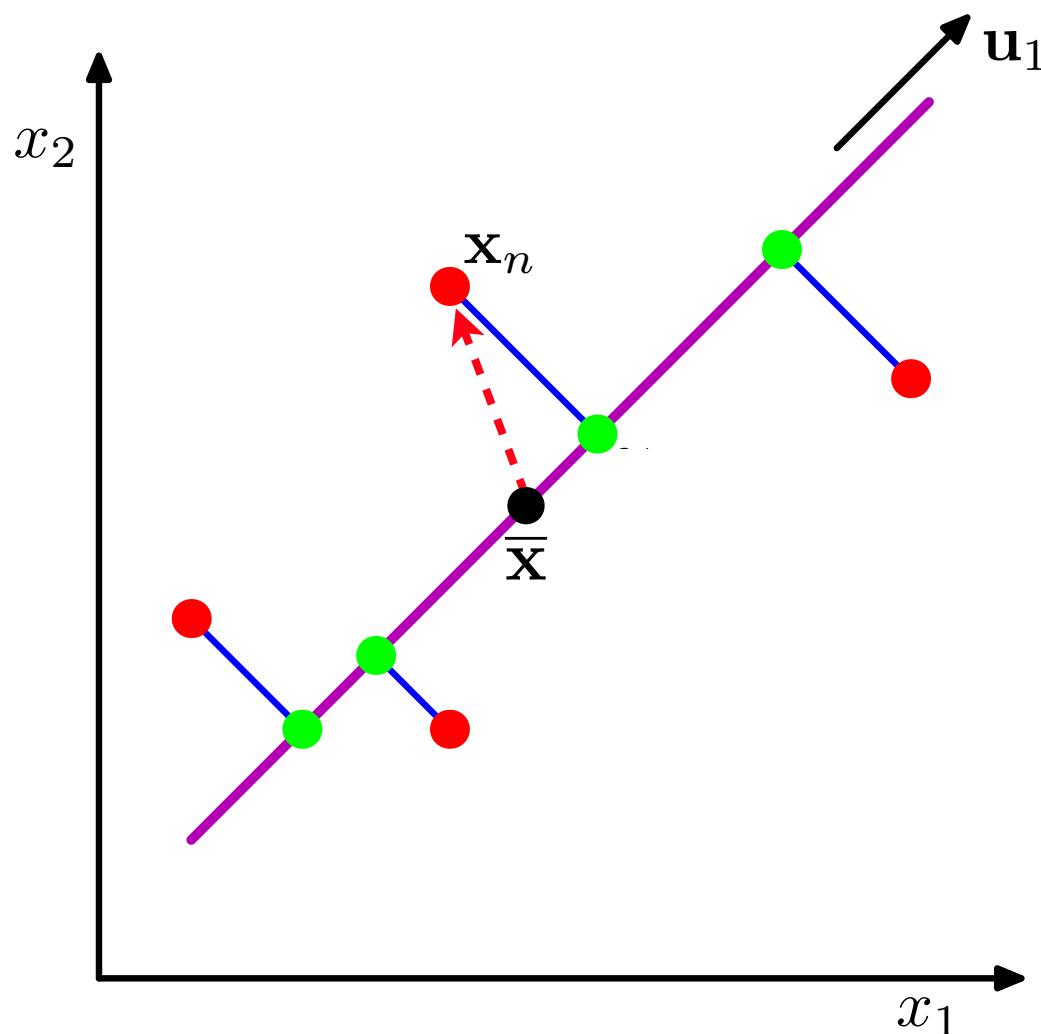
- On peut utiliser l'ACP pour compresser les données
 - on peut décompresser en multipliant chaque dimension par son vecteur \mathbf{u}_i



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: compression avec ACP

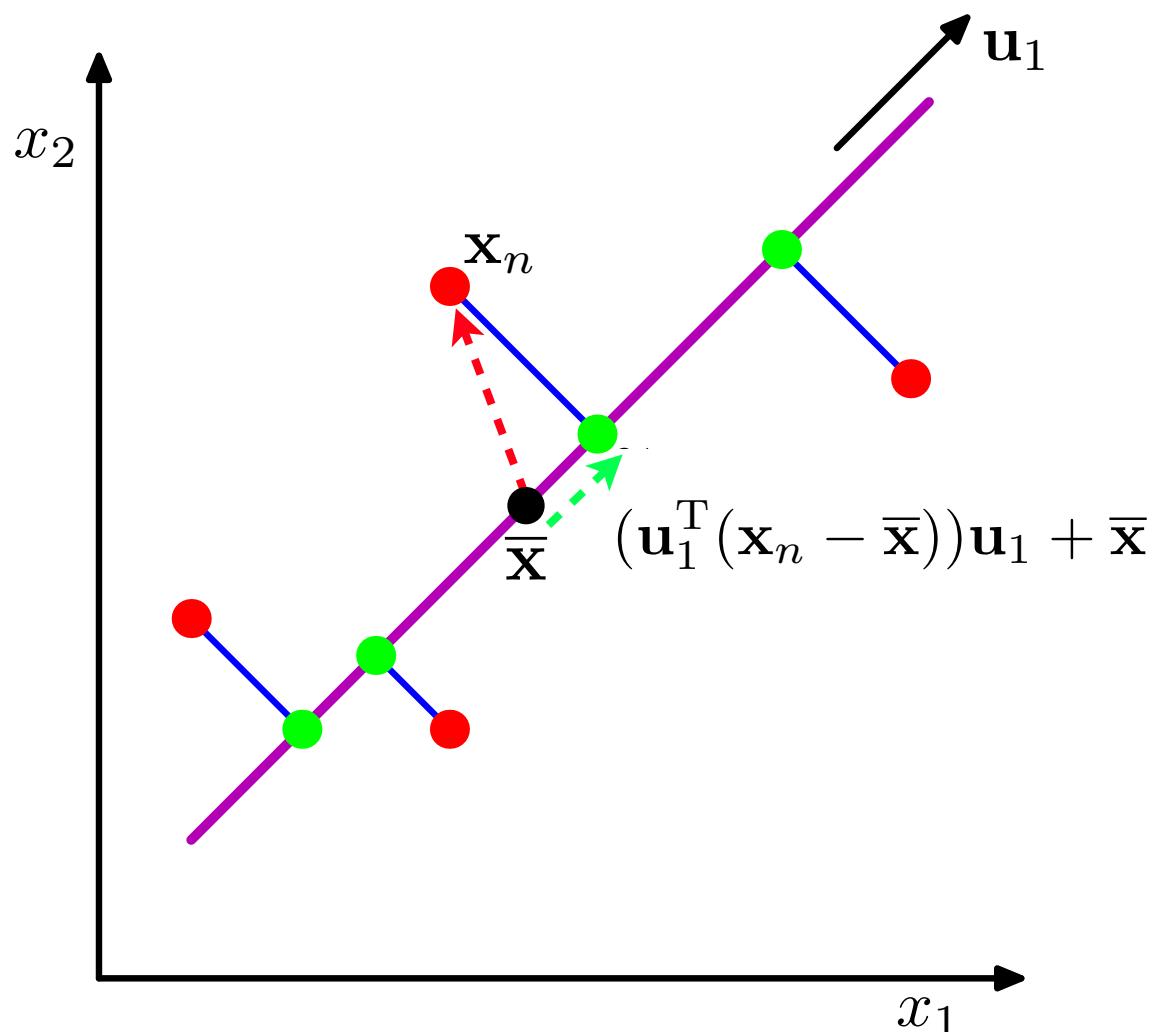
- On peut utiliser l'ACP pour compresser les données
 - on peut décompresser en multipliant chaque dimension par son vecteur u_i



