

Formation Machine Learning

Réduction de dimensionnalité, clustering et détection d'anomalie

Giraud François-Marie

5 Juin 2019



ENI Service

Machine Learning

Réduction de la dimensionalité

Comment appréhender des données en grande dimension ?

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,D} \\ X_{2,1} & X_{2,2} & \dots & X_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N,1} & X_{N,2} & \dots & X_{N,D} \end{bmatrix}$$

Curse of dimensionality ?

- Sélection de dimensions
- Projections linéaires (ACP, LDA, ...)
- Projections non-linéaires (kernels, neural network embeddings, ...)

Sélection de dimensions :

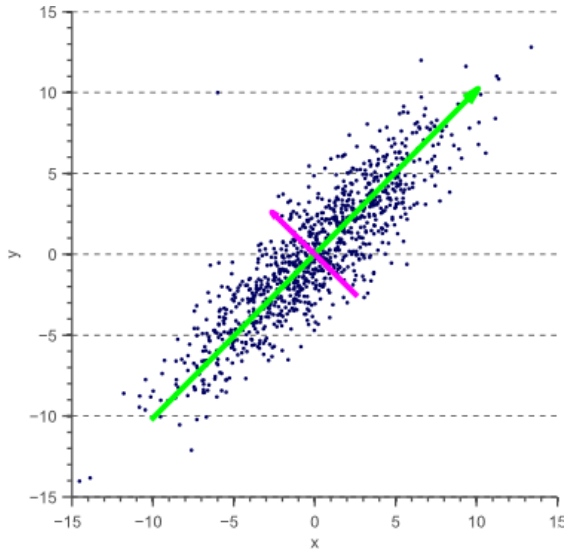
- Random forest
- SVM
- ...

Machine Learning

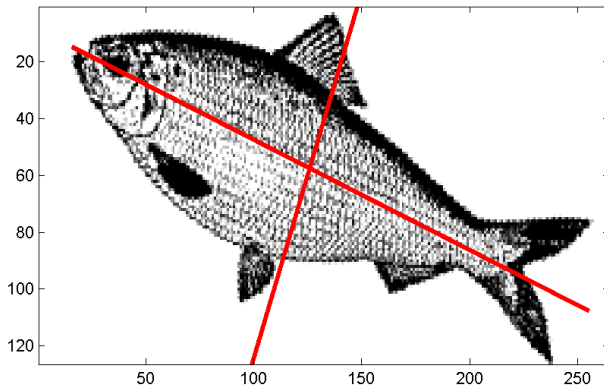
Réduction de la dimensionalité : Projections linéaires

- Principal Component Analysis (Non-supervisée)
- Linear Discriminant Analysis (Supervisée)

Réduction de la dimensionalité : PCA



Réduction de la dimensionalité : PCA



$$X = \begin{bmatrix} X_{1,1} & \dots & X_{1,D} \\ \vdots & \ddots & \vdots \\ X_{N,1} & \dots & X_{N,D} \end{bmatrix}$$

Réduction de la dimensionalité : PCA

chaque dimension est centrée (et réduite) :

$$\bar{X} = \begin{bmatrix} X_{1,1} - \bar{X}_1 & \dots & X_{1,D} - \bar{X}_D \\ \vdots & \ddots & \vdots \\ X_{N,1} - \bar{X}_1 & \dots & X_{N,D} - \bar{X}_D \end{bmatrix}$$

ou

$$\tilde{X} = \begin{bmatrix} \frac{X_{1,1} - \bar{X}_1}{\sigma(X_1)} & \dots & \frac{X_{1,D} - \bar{X}_D}{\sigma(X_D)} \\ \vdots & \ddots & \vdots \\ \frac{X_{N,1} - \bar{X}_1}{\sigma(X_1)} & \dots & \frac{X_{N,D} - \bar{X}_D}{\sigma(X_D)} \end{bmatrix}$$

Réduction de la dimensionalité : PCA

Matrice de covariance (resp. corrélation) :

$$\frac{1}{N} * \bar{X}^T * \bar{X}, \left(\frac{1}{N} * \tilde{X}^T * \tilde{X} \right)$$

ACP :

