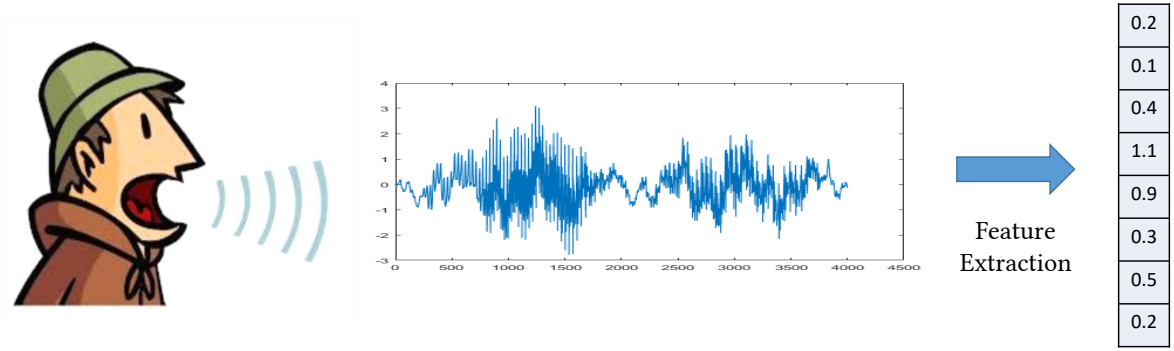
Intelligence Artificielle & Machine Learning pour la modélisation de séries temporelles et de signaux