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: compression avec ACP

- On peut utiliser l'ACP pour compresser les données
 - on peut décompresser en multipliant chaque dimension par son vecteur \mathbf{u}_i



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: compression avec ACP

- On peut utiliser l'ACP pour compresser les données
 - on peut décompresser en multipliant chaque dimension par son vecteur \mathbf{u}_i
 - de façon générale :

$$\bar{\mathbf{x}} + \sum_{i=1}^M (\mathbf{u}_i^T (\mathbf{x}_n - \bar{\mathbf{x}})) \mathbf{u}_i$$

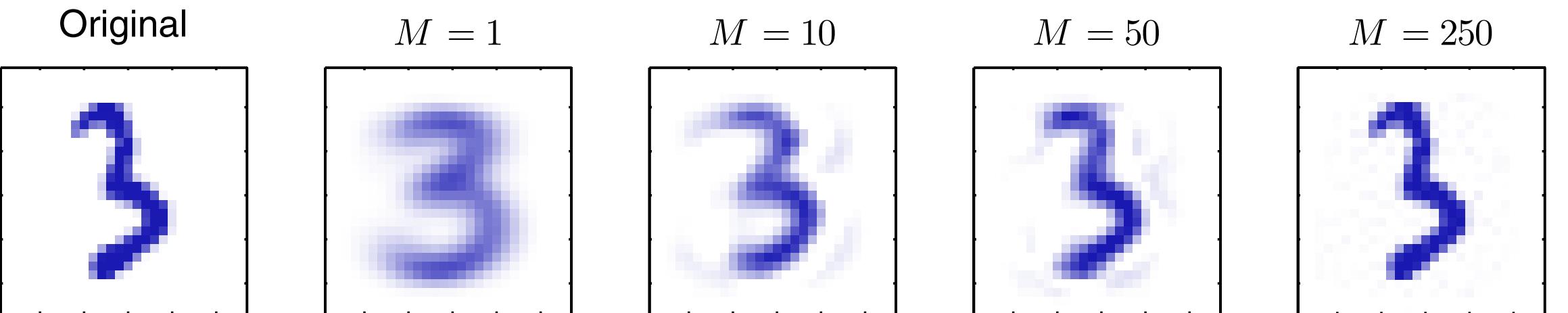
ou

$$\bar{\mathbf{x}} + (\mathbf{U}_{:,1:M})(\mathbf{U}_{:,1:M})^T (\mathbf{x} - \bar{\mathbf{x}})$$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: compression avec ACP

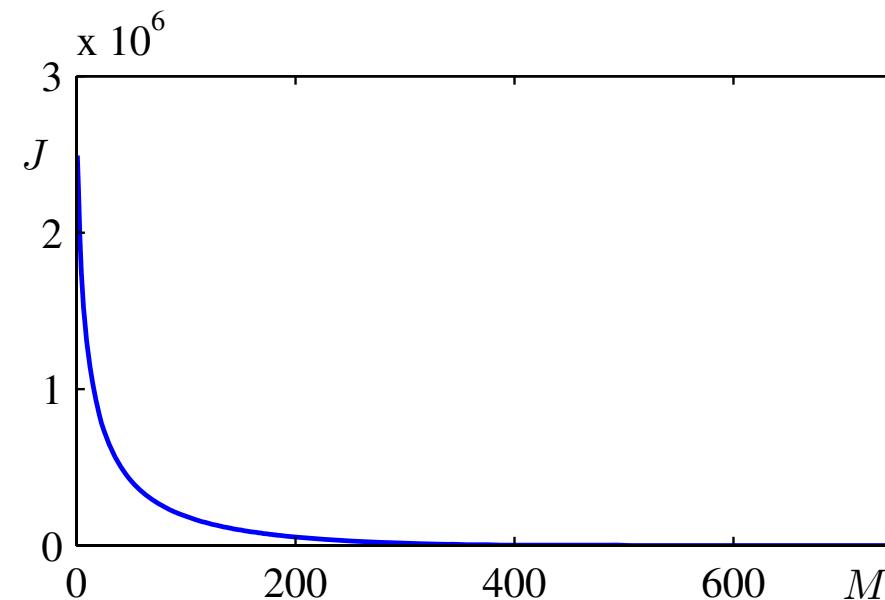
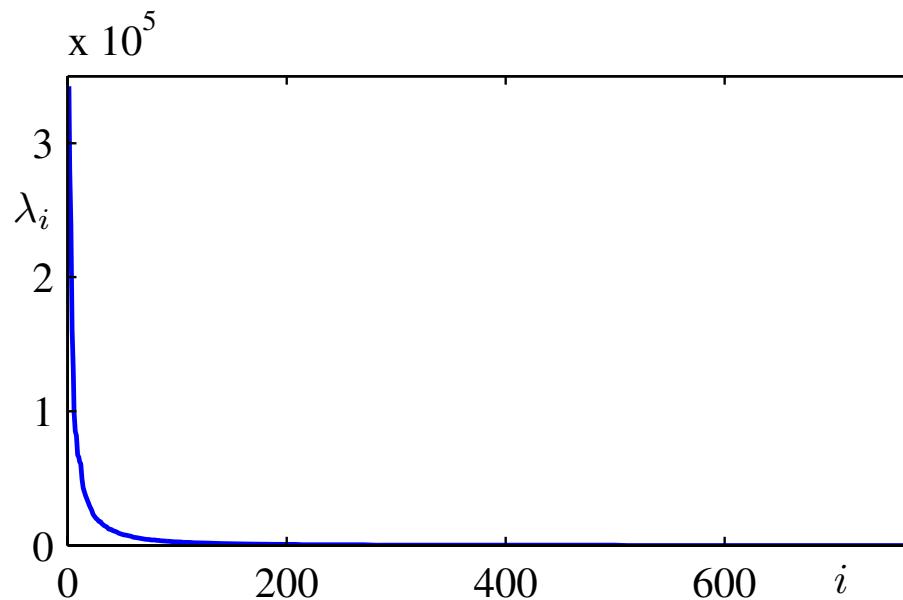
- Exemple : compression d'images de «3» avec M croissant



ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: erreur de compression et valeurs propres

- Exemple : compression d'images de «3» avec M croissant
 - il y a un lien intime entre l'erreur de compression d'entraînement et la valeurs propres de S
 - on peut montrer que l'erreur est égal à la somme des valeurs propres de $i=M+1$ à D (voir section 12.1.2 du livre)



$$J = \sum_{i=M+1}^D \lambda_i$$

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: normalisation

- En plus de center les données, on divise chaque dimension par son écart type (empirique)

