A Hybrid GP-KNN Imputation for Symbolic Regression with Missing Values

Baligh Al-Helali(✉), Qi Chen, Bing Xue, and Mengjie Zhang

School of Engineering and Computer Science, Victoria University of Wellington,
PO Box 600, Wellington 6400, New Zealand
{baligh.al-helali,Qi.Chen,Bing.Xue,Mengjie.Zhang}@ecs.vuw.ac.nz

Abstract. In data science, missingness is a serious challenge when dealing with real-world data sets. Although many imputation approaches have been proposed to tackle missing values in machine learning, most studies focus on the classification task rather than the regression task. To the best of our knowledge, no study has been conducted to investigate the use of imputation methods when performing symbolic regression on incomplete real-world data sets. In this work, we propose a new imputation method called GP-KNN which is a hybrid method employing two concepts: Genetic Programming Imputation (GPI) and K-Nearest Neighbour (KNN). GP-KNN considers both the feature and instance relevance. The experimental results show that the proposed method has a better performance comparing to state-of-the-art imputation methods in most of the considered cases with respect to both imputation accuracy and symbolic regression performance.

Keywords: Symbolic regression · Genetic programming
Incomplete data · Imputation

1 Introduction

Symbolic Regression (SR) is a crucial machine learning field the task of which is to construct a mathematical model that best fits a given data set. Different from traditional regression, no priori assumption is required in SR. This means many benefits to real-world applications, especially when dealing with multivariate data from unknown systems, such as real-time forecasting and physical model integration [1]. Genetic Programming (GP) is an evolutionary computation technique which is inspired by the biological evolution analogy. It creates new solutions from the current ones using mutation and crossover processes with the expectation to find a good solution in the evolution process. SR problems have been typically solved via GP [13].

Many real-world data sets have instances with missing values due to some common reasons such as unfilled survey fields and sensor failures. When analyzing the regression data sets in the UCI machine learning repository [6], among

T. Mitrovic et al. (Eds.): AI 2018, LNAI 11320, pp. 345–357, 2018.
https://doi.org/10.1007/978-3-030-03991-2_33

about the 80 available data sets, more than 20 data sets are annotated as having missing values.

There are three main types of missing data: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [8]. MCAR implies that the events that lead to any missing value happen independently of both unobservable parameters and observable variables of interest, i.e. no relationship presents between the missingness of values and other values, observed or missing. In MAR, the missingness is related to some observed data rather than to the missing data itself. MNAR means that the missingness is related to the reason it's missing (neither MAR nor MCAR).

Imputation is the process of filling missing values with plausible ones and it can be categorized into single imputation and multiple imputation [8]. Single imputation provides a specific value in place of the missing data directly. While multiple imputation selects such imputed value from several possible responses based on the variance/confidence interval analysis. Some methods are widely used for imputation. K-nearest neighbour (KNN) is used to impute the missing values with the average of the k most similar instances. Classification and regression trees (CART) is used for imputation by employing decision trees to predict the missing values based on the non-missing ones. Another method adopting the decision trees approach is random forest (RF). It starts from replacing the missing data with the average of the corresponding complete values and then iteratively improves the missing imputation using proximity. One of the most flexible and powerful imputation methods is multivariate imputation by chained equations (MICE). MICE is an iterative method based on chained equations that generates an imputation model for each feature and involves other features as predictors.

GP-based imputation has been investigated on the classification tasks and has shown better performance than some popular imputation methods. In [17], GP-based multiple imputation method is introduced. This method utilizes the robust SR method to predict the missing values in classification data sets. In [18], the GP-based imputation is separated into two stages: the training process and the imputation process. In the training process, imputation regression functions are constructed using chunks of training instances. The imputation process is performed on individual instances by applying the constructed predictors. In [19], multiple imputation and GP are combined to evolve classifiers on data with missing values. Common patterns of missing values are firstly extracted and GP is then used to construct a classifier for each pattern.

The existing research on dealing with missing values mainly focus on the classification tasks. The impact of missing values when performing traditional regression has been considered in several studies [11,12,14]. In SR research, the most common strategy to deal with incomplete data is to delete the instances having missing values [5,7,9]. The only studies that consider imputation for SR are [3,15]. However, they have some limitations. [3] considered only artificial functions, while in [15], missing values are simply replaced with corresponding

feature values from other instances. Therefore, how to deal with missing values in SR is still an open issue.

