COURS DE DEUXIÈME ANNÉE
INGÉNIEUR ISIMA

Machine-Learning
Réduction de dimension

Vincent Barra

ISIMA

Table des matières

On s'intéresse ici à n individus $x_i, 1 \leq i \leq n$ que l'on suppose vivre dans $\mathbb{R}^d$. Chaque x_i est donc décrit par un ensemble de d valeurs quantitatives, ou caractéristiques (features). Avec l'avènement des Big Data, et la généralisation des capteurs, d peut être très grand (plusieurs milliers), et analyser telles quelles les données brutes devient difficile d'un point de vue calculatoire et interprétation. De plus, il est rare que les caractéristiques soient totalement utiles et indépendantes.
Une étape souvent utilisée en analyse de données consiste donc à prétraiter cet espace, par exemple pour :

— le transformer en un format compatible avec des algorithmes qui seront utilisés
— réduire la complexité temporelle des algorithmes qui seront utilisés
— réduire la complexité spatiale du problème traité
— découpler des variables et chercher les dépendances

— introduire des a priori, ou des propriétés importantes pour les algorithmes (données centrées normées, descripteurs épars...)
— permettre une interprétation plus intuitive et/ou graphique
— ...

Deux approches sont alors classiquement utilisées : soit sélectionner parmi les caractéristiques initiales, celles jugées les plus pertinentes, et éliminer les autres ; soit construire, à partir de ces valeurs quantitatives, de nouvelles caractéristiques par des transformations linéaires, ou non linéaires.

1 SÉLECTION DE CARACTÉRISTIQUES

1.1 Définitions

La sélection de caractéristiques consiste à choisir parmi les d descripteurs d'un ensemble d'individus $x_i, 1 \leq i \leq n$, un sous-ensemble de $t < d$ caractéristiques jugées "les plus pertinentes", les $d-t$ restantes étant ignorées.
On note $F = (f_1 \cdots f_d)$ les d caractéristiques. On note $Perf$ une fonction qui permet d'évaluer un sous-ensemble de caractéristiques, et on suppose que $Perf$ atteint son maximum pour le meilleur sous-ensemble de caractéristiques ("le plus pertinent"). Le problème de sélection se formule donc comme un problème d'optimisation

$$\hat{F} = Arg \max_{U \subset F} Perf(U)$$

le cardinal $|\hat{F}|$ de $\hat{F}$ étant soit contrôlé par l'utilisateur, soit défini par l'algorithme de sélection.
On distingue alors trois stratégies :

— $|\hat{F}|$ est défini par l'utilisateur et l'optimisation s'effectue sur tous les sous-ensembles ayant ce cardinal
— on connaît une mesure minimale de performance γ et la sélection recherche le plus petit sous-ensemble U dont la performance $Perf(U)$ est supérieure ou égale à γ
— On cherche un compromis entre l'amélioration de la performance $Perf(U)$ et la réduction de la taille du sous ensemble.

La mesure de pertinence d'une caractéristique est donc au centre des algorithmes de sélection. Plusieurs définitions sont possibles, et nous dirons ici qu'une caractéristique f_i est :

— pertinente si son absence entraîne une détérioration significative de la performance de l'algorithme utilisé en aval (classification régression)
— peu pertinente si elle n'est pas pertinente et s'il existe un sous-ensemble U tel que la performance de $U \cup \{f_i\}$ est significativement meilleure que la peformance de U
— non pertinente, si elle ne rentre pas dans les deux premières définitions. En général, ces caractéristiques sont supprimées.

1.2 Caractéristiques des méthodes de sélection

Une méthode de sélection basée sur l'optimisation de $Perf$ utilise généralement trois étapes. Les deux dernières sont itérées jusqu'à un test d'arrêt.

1.2.1 Initialisation

L'initialisation consiste à choisir l'ensemble de départ des caractéristiques. Il peut s'agit de l'ensemble vide, de F tout entier, ou un sous-ensemble quelconque $U \subset F$

1.2.2 Exploration des sous-ensembles

A partir de cette initialisation, les stratégies d'exploration des sous-ensembles de caractéristiques se déclinent en trois catégories :

1. génération exhaustive : tous les sous-ensembles de caractéristiques sont évalués. Si elle garantit de trouver la valeur optimale, cette méthode n'est que peu applicable dès que $|F|$ devient important ($2^{|F|}$ sous-ensembles possibles)
2. génération heuristique : une génération itérative est effectuée, chaque itération permettant de sélectionner ou de rejeter une ou plusieurs caractéristiques. La génération peut être ascendante (ajout de caractéristiques à partir de l'ensemble vide), descendante (suppression de caractéristiques à partir de F), ou mixte.
3. génération stochastique : pour un ensemble de données et une initialisation définie, une stratégie de recherche heuristique retourne toujours le même sous-ensemble, ce qui la rend très sensible au changement de l'ensemble de données. La génération stochastique génère aléatoirement un nombre fini de sous-ensembles de caractéristiques afin de sélectionner le meilleur. La convergence est sous-optimale mais peut s'avérer préférable dans des algorithmes d'apprentissage, par exemple pour éviter le phénomène d'overfitting.

1.2.3 Evaluation des sous-ensembles

Filtres Le critère d'évaluation utilisé évalue la pertinence d'une caractéristique selon des mesures qui reposent sur les propriétés des données d'apprentissage.
Pour n exemples $x_i, 1 \leq i \leq n$, on note $x_i = (x_{i1} \cdots x_{id})^T \in \mathbb{R}^d$ une donnée d'apprentissage (la j^e caractéristique f_j ayant donc pour valeur x_{ij}) , d'étiquette y_i (en classification ou régression). Les méthodes de type filtres calculent un score pour évaluer le degré de pertinence de chacune des caractéristiques f_i , parmi lesquelles on peut citer

— Le critère de corrélation, utilisé en classification binaire :

$$C_i \frac{\sum_{k=1}^{n} (x_{ki} - \mu_i)\left(y_k - \mu_y\right)}{\sqrt{\sum_{k=1}^{n} (x_{ki} - \mu_i)^2 \sum_{k=1}^{n} \left(y_k - \mu_y\right)^2}}$$

où μ_i (resp. μ_k) est la moyenne de la caractéristique f_i observée sur $x_1 \cdots x_n$ (resp. moyenne des étiquettes)

— Le critère de Fisher, qui permet de mesurer dans un problème de classification multiclasses le degré de séparabilité des classes à l'aide d'une caractéristique donnée

$$F_i = \frac{\sum_{c=1}^{C} n_c \left(\mu_c^i - \mu_i\right)^2}{\sum_{c=1}^{C} n_c (\sigma_c^i)^2}$$

où n_c, μ_c^i et σ_c^i sont l'effectif, la moyenne et l'écart-type de la caractéristique f_i dans la classe c

— l'information mutuelle

$$I(i) = \sum_{x_i} \sum_{y} P(X = x_i, Y = y) log \left(\frac{P(X = x_i, Y = y)}{P(X = x_i)P(Y = y)} \right)$$

qui mesure la dépendance entre les distributions de deux populations. Ici X et Y sont deux variables aléatoires dont les réalisations sont les valeurs de f_i et des étiquettes de classes. Les probabilités sont estimées de manière fréquentiste.

Méthodes enveloppantes Le principal inconvénient des approches précédentes est le fait qu'elles ignorent l'influence des caractéristiques sélectionnées sur la performance de l'algorithme à utiliser par la suite. Les méthodes de type enveloppantes (wrappers) évaluent un sous-ensemble de caractéristiques par sa performance de classification en utilisant un algorithme d'apprentissage. Les sous-ensembles de caractéristiques sélectionnés par cette méthode sont bien adaptés à l'algorithme de classification utilisé, mais ils ne sont pas nécessairement pour un autre. De plus, la complexité de l'algorithme d'apprentissage rend ces méthodes coûteuses.

