

# Apprentissage Automatique (4/7): Apprentissage non supervisé

S. Herbin, **B. Le Saux**, A. Chan Hon Tong

8 avril 2019

# Introduction

# Apprentissage non-supervisé : pourquoi ?

Problèmes de l'analyse de données:

Données de grande dimension:

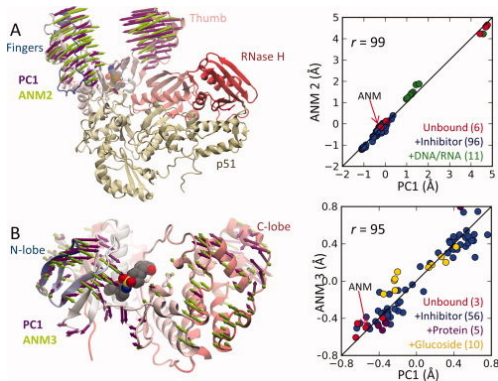
- ▶ Ex : image = qq MPixels
- ▶ Info redondante, non-pertinente
- ▶ Fléau de la dimension : espace vide (exemple : 50 dimensions, 20 niveaux par dimensions  $\implies 20^{50}$  cellules...)

Grand volume de données:

- ▶ Ex: 1 minute de données produites par le collisionneur de particules du CERN = 100 POctets...
- ▶ Temps de traitement extrêmement longs

# Apprentissage non-supervisé : pourquoi ?

Exemple : visualiser et analyser la structure de protéines



Morelles et al., Protein Science 2011

# Apprentissage non-supervisé : objectif

## Définition:

Trouver des structures sous-jacentes à partir de données non étiquetées

## Motivations :

- ▶ La plupart des données ne sont pas étiquetées !
- ▶ Réduire la dimension des données (en enlevant l'information non pertinente, pour réduire les temps de calcul)
- ▶ Catégoriser, classifier (mais sans guide!)
- ▶ Avoir un modèle simple des données, humainement compréhensible

# Apprentissage non-supervisé : plan

## Réduction de dimensionalité

- Analyse en Composantes Principales

- Analyse Discriminante Linéaire

- Pour aller plus loin: t-SNE

## Catégorisation

- K-means

- Approches spectrales, DBSCAN

- Pour aller plus loin : apprentissage de dictionnaire

## Conclusion

# Réduction de dimensionalité

# Analyse en Composantes Principales

Dessinez un poisson...



# Analyse en Composantes Principales

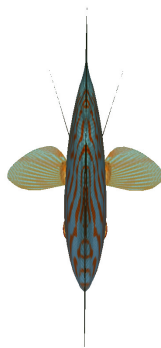
Dessinez un poisson...



# Analyse en Composantes Principales

Dessinez un poisson...

- Les poissons vivent en 3D...



# Analyse en Composantes Principales

Dessinez un poisson...

- ▶ Les poissons vivent en 3D...
- ▶ Comment les représenter sur une feuille 2D ?



# Analyse en Composantes Principales

## Dessinez un poisson...

- ▶ Les poissons vivent en 3D...
- ▶ Comment les représenter sur une feuille 2D ?
- ▶ En choisissant le meilleur point de vue



# Analyse en Composantes Principales

## Dessinez un poisson...

- ▶ Les poissons vivent en 3D...
- ▶ Comment les représenter sur une feuille 2D ?
- ▶ En choisissant le meilleur point de vue
- ▶ Encore mieux : en perspective (Giotto, 1420)



# Analyse en Composantes Principales



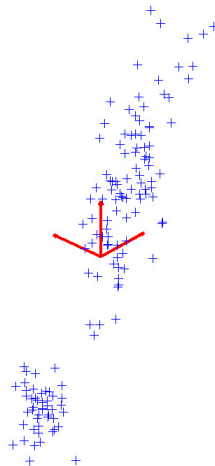
## Analyse en Composantes Principales

L'ACP (ou *PCA* - *Principal Component Analysis*) est une méthode de projection qui permet de représenter au mieux les données d'origine en réduisant le nombre de dimensions.

