Comparative Study of Performance of Neural Networks with Other Non-Parametric Regression Estimators

Robert Kasisi

Jomo Kenyatta University, School of Mathematical Sciences, Department of Statistics and Actuarial Sciences
P.O Box 62000, Nairobi-Kenya

Abstract: Neural networks have drawn attention to researchers in recent years. This is because they show superiority as a modeling technique for datasets showing nonlinear relationships and thus for both data fitting and prediction abilities. In this study, we derive a neural network estimator of the finite population mean. This study shows that the mean square error values of the neural network estimator are minimal compared to those of other nonparametric estimators. This implies that neural networks are a better estimation technique for estimating population mean.

Keywords: Neural networks, nonlinear model, nonparametric regression, auxiliary information, survey sampling.

1. Introduction

Availability of auxiliary information to estimate parameters of interest of a survey variable has become very common. Such information is well available on census data, administrative registers and even on previous surveys. A population is the entire collection of identifiable units. Finite populations are of interest to government for policy making.

A simple way to incorporate known population totals of auxiliary variables is through ratio and regression estimation. Other general situations are handled by means of generalized regression estimation (Särndal, 1980) and calibration estimation (Deville and Särndal, 1992). Estimation procedures have been employed in getting information from the census data, administrative registers, and other surveys. However, in most cases these are challenging due to cost, time, literacy levels, and other geographical factors. In these methods, part of the population referred to as the sample is used, and the information about the population is inferred into the sample.

In this paper, we introduce a new type of nonparametric estimator for the finite population mean based on neural network learning.

2. A Neural Network Estimator

A neural network is a nonlinear model transforming real input variables into one or several output variables using several intermediate steps. The goal is to estimate the population mean of the survey, that is

\[ T = \frac{1}{N} \sum_{i=1}^{N} T_i \]

T is the survey variable and N is the size of the population.

Using calibration technique (Deville and Särndal, 1992), we can define our neural network estimator to be a linear combination of the observations

\[ f_\theta = \sum_{i=1}^{N} w_i T_i \]

With weights chosen to minimize an average distance measure from the basic design weights

\[ \hat{d}_i = \frac{1}{n_i} \]

Minimization is constrained to satisfy

\[ \frac{1}{N} \sum_{i=1}^{N} w_i x_i = \bar{x} \]

where \( \bar{x} \) is the known vector of population means for the auxiliary variables. Although alternative distance measures are available in Deville and Särndal (1992), all resulting estimators are asymptotically equivalent to the one obtained from minimizing the chi-squared distance

\[ \phi_2 = \sum_{i=1}^{N} (w_i - \hat{d}_i)^2 / d_i q_i \]

where the \( q_i \) are known positive weights unrelated to \( d_i \), i.e.

\[ t_{\text{nh}} = \hat{f} + (\bar{x} - \hat{x})^T \beta \]

where \( \hat{f}_{\text{nh}} \) and \( \hat{f} \) are the Horvitz-Thompson estimators of \( T \) and \( \bar{x} \), respectively, and

\[ \beta = \left( \sum_{i=1}^{N} d_i q_i x_i \right) \left( \sum_{i=1}^{N} d_i q_i T_i \right)^{-1} \]

Consider the following super-population model

\[ E_\epsilon(T_i) = f(x_i)^2 \quad \text{for} \ i = 1, 2, \ldots, N \]

\[ V_\epsilon(T_i) = \sigma^2 f(x_i)^2 \quad \text{for} \ i = 1, 2, \ldots, N \]

\[ C_\epsilon(T_i, T_j) = 0 \quad \text{for} \ i \neq j \]

Where \( E_\epsilon \) and \( V_\epsilon \) denote expectation and variance, respectively, with respect to \( \epsilon \); \( f(x_i) \) takes the form of a feed forward neural network with skip-layer connections and \( V(\cdot) \) is a known function of \( x_i \). Hence,

\[ f(x_i) = \sum_{q=1}^{Q} \beta_q x_{i,q} + \sum_{m} \alpha_m \sigma(\sum_{q=1}^{Q} \gamma_{qm} x_{i,q} + \gamma_{rm}) + \alpha_0 \]

