Stats 790 project report

Gujie Fu 400067957

April 2023

1 Introduction

Quantile regression is a type of regression analysis used in statistics. Unlike ordinary least squares regression, which aims to estimate the conditional mean of the response variable across values of the predictor variables, quantile regression aims to estimate based on different quantiles. This makes quantile regression particularly useful when the data are skewed or heterogeneous, as it provides a more complete view of the possible conditional quantitative relationships between variables. The method of quantile regression was first introduced by the Robert Koenker and Gilbert Bassett Jr[KD87]. They introduced this approach to enhance the exploratory power of regression models and enable them to describe not just the average effect of a set of predictors, but their effect at different points in the distribution of the response variable. While the OLS regression model is based on minimizing the sum of the squared errors , quantile regression is based on minimizing the sum of the absolute differences of the errors for the specified quantile. Since its introduction, quantile regression has been particularly useful in situations where the assumption of normality is violated. This includes scenarios where the data have heavy tails or where the variability of the response variable changes across values of the predictor variables. Similar to the linear regression, we can put penalty term on the loss function. The L_1 penalty, also known as Lasso regularization is a well known regularization method. In this project, we pick up two data set and then perform quantile regression with or without L_1 penalty.

2 Methodology

The quantile q_τ where $\tau \in (0,1)$ of a random variable y is defined by

$$P(y \le q_\tau) \le \tau \quad or \quad P(y \le q_\tau) \ge 1-\tau$$

However, based on the data we observe, assuming there are n observations, we can estimate the empirical quantile by

$$\frac{1}{n}\sum_{i=i}^{n} I(y_i \le \hat{q}_\tau) \le \tau \quad or \quad \frac{1}{n}\sum_{i=1}^{n} I(y_i \ge \hat{q}_\tau) \ge 1-\tau$$

where I is the indicator function. A common loss function to be minimized is defined as

$$\hat{q}_\tau = \underset{q}{\operatorname{argmin}} \sum_{i=1}^{n} w_\tau(y_i, q) \cdot |y_i - q| \quad where$$

$$w_\tau(y_i, q) = \begin{cases} 1-\tau & y_i < \tau \\ 0 & y_i = \tau \\ \tau & y_i > \tau \end{cases}$$

which is a weighted sum of absolute deviations. It is a quantile regression loss function which is called check function of Koenker and Bassett (1978).[LZ12] Instead of one quantile q_τ, we can natually replace the quantile q_τ by β_τ with

$$\hat{\beta}_\tau = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} w_\tau(y_i, \eta_{i\tau}(\beta)) \cdot |y_i - \eta_{i\tau(\beta)}| \quad where \eta_{i\tau}(\beta) = \beta^T x_i$$

$$w_\tau(y_i, \eta_{i\tau}(\beta)) = \begin{cases} 1-\tau & y_i < \eta_{i\tau}(\beta) \\ 0 & y_i = \eta_{i\tau}(\beta) \\ \tau & y_i > \eta_{i\tau}(\beta) \end{cases}$$

Since the check function is not differentiable at 0, we can't find a solution similar as in the linear regression problem. Thus we will use a check function which is smooth and differentiable.[Zhe11] The check function is

$$S_{\tau,\alpha}(u) = \tau u + \alpha log(1 + e^{-\frac{u}{\alpha}})$$

The following figure is a comparison of two different check function. The red curve is more smooth than the usual check function around 0.

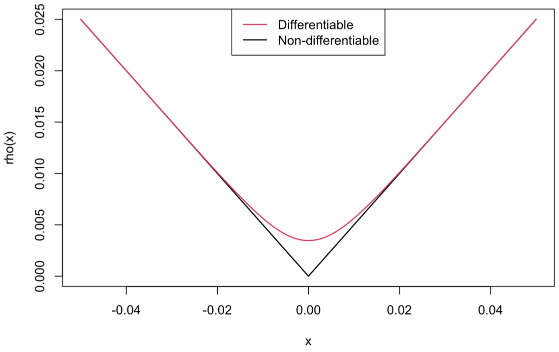

Figure 1: Comparison of values of two check function

Next, consider the quantile regression with L_1 penalty and we the loss function is defined as the following

$$\underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho_\tau(y_i - \beta_0 - \boldsymbol{\beta}^T \boldsymbol{x_i} + \lambda ||\boldsymbol{\beta}||_1)$$