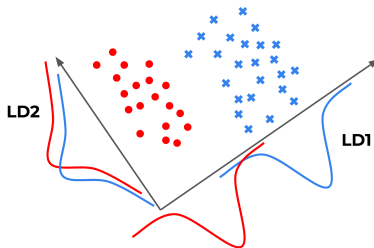
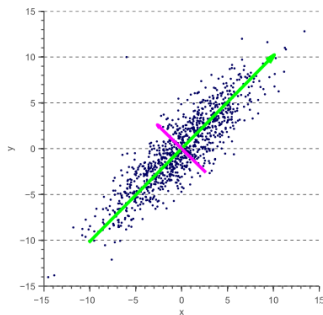
Retrouver les valeurs et vecteurs propres de de la matrice de covariance (resp. corrélation), donc diagonaliser la matrice carrée obtenue.

Vecteur propre : vecteur permettant de projeter les données

Valeur propre : “proportion d’information” conservée par la projection suivant le vecteur propre correspondant

Réduction de dimension : On ne projette que suivant le nombre de vecteurs propres voulus

Linear Discriminant Analysis

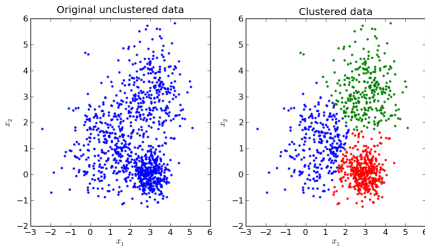


Machine Learning

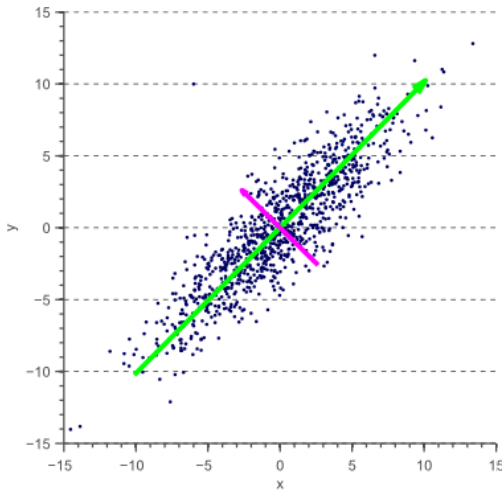
Données Non-Supervisées

Données Non-Supervisées

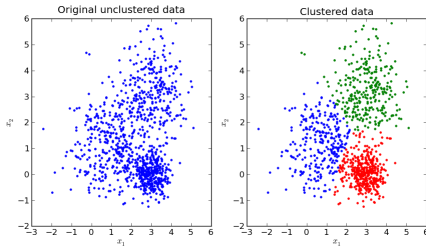
Problème : détection de variables cachées, de “structures” cachées (clusters, variété topologique (manifold), ...)



≈ Réduction de Dimensionnalité



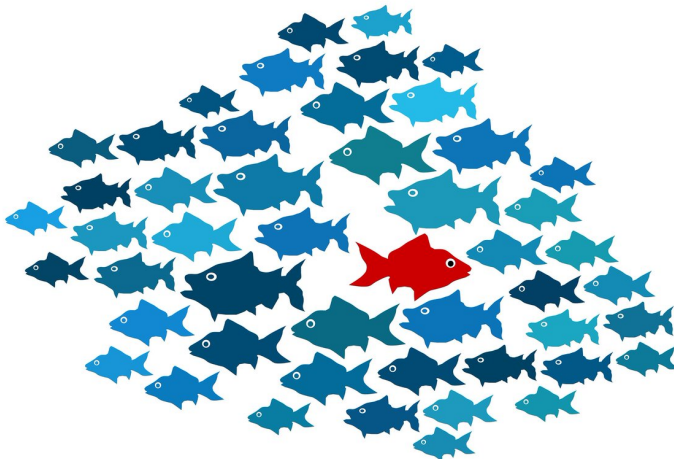
Données Non-Supervisées



Clustering de Clients



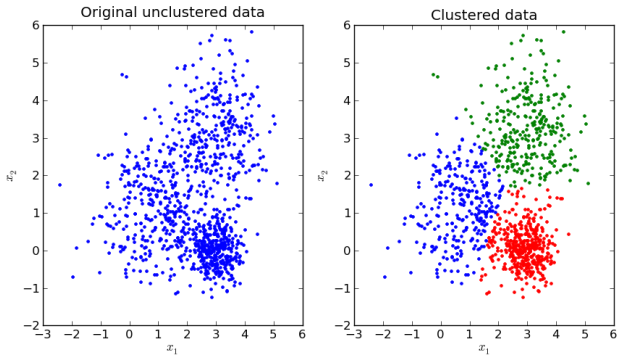
Détection d'anomalie :



