International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 10 ǁ October 2022 ǁ PP. 414-419

Differences and Similarities between Support Vector Regression (SVR) and Neural Network (NN)

Jude Chukwura Obi
Department of Statistics,
Chukwuemeka Odumegwu Ojukwu University, Anambra State, Nigeria

Abstract
The support vector regression and the neural networks are both nonparametric regression procedures. Both regression tools have modifications for use as classification tools. This study, leveraging on these similarities, sought to under the behaviours of both regression tools on data. A total of 12 datasets were used in the study and the result of data analysis shows that there is no difference in the performances of both regression tools based on the datasets used in the study.
Keywords*: Support Vector Regression, Neural Networks, Supervised Learning, Predictive Modelling, Multiple Regression*

Date of Submission: 02-10-2022 Date of acceptance: 15-10-2022

I. Introduction

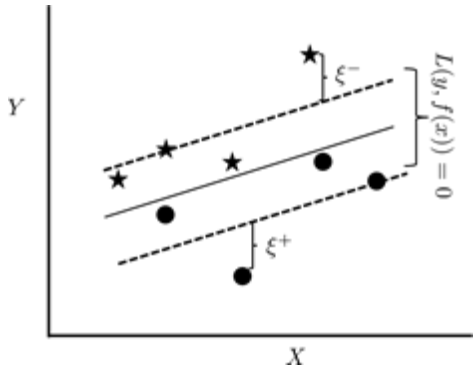
The support vector regression and the neural network are nonparametric tools, becauseno parametric assumptions are made prior to using any of them. The neural network can be used as a classification and regression techniques, but my focus is using it as a technique for solving regression problems. It should be noted that both SVR and NN can be used on datasets that meet parametric assumptions, and in some cases, outperform some parametric tools.They can perform optimally with nonlinear datasets because of their ability to generate nonlinear decision boundaries. On the other hand, while the support vector regression will not rely on all the explanatory variables in predicting the response variables, the neural network utilizes all the explanatory variables in this respect. Their individual approaches to predicting the response variables differ and in the subsections that follows,we shall dwell on these facts.

1.1 Support Vector Regression

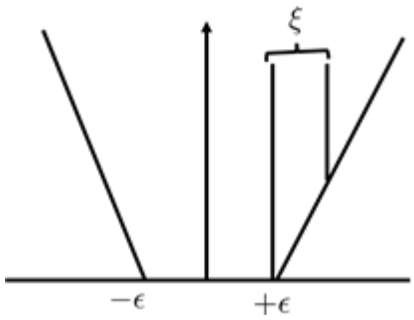
The Support Vector Regression(Cortes &Vapnik, 1995)is a regularization technique, by virtue of the fact that it can be used to correct an ill posed regression problem. The construction of the SVR model is based on ε-insensitiveloss with a threshold ε set by the user(Awad& Khanna, 2015). The SVR will leave behind a sparse model (a model with fewer explanatory variables than the original number of input variablesin the study). As presented in(Obi, 2020), when a threshold is specified by the user, the samplepoints with small residuals and within a specified threshold do not have effect on theregression model. Now, if you set a thresholdto be large comparatively, it willbe discovered that only the outliers will form the points that eventually defines the SVRmodel. The ε-insensitive loss function can be defined as follows:

$$L\big(y, f(x)\big) = \begin{cases} 0 & if\ |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & Otherwise \end{cases} \tag{1.1}$$

Based on (1.1), the SVR model is not sensitive where the error $\big(|y - f(x)\big)|$is smallerthan a specified threshold ε, otherwise it is sensitive. Further illustration on the εinsensitive loss function is given in Figure 1 that follows:



(a) The distance between margin lines, and data pointsoutside the margin lines is denoted by ξ. The data pointswithin the margin lines are disregarded in the construction ofthe regression model. The support vectors are those datapoints on the margin lines.



(b) The graph of ε-insensitive loss function. Here, *errors*∈(-ϵ, +ϵ) are not regarded. But outside the interval, the regression algorithm is sensitive to such errors.