Méthodes intégrées Les méthodes intégrées incluent la sélection de variables lors du processus d'apprentissage. Un tel mécanisme intégré pour la sélection des caractéristiques peut être trouvé, par exemple, dans les algorithmes de type SVM, AdaBoost ou dans les arbres de décision (voir cours correspondants).

1.3 Quelques méthodes de sélection

1.3.1 Algorithmes de sélection séquentielle

Les algorithmes SFS (Sequential Forward Selection [7], algorithme 1) et SBS (Sequential Backward Selection [12], algorithme 1-rouge) ont été les premiers à être proposés. Ils utilisent des approches heuristiques de recherche en partant, pour la première, d'un ensemble de caractéristiques vide et pour la secondede F tout entier.
Des variantes autour de ces algorithmes simples ont été proposées depuis et par exemple :

— il est possible à chaque itération d'inclure (ou d'exclure) un sous-ensemble de caractéristiques, plutôt qu'une seule (méthodes GSFS et GSBS)

— on peut appliquer p fois SFS puis q fois SBS, de manière itérative, avec p, q des paramètres qui peuvent évoluer au cours des itérations (algorithme SFFS et SFBS)

Algorithm 1: Algorithmes SFS et SBS en rouge

Entrées: $F = (f_1 \cdots f_d)$, taille de l'ensemble final T
Sorties: $\hat{F}$
$\hat{F} \leftarrow \emptyset$ $(\hat{F} \leftarrow F)$
pour $i = 1$ *à* T $(i = 1$ *à* $d - T)$ **faire**
  pour $j = 1$ *à* $|F|$ $(j = 1$ *à* $|\hat{F}|)$ **faire**
    Evaluer $\{f_j\} \cup \hat{F}$ $(\hat{F} \setminus \{f_j\})$
  f_{max} = meilleure caractéristique (f_{min}=moins bonne caractéristique)
  $\hat{F} \leftarrow \hat{F} \cup \{f_{max}\}, F = F \setminus f_{max}$ $(\hat{F} \setminus \hat{F}\, f_{min})$

1.3.2 Algorithme Focus

L'algorithme de filtrage Focus [1] (algorithme 2) repose sur une recherche exhaustive sur F pour trouver le sous-ensemble le plus performant de taille optimale.

Algorithm 2: Algorithme Focus

Entrées: $A = \{x_i = (x_{i1} \cdots x_{id})^T \in \mathbb{R}^d, 1 \leq i \leq n\}$, taille de l'ensemble final T, seuil ε
Sorties: $\hat{F}$
$\hat{F} \leftarrow \emptyset$
pour $i = 1$ *à* T **faire**
  pour *chaque sous-ensemble* S_i *de taille* i **faire**
    si *Inconsistance*$(A,S_i)<\varepsilon$ **alors**
      $\hat{F} \leftarrow S_i$
      Retourner $\hat{F}$

1.3.3 Algorithme relief

La méthode relief [6] en classification binaire (algorithme 3), propose de calculer une mesure globale de la pertinence des caractéristiques en accumulant la différence des distances entre des exemples d'apprentissage choisis aléatoirement et leurs plus proches voisins de la même classe et de l'autre classe.

1.3.4 Méthode SAC

L'algorithme SAC (Selection Adaptative de Caractéristiques, [5]) construit un ensemble de classificateurs SVM $(M_1 \cdots M_d)$ appris sur chacun des descripteurs et sélectionne les meilleurs par discrimination linéaire de Fisher. Pour ce faire, l'algorithme construit un vecteur dont les éléments sont les performances $Perf(M_i)$ des modèles M_i, triés par ordre décroissant. Deux moyennes $m_1(i)$ et $m_2(i)$ sont calculées, qui représentent les deux moyennes de performance d'apprentissage qui ont une valeur respectivement plus grande (plus petite) que la performance du modèle M_i. Deux variances des performances $v_1^2(i)$ et

Algorithm 3: Algorithme relief en classification binaire

Entrées: $A = \{x_i = (x_{i1} \cdots x_{id})^T \in \mathbb{R}^d, 1 \leq i \leq n\}$, nombre d'itérations T
Sorties: $w \in \mathbb{R}^d$ un vecteur de poids des caractéristiques, $w_i \in [-1,1], 1 \leq i \leq d$
pour $i = 1$ *à* d **faire**
  $w_i \leftarrow 0$
pour $i = 1$ *à* T **faire**
  Choisir aléatoirement un exemple x_k
  Chercher deux plus proches voisins de x_k, l'un (x_p) dans sa classe, l'autre (x_q) dans l'autre classe
  pour $j = 1$ *à* d **faire**
    $w_j \leftarrow w_j + \frac{1}{nT}\left(|x_{kj} - x_{qj}| - |x_{kj} - x_{pj}|\right)$

$v_2^2(i)$ sont alors calculées à partir de ces moyennes, et le sous-ensemble de caractéristiques sélectionné est celui qui maximise le discriminant de Fisher

$$\frac{|m_1(i) - m_2(i)|}{v_1^2(i) + v_2^2(i)}$$

1.3.5 Algorithmes génétiques

Les algorithmes génétiques ont été utilisés dans le domaine de la sélection de caractéristiques afin d'accélérer la recherche et d'éviter les optima locaux. Les méthodes qui utilisent les techniques génétiques ont donné de meilleurs résultats que les résultats obtenus par les autres méthodes de sélection [13].

2 Extraction de caractéristiques

Les méthodes que nous décrivons ici supposent que des points y_i sont tirés aléatoirement sur une variété $\mathcal{M}$ de dimension t, munie d'une métrique $d_{\mathcal{M}}$. Ces points sont envoyés par un plongement ψ, dans un espace $\mathcal{X}$ de dimension $d, t << d$,, muni de la métrique euclidienne, résultant en les points d'observation x_i. L'objectif des paragraphes suivants est de proposer des algorithmes permettant de retrouver $\mathcal{M}$, et d'expliciter ψ et les y_i, étant donnés soit les individus $x_i \in \mathcal{X}$, soit des informations de distances entre chaque paire de x_i.
Dit autrement, nous supposons que les variables mesurées sur les individus ne sont pas indépendantes, et qu'il est possible de représenter les x_i sur un espace de dimension réduit, variété de dimension t. On parle souvent de réduction de dimension, ou pour certaines des méthodes présentées ici de manifold learning.

2.1 Méthodes linéaires

Une variété linéaire $\mathcal{M}$ est un hyperplan. Les données observées sont dans $\mathcal{X}$, mais leurs relations peuvent être synthétisées sur une variété linéaire de dimension t très réduite par

rapport à d. Si les données ne vivent pas précisément sur $\mathcal{M}$, elles s'en approchent et la distance à la variété est déterminée par une composante de bruit.
L'outil classique permettant d'effectuer de la réduction de dimension linéaire est la projection, et les méthodes sous-jacentes se concentrent sur la définition de projections des données initiales sur des sous-espaces de dimension réduite, au sens l'optimisation de critères statistiques à définir. De nombreux algorithmes ont été décrits dans la littérature, et nous proposons dans la suite deux d'entre eux, l'Analyse en Composantes Principales (ACP) et le Positionnement multidimensionnel (MDS, pour Multidimensional Scaling).