# Analyse en Composantes Principales : Formalisme

## Algèbre Linéaire (*Rappel*)

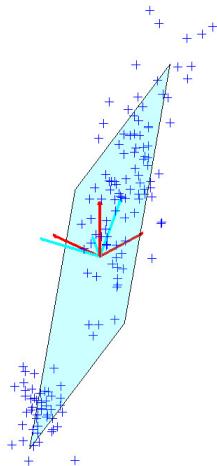
- ▶ Espace vectoriel  $E$ : structure permettant des combinaisons linéaires de *vecteurs*  
 $\mathbf{x}_k = (x_k^1, \dots, x_k^n)$
- ▶ Base  $B$  : famille de vecteurs *libre* et *génératrice*



# Analyse en Composantes Principales : Formalisme

## Algèbre Linéaire (*Rappel*)

- ▶ Espace vectoriel  $E$ : structure permettant des combinaisons linéaires de *vecteurs*  
 $\mathbf{x}_k = (x_k^1, \dots, x_k^n)$
- ▶ Base  $B$  : famille de vecteurs *libre* et *génératrice*
- ▶ Changement de base :  
endomorphisme  $E \rightarrow E$ ,  $B \mapsto B'$ .
- ▶ Projection : Application linéaire de  $E \rightarrow F$ , ss-EV de  $E$ .

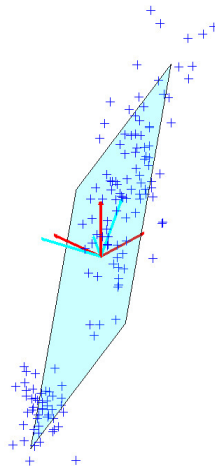




# Analyse en Composantes Principales : Formalisme

## Objectif géométrique de l'ACP

L'ACP est la recherche du sous-espace de projection qui permet la représentation la plus fidèle des variables dans un sous-espace de dimension réduite.



# Analyse en Composantes Principales : Formalisme

## Statistiques (*Rappel*)

Soient  $X, Y$  2 Variables Aléatoires

- ▶ Moyenne  $\bar{x} = \frac{1}{N} \sum x$
- ▶ Variance  $\sigma_X^2 = \frac{1}{N-1} \sum (x - \bar{x})^2$  : mesure de la dispersion
- ▶ Covariance  $\sigma_{X,Y} = \frac{1}{N-1} \sum (x - \bar{x})(y - \bar{y})$  : mesure prop. à la corrélation

Soit  $\mathbf{X} = (X^1, \dots, X^n)$  un vecteur aléatoire :

- ▶ Matrice de Variance-Covariance

$$\text{Var}(\mathbf{X}) = \begin{pmatrix} \sigma_{X^1}^2 & \sigma_{X^1 X^2} & \cdots & \sigma_{X^1 X^n} \\ \sigma_{X^1 X^2} & \sigma_{X^2}^2 & \cdots & \sigma_{X^2 X^n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X^1 X^n} & \sigma_{X^2 X^n} & \cdots & \sigma_{X^n}^2 \end{pmatrix}$$

# Analyse en Composantes Principales : Formalisme

## Objectif statistique de l'ACP

$$\begin{pmatrix} \sigma_{X^1}^2 & \sigma_{X^1 X^2} & \cdots & \sigma_{X^1 X^n} \\ \sigma_{X^1 X^2} & \sigma_{X^2}^2 & \cdots & \sigma_{X^2 X^n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{X^1 X^n} & \sigma_{X^2 X^n} & \cdots & \sigma_{X^n}^2 \end{pmatrix} \rightarrow \begin{pmatrix} \sigma_{X^1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{X^2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{X^n}^2 \end{pmatrix}$$

L'ACP cherche à :

- ▶ Maximiser la dispersion sur les 1ères dimensions de la nouvelle base :  $\sigma_{X^i} \gg 0$  et  $\sigma_{X^i} > \sigma_{X^j} \forall i > j$
- ▶ Décorréliser chaque dimension :  $\sigma_{X^i X^j} \rightarrow 0$

# Analyse en Composantes Principales : Algorithme

## Algorithme

Échantillon  $\{\mathbf{x}_k = (x_k^1, \dots, x_k^n)_{1 \leq k \leq p}\}$ , réalisations d'un vecteur aléatoire  $\mathbf{X} = (X^1, \dots, X^n)$ .  $M$  matrice  $n \times p$  des vecteurs en lignes.

