A speech recognition problem: discriminating log-periodograms

- **Statistical aim**

  We recall that we observe $n = 2000$ pairs $(\boldsymbol{x}_i, y_i)_{i=1,\dots,n}$ where the $\boldsymbol{x}_i$'s correspond to discretized log-periodograms ($\boldsymbol{x}_i = (\chi(f_1), \chi(f_2), \dots, \chi(f_{150}))$ is the ith discretized functional data) whereas the y_i's give the associated class membership (five phonemes). The file "npfda-phoneme.dat" contains the pairs $(\boldsymbol{x}_i, y_i)_{i=1,\dots,215}$. Given a new log-periodogram $\boldsymbol{x}$, our main task is to predict the corresponding class of phoneme y^{LCV}.

- **Measuring performance**

  For measuring the performance of our functional nonparametric discrimination method, we build two samples from the original dataset. The first one, the learning sample, contains the 5×50 units ($(\boldsymbol{x}_i, y_i)_{i\in\mathcal{L}}$, each group containing 50 observations). The second one is the testing sample and contains 5×50 units ($(\boldsymbol{x}_{i'}, y_{i'})_{\in i'\mathcal{T}}$ with 50 observations by group). The learning sample allows to estimate the posterior probabilities with optimal smoothing parameter ; both the $\boldsymbol{x}_i$'s and the corresponding y_i's are used at this stage. The testing sample is useful for measuring the discriminant power of such a method; we evaluate the posterior probabilities (obtained with the learning sample) at $\{\boldsymbol{x}_{i'}\}_{i'\in\mathcal{T}}$ ($\{y_{i'}\}_{i'\in\mathcal{T}}$ being ignored) which allows us to get the predicted class membership $\{y_{i'}^{LCV}\}_{i'\in\mathcal{T}}$. It remains to compute the misclassification rate

  $$Misclas_{Test} \longleftarrow \frac{1}{250}\sum_{i'\in\mathcal{T}} 1_{[y_{i'} \neq y_{i'}^{LCV}]}.$$

  We repeat 50 times this procedure by building randomly 50 learning samples $\mathcal{L}_1, \dots, \mathcal{L}_{50}$ and 50 testing samples $\mathcal{T}_1, \dots, \mathcal{T}_{50}$. Finally, we perform 50 misclassification rates $Misclas_1, \dots, Misclas_{50}$ and the distribution of these quantities gives a good idea on the discriminant power of such a functional nonparametric supervised classification. This procedure is entirely repeated, by running the routine `funopadi.knn.lcv`, for various semi-metrics in order to highlight the importance of such a proximity measure:

  - pca-type semi-metrics (routine semimetric.pca) with a number of dimension taking its values in 4, 5, 6, 7 and 8 successively,

- pls-type semi-metrics (routine semimetric.mplsr) with a number of factors taking its values in 5, 6, 7, 8 and 9 successively,
- derivate-type semi-metrics (routine semimetric.deriv) with a number of derivatives equals to zero (classical L_2 norm; the results obtained with a larger number of derivatives are worse).

Remark: the commandlines for R or S+ are the same.

- **Entering phoneme data**

```
PHONDAT <- as.matrix(read.table("npfda-phoneme.dat"))
attributes(PHONDAT)$dimnames[[1]] <- character(0)
PHONCURVES <- PHONDAT[,1:150]          # sample of curves
Learn.sh <- sample(1:400,50)
Learn.iy <- sample(401:800,50)
Learn.dcl <- sample(801:1200,50)
Learn.aa <- sample(1201:1600,50)
Learn.ao <- sample(1601:2000,50)
Test.sh <- sample((1:400)[-Learn.sh],50)
ind <- (1:800)[-Learn.iy]
Test.iy <- sample(ind[ind>401],50)
ind <- (1:1200)[-Learn.dcl]
Test.dcl <- sample(ind[ind>801],50)
ind <- (1:1600)[-Learn.aa]
Test.aa <- sample(ind[ind>1201],50)
ind <- (1:2000)[-Learn.ao]
Test.ao <- sample(ind[ind>1601],50)
Learning <- c(Learn.sh,Learn.iy,Learn.dcl,Learn.aa,Learn.ao)
Testing <- c(Test.sh,Test.iy,Test.dcl,Test.aa,Test.ao)
PHONLEARN <- PHONCURVES[Learning,]     # learning sample of curves
PHONTEST <- PHONCURVES[Testing,]       # testing sample of curves
Classlearn <- sort(rep(1:5,50))        # learning class numbers
Classtest <-  sort(rep(1:5,50))        # testing class numbers
```

