Datamining 2: Methodes de projections non linéaires

M2 STD

September 16, 2015

projection et réduction de dimension

But : réaliser un "équivalent d'ACP" quand:

- On a des données présentées sous une forme "différente"
- Les données sont non-linéaires

le multidimensional scaling classique

- cmdscale sous R ou matlab

Idée : comment faire une ACP a partir d'un tableau de distances

Parfois on n'a pas les données représentées comme un tableau individu*variables mais un tableau de dissimalarité entre individu $D \in \mathcal{M}_n(\mathbb{R})$ ou $D_{i,j}$ mesure la "dissimilarité" entre l'individu i et l'individu j

faire une *ACP* a partir d'un tableau de distance

- Dans un premier temps on va supposer que $D_{i,j}$ est la distance euclidienne entre $X_i = (x_{i,1}, \ldots, x_{i,p})$ et $X_i = (x_{j,1}, \ldots, x_{j,p})$.
- On appelle X la matrice des $X_{i,j}$ (tableau de données en ACP) qu'on suppose centrée (comme pour l'ACP)
- On rappelle que l'ACP consiste a diagonaliser $X'X$ et que les projections des individus sur les axes sont les $X.u_k$ ou u_k est le $k$eme vercteur propre
- On rappelle que Xu_k est le $k$eme vecteur propre de XX'

théoreme

Si la distance dérive d'un produit scalaire dans $\mathbb{R}^d$ alors on peut le retrouver par la formule suivante :

$$(XX')_{i,j} = \frac{-n^2 D_{i,j}^2 + n\sum_i D_{i,j}^2 + n\sum_j D_{i,j}^2 - \sum_{i,j} D_{i,j}^2}{n^2}$$

MDS: le principe

- A partir de la matrice D des dissimilarités (qui peuvent ne pas etre symetrique) on calcule la matrice Q telle que

$$(XX')_{i,j} = \frac{-n^2 D_{i,j}^2 + n\sum_i D_{i,j}^2 + n\sum_j D_{i,j}^2 - \sum_{i,j} D_{i,j}^2}{n^2}$$

- Les projections des individus sur les k premiers axes correspondent aux k valeurs propres associées aux k plus grandes valeurs propres de Q

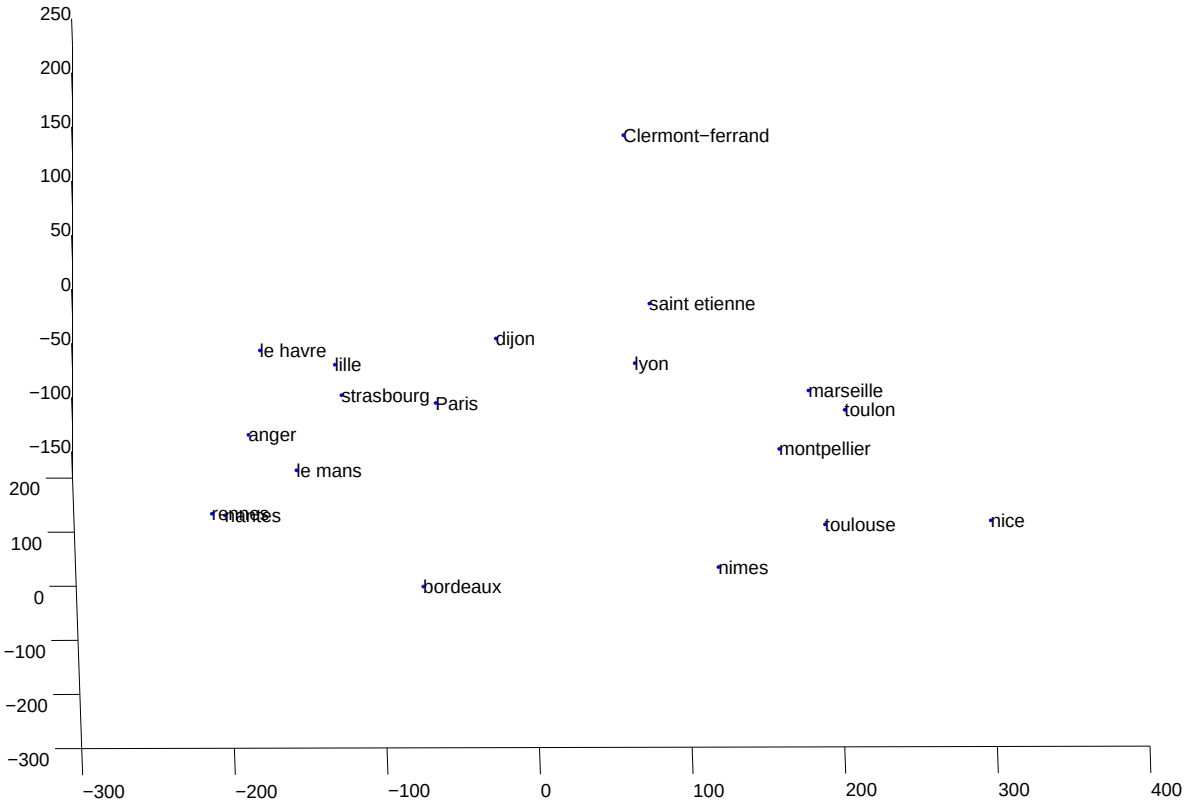
Propriété : on retrouve exactement les résultats d'une ACP lorsque les distances sont euclidiennes (et donc on généralise l'ACP)

