UFR de Mathématiques
Master 2 Ingénierie Statistique et Informatique
de la Finance de l'Assurance et du Risque

Rapport Projet

Apprentissage Statistique 2022
Sujet : Support Vector Machines (SVM)

Rédigé et présenté par :

Diamé SENGHOR
Oumar MARIKO

Année académique : 2021/2022

Table des matières

1 Introduction

Dans de nombreuses applications, les données d'entrée sont des fonctions échantillonnées prenant leurs valeurs dans des espaces dimensionnels infinis plutôt que dans des vecteurs standards. Ce fait a des conséquences complexes sur les algorithmes d'analyse des données, motivant ainsi leurs modifications. En fait, la plupart des outils traditionnels d'analyse de données pour la régression, la classification et le clustering ont été adaptés aux entrées fonctionnelles sous le nom général d'analyse de données fonctionnelles (FDA). L'article que nous exposons , étudie l'utilisation de l'analyse de données fonctionnelles et nous nous concentrons sur le problème de la discrimination des courbes. Les SVM sont des outils de classification à grande marge basés sur des mappages non linéaires implicites des données considérées dans des espaces de grande dimension grâce à des noyaux. L'article montre comment définir des noyaux simples qui prennent en compte la nature fonctionnelle des données et conduisent à une classification cohérente. Des expérimentations menées sur des données du monde réel soulignent l'intérêt de prendre en compte certains aspects fonctionnels des problèmes.

2 Définition

2.1 Données fonctionnelles

Une variable aléatoire est dite fonctionnelle si ses valeurs sont dans un espace de dimension infinie. Une observation d'une variable fonctionnelle est appelée donnée fonctionnelle.
$\mathbf{X} = \{\mathbf{X_t} : t \in \mathcal{T}\}$
$\mathbf{X_t} : \Omega \mapsto \mathbb{R}$
$\mathcal{T} \sqsubseteq \mathbb{R}$: courbe
$\mathcal{T} \sqsubseteq \mathbb{R}^2$: image
En pratique, une donnée fonctionnelle est observée dans un nombre fini d'instants.

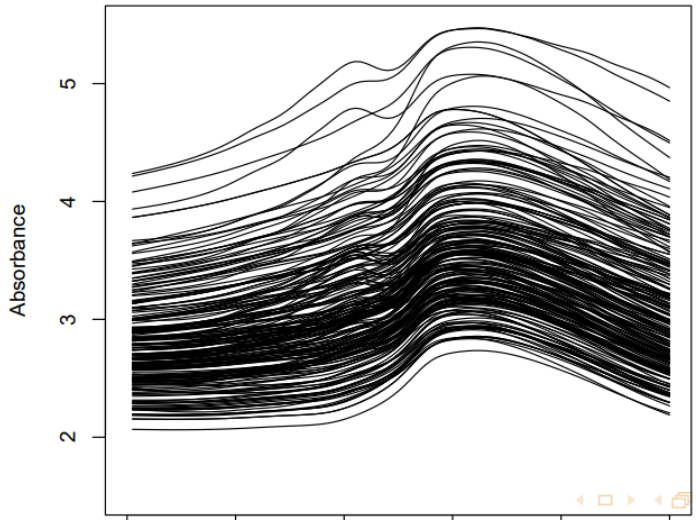
2.2 Analyse des données fonctionnelles

Le développement des appareils de mesure de grandeurs physiques a vu l'apparition, de manière de plus en plus fréquente, de données qui prennent la forme de fonctions discrétisées. De nombreux exemples de ce type de données se retrouvent dans des domaines d'applications divers comme la reconnaissance vocale, l'analyse des séries temporelles, la spectrométrie, etc. Ces données sont de nature tr'es particulière : la dimension de l'espace dans lequel elles prennent leurs valeurs est infinie (et, de manière pratique, le nombre de points de discrétisation est souvent très supérieur au nombre de points d'observations) et il éxiste une grande corrélation entre les différents points de discrétisation d'une même observation fonctionnelle. Ainsi, les outils de la statistique classique conduisent, si ils sont appliqués directement aux données discrétisées, à des problèmes mal posés et ne permettent pas de construire des classifieurs ou des régresseurs pertinents sur ces données.

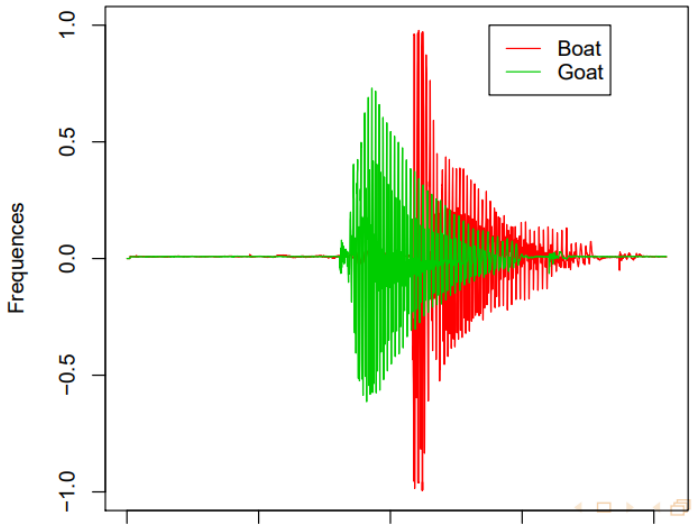
2.2.1 Exemples

Voici quelques exemples d'applications en Analyse des Données Fonctionnelles (FDA)