2.1.1 Analyse en composantes principales

L'Analyse en composantes principales (ACP, [4]) a été définie comme une méthode permettant de construire un ensemble de projections orthogonales d'un ensemble de données correlées, les projections étant ordonnées par variance décroissante.
Supposons que les variables soient les réalisations d'un vecteur aléatoire de dimension d $X = (X_1 \cdots X_d)^T$. On suppose que $\mathbb{E}(X) = \mu_X$ et que la matrice de covariance de X est notée Σ_X. L'ACP remplace les variables $X_1 \cdots X_d$ par un ensemble de nouvelles variables $\xi_1 \cdots \xi_t, t \leq d$ avec

$$\forall i \in \{1 \cdots d\} \quad \xi_i = b_i^T X$$

Les vecteurs $b_i, 1 \leq i \leq d$ sont obtenus en minimisant la perte d'information due à la projection des données. L'ACP mesure l'information comme la variation totale des variables initiales, ou inertie :

$$\sum_{i=1}^{d} Var(X_i) = Tr(\Sigma)$$

En utilisant le théorème de décomposition spectrale, Σ étant symétrique, on a $\Sigma = U\Lambda U^T$, avec U orthogonale, $\Lambda = diag(\lambda_1 \cdots \lambda_d)$. La variation totale est alors donnée par $\sum_{i=1}^{d} \lambda_i$.
Le vecteur b_i est alors choisi de la manière suivante :

- les t premières projections de X $\xi_1 \cdots \xi_t$, appelées t premières composantes principales, sont ordonnées par variance décroissante
- ξ_i est non corrélé à $\xi_j, j < i$

Notons $B = (b_1 \cdots b_t)^T$ la matrice de taille $t \times d$ et $\Xi = BX$, $\Xi = (\xi_1 \cdots \xi_t)^T$. L'ACP cherche un vecteur de dimension d μ et une matrice $A \in \mathcal{M}_{d,t}(\mathbb{R})$ telles que $X \approx \mu + A\Xi$ au sens des moindres carrés :

$$\min_{\mu,A,\Xi} \mathbb{E}\left((X-\mu-A\Xi)^T(X-\mu-A\Xi)\right)$$

Puisque $\Xi = BX$, l'ACP est un problème d'optimisation de la forme

$$\min_{\mu,A,B} \mathbb{E}\left((X-\mu-ABX)^T(X-\mu-ABX)\right)$$

La minimisation donne $A = B = (v_1 \cdots v_t) \in \mathcal{M}_{d,t}(\mathbb{R})$ et $\mu = (I-AB)\mu_x$, où $v_i, 1 \leq i \leq t$ est le vecteur propre associé à la i^e valeur propre λ_i de Σ_X. La reconstruction optimale de

X sur une variété linéaire de dimension t est donc donnée par

$$\begin{aligned}\hat{X} &= \mu + ABX \\ &= \mu_X + AB(X - \mu_X)\end{aligned}$$

avec

$$AB = \sum_{i=1}^{t} v_i v_i^T$$

La valeur de la fonction objectif à l'optimum est $\sum_{i=t+1}^{d} \lambda_i$ et les t premières composantes principales de X sont telles que

$$(\forall i \in \{1 \cdots t\}) \quad \xi_i = vj^T X$$

La covariance entre ξ_i et ξ_j est alors

$$cov(\xi_i, \xi_j) = cov(v_i^T X, v_j^T X) = v_i^T \Sigma_X v_j = \lambda_j v_i^T v_j = \lambda_j \delta_{ij}$$

où δ_{ij} est le symbole de Kronecker. On a ainsi $\forall i, var(\xi_i) = \lambda_i$.

En pratique, on ne connaît pas μ_x et Σ_x, puisqu'on a uniquement accès à un ensemble de données x_i représentant un échantillon de la population.
On doit donc estimer ces paramètres statistiques. On suppose donc avoir n observations $x = (x_1 \cdots n_n)$ i.i.d x_i de X. On approche μ_X par la moyenne arithmétique des observations $\bar{x}$, et Σ_X par la matrice de covariance sur l'échantillon : si $x_{c,i} = x_i - \bar{x}$ et $Z = (x_{c,1} \cdots x_{c,n}) \in \mathcal{M}_{d,n}(\mathbb{R})$ alors $\hat{\Sigma}_X \approx n^{-1} ZZ^T$. Comme dans le cas général, la reconstruction optimale de x est donnée par

$$x = \bar{x} + \left(\sum_{i=1}^{d} v_i v_i^T \right) (x - \bar{x})$$

les v_j étant les vecteurs propres de $\hat{\Sigma}_X$ associés aux λ_i. Le score principal de x associé à la i^e composante principale est $\xi_i = v_i^T (x - \bar{x})$ et la variance de cette composante est estimée par λ_i.
En pratique, t est inconnu. On utilise plusieurs critères pour estimer la dimension de la variété linéaire (et donc le nombre de composantes principales à retenir) :

— Critères théoriques : on détermine ici si les valeurs propres sont significativement différentes entre elles à partir d'un certain rang : si la réponse est négative on conserve les premières valeurs propres. On fait l'hypothèse que les n individus proviennent d'un tirage aléatoire dans une population gaussienne où $\lambda_{t+1} = \cdots = \lambda_d$. Si l'hypothèse est vérifiée, la moyenne arithmétique α des $d - t$ dernières valeurs propres et leur moyenne géométrique γ sont peu différentes. On admet que

$$c = \left(n - \frac{2d + 11}{6} \right) (d - t) ln \frac{\alpha}{\gamma}$$

suit une loi du χ^2 de degré de liberté $\frac{(d-t+2)(d-t-1)}{2}$ et on rejette l'hypothèse d'égalité des $d - t$ valeurs propres si c est trop grand.

— Pourcentage d'inertie : le critère couramment utilisé est le pourcentage d'inertie totale expliquée, qui s'exprime sur les t premiers axes par

$$\frac{\sum_{i=1}^{t} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$$

Un seuil par exemple de 90% d'inertie totale expliquée donne une valeur de t correspondante (figure 1). Attention cependant, le pourcentage d'inertie doit faire intervenir le nombre de variables initiales.

— Mesures locales : le pourcentage d'inertie expliquée est un critère global qui doit être complété par d'autres considérations. Supposons que le plan P_{12} des deux premières composantes principales explique une part importante d'inertie, et que, en projection sur ce plan, deux individus soient très proches. Cette proximité peut être illusoire si les deux individus se trouvent éloignés dans l'orthogonal de P_{12}. Pour prendre en compte ce phénomène, il faut envisager pour chaque individu la qualité de sa représentation, souvent exprimée par le cosinus de l'angle entre le plan principal et le vecteur représentant l'individu i. Si ce cosinus est grand, l'individu est voisin du plan, on peut alors examiner la position de sa projection sur le plan par rapport à d'autres points.

— Critères empiriques : lorsqu'on travaille sur données centrées réduites on retient les composantes principales correspondant à des valeurs propres supérieures à 1 (critère de Kaiser) : en effet les composantes principales étant des combinaisons linéaires des v_i de variance maximale $V(c) = \lambda$, seules les composantes de variance supérieure à celle des variables initiales présentent un intérêt.

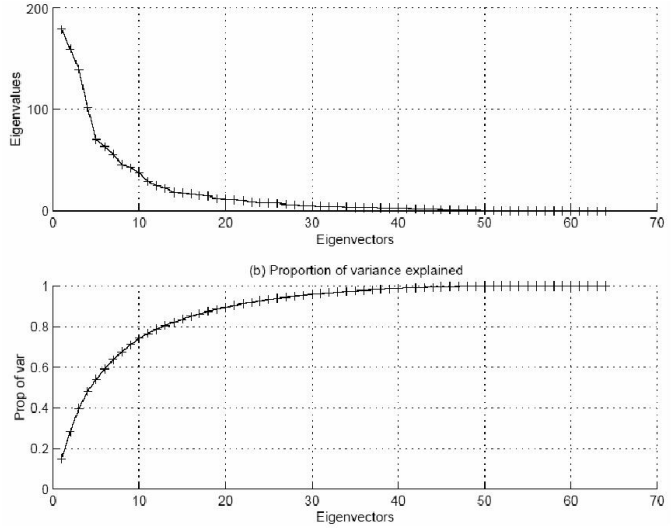


Fig. 1: Graphique du pourcentage de variance expliquée, permettant de choisir t

2.1.2 Positionnement multidimensionnel

Le Positionnement multidimensionnel (ou multidimensional scaling, MDS[3]) est un ensemble de méthodes dont l'objectif est d'identifier la variété linéaire $\mathcal{M}$ à partir de données de proximité (ou de dissimilarité) observées sur les individus $x_i, 1 \leq i \leq n$ (figure 2). Ces proximités ne doivent pas nécessairement être des distances, mais peuvent être des associations (valeur absolue d'un coefficient de corrélation par exemple), ou toute autre mesure quantitative ou qualitative permettant de discriminer deux individus entre eux. La seule contrainte est une contrainte de monotonie sur cette mesure de proximité.
On note δ_{ij} la dissimilarité entre x_i et x_j. Une matrice de proximité Δ est une matrice symétrique, dont la partie triangulaire supérieure (ou inférieure) et formée des $\delta_{ij}, i \leq j$ (resp. $i \geq j$), avec la convention $\delta_{ii} = 0, 1 \leq i \leq n$. On supposera de plus que l'inégalité triangulaire est satisfaite sur les éléments de Δ (i.e $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$) pour associer une métrique à Δ. Dans certaines applications, la contrainte de symétrie peut être relâchée.
Dans la suite, nous décrivons la méthode classique de positionnement multidimensionnel. On suppose que $x_i \in \mathbb{R}^d$ et que Δ est défini par la métrique euclidienne