Séance 6 : Extraction et sélection de caractéristiques

Laurent Oudre
laurent.oudre@ens-paris-saclay.fr

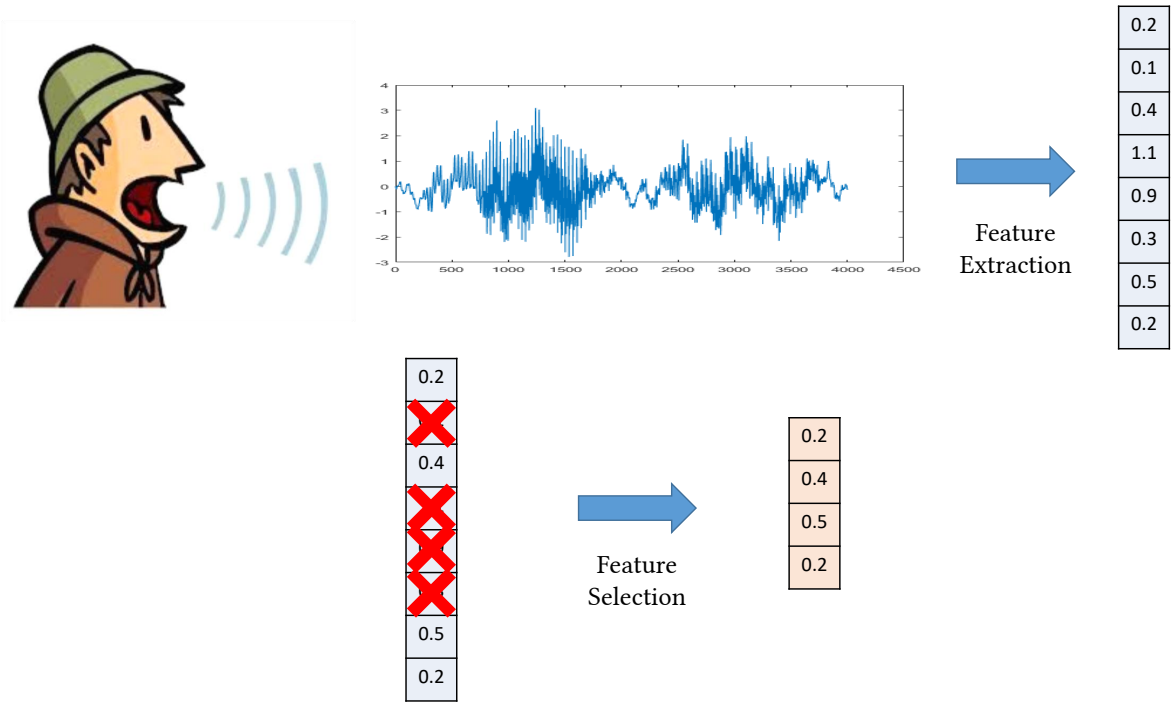
Diplôme ARIA
ENS Paris Saclay
2025-2026

Problème 1 : extraction de caractéristiques



- Etant donnée une série temporelle **x** de longueur N, extraire un ensemble de D caractéristiques (features) **y** qui la caractérisent
- Représentation sous forme de *bag of features*
- Utile pour l'indexation, la classification, le partitionnement...

Problème 2 : sélection de caractéristiques



- Etant donnée une dataset de M séries temporelles $\mathbf{X}$ représentées chacunes par D caractéristiques, sélectionner les $K < D$ caractéristiques qui sont les plus pertinentes pour une tâche donnée
- Tâche qui peut se faire de façon supervisée (avec des annotations) ou non supervisée

Notion de caractéristique

- Une caractéristique (ou feature en anglais) est une valeur réelle qui permet de rendre compte d'une propriété du signal
- Lorsque l'on travaille sur des séries temporelles, la première étape avant de faire de l'apprentissage est de déterminer la liste des features que l'on va extraire du signal et de choisir celles qui sont les plus pertinentes pour la tâche (classification, prédiction, clustering...)
- Ce champ disciplinaire appelé **feature engineering** est un des piliers du data mining et du machine learning

Expertise

- Le choix des features à utiliser peut se baser soit des principes généraux avec des features classiques, soit sur des features parfaitement adaptées à la tâche et donc liées à l'expertise des experts domaine
- Certaines méthodes supervisées peuvent également être utilisées pour choisir les meilleures features (mais attention à l'overfitting !)
- Il n'y a pas de solution miracle et le choix des features sera toujours lié aux données que l'on souhaite traiter et de la tâche à réaliser

Programme de la séance

- Connaître les principales caractéristiques que l'on peut extraire de façon systématique
- Connaître des méthodes simples permettant de sélectionner les features pertinentes dans un contexte supervisé et non supervisé

Session 6 : Feature extraction and selection

Plan du cours

Plan du cours

Caractéristiques classiques

- Sans information préalable sur les données, certaines caractéristiques standard peuvent être utilisées
- Caractéristiques statistiques : moyenne, variance, kurtosis

$$\hat{\mu} = \frac{1}{N}\sum_n x[n] \quad \hat{\sigma^2} = \frac{1}{N}\sum_n (x[n] - \hat{\mu})^2$$

- Caractéristiques liées à l'amplitude : maximum, minimum, Root Mean Square (RMS)

$$\text{RMS} = \frac{1}{N}\sum_n |x[n]|^2$$

- Caractéristique statistiques robustes : médiane, percentiles (5%, 25%, ...)

Caractéristiques fréquentielles

- On peut également caractériser le signal grâce à sa transformée de Fourier discrète : on sélectionnera dans ce cas un ensemble de fréquences qui nous intéressent et on sommera les valeurs $|X[k]|^2$ pour ces fréquences
- L'**énergie relative** dans la bande de fréquence $[f_1, f_2]$ pourra s'exprimer par :

$$E_{[f_1,f_2]} = \frac{\sum_{k, \frac{kF_e}{N} \in [f_1,f_2]} |X[k]|^2}{\sum_{k=0}^{\frac{N}{2}} |X[k]|^2}$$

- Si on échantillonne à 100 Hz, on peut par exemple calculer l'énergie relative dans les bandes 0-10, 10-20, etc...

