Academic Journal of Nawroz University (AJNU), Vol.10, No.2, 2021

https://doi.org/10.25007/ajnu.v10n2a505

A Comparative Study of Nearest Neighbor Regression and Nadaraya Watson Regression

Sarwar A. Hamad [1], Kawa S. Mohamed Ali [2]
[1,2] Department of Mathematics, University of Zakho, Kurdistan Region, IRAQ

ABSTRACT

Two non-parametric statistical methods are studied in this work. These are the nearest neighbor regression and the Nadaraya Watson kernel smoothing technique. We have proven that under a precise circumstance, the nearest neighborhood estimator and the Nadaraya Watson smoothing produce a smoothed data with a same error level, which means they have the same performance. Another result of the paper is that nearest neighborhood estimator performs better locally, but it graphically shows a weakness point when a large data set is considered on a global scale.

KEYWORDS: Nadaraya Watson regression, nearest neighbor regression, Monte Carlo simulation.

1. Introduction

In general, a data set collected from a continuous and stable process has a trend. Such data set is referred to as parametric data set. When clearly identified, trend in data set is often used for forecasting. Beside there exist process that produce trendless data set, which is also call non-parametric data set. A special branch of statistics called non-parametric statistics is dedicated to study of non-parametric data set. In general, data are analyzed with the aims to extract knowledge and information. Automatic system, recommended system and machine learning algorithm [6] are built upon existing collected data set, that are used to predict or forecast a system behavior. In this regard, scientists are often interested in finding trend in the data. Such trend would definitely help in predicting next outcome depending on the feature of the data set. In statistical data analysis, there exists a classification of a data set depending on whether or not it has a known trend. In this regard, a parametric data analysis approach is used when a data set has a known trend; these methods are concerned but not limited with simple linear and multi-linear regression [7-9], logistic model [10-13], quadratic regression [14-16]. It is also common in practice to have a data set that doesn't show any of the above mentioned trends. In such case, a non-parametric method is used to fit the data set. Existing non-parametric include but not limited, nearest neighbor nonparametric regression [17-19], Nadaraya Watson regression [20-21] , Kernel smoothing [23-24]. In this work we focused on two nonparametric regression methods namely the Nadaraya Watson regression and the nearest neighbor nonparametric regression. We studied their specificities and their performance in fitting nonparametric data set.

2. Preliminaries

In this section a review of Nearest Neighbor regression and the Nadaraya Watson regression for the smoothing of nonparametric data set is discussed. As a reminder, these are both nonparametric regression methods that are used to fit data set that doesn't show any known parametric trend.

2.1 Nearest Neighbor estimator

The K nearest neighbor regression also known as K-NN is one of the commonly used nonparametric regression algorithms. It helps to build a smooth trend of a data set by averaging the most K nearest neighbors of a selected point. Considering a set of bijective

correspondence between an input variable and its response denote $\{(X_i, Y_i)\}_{i=1}^{n}$, where Y_i form a vector and each X_i can be a single element in the univariate or a vector in the multivariate case [19-25].

The K-NN aims to fit the experimental data set using a model which analytically written as follow

$$\{Y_i = g(X_i) + \varepsilon_i\}_{1 \le i \le n} \quad (1)$$

Where *g* is an unknown function which is assumed to be is continuous and it is at least twice differentiable, $g \in C_{\square}^{2}$. The variable ε_i represent the errors terms, and it is randomly distributed with null expectation, $E(\varepsilon_i | X_i) = 0$. This setting motivates the estimation on the response of a point *x* by simply averaging Y_i's values that are such that X_i 's are close or surround the point *x*. The number K, of nearest neighbors to be sum up is an important parameter in the whole process. That is why the method often referred to ask K-nearest neighbor.

Theorem 1: [19] Given a data set $\{(X_i, Y_i)\}_{i=1}^{n}$, obtained through a stochastic process, where Y_i is considered as the predicted value obtained from a random variable X_i, through an unknown function *f*. An estimation of the unknown function is defined using K Nearest Neighbor estimator as follow

$$\hat{f}_{NN}(x) = \frac{1}{k}\sum_{i=1}^{n} \delta_{ik}(x) Y_i \quad (2)$$

where $\delta_{ik}(x) = I(X_i \in \gamma_k(x))$ is the Kronecker symbol returning 1 if $X_i \in \gamma_k(x)$ and 0 otherwise. Moreover, $\gamma_k(x)$ represent the set of the *k* nearest neighbor data points to *x*.

