IMPUTATION OF MISSING DATA USING CO-CLUSTER SPARSE MATRIX LEARNING MODEL FOR GENOME DATA

[1]Mrs. J. Sujitha, [2]Mrs. S.R. Lavanya,
[1]Research Scholar, [2]Assistant Professor,
Department of Computer Science,
Sri Ramakrishna College of Arts and Science for Women,
Coimbatore, Tamil Nadu, India.

Abstract: Missing data imputation is a challenging issue in data mining pre-processing techniques. Most of the data mining algorithms cannot process a dataset with continuous missing data. The continuous missing data in a dataset can affect the performance of the mining process of data which leads to the difficulty of extracting information from datasets. The existing system's MIAEC algorithm is not applicable to find the continuous missing data and takes much time to find the missing values in large datasets with continuous data loss. To overcome this problem, the proposed method is meant for continuous missing data imputation based on co-cluster sparse matrix learning (CCSML) model. This algorithm learns without reference class, and even with continuous missing rate data as high as 90%. The method works based on a tensor optimization model and labeled maximum block. The computational method of sparse recovery learning model is built on low-rank, co-clusters of GWAS data matrices and the performance is better than the existing method of missing data imputation using the Evidence Chain (MIAEC) algorithm.

Keywords: **Missing at Random, Missing completely at Random, Missing Not at Random, MIAEC algorithm, sparse low-rank matrix completion, Co-clustering factorization, Maximum blocks improvement.**

I. INTRODUCTION

Missing data is a common problem in many fields of human endeavor ranging from Social sciences to Economics and from Political research to the Entertainment industry. In survey conducting fields, missing data occurs when people refuse to answer specific questions or some people cannot be contacted. Missing data may arise from numerous different reasons. In surveys, missing data is typically a consequence of non-response; some questions might be irrelevant for some respondents, respondents may end up interrupting the survey or results of multiple different surveys with different questions may be analyzed together. In other contexts, missing data may be caused, for example, by equipment failure or data corruption. The existence of missing values is common in real-world data. If the occurrence of the missing data is completely at random, we can simply remove it. In many cases missing values happen when there is a human mistake when collecting the data or some information is damaged and it will make the data biased if we remove those missing observations. With statistical tools, the missing values can be imputed so that the maximum amount of information is restored while keeping the data unbiased. This work demonstrates some of the popular statistical methods for imputing missing values. The challenge of missing data pose to the decision-making process is more evident in online applications where data have to be used almost instantly after being obtained. Computational intelligence techniques are neural networks and other pattern recognition. In the mentioned techniques, have newly become very common tools in decision-making processes. In a case where some variables are not considered, it becomes hard to continue with the decision-making process. The major challenge is that the standard computational intelligence techniques are not able to process input data with missing values and hence, cannot perform classification or regression. Some of the reasons for missing data are sensor failures, omitted entries in databases and non response to questions in questionnaires. There have been many techniques to estimate the missing data for some applications. For the majority of the techniques that have been discussed in the literature, significant the reasons why the data are missing become very helpful in choosing the right technique to approximate missing data. In most applications, there is restricted time between the readings depending on how normally the sensor is sampled. In classification and regression tasks, all decisions concerning how to proceed must be taken during this time period.

1.1 Missing at Random (MAR)

Missing at Random requires that the cause of missing data be unconnected to the missing values themselves. However, the cause may be linked to other observed variables. MAR is also known as the ignorable case and occurs when cases with missing data are different from the observed cases but the pattern of missing data is predictable from other observed variables. Separately said, the cause of the missing data is due to external effect and not to the variable itself. Suppose there are two sensors namely S and T. For MAR to set, the probability of datum d from a sensor S to be missing at random should be dependent on other measured variables in the database. Data item are not missing at random, but the probability that a value is missing depends on values of variables that were accurately measured. As an instance, look at a survey in which females are less likely to provide their private income in general (but the possibility of acknowledging is independent of her real income). If we know the sex of every subject and have income levels for some of the females, equitable sex-specific income estimates can be finished. That is because the incomes for some of the females are a random sample of all females' incomes.

