Missing Value Imputation Method using Ensemble Technique For Microarray Data

Kairung Hengpraprohm* and Suwimol Jungjit**

Abstract

This paper proposes a new missing value imputation method for microarray data using ensemble technique (KNN-Ensemble). We run an experiment on three standard benchmark microarray datasets: Colon, Prostate and Ovarian. Four different distance functions for KNN imputation method were studied. Our experiment can be separated into three steps: (1) selecting two best distance functions for KNN imputation; one distance function for evaluating sample distance and another one is the best distance function used to evaluate distance between features (2) estimating missing values using KNN-Ensemble based on two selected functions from the first step and (3) evaluating the performance of new imputation method for microarray data using ensemble approach with other well-known imputation algorithms: original KNN and Row-Average imputation. The experimental results show that KNN-Ensemble method using Manhattan and Euclidian distance function outperformed other baseline imputation methods on three datasets.

Keywords: Missing Value, Data Imputation, Microarray Data, K-nearest neighbor, Euclidian Distance, Manhattan Distance, Cosine, Pearson, Data Mining.

1. Introduction

DNA microarray is a chip-based technology which developed for measure the gene expression levels of tens of thousands of gene simultaneously. A microarray may contain thousands of spots, which contain the patient's tissue, each spot on the chip represents a different coding sequence from different genes. Gene expression levels can help the physicians to diagnose the patient's condition whether a patient has disease or not [1].

In the context of microarray datasets, the main challenge for data mining is missing values. Missing values always occur in microarray dataset for many reasons such as an insufficient image resolution, image fraud, or the scratches on the microarray chips [2]. Instead of repeating the biological experiment, which too expensive to run and take time, or discarding all observations with missing values (negative impact on microarray analyses), many missing value imputation methods have been proposed in many publish literatures [3], [4] such as [5], [6] and [7].

The simplest way of dealing with missing value is to discard the samples that contain missing value. However, this method has a negative impact on data analysis and it will not have any impact if the data contain a relatively small number of missing value [3]. Zero imputation and Row average imputation method which replacing the missing values by zero and the average value, respectively, can be helpful instead of eliminating the sample which contain missing values [8]. For more sophisticated methods, KNN imputation is the best among those aforementioned methods. However, there are still some points to improve according to increase the performance of estimating missing value.

Based on our knowledge, there are many studies on KNN imputation method for microarray data such as KNNFS proposed by [8] and Weight KNN imputation proposed by

* *Program in Data Science, Faculty of Science and Technology, Nakhon Pathom Rajabhat University, Thailand.*

** *Department of Computer and Information Technology, Faculty of Science, Thaksin University, Thailand.*

[9] but none of them do study the effect of different distance measure on KNN imputation method. In this paper, KNN-Ensemble imputation method is proposed. The main point of our approach is using an appropriate distance function for KNN. Two best distance functions will be used in KNN-Ensemble method in two different ways: (1) one for calculating the distance between rows and (2) another one for calculating the distance between columns simultaneously.

The rest of this paper is organized as follows. Section II gives a background of KNN imputation, distance measures and literature review. Section III the proposed method was describe. Section IV reports the experimental results. Section V concludes the paper and mentions future work.

2. Background

2.1 KNN Imputation Method

Due to its simplicity, one of the well-known missing value imputation methods is K-Nearest Neighbor (KNN) method especially to deal with missing value in microarray data. The KNN method imputes missing values by selecting genes with expression values similar to the gene of interest. The steps of KNN imputation are as follows.

First, k genes that are most similar to the gene with the missing value were chosen. Second, measure the distance between a gene that contain missing value and genes obtained from the first step. At this point, the Euclidian distance will be used to calculate the distance between those genes. Third, estimate the missing value as an average of the k nearest neighbor genes, corresponding entries in the selected k expression. There are many distance functions can be used to calculate the distance for KNN imputation [10] such as Euclidian distance, Manhattan distance, Cosine and Pearson correlation. Note that, for more detail of each distance function see in Section 2.2 – 2.5.

2.2 Euclidian Distance

Euclidian distance use to measure the distance between two expression vectors xi and xj. Euclidean distance between xi and xj can be calculated from Equation (1)

$$dist(x_i, x_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \tag{1}$$

Where dist(x_i, x_j) is the Euclidean distance between samples x_i and x_j; n is the number of dimension of microarray; and x_{ik} is the k th feature of sample x_i.

