Neural Network Approach to Estimating Conditional Quantile Polynomial Distributed lag (QPDL) model with an application to Rubber Price returns

Kwadwo Agyei Nyantakyi*
Postgraduate Institute of Agriculture, University of Peradeniya, Sri Lanka - knyantakyi@yahoo.com
B. L. Peiris
Postgraduate Institute of Agriculture, University of Peradeniya, Sri Lanka - laknathp@pdn.ac.lk
L. H. P. Gunaratne
Postgraduate Institute of Agriculture, University of Peradeniya, Sri Lanka - lhpguna@pdn.ac.lk

Abstract

In this paper we consider the estimation of the conditional quantile polynomial distributed lag (QPDL) using neural network to investigate the influence of the conditioning variables on the location, scale and shape parameters of the new QPDL model developed. This method avoids the need for a distributional assumption and applies conditional quantiles approach which allows the investigator to employ a range of conditional functions which exposes a variety of forms of conditional heterogeneity to give a more comprehensive picture of the effects of the independent variable on the dependent variable. The models fitted were adequate with high R-square values. Also from the actual and the predicted plots we observed that there was no significant difference between them. The results suggest that neural network used in estimating the QPDL model offers a useful alternative for estimating the conditional density, as the use of artificial neural networks in this study have also proven to produce good prediction results in regression problems.

Keywords: Backpropagation, feedforward, Hidden neurons, polynomial transformation.

Introduction

Koenker and Bassett (1978), introduced quantile regression to estimate the conditional quantile function of the response, by using the idea of generalized least absolute deviation regression. As a result, they provide much more information about the conditional distribution of a response variable.

Artificial neural networks represent an excellent tool that has been used to develop a wide range of real-world applications, especially when traditional methods of solving problems fail. This method avoids the need for a distributional assumption, (Scott, 1993).

In this paper we used artificial neural networks with simple architecture (one or two hidden layers) because they can approximate any function and its derivatives with any desired accuracy (Cybenko, 1989; Hornik et al., 1990; Hornik et al., 1993)

The objectives of this study are, to estimate the new Conditional Quantile Polynomial distributed lags model using neural network, to investigate the influence of the conditioning variables on the location, scale and shape parameters of the QPDL model, and to study the effects of rubber production on its price returns at different quantiles.

2.0 Materials and Methods

2.1 Data Source

Secondary annual data was collected from FAOSTAT, food balance sheet, price statistics, available with the Department of Census and Statistic Sri Lanka, and the World Bank (pink sheet). These data comprises of the production, imports, exports and prices of rubber. The rubber data ranges from 1961-2011.

2.2 **Statistical Software**

The R software, with the package quantile neural network (Qrnn) was used in fitting the conditional quantile polynomial distributed lag models.

2.3 Methodology

Given the autoregressive distributed lag

$$y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \beta_3 X_{t-3} + --- + \beta_n X_{t-k} + \varepsilon_t \quad .....(2.1)$$

With y_t dependent variable and X_t independent variable and $\{\varepsilon_t, 1 \leq t \leq k\}$ are independent identically distributed random errors.

$X_{t-1} = X_{t-2}, - - -, X_{t-k},$

$F_{t-1} = \sigma\{X_u, u \leq t-1\}$ the $\sigma - algebra$ generated by the observations up to time $t-1$.

The polynomial distributed lag by Almon, (1965) can be written as

$$y_t = \varphi + \sum_{i=0}^{k} \beta_i X_{t-i} + \varepsilon_t \quad ......(2.2)$$

With $k\ number\ of\ lags$, and the $\beta_i's$ can be approximated by suitable polynomials. That is

$$\beta_i = a_0 + a_1 i + a_2 i^2 + a_3 i^3 + --- + a_n i^n$$

For the n^{th} degree polynomial with $k\ number\ of\ lags$ we have

$$y_t = \varphi + \sum_{i=0}^{k}(a_0 + a_1 i + a_2 i^2 + a_3 i^3 + --- + a_n i^n) X_{t-i} + \varepsilon_t$$

$$y_t = \varphi + \sum_{i=0}^{k} a_0 X_{t-i} + \sum_{i=0}^{k} a_1 (i) X_{t-i} + \sum_{i=0}^{k} a_2 (i^2) X_{t-i} + \sum_{i=0}^{k} a_2 (i^2) X_{t-i} + \sum_{i=0}^{k} a_3 (i^3) X_{t-i} + - - - + \sum_{i=0}^{k} a_n (i^n) X_{t-i} + \varepsilon_t \quad --- (2.2)$$

The conditional $\alpha - quantile$ function (QPDL) can be written as

$Q_{y_t}(\alpha|Z_{0t},- - -,Z_{nt}) = \varphi(\alpha) + a_0(\alpha)Z_{0t} + a_1(\alpha)Z_{1t} + a_2(\alpha)Z_{2t} + a_3(\alpha)Z_{3t} + - - - + a_n(\alpha)Z_{nt} + Q_{yt}(\varepsilon_t)$. ………………..(2.3)

Where $Q_{yt}(\varepsilon_t) = u_t$.