Pré-traitements et modèles

- La plupart des caractéristiques déjà mentionnées sont très sensibles aux différentes perturbations (bruit, outliers, tendances, ...) présentes dans les signaux
- Il est donc indispensable d'utiliser les méthodes déjà vues (cours 4) **avant** de les calculer
- Les différents modèles déjà vus (y compris ceux basés sur des dictionnaires) peuvent également être utilisés pour construire des features : on utilisera dans ce cas soit les paramètres estimés du modèle (ex : fréquence fondamentale et amplitude relative des harmoniques, paramètres AR...), soit les activations obtenues en projetant tous les signaux dans un dictionnaire commun correctement choisi ou appris sur les données

Plan du cours

Choix des caractéristiques

- L'une des questions fondamentales en feature engineering consiste à savoir si l'ensemble des features construites permet bien de caractériser la base de signaux
- Les features sont elles robustes, bien calculées et pertinentes ?
- Dans le cas d'une classification, sont-elles représentatives des différences qui existent entre les signaux ?
- Mais surtout, ont-elles un pouvoir de généralisation ?
- Deux stratégies : approche non supervisée (on ne connaît pas les étiquettes) ou approche supervisée (étiquettes connues sur une base d'apprentissage)

Notations

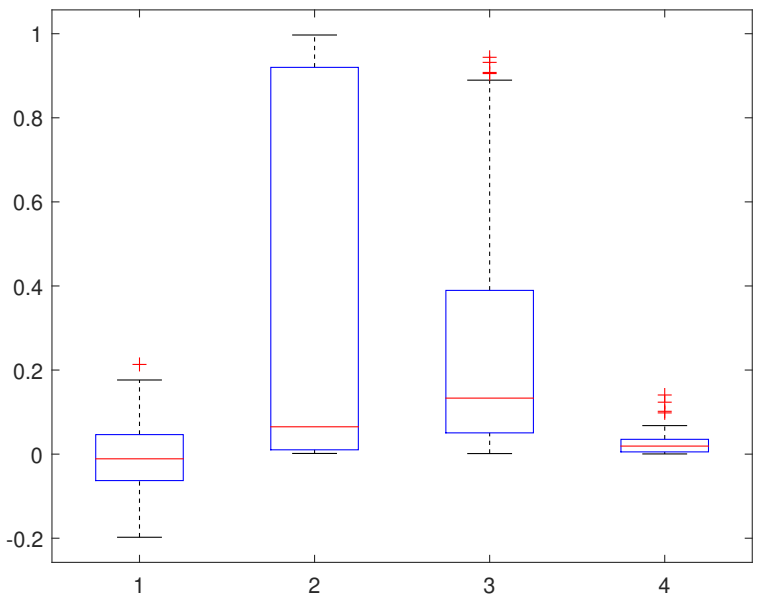
On suppose que le traitement des séries temporelles a été effectué au préalable : chaque signal (ou observation) est représenté par un vecteur de dimension d

- M : nombre d'observations
- D : nombre de features
- $\mathbf{x_1}, \ldots, \mathbf{x_M}$: observations
- Matrice de données $\mathbf{X} \in \mathbb{R}^{D \times M}$
- Annotations (éventuelles) : $\mathbf{y} \in \{-1, +1\}^M$

Cas non supervisé

- Dans le cas où aucune annotation n'est disponible, l'évaluation est empirique et se base sur l'observation de plusieurs critères
- Fiabilité des features : présence de valeurs aberrantes ?
- Utilité des features : permettent-elles de voir des différences entres les observations ?
- Visualisation et clustering : peut-on distinguer des groupes ?

Fiabilité des features

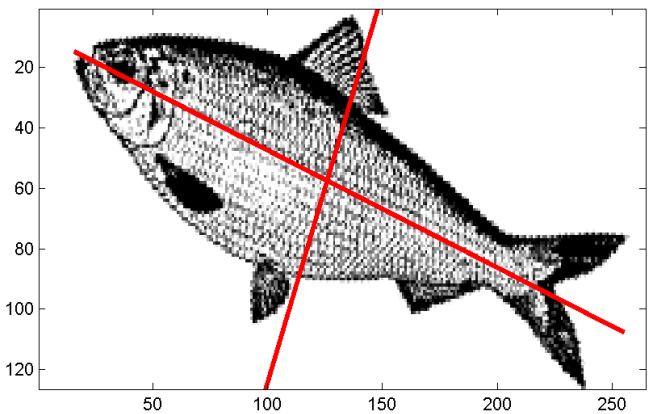


- Tests statistiques simples
  - Repérage des valeurs aberrantes par comparaison avec l'écart-type sur l'ensemble des observations