1. Centrer l'échantillon  $\forall i X^i \mapsto X^i - \bar{X}^i$ , tq :  $B = M - \bar{M}$
2. Construire la matrice de variance-covariance  
$$\text{Var}(\mathbf{X}) = \frac{1}{p-1} B^T B$$
3. Diagonaliser la matrice de variance-covariance <sup>1</sup>:

$$\text{Var}(\mathbf{X}) = P \Delta P^T$$

4. Trier les valeurs propres par ordre décroissant (et les vecteurs propres de  $P$ )

⇒ On obtient la matrice de passage  $P$  et les valeurs propres  $\Delta_i$

# Analyse en Composantes Principales : Propriétés

## Propriétés

- ▶ Matrice de passage  $P = (\mathbf{u}^1, \dots, \mathbf{u}^n)$  composée des vecteurs de la nouvelle base (même dimension) :

$$T = PM$$

- ▶ Matrice de projection dans la base d'un sous-espace vectoriel optimal pour la représentation  $P_{1 \rightarrow l} = (\mathbf{u}^1, \dots, \mathbf{u}^l)$  pour  $l \leq n$  :

$$T = P_{1 \rightarrow l} M$$

# Analyse en Composantes Principales : Propriétés

## Propriétés

- ▶ Vecteurs propres  $P = (\mathbf{u}^1, \dots, \mathbf{u}^n)$  associés aux valeurs propres  $\Delta_i \propto \sigma_{X_i}^2$  triées par ordre décroissant.
- ▶ Variance  $\propto$  information statistique portée par la dimension.  
Lien avec la théorie du signal :
  - ▶ les composantes principales avec une large dynamique représentent le signal,
  - ▶ celles avec une faible variance constituent le bruit.

# Analyse en Composantes Principales : Exemples

## Revenons à nos poissons

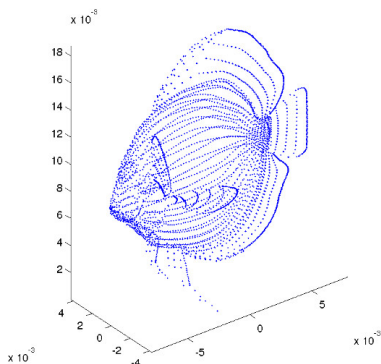
Points  $\in \mathbb{R}^3$  répartis sur la surface du Discus Alenquer

Variances:

$$\Delta \propto \begin{pmatrix} 0.17 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

Base des vecteurs propres:

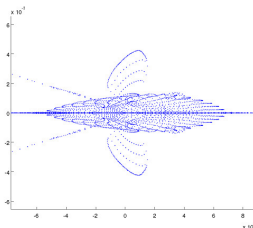
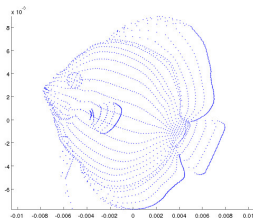
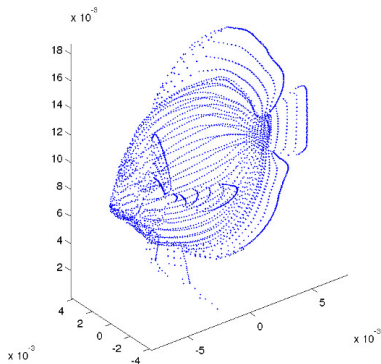
$$P = \begin{pmatrix} 0.91 & -0.42 & 0 \\ 0 & 0 & 1 \\ 0.42 & 0.91 & 0 \end{pmatrix}$$