For a QPDL (2) we have

$Y_t = \varphi(\alpha) + a_0(\alpha)Z_{0t} + a_1(\alpha)Z_{1t} + a_2(\alpha)Z_{2t} + u_t$. ------------------------------------- (2.4)

Neural Network

Let $\theta = (V_0,- - -,V_H,\alpha_1,- - -,\alpha_H,\beta_H)$ with $\alpha_j = (\alpha_{j1},- - -,\alpha_{jp})$ then

$f(Y_t,\theta) = V_0 + \sum_{h=1}^{H} V_h\varphi(<\alpha_h,Y_t> +\beta_h)$ ……………………..(2.5)

Denote a one layer feed forward neural network with H hidden neurons and $<,>$ is the classical scalar product on R^p. Assuming φ to be twice continuous differentiable and belongs to the class of sigmoid functions satisfying

$\lim_{x\to-\infty}\varphi(y) = 0$, $\lim_{x\to\infty}\varphi(y) = 1$ and $\varphi(y) + \varphi(-y) = 1$.

Example is the logistic function $\varphi(y) = (1 + e^{-y})^{-1}$

For the 2nd degree polynomial conditional quantile model given by (2.4)

$Y_t = \varphi(\alpha) + a_0(\alpha)Z_{0t} + a_1(\alpha)Z_{1t} + a_2(\alpha)Z_{2t} + u_t$

where

$Z_{it} = (Z_{0t},- - -,Z_{nt})$, $\theta_0$is a fixed but unknown, ε_t is independent of

$q_{t-1} = \sigma\{X_u, u \le t-1\}$, the sigma algebra generated by the observations up to time $t-1$.

Furthermore $\{\varepsilon_t, 1 \le t \le n\}$ are independent identically distributed random errors.

To estimate the conditional function q(Y_t) given the input

$Z_{it} = (Z_{0t},- - -,Z_{nt})^T \in R^d$, using neural network with one hidden layer consisting of $H \ge 1$ neurons defined by the feedforward function

$f(Y_t) = f_H(Y_t,\theta)$ of the following form

$f(Y_t,\theta) = V_{0j} + \sum_{h=1}^{H} V_h\varphi({Z_{it}}^T\omega_h, +\omega_{h0})$ …………………………………(2.6)

where $\omega_h = (\omega_{h1},- - -,\omega_{hd})^T$.

We consider a bipolar sigmoid function with range (-1,1) given by

$\varphi(y) = \frac{2}{1+\exp(-\sigma y)} - 1$ and $\varphi'(y) = \frac{\sigma}{2}[1 + f(y)][1 - f(y)]$ where σ is the steepness parameter. Then for the $Q_{y_t}(\alpha|Z_{it}) = {Z_{it}}^T\,\beta(\alpha)$ as defined before for QPDL(2), (2.4)

$Y_t = \varphi(\alpha) + a_0(\alpha)Z_{0t} + a_1(\alpha)Z_{1t} + a_2(\alpha)Z_{2t} + u_t$

Using the activation function

$\varphi(y_t) = \frac{2}{1+\exp(-\sigma y_t)} - 1$ and $\varphi'(y_t) = \frac{\sigma}{2}[1 + q(Z_{it})][1 - q(Z_{it})]$ we can estimate the neural network by using the following algorithm and the input feedforward

$(Y_t, t = 1, ---, n)$, and output target vector $(t = t_1, ---, t_k, ---, t_n)$, then for Z_j hidden unit z_j the net input to Z_j denoted by $z_{inj}; J = 1, ---, p$

$z_{inj} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$ where v_{0j} is the bias. The output signal (activation) of Z_j is denoted by z_j, $z_j = f(z_{inj})$.

Also for Y_k for the output unit K: the net input to Y_k denoted by

y_{ink} , $k = (1, ---, m)$, then

$y_{ink} = \omega_{0k} + \sum_{j=1}^{p} z_j \omega_{jk}$ with an output (activation) signal of Y_k given by

$y_k = f(y_{ink})$. This activation function was chosen because for a back propagation net, the activation function should be continuous, differentiable and monotonically non-decreasing and also easy to compute.