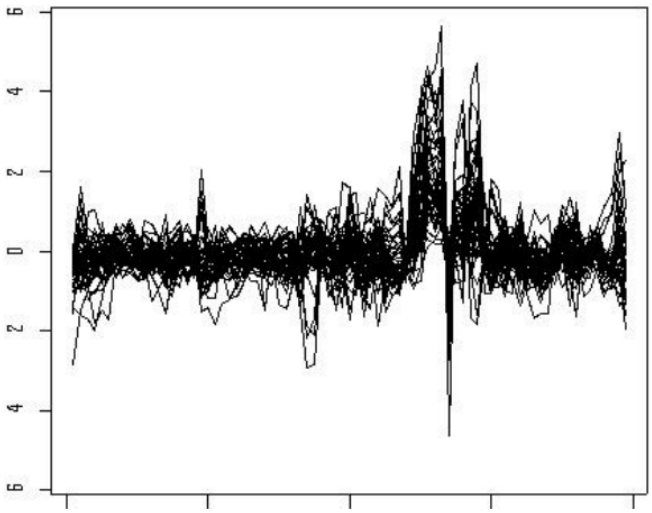
1. Analyse de données spectrométriques



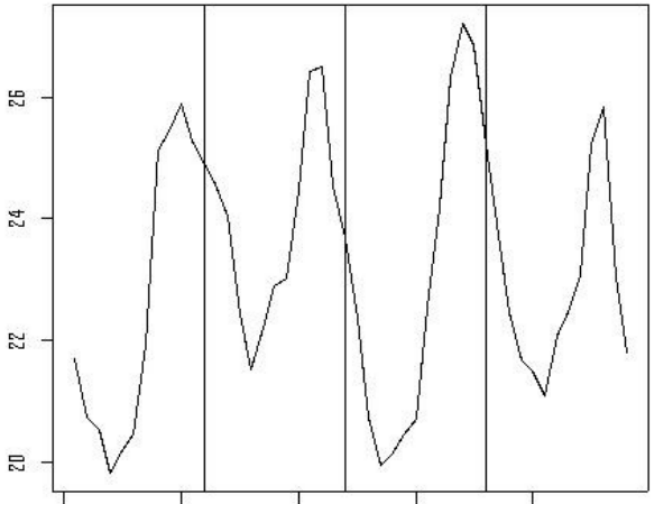
2. Reconnaissance vocale



3. Analyse de puces à ADN



4. Séries temporelles



2.2.2 Formulation mathématique

Le cadre

- X *(Variable explicative)* $\in (H, \langle ., . \rangle)$ *o*ù $(H, \langle , \rangle)$ est un espace de Hilbert ;
- Y *(Variable dépendante)* $\in \{-\mathbf{1}, \mathbf{1}\}$: ***Classification***
  ou $Y \in \mathbb{R}$: ***Regression***
- On cherche à prédire Y à partir de X.

Pour cela, on dispose d'un ensemble d'apprentissage $(X_1, y_1), ..., (X_n, y_n)$ tel que

- $x_i = (x_i(t_1), ..., x_i(t_d))$;
- (x_i, y_i) sont des réalisations du couple (X,Y).

Objectif : construire un prédicteur φ, à partir des observations, tel que $\mathbb{E}[E(Y, \varphi)]$ soit petit où E est une fonction d'erreur que l'on se fixe.

3 Introduction aux SVM

Les Support Vector Machines souvent traduit par l'appellation de Séparateur à Vaste Marge (SVM) sont une classe d'algorithmes d'apprentissage initialement définis pour la discrimination c'est-à-dire la prévision d'une variable qualitative binaire. Ils ont été ensuite généralisés à la prévision d'une variable quantitative. Dans le cas de la discrimination d'une variable dichotomique, ils sont basés sur la recherche de l'hyperplan de marge optimale qui, lorsque c'est possible, classe ou sépare correctement les données tout en étant maximisant la marge i.e maxismiser la distance minimale qui la sépre des points qui lui sont le plus proches. Dans le ces où cs données ne sont

pas linéairement séparables, une technique consiste à effectuer une transformation non linéaire des données (en utilisant un noyau) pour les passer dans un espace de dimension supérieure et y appliquer la séparation. Il convient de trouver un classifieur, ou une fonction de discrimination, dont la capacité de généralisation (qualité de prévision) est la meilleure possible.

SVM = Machines à Vecteurs de Support ; très populaires depuis les travaux sur l'apprentissage statistique [Vapnik, 1995]. Deux types de régularisation efficace :

- Régularisation par projection
- Régularisation par dérivation

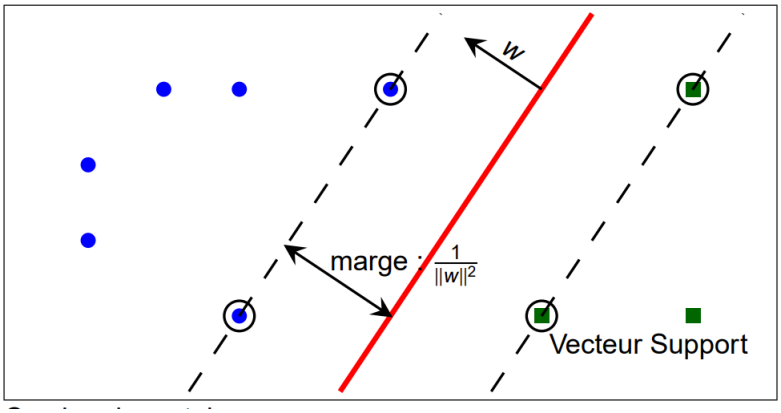
3.1 Discrimination linéaire à marge optimale

On veut trouver l'hyperplan de support w et de biais b qui permettent de maximiser cette marge, c'est-à-dire qu'on veut trouver un hyperplan avec la plus grande marge possible. Le principe du SVM est d'effectuer une discrimination affine des observations avec une marge maximale, c'est-à-dire minimiser $\|w\|$
On résoud le problème d'optimisation en minimisant $\frac{1}{2}\|w\|^2$ sous mles contraintes :

$$y_i(\langle w, x_i\rangle + b) \geq 1 \tag{1}$$

$$o\grave{u}\ 1 \leqslant i \leqslant n \tag{2}$$

Remarque : Cette formulation est valide pour des classes linéairement séparables. La règle de classification associée avec (w,b) est simplement $f(x) = signe(\langle w, x\rangle + b)$
Dans cette situation (appelée SVM à marge marge dure), nous exigeons que la règle ait une erreur nulle sur l'ensemble d'apprentissage