In this work, we aim to develop a new imputation method to handle missing values for SR. This implies conducting SR research with incomplete data. Specific objectives include:

1. developing a new hybrid imputation method to utilize two existing approaches: KNN and GPI;
2. investigating whether the proposed method can outperform state-of-the-art imputation methods on obtaining a small imputation error; and
3. investigating whether the proposed method can outperform state-of-the-art imputation methods on achieving a good regression performance.

2 The Proposed Method

In this section, a new imputation method is proposed. An overall structure of the imputation treatment is firstly introduced and the proposed method is then presented and described.

2.1 The Overall Structure and Evaluation Measures

The framework of imputation for incomplete data is shown in Fig. 1. The first step is to divide a data set into the training and test sets by ratios (70:30). After that the imputation method is performed on the incomplete training and test data sets independently and the imputed complete sets are then fed into the evaluation process. Usually, two measures are used for evaluating the performance of the imputation methods: the imputation error and the regression performance.

For measuring the imputation error, complete regression data sets are used to produce incomplete data sets by generating different percentages of missing values. These synthetic incomplete data sets are then imputed and the imputation error is measured by the difference between the original complete data sets and the imputed ones. In this work, the relative squared error (RSE) shown in the following equation, is used to measure the imputation error:

$$RSE = \frac{\sum_{i=1}^{n}(y_i - t_i)^2}{\sum_{i=1}^{n}(t_i - \bar{t})^2} \tag{1}$$

where n is number of instances, y_i is the i^{th} predicted value, t_i is the i^{th} desired value, and $\bar{t}$ is the average of the desired values t_i, $i = 1, 2, 3 \ldots, n$.

In addition to the synthetic incomplete data sets, real-world regression data sets with missing values are also used in the experiments, where the regression performance is used in both cases to evaluate the imputation methods. The imputed complete training data sets are fed to GP-based SR to build the regression model and the obtained model is evaluated on the unseen test data sets. RSE (Eq. 1) is used as the fitness function.

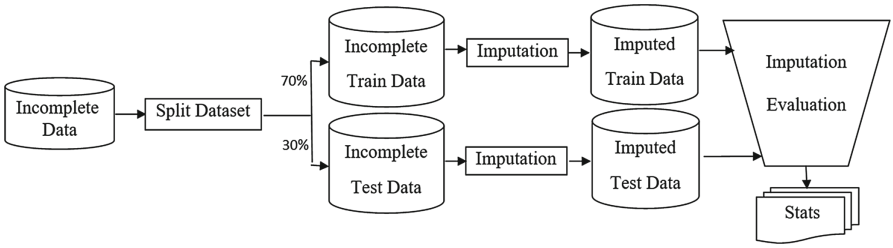

Fig. 1. Incomplete data sets treatment using imputation methods.

2.2 The Proposed GP-KNN Imputation

GPI methods are adopted for classification with missing values in [17–19]. The main idea is to consider each feature having missing values as the target variable while using other features as predictive variables. Instances with complete corresponding feature values are used to build the regression functions and these functions are then used to predict the missing values. This method has the advantage of not requiring any presumptions. However, it performs the regression on all instances regardless of the potential variation. Such variation might due to the imbalanced distribution of certain features. GPI might use some instances that are irrelevant to the instance to be imputed. On the other hand, KNN imputation replaces the missing value with the weighted average (weighted based on distance) of the 'k' closest instances [2]. Although KNN clearly takes the instance-based relevance into account, it ignores the feature-based relevance.

To overcome the limitations of GPI and KNN imputation by taking both instance-based and feature-based relevance into consideration, this work proposes a new imputation method named GP-KNN. The proposed method is formed by combining the two methods GPI and KNN to handle the missing values. The main idea is that, instead of using all instances to build the SR regression functions for features having missing values, only k nearest instances are used to build such predictors for the missing values. This modification is to get the benefits of both GPI and KNN. It firstly explores the instance-based similarity to extract k closest instances and then employs the feature-based predictability power of GPI to impute the missing value. This method considers the variance in each feature to be imputed. More specifically, one feature might require more than one imputation regression function according to the distances between the corresponding instances.

Without loss of generality, the main steps of the proposed method assuming that the input data $\mathcal{X}$ has a missing value at the position i, j (the i^{th} instance and the j^{th} feature) are described as follows:

1. From the i^{th} instance, extract the non-missing values to form a complete instance $V_{i,j}$.
2. Obtain a sub data set $\mathcal{X}^T_{i,j}$ by excluding the features that are not included in $V_{i,j}$.

