KNN and ARL Based Imputation to Estimate Missing Values

Thirumahal R, Deepali A Patil
Departement of Information Technology, Thadomal Shahani Engineering College (TSEC)
Email: deep.patil1987@gmail.com

Abstract

Missing data are the absence of data items for a subject; they hide some information that may be important. In practice, missing data have been one major factor affecting data quality. Thus, Missing value imputation is needed. Methods such as hierarchical clustering and K-means clustering are not robust to missing data and may lose effectiveness even with a few missing values. Therefore, to improve the quality of data method for missing value imputation is needed. In this paper KNN and ARL based Imputation are introduced to impute missing values and accuracy of both the algorithms are measured by using normalized root mean sqare error. The result shows that ARL is more accurate and robust method for missing value estimation.

***Keywords**: Autoregressive model (AR), K-nearest neighbor, Missing value estimation, Time series analysis.*

1. Introduction

To get the required information from huge, incomplete, noisy and inconsistent set of data it is necessary to use data preprocessing. Data cleaning phase of preprocessing is used because the aim here is to impute (fill) the missing values using estimated ones. During filling of the missing values three things are important, they are as follows: A. Estimated values without bias, B. The relation between attributes should be maintained, and C. Minimize the cost.

In most cases, a data set's attributes are not independent of each other. Thus, through the identification of relationships among attributes, missing values can be determined. K-nearest neighbor imputation [1] can find missing values in a particular column but if many missing values are present then this method will give incorrect result. Also, the choice of distance function must be properly selected to estimate the missing values. ARL [2] will perform when there are many missing values at particular time points, and even when the experiments for time points fail or are missing.

In this paper, Normalized root mean square error is calculated based on the experimental results which are then used to compare both the algorithms. Comparision results will show the accuracy of algorithm.

2. Earliear Work

Many existing, industrial, and research data sets contain missing values (MVs). There are various reasons for their existence, such as manual data entry procedures, equipment errors, and incorrect measurements. The presence of such imperfections usually requires a preprocessing stage in which the data are prepared and cleaned. Missing Value Analysis provides a slightly different set of descriptive tools for analyzing missing and includes a variety of single imputation methods**.**

Simplest technique to impute the missing values that is mean imputation was metioned in paper “dealing with missing software project data” by M.J. Shepperd and M.H. Cartwright [3]. Bu it distorts relationships between variables by “pulling” estimates of the correlation toward zero.

Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, David Botstein described Singular value decomposition method (SVD) in paper “Imputing Missing Data for Gene Expression Arrays” [4] to obtain a set of mutually orthogonal values. But the drawbacks of SVD are that it can only be performed on complete matrices and the SVD-based

Received June 25, 2014; Revised August 6, 2014; Accepted September 1, 2014

method shows sharp deterioration in performance when a non-optimal fraction of missing values is used. Therefore, KNN Impute method is introduced which is less sensitive to the exact parameters used (number of nearest neighbors).

The LLSimpute (Local Least Squares Imputation) algorithm [5] predicts the missing value using the least squares formulation for the neighborhood column and the non-missing entries. It works well but the time complexity is higher. Due to above disadvantages in [6] they discussed an autoregressive-model-based missing value estimation method that takes into account the dynamic property of temporal data and the local similarity structures in the data. This method is especially effective for the situation where a particular time point contains many missing values or where the entire time point is missing.

In this work, we describe and evaluate two methods of estimation for missing values in Synthetic control dataset [2]. We compare our KNN and ARL based methods by calculating error rate using NRMSE to show the accuracy.

3. Problem Statement

Given a record of all the time series data, first find the missing values which are indicated by zeros from each column, estimate that values and finally impute those values which are same as original values. To impute the missing values KNN Impute and ARL Impute algorithms are used.

3.1 KNN Impute Algorithm:

KNNI [7] computes the k nearest neighbors and a value from them is imputed. For nominal values, the most common value among all neighbors is taken, and for numerical values, the average value is used. Therefore, a proximity measure between instances is needed for it to be defined. The Euclidean distance is most commonly used in the literature.

The most frequent value among k-nearest neighbor and the mean among the k-nearest neighbor can be predicted by k-nearest neighbor imputation. In this method the main factor is Distance metrics. In 1NN imputation method we can replace the missing values with the nearest neighbor. But if the value of K is greater than one then replace the missing values with the mean or weighted average of K-nearest neighbors. By setting k-value between 10 and 20 brings the best results for KNN imputation.