- La variance empirique de la projection $\mathbf{u}_i^T \mathbf{x}$ est

$$\mathbf{u}_i^T \mathbf{S} \mathbf{u}_i = \lambda_i$$

- On utilise alors la transformation

$$\mathbf{y}(\mathbf{x}) = \Lambda_{1:M, 1:M}^{-1/2} (\mathbf{U}_{:, 1:M})^T (\mathbf{x} - \bar{\mathbf{x}})$$

$$\Lambda^{-1/2} = \begin{pmatrix} \frac{1}{\sqrt{\lambda_1}} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{\sqrt{\lambda_D}} \end{pmatrix}$$

Apprentissage automatique

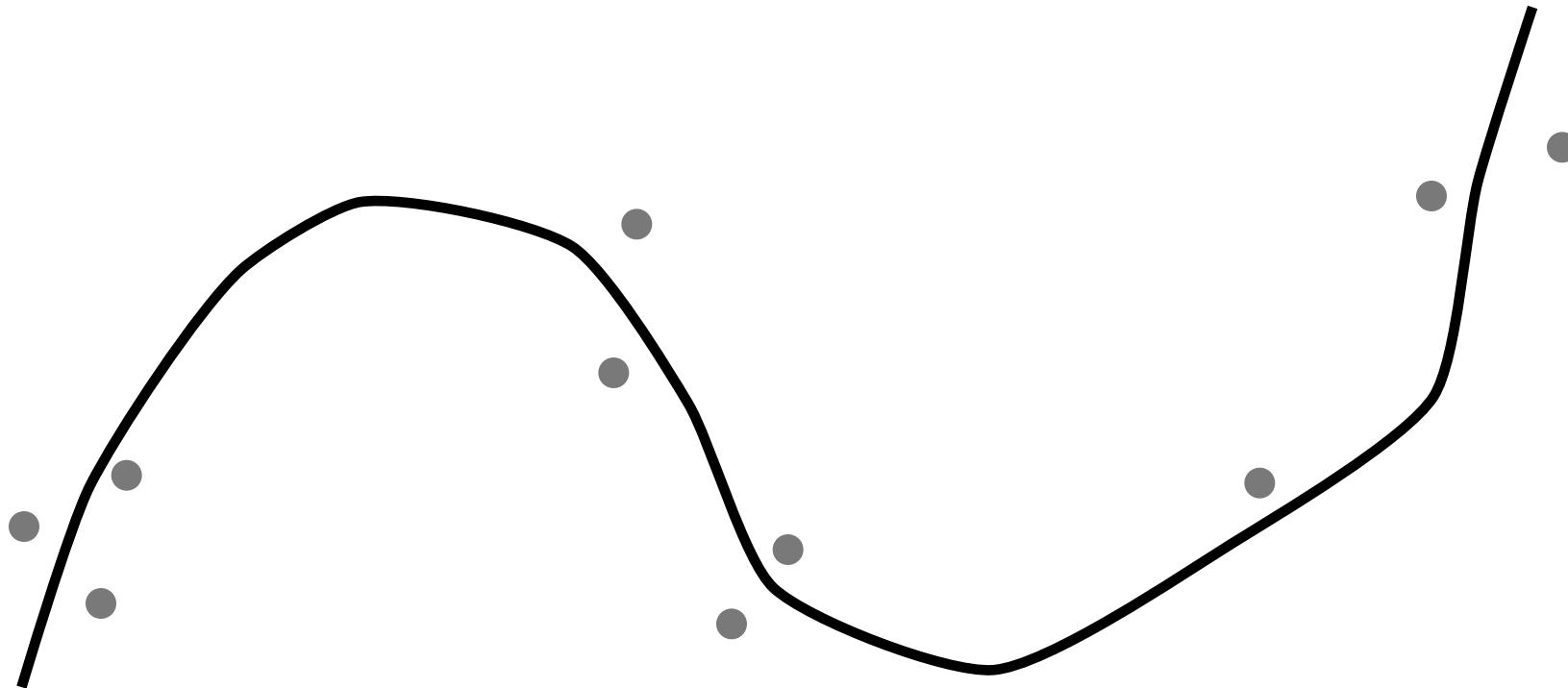
Réduction de dimensionnalité - ACP à noyau

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété, *manifold*

RAPPEL

- De façon générale, lorsque D est grand, on s'attend à ce que les données se trouvent surtout autour d'une **variété (*manifold*)** de dimensionnalité $M < D$

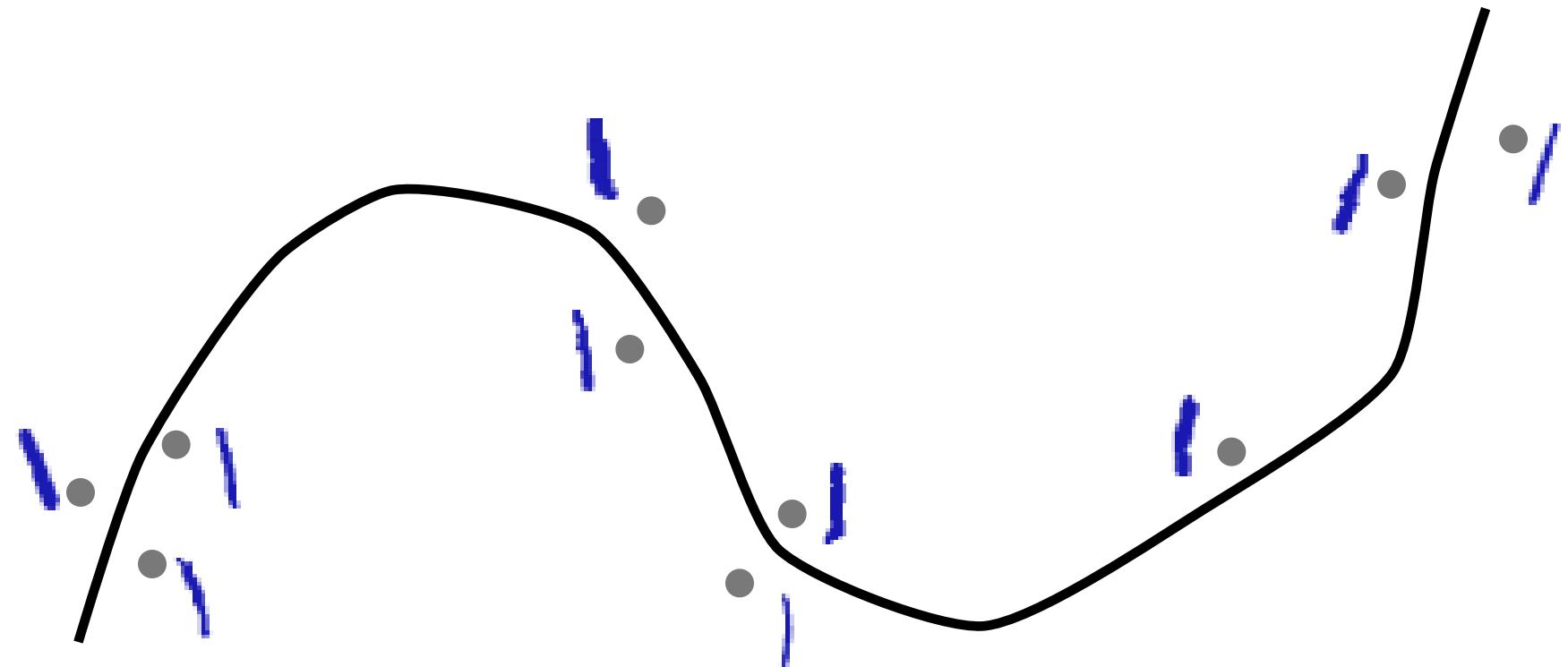


RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété, *manifold*

RAPPEL

- De façon générale, lorsque D est grand, on s'attend à ce que les données se trouvent surtout autour d'une **variété (*manifold*)** de dimensionnalité $M < D$