1.2 Missing Completely at Random (MCAR)

Missing completely at Random refers to a condition where the possibility of data missing is distinct to the values of any other variables, whether missing or observed. In this mechanism, cases with entire data are impossible to differentiate from cases with incomplete data. In this case, the probability of sensor N values missing is independent of any experiential data and the missing value is not dependent on the prior state of the sensor or any reading from any other sensor.Data elements are missing for reasons that are unrelated to any characteristics or responses for the subject, including the value of the missing data, where it to be known. Examples contain missing laboratory measurements because of a dropped test tube (if it was not dropped because of knowledge of any measurements) and a review in which a subject omitted its response to a question for reasons unrelated to the response, the subject would have made or to any other of her characteristics.

1.3 Missing Not at Random (MNAR)

Missing not at random (MNAR) indicates that the missing data method is related to the missing values. An example of data missing not at random can result from a position where two databases from different cities where merged. Assume one database lacks some features that have been considered on the other database. In this condition, why a few data are missing can be explained. However, this explanation is only dependent on the same variables that are missing and cannot be explained in terms of any other variables in the database. Another case of MNAR will be when a sensor excursion if the value read, is above a certain threshold. In this case, the probability of L missing is reliant on L itself. MNAR is also referred to as the non-ignorable case as the missing observation is dependent on the outcome of interest. In this case, the measuring from S might be missing only because sensor T is not working. Elements are more likely to be missing if their true values of the variable in question are scientifically higher or lower. In a discussion, this situation can be given as an example of not missing at random mechanism when subjects with lower income levels or very high incomes are less likely to provide their personal income. These distinctions of mechanisms are important because when missing data mechanism is MCAR unbiased estimates will be produced even with rather primitive analysis methods. When missing data mechanism is MAR, unbiased measures will be created if a model and estimation technique is used that renders the missing value mechanism ignorable. When missing data mechanism is MNAR, an analysis method must be used that includes both a model for the observed data and a model for the missing value mechanism. For missing data that are MCAR or MAR, common modeling software is available, that produces balanced using all the available information. There are no easy solutions for missing data in MNAR. It is impossible to remove completely missing data. The requirement is to use missing data estimation methods which base estimation on the observed (non-rectangular) data only.

II. RESEARCH METHODOLOGY

2.1 MIAEC ALGORITHM

Most existing missing data imputation methods can be divided into two categories based on the probability of statistical analysis and data mining. The existing algorithm MIAEC obtains all relevant evidence of missing data in each tuple of missing data by mining, which is further combined to form a chain of evidence to estimate the missing attribute value. Finally, the value of the missing data is estimated by the chain of evidence. It does not need to master the distribution of data in the dataset, domain knowledge, and does not need to train the dataset estimation model for the imputation to reduce time cost. Most existing algorithms are designed to deal with small datasets on a single machine, but now with the development of information technology, the rapid growth of data, large-scale data processing on a single machine is clearly inappropriate. In MIAEC, Map-Reduce programming model is used to implement the proposed MIAEC algorithm for imputation of large-scale datasets on distributed platforms.

MIAEC is based on the set of associated attribute value combinations of missing data as a chain of evidence to estimate the value of missing data. According to the idea of data mining, there is a certain relationship between data attribute values in large-scale datasets. In the imputation process, the algorithm will first estimate the missing value of the imputation value, the algorithm scans each data tuple in the entire dataset, marking tuples with missing values `?' as incomplete data tuples, and combine the different associated attribute values of the missing data in the incomplete data tuple as evidence of the estimated missing value. So that a large number of combinations of the relevant attributes for missing data in the incomplete tuple constitute the estimated missing data value of the chain of evidence. The algorithm scans the entire dataset again and counts the combination of attribute values in all data tuples, and uses the relevant theorem in the prerequisite knowledge to calculate the value of the missing data. The core task of the algorithm is to calculate the reliability of each estimated missing value in the chain of evidence. Thus, gives the sum of the confidence of all the evidence for the estimated missing data, the maximum estimate of the sum of the confidence values is selected as the imputation value.