2.3 Manhattan Distance

Manhattan distance use to measure the distance between two vectors (also known as the L1 distance). Manhattan distance between x_i and x_j is defined as below-see Equation (2)

$$dist(x_i, x_j) = \sqrt{\sum_{k=1}^{N}\left|x_{ik} - x_{jk}\right|} \tag{2}$$

Where dist(x_i, x_j) is the Manhattan distance between samples x_i and x_j; N is the number of dimension of microarray; and x_{ik} is the k th feature of sample x_i .

2.4 Cosine Distance

Cosine distance distcos(x_i, x_j), gives angular cosine distance between the two vectors, is defined as below

$$dist_{\cos}(x_i, x_j) = \frac{\sum_{k=1}^{N} x_{ik}x_{jk}}{\sqrt{\sum_{k=1}^{N} x_{ik}^2}\sqrt{\sum_{k=1}^{N} x_{jk}^2}} \tag{3}$$

Where distcos(x_i, x_j) is the Cosine distance between samples x_i and x_j; N is the number of dimension of microarray

2.5 Pearson Correlation

Pearson correlation coefficient (r) between x and y is defined as

$$r(x, y) = \frac{\sum_{k=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{k=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{4}$$

Where $r(x, y)$ is the Pearson correlation coefficient between samples x and y; N is the number of dimension of microarray Where x_i and $\bar{x}$ are the i-th value of variable x and average value of x; y_i and $\bar{y}$ are the i-th value of variable y and average value of y.

2.6 Relate Works

As mention before, there are many studies about missing value imputation method. In this section, we classify reviewed literatures into two groups: (1) literature related to new imputation methods in general and (2) literature related to KNN imputation method especially.

As mention before, for the first group, there are many literatures proposed new imputation methods such as [5] proposed BPCA-iLLS imputation method. This method combined Bayesian principal component analysis (BPCA) and local least squares (LLS) for estimate missing value. The result of their method show that BPCA-iLLS imputation method obtained the good result on datasets.

In [6] proposed the hybrid imputation method using neural network and weighted KNN. They ran experiment on 4 different datasets. The experimental result show that proposed method obtained the better result when compare with neural network and genetic imputation method. In this paper, RMSE was used to measure the performance of imputation method.

A new missing value imputation using genetic algorithms was proposed by [11]. In this work, authors introduced the estimation method for missing value by applying the GA-based feature selection. The aim of GA in this work is to selecting appropriate features for each missing value.

Another study proposed by [7]. In this paper, they performed a comprehensive comparison by comparing nine imputation methods on thirteen datasets and three types of measures were used to evaluate the performance of each imputation algorithm. This study found that local-least-squares-based methods are a good choices to handle missing values for most of the microarray datasets.

More precisely, in this work, we focus on KNN-based imputation methods. There are some literatures which proposed a variety version of KNN imputation method for example; KNN-Weight imputation method proposed by [9]. This method estimate the values of missing data using a weighted–nearest neighbor algorithm. The Support Vector Regression (SVR) was used to quantify the weights for KNN. However this approach use the original Euclidian distance function and ignore the influence of different distance function on KNN.

KNNFS proposed by [8]. The idea behind proposed methodology is a combination of KNN-Based Feature Selection and KNN-based imputation (KNNFS Impute). Authors compared the performance of the proposed method with traditional KNN and Row average imputation methods on three microarray data sets. The best estimation results are measured by the minimum Normalized Root Mean Squared Error (NRMSE). The results show that the proposed method outperformed other baseline imputation methods on the three data sets with smaller NRMSE. Again, this approach consider the importance of attributes which can affect the performance of imputation method but still use only Euclidian distance for measure the distance between rows.

In [12] proposed a novel KNN imputation procedure using a feature-weighted distance metric based on mutual information (MI), called MI-KNNimpute. This method selects the k nearest cases considering the input attribute relevance to the target class. Then, MI metric was compute for an incomplete input feature (considering only the training cases with known values in the attribute of interest). Next, the weighted imputation schemes was used for MI-KNNimpute method to estimate missing value. However, this method consider the MI between attribute and target class to improve the predictive accuracy. It suits for only classification task while our propose method did not take target class into account which make our method more generic for both classification and clustering task.

