K-Nearest Neighbor

Komlan Jean-Marie DANTODJI

Etudiant en M1 Big Data

Université Paris 8

27 novembre 2020

Plan

Structure globale des algorithmes du machine learning 3/12

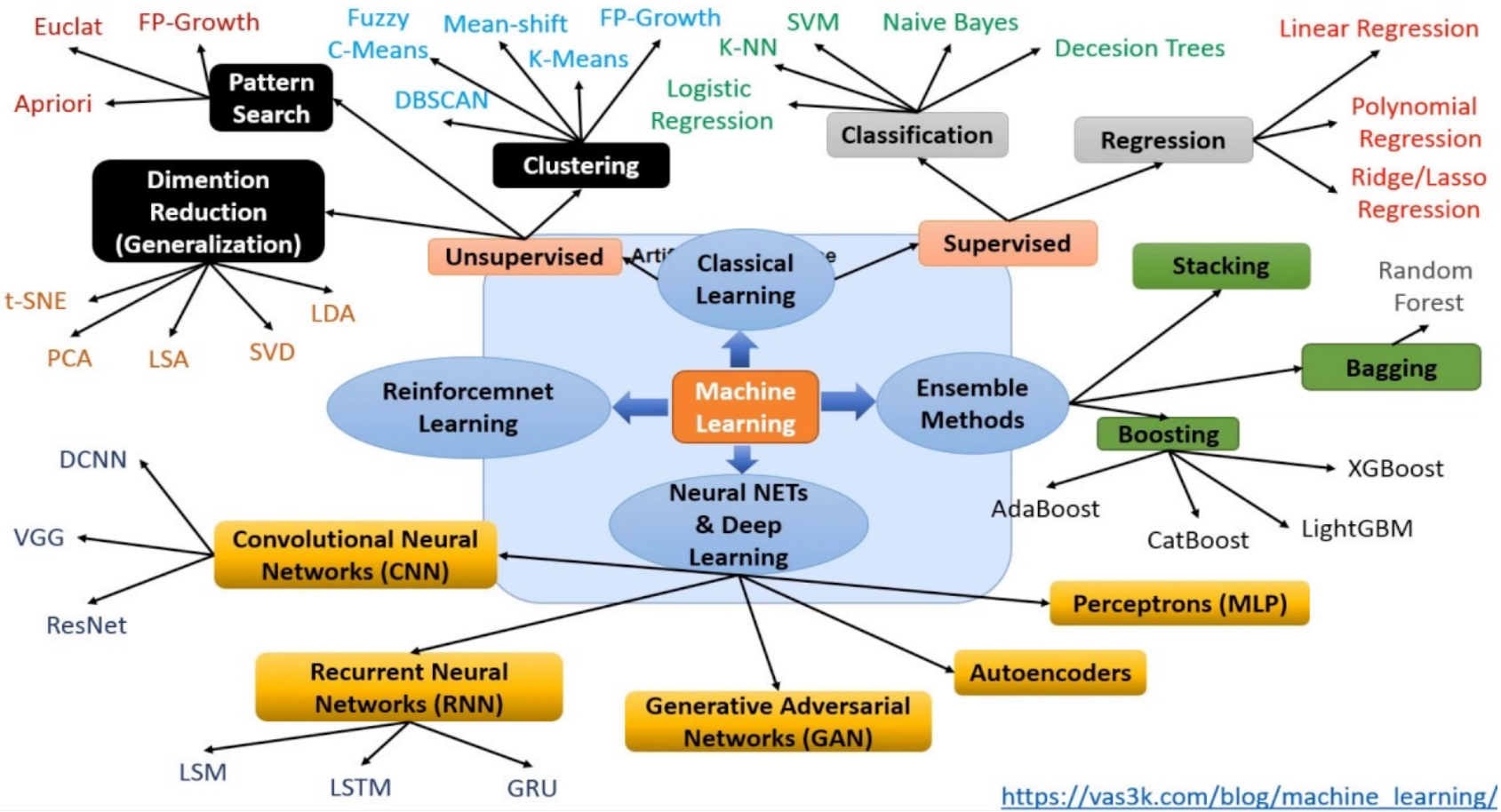

FIGURE – Structure du machine learning

Préparation des données

- Base de donnée
- Traitement préalables nécessaires

Fonctionnement du K-NN

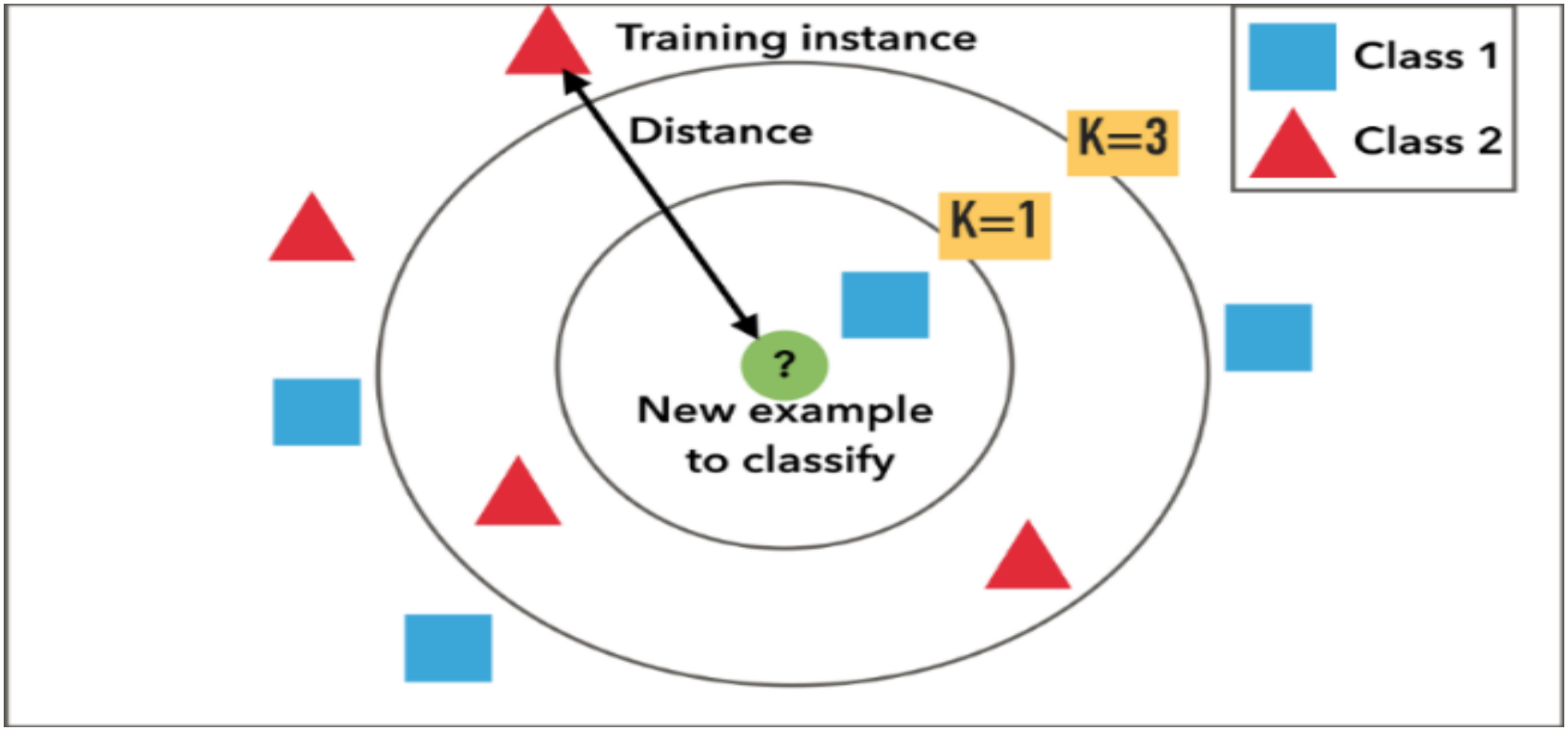

FIGURE – K-Nearest Neighbor

Les types de distances

- Distance euclidienne

$$d(A,X) = \sqrt{\sum_{i=1}^{n}(a_i - x_i)^2}$$

- Distance de Manhattan

$$d(A,X) = \sum_{i=1}^{n} \mid a_i - x_i \mid$$

- Distance de Minkowski

$$d(A,X) = \sqrt[p]{\sum_{i=1}^{n} \mid a_i - x_i \mid^p}$$

Choix du paramètre K

- Utilisation de K

$$K = \sqrt{nombre - de - donnees}$$

- Choisir K suivant celui qui donne une meilleure prédiction

KNN implémenté en Python


```
159 class KNN:
160     def __init__(self, nb_neighbours = 5):
161         self.nb_neighbours = nb_neighbours
162 
163     def fit(self, X, y):
164         self.X = X.astype(np.int64)
165         self.y = y
166 
167     def predict(self, datas_input):
168         results = []
169         for point in datas_input:
170             results.append(self.predict_point(point))
171         return np.array(results)
172 
173     def predict_point(self, point):
174         #Calculate distance from point to each element in the dadabase
175         list_dist = []
176         for x_point, y_label in zip(self.X, self.y):
