

## Chapitre 2 Apprentissage à base d'instances

---

**Résumé.** Dans ce chapitre, nous posons le cadre formel de l'apprentissage supervisé et de sa validation. Nous nous intéressons plus particulièrement à l'apprentissage à base d'instances permettant de définir un contexte à un individu plongé dans l'espace des descripteurs. Après avoir décrit la méthode des  $k$  plus proches voisins, nous montrons des anomalies nous faisant préférer l'utilisation des graphes de voisinage. Ces derniers génèrent une structure définissant clairement les relations de voisinage entre individus.

Enfin, nous décrivons les méthodes de validation usuelles ainsi que leurs mises en œuvre. Dépendamment des performances à évaluer, il existe plusieurs méthodes. Nous présentons ici celles principalement utilisées par la communauté de l'apprentissage automatique et celles utilisées par la communauté de la recherche d'information.

**Mots clefs :** Apprentissage supervisé,  $k$  plus proches voisins, graphe de voisinage, validation croisée, rappel, précision.

---

### 2.1 Introduction

Nous avons vu au Chapitre 1 qu’une population  $\Omega$  d’individus ou d’objets concernés par le problème d’apprentissage était décrite par un ensemble de descripteurs  $\Gamma = \{X^1, \dots, X^p\}$ , appelé encore ensemble de caractéristiques ou variables exogènes. Cet ensemble définit un espace de représentation noté  $\mathfrak{R}$  sans structure mathématique particulière, voir équation (1.2.1). En apprentissage supervisé, il s’agit, à partir de cet ensemble d’objets caractérisés, d’induire en utilisant un ensemble d’apprentissage, un modèle, s’il en existe, qui permet de relier un ou plusieurs descripteurs particuliers, appelés variables endogènes, classes, étiquettes, ..., à tout ou partie des autres variables. Ces attributs particuliers sont en fait des variables statistiques appelées variables endogènes. Dans la suite de l’exposé nous utiliserons indifféremment les expressions « attribut classe », « variable endogène », ou simplement « étiquette ». De même, pour simplifier la présentation générale de l’apprentissage supervisé, nous considérons une seule variable classe notée  $Y$ . La généralisation au cas des  $m$  variables endogènes est immédiate et naturelle. Nous y reviendrons en temps voulu.

A chaque individu  $i \in \Omega$  peut être associée son étiquette  $Y(i)$  ;  $Y$  prend ses valeurs dans l’ensemble des étiquettes noté  $\mathcal{E}$ , voir équation (1.2.2). Par exemple, si la population  $\Omega$  est celle de patients en cancérologie et  $Y$  le résultat du diagnostic du médecin *normal*, *bénin* ou *malin*, alors  $Y(i)$  sera le résultat de l’examen clinique de l’individu  $i$ .

Dans la réalité, il peut s’avérer être difficile ou coûteux de pouvoir observer directement  $Y(i)$ . Par exemple, pour l’établissement du diagnostic des analyses cliniques invasives telles des prélèvements d’éléments organiques, peuvent être nécessaires. Pour cela, nous cherchons un moyen  $\varphi$  permettant de prédire la classe  $Y$  et ainsi limiter les analyses complémentaires pour connaître le résultat.

La détermination du modèle de prédiction  $\varphi$  est liée à l’hypothèse selon laquelle les valeurs prises par la variable statistique  $Y$  ne relèvent pas du hasard mais de certaines situations particulières pouvant être caractérisées. Pour cela l’expert du domaine concerné établit une liste *a priori* de variables exogènes. Le Chapitre 1 a détaillé ce point.

L'objectif est de rechercher un modèle de prédiction  $\varphi$  permettant, pour un individu  $i$  issu de  $\Omega$ , pour lequel nous ne connaissons pas l'étiquette  $Y(i)$  mais dont nous connaissons les valeurs  $(x_i^1, \dots, x_i^p)$  pour chacune des variables exogènes  $\Gamma(i)$ , de prédire cette valeur grâce à  $\varphi$ , voir équation (1.2.3).

L'apprentissage supervisé se propose donc de fournir des outils permettant d'extraire, à partir de l'information dont on dispose sur un échantillon dit d'apprentissage  $\Omega_{App}$ , le modèle de prédiction  $\varphi$ . Ce modèle  $\varphi$  peut prendre des formes diverses : une expression algébrique, une expression logique, un algorithme plus ou moins complexe, un réseau neuronal, ... Il convient de préciser cependant que certaines approches ne cherchent pas un modèle de prédiction explicite. Par exemple, dans les méthodes à base d'instance, comme les kppv ou les modèles à base de graphe de voisinage, la fonction de classement est basée sur une agrégation des étiquettes des voisins de l'individu dont nous cherchons son étiquette. Cette fonction d'agrégation est généralement l'étiquette majoritaire. Ainsi, un individu à classer hérite de l'étiquette la plus fréquente dans son voisinage.

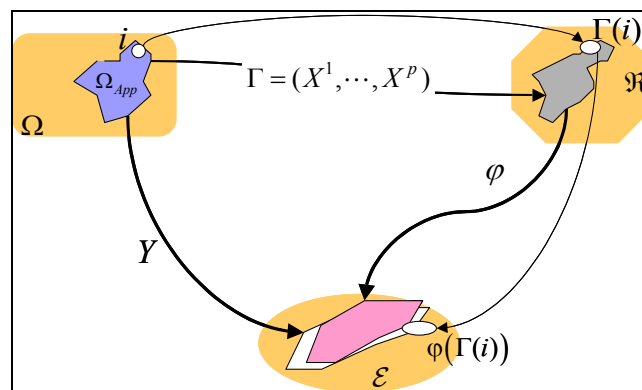


Figure 2-1 Représentation du processus d'apprentissage supervisé

Le processus général d'apprentissage comporte ainsi trois étapes (cf. Figure 2-1) pouvant se résumer comme suit (ZIGHED et RAKOTOMALALA, 2000, pp. 38-39) :

1. La première porte sur l'élaboration du modèle. Elle fait appel à un échantillon d'apprentissage noté  $\Omega_{App}$  dont tous les individus sont décrits dans un espace de représentation noté  $\mathfrak{R}$  et appartiennent à l'une des  $m$  étiquettes notées  $e_k; k = 1, \dots, m$ . Il s'agit alors de construire l'application  $\varphi$  qui permet de calculer la classe à partir de la représentation ;
2. La deuxième étape est celle de la validation. Il s'agit de vérifier, sur un échantillon test  $\Omega_{Test}$  (n'ayant pas servi à l'apprentissage) dont nous connaissons, pour chacun de ses individus, la re-

## Chapitre 2 – Apprentissage à base d’instances

---

présentation et la classe, si le modèle de prédiction  $\varphi$  issu de l’étape précédente donne bien la classe attendue ;

3. Enfin, la dernière phase est la généralisation qui consiste à étendre l’application du modèle à tous les individus de la population  $\Omega$ .

Dans la suite de ce chapitre, nous allons nous intéresser aux méthodes d’apprentissage à base d’instances, qui ont la particularité de déterminer la variable à prédire de l’individu à partir des variables endogènes d’un sous-ensemble d’individus déterminé à l’aide d’une mesure de similarité dans l’espace des descripteurs. L’apprentissage à base d’instance est également dénommé apprentissage en extension ou encore apprentissage à base de cas pour rappeler sa parenté avec le raisonnement à partir de cas. Nous allons aborder dans un premier temps, section 2.2, une méthode bien connue en apprentissage, les  $k$  plus proches voisins. Nous verrons à travers des exemples que cette dernière comporte quelques anomalies qui, à notre sens, révèlent des insuffisances de ce modèle.