Machine Learning

Expectation-Maximisation

Expectation-Maximisation



Expectation-Maximisation

Input : Données, nombre de clusters, métrique

Initialisation Aléatoire

Jusqu'à clusters "stables" :

1. Calculer les "centres" de chaque cluster
2. Réassigner les clusters à tous les points

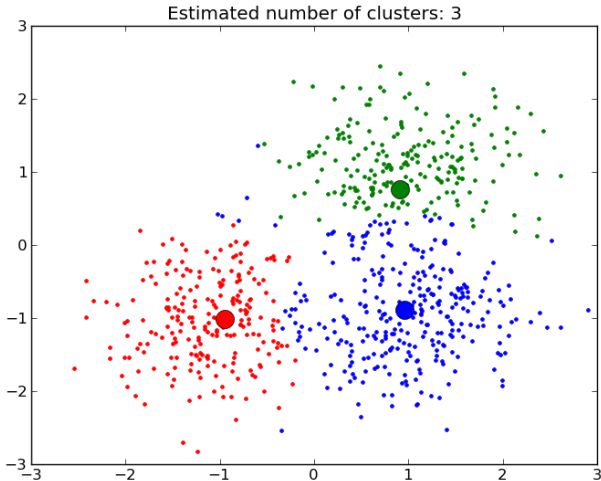
video time



Machine Learning

K-Means

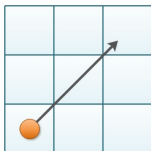
Algorithme EM



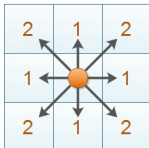
K-Means

distance euclidienne, Manhattan, Chebychev

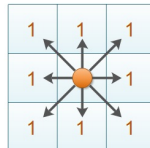
Euclidean Distance



Manhattan Distance



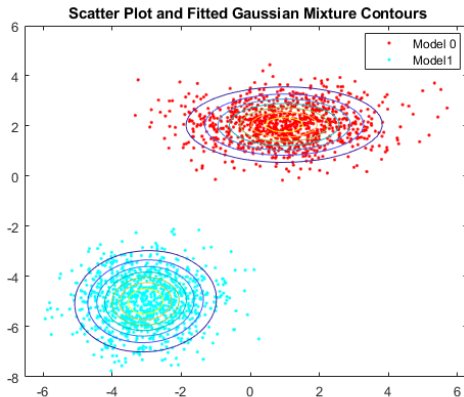
Chebyshev Distance



$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad |x_1 - x_2| + |y_1 - y_2| \quad \max(|x_1 - x_2|, |y_1 - y_2|)$$

Gaussian Mixture Model

Les clusters sont représentés par un centre et une matrice de covariance.



Machine Learning

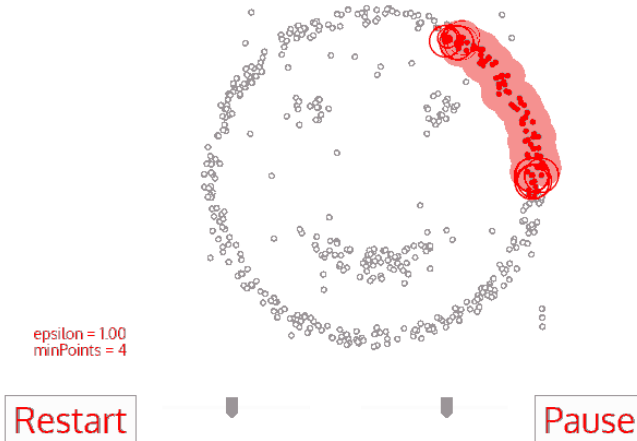
DBSCAN

Density-Based Spatial Clustering of Applications with Noise...