# Analyse en Composantes Principales : Exemples

Revenons à nos poissons

Projections sur les 2 premières (ou dernières) composantes

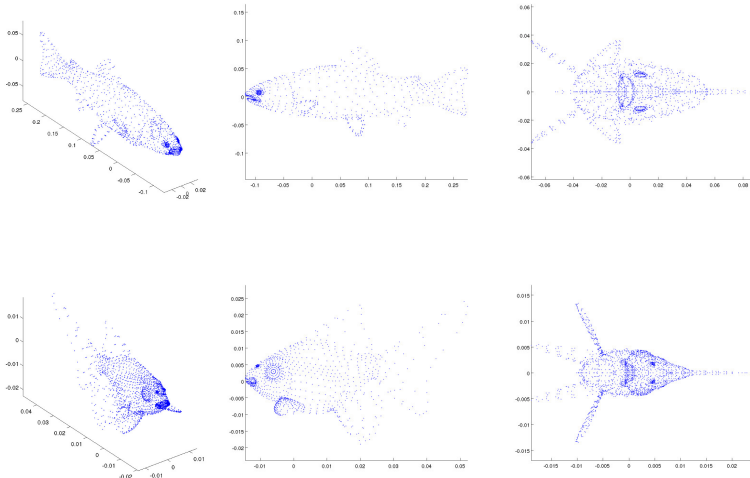




# Analyse en Composantes Principales : Exemples

Plus de poissons...

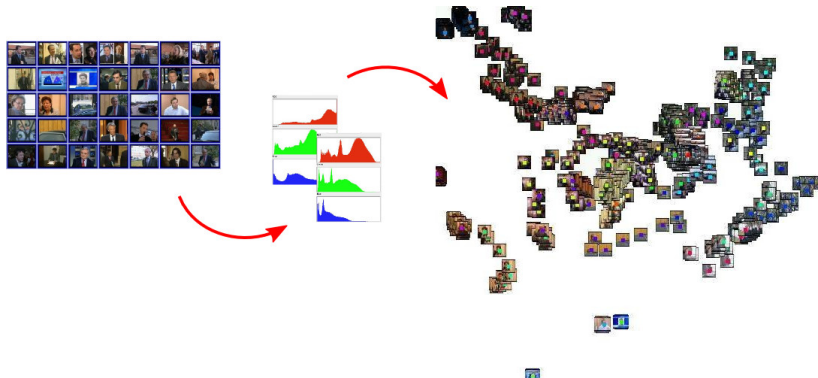
3D vs. 1ères CP (= représentation canonique) vs. dernières CP



# Analyse en Composantes Principales : Exemples

## Plus complexe : analyse de vidéos

Images vidéos caractérisées par des histogrammes de couleurs et visualisées selon les 2 premières composantes issues de l'ACP



Le Saux, 2003

# Analyse en Composantes Principales : Résumé

---

## Points clés de l'ACP

- ▶ Représenter des données de grande dimension
- ▶ Réduire la dimension
- ▶ Décorrélérer les variables
- ▶ Basé sur la diagonalisation de la matrice de variance-covariance des données (*vecteurs*)

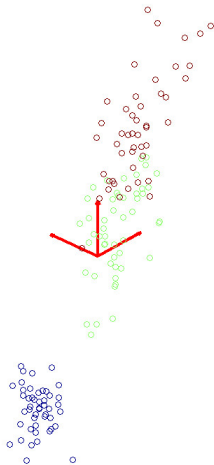
---

## Utilisations

- ▶ Pré-traitement pour l'analyse de données (cf. cours 1, 2, 3)
- ▶ Visualisation

# Analyse Discriminante Linéaire

- ▶ L'ACP optimise la variance globale d'un ensemble de données  
 $X = \{\mathbf{x}_k \in \mathbb{R}^n\}_{1 \leq k \leq p}$
- ▶ Peut-on faire mieux quand les données appartiennent à des sous-groupes connus ?
- ▶ Données étiquetées (ou *labellisées*)  
- une couleur par groupe



# Analyse Discriminante Linéaire

- ▶ Données étiquetées (ou *labellisées*)
- ▶ Ensemble de couples de données avec leur groupe respectif noté  $(X, Y) = \{(\mathbf{x}_k, y_k), \mathbf{x}_k \in \mathbb{R}^n, y_k \in \{1, \dots, C\}\}_{1 \leq k \leq p}$

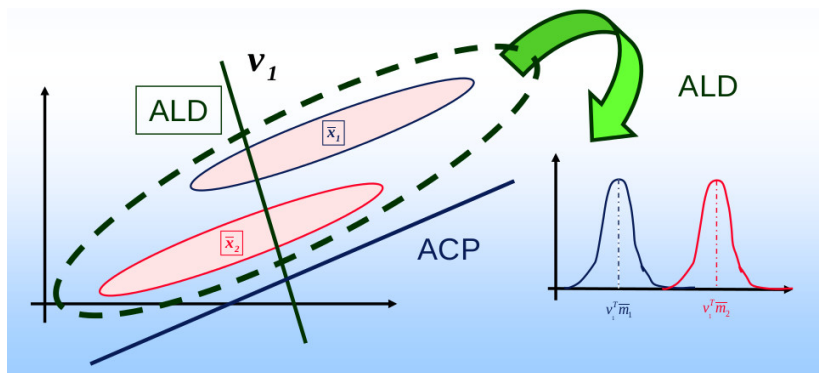
## Objectifs

- ▶ En classification on vise la **séparabilité** des données
- ▶ Mettre en évidence les **différences** entre classes

