CloHe: Clustering and Classification of Multidimensional Functional Data with Missing Values

Serge Iovleff
University Lille 1

Abstract

This vignette describe shortly how to use the package CloHe.

Keywords: Functional Data, STK++, rtkore, Classification, Clustering, Missing Values.

---

1. Formosat data description

The package CloHe is able to read the Formosat data set (located in the data folder of the package) using the function `readFiles`. There is however a subset of the Formosat data set distributed with the package that can be used directly.

```
> #formosat <- readFiles(path="~/Developpement/workspace/CloHe/data/Formosat/");
> data(formosat)
> names(formosat)

[1] "labels" "times"  "xb"     "xg"     "xr"     "xi"     "clouds"

> length(formosat$labels)

[1] 1029

> length(formosat$times)

[1] 96

> dim(formosat$xb) # we get same result with xg,xr,xi,clouds

[1] 1029   96
```

This data set contains the years 2008 to 2014 and the observations 15 and 65 of the original data set are removed. It contains 1029 multidimensional times series (dimension 4) and 96 dates. The vector `formosat$labels` contains the class number of each observations. There is 13 classes in this data set.

```
> # levels of the labels in integer format
> as.integer(levels(factor(formosat$labels)))

 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13
```

The matrices `formosat$xb`, `formosat$xg`, `formosat$xr`, `formosat$xi` contain the spectra values (blue, green, red, infrared). The matrix `formosat$clouds` contains an integer indicating the presence of clouds, shadows,... If there is no clouds the value is 0.

2. GaussianMutSigmat Model description

Let us denotes by n the number of times series (in the Formosat data set $n = 1029$), by T the number of dates (in the Formosat data set $T = 96$) and by K the number of class (in the Formosat data set $K = 13$). For each class $n_k,\ k = 1, \dots K$ denote the number of sample, so that $n_1 + \dots + n_k = n$.

For the Formosat data set we have the following counts

```
> table(factor(formosat$labels))

  1   2   3   4   5   6   7   8   9  10  11  12  13
 85 115 145  60  75  80 142  47  60  55  44  75  46
```

We denote by $\mathbf{X}_k = (\mathbf{x}_{ki},\ i = 1, \dots n_k)$ the observations in the class k. The ith sample $\mathbf{x}_{ki}$ is a multidimensional times series of length T. We denote

$$\mathbf{x}_{kit} = \begin{pmatrix} x^b_{kit} \\ x^g_{kit} \\ x^r_{kit} \\ x^{ir}_{kit} \end{pmatrix}, \quad t = 1, \dots, T, \quad i = 1, \dots, n_k, \quad k = 1, \dots, K$$

The package CloHe proposes to estimate only one model denominated `GaussianMutSigmat`. This model assumes that all the vectors are independents and

$$\mathbf{x}_{kit} \sim \mathcal{N}\left(\boldsymbol{\mu}_{kt};\ \boldsymbol{\Sigma}_{kt}\right)$$

where $\boldsymbol{\mu}_{kt}$ denotes a vector of size 4, and $\boldsymbol{\Sigma}_{kt}$ a variance matrix of size 4×4. If for some (i, t) there is clouds, the values of the vector $\mathbf{x}_{kit}$ are assumed as missing.

In the Formosat data set there is 2561 missing values (over a total of 98784)

```
> c(sum(formosat$clouds != 0),sum(formosat$clouds == 0))

[1]  2561 96223
```

Let $\mathbf{X} = (\mathbf{x}_t,\ t = 1, \dots, T)$ be a new times series, the classification rules for this observation will be

$$\hat{k} = \arg\max_{k=1}^{K} \sum_{t=1}^{T} -\frac{1}{2}\left((\mathbf{x}_t - \boldsymbol{\mu}_{kt})' \boldsymbol{\Sigma}_{kt}^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_{kt}) + \log(|\boldsymbol{\Sigma}_{kt}|)\right)$$

3. GaussianMutSigmat model estimation

A `GaussianMutSigmat` model is estimated using the `learnGaussian` function.

```
> res <- learnGaussian(formosat)
> names(res)

[1] "predict" "models"
```

This function return two results : the predicted class for the observations in the Formosat data set and the model parameters.