Exemple 1: carte de France distance sncf

La dissimilarité entre deux ville est le temps de transport en train...
Sur les principales villes de Frances on obtient le tableau suivant

	paris	marseille	lyon	toulouse	nice	nantes
paris	0	196	118	368	341	155
marseille	196	0	100	297	163	417
lyon	118	100	0	245	267	272
toulouse	368	297	245	0	437	420
nice	341	163	267	437	0	555
nantes	155	417	272	420	555	0
strasbourg	490	675	385	540	488	330

Exemple 1: carte de France distance sncf



Probleme

Discussion : voyez vous le(s) problèmes qu'il peut y avoir ?

Probleme : Indice....

Si la distance dérive d'un produit scalaire dans $\mathbb{R}^d$ alors on peut le retrouver par la formule suivante :

$$(XX')_{i,j} = \frac{-n^2 D_{i,j}^2 + n\sum_i D_{i,j}^2 + n\sum_j D_{i,j}^2 - \sum_{i,j} D_{i,j}^2}{n^2}$$

Probleme : Indice....

Si la distance dérive d'un produit scalaire dans $\mathbb{R}^d$ alors on peut le retrouver par la formule suivante :

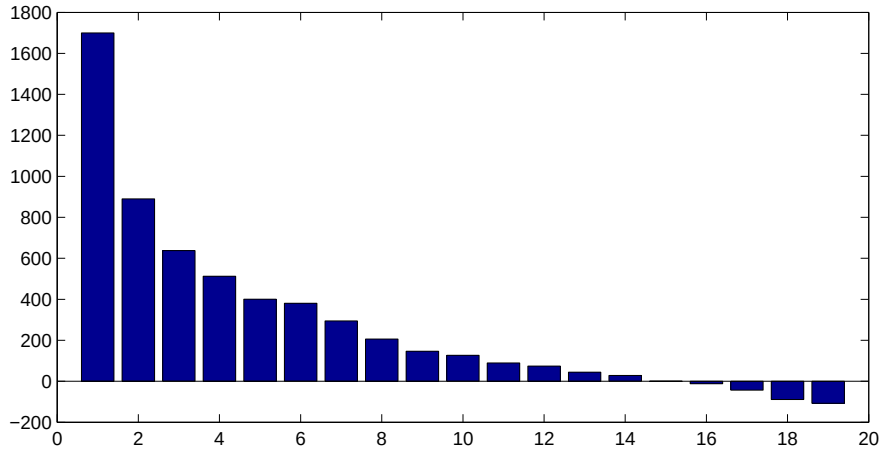
$$(XX')_{i,j} = \frac{-n^2 D_{i,j}^2 + n\sum_i D_{i,j}^2 + n\sum_j D_{i,j}^2 - \sum_{i,j} D_{i,j}^2}{n^2}$$

Probleme : Indice....

Si la distance dérive d'un produit scalaire dans $\mathbb{R}^d$ alors on peut le retrouver par la formule suivante :

$$(XX')_{i,j} = \frac{-n^2 D_{i,j}^2 + n\sum_i D_{i,j}^2 + n\sum_j D_{i,j}^2 - \sum_{i,j} D_{i,j}^2}{n^2}$$

les valeurs propres dans le cas de la distance sncf...



le multidimensional scaling non metrique

mscale sous R ou matlab

le multidimensional scaling non metrique

On choisit une dimension cible p.

- On munit $\mathcal{M}_n(\mathbb{R})$ d'une distance d (le plus souvent la distance euclidienne : $d(M,N) = \sum_{i,j}(M_{i,j} - N_{i,j})^2$).
- On cherche $Y \in \mathbb{M}_{n,p}(\mathbb{R})$ qui minimise $d(D, M(Y))$ ou $M(Y)_{i,j} = ||Y_{i,.} - Y_{j,.}||^2$.