Nous aborderons alors dans un second temps, section 2.3, les méthodes de graphes de voisinage. Ces dernières ne rencontrent pas les anomalies évoquées précédemment notamment grâce à l’ajout d’une structure reliant les différents individus de la population. Nous fournirons des exemples de représentations de ces diverses méthodes en nous basant essentiellement sur un jeu de données comportant deux variables exogènes  $\Gamma = \{X^1, X^2\}$ , et comportant une endogène à deux étiquettes  $\mathcal{E} = \{\text{rouge, vert}\}$ . Ce jeu de données est représenté en Figure 2-2.

Enfin, nous avons vu que pour déterminer si la phase d’apprentissage a atteint son but, il fallait évaluer la qualité de ses résultats. Ainsi, en dernière section (cf. 2.4), nous passerons en revue les principales méthodes de validation permettant d’évaluer les performances des classifieurs.

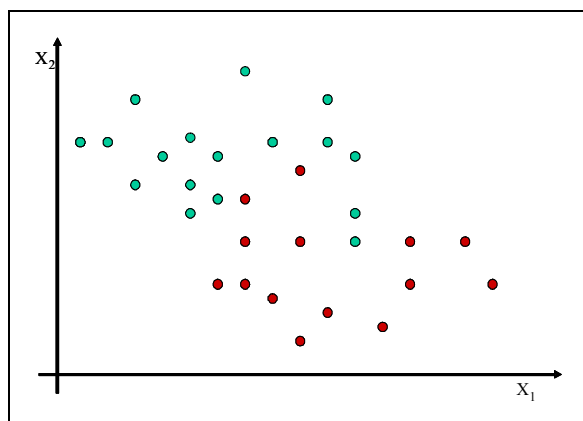


Figure 2-2 Données exemples

---

## 2.2 Les k Plus Proches Voisins

### 2.2.1 Introduction

L'apprentissage à base d'instances, encore appelé apprentissage à base d'exemples, vise à déterminer l'étiquette d'un individu à travers l'usage des individus appartenant à la population  $\Omega_{App}$ . Ici, nous allons décrire une méthode de cette nature, appelée *k* Plus Proches Voisins (kPPV).

Cette méthode non-paramétrique provient de (FIX et HODGES, 1951 ; COVER et HART, 1967). Les individus de l'espace d'apprentissage sont projetés dans l'espace multidimensionnel  $\Gamma$ . L'hypothèse sous-jacente à l'utilisation des kPPV est de supposer que les individus situés dans une même région de l'espace de représentation ont une valeur endogène similaire.

Dans la suite de cette section, nous allons d'abord présenter en 2.2.2 les phases d'apprentissage et de classement. Nous y détaillerons les principaux procédés de prise de décision consistant à retourner une étiquette ou un ensemble d'étiquettes en fonction de celles des plus proches voisins. Enfin, en 2.2.3 nous soulignerons des anomalies que nous pouvons rencontrer lors de l'utilisation de telles méthodes.

### 2.2.2 Présentation de la méthode

L'algorithme des kPPV effectue un apprentissage dit « paresseux » dans le sens où le modèle  $\varphi$  mémorise seulement l'ensemble des  $n_{App}$  individus de l'ensemble d'apprentissage dans l'espace des  $p$  descripteurs (Algorithme 2-1). Les versions plus sophistiquées du kPPV vont consister principalement à élaborer des stratégies permettant une recherche plus efficiente des  $k$  voisins plus proches de l'individu à classer (e.g. utilisation d'index).

---

#### Algorithme 2-1 Apprentissage des kPPV

---

```
ApprentissagekPPV(entrées:  $n_{App}$ , D; entrée-sortie:  $\varphi$ )
Début
     $\varphi \leftarrow \emptyset$ ; //Initialisation
    Pour  $i \leftarrow 1$  à  $n_{App}$  Faire
         $\Phi[i] \leftarrow D[i]$ ;
    FinPour
Fin.
```

---

## Chapitre 2 – Apprentissage à base d’instances

---

La partie classement ou prédiction est en contre-partie plus coûteuse en temps. En effet, pour prédire la valeur  $y$  de l’endogène d’un individu  $x$ , le kPPV doit d’abord chercher les  $k$  individus de  $\Omega_{App}$  plus proches de  $x$  au sens d’une fonction de similarité  $\delta$  définie par l’utilisateur (Algorithme 2-2). Dans sa version élémentaire, cette étape a une complexité en  $O(pn_{App})$  pour chaque individu à prédire puisqu’il faut calculer la distance entre chaque individu de l’ensemble d’apprentissage et l’individu à prédire.

---

### Algorithme 2-2 Classement par les kPPV

---

```
ClassementkPPV(entrées:  $k, \phi, n_{App}, \delta, x$ ; sortie:  $E$ )
Début
  ppv[]  $\leftarrow \emptyset$ ; //Initialisation du tableau des  $k$  plus proches voisins
  dppv[]  $\leftarrow \emptyset$ ; //Initialisation du tableau contenant leur similarité
  Pour  $i \leftarrow 1$  à  $k$  Faire //La seule condition sur  $k$  est :  $k \leq n_{App}$ 
    ppv[ $i$ ]  $\leftarrow \Phi[i]$ ;
    dppv[ $i$ ]  $\leftarrow \delta(x, \Phi[i])$ ;
  FinPour
  Trier(ppv[], dppv[], descendant); //Trier selon la similarité dppv
  Pour  $i \leftarrow k+1$  à  $n_{App}$  Faire
    tmp  $\leftarrow \delta(x, \phi[i])$ ;
    //i est-il un point candidat au kPPV de  $x$  ?
     $j \leftarrow 1$  à  $k$ ;
    Tant que ( $j \leq k$ ) Faire
      Si (tmp < dppv[ $j$ ])
        Alors
          ppv[ $k$ ]  $\leftarrow \phi[i]$ ;
          Trier(ppv[], dppv[], descendant);
           $j \leftarrow k$ ;
        FinSi
       $k \leftarrow k+1$ ;
    FinTantQue
  FinPour
  //Prise de décision
   $E \leftarrow \Psi(ppv[], dppv[], k)$ ;
Fin.
```

---

La prise de décision de cette méthode s’appuie donc sur les valeurs des  $k$  individus ainsi sélectionnés ; nous notons par  $\Psi$  la fonction qui produit cette valeur. Il en existe plusieurs définitions dépendant principalement de la nature de l’endogène : numérique (2.2.2.1), catégorielle (2.2.2.2) ou composée de plusieurs endogènes (2.2.2.3).

### 2.2.2.1 Cas d'une endogène numérique

Dans le cadre où  $E$  est une endogène numérique unique, notée  $Y$ , alors  $\Psi$  est définie comme étant la moyenne des valeurs de  $Y$  des  $k$  voisins de  $x$  (2.2.1). Afin d'être plus résistante au bruit provenant des individus relativement éloignés, cette moyenne peut être pondérée par une fonction de la distance, notée  $\varpi$ , séparant  $x$  de ses voisins (2.2.2) (DUDANI, 1976). Généralement, cette fonction de pondération est définie comme l'inverse de la distance (2.2.3) amenant ainsi à prendre faiblement en considération les voisins fortement éloignés pour privilégier les plus proches. Enfin, MITCHELL (1997) propose d'étendre la définition des méthodes d'apprentissage à base d'instances à toute méthode dont la prédiction se base sur les valeurs endogènes (et non exogènes) d'un sous-ensemble d'individus appartenant à l'ensemble d'apprentissage. Dès lors, comme rappelée dans (MUHLENBACH, 2002), la prédiction peut se baser sur une régression locale, visant à adapter une fonction linéaire ou quadratique aux  $k$  plus proches voisins, ou en utilisant des fonctions à bases radiales approximant  $Y$  par des combinaisons linéaires d'approximations locales.

