

Machine Learning

Supervised
Unsupervised
Reinforcement

Machine Learning (Apprentissage automatique)

Hatem KALLAL

2018-2019

3 Opérations courantes

Régression

- Traitement des valeurs numériques : prédiction

Classification

- Affectation des individus dans des classes

Clustering

- Création de groupes dans une population

Les principales variantes

- **Supervised** : Entraînement sur des individus connus (Les données sont étiquetées).
- **Unsupervised**: Entraînement sur des exemples non connus.
- **Semi-supervised**: Individus étiquetés/non étiquetés).
- **Reinforcement** : observations/récompenses. (Interaction entre l'environnement et l'algorithme)

Principe

A partir des données d'entraînement $E = (x_i ; y_i)$, on cherche une loi de dépendance entre x et y .

Par exemple,

une fonction h aussi proche de f (fonction cible) que possible tel que $y_i = f(x_i)$

une distribution de probabilité $P(x_i ; y_i)$

Règles :

f est continue : régression.

f est discrète : classification.

f est binaire : apprentissage de concept.

Les principaux algorithmes :

- La régression linéaire ou polynomiale
- Les k plus proches voisins
- Les arbres de décisions
- Le perceptron
- Les réseaux de neurones
- Classification naïve bayésienne

KNN : The K Nearest Neighbors (KPPV)