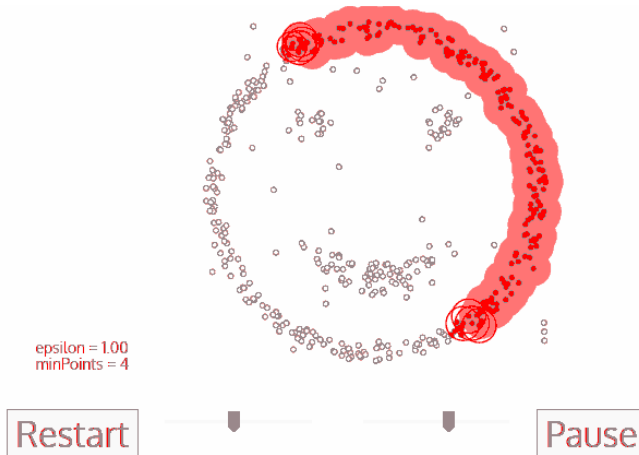
Tant qu'il reste des points non-étiquetés :

1. Prend un point non-étiquetés au hasard et on regarde son voisinage
2. Si (densité $>$ seuil) alors (Nouveau cluster)
 - 2.1 Expansion du cluster de proche en proche dans le voisinage
3. Sinon (Bruit)