$$\Psi(ppv, k) = \frac{1}{k} \sum_{i=1}^k Y(ppv(i)) \quad (2.2.1)$$

$$\Psi(ppv, dppv, k) = \sum_{i=1}^k \varpi(dppv(i)) \cdot Y(ppv(i)) \quad (2.2.2)$$

$$\varpi(z) = \begin{cases} 1 & \text{si } z = 0 \\ \frac{1}{z} & \text{sinon} \end{cases} \quad (2.2.3)$$

### 2.2.2.2 Cas d'une endogène catégorielle

Lorsque  $E$  est une endogène catégorielle, notée  $Y$ , la prise de décision consiste alors à déterminer l'étiquette de  $x$  en fonction de celles de ses voisins. Dans le cas où l'individu ne porte qu'une et une seule étiquette  $e$  parmi l'ensemble des étiquettes  $\mathcal{E}$ , alors le vote à la majorité est largement utilisé. Il consiste à assigner comme étiquette celle se trouvant majoritaire parmi les  $k$  voisins, voir l'équation (2.2.4). Enfin, pour les mêmes raisons qu'au 2.2.2.1, le vote peut également être pondéré par  $\varpi$ , voir l'équation (2.2.5).

$$\Psi(ppv, k) = \arg \max_{e \in \mathcal{E}} \left( \sum_{i=1}^k g(e, Y(ppv(i))) \right) \quad (2.2.4)$$

## Chapitre 2 – Apprentissage à base d’instances

---

$$\Psi(ppv, dppv, k) = \arg \max_{e \in \mathcal{E}} \left( \sum_{i=1}^k \varpi(dppv(i)) \cdot g(e, Y(ppv(i))) \right) \quad (2.2.5)$$

avec  $\forall a, b \in \mathcal{E}, g(a, b) = 1$  si  $a = b$  et  $g(a, b) = 0$  sinon.

### 2.2.2.3 Cas de plusieurs endogènes

Dans un cadre plus général, un individu peut appartenir à plusieurs étiquettes. Cela revient à considérer  $E = (Y^1, \dots, Y^k, \dots, Y^m)$  et  $y_i^k \in [0, 1]$  le degré d’appartenance qu’a l’individu  $i \in \Omega$  d’appartenir à  $e_k \in \mathcal{E}$ . La prise de décision ne consiste donc plus à déterminer l’étiquette dont le degré est le plus important, mais le degré d’appartenir à chacune des  $m$  étiquettes :

$$\Psi(ppv, k) = \frac{1}{k} \sum_{i=1}^k Y^j(ppv(i)), \forall j \in [1, m] \quad (2.2.6)$$

De même que pour les cas précédents, un système de pondération fonction de la distance peut être mis en œuvre.

Enfin, pour se ramener dans le cas binaire, i.e.  $y_i^k \in \{0, 1\}$ , il faut adjoindre une étape supplémentaire consistant à déterminer à quel seuil il faut considérer ou non l’appartenance à une étiquette.

### 2.2.3 Anomalies des kPPV

Nous l’avons vu, le principe du kPPV repose sur la proximité entre les individus. Cependant, l’utilisation de ce seul critère nous semble potentiellement insuffisante. En effet, d’un point de vue cognitif, lorsqu’un individu  $a$  a pour plus proche voisin un individu  $b$ , nous nous attendons que réciproquement  $b$  soit voisin de  $a$ .

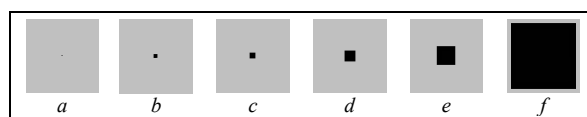


Figure 2-3 Liste de 6 images exemples

Pour illustrer cette remarque, nous avons construit un exemple : nous avons calculé la matrice des distances (Tableau 2-1.a) entre les 6 documents images de la Figure 2-3. Nous avons défini la distance



entre deux images comme étant la différence des sommes des valeurs en niveau de gris de chaque pixel.

Supposons que nous cherchions les voisins de l'image  $d$ . Un système basé sur le 2PPV retourne alors les images  $b$  et  $c$  (Tableau 2-1.b), puisque, d'après la matrice des distances, elles sont les plus proches de  $d$ . Dès lors, il paraît légitime de s'attendre à retrouver au moins l'image  $d$  lors de la recherche des 2PPV de  $b$  ou  $c$ . Pourtant, le système basé sur le 2PPV retourne les documents  $a$  et  $c$  dans le premier cas et  $a$  et  $b$  dans le second. Cette propriété de non-symétrie des kPPV a été soulignée plusieurs fois, et n'est pas à considérer comme étant un écueil à la catégorisation puisque, dans ce contexte, la prise de décision pondérée par l'inverse de la distance permet d'atténuer fortement cet effet. Toutefois, lorsque la stabilité des résultats doit être garantie<sup>2</sup>, nous assumons que le critère de symétrie est une des conditions à observer. Comme nous le verrons en section 2.3 des graphes de voisinage comme celui des voisins relatifs (GVR) garantissent cette condition (Tableau 2-1.c).

Distance	$a$	$b$	$c$	$d$	$e$	$f$
$a$	-					
$b$	0,01	-				
$c$	0,03	0,02	-			
$d$	0,09	0,08	0,06	-		
$e$	0,20	0,19	0,16	0,11	-	
$f$	0,58	0,57	0,55	0,50	0,40	-

(a)

2 Plus Proches Voisins	$a$	$b$	$c$	$d$	$e$	$f$
$a$		■	■			
$b$		■	■			
$c$		■	■			
$d$				■	■	
$e$				■	■	
$f$						■

(b)

Graphe des Voisins Relatifs	$a$	$b$	$c$	$d$	$e$	$f$
$a$		■				
$b$	■					
$c$		■				
$d$			■			
$e$			■			
$f$						■

(c)

Tableau 2-1 Matrice des distances (a) ; Modèle du 2PPV (b) ; Modèle du GVR (c)  
les cellules noires indiquent que l'élément colonne est voisin de l'élément ligne

Afin de montrer que la notion de voisinage doit être encore mieux définie que par une simple mesure de similarité, considérons un autre exemple. Soient les individus de la Figure 2-4. La recherche des plus proches voisins de l'individu  $\alpha$  va donner des résultats différents par rapport aux voisins (reliés par une arête à  $\alpha$  sur la figure) obtenus par la méthode des voisins relatifs (cf. 2.3.3.3). En fixant  $k = 4$ , les 2 derniers plus proches voisins sont sélectionnés par la méthode. Cependant, intuitivement

<sup>2</sup> Nous verrons à la section 4.4.2 du Chapitre 4 un système nécessitant cette propriété.

## Chapitre 2 – Apprentissage à base d'instances

---

nous les considérons davantage comme des voisins de voisins de  $\alpha$ . Quel est le résultat le plus pertinent ? Notre choix se porte sur les graphes de voisinage.

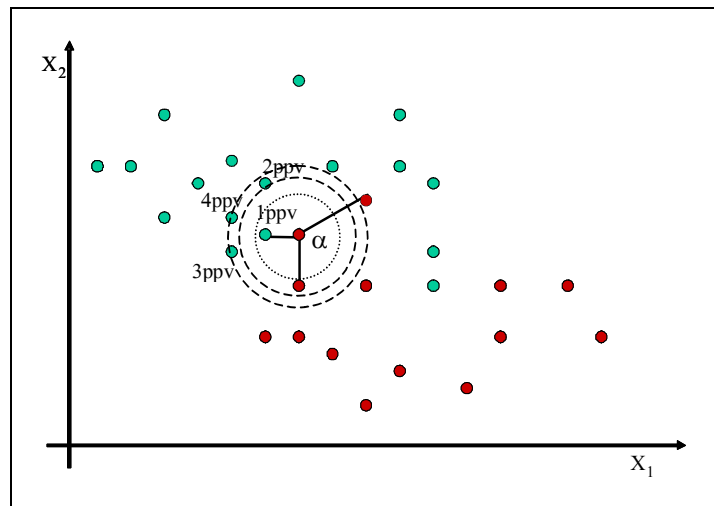


Figure 2-4 kPPV et voisins relatifs de l'individu  $\alpha$

Enfin, un inconvénient intrinsèque à la méthode est le choix du paramètre  $k$ . L'impact de son réglage se fait ressentir sur les performances du classifieur. En effet, comme l'illustre la Figure 2-5, la prédiction de l'étiquette de l'individu  $\alpha$  varie en fonction du choix de  $k$  :  $\beta$  est prédit vert pour  $k$  fixé à 1, et rouge pour  $k$  fixé à 3.