Euclidean distance is calculated by following equation:

$$dist\ (V_x, V_y) = \sqrt{\sum_{t=1}^{n} \left(e_{x,t} - e_{y,t}\right)^2} \tag{1}$$

Weighted average of corresponding entries is calculated by following equation:

$$G_{IJ} = \sum_{i=1}^{k} W_i \times e_{ij} \tag{2}$$

Weighted value is given in following equation:

$$W_i = \frac{1}{dis\ (g^*, g_i) \times \Delta} \tag{3}$$

Where,

$$\Delta = \sum_{i=1}^{k} \left[1 / dist\ \left(g^*, g_i\right)\right] \tag{4}$$

Example 2 (KNNI)

Nearest neighbor: 2

Inserted Array:

Gx	Gy
29.2171	32.8717
32.337	36.0253
32.8717	34.5249
0.0	34.1173
26.3693	27.6623
24.8923	25.7744
29.9423	30.9493
35.6805	35.2623

Gy: reference column

Predicted Missing gx: 27.8725

3.2 ARL Impute Algorithm

KNN impute, is not able to deal with the situation where a particular time point (column) of the data is missing entirely. ARL Impute [1] and [6] is an effective method for the situation where a particular time point contains many missing values or where the entire time point is missing. ARL impute, consists of two major processes: in the first one, we assume no missing data and estimate the AR coefficients, and in the second process, we assume that the AR coefficients are known and we estimate the missing data.

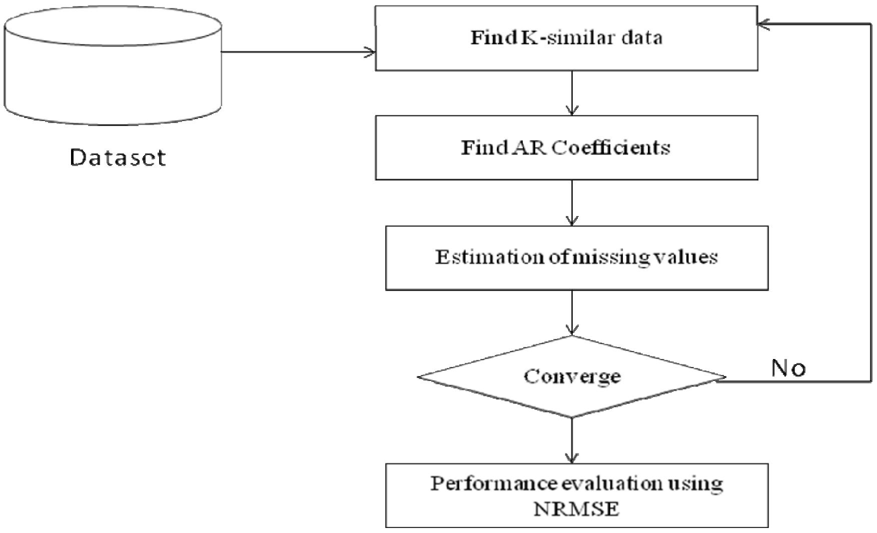

Dataset
Find K-similar data
Find AR Coefficients
Estimation of missing values
Converge
No
Performance evaluation using NRMSE

Figure 1. An overview of ARL Impute

AR coefficients [1] can find by using least square approximation [1] or cubic spline interpolation method. AR model can be described in equation as follows:

$$y_j = Y_j a_j + \in_j \tag{5}$$

where, aj is the coefficient matrix., $\in$j is a sequence of independent identically distributed normal random variable with mean zero. It means that any given value y_j in the time series is directly proportional to the pervious value Y_j plus some random error $\in$j. As the number of AR parameters increase, y_j becomes directly related to additional past values.

$$\begin{bmatrix} y[p+1] \\ y[p+2] \\ \vdots \\ y[n] \\ y[1] \\ y[2] \\ \vdots \\ y[n-p] \end{bmatrix} = \begin{bmatrix} y[p] & y[p-1] & \cdots & y[1] \\ y[p+1] & y[p] & \cdots & y[2] \\ \vdots & \vdots & & \vdots \\ y[n-1] & y[n-2] & \cdots & y[n-p] \\ y[2] & y[3] & \cdots & y[p+1] \\ y[3] & y[4] & \cdots & y[p+2] \\ \vdots & \vdots & & \vdots \\ y[n-p+1] & y[n-p+2] & \cdots & y[n] \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} + \varepsilon_j \qquad (6)$$

In this paper AR coeffiecients are calcualed by using least sqare approximation principle [1]. Let us assume that (y1,y2,….ys) are the observed data and {x1, . . . , xm} are the missing data. Estimation of missing data in matrix form is given by:

$$e=Az \qquad (7)$$

where z is a column vector that consists of the observed data y and the missing data x, and A is a Toeplitz matrix whose column number is n and row number is n- p , which is written as:

$$A = \begin{bmatrix} -a_p & \cdots & -a_1 & 1 & 0 & \cdots & 0 \\ 0 & -a_p & \cdots & -a_1 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & & -a_p & -a1 & 1 \end{bmatrix} \qquad (8)$$