(1)

M is the number of neurons at the hidden layer (Ripley, 1996, Chapter 5). Since we consider \( M \) as fixed, we can denote the set by all parameters of the network, and write \( f(x_i) \) in (1) as \( f(x_i, \theta) \).
\[ \theta = \{ \theta_1, \ldots, \theta_p, \alpha_1, \ldots, \alpha_M, \gamma_{0M}, \gamma_{1M}, \ldots, \gamma_{pM} \} \]

From Wu and Sitter (2001) to estimate \( \hat{\theta} \), the first step is to obtain a design-based method for estimating the model parameters and therefore obtain estimates of the regression function at \( x_i \) for \( i=1, \ldots, N \), through the resulting fitted values. In other words, we first seek for an estimate \( \hat{\theta} \) of the model parameters \( \theta \) based on the data from the entire finite population. We then obtain \( \hat{\theta} \) a design-based estimate of \( \theta \) based on the sampled data only. The population parameter \( \theta \) is defined by weighted least squares with a weight decay penalty term, i.e.

\[
\hat{\theta} = \text{argmin}_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (T_i - f(x_i, \theta))^2 + \frac{\lambda}{N} \sum_{i=1}^{N} \theta_i^2 \right\} \tag{2}
\]

Where \( \lambda \) is a tuning parameter and \( p \) is the dimension of the parameters vector \( \theta \). The estimate \( \hat{\theta} \) is defined as the solution of the design-based sample version of (2), that is

\[
\hat{\theta} = \text{argmin}_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^{n} (T_i - f(x_i, \theta))^2 + \frac{\lambda}{N} \sum_{i=1}^{N} \theta_i^2 \right\} \tag{3}
\]

Once the estimates \( \hat{\theta} \) are obtained, the available auxiliary information is included in the estimator through the fitted values \( \hat{f} = f(x_i, \hat{\theta}) \) for \( i=1, \ldots, N \). Then, we can define the neural network estimator as

\[
\hat{\theta}_{nn} = \frac{1}{n} \sum_{i=1}^{n} w_i x_i, \quad \text{where the calibrated weights} \quad w_i \text{ \text{are sought to minimize the distance measure} } \quad \hat{T}_i = \frac{1}{n} \sum_{i=1}^{n} w_i x_i - f(x_i, \hat{\theta}) \tag{4}
\]

Using the technique of Deville and Särndal (1992) to derive the optimal weights, we can propose that

\[
\hat{T}_{nn} = \hat{T}_{nn} + \frac{1}{n} \sum_{i=1}^{n} \left( \sum_{i=1}^{n} w_i x_i - \frac{1}{n} \sum_{i=1}^{n} w_i x_i \right) \tag{5}
\]

Where

\[ f = \sum_{i=1}^{n} f(x_i, \hat{\theta}) \quad \text{And} \quad \hat{f} = \sum_{i=1}^{n} f(x_i, \hat{\theta}) \]

We wish to combine the kernel technique to our neural network estimation. Therefore we briefly describe kernel smoothing.

A continuous kernel is denoted as \( k(.) \) and the bandwidth as \( h \). The conditional regression estimator \( \mu(.) \) is the solution to a natural weighted least squares problem being the minimizer \( \hat{\beta}_e \) of

\[
S(\beta_e) = \sum_{i=1}^{n} \left( T_i - \hat{\mu}_{\beta_e} \right)^2 k \left( \frac{x_i - \bar{x}}{h} \right) \]

\[
= \sum_{i=1}^{n} \left( T_i - \beta_0 - \beta_1 x_i \right)^2 w_i \tag{6}
\]

Where

\[ w_i = k \left( \frac{x_i - \bar{x}}{h} \right) \]