Le choix de la distance

Si les entrées sont un tableau de distance il y a juste a "lancer l'algorithme" sinon il faut trouver une distance adequat (ce qui est la partie la plus difficile)

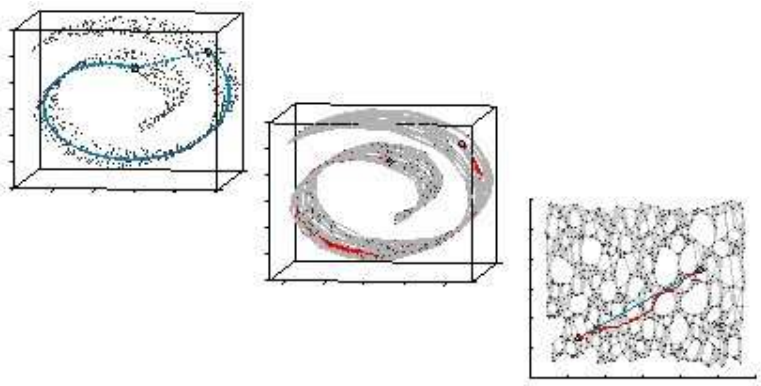
quelques dissimilarité dans le cas des données quantitatives

Ici on se donne deux vecteurs X et Y dans $\mathbb{R}^p$

- Distance euclidienne $d(X,Y)^2 = \sum (X_i - Y_i)^2$, distance L^p: $d(X,Y)^p = \sum (X_i - Y_i)^p$
- Si les données sont des pourcentages χ^2 $d(X,Y)^2 = \sum \frac{(X_i - Y_j)^2}{X_i + Y_j}$ ou divergence de Kulback Lieber $d(X,Y) = \sum X_i \log(X_i/Y_i)$
- Le cas des données fonctionnelles (lissage distances entre les épigraphes)

données "non linéaires" Isomap

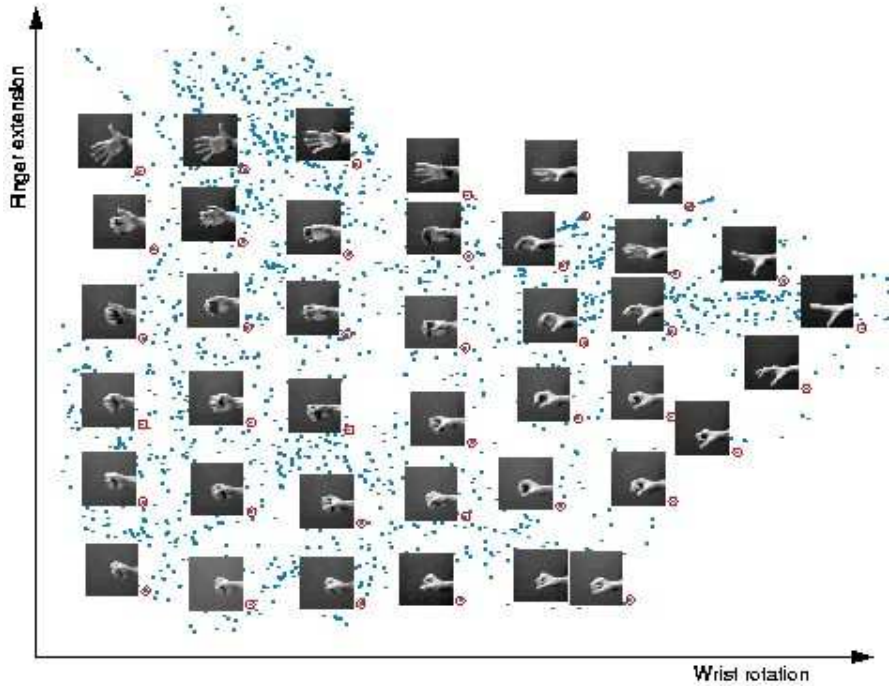
En plus des différentes dissimilarités proposées on peut appliquer le MDS directement... ou... travailler sur les distances géodesiques (idée d'isomap)



Comment calculer la distance géodesique en pratique

- Choix d'un parametre de voisinage (rayon ou nombre de voisin)
- Calcul d'un graph local (a partir des parametres)
- Algorithme de Didjkstra

Quelques résultats (tirés de l'article fondateur)



coté pratique

package sombrero sous R sous sas de bons programme a télécharger sur :
http://samm.univ-paris1.fr/Programmes-SAS-de-cartes-auto

Histoire (wikipédia dit)

"Ces structures intelligentes de représentation de données sont inspirées, comme beaucoup d'autres créations de l'intelligence artificielle, par la biologie ; Il s'agit de reproduire le principe neuronal du cerveau des vertébrés : des stimuli de même nature excitent une région du cerveau bien particulière. Les neurones sont organisés dans le cortex de façon à interpréter tous les types de stimuli imaginables. De la même manière, la carte auto adaptative se déploie de façon à représenter un ensemble des données, et chaque neurone se spécialise pour représenter un groupe bien particulier des données selon les points communs qui les rassemblent. Elle permet une visualisation en dimension multiple de données croisées."

choix d'une carte

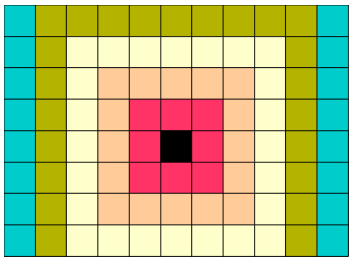
On se définit une structure de projection qui représente la forme des données (connaissance a priori forte) par exemple:

- une ficelle
- une carte
- un cylindre
- une sphère
- un tore
- ....