3.2 Discrimination linéaire à marge souple

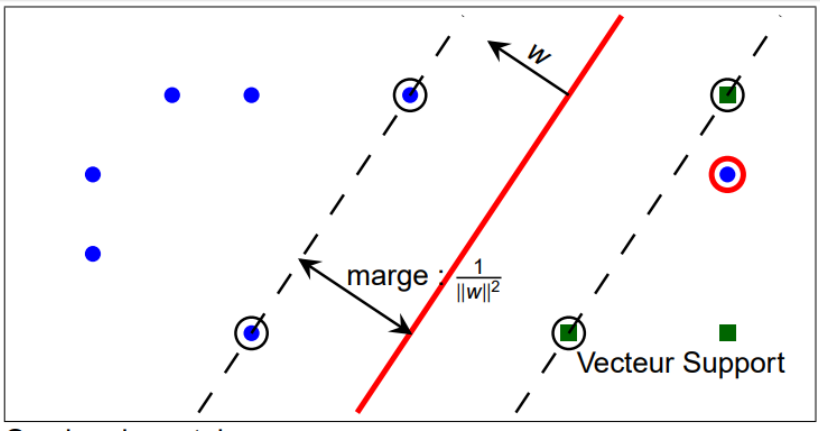
On cherche w tel que :

$$min_{w,b}\,\langle w, w\rangle + C\sum_{i=1}^{n}\xi_i, \tag{3}$$

$$\tag{4}$$

Sous les contraintres

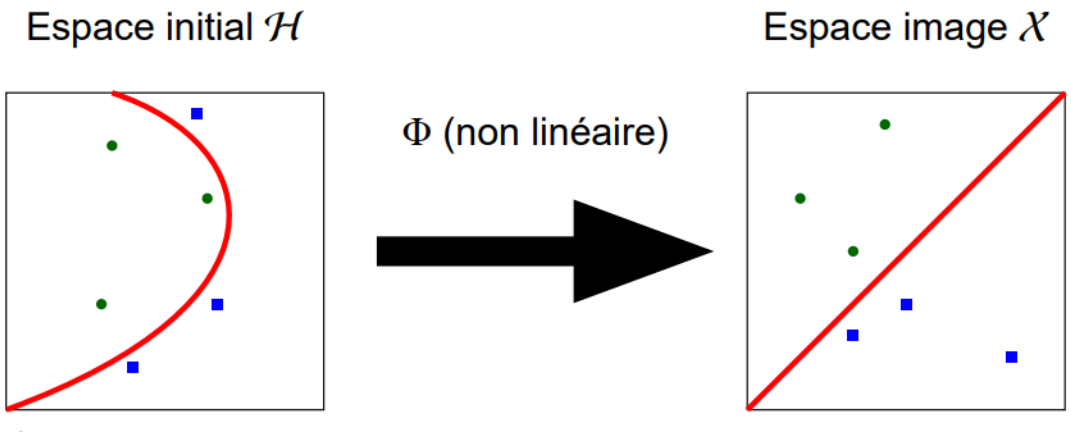
$$y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \tag{5}$$
$$o\grave{u} 1 \leqslant i \leqslant n \tag{6}$$
$$\xi_i \geq 1\,; 1 \leqslant i \leqslant n; \tag{7}$$
$$\tag{8}$$



Les erreurs de classification sont autorisées grâce aux variables xi. Le paramètre C agit comme un paramètre de régularisation inverse. Lorsque C est petit, le coût de la violation des contraintes de marge dure, c'est-à-dire le coût de l'existence de certaines variables $\xi_i > 0$ est faible et une solution avec une petite norme pour w sera choisie, même si cela conduit à de nombreuses erreurs de classification. Au contraire, lorsque C est grand, les erreurs de classification dominent.

3.3 SVM non linéaire

Abordons maintenant le problème des données non linéairement séparables. Pour rappel, des données sont non linéairement séparables quand il n'existe pas d'hyperplan capable de séparer correctement les deux catégories. Ce qui, d'ailleurs, arrive quasiment tout le temps en pratique. Certains problèmes de classification n'ont pas de solution linéaire satisfaisante mais une solution non-linéaire. Les SVM non linéaires sont obtenus en transformant les données originales.
Il est courant ne pas pouvoir séparer les données parce que l'espace est de trop petite dimension. Si l'on arrivait à transposer les données dans un espace de plus grande dimension, on arriverait peut-être à trouver un hyperplan séparateur.
Supposons que l'on donne un espace de Hilbert $\mathcal{H}$ et une fonction $\mathbf{\Phi} : \mathbb{X} \mapsto \mathcal{H}$ non linéaire.
L'idée est d'envoyer les données dans un espace de grande plus dimension. Cependant plus on se plonge dans une grande dimension, plus les calculs sont longs. D'où les difficultés lorsque l'espace est de dimension infinie, comme c'est le cas pour les FDA
Il convient de noter que ce mappage de caractéristiques permet de définir les SVM sur des espaces d'entrée presque arbitraires.



On cherche w tel que :

$$\begin{aligned} \min_{w,b,\xi} \quad & \langle w, w\rangle + C\sum_{i=1}^{N}\xi_i \\ s.t. \quad & y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \quad 1 \leq i \leq N, \\ & \xi_i \geq 0, \quad 1 \leq i \leq N. \end{aligned} \tag{P_C}$$

3.3.1 Double formulation et noyau

Les problèmes ont plusieurs formulations possibles, l'équation P_C peut aussi s'écrire comme D_C, les formulations étant équivalentes.

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle \\ s.t \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad 1 \leq i \leq N. \end{aligned} \tag{D_C}$$

Ce résultat est aussi valable pour les fonctionnelles Hilbertiennes, de telle sorte que l'on peut remplacer les x_i par les $\psi(x_i)$ pour obtenir le problème suivant :

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \langle \phi(x_i), \phi(x_j)\rangle_{\mathcal{H}} \\ s.t \quad & \sum_{i=1}^{N}\alpha_i y_i = 0, \\ & 0 \leq \alpha_i \leq C, \quad 1 \leq i \leq N. \end{aligned} \tag{$D_{C,\mathcal{H}}$}$$

Le problème d'optimisation dual s'exprime ainsi :
Minimiser

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=j}^{n} \alpha_i \alpha_j y_i y_j \langle \Psi(x_i), \Psi(x_j) \rangle \tag{9}$$

$$Sous\ les\ contraintes : \tag{10}$$

$$\sum_{i=j}^{n} \alpha_i y_j = 0; \;\; 0 \leqslant \alpha_i \leqslant C \;\; \forall i = 1, ..., n; \tag{11}$$

La solution du problème s'écrit :

$$d(x, \alpha^*, b^*) = sign(\sum_{i=j}^{n} \alpha_i^* y_i \langle \Psi(x), \Psi(x_i) \rangle + b^*) \tag{12}$$

Problèmatique : L'infinie dimensionalité des espaces fonctionnels implique qu'une transposition basique des algorithmes standard, introduit des difficultés à la fois théoriques et pratiques. Les problèmes deviennent mal posés car la covariance de X (espace de Hilbert quelconque) est un opérateur de Hilbert Schmidt et donc n'a pas d'inverse continue. L'estimation par une méthode directe de cet opérateur pose un problème car elle ne fournit pas une estimation fiable.