3. Form a complete sub data set $\mathcal{X}_{i,j}$ by removing instances having missing values from $\mathcal{X}^T_{i,j}$.
4. Get k nearest instances to $V_{i,j}$ from $\mathcal{X}_{i,j}$ and put them in a new matrix $\mathcal{X}^k_{i,j}$.
5. Build an SR function $f_{i,j}$ using $\mathcal{X}^k_{i,j}$ considering the corresponding j^{th} feature of $\mathcal{X}$ as a target variable.
6. Impute the missing value $\mathcal{X}[i,j]$ using the prediction value obtained by applying the constructed function $f_{i,j}$ on the i^{th} instance $V_{i,j}$, i.e. $\mathcal{X}^C[i,j] = f_{i,j}(V_{i,j})$, where $\mathcal{X}^C$ is the output compete imputed data.

An illustrative example for the main steps of the proposed method is presented (see supplementary Example S1 that can be found online at: http://ecs.victoria.ac.nz/foswiki/pub/Groups/ECRG/OnlineSupplimentaryMaterials/GPKNN_AI2018.pdf).

The above is a high level introduction of the proposed method. The psoduecode of the proposed method is shown in Algorithm 1 and more detailed description of this procedure is given below.

Algorithm 1. Modified GP-KNN Imputation

Input : Data set $\mathcal{X}$ with missing values
Output: Complete data set $\mathcal{X}^C$

Let $F = \phi$ $R = \phi$ $D = \phi$, where F: regression functions set, R: instance-based references set, D: the corresponding distance thresholds set;
foreach *missing value* $\mathcal{X}[i,j]$ **do**
  From the i^{th} instance, extract the non-missing values to form a complete instance $V_{i,j}$;
  if $\exists V_{\hat{i},j} \in R$ *s.t* $distance(V_{\hat{i},j}, V_{i,j}) \leq d_{\hat{i},j}$ **then**
    $\mathcal{X}^C[i,j] \leftarrow f_{\hat{i},j}(V_{i,j})$
  else
    Extract a sub data set $\mathcal{X}^T_{i,j}$ by excluding the features that are not included in $V_{i,j}$;
    Obtain $\mathcal{X}_{i,j}$ as the non-missing sub matrix of $\mathcal{X}^T_{i,j}$;
    $k \leftarrow min(max(|J_{i,j}|, |I_{i,j}|/3), |I_{i,j}|)$, where $I_{i,j}$ and $J_{i,j}$ are the instance and feature indexes of $\mathcal{X}_{i,j}$;
    $\mathcal{X}^k_{i,j} \leftarrow KNN(\mathcal{X}_{i,j}, V_{i,j}, k)$;
    $d_{i,j} \leftarrow max(distance(V, V_{i,j}))$, $\forall$ $V_{i,j}$ an instance in $\mathcal{X}^k_{i,j}$;
    for $r = 1$ *to* N **do**
      $f_r \leftarrow \mathrm{SR}(\mathcal{X}^k_{i,j}, \mathcal{X}[I^k_{i,j}, j])$, where $I^k_{i,j}$ is the instance indexes of $\mathcal{X}^k_{i,j}$;
    end
    $\hat{r} \leftarrow arg(\min_{r=1,\ldots N} f_r)$;
    $f_{i,j} \leftarrow f_{\hat{r}}$;
    $\mathcal{X}^C[i,j] \leftarrow f_{i,j}(V_{i,j})$;
    Append $f_{i,j}, d_{i,j}, V_{i,j}$ to F, D, R, respectively
  end
end

Step 1. Initialize empty sets F, R, and D. These sets are used to store the necessary parameters during the imputation process. The set F contains the constructed imputation SR regression functions. R is a reference set formed by extracting complete samples from the instances having missing values. D is a set of distance thresholds representing the neighborhood diameter of the processed missing values.

Step 2. Extract the non-missing values from the i^{th} instance forming a complete instance $V_{i,j}$.

Step 3. Compare $V_{i,j}$ with the existing imputed instances, if there is already a similar one $V_{\hat{i},j}$ then use the corresponding stored imputation function $f_{\hat{i},j}$ to impute $\mathcal{X}[i,j]$ directly, i.e. $\mathcal{X}^C[i,j] = f_{\hat{i},j}(V_{i,j})$. The similarity is measured by the Euclidean distance.

