

Apprentissage Statistique

Juste Raimbault¹

¹LaSTIG, IGN-ENSG-UGE



Apprentissage d'une fonction f_θ telle que

$$\min_{\theta \in \Omega} \sum_{i=1}^N l(f_\theta(x_i), y_i)$$

avec θ paramètres, x_i données d'entrée, y_i données de sortie (vérité terrain), l fonction de perte (cross-entropy pour la classification, MSE pour la régression, Lasso, etc.)

- Classification : prédire l'appartenance à des classes discrètes
- Régression : prédire une valeur numérique vectorielle
- Clustering : identifier des classes endogènes
- Matching : faire des associations entre éléments de deux ensembles
- Génération : générer des données synthétiques avec certaines caractéristiques statistiques
- Denoising / données incomplètes
- ...

- Non-supervisé : pas de label
- Supervisé : label pour chaque observation
- Faiblement supervisé : labels indirects
- Few-shot : peu de labels (modèles pré-entraînés)
- Self-supervisé : génération endogène des labels

Métriques : accuracy, précision, recall, F-score, . . .

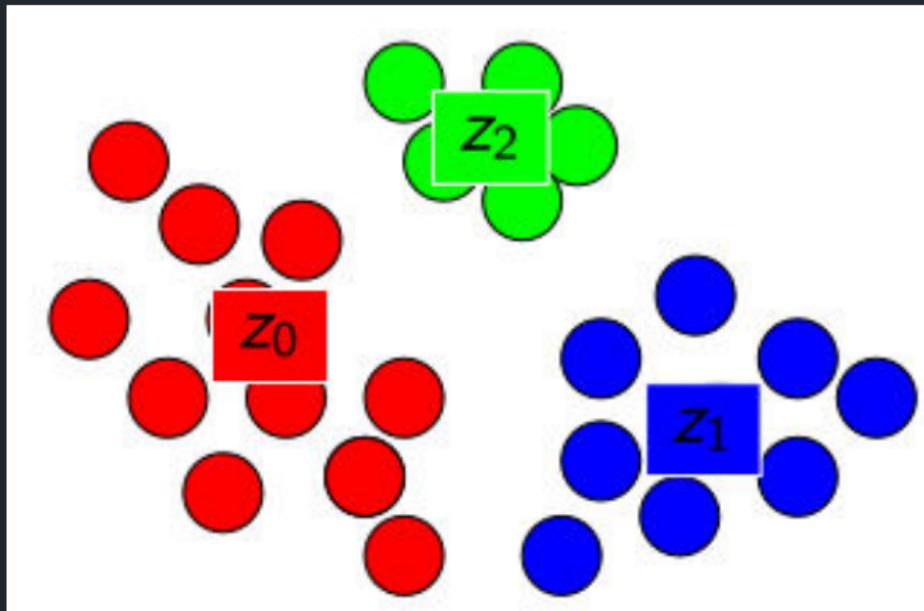
Entraînement :

- Séparation des données en train/test (80/20%)
- Cross-validation pour éviter le sur-apprentissage
- Optimisation des hyper-paramètres

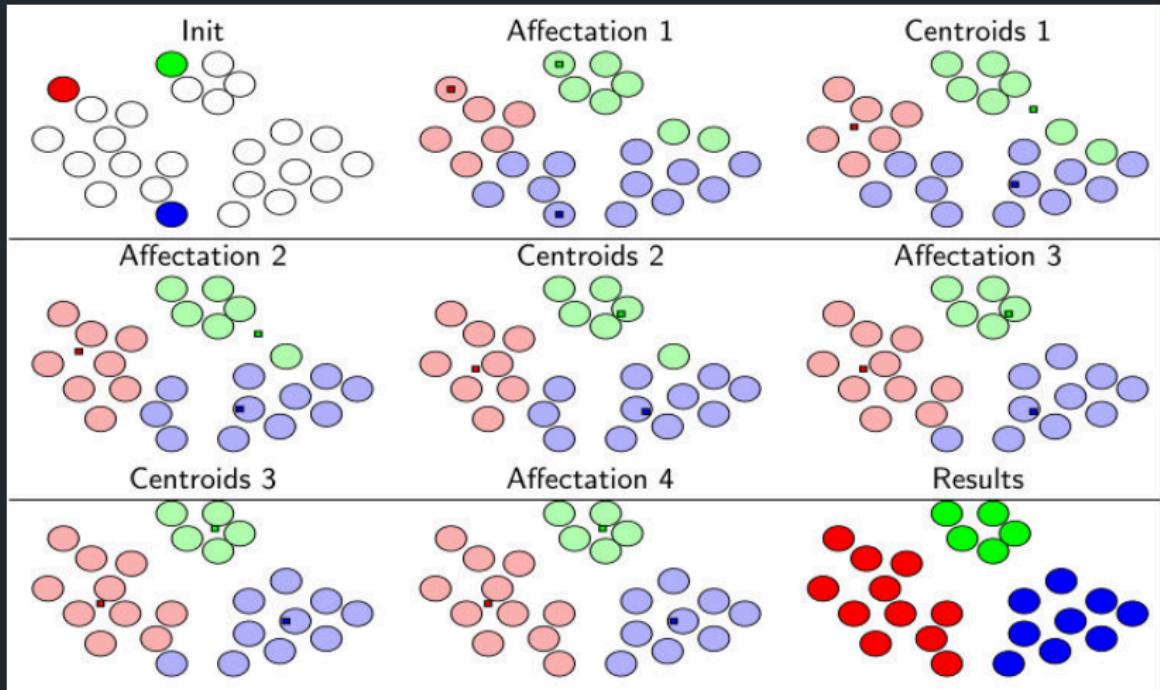
- k-NN (classification)
- Arbres de décisions, Forêts aléatoires (classification)
- Support Vector Machine linéaire et Kernel (classification)
- Régression logistique (classification)
- Régression linéaire (régression)
- k-means (classification non-supervisée)
- DB-SCAN (classification non-supervisée)

Classification non-supervisée : k-means

Minimise $\sum_c \sum_i \|x_i - z_c\|^2$ de manière itérative



k-means



Classification d'espèces d'arbres en utilisant différents algorithmes :
SVM, k-NN, Random Forest

Notebook python sur Google Colab :

[https://colab.research.google.com/drive/
1n7atelJAPWZVmfmFEdksnVLcHH9c6vdV](https://colab.research.google.com/drive/1n7atelJAPWZVmfmFEdksnVLcHH9c6vdV)