One usually refers to *Eq(2)* as a smoothing function and its main parameter *k* is called the smoothing parameter. The consistency of this parameter is given by $k = k_n \to +\infty$ and $\frac{k}{n} \to 0$ for $n \to +\infty$. It is important to select suitable value of *k* in the process. In fact, depending on the data size, a too large value of *k* may lead to an over smoothed process whereas the otherwise case will lead to an under smoothed process. In application and during implementation of the K nearest neighbor algorithm, different metrics are used to enable the identification of the k nearest neighbor points, such as Euclidian distance, maximum linkage distance, minimum linkage distance, Dirac distance, among others. It is worthy noted that the Euclidean distance is by far preferred by scientists to implement K nearest neighbor algorithm.

Definition 1 [1-2]: Euclidean distance, given two elements $x(a_1, \ldots, a_p)'$ and $y(b_1, \ldots, b_p)'$ of the normed space $\square^{p}$, the Euclidean distance between *x* and *y* is given by

$$dist_{Eucl}(x, y) = \sqrt{\sum_{i=1}^{p} (a_i - b_i)^2} \quad (3)$$

Below is the K nearest neighbor algorithm. Commonly used machine learning nomenclature is used in this algorithm. In fact the initial data set $\{(X_i, Y_i)\}_{i=1}^{n}$ is considered as the train data set, subset of the train data set is then used to test the algorithm. This is useful to select the test set as a subset of the train set because knowing the real outcome value, that will help to measure the error level and thus to evaluate the performance of the algorithm. Later on it will be possible to predict the outcome of any other data point in the domain. The following is a K nearest neighbor algorithm that evaluates the outcome *y* of a given point *x* based on its K nearest neighbor.

Algorithm 1: K nearest neighbor

Input: *x* point to estimate outcome,

$\{(X_i, Y_i)\}_{i=1}^{n}$ the train data set,

k number of neighbor to consider.

Output: y estimated value

BEGIN

S← 0, mt←0;

While (mt < = k)

Return $\left(x_j, y_j\right)$ such that

$$dist_{Eucl}\left(x_j, x\right) = Arg_{\min}\left\{dist_{Eucl}\left(x_j, \{x\}_{i=1}^{n-mt}\right)\right\};$$

S←S+y_j;

delete $\left(x_j, y_j\right)$;

mt←mt+1;

End While

y←Average(S);

return y;

END

Algorithm 1 is used in sequel for the evaluation of predictor values using a train data set.

2.2 Kernel Estimator and Nadaraya Watson Estimator

This method is used to fit pairs of data set $\left\{\left(X_i, Y_i\right)\right\}_{i=1}^{n}$ obtained through stochastic process. The unknown function is fitted based on two main parameters. These are the smoothing parameter h, which is also called the bandwidth and the kernel function K, which is a function with some specifics characteristics. There exist various approaches and formulae through which a kernel estimator function is built. In this work we will focus on two of them. The first formula [22] is given by

$$\hat{f}_{KE}\left(x\right) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K\left(\frac{x - X_i}{h}\right) \qquad (4)$$

where $K\left(.\right)$ is called the kernel function or smoother function and h is the bandwidth.

The second approach is called the Nadaraya-Watson smoothing [21] function and is given by the formula

$$\hat{f}_{NW}\left(x\right) = \frac{\sum_{i=1}^{n}K\left(\frac{x - X_i}{h}\right)Y_i}{\sum_{i=1}^{n}K\left(\frac{x - X_i}{h}\right)} \qquad (5)$$

Picking a suitable bandwidth value h for *Eq.(5)* is crucial for the model. Hence, in this work we also focused on selection of suitable bandwidth. The kernel function $K\left(.\right)$ has some properties that are found in detail in literature (see[22]). So far seven functions which fulfill the bandwidth requirements have been exhibited by researchers. These are Uniform, Triangle, Epanechnikov, Quadratic, Triweight, Gaussian, Cosinus. The Gaussian Kernel is the most used kernel. Further studies on kernel are given in the experimental part below.