# Analyse Discriminante Linéaire

## Cas à 2 classes

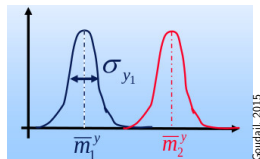
- ▶ Recherche du vecteur unitaire  $\mathbf{u}$  de la droite ALD tel que les 2 groupes sont séparés au mieux après projection
- ▶ Après projection en 1D,  $\mathbf{x}'_k = \mathbf{u}^T \mathbf{x}_k$



# Analyse Discriminante Linéaire

## Cas à 2 classes

- ▶ Après projection en 1D,  $\mathbf{x}'_k = \mathbf{u}^T \mathbf{x}_k$
- ▶ La séparabilité des données est quantifiée par le critère de Fisher : 
$$f(\mathbf{u}) = \frac{(m'_1 - m'_2)^2}{\sigma_1^2 + \sigma_2^2}$$

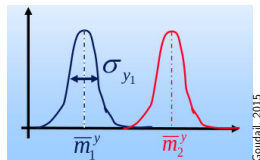


- ▶ Recherche de  $\mathbf{u}$  tel que  $f(\mathbf{u})$  est maximisée.

# Analyse Discriminante Linéaire

## Cas à 2 classes

- Critère de Fisher :  $f(\mathbf{u}) = \frac{(m'_1 - m'_2)^2}{\sigma_1^2 + \sigma_2^2}$



Goudail, 2015

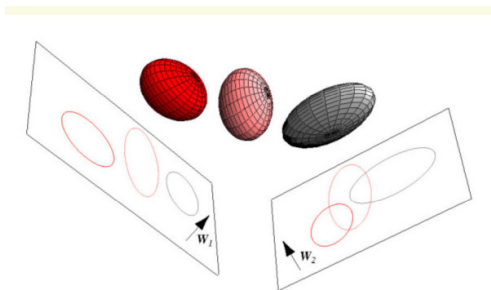
- Matrice de covariance inter-classe  $\Gamma_B$  à maximiser:  
 $(m'_1 - m'_2)^2 = \mathbf{u}^T \Gamma_B \mathbf{u}$
- Matrices de covariance intra-classe  $\Gamma_c, c = 1, 2$  à minimiser:  
 $\Gamma_c = \frac{1}{N}(\mathbf{x} - \mathbf{m}_c)(\mathbf{x} - \mathbf{m}_c)^T$



# Analyse Discriminante Linéaire

## Cas multiclass

- ▶ Projection dans un sous-espace  $\mathbf{x}_k' = V\mathbf{x}_k$
- ▶ Soient:
  - ▶  $p^i$  le nombre d'exemples de la classe  $i$ ,
  - ▶  $\bar{\mathbf{x}}^i = \frac{1}{p^i} \sum_{\mathbf{x} \in \text{classe } i} \mathbf{x}_k^i$  la moyenne de la classe  $i$ ,
  - ▶ et  $\bar{\mathbf{x}} = \frac{1}{p} \sum_{\mathbf{x}_k} \mathbf{x}_k^i$  la moyenne globale.



in: Pattern Classification, Duda, Hart & Stork, 2001

## Cas multiclasse

- ▶ Fonction objective à minimiser de l'Analyse Discriminante

Multiple:  $j(V) = \frac{\det(V^t S_B V)}{\det(V^t S_W V)}$

- ▶ Avec la matrice de variance intra-classe :

$$S_W = \sum_{i=1}^c \sum_{\mathbf{x} \in \text{classe } i} (\mathbf{x}_k - \bar{\mathbf{x}}^i)(\mathbf{x}_k - \bar{\mathbf{x}}^i)^t$$