- **Computing predicted class membership and misclassification rates** (for various semi-metrics)

```
res.mplsr5 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,5,
```

```
              kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.mplsr5 <- sum(res.mplsr5$Predicted.classnumber !=
                           Classtest)/250
res.mplsr6 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,6,
              kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.mplsr6 <- sum(res.mplsr6$Predicted.classnumber !=
                           Classtest)/250
res.mplsr7 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,7,
              kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.mplsr7 <- sum(res.mplsr7$Predicted.classnumber !=
                           Classtest)/250
res.mplsr8 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,8,
              kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.mplsr8 <- sum(res.mplsr8$Predicted.classnumber !=
                           Classtest)/250
res.mplsr9 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,9,
              kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.mplsr9 <- sum(res.mplsr9$Predicted.classnumber !=
                           Classtest)/250
res.pca4 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,4,
            kind.of.kernel = "quadratic",semimetric="pca")
Misclas.pca4 <- sum(res.pca4$Predicted.classnumber !=
                         Classtest)/250
res.pca5 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,5,
            kind.of.kernel = "quadratic",semimetric="pca")
Misclas.pca5 <- sum(res.pca5$Predicted.classnumber !=
                         Classtest)/250
res.pca6 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,6,
            kind.of.kernel = "quadratic",semimetric="pca")
Misclas.pca6 <- sum(res.pca6$Predicted.classnumber !=
                         Classtest)/250
res.pca7 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,7,
            kind.of.kernel = "quadratic",semimetric="pca")
Misclas.pca7 <- sum(res.pca7$Predicted.classnumber !=
                         Classtest)/250
res.pca8 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,8,
            kind.of.kernel = "quadratic",semimetric="pca")
Misclas.pca8 <- sum(res.pca8$Predicted.classnumber !=
```

```
                        Classtest)/250
res.deriv0 <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,0,20,c(0,1),
              kind.of.kernel = "quadratic",semimetric="deriv")
Misclas.deriv0 <- sum(res.deriv0$Predicted.classnumber !=
                        Classtest)/250
```

- **Plotting misclassification rates over 1 run**
  The following commandlines allow to obtain Figure 1 (don't forget to active your graphics device!):

```
Misclas.rates <- c(Misclas.mplsr5,Misclas.mplsr6,Misclas.mplsr7,
                   Misclas.mplsr8,Misclas.mplsr9,Misclas.pca4,
                   Misclas.pca5,Misclas.pca6,Misclas.pca7,Misclas.pca8,
                   Misclas.deriv0)
Misclas.names <- c("mplsr5","mplsr6","mplsr7","mplsr8","mplsr9","pca4",
                   "pca5","pca6","pca7","pca8","deriv0")
%par(mai=c(0.8,.1,0.1,0.1))
dotchart(Misclas.rates, Misclas.names, cex=1, xlab="MISCLASSIFICATION RATES")
```