2.3 Relationship between Nearest Neighbor estimator and Kernel Estimators

There exists a large literature on how to select the bandwidth h in *Eq.(4)* and *Eq.(5)*. However, there isn't a steady formula that computes the number of nearest neighborhood points to be used in the process of smoothing using *Eq.(2)*. In this regard, the said selection is either done by a simple rule of thumb or even a try and error method. Fortunately, there is a relationship between the number of nearest neighborhood points k, to be used (see *Eq.(2)*) and the bandwidth (see *Eq.(4)* and *Eq.(5)*). The said relation is given by

$$\frac{k}{n} \propto h \qquad (6)$$

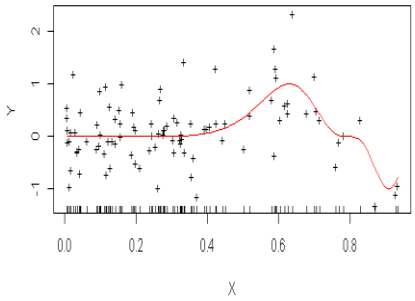
where k, n, and h are the parameters as referred to in *Eq.(2)* , *Eq.(4)* and *Eq.(5)*.

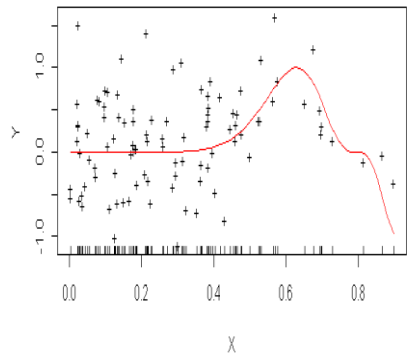
3. Experimental Studies

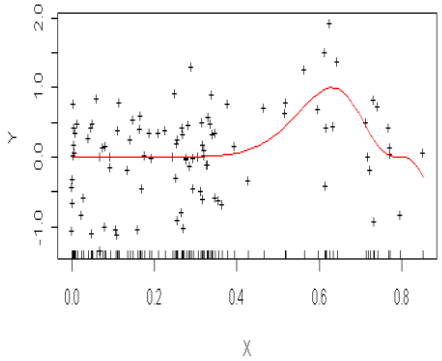
In this section an experimental study is carried out to investigate the performance of Nearest Neighbor estimator and the Nadaraya Watson estimator. In order to ease computation and without loss of generality, the Gaussian Kernel function will be used in sequel.

The experimental data set is obtained through a stochastic process simulated as $\{(X_i, Y_i)\}_{i=1}^{n}$, $n = 100$ from the regression model defined by $Y_i = m(X_i) + 0.5\varepsilon_i$, $i = 1, \ldots, n$. Where $X_i \overset{iid}{\sim} \beta(a = 0.8, b = 2)$ and $\{\varepsilon_i\} \overset{iid}{\sim} N(0,1)$, and $m(x) = \left(\sin\left(2\pi x^3\right)\right)^3$.

Before investigating the performance of the nearest neighbor estimator and the kernel estimator, let us first of all show the random effect of the stochastic data generation.







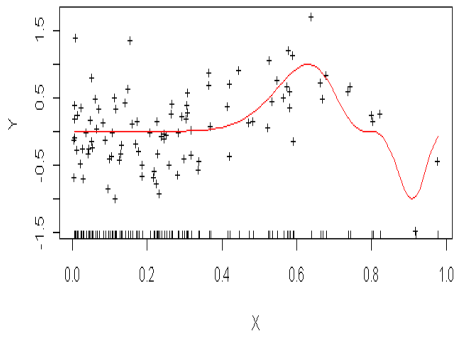
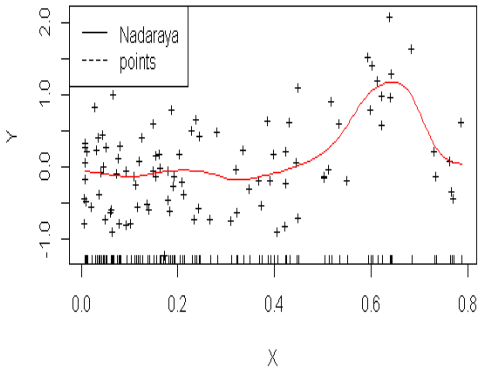


Figure 1: 4 samples of stochastic data set.

Figure 1 shows 4 samples of stochastically generated data sets. Indeed the aim of this figure is sure evidence of the random behavior of the process. It clearly appeared that the randomness in the generated data sets is obvious. At this point, a single sample data will be generated and will be used in what will follow for experiment on the K nearest neighbor and Nadaraya Watson methods.