Figure 1: Graphical illustrations on ε-insensitive loss function

1.1.1 Optimization Procedure

The optimization procedure of the support vector regression is used to find optimized support vectorregression model given as:

$$f(\underline{\mathbf{x}}) = \sum_{i \in sv}^{n} \alpha_i y_i K(x_i, \underline{\mathbf{x}}) + w_0. \tag{1.2}$$

Prior to arriving at (1.2), we first

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 \tag{1.3}$$

$$\text{subject to} \begin{cases} y_i - \mathbf{w}^T\mathbf{x}_i \leq \epsilon \\ \mathbf{w}^T\mathbf{x}_i - y_i \leq \epsilon \end{cases}. \tag{1.4}$$

Note that the assumption implied in (1.4) is that a convex optimization problem is feasible. Now, withdeviation outside the ε-insensitive region, this may not be the case. Hence, slack variables are introduced in the SVR, thus the optimization problem becomes:

$$\text{minimize} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-). \tag{1.5}$$

$$\text{subject to} \begin{cases} y_i - (\mathbf{w}^T\mathbf{x}_i + w_0) \leq \epsilon + \xi^+ \\ (\mathbf{w}^T\mathbf{x}_i + w_0) - y_i \leq \epsilon + \xi^-. \\ \xi_i^+\xi_i^- \qquad\qquad \geq 0 \end{cases} \tag{1.6}$$

Now, C is a cost parameter that determines the trade-off between how flat the function $f$is, and amount of deviation larger than ε that is accepted. It is important to note that entire optimization procedure can be carried out using the e1071 package(Meyer et al., 2019) in R. The Appendix contains the complete program in R on how to use the package.

1.2 Regression Neural Network

Neural Network (NN), otherwise called Artificial Neural Network (ANN) has a structure that mimics the human brain.It consists of the input, one or more hidden layers and the output. The input and output are interconnected, with each connection having a weight that associates with it. The NN can be used to model both linear and nonlinear datasets, but it performs optimally when nonlinear datasets are involved. We can use the NN model to solve problems on both regression and classification.A schematic diagram of the network is contained in Figure 2.

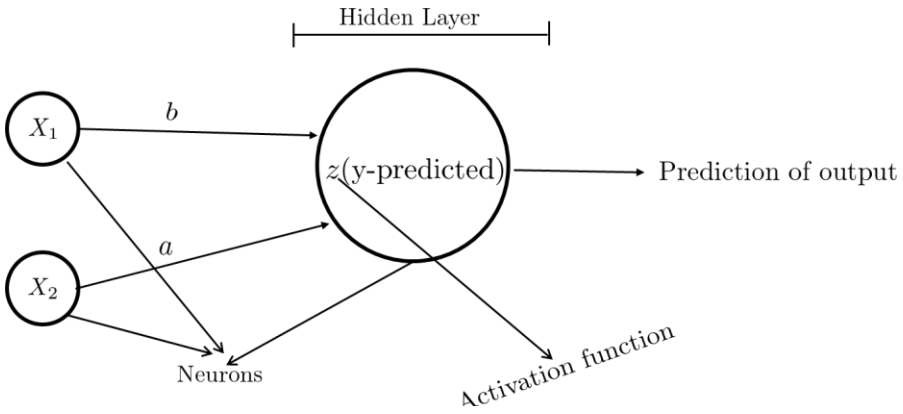


Figure 1: Diagram of a simple Neural Network without complexities.

Figure2 is a NN diagram that can be used to model a regression problem whose linear regression model is given by $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$. The components of the NN as revealed in Figure 2 are the neurons, weights, input layer, output layer and lastly the hidden layer. The weights are a and b, and the input layer consists of the independent variables which in this case are X_1 and X_2. The output layer is the prediction of output, whereas the hidden layer represents all layers of the neurons between the input layer and output layer.

The activities that take place at the hidden layer is of particular importance to the neural network. This is because here, we find the activation function at work. The activation function is used to obtain an output from given input or a set of input. Depending on the function used, it maps the resulting values from 0 to 1 inclusively or -1 to 1.

The neuralnet package in R, with also a neuralnet functioncan be used to implement the regression Neural Network. The syntax is neuralnet(formula,data, hidden, linear.output, threshold), where formula is symbolic description of the model to be fitted and data is a dataframe to be supplied. Hidden is used to specify the number of hidden layers as well as the number of neurons each layer consists of. Linear.outputis logical whereas threshold is used to specify the minimum threshold for iterations to stop. Also contained in the Appendices is an R program on how to implement the neural network using neuralnet package.

II. Aim and Objectives of the Study

The SVR and NN appear to share a number of similarities, but what remains to be known is if the performances of both regression procedures on data are the same statistically or not. As a result, this study aims to make a statement on the performances of SVR and NN on data. We are further interested in what happens (the objectives of the study) when:

a. p (the number of explanatory variables) is equal to n (the number of observations).
b. p > n
c. n > p

III. Research Methodology

We shall use both simulated and some real-world datasets in this study. Our focus is set on addressing all the issuesraised in the objectives of the study. Twomodels shall be constructed, namely the Support Vector regression model and the Neural Network regressionmodel. We shall examine the magnitude of the Root Mean Square Error (RMSE)output by both models given different datasets used in the study. Here, a better performing model will be adjudged by the overall magnitude of the Root Mean Square Error (RMSE) obtained. A statistical test of significance must be carried to ascertain if any of the models has the smallest RMSE comparatively.