Approches de résolution : Pour surmonter le problème de l'infinité de dimensions, la plupart des méthodes de la FDA jusqu'à présent ont été construites grâce à deux principes généraux : le filtrage et la régularisation. Dans l'approche de filtrage, l'idée est d'utiliser des méthodes de représentation qui permettent de travailler en dimensions finies c'est-à-dire en faisant des transformations grâce à des Noyaux. Dans l'approche par régularisation, la complexité de la solution est limitée grâce à des contraintes de lissage. Dans cette approche, h est choisie parmi des candidats lisses (par exemple des fonctions deux fois dérivables avec une courbure minimale). Grâce à ces deux approches, de nombreux algorithmes d'analyse de données ont été adaptés avec succès aux données fonctionnelles. Notre objectif dans le présent article est d'étudier le cas des SVM, principalement grâce à une approche de filtrage.

3.4 SVM pour les données fonctionnelles

3.4.1 L'astuce du noyau pour les données fonctionnelles

Un autre type de transformation peut être utilisé afin de définir des noyaux adaptés. L'idée est de réduire la dimension de l'espace d'entrée, c'est-à-dire d'appliquer l'approche de filtrage standard de FDA.

- Prétraitement : $\mathbf{P} : \mathcal{H} \mapsto \mathbb{R}$

$\forall u, v \in \mathcal{H}$,Q(u, v) = K(P(u), P(v))..,

- Projections : Pour $V_D = Vect(\psi_1, ..., \psi_D)$ $P(x) = \sum_{j=1}^{D} \langle x, \psi_j \rangle \psi_j$
- Transformations fonctionnelles : P(x) = $D^q x$,.....

3.4.2 Une approche consistante : Approche par projection

- $(\psi_j)_j$ base Hilbertienne de $\mathcal{H}$: projection sur $(\psi_j)_j = 1, ..., D$;
- **Choix des paramètres** : $a = d \in \mathcal{N}, \mathcal{K} \in \mathcal{J}_d, C \in [\![0; C_d]\!]$

☐ partage des données : $\mathcal{B}_1 = (X_1, y_1), ..., (X_I, y_I)\ et\ \mathbf{B}_2 = (X_(I+1), y_{I+}1), ..., (X_n, y_n)$

☐ Construction du SVM sur $\mathcal{B}_1$: f_a ;

☐ choix du paramètre optimal sur $\mathcal{B}_2$:
$a^* = argmin\widehat{L}_(n-1)f_a + \frac{\lambda_d}{\sqrt{n-1}}$
Avec $\widehat{L}_{n-1}f_a = \frac{1}{n-1}\sum_{i=I+1}^{n} \mathcal{I}_{\{f_a(x_i)\neq(y_i)\}}$

3.4.3 Hypothèses

- Hypoyhèses sur la distribution de X :

  (H1) X prend ses valeurs dans un borné de $\mathcal{H}$
- Hypothèses sur les paramètres : $\forall\ d > 1$,
  (H2) $\mathcal{J}_d$ est un ensemble fini ;
  (H3) $\exists\ \mathcal{K}_d \in \mathcal{J}_d$ tel que $\mathcal{K}_d$ est universel et $\exists\ \nu_d > 0$ tel que $\mathcal{N}(\mathcal{K}_d, \epsilon) = O(\epsilon^{-\nu_d})$
  (H4) $\mathcal{C}_d > 1$
  (H5) $\sum_{d\geq 1} |\mathcal{J}_d| e^{-2\lambda_d^2} < \infty$

- Hypothèses sur la validation :

  (H6)

$$\lim_{n\to+\infty} I = +\infty; \tag{13}$$

  (H7)

$$\lim_{n\to+\infty} n - I = +\infty; \tag{14}$$

  (H8)

$$\lim_{n\to+\infty} \frac{Ilog(n-I)}{n-I} = 0; \tag{15}$$

4 Convergence par procédure de validation

Sous les hypothèses **(H1)**.....**(H8)**, f_n est consistant

$$Lf_n \underset{n\to+\infty}{\longrightarrow} L^*, \tag{16}$$

Où $Lf_n = \mathbb{P}(f_n(X) \neq Y)$ et $L^* = \mathbb{P}(f^*(X) \neq Y)$ avec

$$f^*(x) = \begin{cases} 1 & \text{si } \mathbb{P}(Y = 1|X = x) > 1/2, \\ -1 & \text{sinon.} \end{cases} \tag{17}$$

4.1 cas des FDA :

Il faut d'abord noter que dans les espaces de Hilbert de dimension infinie, le problème de marge dure PC a toujours une solution lorsque les données d'entrée sont en position générale, c'est-à-dire lorsque N observations couvrent un sous-espace de dimension N de X. Une solution très naïve consisterait donc à éviter les marges douces et les noyaux non linéaires. Cela ne donnerait pas des résultats très intéressants en pratique en raison du manque de régularisation. De plus, le SVM Linéaire avec une faible marge a de mauvaises performances. On peut noter cela en écrivant Pc sous la forme d'un problème d'optimisation sans contraintes et remarquer l'aspect régularisation et la proximité avec la Régression Ridge. La pénalisation dans Ridge a de mauvais résultats sur les données fonctionnelles (celles-ci ayant un nombre de points discrets très grand, l'approximation de la pénalité de Ridge devra se faire avec un produit matriciel avec des matrices de très grande dimension). C'est pour ces raisons qu'il est intéressant de considérer un SVM non linéaire sur des données fonctionnelles en introduisant les bons noyaux.