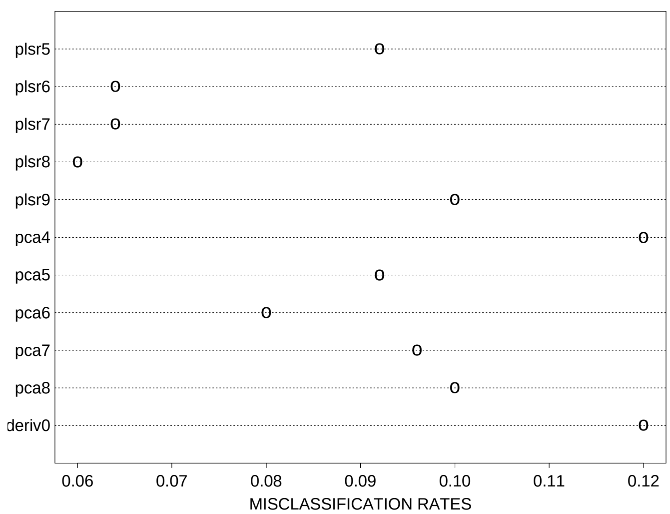

Figure 1: Results for only one run

- **Computing and plotting misclassification rates over 50 runs**
  We repeat 50 times the previous commandlines by means of a loop:

```
Misclas.of.phon.over.50.samples.with.pca4 <- 0
Misclas.of.phon.over.50.samples.with.pca5 <- 0
Misclas.of.phon.over.50.samples.with.pca6 <- 0
Misclas.of.phon.over.50.samples.with.pca7 <- 0
Misclas.of.phon.over.50.samples.with.pca8 <- 0
Misclas.of.phon.over.50.samples.with.mplsr5 <- 0
Misclas.of.phon.over.50.samples.with.mplsr6 <- 0
Misclas.of.phon.over.50.samples.with.mplsr7 <- 0
Misclas.of.phon.over.50.samples.with.mplsr8 <- 0
Misclas.of.phon.over.50.samples.with.mplsr9 <- 0
Misclas.of.phon.over.50.samples.with.deriv0 <- 0
for(i in 1:50){
        set.seed(sample(0:1000,1))
        Learn.sh <- sample(1:400,50)
        Learn.iy <- sample(401:800,50)
        Learn.dcl <- sample(801:1200,50)
        Learn.aa <- sample(1201:1600,50)
        Learn.ao <- sample(1601:2000,50)
        Test.sh <- sample((1:400)[-Learn.sh],50)
        ind <- (1:800)[-Learn.iy]
        Test.iy <- sample(ind[ind>401],50)
        ind <- (1:1200)[-Learn.dcl]
        Test.dcl <- sample(ind[ind>801],50)
        ind <- (1:1600)[-Learn.aa]
        Test.aa <- sample(ind[ind>1201],50)
        ind <- (1:2000)[-Learn.ao]
        Test.ao <- sample(ind[ind>1601],50)
        Learning <- c(Learn.sh,Learn.iy,Learn.dcl,Learn.aa,Learn.ao)
        Testing <- c(Test.sh,Test.iy,Test.dcl,Test.aa,Test.ao)
        PHONLEARN <- PHONCURVES[Learning,]
        PHONTEST <- PHONCURVES[Testing,]
        res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,4,
          kind.of.kernel = "quadratic",semimetric="pca")
        Misclas.of.phon.over.50.samples.with.pca4[i] <-
          sum(res$Predicted.classnumber != Classtest)/length(Classtest)
```

```
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,5,
  kind.of.kernel = "quadratic",semimetric="pca")
Misclas.of.phon.over.50.samples.with.pca5[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,6,
  kind.of.kernel = "quadratic",semimetric="pca")
Misclas.of.phon.over.50.samples.with.pca6[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,7,
  kind.of.kernel = "quadratic",semimetric="pca")
Misclas.of.phon.over.50.samples.with.pca7[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,8,
  kind.of.kernel = "quadratic",semimetric="pca")
Misclas.of.phon.over.50.samples.with.pca8[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,5,
  kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.of.phon.over.50.samples.with.mplsr5[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,6,
  kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.of.phon.over.50.samples.with.mplsr6[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,7,
  kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.of.phon.over.50.samples.with.mplsr7[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,8,
  kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.of.phon.over.50.samples.with.mplsr8[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,9,
  kind.of.kernel = "quadratic",semimetric="mplsr")
Misclas.of.phon.over.50.samples.with.mplsr9[i] <-
  sum(res$Predicted.classnumber != Classtest)/length(Classtest)
res <- funopadi.knn.lcv(Classlearn,PHONLEARN,PHONTEST,0,30,c(0,1),
  kind.of.kernel = "quadratic",semimetric="deriv")
```

```
        Misclas.of.phon.over.50.samples.with.deriv0[i] <-
          sum(res$Predicted.classnumber != Classtest)/length(Classtest)
}
Misclas.mplsr5 <- Misclas.of.phon.over.50.samples.with.mplsr5
Misclas.mplsr6 <- Misclas.of.phon.over.50.samples.with.mplsr6
Misclas.mplsr7 <- Misclas.of.phon.over.50.samples.with.mplsr7
Misclas.mplsr8 <- Misclas.of.phon.over.50.samples.with.mplsr8
Misclas.mplsr9 <- Misclas.of.phon.over.50.samples.with.mplsr9
Misclas.pca4 <- Misclas.of.phon.over.50.samples.with.pca4
Misclas.pca5 <- Misclas.of.phon.over.50.samples.with.pca5
Misclas.pca6 <- Misclas.of.phon.over.50.samples.with.pca6
Misclas.pca7 <- Misclas.of.phon.over.50.samples.with.pca7
Misclas.pca8 <- Misclas.of.phon.over.50.samples.with.pca8
Misclas.deriv0 <- Misclas.of.phon.over.50.samples.with.deriv0
```