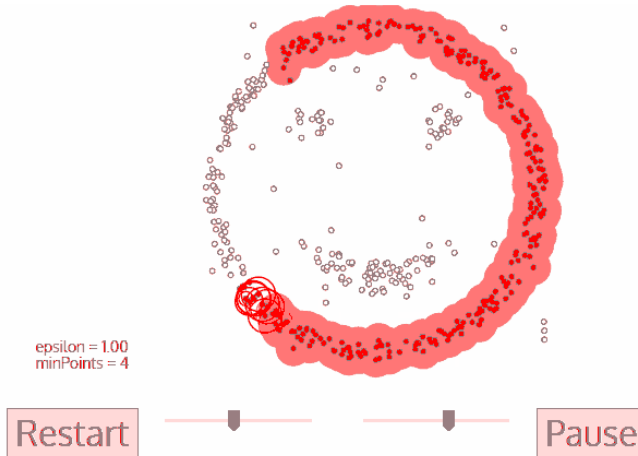
DBSCAN



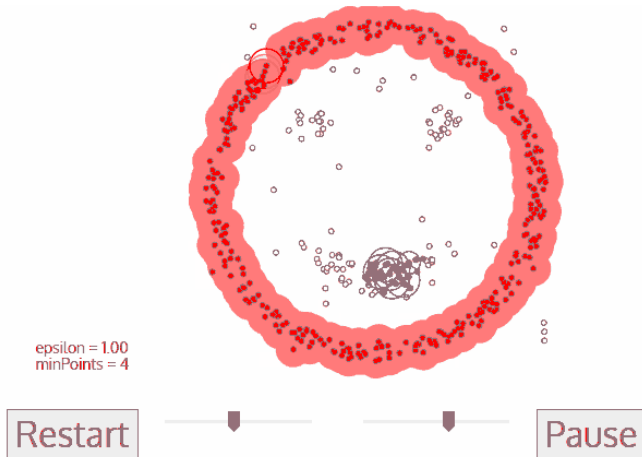
DBSCAN



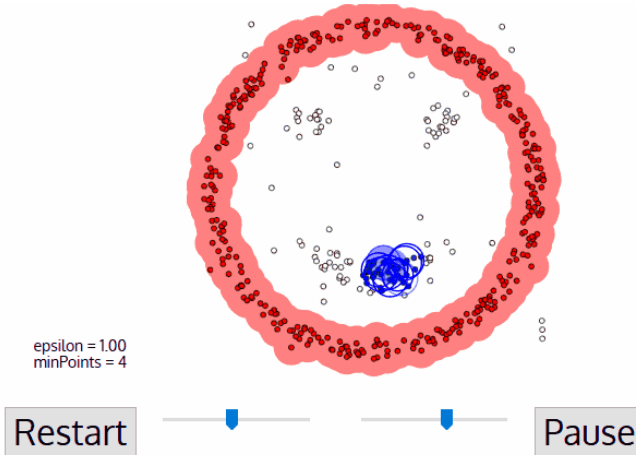
DBSCAN



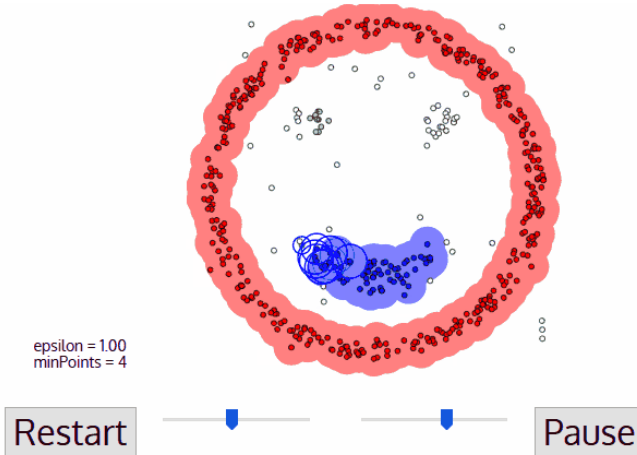
DBSCAN



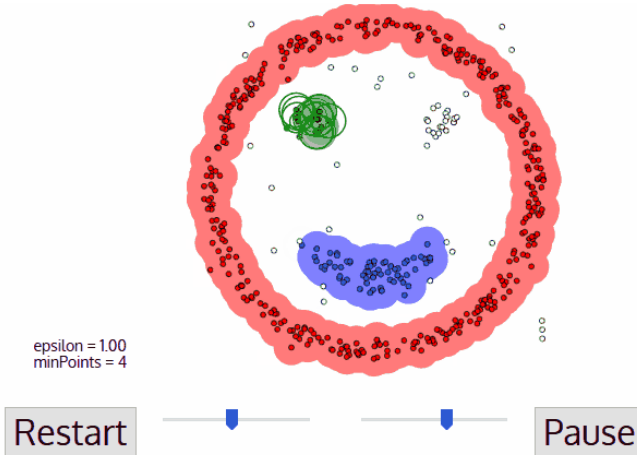
DBSCAN



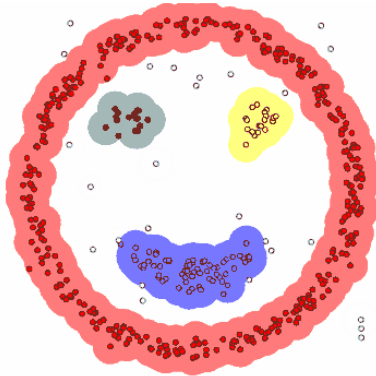
DBSCAN



DBSCAN



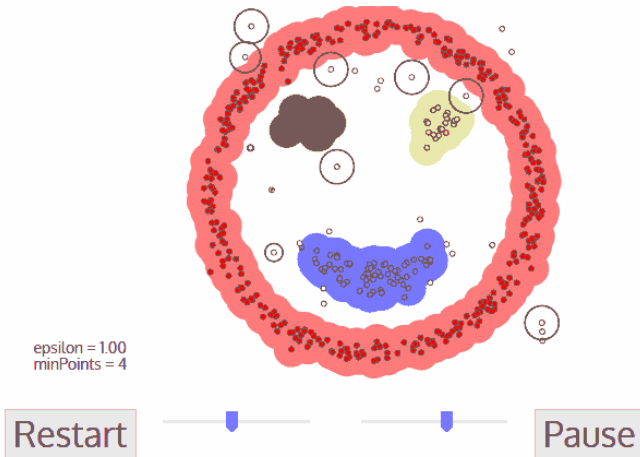
DBSCAN



$\epsilon = 1.00$
 $\text{minPoints} = 4$

Restart

DBSCAN



Machine Learning

Métriques en Non-Supervisé

$$\text{coût} = \sum_i \sum_j \delta_{i,j} |x_j - \mu_i|$$

où $\delta_{i,j}$ vaut 1 si le cluster μ_i est le plus proche du point x_j , 0 sinon