Step 4. Obtain a sub data set $\mathcal{X}^T_{i,j}$ by excluding features having missing values at the i^{th} instance and instances having missing values at the j^{th} feature. After that, delete incomplete instances forming a complete sub data set $\mathcal{X}_{i,j}$.

Step 5. Instead of using all instances in $\mathcal{X}_{i,j}$ to build the regression function as in GPI, the KNN method is employed to extract $\mathcal{X}^k_{i,j}$ which contains the k nearest instances of $\mathcal{X}_{i,j}$ to $V_{i,j}$. For the selection of k, the lower bound is set to the number of features in $V_{i,j}$ ($|J_{i,j}|$) to avoid the curse of dimensionality problem. The upper bound is chosen empirically as one-third of the number of the instances ($|I_{i,j}|/3$). However, if these constraints can not be satisfied, i.e. small complete sub set, the whole set is used ($|I_{i,j}|$). k is selected by the following equation.

$$k = min(max(|J_{i,j}|, |I_{i,j}|/3), |I_{i,j}|) \tag{2}$$

Step 6. The sub-data set $\mathcal{X}^k_{i,j}$ is then used to build N regression functions $\{f_r\}_{r=1}^N$ via SR where the j^{th} feature is the target variable. The value of N is set to 10 empirically.

Step 7. The best constructed SR function (the one having the least fitness value), $f_{i,j}$, is used to predict (impute) the value of $\mathcal{X}[i,j]$ and put it in $\mathcal{X}^C[i,j]$, i.e. $\mathcal{X}^C[i,j] = f_{i,j}(V_{i,j})$. The fitness function is computed using Eq. 1.

Step 8. To avoid the time consuming process of performing GP-KNN imputation for each missing value, the maximum distance $d_{i,j}$ of the returned k nearest instances is computed and stored. This distance can be seen as the diameter of this set of samples w.r.t $V_{i,j}$. It is used to compare the new missing values with the previously imputed ones to check whether the already stored functions can be used directly.

3 Experimental Setup

A set of the experiments has been conducted to evaluate the performance of the proposed imputation method and compare it with state-of-the-art imputation methods, i.e. MICE, KNN, CART, RF, and GPI, using two measures: the imputation error and the SR performance.

As mentioned above, the first evaluation approach requires the complete data sets as ground truth. Table 1 shows the statistics of the complete data sets used in this work. The instances having missing target values are deleted and some non-numerical features are ignored. For each data set, 30 data sets of five instance MAR missingness probabilities (10, 20%, 30%, 40%, 50%) are generated on 40% of the features, i.e. 150 incomplete data sets are obtained. The imputation and missing imposing methods are implemented using R packages: mice [4] and simsem [16] with the default settings. After applying the imputation method, the statistics are aggregated to evaluate the performance. However, to validate the proposed method on reality, real-world data sets with different probabilities of missing values are used. The information of these data sets are shown in Table 2. More details on the used data sets can be found in the UCI repository [6].

Table 3 shows the parameters for the GP runs that used for both imputation (GPI) and regression (SR). They are common settings in GP research. For each experiment, 30 independent GP runs are performed and the implementation is carried out under the GP framework provided by distributed evolutionary algorithms in python (DEAP) [10].

Table 1. Statistics of the used complete data sets

Data set	#Features	#Instances
Yacht-hydrodynamics	7	308
Forestfires	13	517
ENB2012	8	768
Concrete	9	1030
Airfoil-self-noise	6	1503

Table 2. Statistics of the used incomplete data sets

Data set	#Features	#Instances	#Instances with missing	% Missing
SkillCraft1	19	3395	57	1.68
Imports-85	15	205	54	26.34
Auto-mpg	7	398	6	1.58
CCN	122	1994	1676	84.05

4 Results and Analysis

This section shows the experimental results of the proposed GP-KNN imputation method, CART, KNN, MICE, RF, and GPI. The comparisons are carried out in terms of both the imputation error and the regression performance. The

Table 3. The used values for GP parameters

Parameter	Value
Generations	100
Population size	512
Crossover rate	0.9
Mutation rate	0.1
Elitism	5
Selection method	Tournament
Tournament size	7
Maximum depth	17
Initialization	Ramped-half and half
Function set	$+$, $-$, *, protected %
Terminal set	features and constants $\in (-1, 1)$

Wilcoxon non-parametric statistical significance test with a significance level of 0.05 has been used to compare the imputation methods with the proposed method. The means of RSEs achieved by the best-of-run GP programs on the imputed test sets using the examined imputation methods are shown.