$$X_{d,m} > 3 \times \text{std}\left(X_{:,m}\right)$$

  - Calcul des percentiles (5% et 95%) et comparaison aux minimum/maximum
  - Simple test de variance pour déterminer les features qui ne varient quasiment pas sur la base de données
- L'idée est surtout de déterminer si les valeurs extrêmes sont liées aux caractéristiques intrinsèques des signaux ou si elles sont dues à des erreurs de pré-traitement (ex : bruit impulsionnel dans les données, non stationnarités, mauvaise estimation des paramètres...)

Utilisation de l'ACP



Analyse en Composantes Principales (ACP)

- Principe de base : détecter les features qui contribuent le plus à la variance
- Transformation des D variables initiales, potentiellement corrélées, à D nouvelles variables appelées *composantes principales* décorrélées
- Les features f_d sont transformées en de nouvelles features selon la relation

$$\tilde{f}_j = \sum_{d=1}^{D} U_{d,j} f_d$$

- Les $\tilde{f}_j$ sont classées selon leur contribution à la variance globale des données : il est courant d'observer les données projetées selon les deux premieres composantes principales (dans le plan)

Utilisation de l'ACP

1. Renomalisation des données pour que chaque ligne (donc chaque feature) ait une moyenne nulle et une variance égale à 1

$$\tilde{X}_{d,:} = \frac{X_{d,:} - \mu_{X_{d,:}}}{\sigma_{X_{d,:}}}$$

2. Décomposition en valeurs singulières (SVD) de la matrice $\tilde{\mathbf{X}}$:

$$\tilde{\mathbf{X}} = \underbrace{\mathbf{U}}_{D\times D}\underbrace{\mathbf{S}}_{D\times M}\underbrace{\mathbf{V}^t}_{M\times M}$$

3. Tracé des contributions de chaque feature aux deux premières composantes principales sur le cercle des corrélations

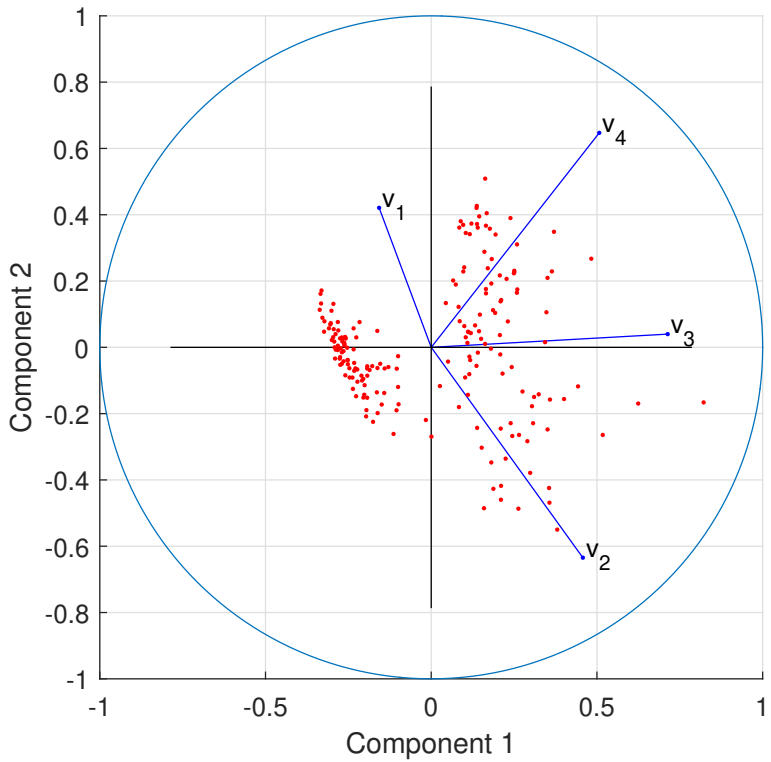
$$U_{d,j} : \text{ contribution de la feature } d \text{ à la } j^{eme} \text{ composante principale}$$

