Documentation for the R-code to implement the clustering methodology in "Clustering for Sparsely Sampled Functional Data"

GARETH M. JAMES* AND CATHERINE A. SUGAR*

Description

This is a set of functions that fit the functional clustering approach as well as plotting discriminant curves and final curve estimates. The main fitting function is "fitfclust". In addition "fclust.pred", "fclust.curvepred", "fclust.plotcurves" and "fclust.discrim" produce the estimated $\hat{\alpha}$'s, cluster predictions, curve estimates (and associated confidence intervals) and discriminant functions. We provide documentation for all these functions below.

Installation

To install the software download the file fclust, open R and type

```
> source(''filename'')
```

where filename is the name you saved the file under (don't forget to give the directory structure in addition if needed.) Now if you type ls() you should see

```
> ls()
 [1] "fclust.curvepred"  "fclust.discrim"    "fclust.plotcurves"
 [4] "fclust.pred"       "fclustconst"       "fclustEstep"
 [7] "fclustinit"        "fclustMstep"       "fitfclust"
[10] "kmeans.rndstart"   "nummax"            "simdata"
```

"simdata" is a test data set consisting of 50 curves from one cluster and 50 from another. The other objects are various functions. Try typing the following command.

```
> testfit <- fitfclust(data=simdata,trace=T)
```

You should then see the following output.

```
[1] "Iteration 1 : Sigma =  0.00016891171719806"
[1] "Iteration 2 : Sigma =  0.0001304362413252 97"
[1] "Iteration 3 : Sigma =  0.00011358611476008"
[1] "Iteration 4 : Sigma =  0.000106583874318897"
```

*Marshall School of Business, University of Southern California

```
[1] "Iteration 5 : Sigma =  0.000103670123735399"
[1] "Iteration 6 : Sigma =  0.000102452631457189"
[1] "Iteration 7 : Sigma =  0.000101943790173395"
[1] "Iteration 8 : Sigma =  0.000101731144636507"
[1] "Iteration 9 : Sigma =  0.000101641283991723"
```

Try typing

```
> fclust.pred(testfit)$class
```

and you should see something like

```
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

though the 1's and 2's may be reversed. These are the cluster predictions for the 100 curves. Since this is an easy data set the predictions are all correct.

Now try

```
> fclust.plotcurves(fit=testfit,index=51,clustermean=T)
```

and you should see a plot with several curves on it. The black lines and dots represent the observed data (for curve 51). The red line is the predicted curve at all time points, the green and blue lines respectively represent 95% confidence and prediction intervals. Finally, the purple dashed lines are the estimated cluster centers for each cluster.

Next try