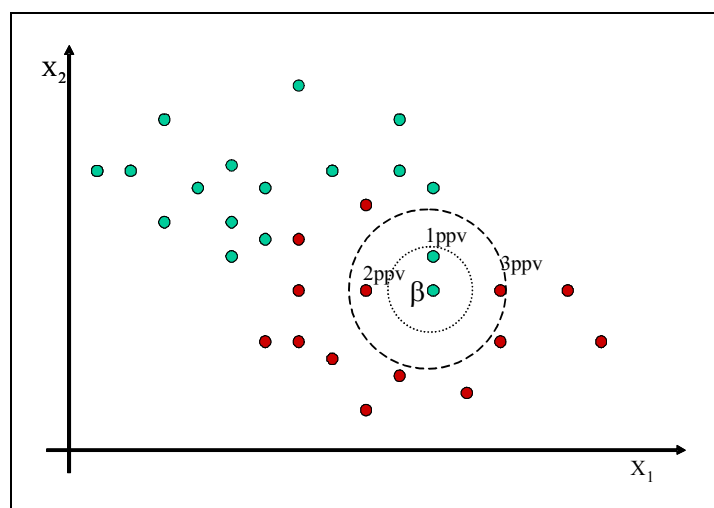


Figure 2-5 Exemple de 1 et 3 ppv de l'individu  $\beta$

---

## 2.3 Les graphes de voisinage

### 2.3.1 Introduction

Nous avons exposé le principe de fonctionnement de la méthode des kPPV. Nous avons également fait remarquer que cette méthode pouvait comporter des anomalies qui, selon les conditions et objectifs d'utilisation de la méthode, peut amener à des résultats troublant pour l'utilisateur. Nous proposons ici d'étudier des méthodes dites graphes de voisinage provenant de la géométrie computationnelle (PREPARATA et SHAMOS, 1985). A notre sens, ces méthodes permettent de mieux définir la proximité entre des individus par la notion de voisinage. De même que pour les kPPV, les graphes de voisinage nécessitent une mesure de similarité  $\delta$ .

Nous allons dans un premier temps, en section 2.3.2, rappeler quelques notions issues de la théorie des graphes afin de permettre la description de telles méthodes. Nous nous restreignons volontairement à la description de quatre méthodes qui ont les propriétés de symétrie et de connexité que nous recherchons (cf. 2.3.3). Enfin, nous aborderons l'aspect prédictif de telles méthodes en section 2.3.4.

### 2.3.2 Définitions et notations

Afin de décrire la conception des graphes de voisinage, nous reprenons ici le formalisme issu de la théorie des graphes, employé par BARTHELEMY et GUENOCHÉ (1988) et MULHENBACH (2002).

**Graphe, sommets, arêtes.** Un graphe est formé d'un ensemble de sommets noté  $\Sigma$  et les sommets sont reliés entre eux par un ensemble d'arêtes noté  $A$ . Le graphe  $G$  est ainsi décrit par la paire  $(\Sigma, A)$  où  $A \subseteq \Sigma \times \Sigma$ . Il permet de représenter l'existence ou l'absence de liaisons entre des objets. L'arête formée par deux sommets  $\alpha$  et  $\beta$  est notée  $(\alpha, \beta)$ .

**Graphe valué, longueur.** Un graphe valué est un couple  $(G, L)$  formé d'un graphe  $G = (\Sigma, A)$  et d'une fonction  $L$ , à valeurs réelles strictement positives, définies sur  $A$ .  $L(\alpha, \beta)$  est appelée la longueur de l'arête  $(\alpha, \beta)$ . Nous utiliserons la notation simplifiée  $L_{\alpha\beta}$  lorsqu'il n'y aura pas d'ambiguïté.

**Voisin.** Un sommet est voisin d'un autre s'il est relié à ce dernier par une arête. L'ensemble des voisins d'un sommet  $\alpha$  est noté  $\mathcal{V}_\alpha$  et est défini comme suit :  $\mathcal{V}_\alpha = \{\beta \in \Sigma \mid (\alpha, \beta) \in A\}$ .

**Connexité, chemin.** Un graphe est dit connexe lorsqu'il existe au moins une suite d'arêtes entre chaque sommet de  $G$ . Ainsi, dans un graphe connexe, pour tout couple de points  $(\alpha, \beta) \in \Sigma^2$ , il existe une succession d'arêtes, appelée chemin, joignant  $\alpha$  à  $\beta$ .

## Chapitre 2 – Apprentissage à base d’instances

---

**Ordre de voisinage.** Un sommet est voisin à l’ordre  $\tau$  d’un autre sommet, si exactement  $\tau$  sommets appartiennent au plus court chemin les séparant.

Nous remarquerons que les voisins d’ordre 1 seront appelés simplement voisins.

**Cycle.** Un cycle est un chemin qui, passant par un sommet donné, retourne à ce même sommet par une suite d’arêtes différentes.

**Arbre.** Un arbre est un graphe connexe sans cycle.

Nous remarquerons que le chemin entre deux sommets d’un arbre est donc unique.

### 2.3.3 Différents graphes de voisinage

#### 2.3.3.1 Graphes des polyèdres de Delaunay

Dans  $\mathbb{R}^p$ ,  $(p+1)$  points définissent un polyèdre de Delaunay si et seulement si l’hypersphère circonscrite aux  $(p+1)$  points ne contient aucun point. Le graphe des polyèdres de Delaunay, noté GPD, sera défini comme étant l’union des polyèdres de Delaunay. Dans  $\mathbb{R}^2$ , comme sur la Figure 2-6, on parle de triangulation de Delaunay, et les points  $\alpha$ ,  $\beta$  et  $\gamma$  forment un triangle de Delaunay car le cercle circonscrit à ces points ne contient aucun autre point.

Cette méthode détermine pour chaque point un nombre important de voisins. Cependant, comme

l’indique (SEBBAN, 1996), sa complexité algorithmique est en  $O\left(\frac{p^4}{2} n^{\frac{p+2}{2}}\right)$ .