The backpropagation error

Each output unit $(Y_k, k = 1, ---, m)$ the error δ_k is calculated by,

$\delta_k = (t_k - y_k)f'(y_{ink})$ and the weight correction term ω_{jk} is updated later, where

$\Delta\omega_{jk} = \alpha\delta_k z_j$ And the bias correction term is $\Delta\omega_{0k} = \alpha\delta_k$ therefore for the sum of the errors for the hidden units $(Z_j, j = 1, ---, p)$ $\delta_{inj} = \sum_{k=1}^{m} \delta_k \omega_{jk}$ and multiplying by the derivative of the activation function we have , $\delta_j = \delta_{inj} f'(z_{inj})$ and the and the correction term weights are calculated by $\Delta v_{ij} = \alpha\delta_j x_i$ which update as $\Delta v_{ij} = \alpha\delta_j$

3.0 Results and Discussions

Neural network parameter estimates for the Conditional Quantile polynomial distributed lag (QPDL) with (3-lags) for rubber production, with Proz0 the first polynomial transformation for production, Proz1 the second polynomial transformation and Proz2 the third polynomial transformation.

Table 1 Conditional quantile polynomial distributed lag (QPDL) .25 Quantile Neural network parameter estimates, standard errors and confidence intervals for production and price of rubber

Quantile	Price	Coef	Std. Err.	P>|t|	[95% Conf. Interval]
0.25	Proz0	16.37	7.68	0.000	1.32, 31.35
	Proz1	-14.31	7.79	0.000	-29.59, -0.94
	Proz2	-1.32	1.16	0.010	-2.48, -0.156
	Constant	-1.41	1.20	0.005	-2.61, 0.21

R-square=0.72, AIC=263.40, RMSE=501.38, P>F=0.000

The resulting estimated equation from Table 1 is:

$\hat{Y}_t = -1.407 + 16.37Z_{0t} - 14.31Z_{1t} - 1.32Z_{2t}$ ……………..(3.1)

The resulting transformed QPDL model for 0.25 quantile neural network was estimated as

$\hat{Y}_t = -1.407 + 16.37X_0 + 0.74X_{t-1} - 17.53X_{t-2} - 38.43X_{t-3}$ ……………(3.2)

Similarly the QPDL estimated model for 0.50 quantile neural network was estimated as

$$\hat{Y}_t = -0.715 + 4.667X_0 + 0.944X_{t-1} - 2.592X_{t-2} - 5.944X_{t-3} \quad \ldots\ldots\ldots\ldots\ldots.(3.3)$$

The QPDL estimated model for 0.75 quantile neural network was estimated as

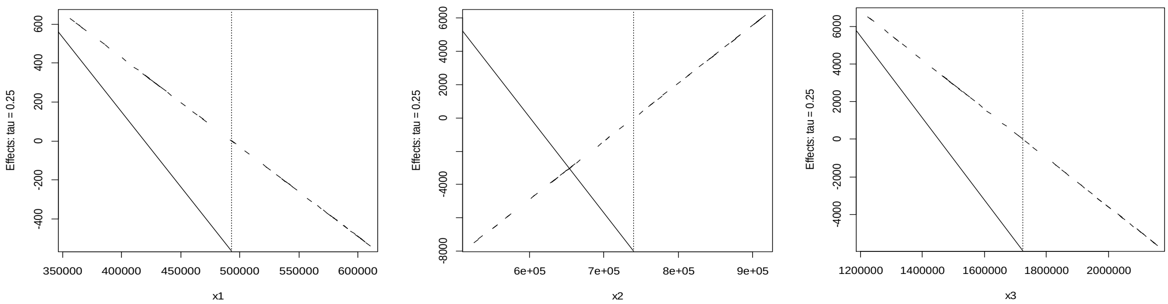
$$\hat{Y}_t = 3.529 - 15.261X_0 - 4.166X_{t-1} + 7.043X_{t-2} + 18.368X_{t-3} \quad \ldots\ldots\ldots\ldots\ldots \text{(3.4)}$$

The QPDL estimated model for 0.95 quantile neural network was estimated as

$$\hat{Y}_t = 2.329 - 7.727X_0 - 2.197X_{t-1} - 0.110X_{t-2} + 1.978X_{t-3} \quad \ldots\ldots\ldots\ldots\ldots \text{(3.5)}$$