L'utilisation d'un noyau correspond à la fois à remplacer un classificateur linéaire par un classificateur non linéaire, mais aussi à remplacer la pénalisation par une pénalisation induite par le noyau, qui pourrait être plus adaptée au problème.

5 Application

5.1 FDA en pratique :

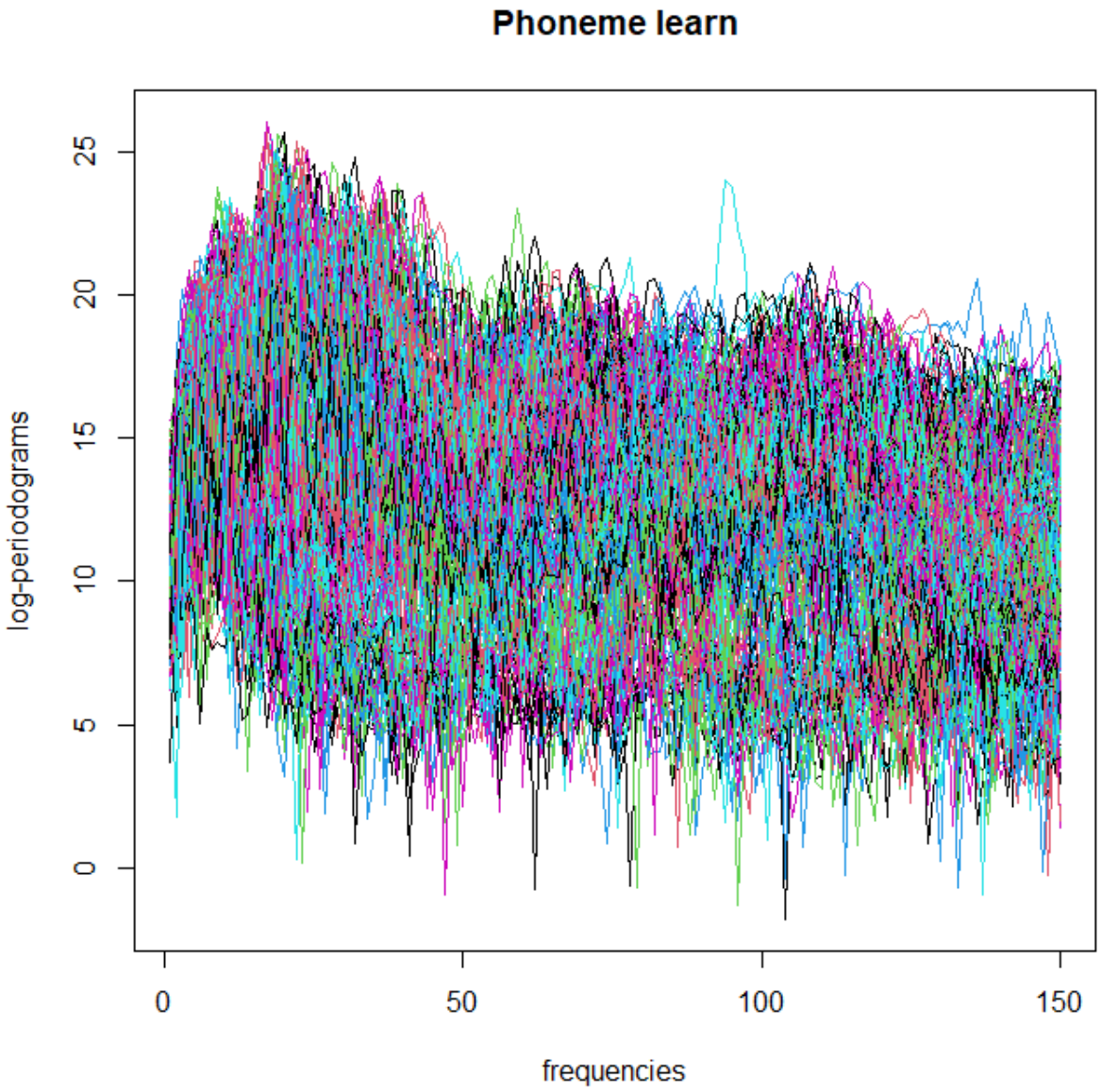
En pratique, les fonctions n'étant jamais vraiment connues, il est difficile d'appliquer les méthodes décrites plus haut. La meilleure situation correspond à celle où la d-discrétisation des points a été choisie dans |R et que les fonctions xi sont décrites par un vecteur de |Rd. Dans ce cas, une solution simple consiste à supposer que les opérations standards dans Rd (combinaisons linéaires, produit interne et norme) sont de bonnes approximations de leurs homologues dans l'espace fonctionnel considéré. Dans certains domaines comme le médical, cela ne marche pas. Une solution possible dans ce contexte consiste à construire une approximation des fonctions basée sur ses valeurs d'observation (grâce, par exemple, aux B-splines) et à travailler ensuite avec les fonctions reconstruites.

5.2 Description du jeu de données

Dans ce projet, nous utiliserons un ensemble de données standard appelé « phoneme ». C'est un exemple de données foctionnelles dans R que l'on peut trouver dans le package **fda.usc**. Il est composé d'un ensemble de fonctions décrites par des matrices de 250 lignes et 150 colonnes. Cela correspond à la situation décrite plus haut où les fonctions sont approcimées par des vecteurs de dimensions 250 (chaque colonne correspondant à une fonction discrétisée en 250 points).

5.2.1 Visualisation des données

Nous pouvons visualiser les données d'apprentissage du dataset "phoneme" dont le résultat est le suivant :



Quelques informations sur le Dataset La plupart des systèmes de reconnaissance de la parole déjà existants sont des systèmes globaux (typiquement des modèles de Markov cachés et des réseaux de neurones à retardement) qui reconnaissent les signaux et n'utilisent pas vraiment les spécificités de la parole. Au contraire, les systèmes analytiques prennent en compte le processus articulatoire menant aux différents phonèmes d'une langue donnée, l'idée étant de déduire la présence de chacune des caractéristiques phonétiques à partir de l'observation acoustique.