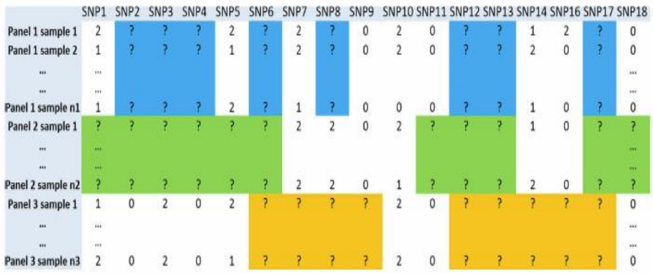
2.2 DISADVANTAGES OF EXISTING SYSTEM

- The missing attributes tuple, nor performs imputation of the missing value, directly in the missing data in the dataset analysis and mining.
- The parameter convergence is very slow and time-consuming.
- Large percentage of missing data, MIAEC filling accuracy will be greatly reduced.
- The linear regression for imputing missing values is extremely poor.
- The estimations of parametric method may be very biased and the optimal control factor settings may be miscalculated.
- Leads to a loss of helpful characteristics in case of the continuous missing attributes.
- May not be sufficient to non-parametrically estimate the relationship during continuous missing attributes in the cell.

III. PROPOSED SYSTEM

The current approaches for haplotype inference (i.e., the details that are not directly available from the high-throughput genotyping platform) and missing data imputation usually process the genotype data, and each sample is phased and modeled as mosaic of those haplotypes of the reference panel. The proposed approach could conduct imputation for both haplotype and genotype data matrices from different cohorts using different genotyping chips. The missing valued matrices of the genotype data is provided with SNPs or the phased haplotype data. The imputation problem is to impute the missing data entries of the data matrices. The illustrative formats of the data matrices are given with the diploid genotype data and the corresponding phased haplotype data. For the data matrices, each row corresponds to one individual sample and each column corresponds to one SNP.

The natural and flexible modeling framework which utilizes information across multiple reference panels and study panels which achieves high recovery accuracy even when the data matrices have high percentages of missing entries. Proposed approach combines the multiple chosen reference panels and the different panels together as a large whole data matrix with missing entries.



	SNP1	SNP2	SNP3	SNP4	SNP5	SNP6	SNP7	SNP8	SNP9	SNP10	SNP11	SNP12	SNP13	SNP14	SNP16	SNP17	SNP18
Panel 1 sample 1	2	?	?	?	2	?	2	?	0	2	0	?	?	1	2	?	0
Panel 1 sample 2	1	?	?	?	1	?	2	?	0	2	0	?	?	2	0	?	0
...	...																...
...	...																...
Panel 1 sample n1	1	?	?	?	2	?	1	?	0	0	0	?	?	1	0	?	0
Panel 2 sample 1	?	?	?	?	?	?	2	2	0	2	?	?	?	1	0	?	?
...	...																...
...	...																...
Panel 2 sample n2	?	?	?	?	?	?	2	2	0	1	?	?	?	2	0	?	?
Panel 3 sample 1	1	0	2	0	2	?	?	?	?	2	0	?	?	?	?	?	0
...	...																...
...	...																...
Panel 3 sample n3	2	0	2	0	1	?	?	?	?	2	0	?	?	?	?	?	0

Figure 1.1 Genotype data matrix with missing value

The idea of proposed approach is based on the following observation: although the large genotype data matrix, which usually has missing entries and which may be contaminated by noises and errors from experimental samples and sequencing technologies, appear to be very complex, the underlying structure of the data matrix contains essential "sparse" information. By "sparse", it means the data matrix has the mathematical property of low-rank or low number of co-clusters. The sparse property using large genotype data matrices and the testing results show the matrices are usually low-ranked.