4.1 Imputation Performance

The imputation performance with different missingness probabilities are shown in Fig. 2(a). It can be seen that the proposed method has the best performance among the examined methods on four of the five data sets with respect to almost all considered missingness probabilities. The differences are all significant on the data sets Yacht, Concrete, and Airfoil. On the Forestfires data set, CART achieves a similar imputation performance to GP-KNN. However, CART and MICE have smaller imputation errors than other imputation methods on ENB2012.

One of the most important advantages of GP-KNN is that it mostly performs well even if one of the two underlying methods, i.e. GPI and KNN, has an undesirable performance. This is indicated by the results on Yacht, Forestfires, and Concrete. On these data sets, using KNN results in the worst imputation while GP-KNN has the best performance. On airofoil data set, the good performance of GPI along with acceptable performance of KNN leads to a highly preferred GP-KNN performance. However, the extremely low performance of KNN on ENB2012 data set seems to affect the overall performance of GP-KNN negatively. On this data set, GPI advances KNN significantly which indicates that the correlation between features might be higher than that between instances. However, it is difficult for GP-KNN to advance GPI notably in this case.

Considering the comparison between the other imputation methods, the GPI method performs better than the rest and the CART method comes next. And

KNN method has the worst imputation performance on the five data sets. A common pattern among all the imputation methods is that the higher probability of the missingness the worse imputation error as there will be less useful data to predict the missing values properly.

4.2 SR Performance on Synthetic Incompleteness

The symbolic regression errors on these synthetic incomplete data sets are shown in Fig. 2(b).

Similar to the pattern on the imputation evaluation, the proposed GP-KNN method achieves the best performance except for the ENB2012 data set. However, the agreement between the imputation performance and the regression performance is not as high as expected. Such agreement can be seen in the Concrete data set with the corresponding results. However, on the Forestfires data set, although CART achieves a similar imputation performance comparing to GP-KNN, the best regression results are obtained by GP-KNN. This is an indicator of the applicability of the proposed method when performing SR.

Unlike the corresponding imputation performance results, the regression errors' curves are not monotonically increasing w.r.t the missingness probabilities. The functionality can be noticed when comparing the mean error obtained on 10% missingness and that on 20% missingness on Airfoil data set. The reason is that the regression models are trained on the imputed data and the regression errors are evaluated on imputed data as well which means the error depends on the modeling process regardless of the missingness itself.

4.3 SR Performance on Real-World Incompleteness

To validate the applicability of the proposed method, real-world incomplete data sets are considered. In this section, four real-world data sets having different ratios of missing values are examined.

As the data sets are incomplete, it is impossible to measure the imputation error. Hence, the regression performance will be the only criterion to compare the imputation methods. The SR performance results on the imputed test data sets are shown in Table 4. The mean, standard deviation and the significant test sign of RSEs achieved by the best-of-run programs on test sets are shown. ST refers to the results of the significance test (Wilcox) against the proposed GP-KNN method where "+" means GP-KNN is significantly better, "−" means GP-KNN is significantly worse, and "=" indicates no significant difference.

GP-KNN achieves the best regression performance on Imports-85, Skill, and CCN while on Auto-mpg, the best results are obtained by the CART method. This may be due to the low percentage of missing values in the Auto-mpg data set. The GP-based imputation methods are not the worst in any of the used data sets. The worst reported results are obtained when using KNN on Auto data, RF on Imports-85 data, and MICE on both Skill and CNN data sets.

The main limitation of the proposed method is the imputation time complexity. This problem is due to the need to go through all missing values. It is also