Dans la suite on ne considérera que la carte ou la ficelle (mais tout fonctionne pareil).

On "découpe" la structure en cellule qu'on munit d'une distance

La structure est découpées en p cases $1, \ldots, p$ et on se définit une distance $d_{carte}(i, j)$.
Le cas le plus simple est celui de la carte et de la distance "de Manhattan".



Choix de deux fonctions qui décroissent en fonction du temps

- Une fonction V_t qui représente le rayon d'un voisinage au temps t
- Une fonction ε_t qui représente la "perturbation" au temps t

Les "parametres" en résumé

- Une structure géométrique, découpée en cellules entre lesquelles on a défini une distance
- deux fonction décroissant au cours du temps

l'algorithme : principe

La structure est découpées en p cases $1, \ldots, p$ et on s'est fixé $d_{carte}(i, j)$. On va chercher p "vecteurs codes" $C_1, \ldots, C_p$ dans $\mathbb{R}^d$ (dimension des données) qui "représente bien les données et qui sont organisés comme la structure"....

dessin au tableau

l'algorithme : initialisation

On affecte a chaque case i un vecteur code C_i par tirage aléatoire (sans remise) d'un individu de la base $C_i = X_{i_0}$ ou i_0 est tirée aléatoirement et pour tout $i \neq j$ $C_i \neq C_j$.

l'algorithme : iterations

de $t = 1$ a T_{max}

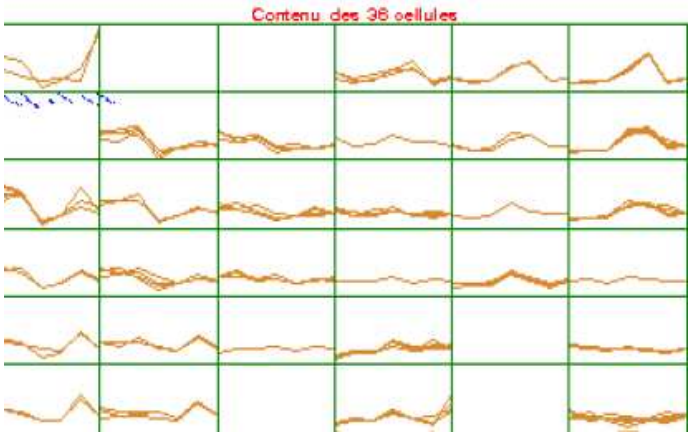
- On tire un point de la base au hasard X_{i_0}
- On recherche $i = \text{argmin} \, ||C_i - X_{i_0}||$
- Pour tout j tel que $d_{carte}(i, j) \leq V_t$ on applique : $C_j := (1 - \varepsilon_t)C_j + \varepsilon_t X_{i_0}$

description des sorties avec un exemple

Un exemple : 96 pays en 1996

ANCRX	Croissance annuelle de la population en %
TXMORT	Taux de mortalité infantile (en pour mille)
TXANAL	Taux d'illetrisme en %
SCOL2	Indice de fréquentation scolaire au second degré
PNBH	PNB par habitant exprimé en dollars
CHOMAG	Taux de chomage en %
INFLAT	Taux d'inflation en %

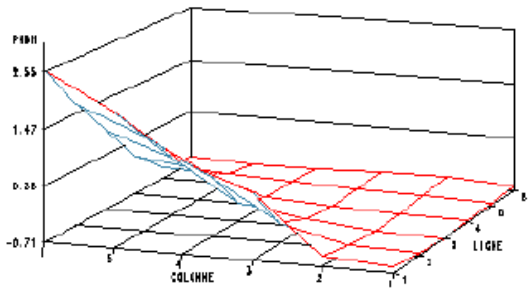
description des sorties avec un exemple: convergence