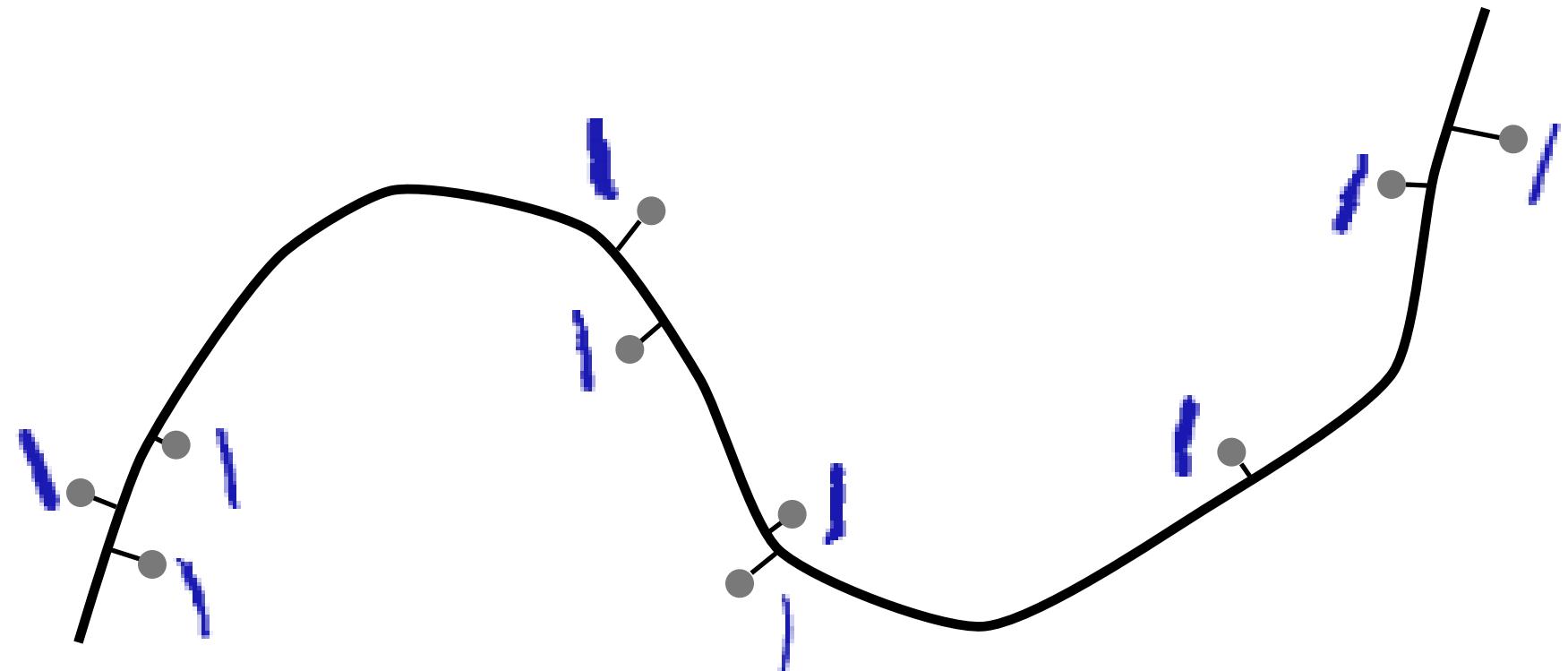


RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété, *manifold*

RAPPEL

- De façon générale, lorsque D est grand, on s'attend à ce que les données se trouvent surtout autour d'une **variété (*manifold*)** de dimensionnalité $M < D$