Graphical Interpretation of the effects

GAM-style effect plots provide a graphical means of interpreting fitted predictor/predictor relationships. From Plate et al. (2000):. The effects are plotted as short line segments, and the slope of the segment is given by the partial derivative. Functions without interactions appear as possibly broken straight lines (linear functions) or curves (nonlinear functions).
Figure 1, 2 and 3 shows the plots of the effects of tau=0.25 on x1=proz0, x2=proz1 and x3=proz2

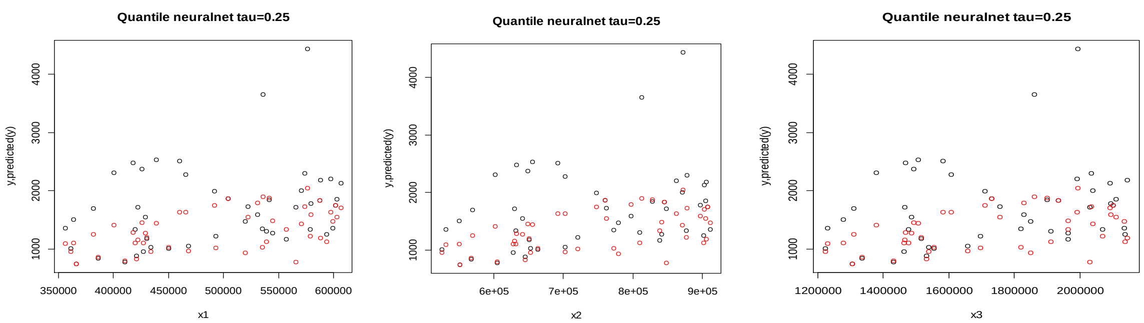


Figures 1, 2 and 3, plot of effects of tau=0.25 on x1, x2 and x3

From figures 1, 2 and 3 we observe that, x1, x2 and x3 all strongly influence the function value since they have a large total vertical range of effects values. Hence there is a large effect of the independent variable on the dependent variable with no interaction between x1 x2 and x3

Graphical Interpretation of the plots of actual and the predicted values

Figures 4, 5 and 6 shows the plots of actual values(black) and predicted values (red) against variables x1=proz0, x2=proz1 and x3=proz2.we present only the results for tau=0.25



Figures 4, 5 and 6, plot of (y, predicted(y)) against X1, X2 and X3

From figures 4, 5 and 6, we observe the plot of the actual (black) values and the predicted (red) values against the x1=proz0, x2=proz1 and x3=proz2, and there is no significant difference between the actual and the predicted values since they show similar clustering pattern.

Conclusions

We have presented a nonparametric approach to estimating the conditional polynomial distributed lag. Neural Network can learn to approximate any function just by using example data that is representative of the desired task. We observe that all the parameter estimates were significant. The models fitted were adequate with high R-square values. Also from the actual and the predicted plots we observed that there was no significant difference between them. We also observed that the quantile regression provides a more complete picture of the conditional distribution. The results suggest that neural network used in estimating the QPDL model offers a useful alternative for estimating the conditional density, as use of artificial neural networks in this study have also proven to produce good prediction results.

References

Barron, A. R. and Barron, R. L., Statistical learning networks: A unifying view, Wegman, E., editor, Computing Science and Statistics: Proc. 20th Symp. Interface, (American Statistical Association, Washington, DC, 1988) 192-203.

Cybenko, G., (1989), Approximations by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2 , 303-314.

FAOSTAT, Sri Lanka Annual Data (1961-2013), [on line].[Accessed on 10.02.2014]. Available at http://faostat.fao.org

Geman, S., Bienenstock, E., and Doursat, R., (1992), Neural networks and the bias-variance dilemma. Neural Computation, 4, 1-58.

Hornik, K., Stinchcombe, M., and White, H., (1990), Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Networks, 3, 551 - 560.

Hornik, K., Stinchcombe, M., White, H., and Auer, P., (1993), Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. (Discussion paper 93-15, Department of Economics, UCSD)

Koenker, R. W. and Bassett, G. W. (1982). Robust Tests for Heteroscedasticity based on Regression Quantiles, Econometrica, 50, 43–61.

Koenker, Roger and Zhijie Xiao (2002), “Inference on the Quantile Regression Process”, Econometrica, 81, 1583–1612.

Scott, G.M. (1993), Knowledge - Based Artificial Neural Networks for Process Modeling and Control. Ph.D. thesis, University of Wisconsin.

World Bank Pink Sheet Annual Data (1961-2013), [on line]. [Accessed on 08.08.2014]. Available at http://econ.worldbank.org