3.1 DATASET

The dataset of chr22 setup information is detailed as follows. The dataset 1KG_chr22 was prepared to represent imputing SNPs from a reference panel typed on a different chip. This dataset consists of haplotypes for 1092 individuals, which split into a study and reference panel by assigning half of the individuals to each such that the distribution of ethnicities was preserved across both groups. For the study panel 60,000 SNPs is chosen from a randomly selected region on chromosome 22 and masked genotypes of all SNPs.

The panel consists of 85.3% systematically missing data for 546 individuals. For MIAEC and CCSML, there are two different models for the same target: matrix completion. Usually if it has a good estimation of the rank of the matrix, then it can decide the size of matrices Y_1, X, Y_2 and thus CCSML is preferred because it also provides the clustering information of individual samples and SNPs.

3.2 SPARSE LOW-RANK MATRIX COMPLETION

The sparse low-rank matrix completion model aims to fill in missing data values of a matrix based on the priori information that the matrix under consideration is of low rank. The low-rank matrix completion model can be formulated as the following optimization problem:

$$min_X \;\; rank(X), s, t., X_{ij} = M_{ij}(i,j) \in \Omega \qquad (1)$$

Where rank(X) denotes the rank of matrix X, and Ω denotes the index set of the known entries of M. That is, it is given a set of known entries of M, and wants to fill in the missing entries such that the completed matrix is of low rank. In the genotype missing data imputation problem, each row of the matrix M represents a patient sample, and each column of the matrix M corresponds to a SNP. That is, M_{ij} represents the $j\ th$ allele of the $i\ th$ patient sample. It is usually believed that patients can be classified into different categories and patients in the same category should have similar genetic patterns. Therefore, believe that the matrix M is low-rank, or at least numerically low-rank.

The sparse low-rank matrix model has been widely used in online recommendation, collaborative filtering, and computer vision and so on. Under certain randomness hypothesis, the model (1) is equivalent to the following convex optimization problem with high probability:

$$min_X \|X\|_*, s, t., X_{ij} = M_{ij}(i,j) \in \Omega \qquad (2)$$

Where ||X||* is called the nuclear norm of matrix X and is defined as the sum of singular values of X. The nuclear norm minimization problem (NNM) is numerically easier to solve than the propose model because it is a convex problem. Many efficient numerical algorithms have been suggested to solve the NNM model, use the fixed-point continuation method (FPCA) proposed.

The proposed imputation method implements Nesterov's accelerated proximal gradient method (APG) to solve (2), while FPCA can be seen as the ordinary version of proximal gradient method for solving (2). Theoretically, APG is faster than FPCA for solving LRMC, because the former attains an ε-optimal solution in $O\left(\frac{1}{\sqrt{\varepsilon}}\right)$ iterations, while the latter one attains an ε-optimal solution in $O\left(\frac{1}{\sqrt{\varepsilon}}\right)$ iterations. Mendel-Impute also implements two important techniques to further accelerate the speed of APG: the sliding window scheme to better balance the trade-offs between accuracy and running time, and the line search technique to find an appropriate step size for the proximal gradient step. From these experiments, it is found that the sliding window scheme is quite helpful for missing data imputation. Thus it is incorporated the sliding window scheme to LRMC, denoted as LRMC-s.