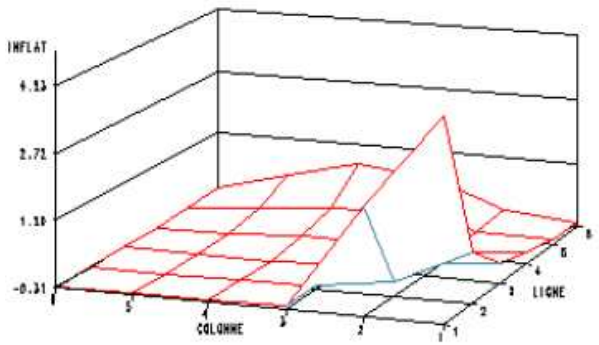


description des sorties avec un exemple: individus dans la carte

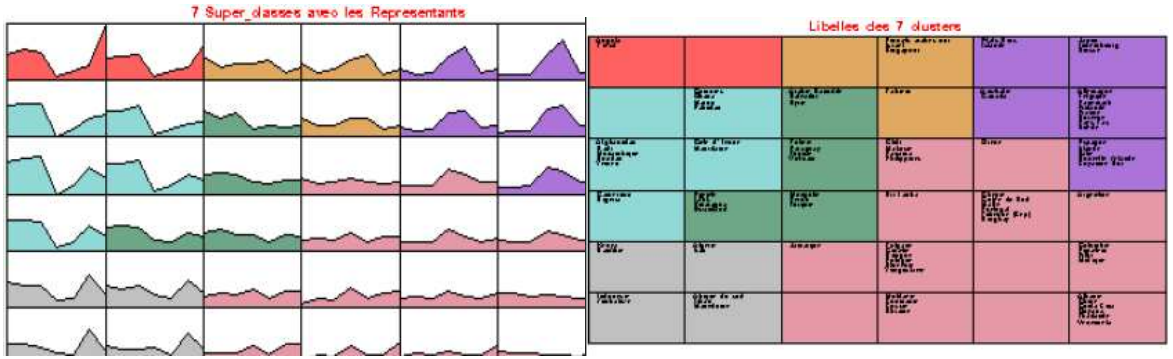
Angola Bresil			Emirats arabes uni Israel Singapour	Etats Unis Islande	Japon Luxembourg Suisse
	Comores Ghana Maroc Pakistan	Arabie Saoudite Salvador Syrie	Bahrein	Australie Canada	Allemagne Belgique Danemark Finlande France Norvege Pays Bas Suede
Afghanistan Haiti Mozambique Soudan Yemen	Cote d'Ivoire Mauritanie	Bolivie Paraguay Tunisie Vietnam	Chili Malaisie Panama Philippines	Grece	Espagne Irlande Italie Nouvelle Zelande Royaume Uni
Cameroun Nigeria	Egypte Laos Nicaragua Swaziland	Mongolie Perou Turquie	Sri Lanka	Chypre Coree du Sud Malte Portugal Tchecoslovaquie (Rep) Uruguay	Argentine
Kenya Namibie	Algerie Iran	Jamaique	Bulgarie Croatie Hongrie Pologne Slovenie Yougoslavie		Colombie Equateur Inde Mexique
Indonesie Zimbabwe	Afrique du sud Liban Macedoine		Moldavie Roumanie Russie Ukraine		Albanie Chine Costa Rica Guyana Thailande Venezuela

description des sorties avec un exemple: interprétation "variables"





description des sorties avec un exemple: ajout d'une CAH sur les vecteurs codes



a rapprocher de la classification mixte

référence

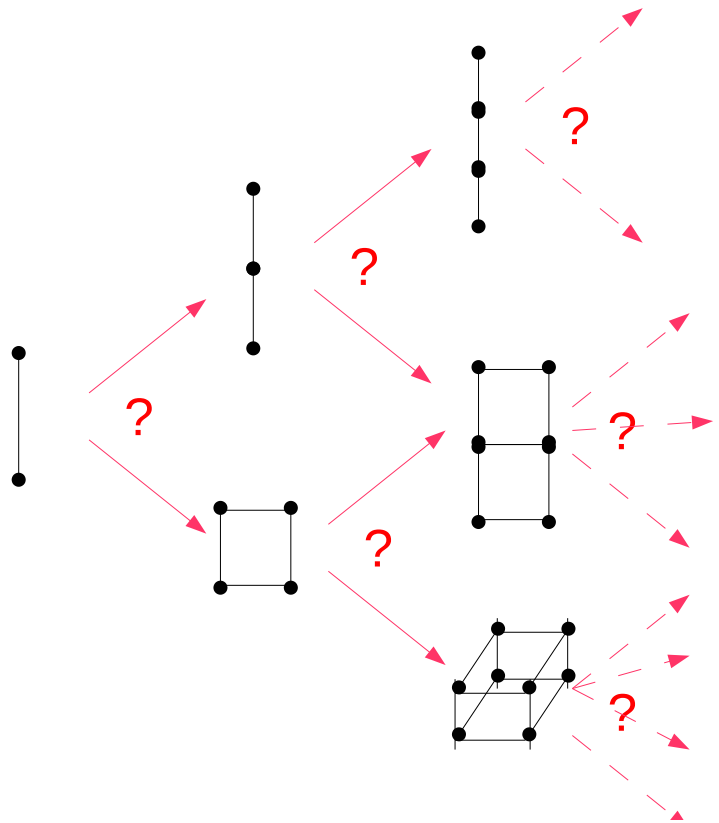
https://samos.univ-paris1.fr/archives/ftp/preprints/samos173.pdf

les Growing SOM

Dans le cas de structures "simples" (des lignes, cartes, pavés, hyperpavés... mais pas de cercles tores sphères bouteilles de Klein (*)) il existe un algorithme permettant "d'auto adapter" les paramêtres des algorithmes SOM.
Th. Villmann, and H. -U. Bauer : Applications of the growing self-organizing map Neurocomputing volume 21, 1998 , p91-100
Il y a une application aux images sattelite dans cette article
(*) Il a récemment été rellement observé une base de donnée en forme de bouteille de Klein