La principale difficulté des systèmes analytiques est d'obtenir des paramètres acoustiques suffisamment fiables. Ces mesures acoustiques doivent :

contenir toutes les informations relatives à la caractéristique phonétique concernée. être indépendantes du locuteur. être indépendantes du contexte. être plus ou moins robustes au bruit. L'observation acoustique primaire est toujours volumineuse (spectre x N moments d'observation différents) et la classification ne peut être traitée directement.

L'objectif de la présente base de données est de distinguer les voyelles nasales des voyelles orales. Il existe donc deux classes différentes : Nasales et Orales.

Ce volet ne sera ici, pas traitées. La base de données disponible sur R, qui est une version modifiée de la vraie base du European ESPRIT 5516 project : ROARS. Nous nous contenteront d'apprendre au modèle reconnaître (classifier) les voyelles, au nombre de 5 et de savoir les prédires.

5.2.2 Application et Résultat :

Dans le package fda.usc, nous utilisons le code suivant sous R, pour importer et nommer les variables que l'on utilisera dans les méthodes qui nous intéressent.

```
4  data(phoneme)
5  mlearn<-phoneme[["learn"]]
6  glearn<-phoneme[["classlearn"]]
7  mtest<-phoneme[["test"]]
8  gtest<-phoneme[["classtest"]]
9
10 dataf<-data.frame(glearn)
11 dat=list("df"=dataf,"x"=mlearn)
12 newdat<-list("x"=mtest)
```

Nous nous proposons de tester plusieurs méthodes sur les donées fonctionnelles et de comparer leurs performances. Cette comparaison se fera grâce notamment à leurs matrices de confusion respectives.

- **Exemple 1 : KNN sur les données fonctionnelles de phoneme**

```
38
39 #KNN
40
41 model_knn <-classif.knn(glearn,mlearn)
42 summary(model_knn)
43 predictions_knn <-predict(model_knn,mtest,type="class")
44 table(gtest,predictions_knn)
45 sum(predictions_knn==gtest)/250
46
```

```
-Probability of correct classification by group (prob.classification):
y
   1    2    3    4    5
1.00 0.98 0.98 0.80 0.80

-Confusion matrix between the theoretical groups (by rows)
  and estimated groups (by column)

    1  2  3  4  5
  1 50  0  0  0  0
  2  0 49  1  0  0
  3  1  0 49  0  0
  4  0  0  0 40 10
  5  0  0  1  9 40

-Vector of probability of correct classification
   by number of neighbors (knn):
    3     5     7     9    11    13    15
0.884 0.912 0.904 0.892 0.892 0.896 0.892

-Optimal number of neighbors: knn.opt= 5
with highest probability of correct classification max.prob=  0.912
```

- **Exemple 2 : SVM sur les données fonctionnelles de phoneme**

```
47
48  #SVM
49
50  model_svm <-classif.svm(glearn~x,data=dat)
51  summary(model_svm)
52  predictions_svm <-predict(model_svm,newdat,type="class")
53  table(gtest,predictions_svm)
54  sum(predictions_svm==gtest)/250
55
```

```
> summary(model_svm)
     - SUMMARY -

-Probability of correct classification by group (prob.classification):
   1    2    3    4    5
0.98 1.00 0.96 0.80 0.78

-Confusion matrix between the theoretical groups (by rows)
  and estimated groups (by column)

    1  2  3  4  5
  1 49  1  0  0  0
  2  0 50  0  0  0
  3  0  2 48  0  0
  4  0  0  0 40 10
  5  0  0  0 11 39

-Probability of correct classification:  0.904

> sum(predictions_svm==gtest)/250
[1] 0.904
>
```

- **Variante : KSVM sur les données fonctionnelles de phoneme**

```
55
56  #KSVM
57
58  model_ksvm <-classif.ksvm(glearn~x,data=dat)
59  summary(model_ksvm)
60  predictions_ksvm <-predict(model_ksvm,newdat,type="class")
61  table(gtest,predictions_ksvm)
62  sum(predictions_ksvm==gtest)/250
63
```

```
> summary(model_ksvm)
     - SUMMARY -

-Probability of correct classification by group (prob.classification):
1 2 3 4 5
1 1 1 1 1

-Confusion matrix between the theoretical groups (by rows)
  and estimated groups (by column)

    1  2  3  4  5
  1 50  0  0  0  0
  2  0 50  0  0  0
  3  0  0 50  0  0
  4  0  0  0 50  0
  5  0  0  0  0 50

-Probability of correct classification:  1

> sum(predictions_ksvm==gtest)/250
[1] 0.908
>
```

Notons ici la différence entre les probabilité de bonne classification pour chaque classe entre svm et ksvm. Ce dernier présente une probabilité de faire de bons classements meilleurs que celui de svm. Cependant la soumission du modèle aux données tests est moins concluant notamment sur les deux dernières classes, et prend plus de temps à s'exécuter que la méthode classif.svm.