L'un des plus réputés

Utilisé en classification et en régression

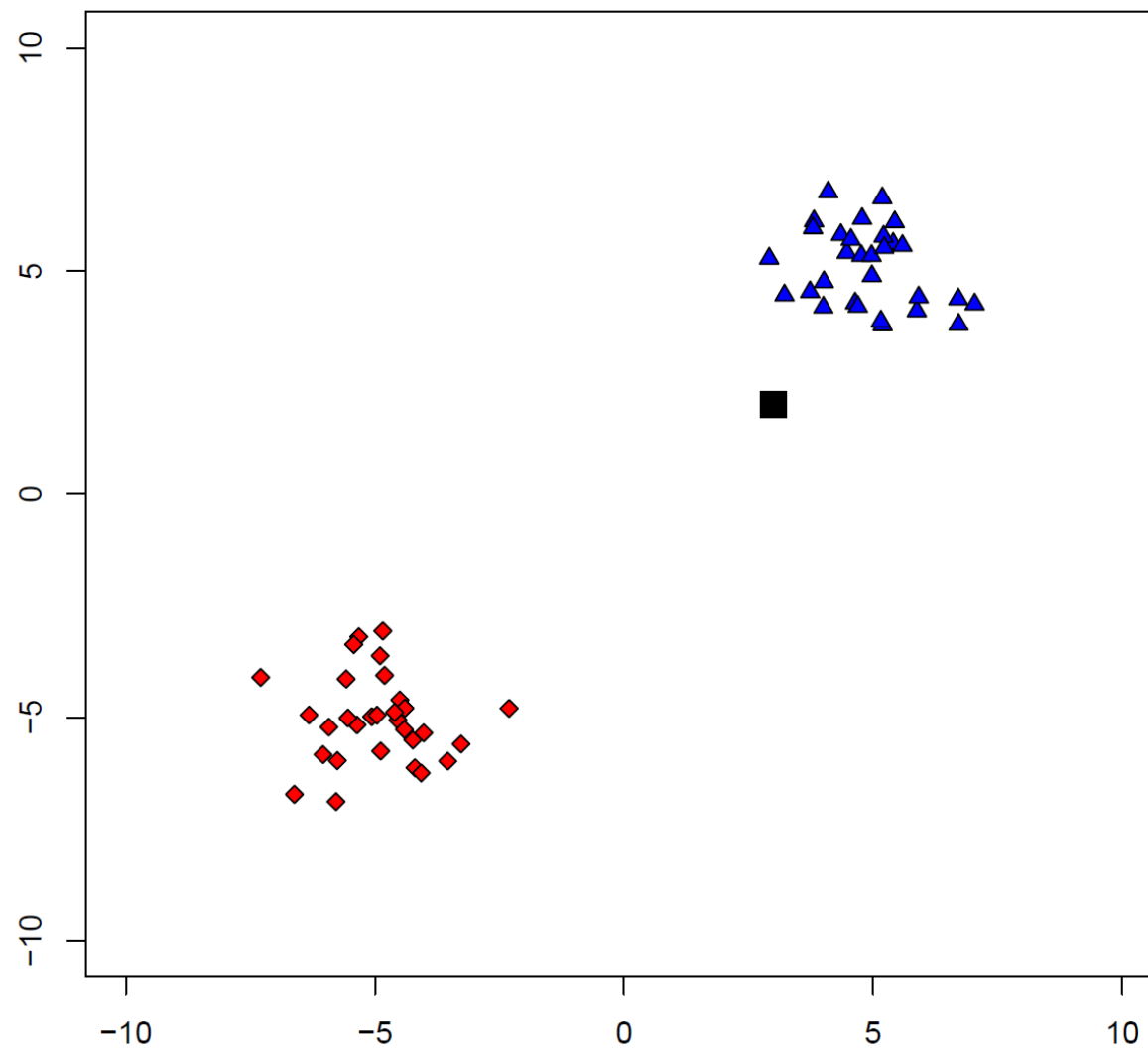
Paramétrage facile

Machine Learning

Supervised

- KNN