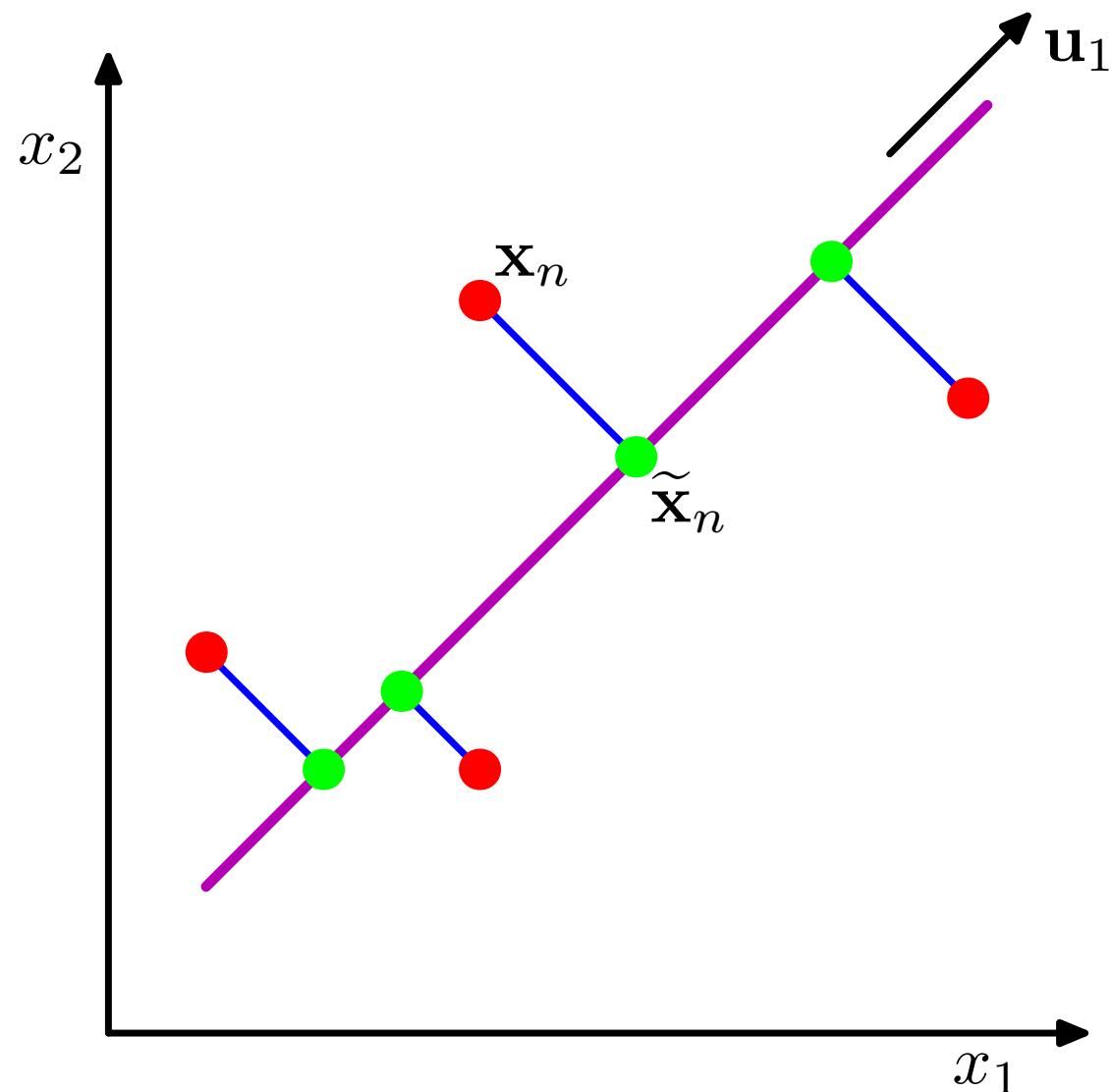


ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: analyse en composantes principales

RAPPEL

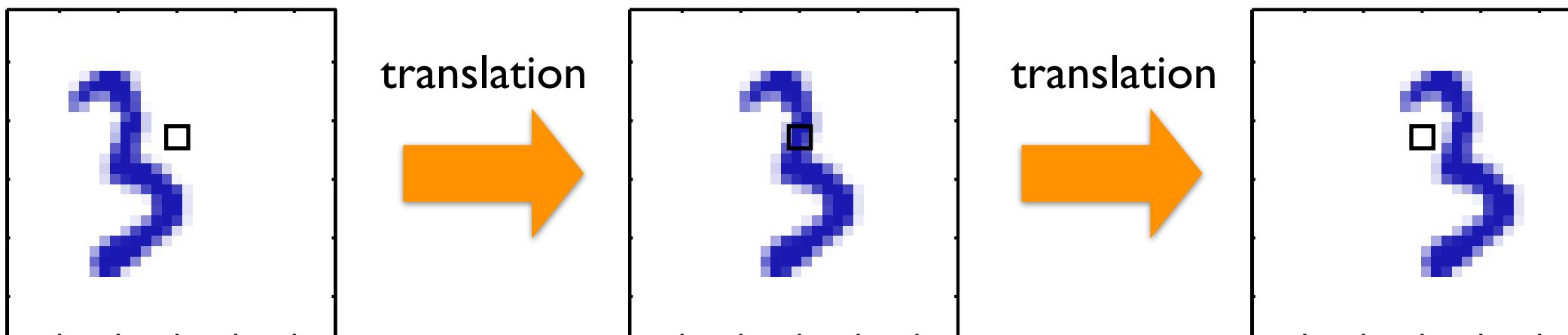
- Exemple : de $D=2, M=1$



RÉDUCTION DE DIMENSIONNALITÉ

Sujets: variété non-linéaire

- Il est fort probable que la variété sous-jacente soit non-linéaire
 - pour des images, même la translation est non-linéaire



- Si la variété est non-linéaire, alors projeter sur cette variété sera aussi une opération non-linéaire

ACP À NOYAU

Sujets: ACP à noyau

- Comment obtenir une version non-linéaire de l'ACP ?
 - avec l'astuce de noyau !

1. On représente nos entrées sous une forme $\phi(\mathbf{x}_n)$

2. On formule l'algorithme de façon à seulement avoir à calculer $k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$

ACP À NOYAU

Sujets: ACP à noyau

- Commençons par supposer que les données (transformées) sont centrées ($\sum_n \phi(\mathbf{x}_n) = 0$)
- La matrice de covariance empirique est alors

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T$$

et on cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- Équivaut à ce que \mathbf{v}_i satisfasse

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \left\{ \phi(\mathbf{x}_n)^T \mathbf{v}_i \right\} = \lambda_i \mathbf{v}_i$$

- On peut donc écrire \mathbf{v}_i sous la forme

$$\mathbf{v}_i = \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n)$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- On remplace \mathbf{v}_i sous cette forme pour obtenir

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \sum_{m=1}^N a_{im} \phi(\mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n)$$

- On multiplie par $\phi(\mathbf{x}_l)^T$ des deux côtés ($\mathbf{x}_l \in \mathcal{D}$)

$$\frac{1}{N} \sum_{n=1}^N k(\mathbf{x}_l, \mathbf{x}_n) \sum_{m=1}^N a_{im} k(\mathbf{x}_n, \mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} k(\mathbf{x}_l, \mathbf{x}_n)$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- On remplace \mathbf{v}_i sous cette forme pour obtenir

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \sum_{m=1}^N a_{im} \phi(\mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n)$$

- On multiplie par $\phi(\mathbf{x}_l)^T$ des deux côtés ($\mathbf{x}_l \in \mathcal{D}$)

$$\frac{1}{N} \sum_{n=1}^N k(\mathbf{x}_l, \mathbf{x}_n) \sum_{m=1}^N a_{im} k(\mathbf{x}_n, \mathbf{x}_m) = \lambda_i \mathbf{K}_{l,:} \mathbf{a}_i$$

\mathbf{K} est la matrice
de Gram

$$\mathbf{K}_{n,m} = k(\mathbf{x}_n, \mathbf{x}_m)$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- On remplace \mathbf{v}_i sous cette forme pour obtenir

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \sum_{m=1}^N a_{im} \phi(\mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n)$$

- On multiplie par $\phi(\mathbf{x}_l)^T$ des deux côtés ($\mathbf{x}_l \in \mathcal{D}$)

$$\frac{1}{N} \sum_{n=1}^N k(\mathbf{x}_l, \mathbf{x}_n) \mathbf{K}_{n,:} \mathbf{a}_i = \lambda_i \mathbf{K}_{l,:} \mathbf{a}_i$$

\mathbf{K} est la matrice
de Gram

$$\mathbf{K}_{n,m} = k(\mathbf{x}_n, \mathbf{x}_m)$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- On remplace \mathbf{v}_i sous cette forme pour obtenir

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \sum_{m=1}^N a_{im} \phi(\mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n)$$

- On multiplie par $\phi(\mathbf{x}_l)^T$ des deux côtés ($\mathbf{x}_l \in \mathcal{D}$)

$$\frac{1}{N} \mathbf{K}_{l,:} \mathbf{K} \mathbf{a}_i = \lambda_i \mathbf{K}_{l,:} \mathbf{a}_i$$