3.3 CO-CLUSTERING FACTORIZATION

In this proposed approach imputation is done based on matrix co-clustering (simultaneous clustering approach for each row) factorization. Co-clustering model for two-dimensional and higher-dimensional matrix co-clustering is based on a tensor optimization model and an optimization method termed Maximum Block Improvement (MBI) Inspired by the idea of matrix co-clustering for imputation, a basic model is developed as follows.

$$min_{\{A,X,Y_1,Y_2\}} \;\; f(A,X,Y_1,Y_2) := \|A - Y_1XY_2\|_F^2, s, t\, A_{ij} = M_{ij}(i,j) \in \Omega \qquad (3)$$

Where $A \in R^{m\times n}, \;\; Y_1 \in R^{m\times k_1}, X \in R^{k_1\times k_2}, Y_2 \in R^{k_2\times n}$

In (3), the Frobenius norm of a matrix X is defined as $\|X\|_F^2 = \sum_{ij} X_{ij}^2$ imputation approach, based on the matrix co-clustering factorization, aims to complete matrix *M* by using a low-rank matrix factorization model. In our framework, *A* is the data matrix with missing entries; Y_1 and Y_2 is the artificial row assignment matrix and the artificial column assignment matrix, respectively, and *X* is the artificial central-point matrix. Note that A is also an unknown decision variable in (3), because only a subset of its entries is known. Moreover, note that (3) requires the input of k_1 , which are closely related to the rank of the matrix to be completed. Therefore, in practice, if it has good estimation to the rank of the matrix, then (3) is a better model to use than (2), because it also provides the clustering information of individual samples and SNPs.

3.4 MAXIMUM BLOCK IMPROVEMENT

The proposed model is non-convex; it has some natural block-structure that can be utilized to adopt an efficient solution method. To solve the model (3) using a block coordinates update (BCU) procedure. There are four block variables in the model (3), namely A, X, Y_1 and Y_2 . The basic idea of BCU is, at iteration to minimize the function f with respect to one block variable while the other three blocks are fixed at the current known values. This idea is effective because it is observed that minimizing f for only one block variable among A, X, Y_1 and Y_2 is always relatively easy. A naive implementation of the BCU idea is to minimize f in the order of $A \rightarrow Y_1 \rightarrow X \rightarrow Y_2$, and in each step only one block variable is updated with the other three blocks being fixed. The proposed model, the matrix *X* actually plays a more important role than the other three blocks. As a result, it is beneficial if it is possible to update the *X* block more frequently than the other three blocks. Therefore, implemented the following four different algorithms based on the BCU idea.

"MBI-BL": This is an alternative of the MBI algorithm MBI-BL applies MBI algorithm in ref. 15 to minimize *f* with four blocks variables: *X*, ($Y_1 - X$), ($Y_2 - X$) and ($A - X$). In each block, for example, ($A - X$), alternating block minimization scheme is used to minimize *f* with respect to *A* and *X* alternating, until the function value ceases to change. After having attempted all four block variables, the block variable with maximum improvement is updated.

Algorithm: Maximum Block Improvement

Given initial iterates X^0, Y_1^0, Y_2^0, A^0, and initial values $v_0 = 0, v_1 = 1$.

For $K = 0,1 \ldots$ run the following until $|v_k - v_{k+1}| < \epsilon$

1) Block Improvement:

$$\bar{X}^{k,1} \leftarrow argmin_X \left\| A^k - Y_1^k X Y_2^k \right\|^2 \qquad (4)$$

$$\left(\bar{Y}_1^k \bar{X}^{k,2}\right) \leftarrow argmin_{(Y_1,X)} \left\| A^k - Y_1^k X Y_2^k \right\|^2 \qquad (5)$$

$$\left(\bar{Y}_2^k \bar{X}^{k,3}\right) \leftarrow argmin_{(Y_2,X)} \left\| A^k - Y_1^k X Y_2^k \right\|^2 \qquad (6)$$

$$(\bar{A}^k, \bar{X}^{k,4}) \leftarrow argmin_{(A,X)} \left\| A - Y_1^k X Y_2^k \right\|^2 s,t\ A_{ij} = M_{ij}(i,j) \in \Omega \qquad (7)$$