3.1 Illustration of Kernel estimation and Nearest Neighbor Estimation

In this section illustration of kernel estimation and Nearest Neighbor estimator are given. One round simulation is performed to generate the stochastic data set and then both the Nearest Neighbor and the Nadaraya estimators are used to fit the data set



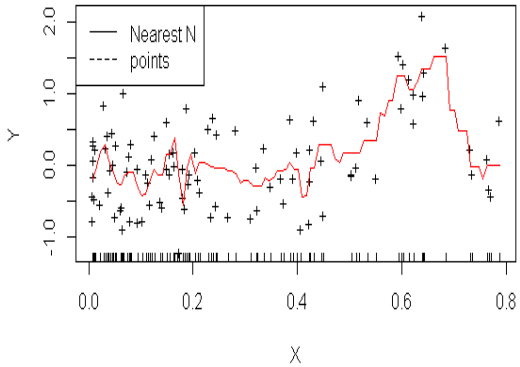


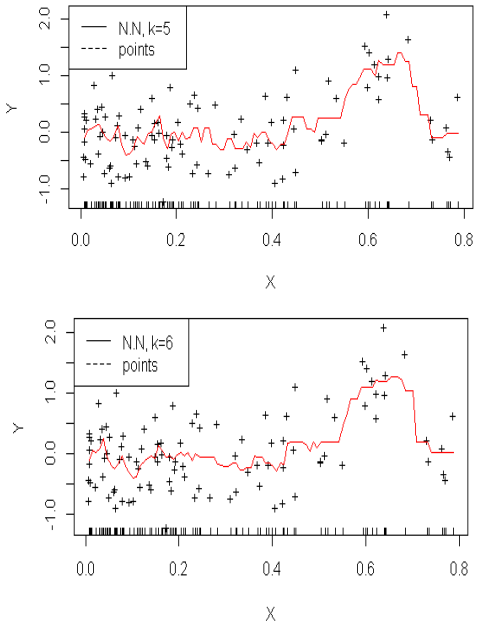
Figure 2: (a) Nadaraya estimator , (b) Nearest Neighbor Estimator

Figure 2 is an illustration of Nadaraya estimator and Nearest Neighbor Estimator, in which their smoothing parameters are respectively $h=0.0448$ and $k=4$. More over these smoothing parameters are chosen in a way to be related by *Eq(6)*. Note that the optimal value of $h=0.0448$, was first of all obtained using the function *dpill()* of R programming language . After using the smoothing parameters to fit the data sets, further studies are done to evaluate the performance of both estimators. For this purpose, the Bias (see [5]), MSE (see [3]), Variance and MISE (see [4]) of each method are computed for performance evaluation purposes.

Table 1: Comparative study

	Bias	*MSE*	*Variance*	*MISE*
Nadaraya	0.1239	0.4514	0.1883	2.0330
Nearest N	0.1361	0.5172	0.2661	2.3452

Given that the Nearest Neighbor estimator shown on Figure 2(b) looks under-smoothed, different larger bandwidth values were used to analysis the process, in order to observe the behavior of the graph as a function of k. R-programming language was used to generate all plots.



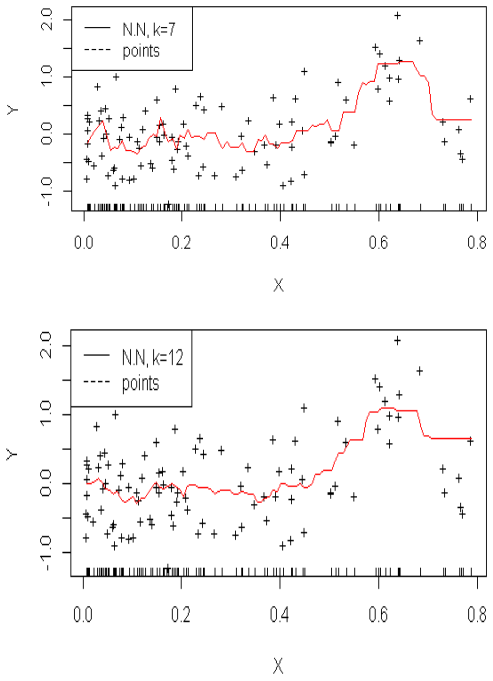


Figure 3: Sample of Nearest Neighbor estimators for different values of k.