$$(\forall i, j \in \{1 \cdots n\}) \quad \delta_{ij} = \|x_i - x_j\|$$

On a alors $\delta_{ij}^2 = \|x_i - x_j\|^2 = \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j$. En notant $\delta_{i0}^2 = \|x_i\|^2$, et en sommant sur i et j, on obtient

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \delta_{ij}^2 &= \frac{1}{n}\sum_{i=1}^{n} (\delta_{i0}^2 + \delta_{j0}^2) \\
\frac{1}{n}\sum_{j=1}^{n} \delta_{ij}^2 &= \delta_{i0}^2 + \frac{1}{n}\sum_{i=1}^{n} \delta_{i0}^2 \\
\frac{1}{n^2}\sum_{i,j=1}^{n} \delta_{ij}^2 &= \frac{2}{n}\sum_{i=1}^{n} (\delta_{i0}^2)
\end{aligned}$$

On pose alors :

$$\begin{aligned}
a_{ij} &= -\frac{1}{2}\delta_{ij}^2, \quad a_{i.} = \frac{1}{n}\sum_{j=1}^{n} a_{ij} \\
a_{.j} &= \frac{1}{n}\sum_{i=1}^{n} a_{ij}, \quad a_{..} = \frac{1}{n^2}\sum_{i,j=1}^{n} a_{ij} \\
b_{ij} &= x_i^T x_j = -\frac{1}{2}\left(\delta_{ij}^2 - \delta_{i0}^2 - \delta_{j0}^2\right)
\end{aligned}$$

de sorte que

$$b_{ij} = a_{ij} - a_{i.} - a_{.j} + a_{..}$$

ou, en notation matricielle, avec $A = (a_{ij})$ et $B = (b_{ij})$

$$B = HAH$$

où $H = I - \frac{1}{n}J$, J étant une matrice $n \times n$ de coefficients tous égaux à 1.
Le positionnement multidimensionnel vise à trouver des points $y_1 \cdots y_n \in \mathbb{R}^t, t < d$, appelés coordonnées principales, qui représentent les $x_i \in \mathbb{R}^d$, de sorte que les distances entre

paires de points soient au mieux respectées. Notons que si les dissimilarités sont définies comme des distances euclidiennes, alors la représentation MDS est équivalente à celle obtenue par l'ACP, les coordonnées principales étant identiques aux t premiers scores sur les composantes principales des x.
Cependant, l'ACP ne peut être appliquée puisque les données d'entrée ne sont pas les individus, mais les δ_{ij}. La démarche est donc de former la matrice A à partir de Δ, puis $B = HAH$. Il s'agit ensuite de trouver $B^* = (b^*_{ij})$, de rang au plus t, qui minimise

$$Tr\left[(B-B^*)^2\right] = \sum_{i,j=1}^{n} (b_{ij} - b^*_{ij})^2$$

Si $Sp(B) = \{\lambda_i\}_{1\leq i\leq n}$ et $Sp(B^*) = \{\lambda^*_i\}_{1\leq i\leq n}$ alors ce minimum est donné par $\sum_{i=1}^{n}(\lambda_i - \lambda^*_i)^2$, avec $\lambda^*_i = max(\lambda_i, 0), 1 \leq i \leq t$ et 0 sinon. En notant $\Lambda = diag(\lambda_1 \cdots \lambda_n)$ et $V = (v_1 \cdots v_n)$ la matrice des vecteurs propres de B alors $B = V\Lambda V^T$.
Si B est semi définie positive avec $Rg(B) = t < n$, les t plus grandes valeurs propres sont positives et les $n-t$ autres nulles. En notant $\Lambda_1 = diag(\lambda_1 \cdots \lambda_t)$ et $V_1 = (v_1 \cdots v_t)$ alors

$$B = V\Lambda V^T = V_1\Lambda_1 V_1^T = (V_1\Lambda_1^{1/2})(\Lambda_1^{1/2}V_1) = YY^T$$

où

$$Y = V_1\Lambda_1^{1/2} = \left(\sqrt{\lambda_1}v_1 \cdots \sqrt{\lambda_t}v_t\right) = (y_1 \cdots y_n)^T$$

Les coordonnées principales sont les colonnes de Y^T dont les distances paire à paire $d^2_{ij} = \|y_i - y_j\|^2$ sont égales aux distances δ_{ij}.
De même que pour l'ACP, la détermination de t reste empirique. Une méthode populaire consiste à tracer un graphe des valeurs propres de B, ordonnées (de la plus grande à la plus petite), et regarder à quel ordre ces valeurs se "stabilisent".

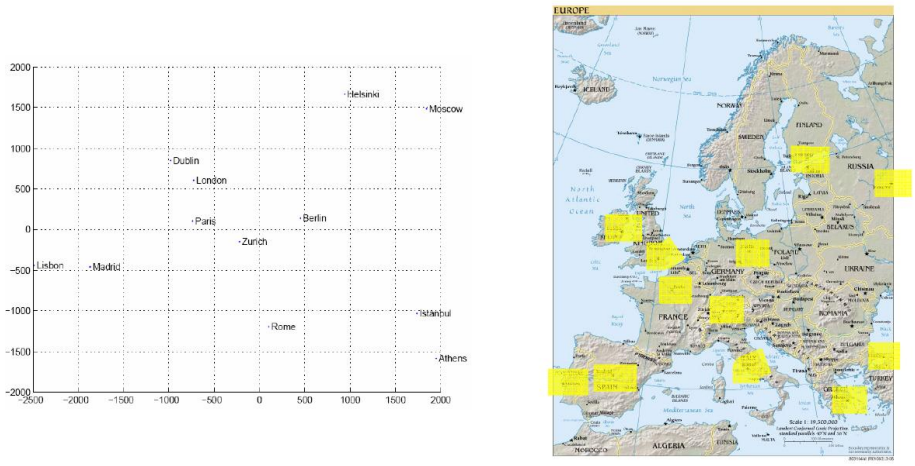


Fig. 2: Une application classique du positionnement multidimensionnel : reconstruction, à translation et rotation près, du positionnement de villes à partir de leur tableau de distances

2.2 Méthodes non linéaires

La variété $\mathcal{M}$ est maintenant supposée non linéaire. L'objectif des algorithmes suivants est de reconstruire une représentation de $\mathcal{M}$, où la structure de voisinage sur cette variété jouera un rôle important (figure 3). En effet, sur l'ensemble des méthodes qui vont être décrites (exceptée l'extension par astuce du noyau), toutes partagent une structure en trois étapes, dont la première (construction d'un graphe de voisinage) et la troisième (algorithme spectral) sont communes.