2) Compute the corresponding objective values:

$$w_1 = f\left(A^k, \bar{X}^{k,1}, Y_1^k, Y_2^k\right) \qquad (8)$$

$$w_2 = f\left(A^k, \bar{X}^{k,2}, Y_1^k, Y_2^k\right) \qquad (9)$$

$$w_3 = f\left(A^k, \bar{X}^{k,3}, Y_1^k, Y_2^k\right) \qquad (10)$$

$$w_4 = f\left(A^k, \bar{X}^{k,4}, Y_1^k, Y_2^k\right) \qquad (11)$$

3) Maximum Improvement: Compare w_1, w_2, w_3, w_4 pick up the smallest value to update the corresponding block variables:

- If w_1 is the smallest, then

$$X^{k+1} \leftarrow \bar{X}^{k,1} \leftarrow w_1 \qquad (12)$$

- If w_2 is the smallest, then

$$Y_1^{k+1} \leftarrow \bar{Y}_1^{k}, X^{k+1} \leftarrow \bar{X}^{k,2}, v_{k+1} \leftarrow w_2 \qquad (13)$$

- If w_3 is the smallest, then

$$Y_2^{k+1} \leftarrow \bar{Y}_2^{k}, X^{k+1} \leftarrow \bar{X}^{k,3}, v_{k+1} \leftarrow w_3 \qquad (14)$$

- If w_4 is the smallest, then

$$A^{k+1} \leftarrow \bar{A}^{k}, X^{k+1} \leftarrow \bar{X}^{k,4}, v_{k+1} \leftarrow w_4 \qquad (15)$$

All the algorithms are terminated when the objective value in the (k + 1)-th iteration does not decrease significantly from that in the k-th iteration.

IV. RESULT AND DISCUSSION

The results consists the combination of two types of imputation methods. They are CCSML (Co-cluster Sparse matrix Learning model) and MIAEC (Missing data imputation using the Evidence Chain).

4.1. Error rate

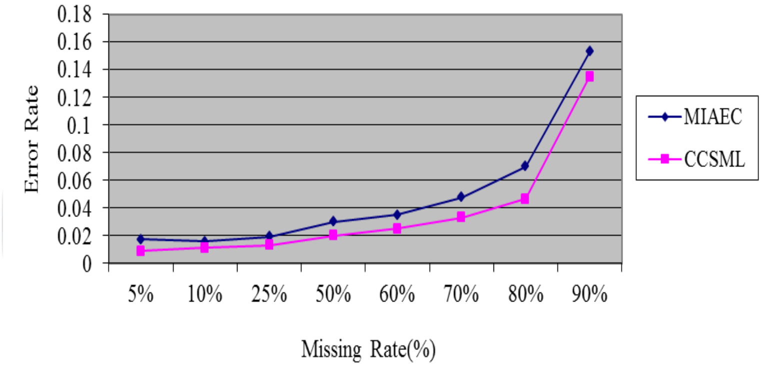


Figure 4.1 Comparison of Error rate estimation of missing genotypes for Chr22.

The above graph shows the error rate estimation between MIAEC and CCSML. The missing rate is calculated from 5% to 90%. In the graph X-axis shows missing rate and the Y-axis shows error rate. From the graph it is understood that the CCSML has lower error rate compare to MIAEC algorithm. In chr22 data, error rate are calculated based on the percentage of positive results returned that are relevant.

The formula for calculating percentage error is simple: [(|Approximate Value - Exact Value|) / Exact Value] x 100.

$$Error\ (\%) = \frac{|Approxmate\ value - exact\ Value|}{exact\ Value} \times 100$$

4.2. CPU Time

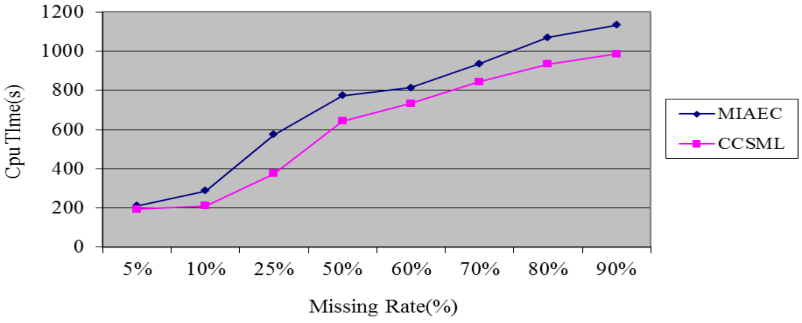


Figure 4.2 Comparison of CPU time for Chr22.