By differentiating equation (6) with respect to \( \beta_e \) and equating to zero we get

\[
\frac{\partial S(\beta_e)}{\partial (\beta_0)} = 0
\]

\[
-2 \sum (T_i - \hat{\beta}_0) w_i = 0
\]

\[
\sum (x_i - \hat{\beta}_0) w_i = 0
\]

\[
\sum (T_i) w_i = -\beta_0 \sum w_i
\]

\[
\hat{\mu}_{\beta_e} - \beta_0 \sum_{i=1}^{n} w_i
\]

\[
\sum_{i=1}^{n} k \left( \frac{x_i - \bar{x}}{h} \right) T_i
\]

\[
= \sum_{i=1}^{n} k \left( \frac{x_i - \bar{x}}{h} \right) \frac{T_i}{\sum_{i=1}^{n} k \left( \frac{x_i - \bar{x}}{h} \right)}
\]

For a target \( x_j, j = 1, 2, \ldots, N \), we have

\[
\hat{\mu}(x_j) = \frac{\sum_{i=1}^{n} w_i T_i}{\sum_{i=1}^{n} k \left( \frac{x_i - \bar{x}}{h} \right)}
\]

Similarly

\[
\hat{\mu}(x_j) = \frac{\sum_{i=1}^{n} k \left( \frac{x_i - x_j}{h} \right) T_i}{\sum_{i=1}^{n} k \left( \frac{x_i - x_j}{h} \right)}
\]

So that

\[
\hat{\mu}(x_j) = \frac{\sum_{i=1}^{n} T_i k \left( \frac{x_i - x_j}{h} \right)}{\sum_{i=1}^{n} k \left( \frac{x_i - x_j}{h} \right)}
\]

\[
= \sum_{i=1}^{n} w_i x_i
\]

\[
\text{i.e.} \quad \hat{\mu}(x_j) \text{ is an approximation of } \mu(x_j) \text{ with a constant weighting value of } T \text{ corresponding to } x_i \text{'s closest to } x_j \text{ more heavily. Alternatively, let } T = \begin{bmatrix} T_1 \end{bmatrix} \in \mathbb{R} \text{ be the n vector of } y_i \text{'s obtained in the sample. Define the } n \times 1 \text{ matrix }
\]

\[
X_{ij} = \begin{bmatrix} x_i \end{bmatrix} \text{ and define the } n \times n \text{ matrix }
\]

\[
W_{ij} = \frac{1}{h} \text{ diag } \left( \begin{bmatrix} k \left( \frac{x_i - x_j}{h} \right) \end{bmatrix} \right)
\]

Then a sample based estimator of \( \mu(x_j) \) is given by

\[
\hat{\mu}(x_j) = \begin{bmatrix} (X_{ij} W_{ij} X_{ij})^{-1} X_{ij} W_{ij} X_{ij} \end{bmatrix} = \hat{\omega}_j T
\]

As long as \( X_{ij} W_{ij} X_{ij} \) is invertible. It follows that

\[
(X_{ij} W_{ij} X_{ij})^{-1} X_{ij} W_{ij} X_{ij} = \hat{\omega}_j T
\]

\[
\sum_{i=1}^{n} k \left( \frac{x_i - x_j}{h} \right) T_i
\]

\[
= \sum_{i=1}^{n} k \left( \frac{x_i - x_j}{h} \right)
\]

\[
\text{We note that we can use the neural network package (nnet) method to obtain the mean function of the fitted values. From the kernel technique,}
\]

\[
\mu(x_j) = \begin{bmatrix} x_j \end{bmatrix} \text{ is subjected to the network and learnt.}
\]

Therefore \( f = \mu(x_j) \) and \( \hat{\mu}(x_j) = \begin{bmatrix} x_j \end{bmatrix} \text{ is subjected to the network.}
\]

In other words \( y, \hat{\mu} = \hat{\mu}(x_j) \)
3. Data Analysis

Using R statistical package we simulate two populations of x as independent and identically distributed uniform (0, 1) and gamma (1,1) random variables.

