Genetic Versus Nearest-Neighbor Imputation of Missing Attribute Values for RBF Networks

Pedro G. de Oliveira and André L.V. Coelho

Graduate Program in Applied Informatics, University of Fortaleza,
Av. Washington Soares 1321/J30, Fortaleza–CE, Brazil
pedgoncalves@gmail.com, acoelho@unifor.br

Abstract. Missing data is a common issue in almost every real-world dataset. In this work, we investigate the relative merits of applying two imputation schemes for coping with this problem while designing radial basis function network classifiers, which show sensitiveness to the existence of missing values. Whereas the first scheme centers upon the *k*-nearest neighbor algorithm and has been deployed with success in other supervised/unsupervised learning contexts, the second is based on a simple genetic algorithm model and has not been fully explored so far.

1 Introduction

Real-life datasets invariably contain examples in which the values of some attributes, for some reason or other, are unknown. Aiming at managing this problem, several preprocessing techniques have been devised so far, ranging from those more generic/simple to those more customized/complicated [5]. A broad class of techniques, referred to as imputation methods, involves the replacement of absent values with plausible ones, usually estimated from the data itself. In this context, a common practice is to substitute missing values (MV) with statistical amounts, such as the most likely values, for discrete attributes, or the mean (median) values, for the continuous case.

Since different learning algorithms show different rates of sensitiveness to different levels of data incompleteness, several comparative studies have been conducted recently over different MV handling methods (with an emphasis on those based on imputation) in order to reveal their advantages and disadvantages [1,3,5,9]. In this paper, we move a step further in this initiative and investigate the relative merits of applying two imputation schemes for coping with missing data while specifically designing radial basis function (RBF) networks [6,7]. Whereas the first method centers upon the *k*-nearest neighbor (KNN) algorithm and has been deployed with success in other supervised/unsupervised learning contexts [3,8], the second is based on a simple genetic algorithm (GA) model [4] and has not been fully explored so far.

In order to contrast, in terms of effectiveness and efficiency, the pros & cons of the genetic and KNN imputation methods for the training and generalization of RBF network classifiers, experiments over UCI benchmark datasets [2] have been conducted, the quantitative results of which are discussed here. In this analysis, we have considered different types of atributes as well as MV rates, and also taken as reference the performance exhibited by mean/mode imputation.

The rest of the paper is organized as follows. Section 2 describes *ein passant* how RBF networks work and comments upon their sensitiveness to MV. The next two sections briefly outline the distinctive aspects behind the nearest-neighbor and genetic imputation methods. Section 5 is dedicated to the analysis of the performance levels exhibited by the contestant imputation schemes, whereas Section 6 concludes the paper and brings remarks on future work.

2 RBF Networks

RBF networks are a popular type of three-layer feedforward networks [6,7]. Each unit i of the hidden layer of this type of network essentially represents a particular point $\mathbf{c}_i$ in the input space, and its output for a given instance $\mathbf{x}$ depends solely on the distance between $\mathbf{c}_i$ and $\mathbf{x}$ (which is achieved by using nonlinear activation functions to convert the distance into a similarity measure). On the other hand, the role of the output layer is to linearly combine the outputs produced at the hidden layer and bring about the actual network's estimates.

An advantage of RBF networks over multilayer perceptrons is that the training of each layer can be conducted separately, bringing efficiency. A disadvantage is that they give every attribute the same weight, meaning that they may show high sensitiveness to the way MV are handled. Although relevant, this issue seems to be overlooked in the literature.

3 Nearest-Neighbor Imputation of Missing Values

This method uses the KNN algorithm [10] to estimate missing data, by considering each time the attribute with MV as class attribute and the others as sources of information for locating the nearest samples. Some advantages of this method include [3]: a) it can estimate both discrete and numeric attribute values; b) it is not necessary to build up a predictive model for each attribute with missing data, as KNN is a lazy-learning method; and c) it can easily treat examples with multiple MV. On ther other hand, efficiency may be a big trouble for this method when considering large datasets if special-purpose data structures (as kD-trees) are not used [10]. Likewise, how to select the value k and the measure of similarity is a decision that may impact the performance results greatly [9].

