International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

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

Robert Kasisi[1]

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 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. More general situations are handled by means of generalized regression estimation (Sarndal, 1980) and calibration estimation (Deville and Sarndal, 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

$$\bar{T} = \frac{1}{N}\sum_{i=1}^{N} T_i$$

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

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

$$\hat{\bar{T}} = \sum_{i=1}^{N} w_i T_i$$

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

$$d_i = \frac{1}{\pi_i}$$

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 S¨arndal (1992), all resulting estimators are asymptotically equivalent to the one obtained from minimizing the chi-squared distance

$$\emptyset_s = \frac{\sum_{i\in s}(w_i - d_i)^2}{d_i q_i}$$

Where the $q_i{}'s$ are known positive weights unrelated to d_i , i.e.

$$\hat{\bar{T}}_{nn} = \hat{\bar{T}} + (\bar{x} - \hat{\bar{x}})^T \hat{\beta}$$

Where $\hat{\bar{T}}_{nn}$ and $\hat{\bar{x}}$ are the Horvitz-Thompson estimators of $\bar{T}$ and $\bar{x}$, respectively, and

$$\hat{\beta} = (\sum_{i=1}^{n} d_i q_i x_i x_i')\sum_{i=1}^{n} d_i q_i T_i$$

Consider the following super-population model

$$\begin{cases} E_\varepsilon(T_i) = f(x_i) & for\ i = 1,2,\cdots,N \\ V \in (T_i) = \sigma^2 v(x_i)^2 & for\ i = 1,2,\cdots N \\ C_\varepsilon(T_i T_j) = 0 & for\ i \neq j \end{cases}$$

Where E_ε and V_ε denote expectation and variance, respectively, with respect to ε; $f(x_i)$ takes the form of a feed forward neural network with skip-layer connections and $V(.)$ is a known function of x_i. Hence,

$$f(x_i) = \sum_{q=1}^{Q}\beta q\ x_{qi} + \sum_{m}^{M} a_m \emptyset\left(\sum_{q=1}^{Q}\gamma_{qm}\, x_{qi} + \gamma_{0m}\right) + a_0 \quad (1)$$

M is the number of neurons at the hidden layer (Ripley, 1996, Chapter 5). Since we consider M as fixed, we can denote by the set of all parameters of the network, and write $f(xi)$ in (1) as $f(xi, \theta)$,

$\theta = \{\beta_1, \cdots, \beta_q, a_0, a_1, \cdots, a_M, \gamma_{0M}, \gamma_1, \cdots, \gamma_M\}$

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

$$\tilde{\theta} = argmin_{\theta\,\epsilon\,\emptyset}\left\{\frac{1}{v(x_i)^2}(T_i - f(x_i,\theta))^2 + \frac{\lambda}{N}\sum_{i=1}^{p}\theta_i{}^2\right\} \quad (2)$$

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

$$\hat{\theta} = argmin_{\theta\,\epsilon\,\emptyset}\left\{\frac{1}{\pi_i}\left[\frac{1}{v(x_i)^2}(T_i - f(x_i,\theta))2 + \frac{\lambda}{N}\sum_{i=1}^{p}\theta_i{}^2\right]\right\} \quad (3)$$

Once the estimates $\hat{\theta}$ are obtained, the available auxiliary information is included in the estimator through the fitted values $\hat{f} = f(x_i,\hat{\theta}), for\ i = 1,2,\cdots N$.Then, we can define the neural network estimator as $\hat{\bar{T}}_{nn} = \frac{1}{N}\sum_{i=1}^{N} w_i x_i$ where the calibrated weights w_i are sought to minimize the distance measure $\emptyset_s$ subject to $\frac{1}{N}\sum_{i=1}^{N} w_i = 1$ and

$$\frac{1}{N}\sum_{i=1}^{n} w_i \hat{f}_i = f(x_i,\hat{\theta})$$

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

$$\hat{\bar{T}}_{nn} = \hat{\bar{T}}_{nn} + \frac{1}{N}\left\{\sum_{i=1}^{N}\hat{f}_i - \sum_{i=1}^{n} d_i \hat{f}_i\right\} \quad (4)$$