In [13] nearest neighbor selection for iteratively KNN imputation, in short, GKNN was proposed. This algorithm selects k nearest neighbors for each missing data via calculating the gray distance instead of the traditional Euclidean distance. However, this distance function used to calculate the distance only one dimension (consider only distance between column) while our approach consider the distance between row and column simultaneously.

Recall that, in this paper, we focus on KNN imputation method on microarray dataset. There are many literatures proposed different versions of KNN imputation method but none of them do study the effect of different distance measures on KNN imputation method. In order to improve the performance of KNN imputation method, two best distance functions will be used in KNN-Ensemble imputation.

3. Proposed Method

In this paper, we proposed the KNN imputation method using ensemble technique. The idea of KNN-ensemble model is using the best distance function in two different ways: (1) KNN Distance-Row (KNN-Row) which uses the best distance function to compute the distance between sample and sample for KNN imputation method and (2) KNN Distance-Column (KNN-Column). In this way, we use the best distance function to compute the distance between feature and feature for KNN imputation. Recall that, in our experiment, the feature is a gene corresponding to each sample.

To find the best distance function for KNN-Ensemble method, we decided to compare four different distance functions in our experiment which are Euclidian distance, Manhattan distance, Cosine distance and Pearson correlation. Note that, the details of each distance function mentions in Section 2. Then, the best distance function for KNN-Row and the best distance function for KNN-Column were selected for KNN-Ensemble method.

The overview of proposed KNN-Ensemble model is shown in Figure 1. From three datasets, first, we randomly generate the missing value on three datasets according to 18 different percentage of missing value. Second, estimate the missing value using KNN imputation method using four distance functions. Four different distance functions were used to calculate are Euclidian distance, Manhattan distance, Cosine distance and Pearson correlation. For KNN-Row perspective, we run KNN imputation using Euclidian distance function (named "KNNR-Euclidian") and replace the missing value with the estimated value obtained by KNNR-Euclidian. Then, KNN imputation with other distance functions will run and replace the missing value with the estimated value obtained by each distance function. We name each approach with "KNNR" following with functions name such as KNNR-Euclidian stands for KNN imputation using Euclidian function, KNNR-Manhattan stands for KNNR using Manhattan function, KNNR-Cosine and KNNR-Pearson stand for KNNR using Cosine and KNNR using Pearson function, respectively.

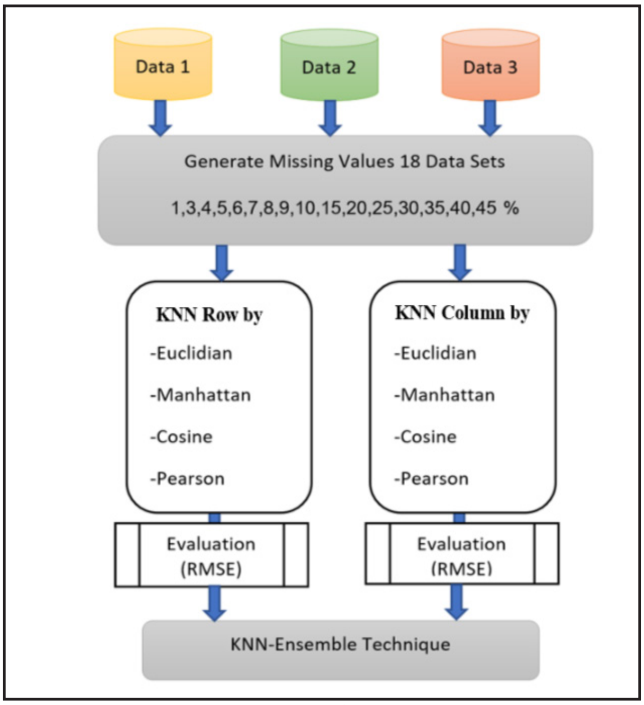

Figure 1. *The overview of KNN-Ensemble Model.*

Then again, those four distance functions were used to calculate the distance between feature and feature (Knn-Row perspective). Those approaches named KNNC-Euclidian, KNNC-Manhattan, KNNC-Cosine and KNNC-Pearson. Next, the Root Mean Square Error: (RMSE) matric was used to evaluate the performance of each approaches. Then, the best distance function for KNN-Row and the best distance function for KNN-Column were selected for KNN-Ensemble method.

More precise, for KNN-Ensemble imputation method, the estimated value is the average value of the estimated value from KNNR imputation and the estimated value from KNNC imputation.