From Figure 3, it is clear that despite the variation of the k values, the obtained curves don't have global smooth shapes. Rather they are globally stretched. As a matter of fact, the Nearest Neighbor is suitable for local estimation but not for global estimation.

3.2 Monte Carlo Simulation

In this section the experiment carried in section 3-1, is repeated many times in order to appraise the behavior of the two estimators. The assumptions are the following, the optimal bandwidth for kernel smoothing is computed using the R function *dpill()*, and the optimal k value for Nearest Neighbor is obtained from *Eq(6)*.

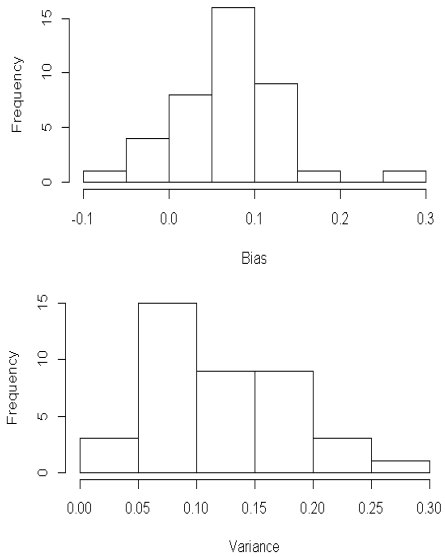
Simulation 1: In this simulation the random process used to generate the stochastic data is repeated a number of $N=40$ times. Each time, the Nadaraya Watson regression is used to fit the random data set and some metrics of the fitted data set are recorded. These metrics are the Bias, MSE, Variance and MISE. The *40* records of each of those metrics are considered as observations of random variables. Moreover, their mean and standard deviation are computed.

Table 2: Monte Carlo of size $N=40$ with Nadaraya Watson Estimator

	Bias	*MSE*	*Variance*	*MISE*
Mean	0.0683	0.4252	0.1206	2.6527
Standard Deviation	0.0626	0.1185	0.0616	0.3028

Figure 4 is the histograms of the Bias, MSE, Variance and MISE obtained from the 40 rounds Monte Carlo simulation, followed by Nadaraya Watson data fitting. These histograms aim is to graphically determine the distribution of each of those parameters. Although it is not clear enough, one can see that the bias tends to be normally distributed, whereas the MSE and Variance follow a chi-square distribution. A higher frequency simulation would probably illustrate with higher precision.

Simulation 2: In this simulation the random process used to generated the stochastic data is repeated a number of *N=1000* times. Each time, the Nadaraya Watson regression is used to fit the random data set and some metrics of the fitted data set are recorded. These metrics are the Bias, MSE, Variance and MISE. The 1000 records of each of those metric are considered as observations of random variables. Moreover, their means and standards deviations are computed.



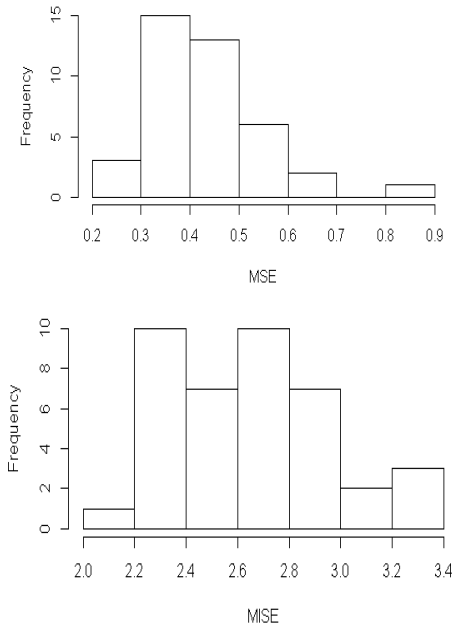
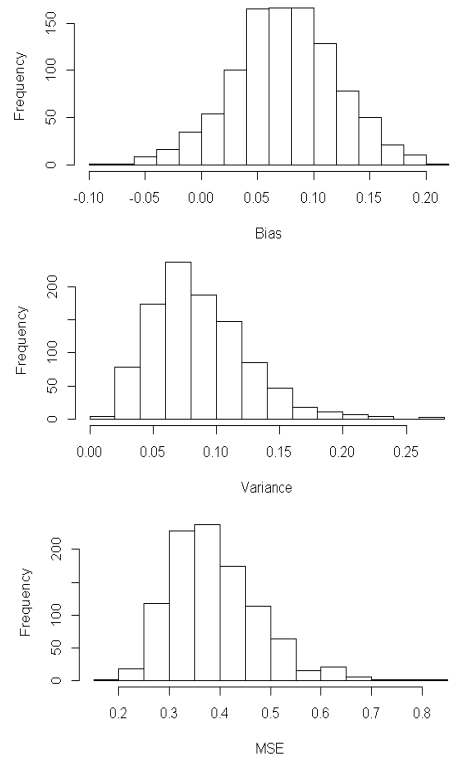