\mathbf{K} est la matrice
de Gram

$$\mathbf{K}_{n,m} = k(\mathbf{x}_n, \mathbf{x}_m)$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- On génère N équations en considérant n'importe quel \mathbf{x}_l de l'ensemble d'entraînement

$$\mathbf{K}^2 \mathbf{a}_i = \lambda_i N \mathbf{K} \mathbf{a}_i$$

- En multipliant par \mathbf{K}^{-1} , on obtient

$$\mathbf{K} \mathbf{a}_i = \lambda_i N \mathbf{a}_i$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- Pour obtenir les \mathbf{a}_i , on trouve les M vecteurs propres (\mathbf{a}_i) de \mathbf{K} ayant les plus grandes valeurs propres ($\lambda_i N$)

$$\mathbf{K}\mathbf{a}_i = \lambda_i N \mathbf{a}_i$$

ACP À NOYAU

Sujets: ACP à noyau

- On cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

- Finalement, on doit s'assurer que les \mathbf{v}_i soient de norme 1

$$1 = \mathbf{v}_i^T \mathbf{v}_i = \sum_{n=1}^N \sum_{m=1}^N a_{in} a_{im} \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = \mathbf{a}_i^T \mathbf{K} \mathbf{a}_i = \lambda_i N \mathbf{a}_i^T \mathbf{a}_i$$

- On divise les \mathbf{a}_i par la racine carré des valeurs propres $\lambda_i N$

$$\mathbf{a}_i \leftarrow \frac{\mathbf{a}_i}{\sqrt{\lambda_i N}}$$

ACP À NOYAU

Sujets: calcul de la projection

- On peut finalement calculer chaque élément $y(\mathbf{x})_i$ de la projection $\mathbf{y}(\mathbf{x})$

$$y_i(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{v}_i = \sum_{n=1}^N a_{in} \phi(\mathbf{x})^T \phi(\mathbf{x}_n) = \sum_{n=1}^N a_{in} k(\mathbf{x}, \mathbf{x}_n)$$

ou, avec \mathbf{A} telle que chaque rangée correspond aux \mathbf{a}_i

$$\mathbf{y}(\mathbf{x}) = \mathbf{A} \mathbf{k}(\mathbf{x})$$

Apprentissage automatique

Réduction de dimensionnalité - centrage du noyau

ACP À NOYAU

Sujets: ACP à noyau

RAPPEL

- Commençons par supposer que les données (transformées) sont centrées ($\sum_n \phi(\mathbf{x}_n) = 0$)
- La matrice de covariance empirique est alors

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T$$

et on cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

ACP À NOYAU

Sujets: ACP à noyau

RAPPEL

- Commençons par supposer que les données (transformées) sont centrées ($\sum_n \phi(\mathbf{x}_n) = 0$)
- La matrice de covariance empirique est alors

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T$$

et on cherche ses vecteurs propres \mathbf{v}_i :

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

ACP À NOYAU

Sujets: centrage du noyau

- On a supposé que les $\phi(\mathbf{x}_n)$ sont centrés
 - pour un noyau $k(\mathbf{x}_n, \mathbf{x}_m)$ donné, c'est probablement pas le cas
- Il faudrait soustraire la moyenne, dans l'espace des $\phi(\mathbf{x}_n)$

$$\tilde{\phi}(\mathbf{x}_n) = \phi(\mathbf{x}_n) - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_l)$$

- par contre, on ne peut pas travailler avec les $\phi(\mathbf{x}_n)$ directement, puisqu'ils peuvent être de taille infinie

ACP À NOYAU

Sujets: centrage du noyau

- On veut travailler avec la matrix \tilde{K} de Gram telle que

$$\tilde{K}_{nm} = \tilde{\phi}(\mathbf{x}_n)^T \tilde{\phi}(\mathbf{x}_m)$$

ACP À NOYAU

Sujets: centrage du noyau

- On veut travailler avec la matrix \tilde{K} de Gram telle que

$$\begin{aligned}\tilde{K}_{nm} &= \tilde{\phi}(\mathbf{x}_n)^T \tilde{\phi}(\mathbf{x}_m) \\ &= \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_l) \\ &\quad - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_l)^T \phi(\mathbf{x}_m) + \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_l)\end{aligned}$$

ACP À NOYAU

Sujets: centrage du noyau

- On veut travailler avec la matrix \tilde{K} de Gram telle que

$$\tilde{K}_{nm} = \tilde{\phi}(\mathbf{x}_n)^T \tilde{\phi}(\mathbf{x}_m)$$

$$= \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_l)$$

$$- \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_l)^T \phi(\mathbf{x}_m) + \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_l)$$

$$= k(\mathbf{x}_n, \mathbf{x}_m) - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_l, \mathbf{x}_m)$$

$$- \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_n, \mathbf{x}_l) + \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N k(\mathbf{x}_j, \mathbf{x}_l)$$

ACP À NOYAU

Sujets: centrage du noyau

- On peut calculer $\tilde{\mathbf{K}}$ comme suit :

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N \mathbf{K} - \mathbf{K} \mathbf{1}_N + \mathbf{1}_N \mathbf{K} \mathbf{1}_N$$

où $\mathbf{1}_N$ est une matrice $N \times N$ où tous les éléments sont $1/N$

- On appelle cette opération **centrer le noyau**

Apprentissage automatique

Réduction de dimensionnalité - résumé

ANALYSE EN COMPOSANTES PRINCIPALES

Sujets: résumé de l'ACP

- Modèle : $\mathbf{y}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$
- Entraînement : on maximise la variance des $y(\mathbf{x})_i$
 - extraire les vecteurs propres \mathbf{U} et valeurs propres Λ de \mathbf{S}
- Hyper-paramètres : M
- Prédiction : $\mathbf{y}(\mathbf{x}) = \Lambda_{1:M,1:M}^{-1/2} (\mathbf{U}_{:,1:M})^T (\mathbf{x} - \bar{\mathbf{x}})$

$$\mathbf{W} = \Lambda_{1:M,1:M}^{-1/2} (\mathbf{U}_{:,1:M})^T$$

$$\mathbf{b} = -\Lambda_{1:M,1:M}^{-1/2} (\mathbf{U}_{:,1:M})^T \bar{\mathbf{x}}$$

ACP À NOYAU

Sujets: résumé de l'ACP à noyau

- Modèle : $\mathbf{y}(\mathbf{x}) = \mathbf{W}\phi(\mathbf{x}) = \mathbf{A}\mathbf{k}(\mathbf{x})$
- Entraînement : on maximise la variance des $y(\mathbf{x})_i$ (implicitement)
 - extraire les M vecteurs propres \mathbf{a}_i avec plus grandes valeurs propres $\tilde{\lambda}_i$ ($\lambda_i N$) de la matrice de Gram **centrée** $\tilde{\mathbf{K}}$
 - $\mathbf{a}_i \leftarrow \frac{\mathbf{a}_i}{\sqrt{\tilde{\lambda}_i}}$
 - construire \mathbf{A} en empilant les \mathbf{a}_i en rangées
- Hyper-paramètres : M et ceux du noyau
- Prédiction : $\mathbf{y}(\mathbf{x}) = \mathbf{A}\mathbf{k}(\mathbf{x})$

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: choisir les hyper-paramètres

- Pour la visualisation
 - M : 2 ou 3
 - pas vraiment de choix ici, puisqu'on peut seulement visualiser en 2D ou 3D
 - hyper-paramètres du noyau (ACP à noyau) : essai et erreur
 - on tente différentes valeurs, où chaque choix est une «fenêtre» sur les données