Where

$$\bar{T} = \frac{\sum_{i=1}^{n} d_i q_i T_i}{\sum_{i=1}^{n} d_i q_i} \text{ And } \bar{f} = \frac{\sum_{i=1}^{n} d_i q_i \hat{f}_i}{\sum_{i=1}^{n} d_i q_i}$$

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 $\mu(x)$is the solution to a natural weighted least squares problem being the minimizer $\hat{\beta}_0$ of

$$S(\beta 0) = \sum_{i=1}^{n}(T - \beta_0)^2 k\left(\frac{x-xi}{h}\right) \quad (5)$$

$$= \sum_{i=1}^{n}(T_i - \beta_0)^2 w_i \quad (6)$$

Where

$$w_i = k\left(\frac{x - x_i}{h}\right)$$

By differentiating equation (6) with respect to β_0 and equating to zero we get

$$\frac{\partial s(\beta_0)}{\partial(\beta_0)} = 0$$

$$-2\sum(T_i - \beta_0)w_i = 0$$

$$\sum(T_i - \beta_0)w_i = 0$$

$$\sum(T_i)w_i = -\beta_0\sum w_i$$

$$\hat{\mu}(xj) = \hat{\beta}_0 = \frac{\sum_{i=1}^{n} w_i T_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} k\left(\frac{x - x_i}{h}\right)T_i}{\sum_{i=1}^{n} k\left(\frac{x - x_i}{h}\right)}$$

For a target $x_j, j = 1,2,\cdots,N$, we have

$$\hat{\mu}(xj) = \hat{\beta}_0 = \frac{\sum_{i=1}^{n} w_{ij} T_i}{\sum_{i=1}^{n} w_{ij}}$$

$$= \frac{\sum_{i=1}^{n} k\left(\frac{x_i - x_{ij}}{h}\right)T_i}{\sum_{i=1}^{n} k\left(\frac{x_i - x_{ij}}{h}\right)}$$

Similarly

$$w_{ij} = \frac{k\left(\frac{x_i - x_j}{h}\right)}{\sum_{i=1}^{n} k\left(\frac{x_i - x_j}{h}\right)}$$

So that

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

$$= \sum_{i=1}^{n} w_{ij}\, T_i$$

i.e. $\hat{\mu}(x_j)$ is an approximation of $\mu(x_j)$with a constant weighting value of T corresponding to x_i's closest to x_j more heavily. Alternatively, let T_s=$[T_i]i \in s$ be the n vector of y_i's obtained in the sample. Define the $n \times 1$ matrix $X_{sj} = [1]_{n\times 1}$and define the $n \times n$ matrix

$$w_{sj} = \frac{1}{h} diag\left\{k\left(\frac{x - x_i}{h}\right)\right\}_{i\epsilon s}$$

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

$$\hat{\mu}(x_j) = \left(X'_{sj}\, W_{sj} X_{sj}\right)^{-1} X'_{sj} W_{sj} X_{sj} = \hat{W}_{sj} T_s$$

as long as $X'_{sj}\, W_{sj}\, X_{sj}$ is invertible.

It follows that $\left(X'_{sj} W_{sj} X_{sj}\right)^{-1} X'_{sj} W_{sj} X_{sj} = \hat{W}_{sj} T_s$

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

$$= \sum_{i=1}^{n} w_{ij}\, T_i$$

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,

$$\hat{\mu}(x_j) = \left(X'_{sj} W_{sj} X_{sj}\right)^{-1} X'_{sj} W_{sj} X_{sj}$$

The weights W_{sj} are subjected to the network and learnt.