4 Genetic Imputation of Missing Values

The second scheme employs a customized GA as a mechanism for imputation, which in turn is cast as an optimization problem. So, the values of missing data may be automatically tailored in accord with the supervised learning algorithm being used, leveraging up its performance. Although appealing, to the best of our knowledge, no work has yet fully explored the idea of applying a GA engine to deal specifically with the MV issue. According to the model adopted here, the representation of the individuals is linear, so that each MV is assigned to a position in the string. The encoding can be homogeneous or not, depending on the types of attributes with MV (continuous or discrete). Moreover, fitness-proportionate selection, two-point crossover, and simple mutation are employed for generating novel solutions. Finally, we have made use of the wrapper approach commonly used in the attribute selection task [4] for assigning fitness values. So, these values are set as the average error rate achieved by the RBF network in a stratified 10-fold cross-validation process conducted over the enhanced dataset.

5 Experiments and Results

To assess the performance of the genetic and KNN imputation methods with respect to the training and generalization of RBF network classifiers, extensive experiments have been conducted over UCI benchmark datasets [2].

Due to space limitation, we focus our analysis here on the six datasets depicted in Table 1, which show different number of samples, class distributions, and number/types of attributes and are roughly organized from the less to the more complex ones. These datasets originally do not show absent items, except given to the fourth and sixth in the list. But even in these cases (where we have maintained the MV items), we have followed the same strategy adopted in other papers [3,9], viz., to artificially inject MV completely at random – by this means, the probability of missing data on any attribute would not depend on any value of that attribute.

In the experiments, we have considered three rates of data missingness: 5%, 10%, and 20%. For every original dataset, 10 pairs of training and test (80/20%) partitions were produced via a stratified manner. Then, MV were injected in each novel dataset; this process was repeated five times, giving birth to 50 derived datasets for each original one. In order to assess the performance of the methods, we have recorded the average error rates produced by RBF networks both over the training (10-fold cross-validation) and test datasets. This division is necessary as the genetic imputation uses the cross-validation results as fitness values for its individuals whereas KNN and mean/mode schemes do not.

Table 1. Configuration of the datasets used in the experiments. **NC**: number of classes; **NS**: number of samples; **CD**: class distribution; **NA**: number of attributes; and **TA**: type(s) of attributes (**D**iscrete $\times$ **C**ontinuous).

Name	NC	NS	CD	NA	TA
IRIS	3	150	(33.3%;33.3%;33.3%)	4	C
ZOO	7	101	(40.6%;19.8%;4.9%;12.9%;4.00%;7.9%;9.9%)	16	D&C
DIABETES	2	768	(65.1%;34.9%)	8	C
BCD699	2	699	(65.5%;34.5%)	9	C
CAR	4	1,728	(70.0%;22.2%;4.0%;3.8%)	6	D
DERMATOLOGY	6	366	(30.6%;16.7%;19.7%;13.4%;14.2%;5.5%)	33	D&C