Exemple 3 : Kernels et illustration de l'impact du choix des kernels

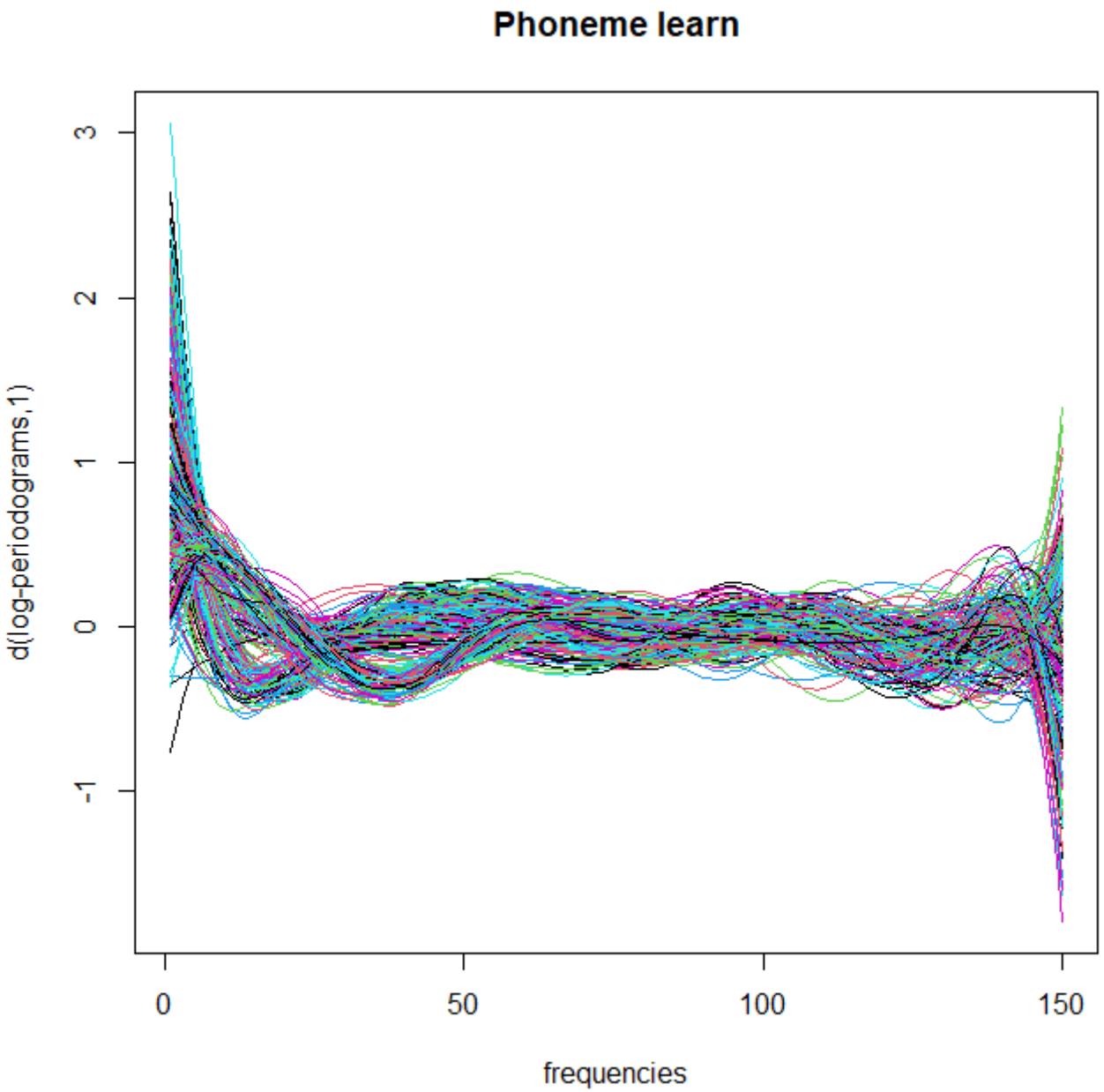
```
#KERNEL
#Ker.epa (0.912) > Ker.cos = Ker.quar = Kern.unif (0.908) > Ker.norm (0.904)

model_kernel <-classif.kernel(glearn, mlearn, Ker = Ker.epa)
predictions_kernel <-predict(model_kernel,mtest,type="class")
table(gtest,predictions_kernel)
summary(model_kernel)
sum(predictions_kernel==gtest)/250
```

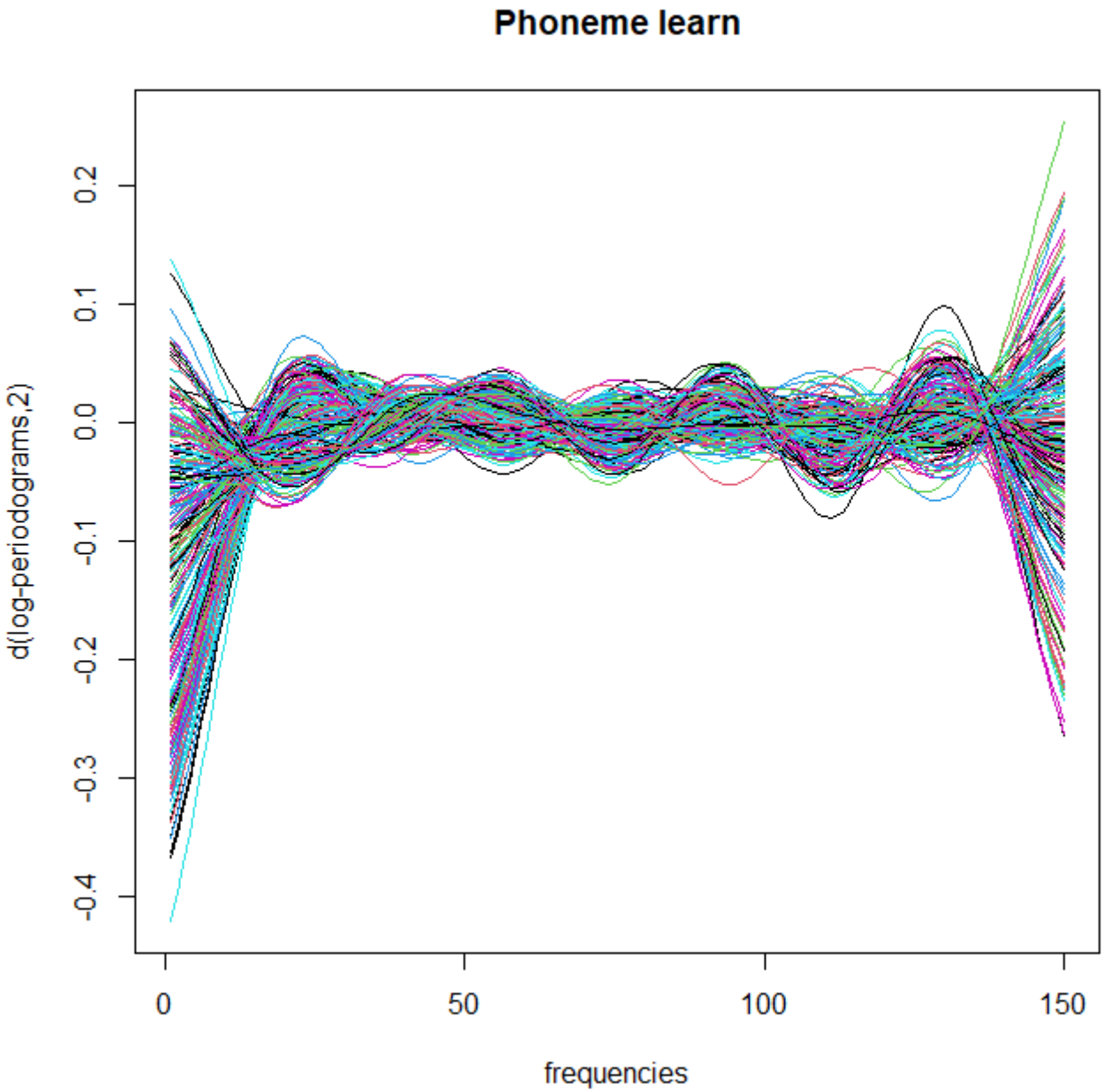
Nous obtenons les résultats suivant à la phase de test (pas pris la probabilité de faire des classifications correctes) pour quelques noyaux : Ker.epa (0.912) > Ker.cos = Ker.quar = Kern.unif (0.908) > Ker.norm (0.904) Ceci permet aussi d'illustrer l'importance du choix des noyaux. Dans le cas des données du phoneme, les performances des kernels sont néanmoins toutes bonnes et ne se différencient que très légèrement. Cependant, ces dernières peuvent être très éloignées comme l'illustre les auteurs de l'article, en fonction de la nature des données.

```
-Optimal bandwidth: h.opt= 36.94332 with highest probability of
correct classification: max.prob= 0.9

-Probability of correct classification:  0.9

> sum(predictions_kernel==gtest)/250
[1] 0.912
>
```