RÉDUCTION DE DIMENSIONNALITÉ

Sujets: choisir les hyper-paramètres

- Pour réduire le sur-apprentissage d'un autre algorithme (par exemple de classification), qui prend $y(x)$ en entrée
 - M et les hyper-paramètres de noyau : sélection de modèle
 - on fait comme pour les autres hyper-paramètres de l'autre algorithme, et on tente de maximiser la performance de généralisation de cet algorithme par rapport aux hyper-paramètres de réduction de dimensionnalité
- Alternative, pour l'ACP (linéaire) : choisir M telle que l'erreur de compression (J) est moins de 1%
 - utile si on veut simplement réduire la taille des données, pour accélérer les calculs

ACP À NOYAU

Sujets: extraction de caractéristique

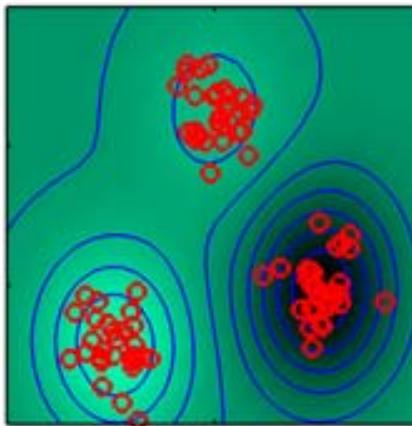
- On note que, pour l'ACP à noyau, M pourrait être plus grand que D
 - c'est la dimensionnalité de $\phi(\mathbf{x}_n)$ qu'on réduit
 - si le noyau est gaussien, la dimensionnalité de $\phi(\mathbf{x}_n)$ est infinie
- On peut donc aussi utiliser l'ACP à noyau pour faire de **l'extraction de caractéristique**
 - la représentation non-linéaire $y(\mathbf{x})$ est possiblement plus riche et utile

ACP À NOYAU

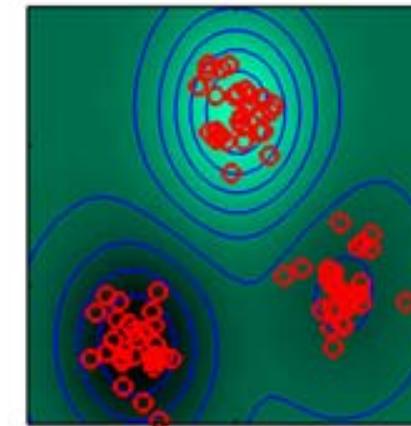
Sujets: extraction de caractéristique

- Exemple :

