Fast Inference of Functional Linear Regression with SGD for Ultra-Large Dataset

Zihan Lou[1]

[1]School of Economics and Finance, Xi'an Jiaotong University, 710049, Shaanxi, China

Abstract

This is a brief demonstration of our SGD method for estimation and inference in functional linear regression.

1. Method and Algorithm

Functional data are widely used across various fields. When dealing with extremely large sample sizes—often comprising tens of millions of observations and high dimensionality—traditional estimation and inference methods typically require storing massive datasets and computing the inverse of matrices, which can be computationally infeasible. To enable rapid estimation and inference in functional linear regression, we developed an online learning method utilizing the Stochastic Gradient Descent (SGD) algorithm. Compared to the Bootstrap method, the SGD approach demonstrates superior efficiency and accuracy. For functional linear regression:

$$Y_i = \alpha + \int x_i(t)\beta(t)dt + \epsilon_i$$

We can use basis expansion to represent $\beta(t)$

$$\beta(t) = \sum_{l=1}^{d} B_l(t)\theta_l = \mathbb{B}'\theta$$

And the regression model can be denoted as:

$$Y_i = \alpha + \int x_i(t)\mathbb{B}'dt\theta + \epsilon_i = Z_i'\omega + \epsilon_i$$

where

$$Z_i = \begin{bmatrix} 1 & \int x_i B_{i1}dt & \cdots & \int x_i B_{id}dt \\ \cdots & \cdots & & \cdots \\ 1 & \int x_n B_{i1}dt & \cdots & \int x_n B_{id}dt \end{bmatrix}$$

To account for the smoothness of our fit, we incorporate a penalty term into the loss function. The loss function is given by:

$$PENSSE_\lambda(\beta) = \sum_{i=1}^{n}(Y_i - \alpha - \int x_i(t)\beta(t)dt)^2 + \lambda \int [L\beta(t)]^2 dt$$

so with the traditional estimation method, and denote $\int [L\beta(t)]^2 dt$ as R, the estimator and covariance matrix is:

$$\hat{\omega} = (Z'Z + \lambda R)^{-1}Z'Y,\ Var(\hat{\omega}) = \sigma_\epsilon^2 (Z'Z + \lambda R)^{-1}Z'Z(Z'Z + \lambda R)^{-1}$$

When it becomes infeasible to compute $\hat{\omega}$ and $Var(\hat{\omega})$ due to the ultra-large size of the dataset, we introduce the SGD algorithm:

$$\omega_i = \omega_{i-1} - \gamma_i \nabla PENSSE_\lambda(\omega_i)$$

When a new sample is received, we can update the estimator. According to asymptotic theory, the average of the updated estimator sequence can be taken, and the covariance matrix can be updated as:

$$\bar{\omega_n} = \frac{1}{n}\sum_{i=1}^{n}\omega_i$$

$$\hat{V_n} = \frac{1}{n}\sum_{s=1}^{n}\left\{\frac{1}{\sqrt{n}}\sum_{i=1}^{s}(\omega_i - \bar{\omega}_n)\right\}\left\{\frac{1}{\sqrt{n}}\sum_{i=1}^{s}(\omega_i - \bar{\omega}_n)\right\}'$$

To improve computational efficiency when updating the estimator and covariance matrix, certain solutions are employed:

$$\omega_i = \omega_{i-1} - \gamma_i \nabla PENSSE$$

$$\bar{\omega}_i = \bar{\omega}_{i-1}\frac{i-1}{i} + \omega_i \frac{1}{i}$$

$$\hat{V}_i = i^{-2}(A_i - \hat{\omega}_i b_i' - b_i \bar{\omega}_i{}' + \bar{\omega}_i \bar{\omega}_i' \sum_{s=1}^{i} s^2)\ \hat{V}_{n,jj}$$

$$A_i = A_{i-1} + i^2 \bar{\omega}_i \bar{\omega}_i', b_i = b_{i-1} + i^2 \bar{\omega}_i$$

The learning rate and penalization parameter are crucial for accurate estimation and inference. First, we select the penalization parameter through Generalized Cross Validation (GCV):

$$GCV(\lambda) = (\frac{n}{n - df(\lambda)})(\frac{SSE}{n - df(\lambda)})$$

When updating the learning rate, the change in the loss is given by:

$$\Delta l = l(X_i; \hat{\omega} - \gamma_i g) - l(X_i; \hat{\omega})$$

We can express $l(X_i; \hat{\omega} - \gamma_i grad)$ using a Taylor expansion around $\hat{\omega}$ as:

$$l(X_i; \hat{\omega} - \gamma_i g) = l(X_i; \hat{\omega}) - \nabla l(X_i; \hat{\omega})\gamma_i g + \frac{1}{2}\gamma_i^2 g' \nabla^2 l(X_i; \hat{\omega}) g + o(\gamma_i^2 g' g)$$

Thus, the change in the loss is:

$$\Delta l = -\nabla l(X_i; \hat{\omega})\gamma_i g + \frac{1}{2}\gamma_i^2 g' \nabla^2 l(X_i; \hat{\omega}) g$$

To determine the optimal learning rate, we set the derivative of the change in loss to zero:

$$\frac{d\Delta l}{\gamma_i} = -\nabla l(X_i, \hat{\omega}) g + r g' \nabla^2 l(X_i, \hat{\omega}) g = 0$$

The optimal learning rate is then given by:

$$\gamma_i^* = \frac{\nabla l(X_i, \hat{\omega}) g}{g' \nabla^2 l(X_i, \hat{\omega}) g}$$