4. *(facultatif)* Tracé des données sur les deux premières composantes principales. En notant $\tilde{\mathbf{S}} = \mathbf{S}\mathbf{V}^t$

$$\tilde{S}_{j,m} : \text{ projection de l'observation } m \text{ sur la } j^{eme} \text{ composante principale}$$

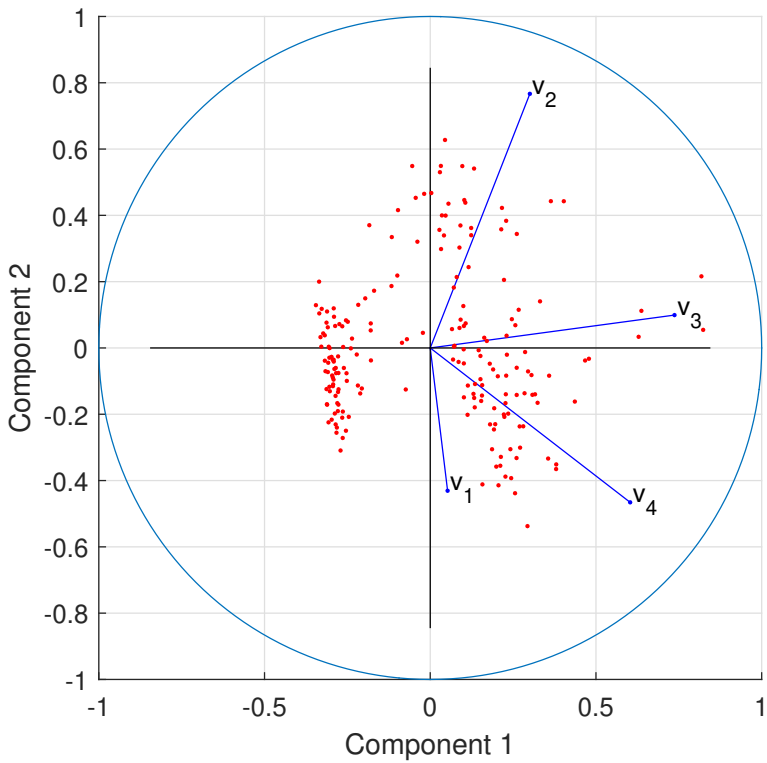
$$\text{var}\left[\tilde{S}_{j,:}\right] : \text{ contribution de la } j^{eme} \text{ composante principale à la variance globale}$$