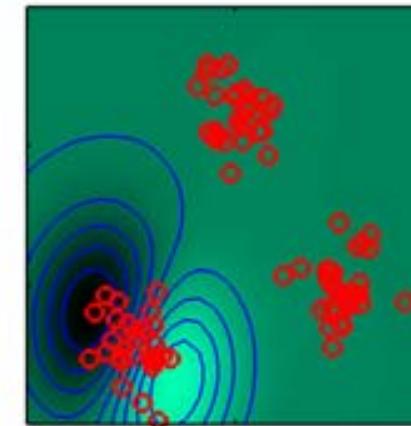
Eigenvalue=21.72



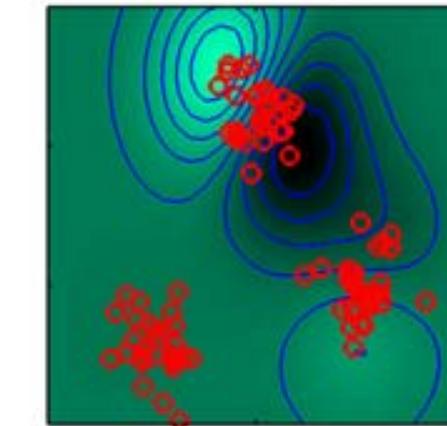
Eigenvalue=21.65



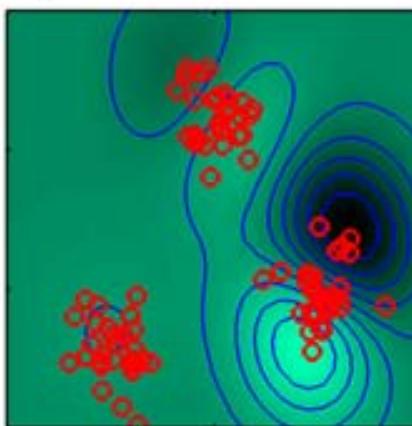
Eigenvalue=4.11



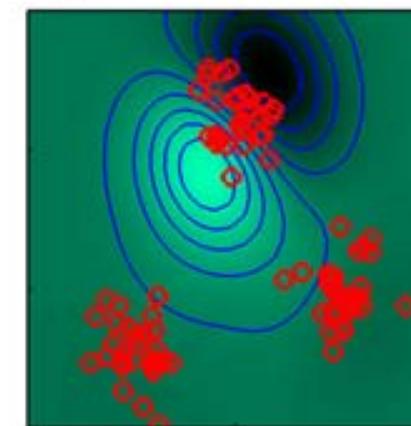
Eigenvalue=3.93



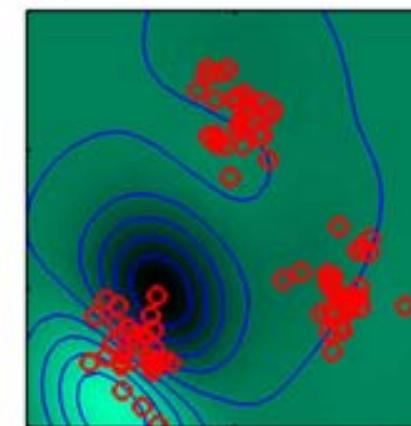
Eigenvalue=3.66



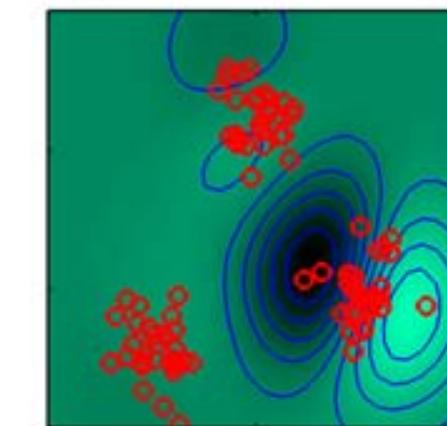
Eigenvalue=3.09



Eigenvalue=2.60



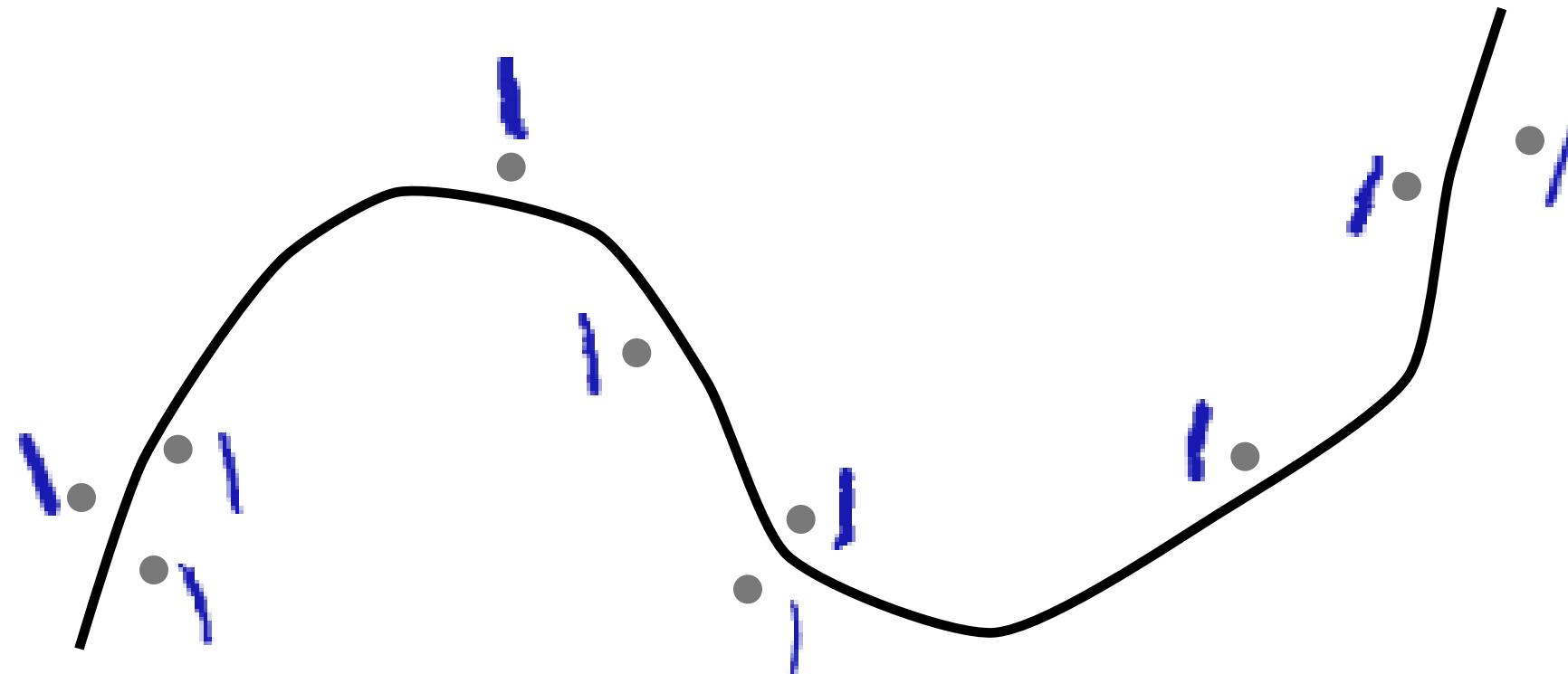
Eigenvalue=2.53



ISOMAP

Sujets: Isomap

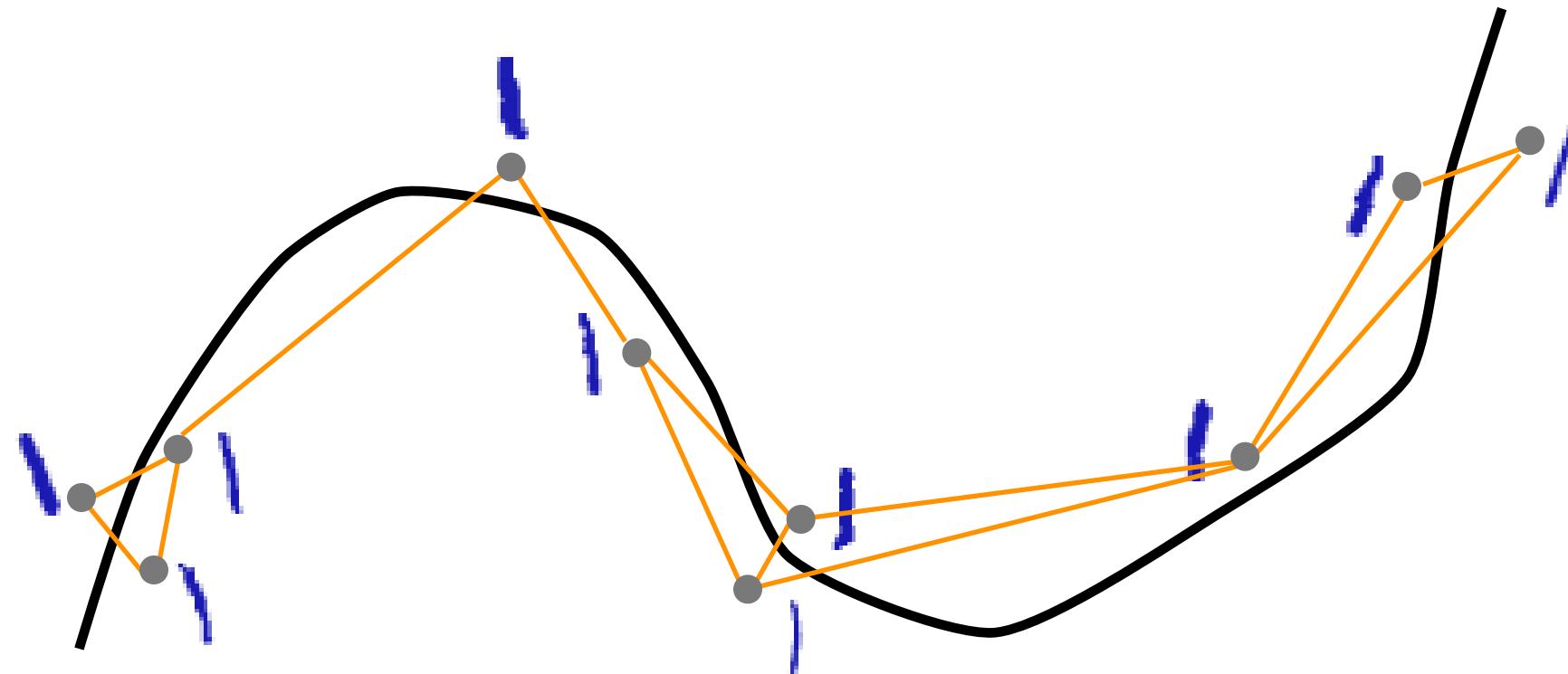
- Pour la réduction de dimensionnalité, les noyaux standards ne fonctionnent pas toujours bien
 - Algorithme **Isomap** : estimer la distance sur la variété à l'aide de Dijkstra, et utiliser cette distance pour dériver un noyau



ISOMAP

Sujets: Isomap

- Pour la réduction de dimensionnalité, les noyaux standards ne fonctionnent pas toujours bien
 - Algorithme **Isomap** : estimer la distance sur la variété à l'aide de Dijkstra, et utiliser cette distance pour dériver un noyau



TYPES D'APPRENTISSAGE

Sujets: apprentissage non-supervisé, visualisation

RAPPEL

- L'apprentissage non-supervisé est lorsqu'une cible n'est pas explicitement donnée
 - visualisation de données

Tenenbaum, de Silva,
Langford, (2000)