4. Experimental Results and Discussion

4.1 Datasets and Experiment Setup

We perform the following experiments on three cancer benchmark microarray datasets: Colon cancer, ovarian cancer and Prostate cancer. The benefit of using benchmark dataset is it already has a complete data which will be used to estimate the error of estimation methods. Dataset characteristic shows in Table 1. The first column is a dataset name, the second and the third column is a number of features and number of samples in dataset, respectively. For the last column shows number of class in dataset when the first number in bracket is the number of sample which have class "has cancer" while another number in bracket is the number of sample which have class "no cancer".

Table 1. *Dataset characteristic.*

Dataset Name	No.features	No.samples	No.classes
Colon	2,000	62	2(40:22)
Ovarian	15,154	253	2(162:91)
Prostate	12,600	102	2(52:50)

Before running experiment, first, we remove all records with missing values from original datasets. Second, missing values were randomly generated with 1, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45 and 50 % on those three datasets. Next, estimated values are calculated and replaced using three imputation methods: (1) the proposed KNN-Ensemble method (2) KNN imputation method and (3) Row-Average imputation method. Last, performance of each methods will be evaluated using Root Mean Square Error: RMSE. The overview of performance evaluation model shows in Figure 2.

4.2 Evaluation Metric

In this paper, we use a Root Mean Square Error: RMSE to evaluate the performance of KNN-Ensemble method and other two baseline methods named (1) Row Average imputation method and (2) KNN imputation. RMSE use to calculate the error between the estimated values and true value in datasets [14]. RMSE equation shows in Equation (5), where N is the number of missing value in dataset, R_i is the real value *i* and E_i is the estimated value *i*.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(R_i - E_i)^2}{N}} \tag{5}$$

4.3 Experimental Results

To finding the best distance function for KNN-Row and KNN-Column, we run experiment on three datasets using KNN imputation with four different distance measures.

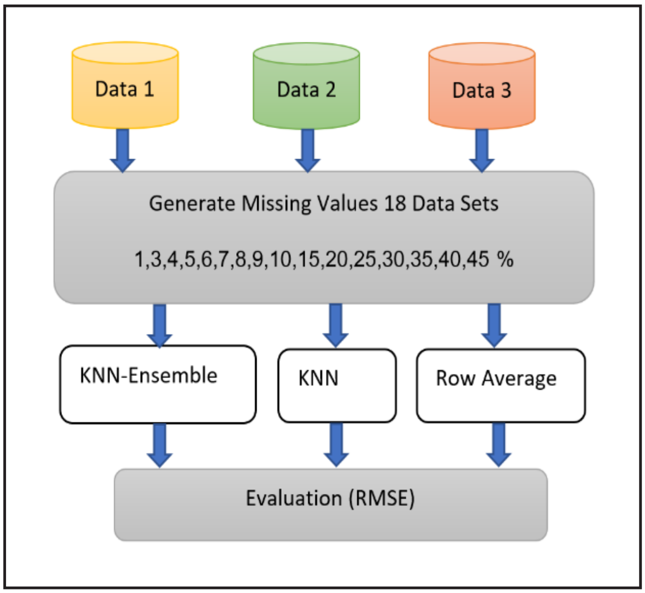

Figure 2. *Evaluation Model.*

Figures 3-5 show RMSE obtained from KNNR-Euclidian, KNNR-Manhattan, KNNR-Cosine and KNNR-Pearson on Colon, Prostate and Ovarian dataset. Recall that, KNNR is a KNN imputation method which consider the distance between sample and sample. Also, note that in our experiment we set the number of k = 5.

Clearly, in Figures 3-5, the KNNR-Manhattan obtained the smaller RMSE than KNNR-Euclidian, KNNR-Cosine and KNNR-Pearson on two out of three datasets (Colon and Prostate datasets).