Métrique : Silhouette

Points $x = \{x_1, \dots, x_n\}$, Clusters $\mu = \{\mu_1, \dots, \mu_k\}$.

$$a(x_i) = \frac{1}{\#\mu_i - 1} \sum_j |x_i - x_j|$$

$$b(x_i) = \min_{i \neq j} \frac{1}{\#\mu_j} \sum_j |x_i - x_j|$$

où :

$\#\mu_i$ est le nombre d'éléments de x dans le cluster μ_i

L'ensemble d'indice j ne représente que ceux des points appartenant au cluster μ_j

$a(x_i)$: distance moyenne aux autres points du cluster contenant x_i

$b(x_i)$: distance moyenne aux points du cluster le plus proche

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} , \quad s_i = \begin{cases} 1 - a_i/b_i & \text{if } a_i < b_i \\ 0 & \text{if } a_i = b_i \\ b_i/a_i - 1 & \text{if } a_i > b_i \end{cases}$$

donc $s_i \in [-1, 1]$

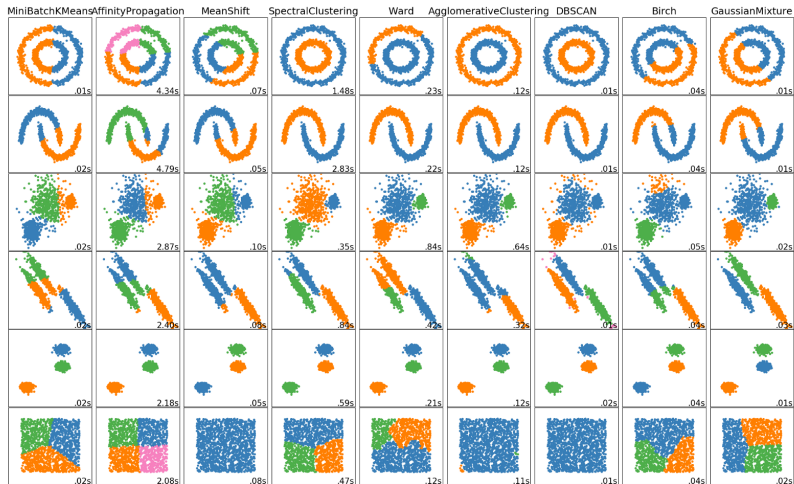
$s_i \approx 1 \iff x_i$ bien clusterisé

$s_i \approx 0 \iff x_i$ au bord de 2 clusters

$s_i \approx -1 \iff x_i$ mal clusterisé

- Calinski-Harabaz index
- Davies-Bouldin Index
- ...

Clustering - Conclusions



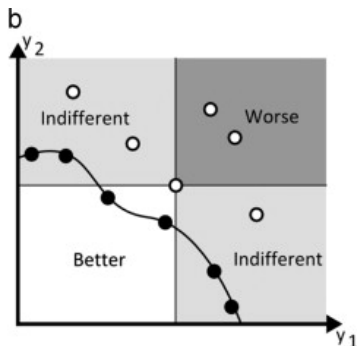
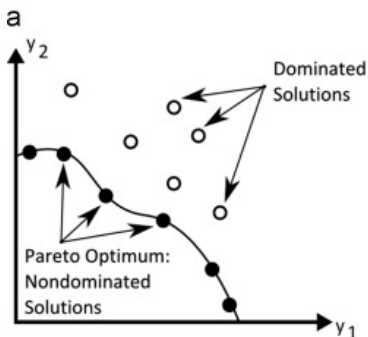
Clustering - Conclusions

Pas de métrique satisfaisante ! \Leftarrow Théorie de la Décision :

Problème qui se mesure en plusieurs dimensions

\Rightarrow

Pas de solution unique !