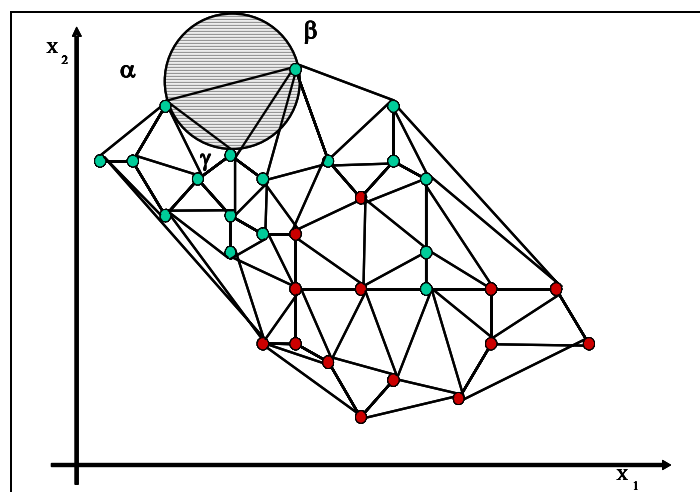


Figure 2-6 Exemple de triangulation de Delaunay

### 2.3.3.2 Le graphe de Gabriel

Le graphe de Gabriel, noté GG, est un graphe où deux points  $\alpha$  et  $\beta$  sont voisins si ils vérifient la propriété suivante : « L'hypersphère de diamètre  $[\alpha, \beta]$  doit être vide, i.e. elle ne contient aucun autre point », voir Figure 2-7. En d'autres termes, en notant  $\mu$  le centre de l'arête  $(\alpha, \beta)$ , alors les points  $\alpha$  et  $\beta$  sont voisins si l'inéquation (2.3.1) est vérifiée :

$$\delta(\alpha, \beta) \geq \delta(\alpha, \mu) = \delta(\beta, \mu) = \frac{\delta(\alpha, \beta)}{2}, \forall \gamma \in \Omega \setminus \gamma \neq \alpha, \beta \quad (2.3.1)$$

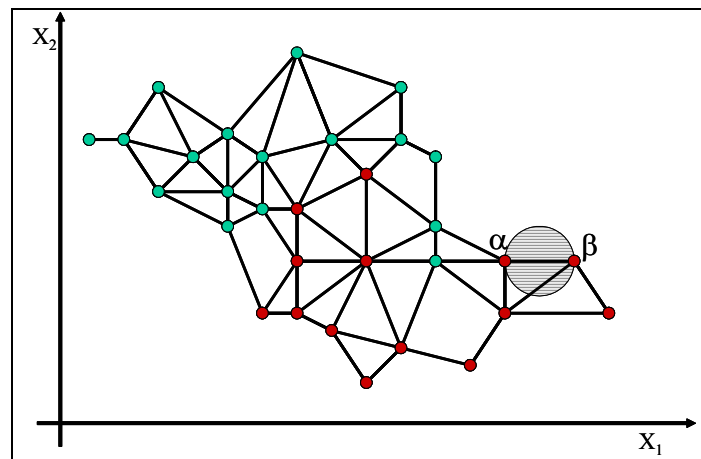


Figure 2-7 Exemple de graphe de Gabriel

### 2.3.3.3 Le graphe des voisins relatifs

Le graphe des voisins relatifs, noté GVR, est un graphe où deux points  $\alpha$  et  $\beta$  sont voisins si ils vérifient la propriété suivante : « La lunule, correspondant à l'intersection des sphères de rayon  $[\alpha, \beta]$  et de centres  $\alpha$  et  $\beta$ , doit être vide », voir Figure 2-8. En d'autres termes, les points  $\alpha$  et  $\beta$  sont voisins si l'inéquation (2.3.2) est vérifiée :

$$\delta(\alpha, \beta) \leq \max(\delta(\alpha, \gamma), \delta(\beta, \gamma)), \forall \gamma \in \Omega \setminus \gamma \neq \alpha, \beta \quad (2.3.2)$$

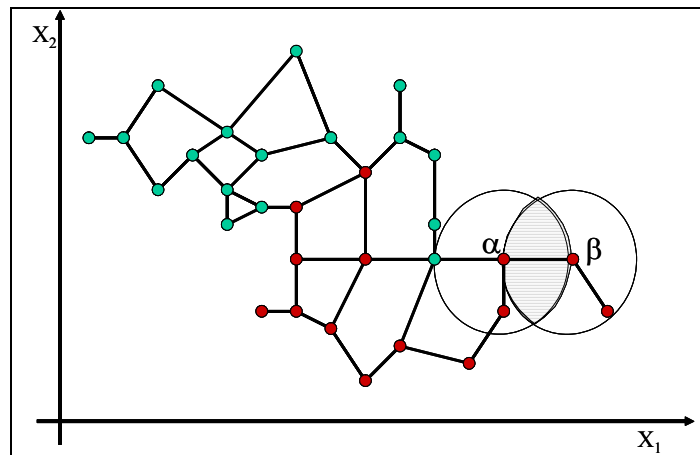


Figure 2-8 Exemple de graphe de voisins relatifs

### 2.3.3.4 L’arbre de recouvrement minimum

Ce graphe de voisinage, noté ARM, est un arbre valué  $(H, L)$  avec  $H = (\Sigma, A)$  dont la somme des longueurs des arêtes le composant est minimale. Plus formellement, l’arbre de recouvrement minimum est la solution de l’équation (2.3.3). La Figure 2-9 représente un tel arbre sur nos données exemples. Ce graphe est un arbre, donc sans cycle, dont le nombre d’arêtes est égal à  $(n-1)$  (BARTHELEMY et LUONG, 1998, p. 10-13).

$$H(\Sigma, A, L) = \underset{A}{\operatorname{Argmin}} \left( \sum_{(\alpha, \beta) \in A} L_{\alpha\beta} \right) \quad (2.3.3)$$

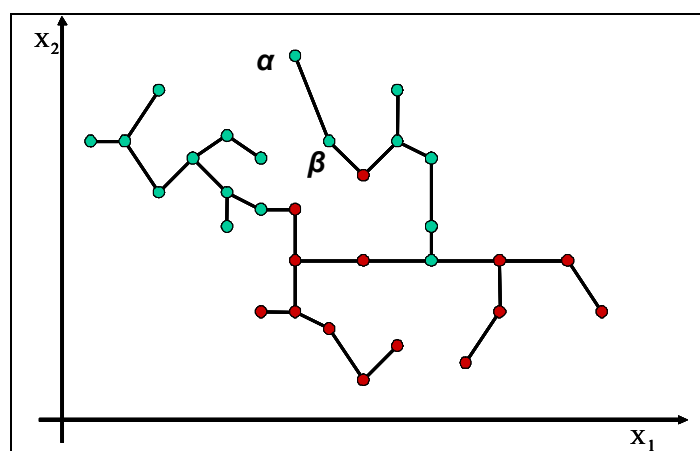


Figure 2-9 Exemple d’arbre de recouvrement minimum

---

Il existe plusieurs algorithmes de construction d'un tel arbre ; nous citerons principalement (KRUSKAL, 1956) et (PRIM, 1957) de complexité  $O(n^2)$  :

- Le premier aborde le problème en construisant la liste exhaustive des arêtes à partir de  $A$ . Cette liste est ordonnée selon leur valuation décroissante et les arêtes sont ajoutées à  $H$  si elles ne forment pas un cycle avec celles déjà sélectionnées ;
- Le second aborde le problème en sélectionnant aléatoirement un sommet de  $\Sigma$ , puis en rejoignant tous les sommets par ajout successif des plus petites arêtes connexes à l'arbre en cours d'élaboration.

#### 2.3.4 Graphes de voisinage et apprentissage supervisé

Intuitivement, nous nous rendons compte que les critères de construction présentés ici sont de plus en plus restrictifs. Ainsi, le GPD (cf. Figure 2-6) offre un nombre important d'arêtes pour chacun des sommets du graphe et à l'opposé, l'ARM (cf. Figure 2-9) propose la structure la moins dense. (PREPARATA et SHAMOS, 1985) rappellent l'existence de cette relation d'inclusion comme suit :

$$ARM \subseteq GVR \subseteq GG \subseteq GPD \quad (2.3.4)$$

Aux deux extrêmes nous trouvons donc le GPD et l'ARM. Le premier de ces graphes propose une exhaustivité des relations de voisinage pour un individu mais au tribut d'un coût prohibitif ; le second offre une structure squelettique des relations de voisinage, comportant 1 à 2 voisins pour un individu et offrant des résultats comparables au 1 ou 2 ppv mais avec une complexité supérieure (MUHLENBACH, 2002).

