Journal of Education and Culture Studies
ISSN 2573-0401 (Print) ISSN 2573-041X (Online)
Vol. 9, No. 3, 2025
www.scholink.org/ojs/index.php/jecs

Original Paper

Teaching Semiparametric Modeling Based on R Programming: Expectile Regression in Monotonic Partially Linear Models

Jun Sun[1] & Mingtao Zhao[1]

[1] School of Statistics and Applied Mathematics, Anhui University of Finance and Economics, Bengbu, 233030, China

Received: May 8, 2025 Accepted: June 27, 2025 Online Published: July 11, 2025

doi:10.22158/jecs.v9n3p33 URL: http://dx.doi.org/10.22158/jecs.v9n3p33

Abstract

Instruction in the expectile regression estimation method for semiparametric partially linear models (PLMs) with monotonic constraints is a crucial component of graduate statistics courses. This paper demonstrates the R programming implementation of this estimation method using monotone B-spline approximation and provides a Monte Carlo simulation example for practical teaching purposes.

Keywords

Partially linear models, Practical pedagogy, R Programming

1. Introduction

In the realm of complicated data analysis, linear models are often inadequate for characterizing relationships between responses and predictors, leading to model misspecification or high prediction variability, therefore many useful statistical tools have been developed for addressing these concerns. Examples of these tools include semiparametric partially linear models (PLMs). The PLMs, introduced by Engle et al. (1986), take the form:

$Y = X^T\beta + f(U) + \varepsilon,$

where Y is the response variable, $X = (X_1, \dots, X_p)^T$ is the p -dimensional covariate vector, β denotes the corresponding linear regression coefficients, $f(\cdot)$ is an unknown univariate link function, U ranges over a non-degenerate compact interval, ε represents the random error term, and "T" denotes the transpose of a vector or matrix throughout this paper. Various estimating approaches have been proposed for PLMs, including least squares, penalized splines, quantile regression, principal components regression, and relative error minimization (Liang, 2006; Du et al., 2013; Liu et al., 2016; Chen & Liu, 2023).

In statistics, expectile regression (Newey & Powell, 1987) applied to PLMs with monotonic constraints

offers a powerful paradigm for comprehensive distributional modeling. This integration makes it a pedagogically rich topic for advanced graduate courses in statistics, econometrics, and related fields. Its instructional value stems from synthesizing several critical concepts: (I) The interpretability of parametric effects within PLMs. (II) The flexibility of nonparametric function estimation. (III) The necessity of monotonicity constraints (which are ubiquitous in fields such as dose-response and economic behavior). (IV) The distributional insights provided by expectiles (generalizing quantile regression). (V) Associated constrained optimization techniques. Teaching this methodology equips students with a powerful toolkit for analyzing complex real-world relationships where standard mean regression or unconstrained nonparametric methods fail, thereby fostering deeper understanding of modern semiparametric estimation and statistical inference.

Our teaching framework comprises three components: (I) Conceptual Foundations, including review of PLMs and expectile regression fundamentals, interpretation of monotonicity constraints for identifiability and realism, and case studies used to demonstrate the performance of expectile regression with monotonic constraints. (II) Model implementation, including detailed model specification and estimation framework, constrained B-spline estimation, R-based computational algorithms, applied case studies, and performance evaluation via Monte Carlo simulations. (III) Theoretical exploration.

This paper focuses specifically on R-based algorithmic development for expectile regression in monotonic PLMs. The included simulation studies enable students to comprehend the significance of R programming for solving statistical modeling challenges.

2. B-spline Approximation and Design of R Programming

2.1 B-spline Approximation

The challenge lies in estimating $f(\cdot)$without assuming a parametric form. In our instructional approach, we first show students how to estimate the monotone nonparametric function $f(\cdot)$. Due to the desirable numerical stability and fast computation of B-spline basis functions approximation, we approximate the unknown function $f(\cdot)$ by a linear combination of B-spline basis functions. Specifically, let $(X_i, U_i, Y_i),\ i = 1, \dots, n$ be independent and identically distributed realizations of $(X, U, Y),$ which are generated from PLMs. Let $B(U) = (B_1(U), \dots, B_J(U))^T$be a set of B-spline basis functions of the order of q with N internal knots, where the order $q \geq 2$ and $J = N + q$. Then the nonparametric function $f(\cdot)$can be approximately expressed as $f(U) \approx B(U)^T\gamma$, where $\gamma = (\gamma_1, \dots, \gamma_J)^T$ is the vector of spline coefficients. To ensure the nondecreasing property of function $f(\cdot)$on the compact support $U \in S_U$, following Schumaker (1981), we impose a nondecreasing constraint on the coefficients $\gamma = (\gamma_m : 1 \leq m \leq J)^T$, i.e., $\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_J$. Thus, the sample version of PLMs can be written as $Y_i \approx {X_i}^T\beta + B(U_i)^T\gamma + \varepsilon_i$. The parameters β and γ are estimated by minimizing the objective function:

$Q_n(\beta, \gamma) = \sum_{i=1}^{n} \rho_\tau\,(Y_i - {X_i}^T\beta - B(U_i)^T\gamma),$

where $\rho_\tau(\cdot)$ is a convex loss function with the form $\rho_\tau(s) = |\tau - I(s < 0)| \cdot s^2$ for any fixed value $\tau \in (0,1)$. Clearly, the above formulation transforms the problem into a tractable constrained convex

optimization via B-spline approximation and linear coefficient constraints.