Machine Learning

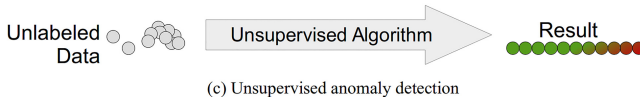
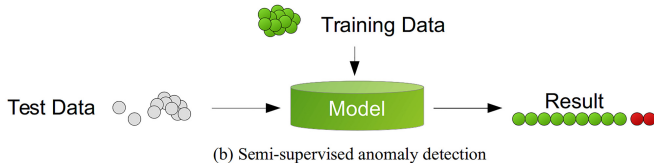
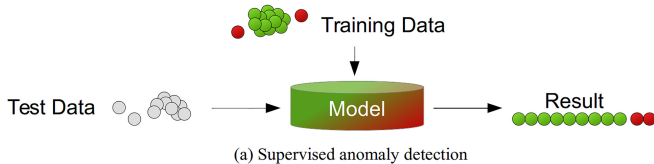
Détection d'Anomalies

Détection :

- de Fraude
- d'Intrusion/Fuite (physique ou électronique)
- Santé (biologique, géologique, machine, ...)

- une anomalie diffère de la norme par ses features
- les anomalies sont rares comparées aux instances normales

Modes de détection d'anomalie



Problème de classification normal.

Réseaux de neurones et SVM très performants.

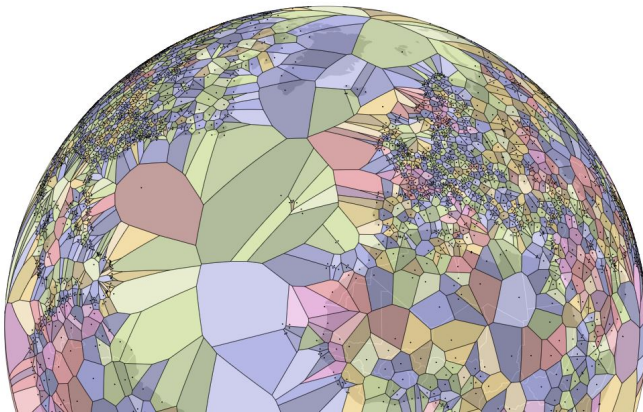
Détection de nouveauté.

Pas traité ici.

One-class SVM très utilisé.

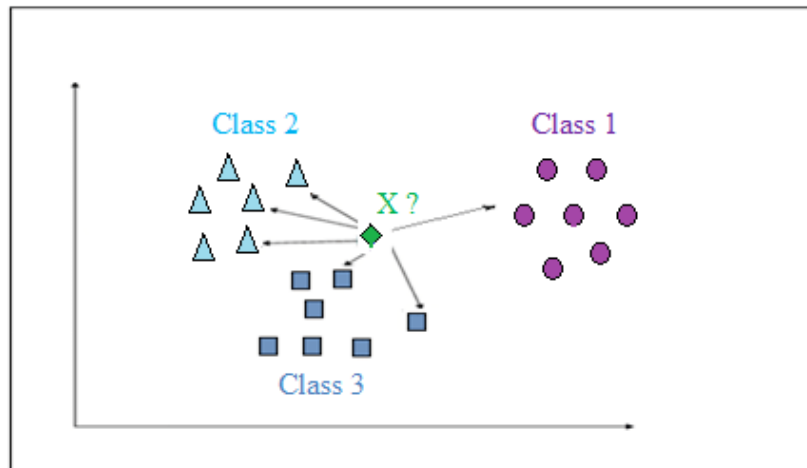
De nombreuses méthodes :

- K-Nearest-Neighbor (KNN)
- Local Outlier Factor (LOF)
- Unweighted Cluster-Based Outlier Factor
- Isolation Forest
- Autoencoder
- ...



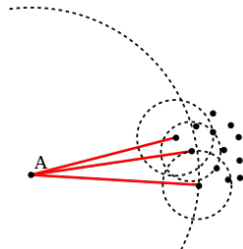
K-Nearest Neighbor

Détection d'Anomalies : KNN



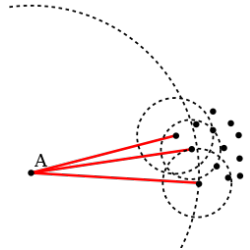
Local Outlier Factor

- anomalies locales



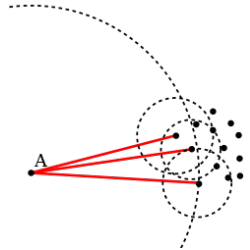
Local Outlier Factor

- anomalies locales
- basé sur les k voisins du point



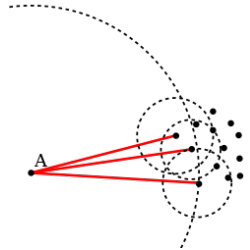
Local Outlier Factor

- anomalies locales
- basé sur les k voisins du point
- définit une « atteignabilité » par les distances de ces voisins