The above graph shows the computation time estimation between MIAEC and CCSML imputations. The missing rate is calculated from 5% to 90%. In the graph X-axis shows missing rate and the Y-axis shows CPU time. From the graph it is easily understood that the CCSML algorithm works faster than the existing MIAEC algorithm as the CPU time of proposed system is lower than the existing system.

The formula of the runtime is,

$$Run\ Time = (Start\ Time - End\ Time)$$

4.3. Imputation Accuracy

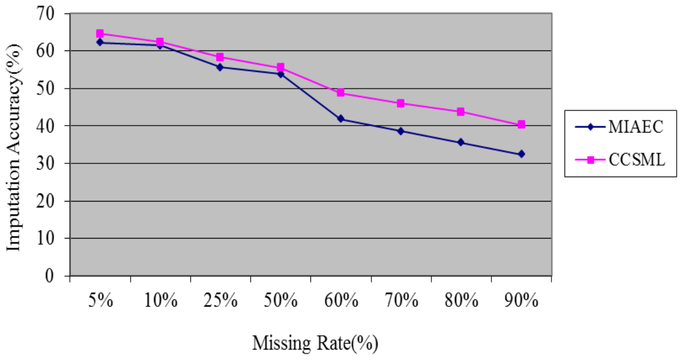


Figure 4.3 Imputation Accuracy of missing genotypes for Chr22

The above graph indicates the comparison of imputation accuracy of proposed (CCSML) algorithm with the existing (MIAEC) algorithm. In this graph X-axis shows the missing rate and Y-axis shows the imputation accuracy. From the graph it is understood that the CCSML algorithm imputation is higher accuracy while compared to MIAEC algorithm imputation. The following formula is used to calculate the accuracy,

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

(TP- True Positive, TN- True Negative, FP- False Positive, FN- False Negative)

V. CONCLUSION AND FUTURE ENHANCEMENT

The proposed system sparse recovery is meant for imputing missing genetic data in genome-wide association studies. Co-cluster sparse matrix learning (CCSML) models of sparse matrix are designed based on the sparse properties of low-rank and low numbers of co-clusters of the large, noisy, genetic datasets of matrices with missing data. The model would like to point out that the low-rank matrix completion model is similar to Mendel-Impute, but the matrix co-clustering factorization model is completely new. The proposed approach is able to effectively find patterns for imputation within study data, both with and without reference panels, and even with data missing rate as high as 90%. The performance of this approach with several other mainstream approaches for genotype imputation, sparse matrix is easy to use for metadata analysis, and it requires very simple, easy-to-process input file format and easy-to-interpret output result files. It has better or comparable performance compared to current state-of-the-art methods, especially for handling large sample size data with very different sets of SNPs and no reference panels.

In future work plan it can be further explored with including the extensive experiments of CCSML on more data sets, and further improvement of CCSML for reducing the time complexity.