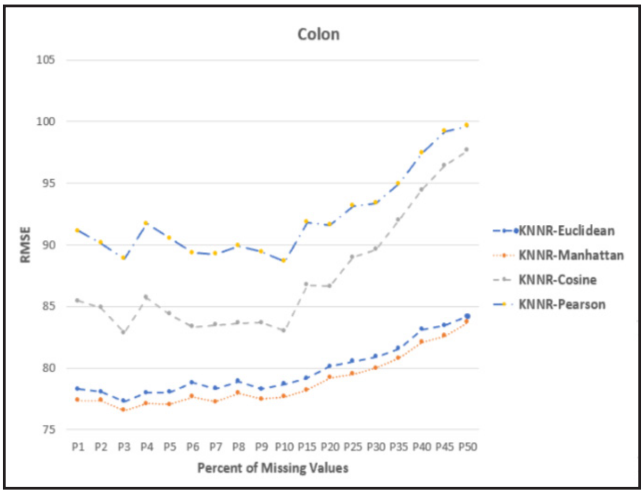

Figure 3. *The experimental results for KNNR and four different distance functions on Colon dataset (when k=5).*

In Figure 3, KNNR-Manhattan outperformed other versions of KNNR (KNNR-Euclidian, KNNR-Cosine and KNNR-Pearson) with the smallest RMSE in every case (1, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45 and 50 % of missing value). The second best is KNNR-Euclidian while KNNR-Pearson obtained the largest RMSE in all case.

In Figure 4, again, KNNR-Manhattan outperformed other versions of KNNR (KNN-Euclidian, KNNR-Cosine and KNNR-Pearson) with the smallest RMSE while KNNR-Cosine obtained the largest RMSE in all case.

In Figure 5, surprisingly, KNNR-Manhattan and KNNR-Euclidian obtained the better result (smaller RMSE) than other two versions of KNNR (KNNR-Cosine and KNNR-Pearson).

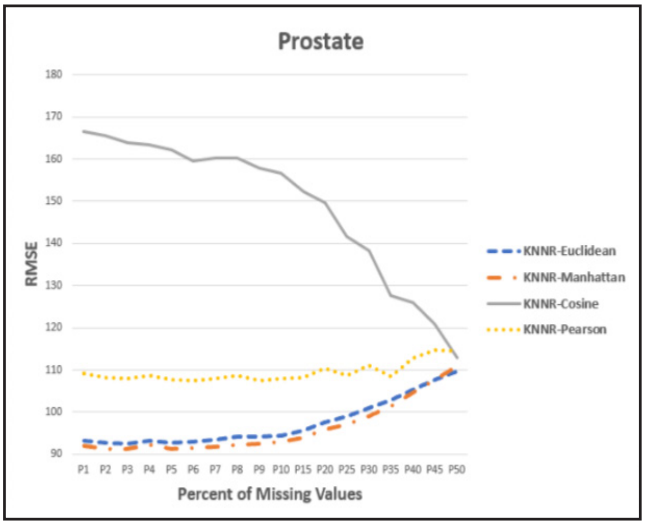

***Figure 4.** The experimental results for KNNR and four different distance functions on Prostate dataset (when k=5).*

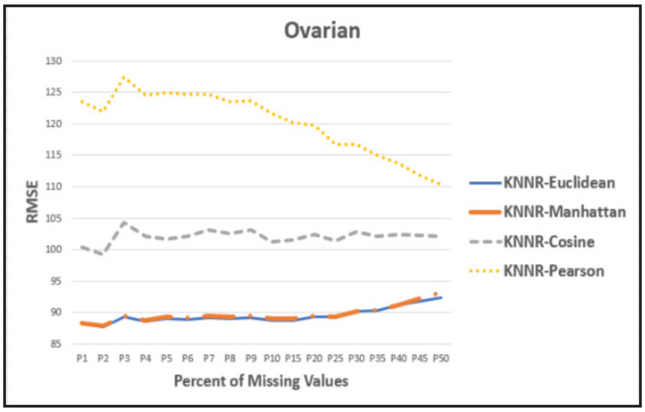

***Figure 5.** The experimental results for KNNR and four different distance functions on Ovarian dataset (when k=5).*

Again clearly, in Figures 6-8, KNNC-Manhattan outperformed with smaller RMSE than other versions of KNNC (KNNC-Euclidian, KNNC-Cosine and KNNC-Pearson) on Colon and Prostate datasets while on Ovarian dataset KNNC-Manhattan and KNNC-Euclidian show the same performance on this dataset.

In Figure 6, KNNC-Manhattan outperformed other versions of KNNC (KNNC-Euclidian, KNNC-Cosine and KNNC-Pearson) with the smallest RMSE in every case. The second best is KNNC-Euclidian while KNNC-Pearson obtained the largest RMSE in all case.