Unsupervised Reinforcement



Machine Learning

Supervised

- KNN

Unsupervised Reinforcement

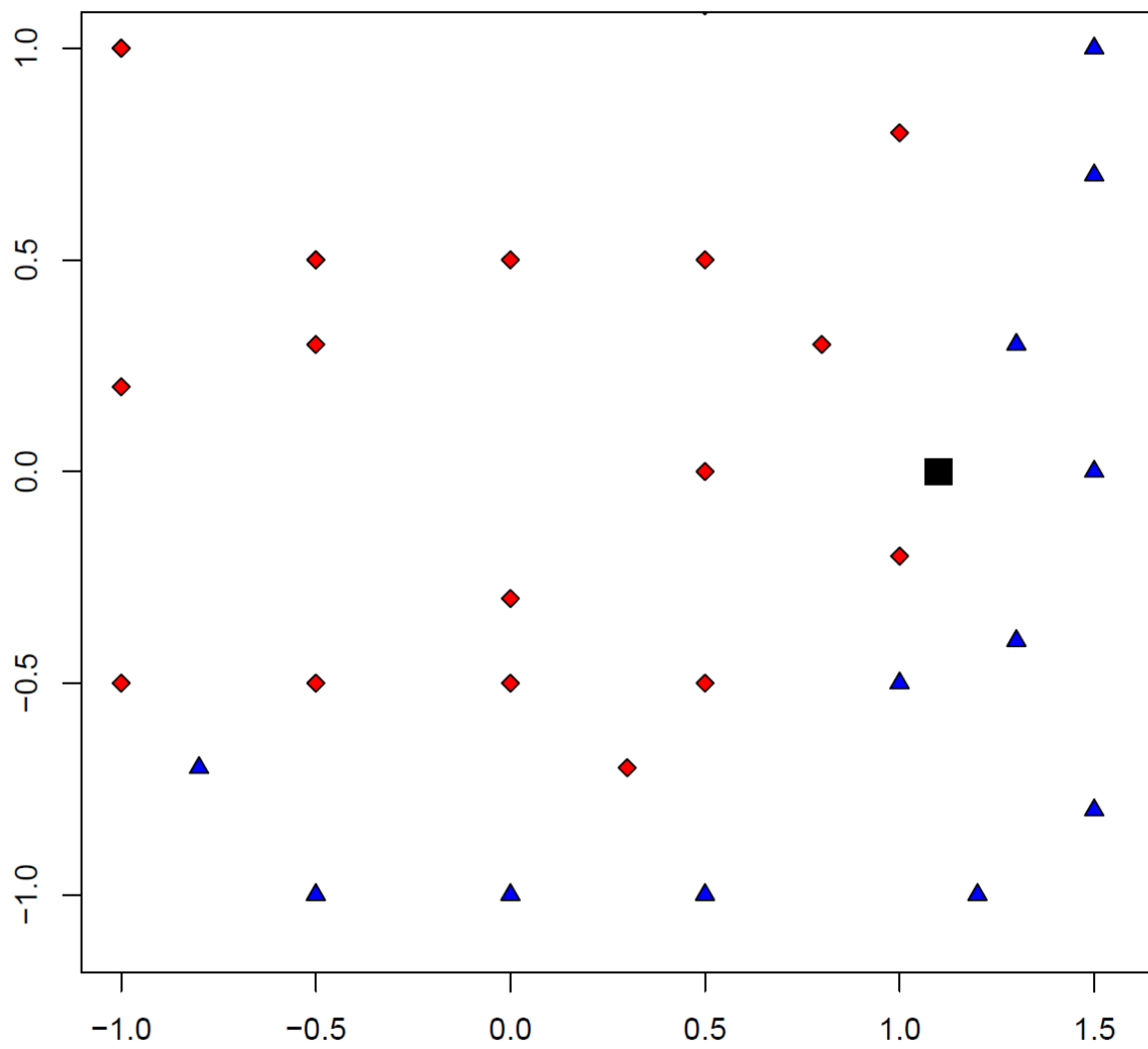
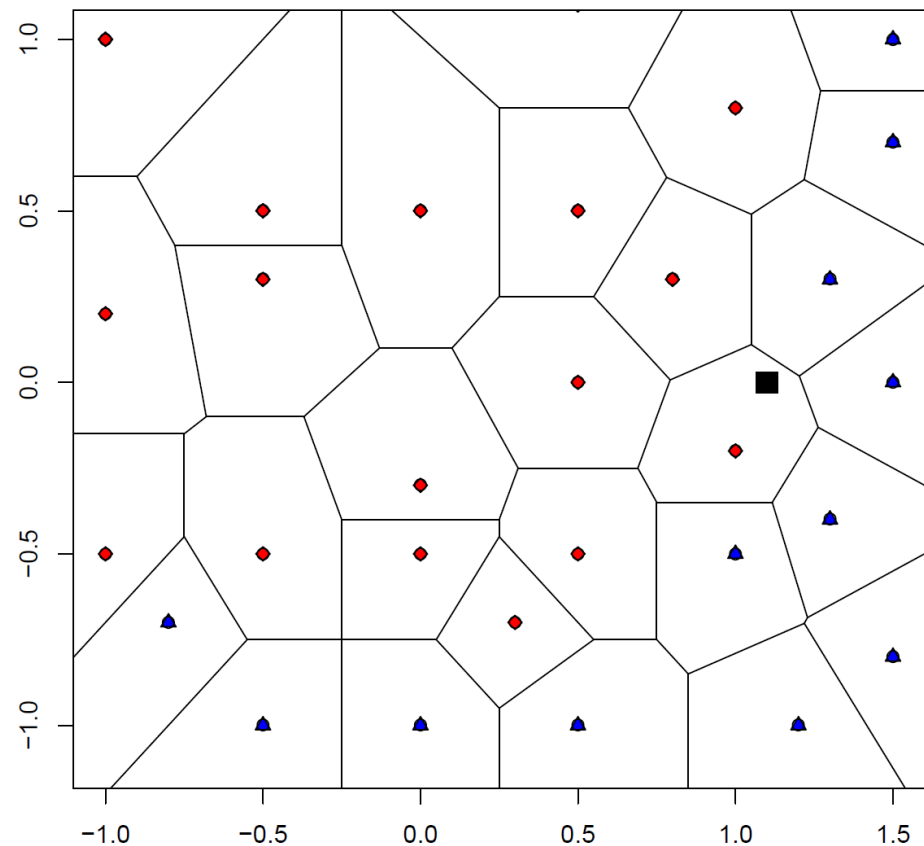