The function ρ_τ is called the check function of Koenker and Bassett[KD87] and is defined in an alternative parameterization because

$$\rho_\tau(y - f(\boldsymbol{x})) = \begin{cases} \tau \cdot (y - f(\boldsymbol{x})) \quad if \quad y - f(\boldsymbol{x}) > 0 \\ -(1-\tau) \cdot (y - f(\boldsymbol{x})) \quad otherwise \end{cases}$$

where $f(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}$ and $\tau \in (0, 1)$. There are other loss function which looks like least square method as the following

$$\underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} (\rho_\tau(y_i - \beta_0 - \boldsymbol{\beta}^T \boldsymbol{x_i})^2 + \lambda ||\boldsymbol{\beta}||_1)$$

However, in this project we will use the loss function with the absolute deviance.

3 Illustration

The first data set I use is a build in data in R which is called "age.income" which has 205 pairs observations on Canadian workers from a 1971 Canadian Census Public Use Tape. The following is a scatter plot of the dataset. From the figure we can see that when age is less than 30 years, the income increases very quickly. When age is larger than 30 years, it seems the mean of income does not change a lot but these data points definitely capture more variance, especially when age is larger than 50 years. Hence, we can not assume constant variance and this data shows heteroskedasticity.

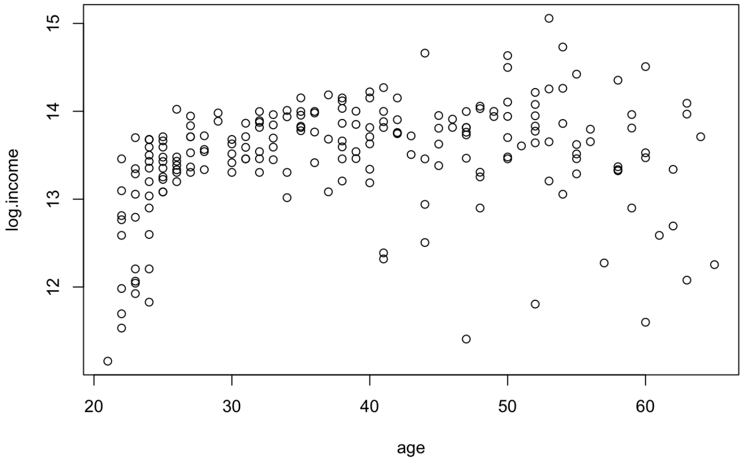

Figure 2: Caption

Here's a brief introduction about the algorithm we use. We define a function called "qlm" and it first fits an ordinary least squares linear regression model to the data, using the specified formula. It then constructs the design matrix and the response vector from the linear model object. We defines an objective function that calculates the sum of a smooth function applied to the residuals of the model. Then we initialize a list to store the estimated coefficients of the quantile regression model for each quantile. A loop is entered over the quantiles , in which it optimizes the objective function to find the coefficients of the quantile regression model for that quantile. The following figure contains three quantile regression curve with respect to different quantiles. If we use the most naive quantile regression, then we would expect a straight line instead of a curve, however, in this project I use different degrees of polynomial to fit the response variable and capture the nonlinear relationship. More specifically , I use 4^{th} and 5^{th} degrees of polynomial generated by function "poly()" to make sure there is no correlation between these higher order terms. From the following figure we can see that the quantile regression plots share the similar pattern. The income increases for all three quantiles and for the median which is 50% quantile, income is stable. For the lower 5% and upper 95% quantile, the behavior of income are in opposite direction but eventually both of them decrease.