Table 2. Average results for 5%-level of missing values

Method	Crossval	Test	Generation	Time
	IRIS			
Mean/Mode	92.78 ± 1.00	97.11 ± 3.42	–	0.00 ± 0.10
KNN ($k = 1$)	93.83 ± 1.20	97.33 ± 2.67	–	0.00 ± 0.00
KNN ($k = 3$)	93.94 ± 0.92	97.11 ± 2.69	–	0.00 ± 0.00
KNN ($k = 5$)	94.17 ± 1.17	96.89 ± 3.01	–	0.00 ± 0.00
GA	$\mathbf{96.67 \pm 0.60}$	$\mathbf{98.28 \pm 1.34}$	5.07 ± 1.01	96.73 ± 4.40
	ZOO			
Mean/Mode	93.17 ± 1.64	$\mathbf{95.67 \pm 4.04}$	–	0.00 ± 0.00
KNN ($k = 1$)	95.39 ± 1.22	94.67 ± 3.51	–	0.07 ± 0.12
KNN ($k = 3$)	94.90 ± 2.40	95.33 ± 4.51	–	0.07 ± 0.12
KNN ($k = 5$)	94.57 ± 1.73	95.00 ± 5.00	–	0.13 ± 0.12
GA	$\mathbf{98.19 \pm 0.71}$	94.29 ± 2.86	3.53 ± 1.42	125.67 ± 9.03
	DIABETES			
Mean/Mode	73.39 ± 0.71	77.82 ± 0.54	–	1.00 ± 0.00
KNN ($k = 1$)	73.03 ± 0.15	$\mathbf{78.87 \pm 2.16}$	–	1.00 ± 0.00
KNN ($k = 3$)	72.60 ± 0.07	78.47 ± 1.90	–	1.00 ± 0.00
KNN ($k = 5$)	72.70 ± 0.28	78.34 ± 2.41	–	1.00 ± 0.00
GA	$\mathbf{75.63 \pm 0.92}$	78.36 ± 1.04	8.20 ± 1.06	181.27 ± 4.05
	BCD699			
Mean/Mode	96.12 ± 0.31	$\mathbf{95.92 \pm 1.22}$	–	1.13 ± 0.23
KNN ($k = 1$)	96.28 ± 0.40	95.39 ± 0.63	–	1.07 ± 0.12
KNN ($k = 3$)	96.20 ± 0.29	95.63 ± 0.98	–	1.00 ± 0.00
KNN ($k = 5$)	96.19 ± 0.42	95.34 ± 0.84	–	1.00 ± 0.00
GA	$\mathbf{97.17 \pm 0.41}$	95.38 ± 1.15	7.67 ± 0.42	192.13 ± 10.64
	CAR			
Mean/Mode	83.64 ± 0.41	87.38 ± 0.79	–	6.20 ± 0.20
KNN ($k = 1$)	82.69 ± 0.66	86.67 ± 0.99	–	5.00 ± 0.00
KNN ($k = 3$)	83.10 ± 0.53	86.42 ± 0.72	–	5.07 ± 0.12
KNN ($k = 5$)	83.17 ± 0.90	86.77 ± 1.16	–	5.13 ± 0.12
GA	$\mathbf{87.47 \pm 0.44}$	$\mathbf{87.51 \pm 0.29}$	8.80 ± 0.60	898.40 ± 79.15
	DERMATOLOGY			
Mean/Mode	95.81 ± 0.72	95.43 ± 1.51	–	4.53 ± 1.42
KNN ($k = 1$)	95.56 ± 0.83	94.89 ± 1.51	–	5.80 ± 1.83
KNN ($k = 3$)	95.84 ± 0.58	95.43 ± 1.98	–	5.20 ± 1.22
KNN ($k = 5$)	95.72 ± 0.74	94.79 ± 1.71	–	5.53 ± 1.63
GA	$\mathbf{97.93 \pm 0.45}$	$\mathbf{95.77 \pm 1.49}$	6.33 ± 2.55	596.27 ± 183.15

For the experiments, we have made use of the RBF network model as well as the validation testbench as implemented in the Weka framework [10]. The machine used was an Intel Pentium Core 2, 1.66 GHz equipped with 2 GB of RAM memory. Concerning the values of the control parameters adopted for the GA, they were calibrated as follows: a population of 20 individuals; 10 generations for each execution; 70% as crossover rate; and 10% as mutation rate. This configuration was adopted taking into account the efficiency issue: The GA was