- ▶ Et la matrice de variance inter-classes :

$$S_B = \sum_{i=1}^c n^i (\bar{\mathbf{x}}^i - \bar{\mathbf{x}})(\bar{\mathbf{x}}^i - \bar{\mathbf{x}})^t$$

# Analyse Discriminante Linéaire

## Algorithme

1. Calcul des moyennes de classe  $\bar{x}^i$  (dimension  $n$ )
2. Calcul des matrices de variance intra-classe  $S_W$  et inter-classes  $S_B$
3. Calcul des vecteurs propres  $\mathbf{v}_1, \dots, \mathbf{v}_n$  et valeurs propres  $\lambda_1, \dots, \lambda_n$  pour les matrices de variance
4. Tri des vecteurs propres par ordre décroissant des valeurs propres
5. Sélection des  $f$  vecteurs propres associés aux plus grandes  $\lambda_k$   
→ matrice de passage  $V$  (dimension  $d * f$ ) où chaque colonne est un vecteur propre  $v_k$ .
6. Projection dans le sous-espace :  $\mathbf{x}_k' = V\mathbf{x}_k$

# Analyse Discriminante Linéaire

## Propriétés

- ▶ Matrice de passage  $V_n = (\mathbf{v}^1, \dots, \mathbf{v}^n)$  composée des vecteurs de la nouvelle base (même dimension) :  $T = PM$
  - ▶ Matrice de projection dans la base d'un sous-espace vectoriel optimal pour la **classification**  $V_{1 \rightarrow f} = (\mathbf{v}^1, \dots, \mathbf{v}^f)$  pour  $f \leq n$  :  $T = P_{1 \rightarrow f} M$
- ▶ L'Analyse Discriminante Linéaire est une ACP sur les vecteurs moyens de chaque classe, normalisée par la variance intra-classe

# Analyse Discriminante Linéaire : Résumé

## Points clés de l'ADL

- ▶ Représenter des données **labelisées** de grande dimension
- ▶ Optimiser la **séparabilité** des données projetées
- ▶ Basé sur la variance intra-classe (à minimiser) et de la variance inter-classes (à maximiser)

## Utilisations

- ▶ Pré-traitement pour l'analyse de données (cf. cours 1, 2, 3)
- ▶ Classification sommaire
- ▶ Visualisation

# Réduction de dimension : Pour aller plus loin, **t-SNE**

## Définition

- ▶ t-SNE = t-distributed stochastic neighbor embedding
- ▶ Approche **non-linéaire** de réduction de dimension, proposée par [Van Der Maaten & Hinton, 2008]

## Objectifs

- ▶ Trouver un mapping en faible dimension qui reflète au mieux les **similarités** entre les observations dans l'espace de départ.

# Réduction de dimension : Pour aller plus loin, **t-SNE**

## Algorithme (1/2)

Soient  $N$  observations  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$  en grande dimension.

1. Calcul des probabilités de **similarité** des observations:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)} \text{ et } p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

2. ...

## Objectifs

t-SNE cherche à apprendre une représentation  $d$ -dimensionnelle  $\mathbf{y}_1, \dots, \mathbf{y}_N$  (avec  $\mathbf{y}_i \in \mathbb{R}^d$ ) qui reflète au mieux les similarités  $p_{ij}$ .

# Réduction de dimension : Pour aller plus loin, **t-SNE**

## Algorithme (2/2)

Soient  $N$  observations ( $\mathbf{x}_1, \dots, \mathbf{x}_N$ ) en grande dimension.

1. Calcul des probabilités de **similarité** des observations:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)} \text{ et } p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

2. Définition des probabilités de **similarité** des représentations cibles selon une distribution de Student :

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k, m, k \neq m} (1 + \|\mathbf{y}_k - \mathbf{y}_m\|^2)^{-1}}$$

3. Minimisation de la divergence de Kullback-Leibler de  $Q$  par rapport à  $P$  :  $KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

- ▶ Distrib. de Student force les individus dissimilaires à être éloignés
- ▶  $KL(P||Q)$  facilement dérivable, donc optimisation par descente de gradient