Principe

On part d'une carte a 2 céllule, on fait converger la carte puis toutes les N_0 itérations on se demande dans quelle direction l'agrandir

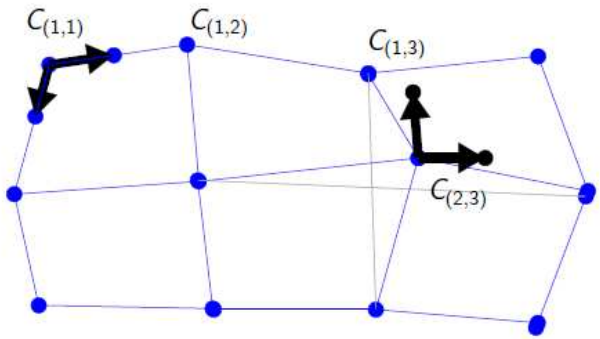


Choix de la direction

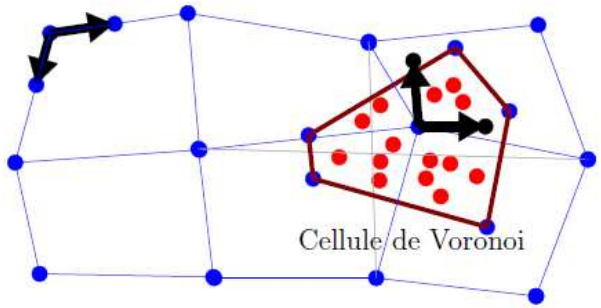
Imaginons qu'on ait obtenu une carte $n_1 \times ... \times n_j$ a une certaine étape Il faut désormais qu'on regarde si on doit augmenter un n_i de 1 ou rajouter une dimension

- Dans un premier temps on calcule $\overrightarrow{u}_i(k)$ le vecteur unitaire tangent a la case k carte dans la direction i.
- Pour chaque point de la base dans la cellule de Voronoi de la carte on recalcule les coordonnées dans la base des $\overrightarrow{u}_i(k)$ complétée par un sous espace orthogonal
- On obtient une décomposition de la variance dans toutes les direction pour l'orthogonal on applique en plus une ACP notons cette variance $\widetilde{a}_i$
- on élit la direction réalisant le max de $\sqrt{n_i(n_i+1)}\widetilde{a}_i$

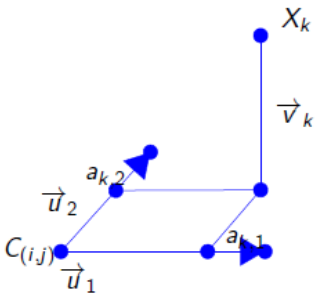
Calcul des vecteurs tangents par cellule



Les points dans la cellule de voronoi



Le calcul des variance



$\tilde{a_i} = Var(a_{k,i})$

$\tilde{a_3}$ est la variane sur le premier axe d'une ACP realisee ur les $\vec{v}_k$

LLE

Nonlinear dimensionality reduction by locally linear embedding. Sam Roweis and Lawrence Saul. Science, v.290 no.5500 , Dec.22, 2000. pp.2323–2326.

Principe

Dans un premier temps on se donne une structure de voisinages sur les poins (par exemple des $k-$plus proches voisins L'idée est simple :

- Si les points sont sur une sous variété lisse (au moins $\mathcal{C}_1$)
- Alors on est localement linéaire (sur le plan tangent)

Score

Alors on devrait pouvoir trouver une matrice W telle que

$$S(W) = \sum_i ||\vec{X_i} - \sum_j W_{i,j} X_j||^2$$

soit petit
avec

- $W_{i,j} = 0$ dès que j n'est pas un voisin de i
- $W_{i,i} = 0$
- $\sum_j W_{i,j} = 1$ (contrainte technique)

En effet on regarde les différences entre les observations et leurs reconstructions linéaires