Figure 4: Bias, Variance, MSE, MISE for N=40, with Nadaraya Watson

Table 3: Monte Carlo of size *N=1000* with Nadaraya Watson Estimator

	Bias	*MSE*	*Variance*	*MISE*
Mean	0.0746	0.3897	0.0865	2.7114
Standard Deviation	0.0467	0.0868	0.0381	0.3066



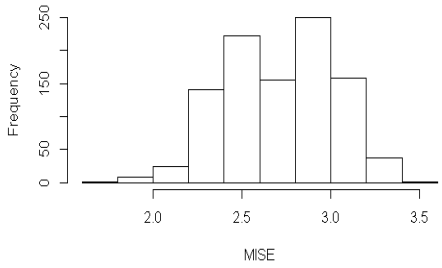


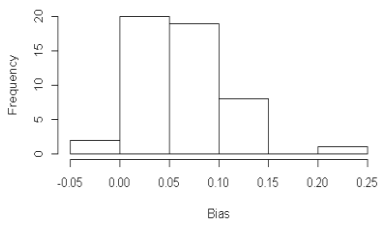
Figure 5: Bias, Variance, MSE, MISE for N=1000, with Nadaraya Watson

Figure 5 is the histograms of the Bias, MSE, Variance and MISE obtained from the 1000 rounds Monte Carlo simulation followed by Nadaraya Watson data fitting. These histograms aim is to graphically determine the distribution of each of those parameters. With 1000 rounds of simulations, it is clear enough to see which distributions theses parameters follow; one can see that the bias is normally distributed, whereas the MSE and Variance follow a chi-square distribution. The MISE doesn't clearly show the shape of a known distribution; however it can be fitted with a normal curve.

Simulation 3: In this simulation the random process used to generate the stochastic data is repeated a number of N=50 times. Each time, the Nearest Neighbor regression is used to fit the random data set and some metrics of the fitted data set are recorded. These metrics are the Bias, MSE, Variance and MISE. The 50 records of each of those metric are considered as observations of random variables. Moreover, their means and standards deviations are computed.

Table 4: Monte Carlo of size *N=50* with Nearest Neighbor

	Bias	***MSE***	***Variance***	***MISE***
Mean	0.0675	0.3981	0.0896	2.7539
Standard Deviation	0.0417	0.0845	0.0363	0.3099



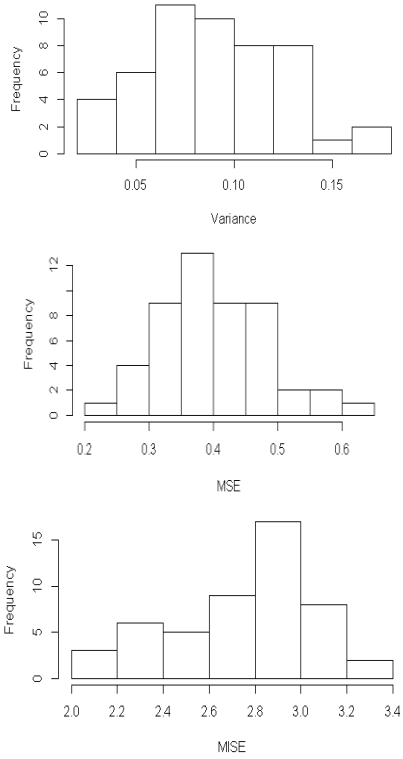


Figure 6: Bias, Variance, MSE, MISE for N=50, with Nearest Neighbor

Figure 6 is the histograms of the Bias, MSE, Variance and MISE obtained from the 50 rounds Monte Carlo simulation, followed by data fitting using the nearest neighbor estimator. These histograms aim is to graphically determine the distribution of each of those parameters. Although it is not clear enough, one can see that the bias tends to be normally distributed, whereas the MSE and Variance follow a chi-square distribution. A higher frequency simulation would probably illustrate with higher precision.