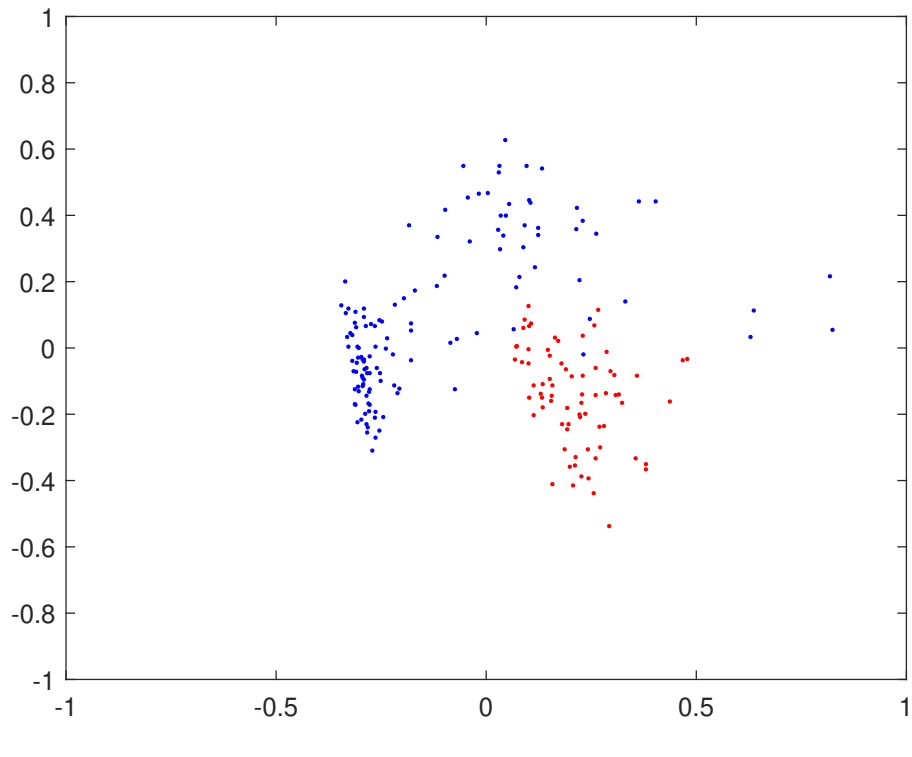
Visualisation des features



- Visualisation de chaque feature : une **bonne** feature est une feature proche du cercle des corrélations
- Visualisation des données dans le plan défini par les deux premières composantes principales

Utilisation du clustering





- Si l'on sait qu'il existe plusieurs classes dans le jeu de données, on peut utiliser un algorithme de clustering (K-means par exemple) pour visualiser les clusters obtenus
- La distance des points aux centroids peut être utilisée pour rendre compte de la pertinence de la représentation

Cas supervisé

- Si l'on dispose d'une base d'apprentissage avec des annotations $\mathbf{y} \in \{-1, +1\}^M$, on peut les utiliser pour déterminer les meilleures features à utiliser
- On distingue en général trois façons de procéder :
  - **Filter methods :** On teste l'adéquation des features avec les annotations, grâce à différents critères (par exemple la corrélation). Ceci permet de sélectionner les features les plus pertinentes.
  - **Wrapper methods :** On teste directement les features sur une tâche de classification supervisée, en essayant plusieurs configurations. On garde les features donnant les meilleurs scores de classification.
  - **Embedded methods :** Approches mixtes où on va conjointement inférer l'importance des features et classer les données (arbres de décision, méthodes parcimonieuses...)

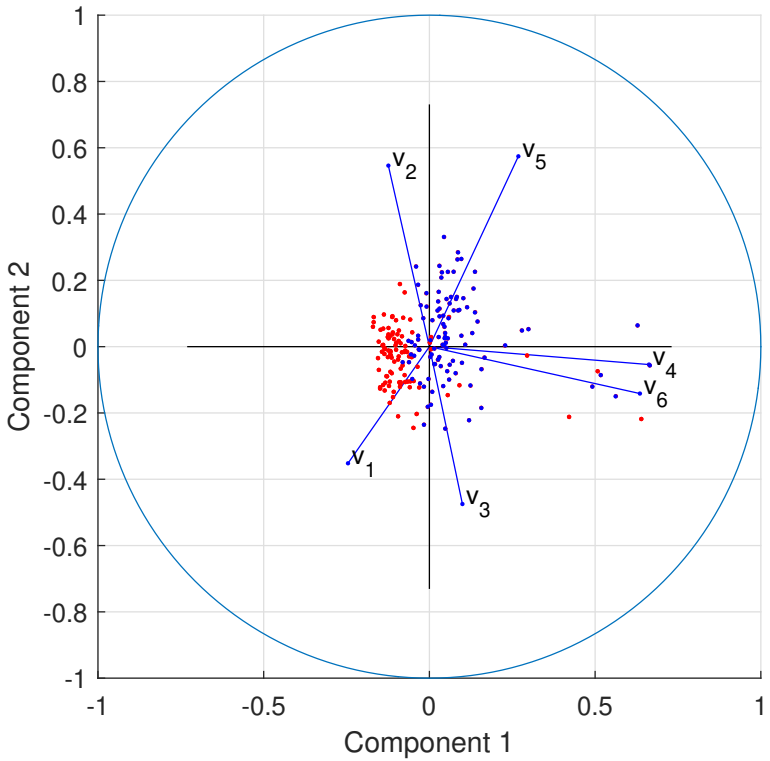
Filter methods

- L'idée consiste à tester (souvent individuellement) la pertinence de chaque feature en étudiant les liens qui existent avec les annotations
- Un exemple de test simple consiste à calculer le coefficient de corrélation de Pearson entre le vecteur de données $X_{i,:}$ correspondant à la feature i et le vecteur d'annotations $\mathbf{y}$

$$\rho(i) = \frac{\text{cov}\left(X_{i,:}, \mathbf{y}\right)}{\sqrt{\text{var}\left(X_{i,:}\right)\text{var}\left(\mathbf{y}\right)}}$$