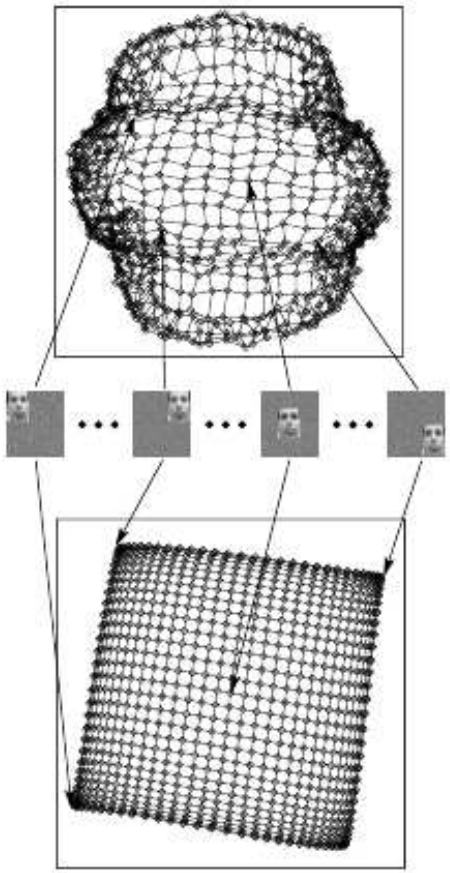
Algorithme

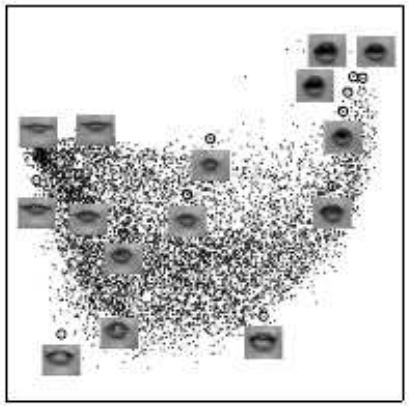
- Recherche du W minimisant $S(W)$ sous les contraintes du transparent précédent. W représente de manière intrinsèque la géométrie des données
- Recherche des Y_i minimisant

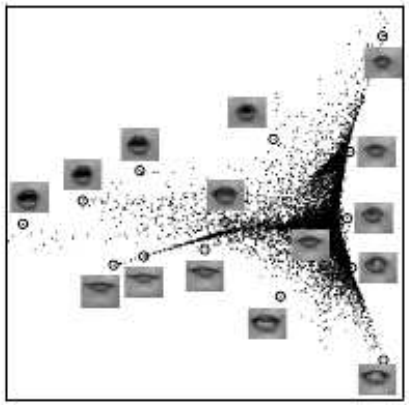
$$S'(Y) = \sum_i ||\vec{Y_i} - \sum_j W_{i,j} Y_j||^2$$

sous la contrainte : les Y_i sont de dimension d fixée (la dimension de l'ensemble sur lequel on souhaite projeter les données)

quelques résultats issus de l'article







"Nonlinear dimensionality reduction"

24 méthodes sur la page wikipédia associée...

La dimension

Sauf pour le cas de Growing Som il faut une idée de la dimension intrinsèque des données. Il existe aujourd'hui deux types de méthodes pour cela :

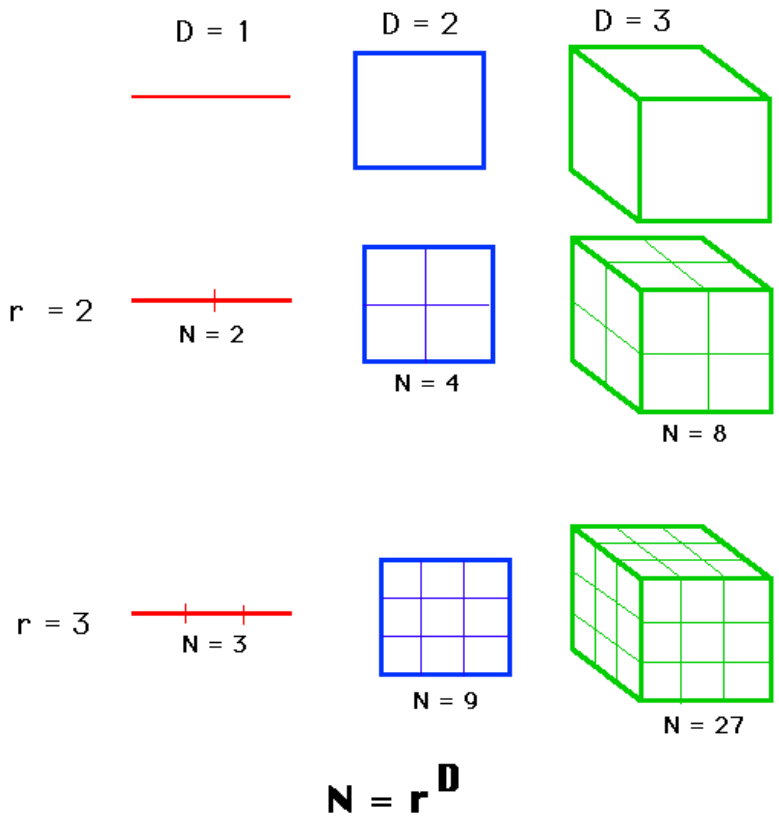
- Les méthodes issues de la dimension fractale
- Les méthodes avec des ACP Locales