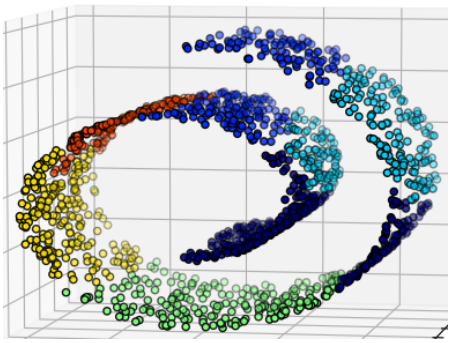


Fig. 3: Un exemple classique de variété non linéaire. Le swissroll est une variété 2D plongée dans un espace 3D. Les points vivent sur cette variété mais peuvent être représentés dans un plan.

2.2.1 Isomap

L'algorithme Isometric Feature Mapping (ISOMAP, [11]) suppose que $\mathcal{M}$ est un convexe de $\mathbb{R}^t$ et que le plongement ψ est une isométrie : pour chaque paire de points $y_i, y_j \in \mathcal{M}$, la distance géodésique entre y_i et y_j est égale à la distance euclidienne entre leurs coordonnées x_i, x_j sur $\mathcal{X}$: $d_{\mathcal{M}}(y_i, y_j) = \|x_i - x_j\|$
L'algorithme ISOMAP utilise ces deux hypothèses pour proposer une généralisation non linéaire de MDS. La méthode tente de préserver les propriétés géométriques globales de la variété $\mathcal{M}$ sous-jacente en approximant toutes les distances géodésiques entre paires de points y_i. Plus précisément, ISOMAP fonctionne en trois étapes :

Recherche des plus proches voisins ISOMAP calcule tout d'abord toutes les paires de distances $\|x_i - x_j\|$, pour $1 \leq i, j \leq n$. Une sélection des points voisins est ensuite effectuée, soit en ne retenant pour chaque point que K plus proches voisins, soit en retenant tous les points inclus dans une boule de rayon $\varepsilon > 0$. K (ou ε) est un paramètre de l'algorithme

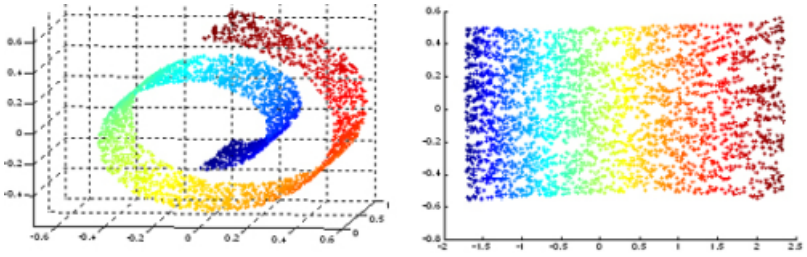
Calcul du graphe de voisinage Un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E})$ est ensuite construit, où $\mathcal{V} = (x_1 \cdots x_n)$ et $\mathcal{E} = (e_{ij})$ est un ensemble d'arcs reliant les points voisins au sens de K (ou ε), pondérés par des poids $w_{ij} = \|x_i - x_j\|$. Si deux points ne sont pas voisins, aucun arc ne relie ces points et le poids est nul. Les distances géodésiques entre points y_i et y_j de

$\mathcal{M}$ sont alors estimées par le calcul des plus courts chemins $d_{ij}^{\mathcal{G}}$ entre les paires de points correspondants x_i et x_j de $\mathcal{G}$.

Plongement spectral par MDS L'application de l'algorithme de positionnement multidimensionnel (cf. paragraphe 2.1.2) à la matrice symétrique de taille n composée des plus courts chemins entre les points x_i permet de reconstruire les points dans un espace de dimension t , de sorte que les distances géodésiques sur $\mathcal{M}$ soient préservées au mieux :

- Si $S = \left(\left(d_{ij}^{\mathcal{G}} \right)^2 \right)$, on calcule $A = -\frac{1}{2} HSH$, $H = I - \frac{1}{n} J$, J étant une matrice $n \times n$ de coefficients tous égaux à 1. A est semi définie positive de rang $t < n$
- Si $\hat{S} = \left(\|y_i - y_j\|^2 \right)$, la minimisation de $\|A - (-\frac{1}{2} H\hat{S}H)\|$ permet d'obtenir les vecteurs reconstruits $\hat{y}_i$. La solution optimale est obtenue à l'aide des vecteurs propres $v_1 \cdots v_t$ correspondant aux t plus grandes valeurs propres $\lambda_1 > \cdots \lambda_t$ de A.
- La i^e colonne de la matrice $Y = \left(\sqrt{\lambda_1} v_1 \cdots \sqrt{\lambda_t} v_t \right)^T$ donne les coordonnées du i^e point dans $\mathbb{R}^t$

ISOMAP fonctionne mal sur des variétés à trous, la contrainte de convexité étant violée. De plus, dans le cas de données bruitées (i.e. n'étant pas précisément sur la variété), l'influence de K (ou ε) joue beaucoup : si les valeurs des paramètres sont trop grandes (faux arcs dans $\mathcal{G}$) ou trop petites ($\mathcal{G}$ a trop peu d'arcs pour approximer de manière correcte les distances géodésiques), l'algorithme échoue.
Enfin, empiriquement, il a été montré que l'algorithme fonctionnait moins bien lorsque n est grand, et on lui préférera l'algorithme Landmark ISOMAP dans ce cas.
La figure 4(a) présente 3000 points échantillonnés sur le swissroll, et l'application de l'algorithme ISOMAP est illustrée sur la figure 4(b)



(a) 3000 points échantillonnés sur le Swissroll.

(b) Plongement ISOMAP, t=2

Fig. 4: Exemple d'application de l'algorithme ISOMAP sur le Swiss Roll.

2.2.2 Plongement localement linéaire

L'algorithme de plongement localement linéaire (Local Linear EmbeddingLLE [9]) est similaire dans l'esprit à ISOMAP, mais agit d'un point de vue local puisqu'il tente de préserver les informations locales sur la variété riemannienne plutôt que d'estimer les distances géodésiques.

Recherche des plus proches voisins pour $K_i << d$, on note $N_i^{K_i}$ le voisinage de x_i composé de ses K_i plus proches voisins, calculés en utilisant la distance euclidienne. K_i est indicé par i car il peut être différent pour chaque point de donnée.

Reconstruction par moindres carrés contraints En faisant l'approximation d'ordre 1 que, localement, une variété est linéaire, on reconstruit x_i comme combinaison linéaire de ses K_i plus proches voisins :

$$x_i = \sum_{j \in K_i} w_{ij} x_j = \sum_{j=1}^{n} w_{ij} x_j$$

On impose aux w_{ij} de sommer à 1 pour être invariant en translation, et si la somme est effectuée pour tous les points, alors $w_{ij} = 0$ si $x_j \notin N_i^{K_i}$.
On note $W = \left(w_{ij}\right)$ et on cherche les pondérations $\hat{w}_{ij}$ résolvant le problème d'optimisation contraint :

$$\begin{aligned} \min_{W} \quad & \sum_{i=1}^{n} \|x_i - \sum_{j=1}^{n} w_{ij} x_j\|^2 \\ s.c \quad & (\forall i \in \{1 \cdots n\}) \quad \sum_{j=1}^{n} w_{ij} = 1 \\ & (\forall i \in \{1 \cdots n\}) \quad w_{ij} = 0 \text{ si } x_j \notin N_i^{K_i} \end{aligned}$$

Pour $1 \leq i \leq n$, on note

$$\left\| \sum_{j=1}^{n} w_{ij}(x_i - x_j) \right\|^2 = w_i^T G_i w_i$$

où $w_i = (w_{i1} \cdots w_{in})^T \in \mathbb{R}^n$ et G_i est la matrice de Gram associée à x_i, de coefficients

$$G_{jk} = (x_i - x_j)^T (x_i - x_k), j, k \in N_i^{K_i}$$

La relaxation lagrangienne du problème d'optimisation amène à minimiser

$$w_i^T G_i w_i - \lambda(1_n^T w_i - 1)$$

où λ est le multiplicateur de Lagrange associé à la contrainte de somme unité, et 1_n est le vecteur de $\mathbb{R}^n$ composé uniquement de 1. L'annulation de la dérivée partielle par rapport à w_i donne $\hat{w}_i = \frac{\lambda}{2} G_i^{-1} 1_n$. En multipliant à gauche par 1_n^T, on obtient finalement

$$\hat{w}_i = \frac{G_i^{-1} 1_n}{1_n^T G_i^{-1} 1_n}$$

On forme alors $\hat{W}$ la matrice dont les colonnes sont les $\hat{w}_i$.