De plus on notera que les performances des toutes les méthodes utilisées sont supérieures à 0.9. Ces méthodes de classification prenant en compte les données fonctionnelles sont toutes très bonnes et rejoignent l'auteur sur l'utlité de la prise en compte de la nature de ces dernières.

- **Prise en compte des dérivées :** Nous allons inclure les dérivées des fonctions pour tenter d'améliorer les résultats en exploitant au mieux le caractère fonctionnel des données. On obtient les dérivées en utilisant la fonction fdata.deriv(). Voici les dérivées première et seconde de mlearn :


Phoneme learn
d(log-periodograms,1)
3
2
1
0
-1
0
50
100
150
frequencies



Les 3 premières dérivées seront utilisées pour améliorer nos résultats :

```
mlearnd1 = fdata.deriv(mlearn,nderiv=1,method="bspline",class.out='fdata',nbasis= 11)
mtestd1 = fdata.deriv(mtest,nderiv=1,method="bspline",class.out='fdata',nbasis=11)

mlearnd2 = fdata.deriv(mlearn,nderiv=2,method="bspline",class.out='fdata',nbasis=11)
mtestd2 = fdata.deriv(mtest,nderiv=2,method="bspline",class.out='fdata',nbasis=11)

mlearnd3 = fdata.deriv(mlearn,nderiv=3,method="bspline",class.out='fdata',nbasis=11)
mtestd3 = fdata.deriv(mtest,nderiv=3,method="bspline",class.out='fdata',nbasis=11)

dataf<-data.frame(glearn)
dat=list("df"=dataf,"x"=mlearn+mlearnd1+mlearnd2+mlearnd3)

moderivsvm<-classif.svm(glearn~x,data=dat)
summary(moderivsvm)

newdat<-list("df" = dataf, "x"= mtest+mtestd1+mtestd2+mtestd3)
predictions<-predict(moderivsvm,newdat,type="class")
table(gtest,predictions)
sum(predictions==gtest)/250
```

Voici les résultats obtenus :

```
-Probability of correct classification by group (prob.classification):
   1    2    3    4    5
1.00 1.00 0.96 0.80 0.78

-Confusion matrix between the theoretical groups (by rows)
  and estimated groups (by column)

    1  2  3  4  5
  1 50  0  0  0  0
  2  0 50  0  0  0
  3  0  2 48  0  0
  4  0  0  0 40 10
  5  0  0  0 11 39

-Probability of correct classification:  0.908

>
> newdat<-list("df" = dataf, "x"= mtest+mtestd1+mtestd2+mtestd3)
> predictions<-predict(moderivsvm,newdat,type="class")
> table(gtest,predictions)
     predictions
gtest  1  2  3  4  5
    1 50  0  0  0  0
    2  0 50  0  0  0
    3  0  0 49  0  1
    4  0  0  0 42  8
    5  0  1  0 12 37
> sum(predictions==gtest)/250
[1] 0.912
```

On observe une amélioration très légère des résultats par rapport au classif.svm appliqué sur mlearn. L'absence d'amélioration très significative des prédications, pourrait s'expliquer par le fait que l'application directe de clasif.svm sur les données fournissait déjà de bons résultats et que les dérivées, sur ce jeu données particulier, n'apportent pas une différenciation supplémentaire suffisante pour améliorer significativement les prédictions. Sur un jeu de données différents où la méthode classif.svm appliquée directement n'est pas suffisament performante, rajouter les dérivées pourrait améliorer la performance significativement.

6 Conclusion et ouvertures

Nous avons introduit une méthode d'utilisation des SVM pour des données de type fonctionnel. L'avantage de cette approche est qu'elle permet d'éffectuer ce pré-traitement de manière transparente en utilisant une simple perturbation du noyau d'origine sur les discrétisations des fonctions. Un résultat de consistance en découle obtenu à partir des résultats de consistance existant en dimension finie. Ce résultat pourrait être étendu, moyennant quelques adaptations, par une approche de lissage qui autoriserait une interpolation imparfaite des données et donc la prise en compte d'un éventuel bruit sur les mesures ; cette nouvelle approche entraînerait néanmoins la détermination d'un paramètre de régularisation (le paramètre de lissage des splines) supplémentaire. L'utilisation d'un noyau correspond à la fois à remplacer un classificateur linéaire par un classificateur non linéaire, mais aussi à remplacer la pénalisation par une pénalisation induite par le noyau, qui pourrait être plus adaptée au problème. Il convient néanmoins de bien choisir le noyau en question selon la nature des données.Il aussi possible d'améliorer les prédictions en prenant encompte les dérivées des données fonctionnelles d'où l'intérêt de la considération du caractère fonctionnelle de ces données.