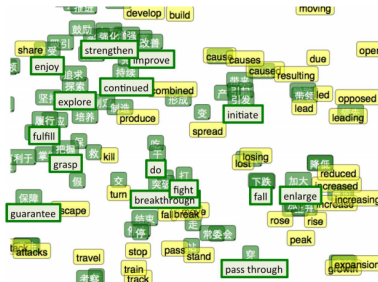
# Réduction de dimension : Pour aller plus loin, **t-SNE**

## t-SNE sur base d'images



Van Maaten, <https://lvdmaaten.github.io/t-sne/>

## t-SNE sur mots chinois + anglais



Zou et al., ENNLP'13

---

## Points clés de t-SNE

- ▶ Approche **non-linéaire** de réduction de données
- ▶ Optimiser la **distrib. des similarités** entre observations et projections
- ▶ Dans l'espace de proj., la **distrib. de Student** éloigne les individus dissimilaires
- ▶ Optimisation par **descente de gradient** (cf. cours réseaux de neurones 5, 6)

---

## Utilisations / liens

- ▶ Pré-traitement pour l'analyse de données (cf. cours 1, 2, 3)
- ▶ Visualisation
- ▶ Autre approche pour apprendre un espace de représentation : les **auto-encodeurs** (cf. cours 6)

# Catégorisation

# Catégorisation

## Définition

- ▶ Trouver des catégories d'objets proches ou similaires
- ▶ Synonymes : partitionnement, *clustering*...
- ▶ ... classification **non-supervisée** : des données  $\{x_i | i \in \{1..N\}\}$  dans  $\mathbb{R}^n$  mais pas de labels

## Objectifs

- ▶ Trouver les groupes de données *proches* dans  $\mathbb{R}^n$  (*notion de distance*)
- ▶ Mettre en évidence des catégories ("*Qui se ressemble s'assemble*")

# Catégorisation : K-means

## K-means

- ▶ Soit  $K$  le nombre de groupe cherchés.
- ▶ Un groupe (d'indice  $j \in \{1 \cdots K\}$ ) = un ensemble de points.
- ▶ Soit  $u_{ji} \in \{0, 1\}$  l'appartenance de chaque  $x_i$  au groupe  $j$
- ▶ Soient  $B = \{\beta_j | j \in \{1 \cdots K\}\}$  les prototypes qui caractérisent ces groupes.

L'algorithme **K-means** minimise :

$$\text{▶ } J_{B,U}(X) = \sum_{j=1}^K \sum_{i=1}^N (u_{ji})^m d^2(x_i, \beta_j)$$

# Catégorisation : K-means

## Algorithme

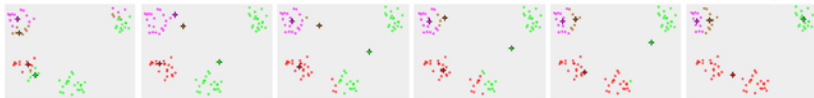
Initialiser les  $\beta_j$ , puis itérer :

1. Assigner chaque donnée  $x_i$  au plus proche  $\beta_j$
2. Recalculer les prototypes selon:  $\beta_j = \frac{\sum_{i=1}^N u_{ji} * x_i}{\sum_{i=1}^N u_{ji}}$  (moyenne des observations du groupe)

# Catégorisation : K-means

## Propriétés

- ▶ L'algorithme fait diminuer la fonction de coût  $J_{B,U}(X)$  à chaque itération.
- ▶ Il y a un nombre fini de  $K$  partitions possible, donc l'algorithme **converge**.
- ▶ Mais la solution peut ne pas être optimale (minimum local) !



- ▶ importance de l'**initialisation**
- ▶ Par exemple: choisir  $\beta_j$  parmi les observations  $x_j$ ...

## Variante statistique :

- ▶ Paramètres d'un *Modèle de Mélange de Gaussiennes (GMM)* estimé par l'algorithme *Expectation-Maximisation*
- ▶  $x_i$  réalisation d'un V.A. modélisée par un Mélange de Gaussiennes :  $p(x_i) = \sum_1^K p(x_i|k)P(k)$
- ▶  $d(x_i, \mu_k) \longrightarrow p(x_i|k) \propto \exp(-||x_i - \mu_k||^2/2\sigma_k^2)$  (Gaussienne simplifiée)
- ▶ Paramètres à estimer :  $\forall k : P(k), \mu_k, \sigma_k$