Plongement spectral On recherche enfin la matrice Y de taille $t \times n$ donnant les vecteurs y_i en résolvant le problème d'optimisation

$$\begin{aligned} \min_{Y} \quad & \sum_{i=1}^{n} \|y_i - \sum_{j=1}^{n} \hat{w}_{ij} y_j\|^2 \\ s.c \quad & Y 1_n = \sum_{i=1}^{n} y_i = 0 \in \mathbb{R}^t \\ & \frac{1}{n} Y Y^T = \frac{1}{n} \sum_{i=1}^{n} y_i y_i^T = I \in \mathcal{M}_t(\mathbb{R}) \end{aligned}$$

Les contraintes déterminent le positionnement (translation, rotation, échelle) des coordonnées y_i.

La fonction objectif peut être réécrite $Tr(YMY^T)$, avec $M = (I - \hat{W})^T (I - \hat{W})$. Cette fonction a un minimum global unique donné par les vecteurs propres correspondant aux t+1 plus petites valeurs propres de M, associées aux vecteurs propres $v_n \cdots v_{n-t}$. La plus petite d'entre elles est 0, correspondant à $v_n = \frac{1}{\sqrt{n}} 1_n$. Puisque la somme des coefficients de chacun des autres vecteurs propres, qui sont orthogonaux à v_n, est zéro, ignorer v_n et sa valeur propre associée contraindra les coordonnées y_i d'être de somme nulle. La solution optimale est donc

$$Y = (v_{n-1} \cdots v_{n-t}) \in \mathcal{M}_{t,n}(\mathbb{R})$$

Contrairement à ISOMAP, LLE gère assez bien les variétés non convexes, puisque l'algorithme préserve les propriétés géométriques locales, plutôt que globales. En, revanche, les variétés à trou sont également difficiles à traiter par cet algorithme.

2.2.3 Laplacian Eigenmaps

Les cartes propres du laplacien ([2]) est un algorithme en trois étapes, comme les précédents, dont les première et troisième sont très proches de celles de LLE :

Recherche des plus proches voisins Pour $K << d$ (ou $\varepsilon > 0$), on définit de manière symétrique les voisinages des points : si N_i^K est le voisinage de x_i composé de ses K plus proches voisins, $x_j \in N_i^K \Leftrightarrow x_i \in N_j^K$ (ou si N_i^ε est le voisinage de x_i alors $x_j \in N_i^\varepsilon \Leftrightarrow \|x_i - x_j\| < \varepsilon$). On notera dans la suite N_i le voisinage de x_i

Calcul de la matrice d'adjacence On définit $W \in \mathcal{M}_n(\mathbb{R})$ la matrice d'adjacence, de paramètre σ, par :

$$w_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} & \text{si } x_j \in N_i \\ 0 & \text{sinon} \end{cases}$$

W définit un graphe pondéré $\mathcal{G}$.

Plongement spectral Pour chaque sous graphe de $\mathcal{G}$ (si le graphe n'est pas complètement connecté, ou pour $\mathcal{G}$ s'il l'est), on calcule la matrice diagonale des degrés $D \in \mathcal{M}_n(\mathbb{R})$, où pour $1 \leq i \leq n$ $d_i = \sum_{j=1}^{n} w_{ij} = (W.1_n)_i$, puis le laplacien du graphe L défini par $L = D - W$.
L est semi défini positif. De plus, lorsque les données sont uniformément échantillonnées depuis une variété $\mathcal{M}$ de dimension t, alors L est une approximation de l'opérateur de Laplace Beltrami défini sur $\mathcal{M}$, et converge vers cet opérateur lorsque σ tend vers 0 et n vers l'infini.
La matrice $Y \in \mathcal{M}_{tn}(\mathbb{R})$ qui permet de plonger $\mathcal{G}$ dans un espace de dimension t (dont les colonnes sont donc les coordonnées du i^e point de données) est obtenue en résolvant le problème d'optimisation

$$\begin{aligned} &\min_{Y} \sum_{i,j=1}^{n} w_{ij} \|y_i - y_j\|^2 = Tr(YLY^T) \\ sc \quad &YDY^T = I \in \mathcal{M}_t(\mathbb{R}) \end{aligned}$$

la contrainte impose de rester dans un espace de dimension t.
La solution est donnée par l'équation propre généralisée $Lv = \lambda Dv$ ou, de manière équivalente, par la recherche des éléments propres de $\hat{W} = D^{-1/2} W D^{-1/2}$. La plus petite valeur propre de $\hat{W}$ étant 0 (de vecteur propre 1_n), on obtient en l'ignorant la solution

$$Y = (v_{n-1} \cdots v_{n-t}) \in \mathcal{M}_{t,n}(\mathbb{R})$$

correspondant aux t plus petites valeurs propres (hors 0) de $\hat{W}$.

2.2.4 Diffusion maps

L'algorithme des cartes de diffusion [8] associe une chaîne de Markov à un graphe construit sur les points de données, et analyse plus spécifiquement les éléments propres de la matrice de transition de cette chaîne. Là encore, cet algorithme fonctionne en trois étapes.

Recherche des plus proches voisins Pour $K << d$ (ou $\varepsilon > 0$), on définit de manière symétrique les voisinages des points : si N_i^K est le voisinage de x_i composé de ses K plus proches voisins, $x_j \in N_i^K \Leftrightarrow x_i \in N_j^K$ (ou si N_i^ε est le voisinage de x_i alors $x_j \in N_i^\varepsilon \Leftrightarrow \|x_i - x_j\| < \varepsilon$). On notera dans la suite N_i le voisinage de x_i

Calcul de la matrice d'adjacence Un graphe $\mathcal{G}(\mathcal{V}, \mathcal{E})$ est ensuite construit, où $\mathcal{V} = (x_1 \cdots x_n)$ et $\mathcal{E} = (e_{ij})$ est un ensemble d'arcs reliant les points voisins au sens de K (ou ε), pondérés par des poids définis par un noyau. Si le noyau gaussien de largeur de bande σ :

$$w_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} & \text{si } x_j \in N_i \\ 0 & \text{sinon} \end{cases}$$

est couramment employé, d'autres noyaux peuvent également être utilisés.

Plongement spectral Comme dans le paragraphe 2.2.3, on forme le laplacien de $\mathcal{G}$, $L = D - W$, où $W = (w_{ij})$ et $D \in \mathcal{M}_n(\mathbb{R})$, avec $d_i = \sum_{j=1}^{n} w_{ij}$. On s'intéresse alors aux solutions du problème propre généralisé $Lv = \lambda Dv$, ou de manière équivalente au laplacien du graphe normalisé

$$P = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

Par construction, P est une matrice stochastique, et définit ainsi une marche aléatoire sur $\mathcal{G}$. Soit alors X_t une telle marche sur $\mathcal{G}$, démarrant en noeud arbitraire du graphe au temps $t = 0$. Le noeud visité en $t = 1$ est tiré selon une distribution de probabilité, obtenue en normalisant les lignes de W. En supposant la chaîne stationnaire, on peut calculer plus généralement la probabilité d'être en x_j à l'instant $t+1$, sachant que la marche visitait x_i à l'instant t :

$$(\forall i, j \in \{1, n\}) \quad p(x_j|x_i) = P(X_{t+1} = x_j | X_t = x_i) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}}$$