In Figure 7, KNNC-Manhattan again outperformed other three versions of KNNC (KNNC-Euclidian, KNNC-Cosine and KNNC-Pearson) with the smaller RMSE while KNNC-Cosine obtained the larger RMSE in all case.

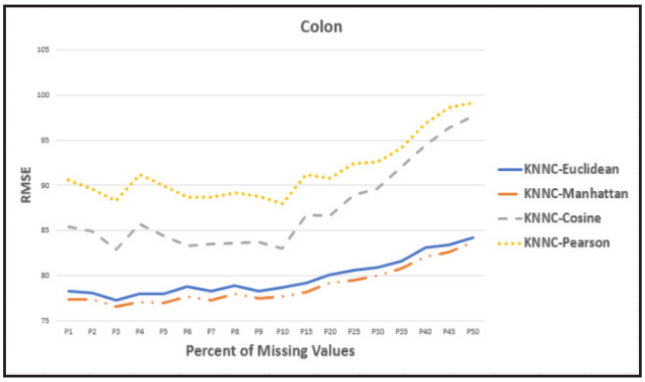

***Figure 6.** The experimental results for KNNC and four different distance functions on Colon dataset (when k=5).*

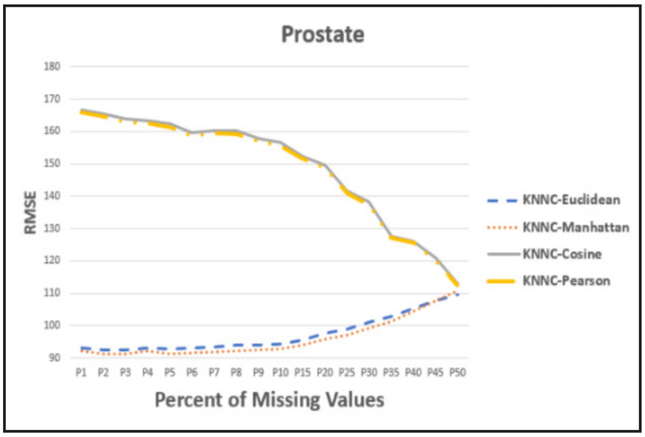

***Figure 7.** The experimental results for KNNC and four different distance functions on Prostate dataset (when k=5).*

In Figure 8, KNNC- Manhattan and KNNC-Euclidian obtained the better result (smaller RMSE) than other two versions of KNNC (KNNC-Cosine and KNNC-Pearson).

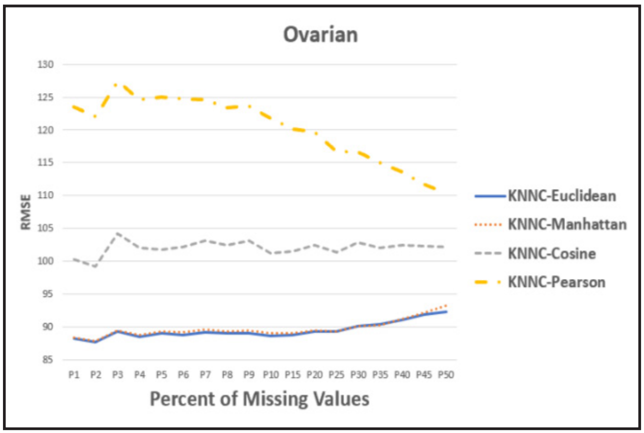

Figure 8. The experimental results for KNNC and four different distance functions on Ovarian dataset (when k=5).

To summarize the result in Figures 3-8, on three dataset, the Experimental results show that Manhattan and Euclidian distance function outperform other distance functions for KNNC and KNNR in our experiment. After that, these two best distance functions will be used in KNN-Ensemble model.

4.4 Performance Comparison and Discussion

The experimental results for KNN-Ensemble, Row-Average and KNN imputation method on three dataset are shown in Figures 9-11. Similarity to the previous section, we run experiment using KNN-Ensemble, Row-Average and KNN imputation method on Colon, Prostate and Ovarian datasets with the different 18 missing value rate.

In general, KNN-Ensemble imputation method outperformed Row-Average and KNN imputation method with obtained the smaller RMSE.

In Figures 9-10, KNN-Ensemble imputation method obtained the smaller RMSE. The second best is KNN imputation method while the Row-Average imputation method obtained the largest RMSE on Colon and Prostate dataset.