Nous allons donc privilégier les graphes GVR et GG. Ces derniers offrent un compromis sur le nombre de voisins, et leur mode de construction locale à un point permet d'obtenir une complexité en  $O(n^3)$ , voir Algorithme 2-3.

Enfin, nous ferons remarquer qu'en raison d'un critère plus restrictif dans le GVR que dans le GG, en pratique l'élaboration du GVR est plus rapide que celle du GG.

## Chapitre 2 – Apprentissage à base d’instances

---

---

### Algorithme 2-3 Construction d’un graphe de voisinage selon un critère local

---

```
ConstructionGV(entrées:  $n$ ,  $D$ ,  $\delta$ ; sortie:  $A$ ,  $L$ )
Début
   $A \leftarrow \emptyset$ ;  $L \leftarrow \emptyset$ ; //Initialisations
  Pour  $i \leftarrow 1$  à  $(n-1)$  Faire
    Pour  $j \leftarrow (i+1)$  à  $n$  Faire //Déterminons si  $(i,j)$  est une arête du GV
       $k \leftarrow 1$ ;
      Arete_ij  $\leftarrow$  vrai; //Initialisation
      Tant que  $(k \leq n)$  et Arete_ij Faire
        Si  $(k \neq i)$  et  $(k \neq j)$ 
          Alors
            Arete_ij  $\leftarrow$  VerifieCritereLocal( $D[i]$ ,  $D[j]$ ,  $D[k]$ ,  $\delta$ );
          FinSi
           $k \leftarrow k+1$ ;
        FinTantQue
      Si Arete_ij //( $i,j$ ) est une arête au sens du critère local
        Alors
           $A \leftarrow A \cup (i,j)$ ; //Ajout de l’arête
           $L[i,j] \leftarrow \delta(i,j)$ ; //Ajout de la longueur de l’arête
        FinSi
    FinPour
  FinPour
Fin.
```

---

---

### Algorithme 2-4 Classement par un graphe de voisinage selon un critère local

---

```
ClassementGV(entrées:  $n_{App}$ ,  $n_{Test}$ ,  $\Sigma_{App}$ ,  $\Sigma_{Test}$ ,  $D$ ,  $\delta$ ,  $x$ ; sortie:  $y$ )
Début
   $V \leftarrow \emptyset$ ; //Liste des voisins de  $x$ 
   $L \leftarrow \emptyset$ ; //Longueur des arêtes des voisins de  $x$ 
  Pour  $j \leftarrow 1$  à  $n_{App}$  Faire //Déterminons si  $(x,j)$  est une arête
     $k \leftarrow 1$ ;
    Arete_xj  $\leftarrow$  vrai; //Initialisation
    Tant que  $(k \leq n_{App})$  et Arete_xj Faire
      Si  $(k \neq j)$ 
        Alors
          Arete_xj  $\leftarrow$  VerifieCritereLocal( $D[x]$ ,  $D[j]$ ,  $D[k]$ ,  $\delta$ );
        FinSi
         $k \leftarrow k+1$ ;
      FinTantQue
    Si Arete_xj //( $x,j$ ) est une arête au sens du critère local
      Alors
         $V \leftarrow V \cup (x,j)$ ; //Ajout de l’arête
         $L[x,j] \leftarrow \delta(x,j)$ ; //Ajout de la longueur de l’arête
      FinSi
    FinPour
  //Prise de décision
   $y \leftarrow \Psi(V[], L[], |V|)$ ; //|V| est le nombre de voisins de  $x$ 
Fin.
```

---



---

Dans un contexte d'apprentissage, nous ne cherchons pas à connaître la structure du graphe de voisinage, mais à prédire la valeur  $y$  de l'individu  $x$ . Pour ce faire, nous utilisons les valeurs des voisins de  $x$  au sens de la structure du graphe de voisinage utilisé. Ainsi, il suffit seulement de déterminer les voisins de  $x$  ce qui est possible avec l'utilisation d'un critère local (cf. Algorithme 2-4).

Les méthodes de prises de décisions sont les mêmes que celles des kPPV (cf. équations (2.2.1) à (2.2.6)), à la seule différence qu'ici  $k$  varie d'un individu à un autre. D'autres méthodes de prises de décisions exploitant davantage la structure du graphe peuvent être envisagées en faisant par exemple intervenir les voisins d'un ordre supérieur (e.g. les voisins d'ordre 2). Cependant, la complexité s'en trouve fortement augmentée.

## 2.4 Validation

### 2.4.1 Introduction

Consensuellement, il est considéré qu'aucune méthode d'apprentissage ne surpasse les autres tant elles ont leurs forces et faiblesses propres. Dès lors, pour un problème spécifique, il est recommandé de tester plusieurs méthodes afin d'utiliser la « meilleure » d'entre elles ou de les faire coopérer.

La définition de la « meilleure » méthode est la plupart du temps liée à un ensemble de critères dont les plus courants peuvent être :

1. La satisfaction de l'utilisateur envers les réponses fournies et sa capacité à les interpréter ;
2. L'adéquation entre le temps d'exécution de l'algorithme avec la périodicité du problème et les contraintes utilisateur. Par exemple, dans le cadre d'un diagnostic médical urgent, il est nécessaire de disposer d'une réponse rapide ;
3. La fiabilité de la méthode doit être analysée afin d'évaluer le risque encouru par l'utilisation de cette méthode ;

Les deux premiers critères sont subjectifs et dépendants de la problématique ; il est donc délicat de formuler des indicateurs généraux. Nous allons aborder dans la suite de cette section les méthodes usuelles permettant d'analyser la fiabilité d'un classifieur. Ces dernières, comme le rappelle (ZIGHED et RAKOTOMALALA, 2000, p. 238), incluent une estimation honnête du taux d'erreur commis en généralisation, mais également l'analyse du type d'erreur commise (e.g. sur quelles classes la confusion

## Chapitre 2 – Apprentissage à base d’instances

est-elle la plus manifeste ? les coûts sont-ils symétriques ?) ainsi que la caractérisation des situations de leur occurrence (e.g. niveau de bruit, « forme » du concept, taille de l’échantillon d’apprentissage).

Bien évidemment, chacun de ces critères caractérise un élément particulier lié aux performances du classifieur, et l’intérêt de son choix dépend de la problématique. C’est pourquoi, différentes communautés scientifiques n’utilisent pas nécessairement les mêmes critères. La communauté de l’apprentissage automatique privilégie davantage les critères globaux comme le taux d’erreur alors que la communauté de la recherche d’information préfère étudier les performances partielles du classifieur. En effet, pour la première, la problématique la plus usuelle est la catégorisation mono-classe, c’est-à-dire que le document à classer n’appartient qu’à une et une seule catégorie. Alors que pour la seconde, un document peut appartenir à aucune ou aux  $m$  classes, engendrant ainsi le besoin d’analyser les performances de chacune de ces classes afin de pouvoir ventiler les qualités et défauts du classifieur. En ce sens, bons nombres de mesures existent. Cependant, une part importante d’entre elles sont basées sur les mesures de rappel et précision que nous allons aborder dans la suite de cette section.

### 2.4.2 Schéma apprentissage-validation

Le procédé habituel de validation consiste à estimer « au mieux » le taux d’erreur du modèle. Il est généralement calculé à partir de la matrice de confusion (cf. Tableau 2-2). Celle-ci donne le pourcentage d’affectation dans les différentes classes en fonction des classes d’origine.