REFERENCES

[1] R. K. Pearson, "The problem of disguised missing data," ACM SIGKDD Explorations News. Lett., vol. 8, no. 1, pp. 83-92, 2006.
[2] R. J. A. Little and D. B. Rubin, “Statistical Analysis With Missing Data,” 2nd ed. Hoboken, NJ, USA: Wiley, 2002, pp. 200-220.
[3] F. Z. Poleto, J. M. Singer, and C. D. Paulino, "Missing data mechanisms and their implications on the analysis of categorical data," Stat. Comput., vol. 21, no. 1, pp. 3143, Jan. 2011.
[4] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed attribute data sets", IEEE Trans. Knowl. Data Eng., vol. 23, no. 1, pp. 110-121, Jan. 2011.
[5] Y. Qin, S. Zhang, X. Zhu, J. Zhang, and C. Zhang, "Semi-parametric optimization for missing data imputation," Appl. Intell., vol. 27, no. 1, pp. 79-88, 2007.
[6] U. Dick, P. Haider, and T. Scheffer, "Learning from incomplete data with innite imputations," in Proc. 25th Int. Conf. Mach. Learn., Jul. 2008, pp. 232-239.
[7] Z. Shan, D. Zhao, and Y. Xia, "Urban road trafc speed estimation for missing probe vehicle data based on multiple linear regression model", in Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC), The Hague, The Netherlands, Oct. 2013, pp. 118-123.
[8] F. Bashir and H. L. Wei, "Parametric and non-parametric methods to enhance prediction performance in the presence of missing data," in Proc. 19th Int. Conf. Syst. Theory, Control Comput (ICSTCC), Cheile Gradistei, Romania, 2015, pp. 337-342.
[9] A. Karmaker and S. Kwek, "Incorporating an EM approach for handling missing attribute values in decision tree induction," In Proc. 5th Int. Conf. Hybrid Intell. Syst. (HIS), 2005, p. 6.
[10] D.-H. Yang, N. N. Li, H. Z. Wang, J.Z. Zhao, and H. Gao, "The optimization of the big data cleaning based on task merging," Chin. J. Comput., vol. 39, no. 1, pp. 97-108, 2016.
[11] M. Zhu and X. B. Cheng, "Iterative KNN imputation based on GRA for missing values in TPLMS", in Proc. 4th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT), Harbin, China, 2015, pp. 94-99.
[12] P. Keerin, W. Kurutach, and T. Boongoen, "Cluster-based KNN missing value imputation for DNA microarray data," in Proc. IEEE Int. Conf. Syst., Man, (SMC), Seoul, South Korea, Oct. 2012, pp. 445-450.
[13] L. Jin, H. Wang, S. Huang, and H. Gao, ``Missing value imputation in big data based-on map-reduce," J. Comput. Res. Develop., vol. 50, no. S1, pp. 312-321, 2013.
[14] M. G. Rahman and M. Z. Islam, "iDMI: A novel technique for missing value imputation using a decision tree and expectation-maximization algorithm," in Proc. 16th Int. Conf. Comput. Inf. Technol. (ICCIT), Khulna, Bangladesh, 2014, pp. 496-501.
[15] S. Wu, X.-D. Feng, and Z.-G. Shan, "Missing data imputation approach based on incomplete data clustering," Chin. J. Comput., vol. 35, no. 28, pp. 1726-1738, Aug. 2012.
[16] C. Liu, D. Dai, and H. Yan, “The theoretic framework of local weighted approximation for microarray missing value estimation,” Pattern Recognition, vol. 43, no. 8, pp. 2993–3002, 2010.
[17] M. G. Rahman and M. Z. Islam, “kdmi: A novel method for missing values imputation using two levels of horizontal partitioning in a data set,” in The 9th International Conference on Advanced Data Mining and Applications (ADMA 13). in press, Hangzhou, China: Springer, 2013.
[18] K. Cheng, N. Law, and W. Siu, “Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data,” Pattern Recognition, vol. 45, no. 4, pp. 1281–1289, 2012.
[19] Browning, S. R. Missing data imputation and haplotype phase inference for genome-wide association studies. Hum Genet. 124, 439–450, doi: 10.1007/s00439-008-0568-7 (2008).
[20] Xiaolong xu, weizhi chong, shancang li, abdullahi arabo and jianyu xiao. "MIAEC: Missing Data Imputation Based on the Evidence Chain", Special section on security and trusted computing for Industrial internet of things, IEEE, VOLUME 6, 2018