IV. Data Analysis/Result

A total of eleven datasets consisting of eight real-world and three simulated datasets will be used for this study. The real-world datasets are largely sourced from the internet and a description of them is contained the section that follows:

4.1 Dataset Descriptions

Abalon Dataset:The dataset contains 4177 observationsand 8 variables from zoology field. The variables consistof physical measurement used to determine age of abalone shell by cutting their cone and counting the rings, this is achieved using a microscope. The original data can be sourced from https://archive.ics.uci.edu/ml/datasets/abalone.

Bodyfat Dataset: The dataset is about estimates of the percentage of body fat determined by underwaterweighing and various body circumference measurements for 252 men. The source of the dataset is https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset.

Dataset_2196_cloud: These are data collected in a cloud-seeding experiment in Tasmania between mid-1964 and January 1971. Their analysis, using regression techniques and permutation tests, is discussed in(Miller et al., 1979).The dataset is contained on https://www.kaggle.com/code/milachadayammuri/clouddataviz/data.

Diamond Dataset: This dataset contains 53940 observation of 7 variables. It is about the analysis of diamonds by their cut, color, clarity, price, and other attributes. The source is https://www.kaggle.com/datasets/shivam2503/diamonds.

Fish Dataset: This dataset is a record of 7 common different fish species in fish market sales. It is used to perform a predictive model using machine friendly data and estimate of the weight of fish can be predicted. The source is https://www.kaggle.com/datasets/aungpyaeap/fish-market.

Garment worker productivity: The dataset consists of important attributes of the garment manufacturing process and the productivity of the employees which had been collected manually. It has 1197 observations and 10 attributes. The source is
https://www.kaggle.com/datasets/ishadss/productivity-prediction-of-garment-employees.

WineQuality Dataset:The wine quality dataset concerns red and white variants of the Portuguese "Vinho Verde" wine(Cortez et al., 2009).
As a result of privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available.The data can be downloaded from
https://www.kaggle.com/datasets/rajyellow46/wine-quality?select=winequalityN.csv.

Salary Dataset: The dataset consists of one input variable and an outcome variable; hence it is suitable for carrying out simple regression modelling. There is no other documentation about the data and it can be downloaded from
https://www.kaggle.com/datasets/karthickveerakumar/salary-data-simple-linear-regression.

Hungary Chickenpox: The dataset consists of a weekly chickenpox (childhood disease) cases from Hungary. It is made of county-level adjacency matrix and time series of the county-level reported cases between 2005 and 2015. There are 2 specific related tasks: County level case count prediction and nation level case count prediction. It can be downloaded from https://archive.ics.uci.edu/ml/datasets/Hungarian+Chickenpox+Cases#.

Simulated Dataset 1, 2 & 3: The datasets were simulated from a multivariate distribution but their variables were varied in all the cases. The R codes used to generate the datasets are contained in the Appendix.

4.2 Dataset Analysis/Result

The result of dataset analysis using the R statistical software(R Core Team, 2018)is presented in Table 4.1.

Table 4.1: The outcome of data analysis showing the RMSE for SVR and RMSE for Neural Network.

S/No.	Dataset	Dimensions	RMSEsvr	RMSEnn
1	Abalone	4177×8	0.0753	0.0627
2	Bodyfat	252×15	0.0388	0.0161
3	Dataset_2196_cloud	108×5	0.2995	0.3915
4	Diamond	53940×7	0.0738	Nil

5	Fish Dataset	159×7	0.0304	0.0217
6	Garment worker productivity	1197×10	0.1475	0.1452
7	Wine quality	1599×12	0.1287	8.4815
8	Simulated Dataset1	2000×7	0.0209	0.0083
9	Simulated Dataset2	2500×31	0.0350	0.1774
10	Salary Dataset	159×6	0.042	0.0371
11	Simulated Dataset3	3000×101	0.0359	0.2245
12	Hungary_chickenpox	522×20	0.0832	0.1059

Based on Table 4.1, it seems that the RMSEs of both repression models are neck and neck in majority of the datasets. With some datasets, particularly wine quality and diamond, the SVR performed relatively better. Further, it seems that where the data is relatively high dimensional, the neural network appear to perform poorly. Such cases include diamond, wine quality, simulated datasets 2 & 3.