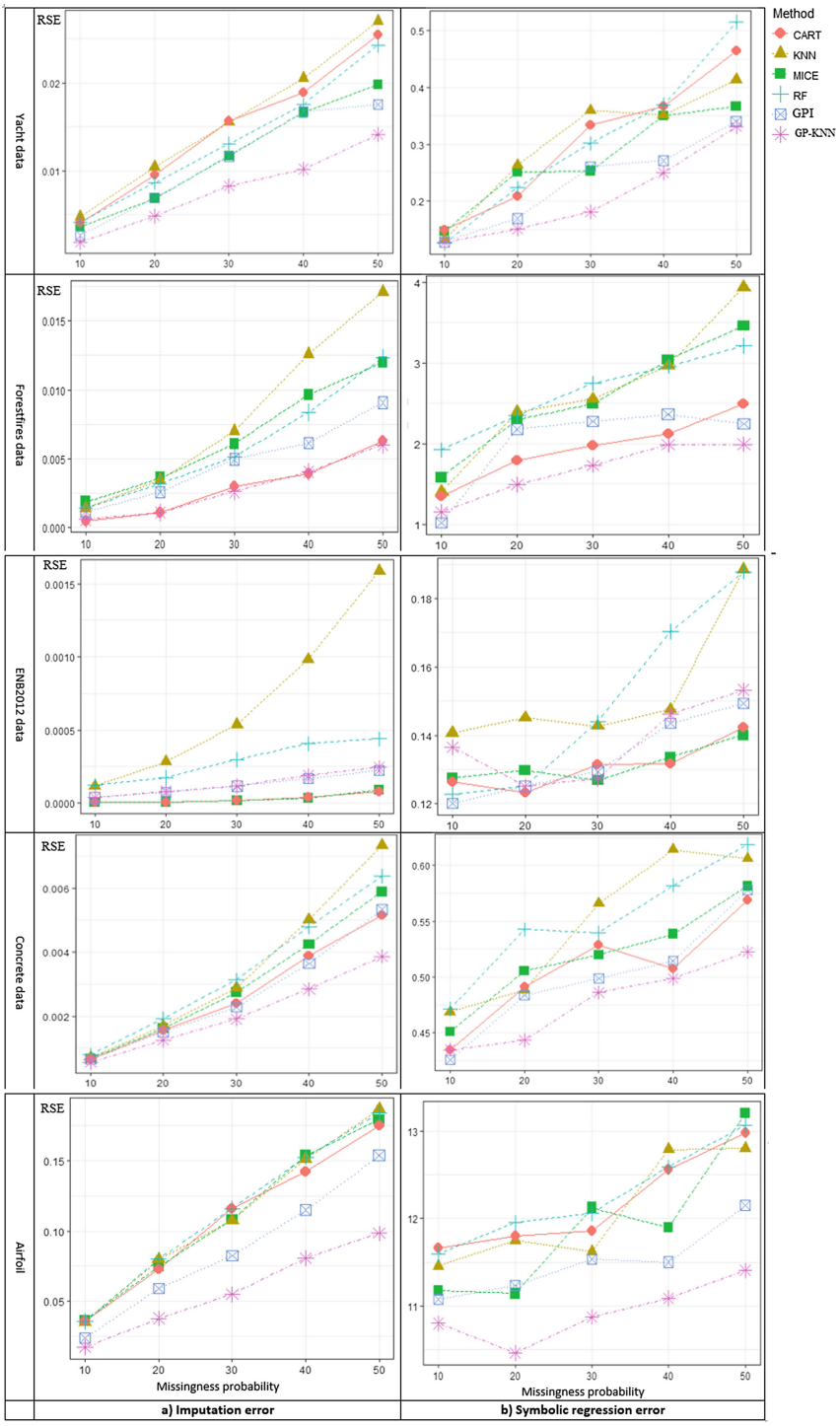

Fig. 2. The experimental results on synthetic incomplete data sets, where the x-axis represents the missingness probability and the y-axis is the RSE error.

required to apply the KNN method and then construct the regression function using SR which means an extra load.

Table 4. The test results of SR error on real incomplete data sets with different imputation methods.

Method	Measure	Auto	Imports-85	Skill	CCN
MICE	Mean	0.248889	0.346285	0.646367	0.546712
	Std	0.05624	0.038141	0.024561	0.056658
	ST	=	+	+	+
KNN	Mean	0.276211	0.33792	0.635711	0.539296
	Std	0.071383	0.042548	0.032341	0.049915
	ST	+	+	+	+
CART	Mean	**0.240706**	0.335545	0.640299	0.509417
	Std	0.046465	0.045827	0.026141	0.038756
	ST	−	+	+	=
RF	Mean	0.24339	0.373118	0.649879	0.517652
	Std	0.058325	0.040726	0.028141	0.028572
	ST	=	+	+	+
GPI	Mean	0.242211	0.331869	0.635089	0.533547
	Std	0.036748	0.031374	0.28758	0.0458
	ST	=	+	+	+
GP-KNN	Mean	0.24411	**0.327196**	**0.633138**	**0.504164**
	Std	0.042643	0.0303757	0.02848	0.033821

5 Conclusions and Future Directions

This work proposed a new genetic programming-based imputation method which combines KNN and GPI. The performance of this method is evaluated from two aspects: the imputation error and the symbolic regression performance. The proposed method has been compared with state-of-the-art imputation methods. The experimental results show that the proposed GP-KNN method significantly outperforms the other methods in most considered cases.