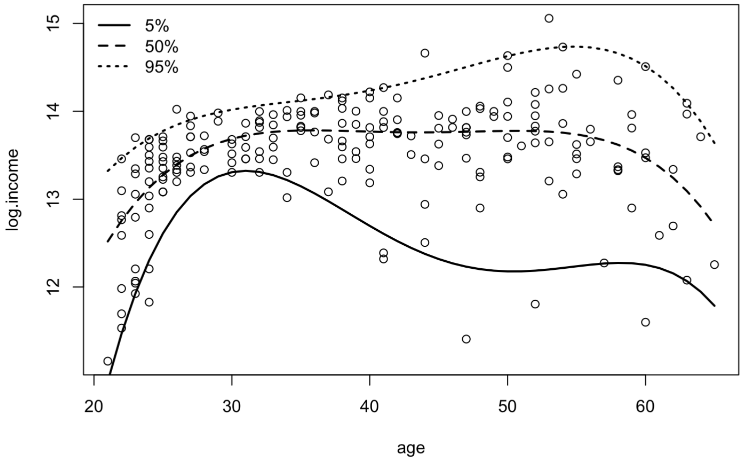

Figure 3: Quantile regression curve of three different quantiles

There is a package in R studio called "quantreg" which can be used to perfrom quantile regression. Thus we can compare the results from my method with the package. The processing time of my method is much longer than the quantile regression package which almost take 6 to seven times longer. My personal guess is that the optimizer takes more iterations to converge. However, the output of my method is very close to the package. The following two figures are comparison of two methods.

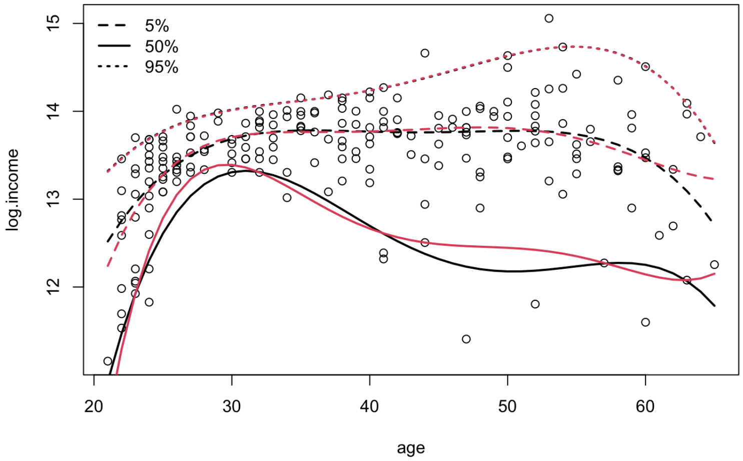

Figure 4: Comparison of two methods (polynomial with degree 4)

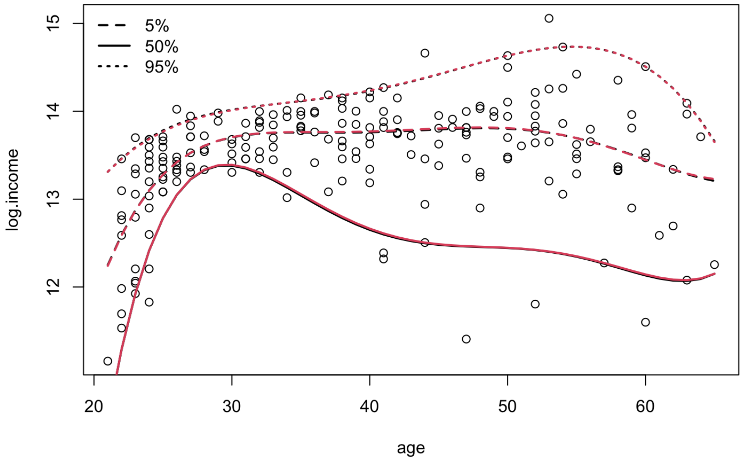

Figure 5: Comparison of two methods (polynomial with degree 4)

From the figure we can see that if we fit the quantile regression model with polynomial with degree 4, then the curve of all three quantiles are very close to the curve from the package. If we use polynomial with degree 5, then there is almost no difference between them. However, the adjusted R^2 of both method is low, which is approximately 32%. The overall performance is not so exciting. My personal guess is that expanding the predictor variables just by applying polynomial does not work so well in this case. The following table shows the mean squared errors of three methods and three quantiles we use : It is surprising that the mean squared error first decreases and then increases. I would expect that the mean square error decreases monotonically as the quantile increases.