La dimension de Hausdorff

La dimension de Hausdorff (définition simplifiée) soit S un sous ensemble compact
soit $N(\varepsilon)$ le nombre minimum de boules de rayon epsilon permettant de recouvrir S
Si la limite existe :

$$d(S) = \lim_{\varepsilon \to 0} \frac{log(N(\varepsilon))}{\log(\varepsilon)}$$

La dimension de Hausdorff



La dimension de Hausdorff

Avantages :

1. Mathématiquement c'est "le mieux"

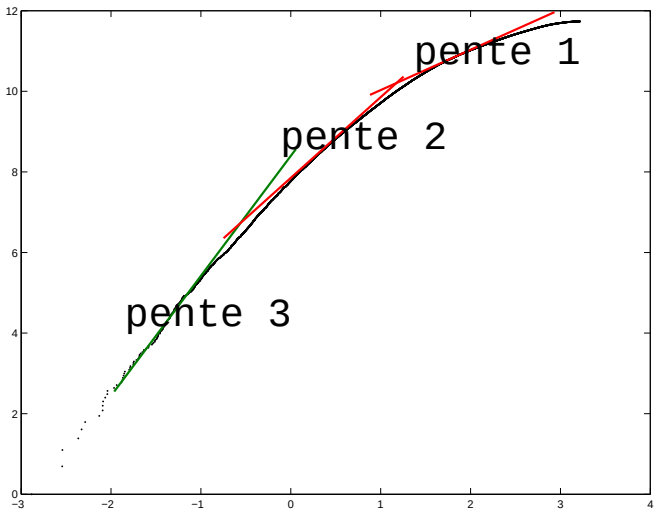
Inconvénients :

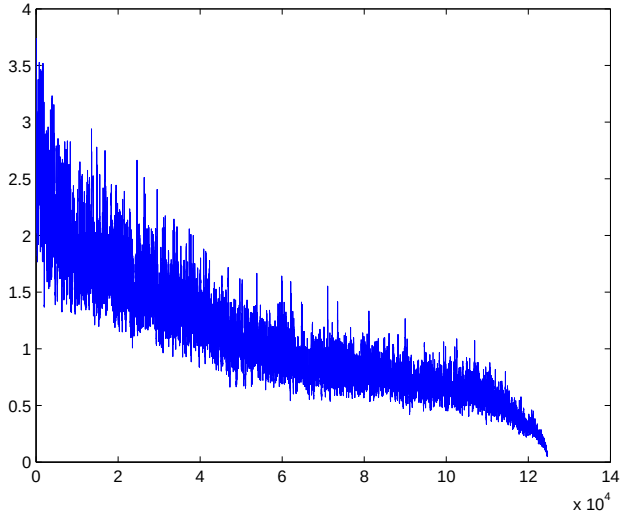
1. Temps de calcul très long
2. En pratique (en statistiques) impossibilité de prendre la limite en 0 donc où prendre la limite ?

La dimension de corrélation

Pour Pallier au problème du temps de calcul de la dimension fractale il est d'usage d'utiliser al dimension de corrélation en statistique :
soit $C(\varepsilon)$ le nombre de couples de points observés distants d'au plus epsilon
sous une hypothèse d'uniformité des données $C(\varepsilon) \sim cste\varepsilon^d$
et on calcule la dimension de corrélation en mesurant la pente de $log(C)$ comme une fonction de $log(\varepsilon)$

La dimension de corrélation





La dimension de corrélation

Avantages :

1. Temps de calcul rapide

Inconvénients :

1. Implicitement lié a un tirage uniforme
2. En pratique (en statistiques) impossibilité de prendre la limite en 0 donc où prendre la limite ?

Les ACP Locales

Plus récemment des méthodes avec des ACP locales ont été crées : en résumé pour chaque point on se définit un vosinage ($k-$plus proches voisins par exemples) et on observe les pourcentages d'inertie expliquée pour l'ensemble des *ACP* locales. Par rapport a la méthode précédente on peut s'abstraire de l'hypothèse du tirage uniforme mais on a toujours le problème du choix des voisinages.

Le choix du voisinage

Beaucoup de méthodes (*Isomap*, *LLE* et *HLLE*) passent par le choix préliminaire d'un voisinage par une méthode des $k-$plus proches voisins.

Attention

- Ce choix est fondamental : il faut trouver un k suffisament petit pour respecter la localité mais pas trop petit pour éviter de créer du bruit