Table 3. Average results for 10%-level of missing values

Method	Crossval	Test	Generation	Time
	IRIS			
Mean/Mode	93.00 ± 1.09	96.44 ± 3.85	–	0.00 ± 0.00
KNN ($k = 1$)	93.50 ± 0.44	96.45 ± 3.42	–	0.00 ± 0.00
KNN ($k = 3$)	93.50 ± 0.50	96.45 ± 3.42	–	0.00 ± 0.00
KNN ($k = 5$)	93.28 ± 0.25	96.22 ± 3.36	–	0.00 ± 0.00
GA	$\mathbf{95.11 \pm 0.54}$	$\mathbf{98.71 \pm 2.24}$	6.47 ± 0.23	123.53 ± 1.62
	ZOO			
Mean/Mode	91.52 ± 1.98	$\mathbf{95.67 \pm 4.04}$	–	0.20 ± 0.20
KNN ($k = 1$)	93.17 ± 1.85	95.33 ± 2.52	–	0.20 ± 0.35
KNN ($k = 3$)	92.59 ± 2.43	93.67 ± 3.06	–	0.47 ± 0.23
KNN ($k = 5$)	93.00 ± 2.95	95.00 ± 2.65	–	0.53 ± 0.12
GA	$\mathbf{97.61 \pm 0.38}$	95.24 ± 2.86	4.13 ± 1.63	139.07 ± 4.66
	DIABETES			
Mean/Mode	72.48 ± 0.85	77.25 ± 2.35	–	1.00 ± 0.00
KNN ($k = 1$)	72.24 ± 0.13	$\mathbf{78.17 \pm 1.93}$	–	1.00 ± 0.00
KNN ($k = 3$)	72.25 ± 0.62	77.91 ± 1.71	–	1.00 ± 0.00
KNN ($k = 5$)	72.47 ± 0.75	77.91 ± 1.37	–	1.00 ± 0.00
GA	$\mathbf{74.44 \pm 0.34}$	76.71 ± 1.04	8.53 ± 1.85	192.67 ± 1.86
	BCD699			
Mean/Mode	96.08 ± 0.26	$\mathbf{95.73 \pm 0.79}$	–	1.00 ± 0.00
KNN ($k = 1$)	95.90 ± 0.32	95.63 ± 1.34	–	1.00 ± 0.00
KNN ($k = 3$)	96.21 ± 0.16	95.49 ± 1.22	–	1.00 ± 0.00
KNN ($k = 5$)	96.07 ± 0.39	95.54 ± 1.04	–	1.00 ± 0.00
GA	$\mathbf{96.88 \pm 0.32}$	95.05 ± 1.15	7.20 ± 1.11	199.40 ± 2.95
	CAR			
Mean/Mode	80.11 ± 0.31	85.45 ± 0.29	–	7.40 ± 0.35
KNN ($k = 1$)	78.88 ± 1.03	85.08 ± 0.66	–	5.13 ± 0.23
KNN ($k = 3$)	79.16 ± 0.54	85.41 ± 0.49	–	5.07 ± 0.12
KNN ($k = 5$)	79.43 ± 0.22	$\mathbf{85.94 \pm 0.85}$	–	5.20 ± 0.20
GA	$\mathbf{84.01 \pm 0.39}$	85.51 ± 0.58	7.87 ± 0.76	872.00 ± 28.93
	DERMATOLOGY			
Mean/Mode	94.58 ± 1.40	$\mathbf{95.25 \pm 0.84}$	–	7.07 ± 2.61
KNN ($k = 1$)	94.95 ± 0.82	94.89 ± 2.02	–	8.47 ± 1.45
KNN ($k = 3$)	95.29 ± 1.24	94.52 ± 1.19	–	6.93 ± 0.76
KNN ($k = 5$)	94.74 ± 0.76	94.15 ± 1.94	–	8.20 ± 1.25
GA	$\mathbf{97.47 \pm 0.79}$	94.68 ± 1.28	6.00 ± 0.35	734.67 ± 120.89

allowed to search for the best MV imputation having available only 200 fitness evaluations in total.