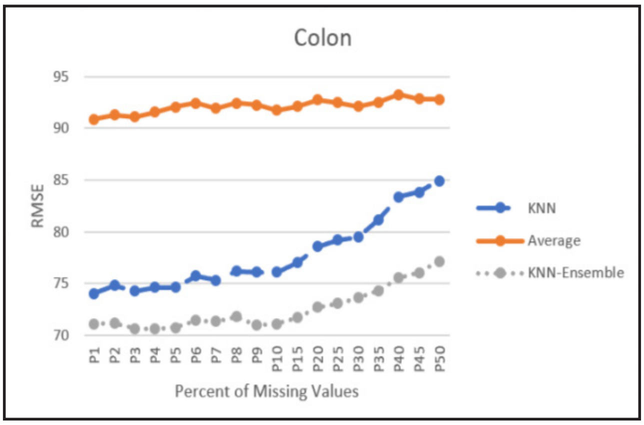

Figure 9. The experimental results for KNN-Ensemble, Row-Average and KNN imputation method on Colon dataset.

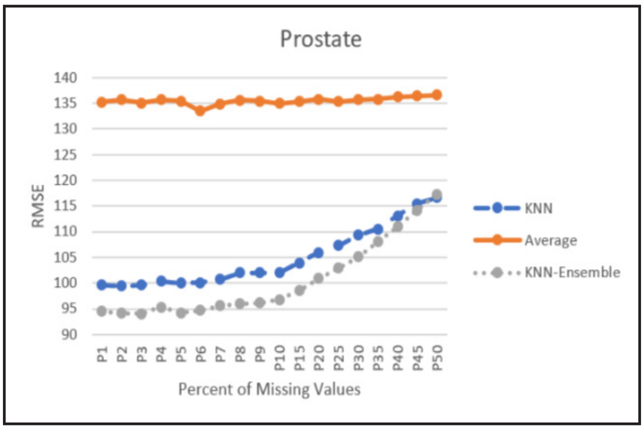

Figure 10. The experimental results for KNN-Ensemble, Row-Average and KNN imputation method on Prostate dataset.

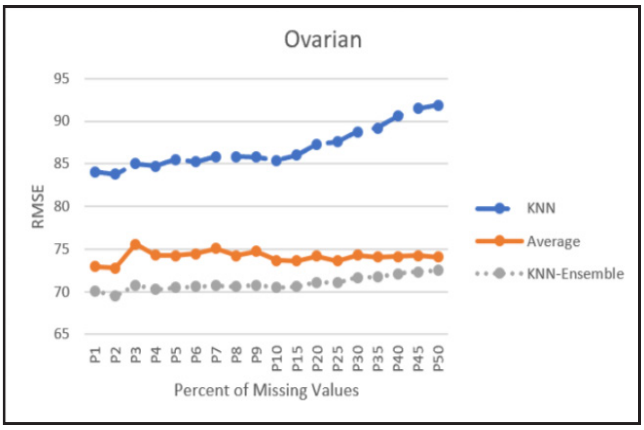

Figure 11. The experimental results for KNN-Ensemble, Row-Average and KNN imputation method on Ovarian dataset.

In Figure 11, KNN-Ensemble imputation method again obtained the smaller RMSE. Surprisingly, the second best is Row-Average imputation method while the KNN imputation method obtained the largest RMSE on Colon and Prostate dataset. Note that, on this dataset the Row-Average outperforms KNN because data points in Ovarian dataset are close to the average value.

To summarize the results, KNN-Ensemble imputation method outperformed (obtained the smaller RMSE) Row-Average and KNN imputation method across three datasets while the second best is KNN imputation method on Colon and Prostate dataset. On Ovarian dataset the second best is Row-Average imputation method.

Row-Average shows the worst result (obtained the larger RMSE) when compare with KNN and KNN-Ensemble in two cases because Row-Average calculates the missing value from all data points in only one particular column and ignores other columns completely.

KNN imputation method takes the k rows into account when estimate missing values. KNN selects k rows based on the distance value which calculated from every features including the outlier feature in dataset.

The reason why KNN-Ensemble is more effective when comparing with Row-Average and KNN imputation method seems to be because KNN-Ensemble estimates missing values based on the selected k rows and selected k columns. In this scenario, KNN-Ensemble calculates each missing values using the most similar data points from selected rows and selected columns.