177             dist = ((point - x_point) ** 2).sum()
178             list_dist.append([dist, y_label])
179         #Sort datas based on thier distances to point
180         sorted_distance = sorted(list_dist, key=lambda x: x[0]) #to sort according to the first column
181         #Get the K nearest in distances
182         k_nearest = sorted_distance[:self.nb_neighbours] # [[10,1],[15,0],[17,1],[23,0],[25,0]]
183         # Get list of the labels in the top K elements ==> [0,1] and the counter [3,2]
184         labels, counts = np.unique(np.array(k_nearest)[:,1], return_counts=True)
185         #Get the label of largest one
186         #print(k_nearest)
187         return labels[np.argmax(counts)]
188 
189     def score(self, X, y):
190         return sum(self.predict(X) == y) / len(y)
```

FIGURE – KNN en python

Domaines d'application

- Ecriture manuscrite
- En médécine, prédire des maladies
- Classification des images

Avantages et désavantage du K-NN

1. Avantages du KNN
   - Facile à comprendre
   - Apprentissage rapide
2. Désavantage du KNN
   - Limité à partir d'une large donnée d'apprentissage
   - La valeur de K optimal non connu au préalable

Références

[1] Audibert, J.Y. , Tsybakov, A.B. (2007) Fast learning rates for plug-in classifiers under the margin condition", Ann. Statist, 35 : 608 633.
[2] Bailey, T. Jain, A. (1978) A note on distance-weighted k-Nearest Neighbor rules, IEEE Tra, Systems, Man, Cybernetics, 8 : 311-313

Merci pour votre attention