Tables 2-4 bring the results achieved for each combination of dataset, MV rate, and imputation method. In these tables, *Crossval* and *Test* denote the average misclassification rates achieved by the schemes for the training/test data partitions, respectively, whereas *Time* refers to the mean simulation time, in seconds, for a single execution of each imputation technique alone – thus, in case of the GA, it takes into account the time elapsed from the first until the

Table 4. Average results for 20%-level of missing values

Method	Crossval	Test	Generation	Time
	IRIS			
Mean/Mode	88.33 ± 1.33	94.22 ± 4.29	–	0.00 ± 0.00
KNN ($k = 1$)	89.39 ± 2.30	96.00 ± 3.71	–	0.07 ± 0.12
KNN ($k = 3$)	89.22 ± 2.12	95.78 ± 4.07	–	0.00 ± 0.00
KNN ($k = 5$)	89.11 ± 2.55	95.78 ± 3.01	–	0.00 ± 0.00
GA	$\mathbf{90.39 \pm 0.54}$	$\mathbf{98.49 \pm 0.37}$	6.93 ± 1.86	104.20 ± 1.56
	ZOO			
Mean/Mode	87.57 ± 1.22	94.67 ± 2.89	–	0.73 ± 0.12
KNN ($k = 1$)	87.90 ± 3.48	93.67 ± 3.79	–	1.20 ± 0.35
KNN ($k = 3$)	90.04 ± 0.87	93.67 ± 3.21	–	0.93 ± 0.12
KNN ($k = 5$)	89.63 ± 0.85	94.33 ± 3.06	–	1.00 ± 0.20
GA	$\mathbf{95.23 \pm 0.38}$	$\mathbf{95.56 \pm 2.40}$	7.13 ± 1.36	162.60 ± 10.34
	DIABETES			
Mean/Mode	71.58 ± 1.32	77.08 ± 1.32	–	2.20 ± 0.20
KNN ($k = 1$)	70.86 ± 0.73	77.30 ± 1.45	–	2.00 ± 0.00
KNN ($k = 3$)	71.45 ± 0.66	$\mathbf{77.65 \pm 1.46}$	–	2.00 ± 0.00
KNN ($k = 5$)	71.47 ± 0.75	77.60 ± 1.74	–	2.00 ± 0.00
GA	$\mathbf{72.47 \pm 0.25}$	73.64 ± 0.39	6.93 ± 1.01	210.53 ± 0.70
	BCD699			
Mean/Mode	95.87 ± 0.32	95.10 ± 0.72	–	2.00 ± 0.00
KNN (K=1)	94.17 ± 0.20	94.96 ± 1.14	–	2.00 ± 0.00
KNN (K=3)	95.48 ± 0.54	$\mathbf{95.39 \pm 0.87}$	–	2.00 ± 0.00
KNN (K=5)	95.56 ± 0.64	95.39 ± 1.12	–	2.00 ± 0.00
GA	$\mathbf{96.13 \pm 0.32}$	94.48 ± 1.19	7.93 ± 1.21	207.00 ± 1.40
	CAR			
Mean/Mode	75.13 ± 0.84	$\mathbf{82.75 \pm 1.12}$	–	9.47 ± 0.46
KNN (K=1)	74.24 ± 0.25	81.99 ± 0.23	–	6.00 ± 0.00
KNN (K=3)	73.82 ± 0.18	81.78 ± 1.87	–	6.00 ± 0.00
KNN (K=5)	74.43 ± 0.23	81.95 ± 1.67	–	6.00 ± 0.00
GA	$\mathbf{77.94 \pm 0.24}$	82.14 ± 1.33	8.27 ± 0.64	838.60 ± 4.10
	DERMATOLOGY			
Mean/Mode	91.63 ± 0.34	94.61 ± 2.46	–	25.47 ± 9.54
KNN (K=1)	91.15 ± 1.24	94.25 ± 1.45	–	24.33 ± 4.20
KNN (K=3)	91.72 ± 0.84	94.70 ± 1.41	–	23.33 ± 1.14
KNN (K=5)	91.29 ± 0.46	94.25 ± 1.45	–	22.93 ± 3.97
GA	$\mathbf{93.99 \pm 0.83}$	$\mathbf{94.96 \pm 2.17}$	6.53 ± 1.90	$1,702.47 \pm 395.70$

last generation of individuals. Conversely, *Generation* indicates the GA iteration at which the best individual was produced for the first time. Highlighted in these tables are the best results achieved (in terms of mean and standard deviation) for both training/test partitions.