Simulation 4: In this simulation the random process used to generated the stochastic data is repeated a number of *N=1000* times. Each time, the Nearest Neighbor regression is used to fit the random data set and some metrics of the fitted data set are recorded. These metrics are the Bias, MSE, Variance and MISE. The 1000 records of each of those metric are considered as observations of random variables. Moreover, their means and standards deviations are computed.

Table 5: Monte Carlo of size N=1000 with Nearest Neighbor

	Bias	***MSE***	***Variance***	***MISE***
Mean	0.0749	0.3993	0.0896	2.7238

Standard Deviation	0.0479	0.0924	0.0416	0.3027

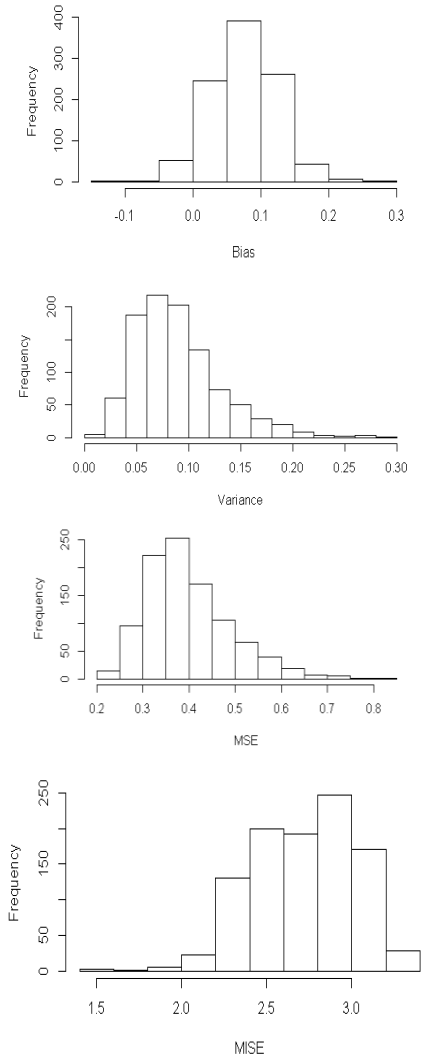


Figure 7: Bias, Variance, MSE, MISE for N=1000, with Nearest Neighbor

Figure 5 is the histograms of the Bias, MSE, Variance and MISE obtained from the 1000 rounds Monte Carlo simulation followed by Nearest Neighbor data fitting. These histograms aim is to graphically determine the distribution of each of those parameters. With 1000 rounds of simulations, it is clear enough to see which distributions theses parameters follow; one can see that the bias is normally distributed, whereas the MSE and Variance follow a chi-square distribution. The MISE doesn't clearly show the shape of a known distribution; however it can be fitted with a normal curve.

3.3 Hypothesis Tests

From the simulations done in the previous section, hypothesis tests are performed to determine which method is the best between Nadaraya Watson estimator and Nearest Neighbor estimator. For both methods and in both cases, we performed the test using data obtained from the 1000 rounds Monte Carlo simulation to be more reliable. On one hand test of difference between two population means is used to check whether or not there is a significant difference between the BIAS obtained from both methods. On the other hand, a test of difference between two population variances is applied to check if there is a significant difference between the variance of the MSE of both methods.

Table 6: Test of difference between Nearest N. and Nadaraya W. *BIAS* population means

	Nearest N.	***Nadaraya W.***
Mean	0.0749	0.0746
Std Dev	0.0479	0.0467
Sample size	1000	1000

A two tails normal test revealed at level 0.05 that the difference between the two population means $-0.004 < \mu_1 - \mu_2 < 0.004$ might be equal to 0, hence $\mu_1 = \mu_2$.

Table 7: Test of difference between Nearest N. and Nadaraya W. *MSE* population variance

	Nearest N.	***Nadaraya W.***
Mean	0.3393	0.3897
Std Dev	0.0924	0.0868
Sample size	1000	1000

A two tails test of ratio between the two population variance revealed that they are equal $\sigma_1 = \sigma_2$.