		Classes d’affectation		
		$\hat{y}^1$	...	$\hat{y}^m$
Classes d’origine	$y^1$	$n_{11}$		$n_{1m}$
	$\vdots$		$\ddots$	
	$y^m$	$n_{m1}$		$n_{mm}$

Tableau 2-2 Matrice de confusion

Dans cette matrice, l’effectif  $n_{ij}$  représente le nombre de cas de la classe  $y^i$  ayant été affectés à la classe  $y^j$ . Ainsi, lorsque  $i = j$ , cette quantité représente le nombre correcte d’affectations. En étendant ce principe aux  $m$  classes, nous pouvons calculer le taux de succès, noté  $Succ$ , comme dans l’équation (2.4.1). Le taux d’erreur, noté  $Err$ , n’est autre que le complément à 1 du taux de succès, voir l’équation (2.4.2).

$$Succ = \frac{\sum_{i=1}^m n_{ii}}{\sum_{i=1}^m \sum_{j=1}^m n_{ij}} \quad (2.4.1)$$

$$Err = 1 - Succ \quad (2.4.2)$$

Ce taux d'erreur calculé sur l'échantillon d'apprentissage  $\Omega_{App}$  est généralement optimiste : il est plus faible que le « vrai » taux d'erreur. Ce dernier, appelé taux d'erreur théorique, serait obtenu en utilisant l'exhaustivité de la population concernée, ce qui ne peut être réalisé. Ce taux d'erreur est appelé « taux d'erreur en resubstitution ». Usuellement le taux d'erreur en validation lui est préféré car plus objectif : il est calculé de la même façon mais sur un échantillon test  $\Omega_{Test}$  qui, rappelons-le, n'a pas servi lors de la phase d'apprentissage.

Cette approche nécessite le découpage aléatoire des données en deux sous-ensembles, l'ensemble d'apprentissage et celui de test. Typiquement, ce découpage s'effectue dans les proportions suivantes :

$$\Omega_{App} = \frac{2}{3}\Omega \text{ et } \Omega_{Test} = \frac{1}{3}\Omega, \text{ avec } \Omega = \Omega_{App} \cup \Omega_{Test}.$$

Cependant, il peut arriver que la méthode nécessite le réglage automatique de paramètres comme dans le cas discuté en 2.2.2.3. Afin de conserver un ensemble test indépendant de toute utilisation en apprentissage, l'ensemble d'apprentissage est alors lui-même décomposé aléatoirement en deux sous-ensembles  $\Omega_{App'}$  pour les données d'apprentissage et  $\Omega_{Reg}$  pour les données servant aux réglages des paramètres. Des proportions typiques de ces trois sous-ensembles sont :  $\Omega_{App'} = \frac{1}{2}\Omega$ ,  $\Omega_{Reg} = \frac{1}{4}\Omega$  et

$$\Omega_{Test} = \frac{1}{4}\Omega, \text{ avec toujours } \Omega = \Omega_{App'} \cup \Omega_{Reg} \cup \Omega_{Test}.$$

Enfin, comme l'illustre la Figure 2-10, lors de l'utilisation de méthodes de pré-traitements (e.g. sélection de termes, élément **1** sur la Figure), les paramètres sont calculés à partir des données d'apprentissage puis le modèle (élément **2**) est élaboré. Lors de la phase de validation, les paramètres des méthodes de pré-traitements sont appliqués tels quels sur les données de validation, ce qui est symbolisé sur la Figure par la liaison entre l'élément **1** et **3**. Pour achever la validation, la prédiction des individus d'apprentissage est réalisée en **4** en appliquant le modèle obtenu en **2** sur les données pré-traitées provenant de **3**.

## Chapitre 2 – Apprentissage à base d’instances

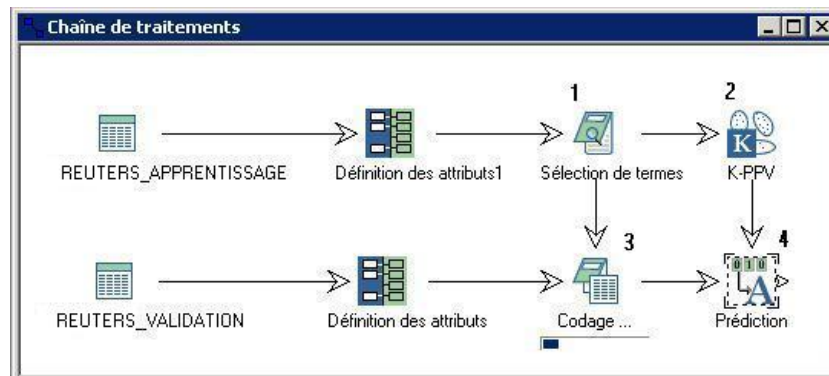


Figure 2-10 Chaîne de traitements en apprentissage-validation

### 2.4.3 Validation croisée

Les méthodes de ré-échantillonnage permettent de travailler avec un nombre plus restreint d’individus tout en s’approchant « au mieux » du taux d’erreur théorique. L’une des plus connues est la validation croisée (STONE, 1974).

La validation croisée consiste à répartir l’échantillon d’apprentissage aléatoirement en  $Q$  paquets d’effectifs identiques. En notant  $\Omega_q$ ,  $q = 1, \dots, Q$ , les différents sous-échantillons, alors le taux d’erreur en validation croisée est calculé en réservant, à tour de rôle, un échantillon  $\Omega_q$  qui servira à mesurer le taux d’erreur en validation, l’apprentissage étant réalisé sur la totalité des individus restants :  $\Omega - \Omega_q$ . Nous obtenons ainsi  $Q$  taux d’erreurs  $Err_q$ ,  $q = 1, \dots, Q$ . Le taux d’erreur en validation croisée est alors la moyenne arithmétique des taux d’erreurs partiels, voir équation (2.4.3), auquel sa variance lui est généralement associée, voir équation (2.4.4). Typiquement,  $Q$  est fixé à 10 qui est selon (KOHAVI, 1995) la solution la plus appropriée car donnant une bonne estimation de l’erreur et de son intervalle de confiance.

$$Err = \sum_{q=1}^Q \frac{|\Omega_q|}{|\Omega|} Err_q \quad (2.4.3)$$

$$\sigma_{Err}^2 = \frac{1}{Q} \sum_{q=1}^Q (Err_q - Err)^2 \quad (2.4.4)$$

Tout comme dans le cadre application-validation (cf. 2.4.2) et pour chacune des itérations, les paramètres du pré-traitement (e.g. sélection de termes, élément **1** sur la Figure 2-11) et du modèle (élément **2**

sur la Figure) doivent être élaborés à partir des individus servant à la  $q^{\text{ième}}$  phase d'apprentissage ( $\Omega - \Omega_q$ ) puis ces paramètres sont appliqués sur les individus test  $\Omega_q$ .

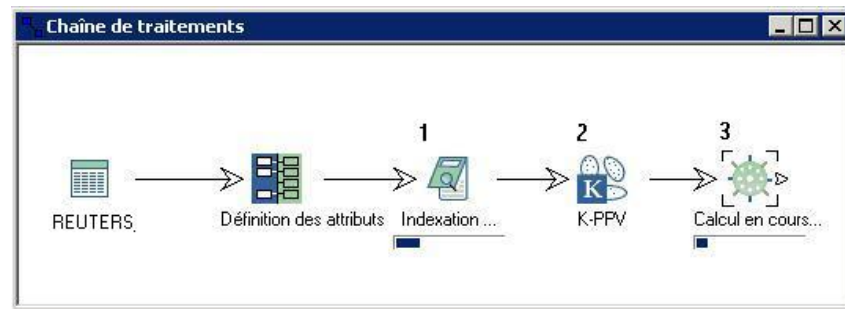


Figure 2-11 Chaîne de traitements en validation croisée

#### 2.4.4 Rappel et précision partiels et globaux

L'évaluation des performances partielles permet de déterminer sur quelles erreurs le classifieur est moins performant, de comparer plusieurs classifieurs, de les faire coopérer en les utilisant sur les catégories qu'ils prédisent le mieux. Les mesures que nous présentons ici sont également issues de la matrice de confusion (cf. Tableau 2-2). Signalons que les éléments de cette matrice sont évalués en généralisation comme en suivant par exemple le mode apprentissage-validation (cf. Figure 2-10) ou encore la validation-croisée (cf. Figure 2-11).