La matrice $P = (p(x_j|x_i))$ définit la chaîne de Markov X_t sur $\mathcal{G}$. Etant une matrice stochastique, elle possède des valeurs propres $\lambda_0 = 1 \geq \lambda_1 \cdots \geq \lambda_{n-1} = 0$ et des vecteurs propres à gauche $\phi_j^T P = \lambda_j \phi_j$ et à droite $P\psi_j = \lambda_j \psi_j$, avec $\phi_i^T \psi_j = \delta_{ij}$. Le théorème spectral permet donc d'écrire $P = \Psi \Lambda \Phi^T$, où Ψ (resp. Φ) est la matrice des vecteurs propres à droite (resp. à gauche), et Λ la matrice diagonale des valeurs propres.
La matrice P élevée à la puissance q donne la probabilité de passer d'un point x_i à un point x_j en q étapes de la marche aléatoire :

$$P^q = (p_q(x_j, x_i)) = P(X_{t+q} = x_j | X_t = x_i)$$

On construit alors une distance sur $\mathcal{G}$ de sorte que deux points x_i, x_j sont proches si les probabilités conditionnelles $p_q(.|x_i)$ et $p_q(.|x_j)$sont proches. On définit ainsi la distance de diffusion :

$$d_q^2(x_i, x_j) = \sum_y w(y) \left(p_q(y|x_i) - p_q(y|x_j) \right)^2$$

où $w(y) = 1/\phi_0(y)$ est l'inverse de la probabilité d'atteindre y après un temps infini, indépendamment de l'état initial de la chaîne de Markov : c'est donc l'inverse de la distribution stationnaire de la chaîne calculée en y.
d_q^2 donne des informations sur la densité de chemins qui existent pour relier deux points donnés dans $\mathcal{G}$: si $d_q^2(x_i, x_j)$ est faible, de nombreux chemins connectent x_i et x_j dans $\mathcal{G}$. Puisque $P^q = \Psi \Lambda^q \Phi$, et que

$$p_q(x_j, x_i) = \phi_0(x_j) + \sum_{k=1}^{n-1} \lambda_k^q \psi_k(x_i) \phi_k(x_j)$$

alors

$$d_q^2(x_i, x_j) = \sum_{k=1}^{n-1} \lambda_k^{2q} \left(\psi_k(x_i) - \psi_k(x_j) \right)^2$$

Les valeurs propres de P décroissant de manière assez rapide, on ne retient que les t premiers termes de la somme, et on obtient une approximation de rang t de P^q :

$$d_q^2(x_i,x_j) = \sum_{k=1}^{t} \lambda_k^{2q} \left(\psi_k(x_i) - \psi_k(x_j)\right)^2 = \|\Psi_q(x_i) - \Psi_q(x_j)\|^2$$

avec la carte de diffusion Ψ_q définie par $\Psi_q(x) = \left(\lambda_1^q \psi_1(x) \cdots \lambda_t^q \psi_t(x)\right)^T$. Cette carte plonge donc les vecteurs x_i dans $\mathbb{R}^t$, de sorte que les distances de diffusion entre deux points soient approchées par la distance euclidienne dans $\mathbb{R}^t$ entre les vecteurs propres à droite pondérés par les λ_k^q.

2.2.5 Astuce du noyau

Il est possible de rechercher une variété non linéaire, en utilisant une méthode linéaire impliquant un produit scalaire, et en remplaçant le produit scalaire canonique par un noyau (Kernel trick, ou astuce du noyau). Pour illustration, nous proposons de montrer comment l'ACP peut être rendue non linéaire.
L'ACP à noyau [10] transforme tout d'abord les données initiales via une transformation non linéaire $\Phi : \mathbb{R}^d \to \mathcal{H}$, $\mathcal{H}$ étant un espace de Hilbert de dimension $n_{\mathcal{H}} >> d$, muni d'un produit scalaire $< . >$. Les points $\Phi(x_i)$ sont ensuite analysés par ACP classique dans $\mathcal{H}$, l'idée étant que des structures de faible dimension peuvent être découvertes plus facilement dans un espace de très grande dimension.
On suppose dans la suite que les vecteurs transformés $\Phi(x_i)$ sont de moyenne nulle. l'ACP dans $\mathcal{H}$ s'effectue en calculant les éléments propres (λ, v) de la matrice de covariance $C = n^{-1} \sum \Phi(x_i)\Phi(x_i)^T$. On a alors :

$$(\forall i \in \{1,n\}) \quad < \Phi(x_i), Cv >= \lambda \left\langle \Phi(x_i), v \right\rangle$$

et $Cv = n^{-1} \sum_{i=1}^{n} \Phi(x_i) \left\langle \Phi(x_i), v \right\rangle$
Ainsi, toutes les solutions telles que $\lambda \neq 0$ sont dans $Im\left(\Phi(x_1 \cdots \Phi(x_n)\right)$ et il existe des réels α_i tels que $v = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$. Et :

$$(\forall i \in \{1,n\}) \quad \frac{1}{n} \sum_{j=1}^{n} \alpha_j \left\langle \Phi(x_i), \sum_{k=1}^{n} \Phi(x_k) \right\rangle \left\langle \Phi(x_k), \Phi(x_j) \right\rangle = \lambda \sum_{k=1}^{n} \alpha_k \left\langle \Phi(x_i), \Phi(x_k) \right\rangle$$

La résolution de cette équation propre dépend du produit scalaire dans $\mathcal{H}$. L'astuce du noyau consiste à utiliser un noyau de Mercer $K_{ij} = K(x_i,x_j) = \left\langle \Phi(x_i), \Phi(x_j) \right\rangle$ (voir le cours sur les méthodes à noyau). Un noyau très utilisé est le noyau gaussien :

$$K(x_i,x_j) = exp(-\frac{\| x_i - x_j \|^2}{2\sigma^2})$$

où σ est un paramètre (bande passante).
L'équation propre se réécrit alors simplement $K^2\alpha = n\lambda K\alpha$, où K est la matrice $n \times n$ des

K_{ij} et α le vecteur des α_i. Pour $\mu = n\lambda$, on a alors $K\alpha = \mu\alpha$.
Les $\mu_i \geq 0$ sont donc les valeurs propres (supposées ordonnées de manière décroissante) de K, associés aux vecteurs propres α_i. En utilisant l'écriture des vecteurs propres v_i de C dans la base des $\Phi(x_i)$, on a alors

$$\begin{aligned}
\langle v_i, v_i \rangle &= \left\langle \sum_{j=1}^{n} \alpha_j \Phi(x_j), \sum_{k=1}^{n} \alpha_k \Phi(x_k) \right\rangle \\
&= \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_{ij} \alpha_{ik} \left\langle \Phi(x_j) \Phi(x_k) \right\rangle \\
&= \sum_{j=1}^{n} \sum_{k=1}^{n} \alpha_{ij} \alpha_{ik} K_{jk} \\
&= \langle \alpha_i, K\alpha_i \rangle \\
&= \mu_i \langle \alpha_i, \alpha_i \rangle
\end{aligned}$$

et en imposant $< v_i, v_i >= 1$ on a la normalisation des vecteurs α_i.
Les coordonnées principales non linéaires d'un vecteur x sont alors données par la projection de $\Phi(x) \in \mathcal{H}$ sur les vecteurs propres v_k de la matrice de covariance C :

$$(\forall k \in \{1, n\}) \quad \left\langle \Phi(x), v_k \right\rangle = \frac{1}{\sqrt{\lambda_k}} \sum_{i=1}^{n} \alpha_{ki} K(x_i, x)$$

où $1/\sqrt{\lambda_k}$ assure une normalisation des vecteurs propres v_k.
On peut montrer qu'en choisissant K de manière appropriée, l'ACP à noyau est fortement reliée à ISOMAP, LLE ou aux cartes propres du laplacien.
Notons pour finir que l'hypothèse de moyenne nulle pour les $\Phi(x_i)$ n'implique pas de connaître explicitement Φ, ce qui est en pratique souvent infaisable. En posant, comme dans le cas du positionnement multidimensionnel $H = I - \frac{1}{n}J = I - \frac{1}{n}1_n 1_n^T$, le noyau $\tilde{K} = HKH$ centre les données dans $\mathcal{H}$.