- Ce coefficient, compris entre -1 et +1, permet de voir la corrélation linéaire qui existe entre les observations et les annotations
- On gardera les features qui ont les coefficients de corrélation les plus élevés (en valeur absolue)

Exemple



- Deux classes de signaux et $d = 6$ features
- Selon l'ACP (non supervisé), les features v_2, v_4, v_5, v_6 semblent pertinentes pour expliquer la variance des données
- En calculant les corrélations avec les annotations (supervisé) on obtient

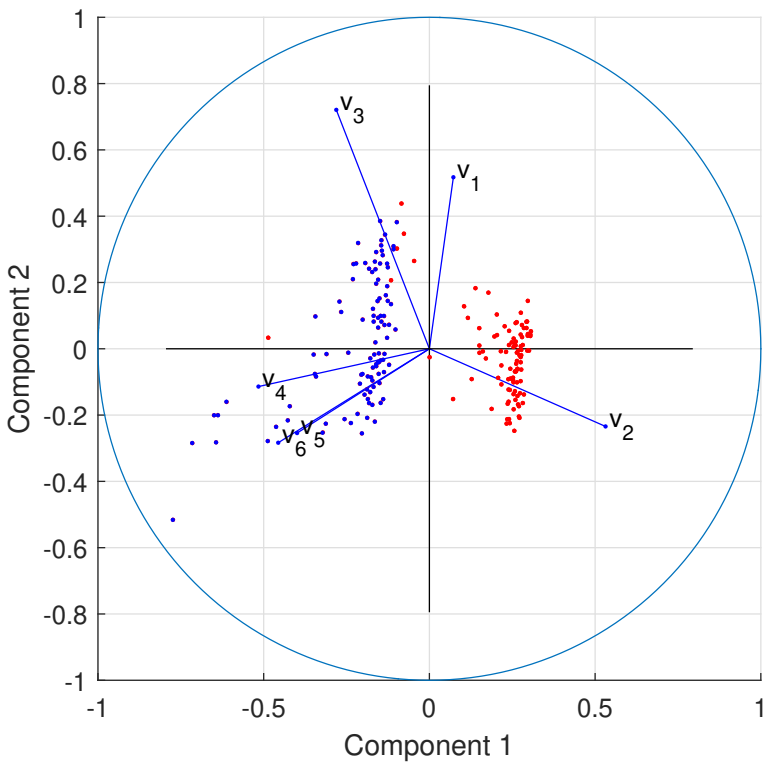
$$-0.02 \quad 0.08 \quad 0.32 \quad 0.45 \quad 0.82 \quad 0.23$$

Les features v_4, v_5 sont ici les plus pertinentes pour la tâche de classification

Wrapper methods

- Utilisation d'un classifieur sur étagère (k-NN plus proches voisins, SVM...), test des différentes variables et calcul de performances par validation croisée
- Sélection de la/des features donnant les meilleures performances sur la base d'apprentissage
- Le nombre de configurations à tester peut se révéler très élevé si d est grand : il existe dans ce cas plusieurs stratégies
  - Ajout itératif de features : on prend la meilleure feature, puis on teste des groupes de deux features la comprenant, etc...
  - Retrait itératif de features : on essaie d'enlever les features une à une en observant les conséquences sur les performances
  - Approche stochastique : on teste des groupes aléatoirement...