Therefore $\hat{f}_i = \hat{\mu}(x_j) = \left(X'_{sj} y.hat X_{sj}\right)^{-1} X'_{sj}\, y.hat_{sj}\, T_s$

In other words $y.hat = k\left(\frac{x_i - x_j}{h}\right)$

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. $Cycle1\ 2 + sin(2\pi x)$

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 $\frac{1}{4}th$ 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 used 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, $\hat{T}$ in $\{T_{lnr}, T_{nw}, T_{nn}, T_{lp}\}$ is evaluated using its relative bias R_B and MSEs.The relative bias is defined as

$$R_B = \frac{\sum_{r=1}^{R}(\hat{T} - T)}{R \times T}$$

R is the replicate number of samples. We evaluate the actual design variance and estimated the mean squared error as

$$MSE(\hat{T}) = var(\hat{T}) + (R_B)^2$$

We also consider an estimate of the mean square error

$$MSE(\hat{T}) = \frac{\sum_{r=1}^{R}(\hat{T}_r - T)^2}{R}$$

Where $\hat{T}_r$ 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

uniform	MSE of nn	MSE of loclp	MSE of local linear	MSE of nw
linear	216.5325	168.6484	216.5325	216.5325
Quadratic	44.0644	50.36182	51.19504	604.57
Exponential	160.5052	210.0316	198.8392	397.6784
cycle2	160.5052	210.0316	198.8392	397.6784

Description
Local polynomial estimator performs better than the other estimators under a linear mean function. But taking into consideration all the mean functions then the neural network is much better.

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

gamma	MSE of nn	MSE of local polynomial	MSE of local linear	MSE of nw
linear	881.1422	11978.5	881.1422	1098.883
Quadratic	22431.97	231829.8	628123.9	81123.79
Exponential	887.0592	883.1512	934.2882	837.5117
cycle2	990.9664	835.5993	721.5811	708.9674

Description
Local polynomial estimator performs better than the other estimators under a linear mean function. But taking into consideration all the mean functions then the neural network is much better performer.

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

uniform	MSE of nn	MSE of local polynomial	MSE of local linear	MSE of nw
linear	216.5325	256.4107	216.5325	437.14
Quadratic	326.0076	228.7917	6.982688	381.1422
Exponential	183.5677	218.4183	206.824	1052.611
cycle2	183.5617	218.4183	206.824	413.611

Description
The neural network estimator and Linear Local polynomial estimator performs almost equally under linear mean function which is better than the other estimators. Taking into consideration all the mean functions, then the artificial neural network is much better performer.

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

gamma	*MSE of nn*	*MSEof local polynomial*	*MSE_of local linear*	*MSE of nw*
linear	881.1422	10687.12	881.1422	897.6784
Quadratic	21805.35	1786035	2627501	1787371
Exponential	865.6058	843.3734	1343.317	910.0316
cycle2	783.988	884.443	659.8745	397.0045

Description
The neural network estimator and Linear Local polynomial estimator performs almost equally under linear mean function, better than the other estimators. Taking into

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_{loclp} 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

[1] Ripley B.D (1996) Pattern recognition and Neural Networks, Cambridge University Press, Cambridge.
[2] Breidt, F.J and Opsormer, J.D.(2000).Local polynomial regression estimators in survey sampling. *Annals of Statistics.28, 1026-1053.*
[3] Wu, C. and Sitter, R. R (2001).Efficient estimation of quadratic finite population functions in the presence of auxiliary information. *Journal of American Statistical Association, 97, 535-43.*
[4] William G. Cochran (1992),Sampling Techniques, *third edition,44-49,364-382*
[5] Godambe, V. P (1995) A Unified Theory of Sampling from finite populations. *journal of Royal Statistical Society.B,17,267-278*
[6] Wu C. and Sitter R.(2001) A model-calibration to using complete auxiliary information from survey data, *Journal of American Statistical Association, 185-193.*

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