The populations are of size N=300. Samples of size n=30 are generated by simple random sampling. The population size is considered large enough for several samples and the sample size is 10 percent of population size. For each population of x, mean function, and bandwidth, 100 replicate samples are generated and the estimates calculated. The population is kept fixed during these 100 replicates in order to be able to evaluate the design averaged performance of the estimators. We consider four mean functions:

1. Linear $2 + 5x$
2. Quadratic $(2 + 5x)^2$
3. Exponential $\exp(-8x)$
4. Cycle $1 + \sin(2\pi x)$

We report on some performance of several estimators. The Epanechnikov kernel

$$k(u) = \frac{3}{4}(1 - u^2), \ u \leq 1$$

is used for all four nonparametric estimators. Several bandwidths are considered (h=0.1, h=0.25, h=0.5, h=0.75, h=1 and h=2) to help see how efficiency of the estimators vary with bandwidth. The second bandwidth is based on the ad hoc rule of $\frac{1}{4\sqrt{d}}$ where d is the data range. The bandwidths h=1 and h=2 are large bandwidths relative to the data range, [0,1].

For the linear mean function, $T_{nn}$ and $T_{lp}$ the results show equal performance evident from equal mean squared errors for both uniform and gamma distributions. We therefore examine how much efficiency is lost if we use the other estimators. The other means functions represent departures from the linear model.

For quadratic function $T_{nn}$ performs better followed by $T_{lp}$ (linear), except for a small portion for the range of x i.e. for(h=0.1,h=0.5, and h=0.75) $T_{lp}$ (linear) performs better under the gamma distribution. The biases at these turning points for $T_{lp}$ (linear) are seen to be less compared to those of $T_{nn}$. For the exponential mean function under uniform distribution, $T_{nn}$ performs better followed by $T_{lp}$(linear). It is interesting to see the cycle and exponential mean functions yield similar MSE values under gamma distribution.

The performance of any estimator $\hat{f}$ in $[T_{nn}, T_{lp}, T_{loclp}, T_{lp}]$ is evaluated using its relative bias $R_{f}$ and MSEs. The relative bias is defined as

$$R_{f} = \frac{\Sigma_{i=1}^{R} (\hat{f}_i - f)}{R \times T}$$

R is the replicate number of samples. We evaluate the actual design variance and estimated the mean squared error as $MSE(\hat{f}) = \text{var}(\hat{f}) + (R_{f})^2$

We also consider an estimate of the mean square error $\text{MSE}(\hat{f}) = \frac{\Sigma_{i=1}^{R} (\hat{f}_i - f)^2}{R}$

Where $\hat{f}_i$ is calculated from the $R^{th}$ simulated sample.

4. Table of Results

- nn: neural network
- loclp: Local-polynomial
- nw: Nadaraya-Watson estimator

Table1: Comparative MSEs for the nonparametric estimators for a sample size n=30 under uniform distribution

<table>
<thead>
<tr>
<th>Distribution</th>
<th>MSE of nn</th>
<th>MSE of loclp</th>
<th>MSE of local linear</th>
<th>MSE of nw</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear</td>
<td>216.5325</td>
<td>168.6484</td>
<td>216.5325</td>
<td>216.5325</td>
</tr>
<tr>
<td>Quadratic</td>
<td>44.0644</td>
<td>50.3618</td>
<td>51.1950</td>
<td>604.57</td>
</tr>
<tr>
<td>Exponential</td>
<td>160.5052</td>
<td>210.0316</td>
<td>198.8392</td>
<td>397.6874</td>
</tr>
<tr>
<td>cycle2</td>
<td>160.5052</td>
<td>210.0316</td>
<td>198.8392</td>
<td>397.6874</td>
</tr>
</tbody>
</table>

Table 2: Comparative MSEs for the nonparametric estimators for a sample size n=30 under gamma distribution