3 Partie pratique

Scikit-Learn propose toutes les implémentations de presque tous les algorithmes précédemment décrits. Le module sklearn.manifold réalise en particulier le méthodes de manifold learning (et implémente d'autres algorithmes dont le très performant t-distributed Stochastic Neighbor Embedding (t-SNE)), et les algorithmes linéaires (ACP, ICA,...) sont proposés dans le module decomposition.

3.1 Sélection de caractéristiques

Implémenter la méthode SAC décrite dans le paragraphe 1.3.4. Pour ce faire, vous utiliserez le module svm pour implémenter les classifieurs M_i sur chaque descripteur. Appliquer alors l'algorithme SAC aux données fournies dans les fichiers `carcinomeX.txt` et `carcinomeY.txt`. Ces fichiers donnent la description (pour `X`) et la classe (pour `Y`) de

174 cellules analysées, décrites chacune par 9182 caractéristiques numériques, et classées dans l'une des 11 catégories de carcinomes. L'algorithme SAC doit permettre de sélectionner parmi les 9182 caractéristiques celles jugées les plus pertinentes. Vous utiliserez ensuite un algorithme de classification de votre choix, appliqué aux caractéristiques extraites, et comparerez le résultat aux classes théoriques contenues dans `carcinomeY.txt`.

3.2 Challenge

Le challenge consiste à explorer des méthodes de manifold learning pour l'analyse d'images de visages. Les objectifs sont multiples :

- Positionner l'ensemble des visages sur une variété non linéaire de faible dimension, en regard de la dimension initiale des données (images 92×112 en niveaux de gris)
- Classer les images sur cette variété (les individus proches se projettent ils dans des zones proches sur la variété ?)
- Donner un sens sémantique à la variété (une direction de cette variété peut correspondre à l'évolution d'un paramètre sémantique haut niveau, modélisant par exemple les sentiments exprimés sur les visages)

Pour ce faire nous utilisons la base de données ORL, que vous pouvez récupérer dans le fichier `orl.zip`. Cette base est constituée d' un ensemble d'images 92×112 de 40 individus, prises dans des conditions d'acquisition (luminosité) et d'expérimentation (yeux ouverts/fermés, sourire, lunettes ou non) différentes. En revanche, le fond (noir) est homogène et les visages sont pris de face. (figure 5).
Il vous est demandé de tester et de commenter les résultats obtenus par les méthodes de manifold learning sur ce jeu de données (quelle dimension pour $\mathcal{M}$? quelle méthode est la plus efficace ? Pourquoi ? que donnent des algorithmes de classification sur la variété ?....)
Vous pourrez utiliser scikit-image pour la manipulation des images.

RÉFÉRENCES

[1] Hussein Almuallim and Thomas G. Dietterich. Learning with many irrelevant features. In *Proc. Ninth National Conference on Artificial Intelligence*, pages 547–552. AAAI Press, 1991.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6) :1373–1396, June 2003.

[3] Trevor F. Cox and M.A.A. Cox. *Multidimensional Scaling, Second Edition*. Chapman and Hall/CRC, 2 edition, 2000.

[4] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.*, 24, 1933.

[5] Rostom Kachouri, Khalifa Djemal, and Hichem Maaref. Adaptive feature selection for heterogeneous image databases. *2010 2nd International Conference on Image Processing Theory, Tools and Applications*, pages 26–31, 2010.

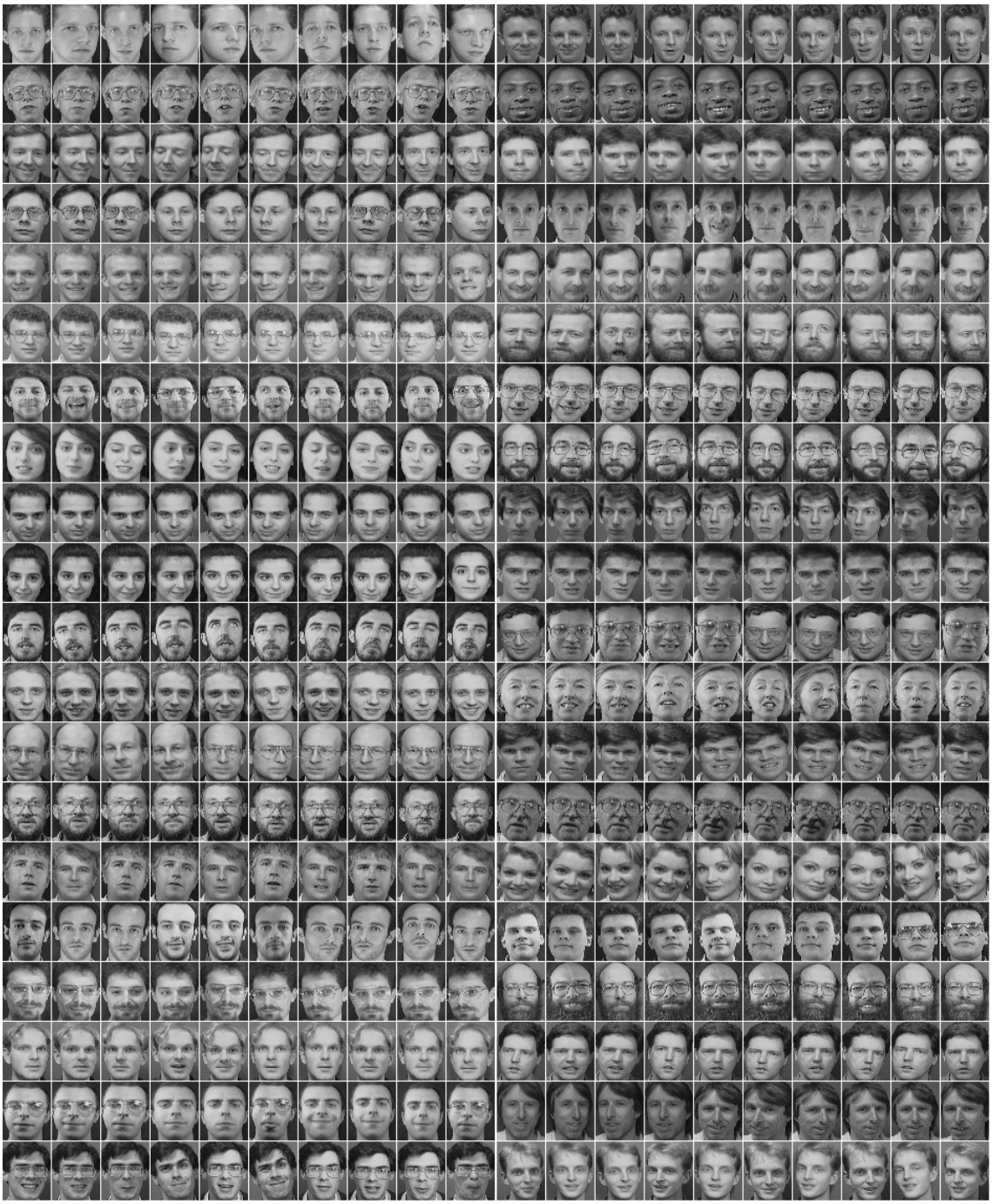


Fig. 5: Base d'images ORL

[6] Kenji Kira and Larry A. Rendell. The feature selection problem : Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI'92, pages 129–134. AAAI Press, 1992.

[7] T Marill and D Green. On the effectiveness of receptors in recognition systems. *IEEE Transactions on Information Theory*, 9(1) :11–17, 1963.

[8] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Dif-

fusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *in Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, 2005.

[9] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290 :2323–2326, 2000.

[10] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. *Kernel principal component analysis*, pages 583–588. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[11] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500) :2319, 2000.

[12] A. W. Whitney. A direct method of nonparametric measurement selection. *IEEE Trans. Comput.*, 20(9) :1100–1103, September 1971.

[13] Jihoon Yang and Vasant G. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2) :44–49, March 1998.