From these quantitative results, it is possible to conclude the following. In terms of effectiveness, the three imputation methods were very competitive, with a prevalence of the genetic scheme, when considering the cross-validation results alone, and the mean/mode and genetic schemes, when considering the test results alone. Taking into account both results simultaneously, it is fair to argue that the genetic imputation has outperformed the others in most of the cases, meaning that it could capture better the different forms of attribute interactions. To ratify this perspective, one should note that, as the mean/mode and KNN imputation schemes do not create models of MV data and do not use the cross-validation results for any purpose, their performance over the training dataset should be considered as important as that over the test dataset.

Moreover, as the MV rate increases, the performance of the three methods decreases slightly, with KNN and mean/mode imputation showing higher sensitiveness to this factor for the training partition. Considering the KNN imputation alone, it seems that the choice of k has not influenced so much both in terms of effectiveness and efficiency. For this last criterion, it is worth mentioning that, in these experiments, we have made use of the implementation of the KNN algorithm available in Weka [10], which is very optimized.

Finally, in terms of efficiency, both mean/mode and KNN imputation methods have prevailed significantly over the genetic one, even considering the fact that, in most of the cases, the number of fitness evaluations necessary to locate the final best individual for the first time was lower than the total number of fitness evaluations available. Therefore, one negative aspect of the genetic scheme is its lack of computational scalability, mainly when considering complex/large datasets.

6 Final Remarks

In this work, we have investigated the relative merits of two machine learning based imputation methods for coping with MV while training RBF network classifiers. Whereas the first method centers upon the k-nearest neighbor algorithm, the second is based on a simple genetic algorithm model and has not been fully explored so far. The performance of these schemes were assessed taking into account different types of atributes, MV rates, and UCI benchmark datasets and also considering as yardstick the performance exhibited by the simple mean/mode imputation. Overall, the simulation results indicate that, in terms of effectiveness, the three methods perform comparatively, with the genetic imputation prevailing in terms of higher generalization endowed to the induced RBF networks. However, if efficiency is a requirement, both KNN and mean/mode imputation are more adequate, mainly for complex datasets.

As future work, we plan to assess the impact of choosing other MV imputation methods [1,3,5,9] over the training/generalization of RBF networks as well as to investigate alternative mechanisms to circumvent the scalability problem presented by the genetic imputation method. A hybrid imputation scheme, combining the KNN and genetic ones, is also underway.

References

1. Acuna, E., Rodriguez, C.: The treatment of missing values and its effect in the classifier accuracy. In: Banks, D., et al. (eds.) Classification, Clustering and Data Mining Applications, pp. 639–648. Springer, Heidelberg (2004)
2. Asunción, A., Newman, D.J.: UCI Machine Learning Repository. University of California at Irvine (1998), `http://ics.uci.edu/~mlearn/MLRepository.html`
3. Batista, G.E., Monard, M.C.: An analysis of four missing data treatment methods for supervised learning. App. Artif. Intel. 17(5), 519–533 (2003)
4. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, Heidelberg (2002)
5. Grzymała-Busse, J.W., Hu, M.: A comparison of several approaches to missing attribute values in data mining. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 378–385. Springer, Heidelberg (2001)
6. Harpham, C., Dawson, W., Brown, R.: A review of genetic algorithms applied to training radial basis function networks. Neural Comp. & App. 13(3), 193–201 (2004)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1998)
8. Hruschka, E.R., Hruschka Jr., E.R., Ebecken, N.F.F.: Missing values imputation for a clustering genetic algorithm. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 245–254. Springer, Heidelberg (2005)
9. Liu, P., Lei, L.: A review of missing data treatment methods. Int. Journal of Intel. Inf. Manag. Syst. and Tech. 1(3), 412–419 (2005)
10. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)