Comme le rappelle (SEBASTIANI, 2002) dans le cadre de la catégorisation de documents textuels (ce qui est tout aussi vrai dans la catégorisation en générale), deux mesures issues de la recherche d'information permettant de nous renseigner sur la qualité de classement de chacune des catégories sont fréquemment utilisées (SALTON, 1989, p. 248) :

- Le rappel d'une classe  $y^k$  est la probabilité qu'un document appartenant à  $y^k$  soit assigné à  $\widehat{y}^k$ . L'estimation du rappel est définie par l'équation (2.4.5) ;
- La précision d'une classe  $y^k$  est la probabilité qu'un document assigné à  $\widehat{y}^k$  appartienne effectivement à  $y^k$ . L'équation (2.4.6) définit l'estimation de la précision.

$$\widehat{\rho}_k = \frac{n_{kk}}{\sum_{j=1}^m n_{kj}} \quad (2.4.5)$$

## Chapitre 2 – Apprentissage à base d’instances

---

$$\widehat{\pi}_k = \frac{n_{kk}}{\sum_{i=1}^m n_{ik}} \quad (2.4.6)$$

Enfin, à partir de ces mesures partielles, nous pouvons estimer des indicateurs globaux, les rappel et précision globaux, tendant à évaluer ces mêmes caractéristiques, mais cette fois sur la globalité du classifieur. Deux méthodes pour les estimer sont définies : les rappel et précision « micro-moyennes » (*micro-averaging* en anglais) et les rappel et précision « macro-moyennes » (*macro-averaging* en anglais) :

- Les rappel et précision micro-moyennes, notés respectivement,  $\rho^\mu$  et  $\pi^\mu$ , sont obtenus par les sommes de toutes les décisions individuelles ; voir les équations (2.4.7) et (2.4.8) ;
- Les rappel et précision macro-moyennes, notés respectivement,  $\rho^M$  et  $\pi^M$ , sont évalués d’abord localement pour chaque classe, puis globalement en les moyennant ; voir les équations (2.4.9) et (2.4.10).

$$\widehat{\rho}^\mu = \frac{\sum_{k=1}^m n_{kk}}{\sum_{i=1}^m \sum_{j=1}^m n_{ij}} \quad (2.4.7)$$

$$\widehat{\pi}^\mu = \frac{\sum_{k=1}^m n_{kk}}{\sum_{j=1}^m \sum_{i=1}^m n_{ij}} \quad (2.4.8)$$

$$\widehat{\rho}^M = \frac{1}{m} \sum_{k=1}^m \widehat{\rho}_k \quad (2.4.9)$$

$$\widehat{\pi}^M = \frac{1}{m} \sum_{k=1}^m \widehat{\pi}_k \quad (2.4.10)$$

Les mesures micro-moyennes tiennent compte directement des effectifs bien classés de chacune des catégories (les cellules  $n_{kk}$ ,  $1 \leq k \leq m$  du Tableau 2-2). Elles privilégient ainsi les catégories de plus fort effectif, ce qui n’est pas le cas pour les mesures macro-moyennes. Par ailleurs, nous ferons remarquer que les rappel et précision micro-moyennes sont ici équivalents au taux de succès :  $\widehat{\rho}^\mu = \widehat{\pi}^\mu = Succ$  ; nous aborderons dans la section suivante (cf. 2.4.5) le cas où ces valeurs sont différentes.

## 2.4.5 Cas multi-étiquettes

Comme nous l'avons vu en introduction (cf. 2.1), le cas multi-étiquette peut être appréhendé de deux façons : une approche binaire consistant en une division en  $m$  sous-problèmes (2.4.5.1) et une approche globale (2.4.5.2).

### 2.4.5.1 Approche binaire

Cette approche, qui consiste en la division en  $m$  sous-problèmes de catégorisation binaire, engendre la réalisation de  $m$  matrices de confusion (cf. Tableau 2-2). Cependant, étant dans le cas binaire, la matrice de confusion pour la catégorisation de la classe  $c_i$  ( $1 \leq i \leq m$ ) se réduit au Tableau 2-3, où  $\overline{c}_i$  signifie « n'appartient pas à la classe  $c_i$  ».

		Classes d'affectation	
		$\widehat{c}_i$	$\overline{\widehat{c}_i}$
Classes d'origine	$c_i$	$n_{11}$	$n_{12}$
	$\overline{c}_i$	$n_{21}$	$n_{22}$

Tableau 2-3 Matrice de confusion – cas binaire

Les mesures partielles de rappel (2.4.5) et précision (2.4.6) restent applicables en considérant  $m = 2$ . De même, les mesures macro-moyennes restent valides, voir les équations (2.4.9) et (2.4.10). Cependant, les définitions des mesures micro-moyennes sont reformulées dans les équations (2.4.11) et (2.4.12).

$$\widehat{\rho}^\mu = \frac{\sum_{i=1}^m n_{11}^i}{\sum_{i=1}^m n_{11}^i + \sum_{i=1}^m n_{12}^i} \quad (2.4.11)$$

$$\widehat{\pi}^\mu = \frac{\sum_{i=1}^m n_{11}^i}{\sum_{i=1}^m n_{11}^i + \sum_{i=1}^m n_{21}^i} \quad (2.4.12)$$

### 2.4.5.2 Approche globale

Cette approche vise à prendre en considération les interactions potentielles entre les différentes catégories. Nous avons vu en 2.2 et 2.3 que les kPPV et les graphes de voisinage peuvent modéliser un tel

## Chapitre 2 – Apprentissage à base d’instances

---

cas en assignant à un document un vecteur de probabilité d’appartenance pour chaque catégorie. En déterminant un seuil au-delà duquel le document est considéré comme appartenant effectivement à cette catégorie, nous pouvons appliquer les formules évaluant les différents critères définis précédemment.

SALTON (1989) propose une alternative à cette approche : la méthode de la précision moyenne des 11 points. Cette méthode provient de la recherche documentaire et ne nécessite pas le choix de seuils pour les probabilités. Elle fixe 11 seuils de rappels allant de 0% à 100% par pas de 10% et la valeur de la précision pour chacune de ces valeurs est calculée. La moyenne de ces 11 valeurs de précision estime la capacité de catégoriser un document. La moyenne de ces résultats obtenus pour les différents documents de l’ensemble de test permet d’évaluer la capacité globale du classifieur sur ce corpus.

### 2.5 Conclusion

Dans ce chapitre, nous avons posé le cadre formel de l’apprentissage supervisé et de sa validation. Nous nous sommes intéressés plus particulièrement aux méthodes d’apprentissage à base d’instances qui ont la particularité de déterminer la variable à prédire de l’individu à partir des variables endogènes d’un sous-ensemble d’individus déterminé à l’aide d’une mesure de similarité dans l’espace des descripteurs.

Nous avons exposé le principe de fonctionnement de la méthode des kPPV dans un processus d’apprentissage supervisé. Nous avons également fait remarquer que cette méthode pouvait comporter des anomalies qui, selon les conditions et objectifs d’utilisation de la méthode, peut amener à des résultats troublant pour l’utilisateur. Il en ressort que la simple utilisation d’une mesure de dissimilarité est insuffisante dans ces cas. Les graphes de voisinage ajoutent une structure entre les divers individus. Cette dernière est à notre sens une solution qui permet de résoudre les inconvénients des kPPV au prix cependant d’un calcul plus important.

Enfin, nous avons décrit les principales méthodes de validation permettant d’évaluer les performances des modèles construits.