Exemple



- Deux classes de signaux et $d = 6$ features.
- Test avec un 1-NN (distance euclidienne)
- Cas d'une variable (taux de classification Leave-One-Out)

$$0.49 \quad 0.97 \quad 0.64 \quad 0.88 \quad 0.94 \quad 0.7850$$

- Cas de deux variables (test de toutes les combinaisons)

$$v_2 + v_5 : 1.0$$

$$v_2 + v_3 : 0.99$$

$$v_3 + v_5 : 0.98$$

$$\ldots$$

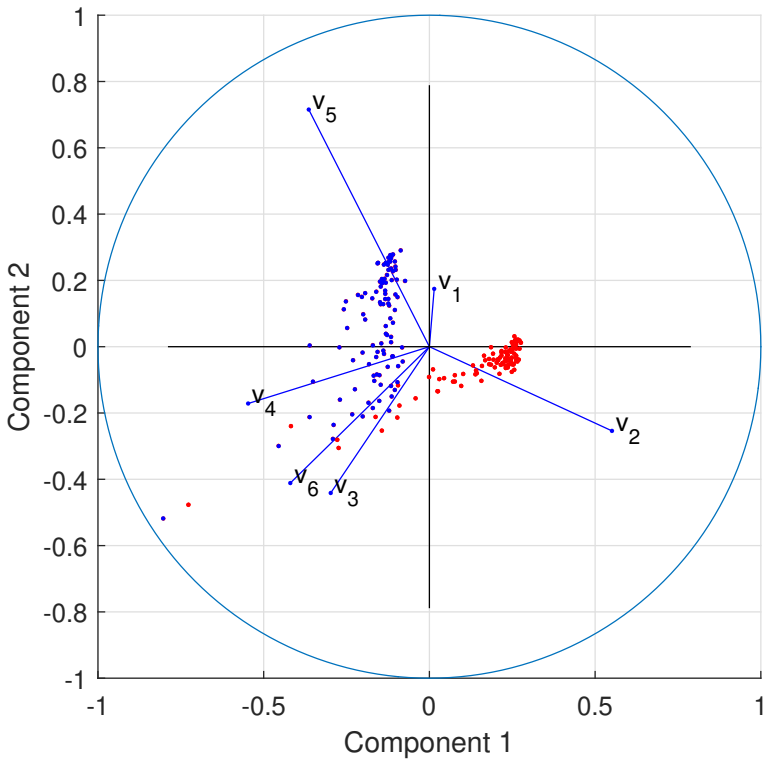
Embedded methods

- Utilisation de méthodes de classification interprétables, qui permettent a posteriori de savoir quelles features ont été utilisées : arbres de décision...
- Une des ces méthodes a déjà été vue dans la séance 2 : méthode du LASSO

$$\mathbf{z}^* = \underset{\mathbf{z}}{\operatorname{argmin}} \left\| \mathbf{y} - \mathbf{X}^T \mathbf{z} \right\|_2^2 + \lambda \left\| \mathbf{z} \right\|_1$$

- Régression parcimonieuse pour retrouver les annotations $\mathbf{y}$ à partir des données $\mathbf{X}$
- Le vecteur $\mathbf{z}$ rend compte de l'importance de chaque feature dans la régression : grâce à la contrainte de parcimonie, beaucoup de coefficients seront à 0, ce qui constitue implicitement une sélection de caractéristiques
- Le score de classification peut s'estimer en regardant le signe de la reconstruction $\mathbf{X}^T \mathbf{z}^*$ et en le comparant au vecteur $\mathbf{y}$

Exemple



- Deux classes de signaux et $d = 6$ features.
- Algorithme du LASSO avec plusieurs valeurs de λ
- 5 variables (1, 2, 4, 5, 6) : 0.97
- 4 variables (2, 4, 5, 6) : 0.96
- 3 variables (2, 5, 6) : 0.95

Références

- Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2019). A review of unsupervised feature selection methods. Artificial Intelligence Review, 1-42.
- J. Li et al. Feature selection : A data perspective. ACM Computing Surveys (CSUR) 50.6 (2018) : 94. https://arxiv.org/pdf/1601.07996.pdf
- T. Kanungo et al. An efficient k-means clustering algorithm : Analysis and implementation. IEEE Transactions on Pattern Analysis & Machine Intelligence 7 (2002) : 881-892. ftp://ftp.umiacs.umd.edu/pub/wexler/kmeans.pdf
- https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114
- https://developers.google.com/machine-learning/crash-course/representation/feature-engineering