If we separate the observed data from missing data and split A in the block matrix, the equation can be written as:

$$e=Bx+Cy \qquad (9)$$

Where B = [B1, B2 · · ·] and C = [C1, C2 · · ·] are block sub matrices of A corresponding to the respective locations of observed data y and missing data x. Finally the missing data can be calculated from B# (pseudo inverse of B).The corresponding equation is given by:

$$x= -B^{\#} Cy \qquad (10)$$

Example 2 (ARLI):
t={1,2,**3,4,5**,6,7}
ft={ 34.4632, 31.2834**, 33.7596, 27.7849,7.569,** 29.2171, 32.337};
f={ 34.4632, 31.2834**,0,0,0,** 29.2171, 32.337};

Predicted missing values are:
28.97423999999749 at t=3.0
27.73036999999391 at t=4.0
27.746439999989708 at t=5.0

3.3 Performance measure:

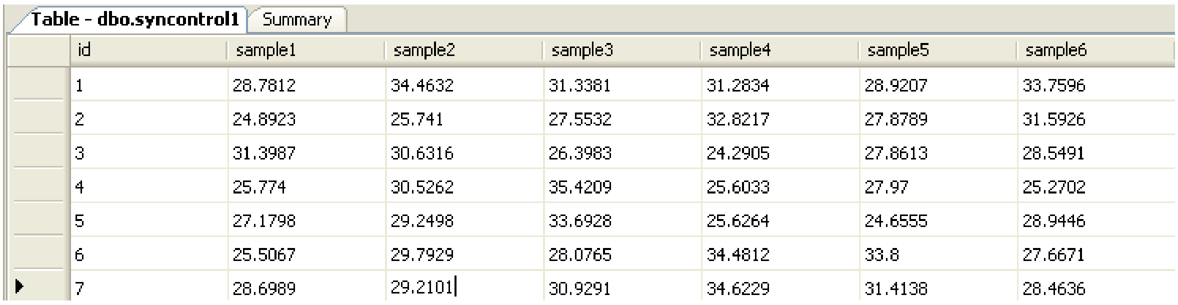
Normalized RMS Error (NRMSE) is used to measure the performance of missing value estimation method, it can be calculated as:

$$NRMSE = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\tilde{Y}(i,j) - Y(i,j)\right]^2}{\sum_{i=1}^{m}\sum_{j=1}^{n}\left[Y(i,j)\right]^2}} \qquad (11)$$

where, Y is the true value, $\tilde{Y}$ is the estimated value, and m and n are the total number of rows and columns respectively.

4. Results and Discussion

Missing values are estimated for Synthetic control chart dataset [2] using K-nearest neighbor and Auto Regressive (AR) Model. Dataset is shown in figure 2.



	id	sample1	sample2	sample3	sample4	sample5	sample6
	1	28.7812	34.4632	31.3381	31.2834	28.9207	33.7596
	2	24.8923	25.741	27.5532	32.8217	27.8789	31.5926
	3	31.3987	30.6316	26.3983	24.2905	27.8613	28.5491
	4	25.774	30.5262	35.4209	25.6033	27.97	25.2702
	5	27.1798	29.2498	33.6928	25.6264	24.6555	28.9446
	6	25.5067	29.7929	28.0765	34.4812	33.8	27.6671
▶	7	28.6989	29.2101	30.9291	34.6229	31.4138	28.4636

Figure 2. Synthetic contol chart Time series dataset

Performance of KNN Impute was assessed over different K values and different percentage of missing values shown in figure 3. Result shows that when the numbers of K values are in the range of 10-20 then, NRMSE error rate is lesser. That is, by setting k-value between 10 and 20 brings the best results for KNN imputation shown in table 1.

Table 1. NRMSE result for KNN Impute over different K values.