Tables 6 and 7 display the results of the hypothesis tests. Form these results; one can observe that, there is no significant difference between the means of BIAS obtained from both methods. On the other hand, it is also clear that there is no significant difference between the variances of MSE obtained from both methods.

4. Conclusion

Based on the Monte Carlo simulation, it appears from the histograms that for both methods (Nearest Neighbor & Nadaraya Watson), the bias are *normally distributed*, Variance and Mean Squared Error (MSE) follow a *Chi-square distribution*, but Mean Integrated

Squared Error (MISE) doesn't present a shape of any of the commonly known distribution. However, it is more likely to be normally distributed. More important is that under the condition of *Eq(6)*, that establishes the relationship between the Nadaraya Watson optimal bandwidth *h* and the optimal number of *K*-Nearest Neighbor to be used, both methods perform equally.

5. REFERENCE

1. Liberti, L., Lavor, C., Maculan, N., & Mucherino, A. (2014). Euclidean distance geometry and applications. *SIAM review*, *56*(1), 3-69..
2. Fabbri, R., Costa, L. D. F., Torelli, J. C., & Bruno, O. M. (2008). 2D Euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, *40*(1), 1-44.
3. Theobald, C. M. (1974). Generalizations of mean square error applied to ridge regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, *36*(1), 103-106.
4. Davis, K. B. (1977). Mean integrated square error properties of density estimates. *The Annals of Statistics*, 530-535.
5. Wahba, G., Lin, X., Gao, F., Xiang, D., Klein, R., & Klein, B. (1999). The bias-variance tradeoff and the randomized GACV. In *Advances in Neural Information Processing Systems* (pp. 620-626).
6. Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). Machine learning. *Neural and Statistical Classification*, *13*.
7. Seber, G. A., & Lee, A. J. (2012). *Linear regression analysis* (Vol. 329). John Wiley & Sons.
8. Weisberg, S. (2005). *Applied linear regression* (Vol. 528). John Wiley & Sons.
9. Kutner, M. H., Nachtsheim, C. J., Neter, J., & Li, W. (2005). *Applied linear statistical models* (Vol. 5). Boston: McGraw-Hill Irwin.
10. Wright, R. E. (1995). Logistic regression.
11. Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression*. New York: Springer-Verlag.
12. Menard, S. (2002). *Applied logistic regression analysis* (Vol. 106). Sage.
13. Pregibon, D. (1981). Logistic regression diagnostics. *The Annals of Statistics*, *9*(4), 705-724.
14. Höskuldsson, A. (1992). Quadratic PLS regression. *Journal of Chemometrics*, *6*(6), 307-334.
15. Cudeck, R., & Du Toit, S. H. (2002). A version of quadratic regression with interpretable parameters. *Multivariate Behavioral Research*, *37*(4), 501-519.
16. Stinchcombe, J. R., Agrawal, A. F., Hohenlohe, P. A., Arnold, S. J., & Blows, M. W. (2008). Estimating nonlinear selection gradients using quadratic regression coefficients: double or nothing?. *Evolution: International Journal of Organic Evolution*, *62*(9), 2435-2440.
17. Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175-185.
18. Stute, W. (1984). Asymptotic normality of nearest neighbor regression function estimates. *The Annals of Statistics*, *12*(3), 917-926.
19. Maltamo, M., & Kangas, A. (1998). Methods based on k-nearest neighbor regression in the prediction of basal area diameter distribution. *Canadian Journal of Forest Research*, *28*(8), 1107-1115.
20. Cai, Z. (2001). Weighted nadaraya–watson regression estimation. *Statistics & probability letters*, *51*(3), 307-318.
21. Devroye, L. P. (1978). The uniform convergence of the nadaraya-watson regression function estimate. *Canadian Journal of Statistics*, *6*(2), 179-191.
22. Wand, M. P., & Jones, M. C. (1994). *Kernel smoothing*. Chapman and Hall/CRC.
23. Daouia, A., Gardes, L., & Girard, S. (2013). On kernel smoothing for extremal quantile regression. *Bernoulli*, *19*(5B), 2557-2589.
24. Altman, N. S. (1990). Kernel smoothing of data with correlated errors. *Journal of the American Statistical Association*, *85*(411), 749-759.
25. Kramer, O. (2011, December). Dimensionality reduction by unsupervised k-nearest neighbor regression. In *2011 10th International Conference on Machine Learning and Applications and Workshops* (Vol. 1, pp. 275-278). IEEE.