For the predicted values we get the following results

```
> buildConfusionMatrix(formosat$labels, res$predict)

    P1  P2  P3 P4 P5 P6  P7 P8 P9 P10 P11 P12 P13
T1  85   0   0  0  0  0   0  0  0   0   0   0   0
T2   0 115   0  0  0  0   0  0  0   0   0   0   0
T3   0   3 142  0  0  0   0  0  0   0   0   0   0
T4   0   1   0 55  0  0   0  2  0   0   0   2   0
T5   0   0   0  0 75  0   0  0  0   0   0   0   0
T6   0   4   0  0  0 76   0  0  0   0   0   0   0
T7   0   9   0  0  0  0 133  0  0   0   0   0   0
T8   0   0   0  0  0  0   0 47  0   0   0   0   0
T9   0   2   0  0  0  0   0  0 57   1   0   0   0
T10  0   1   0  0  0  0   0  0  0  54   0   0   0
T11  0   1   0  0  0  0   0  0  0   0  43   0   0
T12  0   1   0  1  0  0   0  0  0   0   0  73   0
T13  0   0   0  0  0  0   0  0  0   0   0   0  46
```

```
> sum(formosat$labels == res$predict)/length(formosat$labels)

[1] 0.9727891
```

The confusion matrix shows the data are well classified. The rate is overestimated as we are classifying the data used in order to estimate the model parameters.

For the model parameters, we get a list of `S4` classes storing the parameters of size K (13 for the Formosat data set).

```
> getSlots("GaussianMutSigmatModel")

           mut          sigmat              xb              xg              xr
        "list"          "list"        "matrix"        "matrix"        "matrix"
            xi            mask              td     classNumber    lnLikelihood
      "matrix"        "matrix"       "numeric"       "numeric"       "numeric"
     criterion nbFreeParameter
     "numeric"       "numeric"
```

```
> ## res$models[[1]] # show (a part of) the members of the class
```

The class is encapsulating the observations, the mask (the presence of clouds), the times samples (in days, not used by this model), the log-Likelihood of the model, the number of free parameters of the model and two lists `mut` and `sigmat` with, for each dates, the estimated mean and estimated variance matrix. The field criterion is not used.

The values of the parameters can be obtained in a matrix using this kind of code (only the parameters corresponding of the 2 first dates of the first class are displayed)

```
> matrix(unlist(res$models[[1]]@mut), nrow=4, byrow=F)[,1:2]
```

```
          [,1]       [,2]
[1,]  34.70588   5.658824
[2,]  48.85882  21.658824
[3,]  63.24706  15.623529
[4,] 171.60000 352.564706
```

```
> matrix(unlist(res$models[[1]]@sigmat), nrow=4, byrow=F)[,1:(2*4)]
```

```
          [,1]        [,2]     [,3]        [,4]       [,5]       [,6]
[1,] 10.772318  3.77024221 2.907958  1.64705882   2.601246   1.942422
[2,]  3.770242  6.09771626 2.423114 -0.06823529   1.942422   5.165952
[3,]  2.907958  2.42311419 7.880138  1.95764706   1.695087   3.024498
[4,]  1.647059 -0.06823529 1.957647 65.84000000 -11.630865 -17.948512
           [,7]      [,8]
[1,]   1.695087 -11.63087
[2,]   3.024498 -17.94851
[3,]   4.540623 -20.79917
[4,] -20.799170 367.28111
```

4. Conclusion

This first model is easy to implement and seems to work fairly well on the Formosat data set.

Affiliation:

Serge Iovleff
Univ. Lille 1, CNRS U.M.R. 8524, Inria Lille Nord Europe
59655 Villeneuve d'Ascq Cedex, France
E-mail: Serge.Iovleff@stkpp.org
URL: http://www.stkpp.org