5. Conclusion and Future Works

In this work, a new missing value imputation method for microarray data using ensemble technique named KNN-Ensemble was proposed. The idea of KNN-ensemble model is using the best distance function in two different ways: (1) to calculate the distance between rows and (2) to calculate the distance between columns. Four different distance functions: Euclidian distance, Manhattan distance, Cosine and Pearson correlation were studied over three benchmark datasets. From experimental results show that the best distance function for evaluating distance between feature and feature is Euclidian function and the best distance function for evaluating distance between samples is Manhattan function. Then, KNN-Ensemble model was built based on two selected functions. Moreover, we compare the performance of KNN-Ensemble with other two well-known imputation methods named KNN imputation and Row-Average imputation method. Finally, the experimental results show that KNN-Ensemble method using Manhattan and Euclidian function outperformed other baseline imputation method across three datasets.

For future work, we will study on the effect of the normalization on KNN-Ensemble imputation method. Also, we will compare the performance of KNN-Ensemble and other methods in different perspective such as the classification predictive accuracy.

6. References

[1] D. M. Dziuda. *Data Mining for Genomics and Proteomics: analysis of gene and protein expression data*. New Jersy: Wiley and Sons., 2010

[2] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, and R. B. Altman. Missing value estimation methods for DNA microarrays. Bioinformatics, Vol. 17, No. 6, pp. 520-525., 2001.

[3] M. C. De Souto, P. A Jaskowiak, and I. G. Costa. “Impact of missing data imputation methods on gene expression clustering and classification.” *BMC bioinformatics*, Vol. 16, No. 1, pp. 64, 2015

[4] M. S. B. Sehgal, I. Gondal, and L. S. Dooley. “Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data.” *Bioinformatics*, Vol. 21, No. 10, pp. 2417-2423, 2005.

[5] F.X. Shi, D. Zhang, J. Chen, and H.R. Karimi. “Missing Value Estimation for Microarray Data by Bayesian Principal Component Analysis and Iterative Local Least Squares.” *Mathematical Problems in Engineering*, pp. 1-5, 2013.

[6] I.B. Aydilek and A. Arslan. “A novel hybrid approach to estimating missing values in databases using K-nearest neighbors and neural networks.” *International Journal of Innovative Computing, Information and Control*. Vol. 8, No. 7, pp. 4705-4717, 2012.

[7] C.C. Chiu, S.Y. Chan, C.C. Wang, and Wu W.S. *Missing value imputation for microarray data: a comprehensive comparison study and a web tool.* BMC Systems Biology, 7 (Suppl 6), S12. Available Online at http://doi.org/10.1186/1752-0509-7-S6-S12, 2013.

[8] P. Meesad and K. Hengpraprohm. "Combination of knn-based feature selection and knnbased missing-value imputation of microarray data." *In Innovative Computing Information and Control, 2008. ICICIC'08. 3rd International Conference on IEEE*. pp. 341-341, June 2008.

[9] W. Ling and F. Dong-Mei. "Estimation of missing values using a weighted k-nearest neighbors algorithm." *In Environmental Science and Information Application Technology, 2009. ESIAT 2009. International Conference on IEEE*. Vol. 3, pp. 660-663, July 2009.

[10] L. Y. Hu, M. W. Huang, S. W. Ke, and C. F. Tsai. "The distance function effect on k-nearest neighbor classification for medical datasets." *SpringerPlus*, Vol. 5, No. 1, pp. 1304, 2016.

[11] K. Hengpraphrom, S. N. Wichian, and P. Meesad. "Missing value imputation using genetic algorithm." *In The 3rd International Symposium on Intelligent Informatics the 1st International Symposium on Information and Knowledge Management*, 2010.

[12] P. J. García-Laencina, J. L. Sancho-Gómez, A. R. Figueiras-Vidal, and M. Verleysen. "K nearest neighbours with mutual information for simultaneous classification and missing data imputation." *Neurocomputing*, Vol. 72, No. 7-9, pp. 1483-1493, 2009.

[13] S. Zhang. "Nearest neighbor selection for iteratively kNN imputation." *Journal of Systems and Software*, Vol. 85, No. 11, pp. 2541-2552, 2012.

[14] I. B. Aydilek and A. Arslan. "A novel hybrid approach to estimating missing values in databases using k-nearest neighbors and neural networks." *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 8, pp. 4705-4717, 2012.