# Catégorisation

## Variantes et trucs :

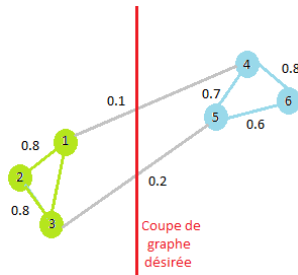
- ▶ Fuzzy C-means : appartenance  $u_i^j \in [0, 1]$
- ▶ Formes variées : distance de Mahalanobis (FCM) / Matrice de covariance complète (GMM)
- ▶ Données aberrantes : si  $\forall k d(x_i, \mu_k), x_i \mapsto$  catégorie bruit
- ▶ Critères pour estimer le nombre de catégorie :
- ▶ Initialisation des  $\mu_k$  : uniformément dispersés parmi les données

# Catégorisation : catalogue de méthodes alternatives

## Partitionnement spectral:

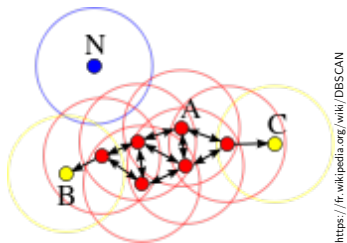
- ▶ Matrice de similarité,
- ▶ Réduction de dimension (1ers vecteurs propres)
- ▶ K-means

⇒ *objets complexes, non vectoriels*



[https://fr.wikipedia.org/wiki/Partitionnement\\_spectral](https://fr.wikipedia.org/wiki/Partitionnement_spectral)

# Catégorisation : catalogue de méthodes alternatives



## DBSCAN:

- ▶ Partitionnement des données en catégories de *MinPts* points se trouvant dans un rayon  $\epsilon$
- ▶ Parcours de proche en proche de tous les points pour ajouter des points à la catégorie courante

⇒ *estime automatiquement le nombre catégories,*

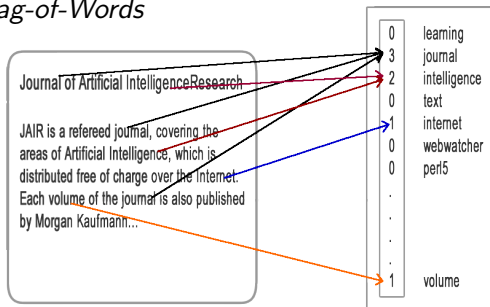
⇒ *gère les données aberrantes*

# Catégorisation : pour aller plus loin

## Apprentissage de dictionnaire:

- ▶ Estime un dictionnaire (=ensemble d'éléments de base) qui représente un ensemble de données
- ▶ Encode chaque donnée en fonction du dictionnaire (*sparse encoding*)

Exemple : *Bag-of-Words*

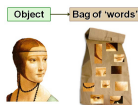


# Catégorisation : pour aller plus loin

## Apprentissage de dictionnaire:

- ▶ Estime un dictionnaire (=ensemble d'éléments de base) qui représente un ensemble de données
- ▶ Encode chaque donnée en fonction du dictionnaire (*sparse encoding*)
- ▶ Notion-clé : parcimonie (*sparsity*); une donnée est représentée par seulement quelques éléments du dictionnaire

Exemple : *Bag-of-Words* pour les images



# Catégorisation : Résumé

---

## Points clés du clustering

- ▶ Regrouper des données **non-labelisées** en catégories
- ▶ Notion de **distance** ou de **similarité** entre échantillons
- ▶ Comme l'ADL, idée de variance intra-classe (à minimiser) et de variance inter-classes (à maximiser)

---

## Utilisations

- ▶ Pré-traitement pour l'analyse de données (cf. cours 1, 2, 3) :  
Bag-of-words, superpixels, etc.
- ▶ Classification **non-supervisée**
- ▶ Visualisation

# Conclusion

# Cours n°3: Sélection de caractéristiques

## Notions phares du jour

- ▶ ACP, LDA, t-SNE
- ▶ Catégorisation, classification non-supervisée
- ▶ Préparation des données (*pre-processing*)

## Concepts généraux

- ▶ Grande dimension / Big data (*curse of dimensionality*)
- ▶ Caractéristiques, espace de représentation (projection et mapping, linéaire et non-linéaire)
- ▶ Similarité intra-classe, variance inter-classes