method,τ	5%	50%	90%
poly(4)	1.323917	0.2864741	0.8112137
ploy(5)	1.191086	0.2810204	0.8113107
build-in	1.17841	0.281677	0.8098487

My guess is that when we use higher quantile, more variability is introduced to the model. So the mean squared error acts like a reversed unimodal model. We can see from the table that our result is very close to the build in quantile regression function.

Now let's have a look at the quantile regression with L_1 penalty. The data we use is called "Medical Cost Personal Datasets" which is publicly available on Kaggle.[Cho18] It has 6 predictor variables and 1 response variable. We make categorical variables to be factors and scale all variables. We use coordinate descent algorithm to find the estimated coefficients.[Fri+07] The following is the plot of quantile regression with L_1 penalty.

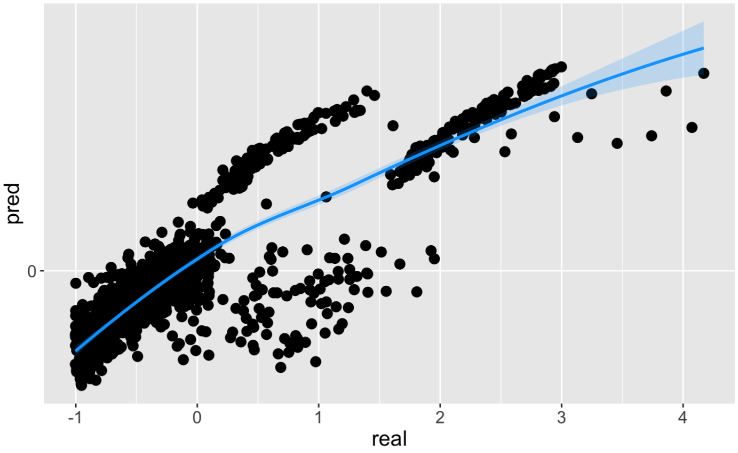

Figure 6: Real value vs predicted value

It is interesting that we can see from this plot that the fit shows a good linear relationship between real value and predicted value except for data points in the middle, especially for the paired data points in the middle and below the fitted curve. These points are sparser than other points. However, if we have a look at the value of mean squared errors and loss functions, we can conclude that as we increase the quantile, the model becomes better.

measure,τ	10%	50%	90%
MSE	0.7037813	0.6820604	0.6599731
Loss	413.6126	407.4874	401.2139

4 Summary

In this project, we introduce quantile regression and use a smooth check function to build it from scratch. The result is graphically satisfying but not numerically both from goodness of fit and processing time. As we increase the degrees of polynomial, the results will be close to the build in function. However, it still inspired me to think about data from non normal distribution, or heterogeneous data. We also try quantile regression with L_1 penalty and we try to use alternative method to estimate the coefficients instead of using loss function directly.

References

[KD87] Roger W Koenker and Valéria D'Orey. "Algorithm AS 229: Computing Regression Quantiles". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 36.3 (1987), pp. 383–393. DOI: 10.2307/2347802.

[Fri+07] Jerome Friedman et al. "Pathwise coordinate optimization". In: (2007).

[Zhe11] Songfeng Zheng. *Gradient descent algorithms for quantile regression with smooth approximation - International Journal of Machine Learning and Cybernetics.* July 2011. DOI: 10.1007/s13042-011-0031-2. URL: https://link.springer.com/article/10.1007/s13042-011-0031-2.

[LZ12] Youjuan Li and Ji Zhu. "L1-Norm Quantile Regression". In: *Journal of Computational and Graphical Statistics* 17 (Jan. 2012). DOI: 10.1198/106186008X289155.

[Cho18] Miri Choi. *Medical Cost Personal Datasets.* Feb. 2018. URL: https://www.kaggle.com/datasets/mirichoi0218/insurance.