For future work, more experimental work should be done to investigate the impact of generating incomplete data sets with more ratios and different missingness kinds. Moreover, various data sets from different applications should be used. The use of the imputation methods can be then studied and analyzed with more statistical evidences. Another plan is to deal with the incompleteness issue in big data such as data sets with high dimensional features. However, this should be done along with handling the problem of time-complexity which represents the main limitation of the proposed method.

References

1. Austel, V., et al.: Globally optimal symbolic regression. arXiv preprint arXiv:1710.10720 (2017)
2. Beretta, L., Santaniello, A.: Nearest neighbor imputation algorithms: a critical evaluation. BMC Med. Inform. Decis. Mak. **16**(3), 74 (2016). https://doi.org/10.1186/s12911-016-0318-z
3. Brandejsky, T.: Model identification from incomplete data set describing state variable subset only - the problem of optimizing and predicting heuristic incorporation into evolutionary system. In: Zelinka, I., Chen, G., Rössler, O., Snasel, V., Abraham, A. (eds.) Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. AISC, vol. 210, pp. 181–189. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-319-00542-3_19
4. van Buuren, S., Groothuis-Oudshoorn, K.: mice: multivariate imputation by chained equations in R. J. Stat. Softw. **45**, 1–68 (2010)
5. Chen, Q., Zhang, M., Xue, B.: Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. IEEE Trans. Evol. Comput. **21**(5), 792–806 (2017). https://doi.org/10.1109/TEVC.2017.2683489
6. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
7. Dick, G.: Bloat and generalisation in symbolic regression. In: Dick, G., et al. (eds.) SEAL 2014. LNCS, vol. 8886, pp. 491–502. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13563-2_42
8. Donders, A.R.T., van der Heijden, G.J., Stijnen, T., Moons, K.G.: Review: a gentle introduction to imputation of missing values. J. Clin. Epidemiol. **59**(10), 1087–1091 (2006)
9. Eggermont, J., et al.: Data mining using genetic programming: classification and symbolic regression. Institute for Programming research and Algorithmics, Leiden Institute of Advanced Computer Science, Faculty of Mathematics & Natural Sciences, Leiden University (2005)
10. Fortin, F.A., Rainville, F.M.D., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. **13**(Jul), 2171–2175 (2012)
11. Haitovsky, Y.: Missing data in regression analysis. J. R. Stat. Soc. Ser. B (Methodol.) **30**, 67–82 (1968)
12. Horton, N.J., Kleinman, K.P.: Much ado about nothing: a comparison of missing data methods and software to fit incomplete data regression models. Am. Stat. **61**(1), 79–90 (2007)
13. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Stat. Comput. **4**(2), 87–112 (1994)
14. Loh, P.L., Wainwright, M.J.: High-dimensional regression with noisy and missing data: provable guarantees with non-convexity. In: Advances in Neural Information Processing Systems, pp. 2726–2734 (2011)
15. Pennachin, C., Looks, M., de Vasconcelos, J.: Improved time series prediction and symbolic regression with affine arithmetic. In: Riolo, R., Vladislavleva, E., Moore, J. (eds.) Genetic Programming Theory and Practice IX. GEVO, pp. 97–112. Springer, New York (2011). https://doi.org/10.1007/978-1-4614-1770-5_6
16. Pornprasertmanit, S., Miller, P., Schoemann, A., Quick, C., Jorgensen, T., Pornprasertmanit, M.S.: Package 'simsem' (2016)
17. Tran, C.T., Zhang, M., Andreae, P.: Multiple imputation for missing data using genetic programming. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 583–590. ACM (2015)

18. Tran, C.T., Zhang, M., Andreae, P.: A genetic programming-based imputation method for classification with missing data. In: Heywood, M.I., McDermott, J., Castelli, M., Costa, E., Sim, K. (eds.) EuroGP 2016. LNCS, vol. 9594, pp. 149–163. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30668-1_10
19. Tran, C.T., Zhang, M., Andreae, P., Xue, B.: Multiple imputation and genetic programming for classification with incomplete data. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 521–528. ACM (2017)