In a bid to compare the performances of both models, the following hypotheses will be tested:

$$H_0: \mu_{RMSEsvr} = \mu_{RMSEnn}$$
$$H_1: \mu_{RMSEsvr} \neq \mu_{RMSEnn}$$

The datasets clearly failed the normality test because in both cases, $n < 30$. For this reason, a parametric student's t-test will be substituted for a non-parametric Mann-Whitney U Test(MacFarland & Yates, 2016). The result of the test shows that at a p-value of 0.5619, the null hypothesis cannot be rejected. It conclusively shows that both regression procedures (the support vector regression and neural network) are not statistically different from each other in terms of their performances.

V. Summary/Conclusions

It is true that the null hypothesis was not rejected, meaning that the performances of both datasets are essentially the same, but based on the contents of Table 4.1, it is my view that for extremely larger or high dimensional datasets, the support vector regression should be preferred to the neural network. Conversely, if a small dataset is involved, the neural network should be preferred. In all, I am of the view that both regression procedures are good non-parametric techniques for solving regression problems.

References

[1]. Awad, M., & Khanna, R. (2015). Support Vector Regression. In Efficient learning machines (pp. 67–80). Springer.
[2]. Cortes, C., &Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273–297.
[3]. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling Wine Preferences by Data Mining from Physicochemical Properties. Decision Support Systems, 47(4), 547–553.
[4]. MacFarland, T. W., & Yates, J. M. (2016). Mann–Whitney U Test. In Introduction to nonparametric statistics for the biological sciences using R (pp. 103–132). Springer.
[5]. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., Lin, C.-C., & Meyer, M. D. (2019). Package 'e1071.' The R Journal.
[6]. Miller, A. J., Shaw, D. E., Veitch, L. G., & Smith, E. J. (1979). Analyzing the Results of a Cloud-Seeding Experiment in Tasmania. Communications in Statistics-Theory and Methods, 8(10), 1017–1047.
[7]. Obi, J. C. (2020). A Foundation Course in Statistics with Applications in R. Favour Fountain Concepts.
[8]. R Core Team. (2018). R: A Language and Environment for Statistical Computing. https://www.R-project.org/

Appendix

Data Generation From a Multivariate Normal Distribution

```
rm(list = ls()) ## Data dimension 3000X500
library(MASS)
Mean.vector = sample(1:1000, 100, replace = F)
library(clusterGeneration)
Sigma = genPositiveDefMat(dim = 100, covMethod = "unifcorrmat")[[2]]
n = 3000

# create multivariate normal distribution
Multi.sample = round(mvrnorm(n, mu = Mean.vector, Sigma = Sigma), digits = 2)
dat = data.frame(Multi.sample)
dat$y = round(abs(dat$X1+2-3^3*sqrt(5)), digits = 2); dat[1:4, ]
```

Support Vector Regression

```
rm(list = ls())
library(hydroGOF)  ## Calculate RMSE = rmse(predY,data$Y)
```

```
library(e1071)  ## SVR
dat = read.table(file = "clipboard", header = T); names(dat)
colnames(dat)[which(names(dat) == 'Weight')] = 'y'; names(dat)

# MAX-MIN NORMALIZATION
normalize = function(x) {
return((x - min(x))/(max(x) - min(x)))
}
maxmindf = as.data.frame(lapply(dat, normalize))
print(maxmindf[1:3, ], row.names=F)

#Create training and test sets
library(dplyr)  ## sample_frac()
#If you do not have an ID per row, use the following code to create an ID

maxmindf$id = 1:nrow(maxmindf)
train = maxmindf %>% sample_frac(0.70)
test  =anti_join(maxmindf, train, by = 'id')
train = subset(train, select = -id)
test = subset(test, select = -id)

modelsvm = svm(y ~., train)
#Predict using SVM regression
predYsvm = predict(modelsvm, test)

#Calculate RMSE
RMSEsvr = rmse(predYsvm,test$y); RMSEsvr
```

Neural Network

```
library(neuralnet)
n = names(train)
f = as.formula(paste("y ~", paste(n[!n %in% "y"], collapse = " + ")))
##  paste("y ~", paste(n[!n %in% "y"], collapse = " + "))

nn = neuralnet(f, data = train, hidden = c(5,3), linear.output = T)
#plot(nn)

## Prediction
pr.nn = compute(nn, test[, -1])    ## check pr.nn$net.result

## MSE Computation
pr.nn_ = pr.nn$net.result * (max(test$y) - min(test$y)) + min(test$y)
test.r = (test$y) * (max(test$y) - min(test$y)) + min(test$y)

RMSEnn = sqrt(sum((test.r - pr.nn_)^2)/nrow(test)); RMSEnn
print(paste("RMSEsvr = ", round(RMSEsvr, digits = 4),"; ", "RMSEnn = ", round(RMSEnn, digits = 4)))
dim(dat)
```