2.2 Design of R Programming

We solve this optimization using the constrOptim package in R, which natively enforces the monotonicity constraints $\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_J$. The implementation requires specifying the analytical gradient of the objective function. Let $\prod_i = (X_i{}^T, B(U_i)^T)^T$ and $\theta = (\beta^T, \gamma^T)^T$, the gradient can be formulated as $\partial Q_n(\theta)/\partial\theta$=-2 $\sum_{i=1}^n \prod_i^T W_i(Y_i - X_i{}^T\beta - B(U_i)^T\gamma)$,

where $W_i = \tau$ if $Y_i - X_i{}^T\beta - B(U_i)^T\gamma \geq 0$ else W_i=1-τ. The following R code implements this constrained optimization:

```
RERest <- function(q,N,y,z,u,tau){
# B-spline basis construction
   tmp <- rep(0, N); p <- ncol(z)
   for(j in 1:N){ tmp[j] <- quantile(sort(u), j/(N+1))}
   b <- max(u) + 10^(-10)
   a <- min(u) - 10^(-10)
   ku <- c(rep(a,q),t(tmp),rep(b,q))
# Create B-spline basis matrix
   bpu   <- splineDesign(knots = ku, x=u, ord = q)

   Phi <- cbind(bpu, z)
   gamma.old <- lm(y ~ Phi + 0)$coef   # OLS initialization (unconstrained)
   gamma.old <- c(sort(gamma.old[1:(q+N)]),gamma.old[-c(1:(q+N))])
# Define optimization functions
      fn<-function(zz){
         resid <- y - Phi %*% zz
         weights <- ifelse(resid >= 0, tau, 1 - tau)
         Ln <- sum( weights * resid^2 )
         Ln
      }
# Gradient function
      gr <- function(zz){
         resid <- y - Phi %*% zz
         weights <- ifelse(resid >= 0, tau, 1 - tau)
         Lnprime <- (-2) * Phi*((weights * resid) %*% seq(1,1,length=ncol(Phi)))
         Lnprime <- colSums(Lnprime)
         Lnprime
      }
```

```
# Monotonicity constraints
      Amat <- matrix(0, nrow = (p+q+N), ncol = (p+q+N))
      for(i in 2:(q+N)){Amat[i,(i-1)] = -1}
      for(i in 2:(q+N)){Amat[i,i] = 1}
      bvec <- rep(0, (p+q+N))    - 1e-10
# Constrained optimization
      RR <- constrOptim(theta=gamma.old, f=fn, grad = gr, ui = Amat, ci = bvec, method = "BFGS")
      theta_hat <- RR$par
# Extract results
   ddd1 <- theta_hat[1:(q+N)]
   fnew <- as.vector(bpu %*% ddd1)
   beta_new <- theta_hat[-c(1:(q+N))]
# Return results
   list(beta_new = beta_new, fnew=fnew, bpu=bpu)
}
```

For comparative analysis, we also provide the R implementation without monotonicity constraints:

```
ERest <- function(q,N,y,z,u,tau){
# B-spline basis construction
   tmp <- rep(0, N); p <- ncol(z)
   for(j in 1:N){ tmp[j] <- quantile(sort(u), j/(N+1))}
   b <- max(u) + 10^(-10)
   a <- min(u) - 10^(-10)
   ku <- c(rep(a,q),t(tmp),rep(b,q))
# Create B-spline basis matrix
   bpu    <- splineDesign(knots = ku, x=u, ord = q)
   Phi <- cbind(bpu, z)
   gamma.old <- lm(y ~ Phi + 0)$coef    # OLS initialization (unconstrained)
# Define expectile loss function
      fsn<-function(zz){
         resid <- y - Phi %*% zz
         weights <- ifelse(resid >= 0, tau, 1 - tau)
         Ln <- sum( weights * resid^2 )
         Ln
}
# Optimization without monotonicity constraints
      RR <- optim(par=gamma.old, fn = fsn, method = "BFGS")
```

```
    theta_hat <- RR$par
# Extract results
  ddd1 <- theta_hat[1:(q+N)]
  fnew <- as.vector(bpu %*% ddd1)
  beta_new <- theta_hat[-c(1:(q+N))]
# Return results
  list(beta_new = beta_new, fnew=fnew, bpu=bpu)
}
```