Now, `Misclas.mplsr5`, `Misclas.mplsr6`,...,`Misclas.deriv0` are vectors of length 50. The following commandlines allow to obtain Figure 2:

```
Misclas.names <- c("plsr5","plsr6","plsr7","plsr8","plsr9","pca4",
  "pca5","pca6","pca7","pca8","deriv0")
boxplot(Misclas.mplsr5, Misclas.mplsr6,Misclas.mplsr7,Misclas.mplsr8,
  Misclas.mplsr9,Misclas.pca4,Misclas.pca5,Misclas.pca6,Misclas.pca7,
  Misclas.pca8,Misclas.deriv0, names=Misclas.names, xlab="SEMI-METRICS",
  ylab="MISCLSSIFICATION RATES", cex=0.8)
```

It is clear that the semi-metric based on the multivariate PLS regression allows to obtain a good discrimination.

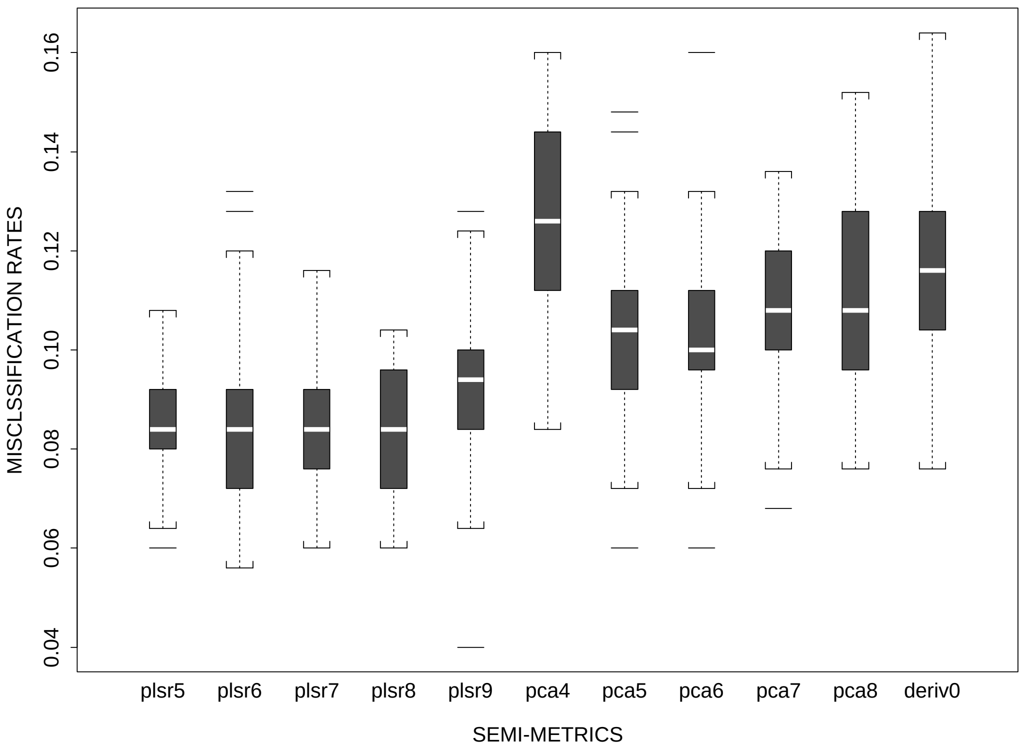

Figure 2: Results over 50 runs