Diagramme de Voronoi



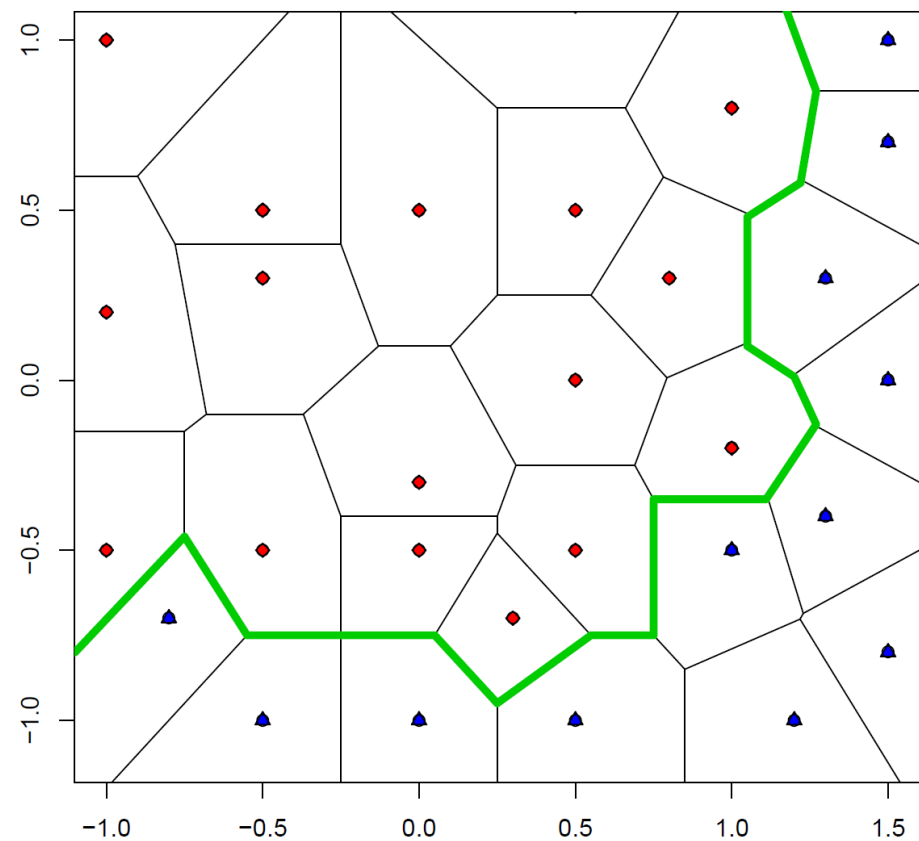
Machine Learning

Supervised

- KNN

Unsupervised Reinforcement

Diagramme de Voronoi



L'algo

Charger les données

Initialiser la valeur de k

Itérez de 1 au nombre total de points de données d'entraînement

- Calculez la distance entre les données de test et chaque ligne de données d'entraînement.
- Triez les distances calculées par ordre croissant en fonction des valeurs de distance.
- Obtenir les k premières lignes du tableau trié
- Obtenir la classe la plus fréquente de ces lignes

Retourner la classe prédite

Dans scikit-learn

```
from sklearn import neighbors,
```

```
neighbors.KNeighborsClassifier(n_neighbors=5,  
                               weights='uniform',  
                               algorithm='auto',  
                               leaf_size=30,  
                               p=2, metric='minkowski',  
                               metric_params=None,  
                               n_jobs=None)
```

Dans scikit-learn

euclidean"	EuclideanDistance		$\text{sqrt}(\text{sum}((x - y)^2))$
"manhattan"	ManhattanDistance		$\text{sum}(x - y)$
"chebyshev"	ChebyshevDistance		$\text{max}(x - y)$
"minkowski"	MinkowskiDistance	p	$\text{sum}(x - y ^p)^{(1/p)}$
"wminkowski"	WMinkowskiDistance	p, w	$\text{sum}(w * (x - y) ^p)^{(1/p)}$
"seuclidean"	SEuclideanDistance	V	$\text{sqrt}(\text{sum}((x - y)^2 / V))$
"mahalanobis"	MahalanobisDistance	V or VI	$\text{sqrt}((x - y)' V^{-1} (x - y))$

Machine Learning

Supervised

- KNN

Unsupervised Reinforcement

Lecture saine

<https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/>

KNN en régression

```
from sklearn.neighbors import KNeighborsRegressor  
from sklearn.metrics import mean_squared_error
```

```
knn = KNeighborsRegressor()
```

```
knn.fit(X_train , Y_train)
```

```
prediction = knn.predict(X_test)
```

```
mse = mean_squared_error(prediction , Y_test)
```

Attention : uniquement pour les « features » numériques et de préférence standardisées « normalized »

Machine Learning

Supervised

- KNN
- Decision Tree

Unsupervised

Reinforcement

Decision Tree