Local Outlier Factor

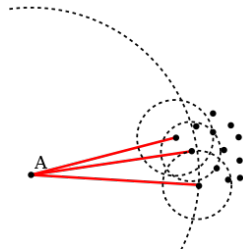
- anomalies locales
- basé sur les k voisins du point
- définit une « atteignabilité » par les distances de ces voisins
- calcule un ratio moyen d'atteignabilité du point et de ses voisins



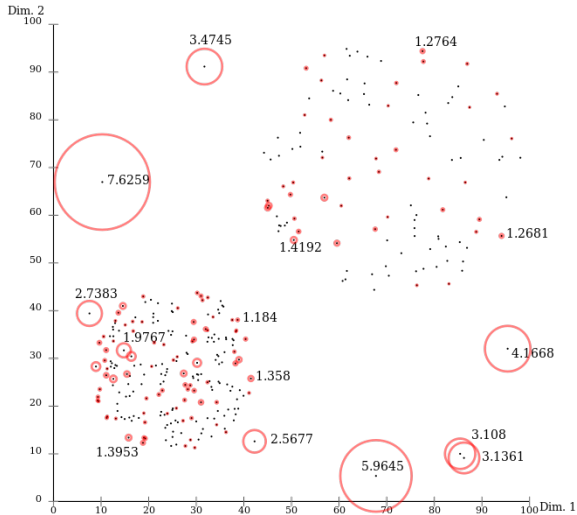
Local Outlier Factor

- anomalies locales
- basé sur les k voisins du point
- définit une « atteignabilité » par les distances de ces voisins
- calcule un ratio moyen d'atteignabilité du point et de ses voisins

→ Anomalie si le ratio moyen d'atteignabilité est beaucoup plus faible que celui de ses plus proches voisins



Local Outlier Factor



- lent (quadratique)
- a des à priori sur la distribution des données

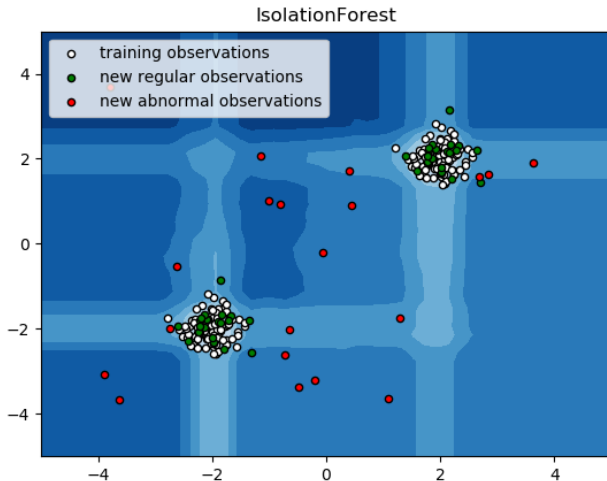
Isolation forest

- arbre aléatoire (comme random forest mais le split est aléatoire, ExtraTree)
- but : isoler une anomalie plus vite qu'un exemple normal
- petit chemin pour arriver à une feuille : anomalie

→ Se sert du fait que les features des anomalies ne sont pas distribuées comme les autres.

- forêt d'isolation trees
- construits sur des sous-échantillons sans remplacement des données
- sous-échantillons plus petits que dans random forest typiquement, pour mieux isoler les anomalies
- converge souvent vite : 100 arbres souvent suffisants

Isolation forest



TP 1

Exploration de données

Support TP 3 : PCA/LDA

www.pca-iris.ipynb

www.pca-lda.ipynb



Support TP 3 : Clustering

www.kmean.ipynb

www.dbscan.ipynb



local-outlier-factor.ipynb
isolation-forest.ipynb

TP-pca.ipynb

TP-clustering.ipynb