
Algorithm 2: Algorithme des K -plus proches voisins

Input: Données d'apprentissage ; $\mathbf{X}^{\text{train}} = (\mathbf{x}_1^{\text{train}}, \dots, \mathbf{x}_n^{\text{train}})$; classes des données d'apprentissage $\mathbf{z}^{\text{train}} = (z_1^{\text{train}}, \dots, z_n^{\text{train}})$; $\mathbf{X}^{\text{test}} = (\mathbf{x}_1^{\text{test}}, \dots, \mathbf{x}_m^{\text{test}})$; nombre des ppv K

Algorithme Knn :

```
for  $i \leftarrow 1$  to  $m$  do
  for  $j \leftarrow 1$  to  $n$  do
    Calculer la distance euclidienne  $d_{ij}$  entre  $\mathbf{x}_i^{\text{test}}$  et  $\mathbf{x}_j^{\text{train}}$  en utilisant l'équation (1)
     $d_j \leftarrow d_{ij}$ 
  end
  Calculer la classe  $z_i^{\text{test}}$  du  $i$ ème exemple qui vaut la classe de son ppv :

  /* trouver les  $K$ -ppv de  $\mathbf{x}_i^{\text{test}}$  */ :

  Trier les distances  $d_j$  selon un ordre croissant pour  $j = 1, \dots, n$ 
  Récupérer en même temps les indices  $IndVoisins$  avant le tri des  $d_j$ 
  Récupérer les classes des  $K$  premiers ppv à partir des indices  $IndVoisins$  et en
  trouver la classe majoritaire :

   $C_k \leftarrow 0$  ( $k = 1, \dots, K$ )
  for  $k \leftarrow 1$  to  $K$  do
     $ind\_voisin_k \leftarrow IndVoisins_k$ 
     $h \leftarrow z_{ind\_voisin_k}^{\text{train}}$ 
     $C_h = z_h + 1$ 
  end

  /* trouver la classe du ppv de  $\mathbf{x}_i^{\text{test}}$  :
  (la classe majoritaire de celles de ses  $K$ -ppv) */ :

   $z_i^{\text{test}} = \arg \max_{k=1}^K C_k$ 
end
```

Result: classes des données de test $\mathbf{z}^{\text{test}} = (z_1^{\text{test}}, \dots, z_n^{\text{test}})$