3. Numerical Examples for Practical Teaching

We provide Monte Carlo simulations to evaluate the feasibility of the proposed algorithm. Suppose the sample datasets come from the model $Y_i = X_{i1}\beta_1 + X_{i2}\beta_2 + f(U_i) + \varepsilon_i$ with sample size $n = 500$, $X_i = (X_{i1}, X_{i2})^T$, X_{i1} and X_{i2} follow a bivariate normal distribution with mean 0, variance 1, and covariance 0.3, the true parameters $\beta_1 = 1$ and β_2=3.5, set $f(U_i)$=$1.2(U_i - 1)^3$ and $\tau = 0.5$, the variable U_i follows the uniform distribution on [0,2], ε_i follows $t(5)$ distribution. To implement the proposed algorithm, we use cubic splines and set $N = 2$ for simplicity. This choice of N is small enough to avoid overfitting in typical problems with sample size not too small and big enough to flexibly approximate many smooth functions. The R code implementing these simulations is presented below:

```
library(MASS)
library(splines)
# Define the true nonparametric function
# This cubic function is monotonically increasing on [0,2]
Gfun <- function(x){
1.2*(x-1)^3
}
# Simulation parameters
n <- 500       # Number of observations
tau <- 0.5     # Expectile level
p <- 2         # Dimension of linear covariates
q <- 4         # Order of B-splines (q=4 for cubic splines)
N <- 2         # Number of interior knots for splines
set.seed(134)   # Set random seed for reproducibility
# Create covariance matrix for X covariates
covMa <- matrix(1, p, p)
  for (i in 1 : p){
    for (j in 1 : p){
```

```
        covMa[i, j] <- 0.3^(abs(i-j))
    }
  }
 # Generate predictor variables
 X <- mvrnorm(n, rep(0, p), covMa)      # Multivariate normal covariates
   U <- sort(runif(n,0,2))                    # Uniform covariate (sorted for plotting)
# Generate error term from t-distribution (heavy-tailed errors)
error <- rt(n,5)       # t-distribution with 5 degrees of freedom
# True regression coefficients
beta <- c(1, 3.5)      # Parameters for linear part
# Generate response variable
Y <- X%*% beta + Gfun(U) + error
# RERest: Monotonic constrained regression using constrOptim
rer_result <- RERest(q,N,Y, X, U, tau)
# ERest: Unconstrained regression using standard optim
er_result <- ERest(q,N,Y, X, U, tau)
# Create plot to compare results
plot(U, Gfun(U), type = "l")
points(U, rer_result$fnew,    type = "l", lty=5, col="red")
points(U, er_result$fnew,    type = "l", lty=4, col="blue")
```

Execution of the provided R code generates the simulation results displayed in Figure 1. The true function is represented by the solid black line, while the unconstrained estimate (without monotone B-spline approximation) appears as the blue dashed line. The red dashed line depicts the constrained estimate obtained through our proposed methodology. Figure 1 demonstrates the effectiveness of both the estimation method and its computational implementation in R.

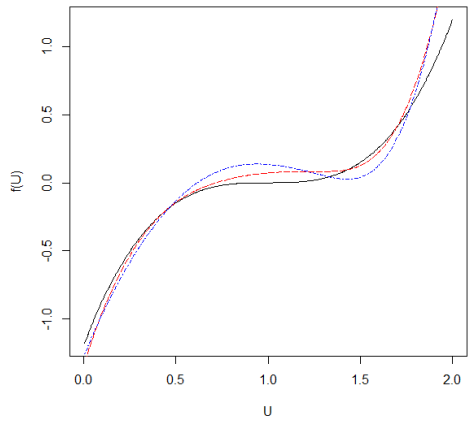

Figure 1. Estimated Nonparametric Curves with Sample Size $n = 500$

4. Conclusion

The integration of expectile regression for monotonic PLMs into postgraduate curricula provides a powerful pedagogical framework that unites advanced statistical theory, computational methodologies, and applied data analysis skills. This structured approach—synthesizing algorithmic principles, software implementation, and critical evaluation through simulation studies—effectively prepares students to master and deploy sophisticated semiparametric modeling techniques for contemporary research challenges.

References

Chen, Y., Liu, H. (2023). A new relative error estimation for partially linear multiplicative model. *Communications in Statistics-Simulation and Computation*, *52*, 4962-4980.

Du, J., Sun, Z., & Xie, T. (2013). M-estimation for the partially linear regression model under monotonic constraints. *Statistics & Probability Letters*, *83*, 1353-1363.

Engle, R., Granger, C., Rice, J., & Weiss, A. (1986). Semiparametric estimates of the relation between weather and electricity sales. *Journal of the American Statistical Association*, *81*, 310-320.

Liang, H. (2006). Estimation in partially linear models and numerical comparisons. *Computational Statistics & Data Analysis*, *50*, 675-687.

Liu, C., Guo, S., & Wei, C. (2016). Principal components regression estimator of the parameters in partially linear models. *Journal of Statistical Computation and Simulation*, *86*, 3127-3133.

Newey, W., & Powell, J. (1987). Asymmetric least squares estimation and testing. *Econometrica*, *55*, 819-847.

Schumaker, L. (1981). *Spline Functions: Basic Theory*. Wiley.