K values	KNN Impute
3	0.179
5	0.15876
10	0.1111
15	0.0889
22	0.1245
25	0.1489

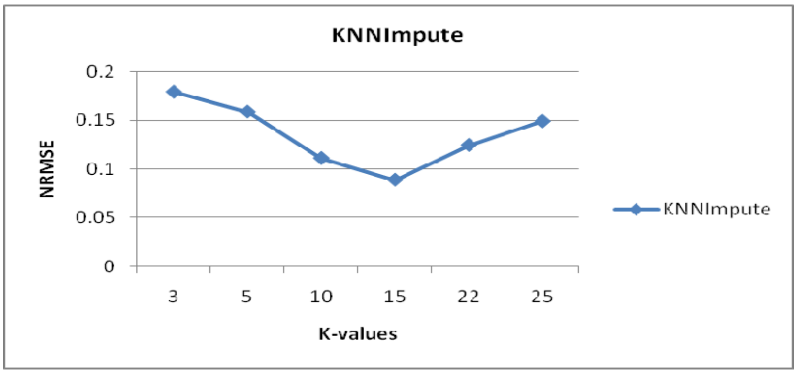


Figure 3. Effect of number of nearest neighbors used for KNN-based estimation

Then Performance of KNN Impute and ARL Impute were measured over NRMSE and % of missing values as shown in figure 4. The graph shows that the error rate of KNNimpute is higher than ARL impute when the percentage of missing values are inceased in a particular column. Comparision of NRMSE result is shown in Table 2.

Table 2. NRMSE result for ARL and KNN Impute over % of missing values

% of missing values	ARL Impute	KNN Impute
10	0.08004	0.12
15	0.0912	0.28
20	0.17	0.33

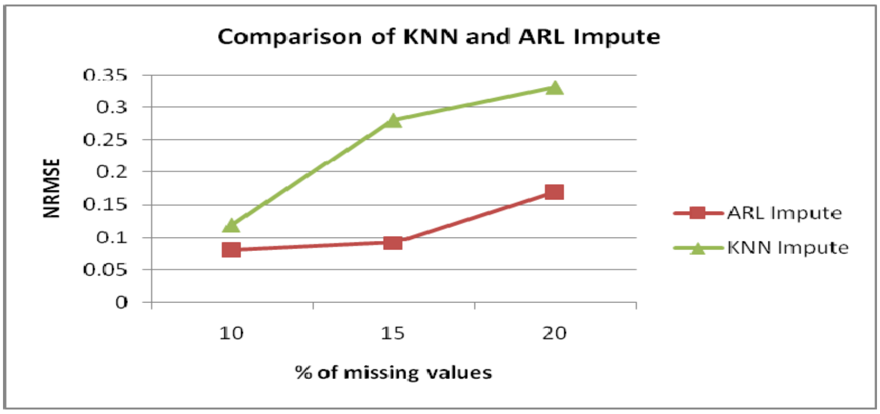


Figure 4. NRMSE comparison pf ARL and KNN Impute algorithm

5. Conclusion And Future Work

We have developed two algorithms KNN impute and ARL impute to estimate and impute the missing values. Both the algorithms give best results to estimate the missing values. KNN Impute gives best results when the K value is in the range of 10-20. KNN Impute estimates missing value from a particular column and ARL Impute estimates many missing values from a particular column. When the performance of both algorithms is measured based on normalized root mean square error (NRMSE) then ARL Impute gives best estimation, as its error rate is less. Our future work aims at finding the missing values for more than two columns of missing data using the probablilistic models like PPCA (Probabilistic principle component analysis).

References

[1] David Sheung Chi Fung. “*Methods for the Estimation of Missing Values in Time Series*”. a thesis Submitted to the Faculty of Communications, Health and Science Edith Cowan University Perth, Western Australia.

[2] UCI machine learning repository, http://archive.ics.uci.edu/ml, 2010.

[3] MJ Shepperd and MH Cartwright. “*Dealing with Missing Software Project Data*”. Empirical Software Engineering Research Group School of Design, Engineering & Computing, Bournemouth University.

[4] Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, David Botstein. “*Imputing Missing Data for Gene Expression Array*s”.

[5] YTao, D Papadias, and X Lian. *Reverse KNN Search in Arbitrary Dimensionality*. Proc. 30th Int’l Conf. Very Large Data Bases (VLDB ’04), 2004.

[6] Miew Keen Choong, Maurice Charbit, and Hong Yan. Autoregressive-Model-Based Missing Value Estimation for DNA Microarray Time Series Data. *IEEE Transactions on Information Technology In Biomedicine*. 2009; 13(1): 131-138.

[7] O Troyanskaya, M Cantor, G Sherlock, et al. "Missing value estimation methods for DNA microarrays". *Bioinformatics*. 2001; 17: 520-525.

[8] PMT Broersen. “*Finite sample criteria for autoregressive order selection*”. *IEEE Trans. Signal Process.* 2000; 48(12): 3550–3558.

[9] MS Sehgal, L Gondal, LS Dooley. “Collateral Missing value imputation: a new robust missing value estimation algorithm for microarray data”. *Bioinformatics*. 2005; 21: 2417-2423.

[10] H Kim, GH Golub, and H Park. “Missing value estimation for DNA microarray gene expression data: local least squares imputation”. *Bioinformatics*. 2005; 21: 187-198.