FEmBa Documentation

Joshua Derenski

10/25/2018

This document discusses the structure of the documentation for the code regarding FEmBa:

"Tweedie's Formula and Score Function.R":

This file contains the functions used in performing bias correction for a given matrix of coefficient vectors, corresponding to basis expansions for modeled curves.

tweedie_correction

Performs bias correction on the coefficient vectors.

Arguments

- X (numeric matrix): A $p \times n$ matrix of coefficient vectors, where p is the dimension of the vectors, and n is the number of vectors you have. Note: this matrix is NOT $n \times p$!
- variance_covariance (numeric matrix, must be positive definite and have no missing values): The conditional covariance of θ, given μ. A $p \times p$ matrix.
- functional_basis (numeric matrix): A $p \times p$ numeric matrix, which is calculated from the basis used for representing the curves. Used in calculating the variance in the basis representation of each curve.

$$\text{basis\_for\_integral\_calculation} = \int \boldsymbol{s}(t)\boldsymbol{s}(t)^t dt$$

- Transformation (boolean, default TRUE): Specifies whether or not the coefficient vectors should be transformed so as to have uncorrelated coordinates before performing bias correction.
- lambda_grid (numeric vector, must be of length at least 1, default 10^seq(-6, 6, 1)): A grid of *lambda*s to be considered for penalization. Optimal value is chosen with K-fold cross-validation. If the length of this vector is 1, the single value in that vector is chosen by default.
- step (numeric, default .001): This parameter determines the density with which to generate the basis used for the score function.
- ICA (boolean, default TRUE): Specifies if Independent Component Analysis (ICA) should be used when transforming the coefficient vectors for bias correction.
- number_of_folds (numeric): The number of folds used in cross validation for estimating λ (*Must be greater than 1!*).
- number_of_knots_score_function: The number of knots to use when generating the cubic B-Spline basis for modeling the score function.

Returns

This function returns a list of three objects:

- bias_corrected_coefficients (numeric matrix): A $p \times n$ matrix containing the bias-corrected coefficients of the basis represenation of each curve.
- chosen_lambda (numeric): The value of λ used in estimating the score function.
- cross_validation_matrix (numeric matrix if cross-validation used, NA otherwise): A matrix containing the cross-validation error for different values of λ.

score_estimator

Estimates the score function, to be used in Tweedie's formula.

Arguments

- thetas (numeric matrix): A $p \times n$ array, whose columns are the basis representations of the curves.
- phi (numeric matrix): A $p \times p$ array, where

$$\text{phi} = \Sigma_{\boldsymbol{\theta}^*} \left(\int \boldsymbol{s}(t) \boldsymbol{s}(t)^T dt \right) \Sigma_{\boldsymbol{\theta}^*}$$

Here, $\Sigma_{\boldsymbol{\theta}_*}$ is the marginal covariance of a vector $\boldsymbol{\theta}_*$, and this coefficient vector will correspond to the data directly, or be a linear transformation of your data, depending on whether or not the user chose to first uncorrelate the coordinates of the coefficient vectors.

- step (numeric,default .001): The granularity of the basis used for generating the score function.
- lambda_grid (numeric vector, must be of length at least 1): A grid of *lambda*s to be considered for penalization. Optimal value is chosen with K-fold cross-validation. If the length of this vector is 1, the single value in that vector is chosen by default.
- num_knots (numeric): The number of knots to use when generating the cubic B-Spline basis for modeling the score function.

"Eigenbasis Estimator with Traditional PCA.R":

coeff_vec

Provides least-squares estimates of the basis representation of a given curve in a given basis space.

Arguments:

- y (numeric vector) The observed points of the curve.
- observed_basis (numeric matrix) A design matrix corresponding to the basis the curve is embedded in.

"basis_estimator_pca"

Arguments

- X (numeric matrix): An $n \times T$ matrix of the observed values of the curves. All curves must be observed at the same time points.
- sample_splitting (boolean): Dictates whether or not to perform sample splitting. If set to TRUE, curves are first separated into two groups: G_1 and G_2. Then, the curves in G_1 is used to estimate a basis for modeling the curves in G_2, and after that the curves in G_2 is used to estimate a basis for modeling the curves in G_1. The roles of each group is then switched and the process is done once more.
- our_times (list of numeric vectors): A list where each element is a numeric vector of times at which the corresponding curve has been observed.
- cumulative_variation_described (numeric): A number corresponding to the desired proportion of variation in the original data the user wants the PCA decomposition of the data to preserve.
- number_of_comps (numeric): Number of components the user wishes to have for PCA. This is equivalent to prespecifying the dimension of the basis you want to use to model the data.
- choose_comp_by_proportion (boolean): Determines which method is used to determine the number of principal components to use for PCA. If set to TRUE, this number is chosen by making sure the appropriate amount of variation the user wants preserved in the PCA representation of the data is actually preserved. If TRUE, cumulative_variation_described must be specified. If choose_comp_by_proportion is set to FALSE, number_of_comps must be specified.
- transformation_for_constraints (function): A function which allows for the imposition of constraints on the curves. If no constraints are needed, set this argument equal to NULL. For example, if one wanted to constrain the curves to be positive, one might set transformation_for_constraints = function(x) x^2
- maximum_number_of_iterations (numeric): The max number of iterations desired when the user is fitting constrained curves. As the optimization problem that needs to be solved when constraints are present does not have a closed form solution in general, this problem must be solved numberically. This variable determines how many iterations at most the optmization algorithm goes through when solving the problem.

"fpca_estimator_trad"

This function performs the Principal Components Analysis on the data, and estimates the coefficient vectors for the curves we will be estimating. This function is used within "basis_estimator_pca".

Arguments

- curves_for_estimating_eigenfunctions (numeric matrix): The curves that will be used when estimating the eigenfunctions.
- curves_to_be_modeled (numeric matrix): The curves for which we will obtain least squares estimates for the coefficients corresponding to their representation in the functional eigenbasis space.
- times_for_curves_to_be_modeled (numeric list): The times corresponding to the curves that we will be modeling.
- variation_described (numeric): A number corresponding to the desired proportion of variation in the original data the user wants the PCA decomposition of the data to preserve.

- n_comps (numeric): Number of components the user wishes to have for PCA. This is equivalent to prespecifying the dimension of the basis you want to use to model the data.
- choose_comps_with_proportion (boolean): Dictates which method is used to determine the number of principal components to use for PCA. If set to true, this number is chosen by thresholding the appropriate amount of variation the user wants preserved in the PCA representation of the data. If this boolean is TRUE, cumulative_variation_described must be specified. If choose_comp_by_proportion is set to FALSE, number_of_comps must be specified.