<table>
<thead>
<tr>
<th>Distribution</th>
<th>MSE of nn</th>
<th>MSE of local polynomial</th>
<th>MSE of local linear</th>
<th>MSE of nw</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear</td>
<td>881.1422</td>
<td>1179.85</td>
<td>881.1422</td>
<td>1098.883</td>
</tr>
<tr>
<td>Quadratic</td>
<td>2243.97</td>
<td>251829.8</td>
<td>628123.9</td>
<td>81123.79</td>
</tr>
<tr>
<td>Exponential</td>
<td>887.0592</td>
<td>883.1512</td>
<td>934.2882</td>
<td>837.5117</td>
</tr>
<tr>
<td>cycle2</td>
<td>990.9664</td>
<td>835.5993</td>
<td>721.5811</td>
<td>708.9674</td>
</tr>
</tbody>
</table>

Table3: Comparative MSEs for the nonparametric estimators for a sample size n=15 under uniform distribution

<table>
<thead>
<tr>
<th>Distribution</th>
<th>MSE of nn</th>
<th>MSE of local polynomial</th>
<th>MSE of local linear</th>
<th>MSE of nw</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear</td>
<td>216.5325</td>
<td>256.4107</td>
<td>216.5325</td>
<td>437.14</td>
</tr>
<tr>
<td>Quadratic</td>
<td>326.0076</td>
<td>228.7917</td>
<td>6982688</td>
<td>381.1422</td>
</tr>
<tr>
<td>Exponential</td>
<td>183.5767</td>
<td>218.4183</td>
<td>206.824</td>
<td>1052.611</td>
</tr>
<tr>
<td>cycle2</td>
<td>183.5617</td>
<td>218.4183</td>
<td>206.824</td>
<td>413.611</td>
</tr>
</tbody>
</table>

Table4: Comparative MSEs for the nonparametric estimators for a sample size n=15 under gamma distribution

<table>
<thead>
<tr>
<th>Distribution</th>
<th>MSE of nn</th>
<th>MSE of local polynomial</th>
<th>MSE of local linear</th>
<th>MSE of nw</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear</td>
<td>884.4434</td>
<td>231829.8</td>
<td>81123.79</td>
<td>7371.174</td>
</tr>
<tr>
<td>Quadratic</td>
<td>21805.35</td>
<td>178603.5</td>
<td>2627501</td>
<td>897.6784</td>
</tr>
<tr>
<td>Exponential</td>
<td>843.3734</td>
<td>1343.317</td>
<td>910.0316</td>
<td>397.0045</td>
</tr>
<tr>
<td>cycle2</td>
<td>783.988</td>
<td>884.443</td>
<td>659.874</td>
<td>397.0045</td>
</tr>
</tbody>
</table>
consideration all the mean functions, then the artificial neural network is much better performer.

5. Conclusions

The aim of this study was to compare the performance of a neural network estimator with other nonparametric estimators. This was achieved. Considering the MSEs of the various estimators, we make several observations. $T_{nn}$ performs exceptionally well under linear and quadratic functions. Also, $T_{loc}$ performs well since it’s itself linear, and hence is almost a true model for the linear function. $T_{nn}$ retained consistent efficiencies in most of the other mean functions.

The only closest competitor of the neural network estimator is the linear local polynomial estimator. However our estimator is more applicable since we do not have to determine the degrees to use. We have also found that if the mean $\mu(x)$ of a sample is known, then we can use this information to find the mean of the non-sampled elements which leads to overall population mean estimation. Our objectives have been achieved that the artificial neural network estimator outperforms kernel estimators and also

References


Author Profile

Robert Kasisi received a BSc in Mathematics and Computer Science from Jomo Kenyatta University of Agriculture and Technology in the year 2011; and a Master of Science in Applied Statistics from Jomo Kenyatta University of Agriculture and Technology in the year 2013. He is currently a Ph.D. student in Applied Statistics in Jomo Kenyatta University of Agriculture and Technology