```
> fclust.discrim(testfit,abs=T)
```

and you should see another plot with a single line. This line represents the absolute discriminant function which gives the regions of greatest discrimination between the clusters. Details of all these functions are provided below.

fitfclust Function

This is the main function that fits the functional clustering algorithm.

Arguments

fitfclust allows for up to fifteen inputs. They are:

x : Suppose we observe N different curves, each consisting of n_i observations. The x should be a vector of length $\sum_i n_i$ with the first n_1 elements representing the observations for curve 1 etc. By default x=NULL.

curve : This is also a vector of length $\sum_i n_i$. The first n_1 elements should be 1's, the next n_2 elements 2's etc. By default curve=NULL.

timeindex : Also a vector of length $\sum_i n_i$. The first n_1 elements provide the locations on grid (see below) of curve 1 etc. So for example if grid is set to $(0,1,2,3)$ and curve 1 is observed at time points 1 and 3 then the first 2 elements of timeindex would be 2 and 4. By default timeindex=NULL.

data : An optional list containing x, curve and timeindex as described above. The used should provide either data or x, curve and timeindex. If data is provided then x, curve and timeindex are ignored. By default data=NULL.

q : This specifies the dimension of the natural spline that will be assumed for the cluster means and the individual curves. Evenly spaced knots are used. By default q=5.

h : This specifies the dimension of the space that the mean coefficients are assumed to lie within. Recall that $h \leq K-1$ where K is the number of clusters. So if $K=2$ then $h=1$ but for $K>2$ larger values are possible for h. Interpretation is considerably easier for $h=1$ but this may be an unrealistic assumption for $K>2$ if the mean coefficients do not lie on a line. By default h=1.

p : This enforces a rank p constraint on the covariance of the γ's, Γ. This could be useful if there are very few observations of each curve but in practice we have not found it to make too much difference. p must be less than or equal to q. By default p=5 (the same as q i.e. no rank constraint).

K : The number of clusters to fit. By default K=2.

tol : The tolerance to use in judging whether the algorithm has converged and should stop. The procedure exits if the percentage change in σ^2 is less than tol. By default tol=.001.

maxit : The maximum number of iterations allowed before the procedure stops. By default maxit=20.

pert : To initialize the algorithm we first compute an initial guess for the spline coefficients for each curve and perform k-means on these coefficients to get cluster memberships. The coefficients are estimated using least squares fits to the observations from each individual curve. This approach can fail if there are few observations per curve. pert adds a ridge term to the least squares fit. For pert=0 standard least squares is performed. As pert gets larger the estimates shrink towards zero. We have found that, unless there are many observations from each curve, better results are obtained using a small (rather than zero) value for pert. By default pert=0.01.

grid : This is the grid of time points that the spline matrix, S, is evaluated over. For example, if gird= seq(0,1,length=100) then S is evaluated at 100 equally spaced points between 0 and 1. Timeindex and grid together give the timepoint for each curve. For example, if the first curve has timeindex=c(1,4,6) then it has been observed at times grid[c(1,4,6)]. By default gird=seq(0,1,length=100).

hard : This is a true/false variable indicating whether to use the classification likelihood approach (hard=T) or the mixture likelihood method (hard=F). By default hard=F.

plot : This is a true/false variable indicating whether to plot the cluster mean curves at each iteration of the algorithm. It can be informative to see how the means change at each iteration. By default plot=F.

trace : This is a true/false variable indicating whether to print the current value for σ^2 at each iteration. By default trace=F.

Value

fitfclust returns a list containing five components. The five components in the list are:

data : A list containing all the original data.

parameters : A list containing all the estimated parameters. This consists of $\lambda_0, \Lambda, \Gamma, \sigma^2$ and the α_k's and π_k's (the probabilities of belonging to each cluster).

vars : This is a list containing results from the Estep of the algorithm. This consists of the γ's given membership in each cluster, the $\pi_{k|i}$'s, the products of $\gamma_i \gamma_i^T$ and the covariance of the γ_i's.

FullS : This is the matrix spline basis matrix S computed on the grid of timepoints.

grid : The grid of timepoints as supplied to the function.

fclust.pred Function

This function takes as input an object from fitfclust and returns the various results such as predicted cluster membership (and posterior probabilities) and the $\hat{\alpha}_i$'s etc. This can either be done on the original data or on a new data set.

Arguments

fclust.pred allows up to three inputs. They are:

fit : A fitted object from fitfclust i.e. the output from fitfclust.

data : A new data set. This is only needed if you require predictions for data other than that which were used in the original fit. Otherwise this can be ignored. By default data=NULL.

reweight : This is a true/false variable indicating whether the cluster probabilities should be calculated using the estimated prior probabilities of cluster membership, π_k, (reweight=F) or simply by assuming equal priors on each cluster (reweight=T). By default reweight=F.

Value

Calpha : An array with the covariances of each $\hat{\alpha}_i$.

alpha.hat : The estimated $\hat{\alpha}_i$'s for each curve.

class.pred : The predicted cluster membership for each curve.

distance : The estimated distance of each curve from each cluster center using the metric given by equation (13) in the paper. Note the distance is also adjusted for the prior π_k if reweight=F.

m : The largest estimated posterior probability for each curve.

probs : All the posterior probabilities of each curve coming from each cluster.

fclust.curvepred Function

This function takes as input an object from fitfclust and returns curve predictions for the entire time interval (red) , confidence intervals (green), and prediction intervals (blue) for each of the curves as well as cluster means (dashed purple).

Arguments

fclust.curvepred takes up to five inputs. They are:

fit : A fitted object from fitfclust i.e. the output from fitfclust.

data : A new data set. This is only needed if you require predictions for data other than that which were used in the original fit. Otherwise this can be ignored. By default data=NULL.

index : A vector indicating the curves that predictions are required for. If this is not provided predictions are produced for all curves. By default index=NULL.

tau : This is the confidence level to use for the confidence and prediction intervals. See Section 3.3 of the paper for further details. By default tau=0.95.

tau1 : See Section 3.3. of the paper for details on the definition of tau1. This should not need to be adjusted. By default tau1=0.975.

Value

etapred : The estimated spline coefficients for each curve using equation (17) from the paper.

gpred : The estimated curves over the time points specified in grid.

upci : The upper confidence interval for each curve.

lowci : The lower confidence interval for each curve.

uppi : The upper prediction interval for each curve.

lowpi : The lower prediction interval for each curve.

index : The index of curves that the predictions have been produced for.

grid : The grid of time points.

data : Either the original data or the new data if it has been supplied.

meancurves : The mean curves for each cluster.

fclust.plotcurves Function

This function plots the predicted curves and confidence intervals from the fclust.curvepred function. Note if you just want to plot predictions and confidence intervals and not save any of the results you can call fclust.plotcurves directly without using fclust.curvepred. See below for details.

Arguments

fclust.plotcurves takes up to six inputs. They are:

object : The output from fclust.curvepred. If you have already called fclust.curvepred then you can save time by using the output from that function. Otherwise fclust.plotcurves will call fclust.curvepred internally each time you produce a plot. Either fit or object must be supplied. By default object=NULL.

fit : The output from fitfclust. If object is supplied then fit is not required. By default fit=NULL.

index : A vector containing the curves for which plots are required. This can contain between 1 and 36 elements. If index is not supplied the function attempts to plot all curves (which will only work if there are no more than 36). By default index=NULL.

ci : This is a true/false variable indicating whether the confidence intervals should be plotted. By default ci=T.

pi : This is a true/false variable indicating whether the prediction intervals should be plotted. By default pi=T.

clustermean : This is a true/false variable indicating whether the cluster mean curves should be plotted. By default clustermean=F.

Value

No values are returned. The function provides plots.

fclust.discrim Function

This function takes as input an object from fitfclust and plots the discriminant curve (or curves) that shows the regions of greatest discrimination between the clusters. Note that if $h = 1$ then only one curve is produced. This is the optimal case in terms of interpretation. If $h > 1$ then h different curves are plotted and interpretation of the results becomes more difficult.

Arguments

fclust.plotcurves takes up to two inputs. They are:

fit : A fitted object from fitfclust i.e. the output from fitfclust.

absvalue : This is a true/false variable indicating whether to plot the discriminant function (absvalue=F) or the absolute value of the function (absvalue=T). Often we are looking for places where the curve is significantly different from zero in which case we may be most interested in the absolute value. By default absvalue=F.

Value

No values are returned. The function provides plots of the discriminant functions.