Cluster-based Algorithms for Filling Missing Values

Yoshikazu Fujikawa and TuBao Ho

Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa 923-1292 Japan

Abstract. We first survey existing methods to deal with missing values and report the results of an experimental comparative evaluation in terms of their processing cost and quality of imputing missing values. We then propose three cluster-based mean-and-mode algorithms to impute missing values. Experimental results show that these algorithms with linear complexity can achieve comparative quality as sophisticated algorithms and therefore are applicable to large datasets.

1 Introduction

The objective of this research is twofold. One is to evaluate several well-known missing value methods in order to get a better understanding in their usage. The other is to develop algorithms to deal with missing values in large datasets. The key idea of these algorithms is to divide a dataset with missing values into clusters beforehand and replace missing values on each attribute by mean or mode value of the corresponding cluster depending on the attribute is numeric or categorical, respectively.

2 Evaluation of existing algorithms for missing values

2.1 Classification of missing values cases

Generally, missing values can occur in data sets in different forms. We roughly classify missing values in datasets into three cases: Case 1: Missing values occur in several attributes (columns); Case 2: Missing values occur in a number of instances (rows); Case 3: Missing values occur randomly in attributes and instances. The occurrence cases of missing values can affect the result of missing value methods, so the selection of suitable missing value methods in each case is significant. For example, a method that ignores instances having missing values cannot be used when most of instances have one or more missing values.

2.2 Existing methods for dealing with missing values

We classify methods to deal with missing values into two groups: (i) *pre-replacing* methods that replace missing values before the data mining process, and (ii) *embedded* methods that deal with missing values during the data mining process.

Table 1. Comparative evaluation of methods to deal with missing values

Method	Group	Cost	Attributes	Case
Mean-and-mode method	Pre-replacing	Low	Num & Cat	Case 2
Linear regression	Pre-replacing	Low	Num	Case 2
Standard deviation method	Pre-replacing	Low	Num	Case 2
Nearest neighbor estimator	Pre-replacing	High	Num & Cat	Case 1
Decision tree imputation	Pre-replacing	Middle	Cat	Case 1
Autoassociative neural network	Pre-replacing	High	Num & Cat	Case 1
Casewise deletion	Embedded	Low	Num & Cat	Case 2
Lazy decision tree	Embedded	High	Num & Cat	Case 1
Dynamic path generation	Embedded	High	Num & Cat	Case 1
C4.5	Embedded	Middle	Num & Cat	Case 1
Surrogate split	Embedded	Middle	Num & Cat	Case 1

Among missing value methods from the literature that we consider in this work, statistics-based methods include linear regression, replacement under same standard deviation [6] and mean-mode method [2]; and machine learning-based methods include nearest neighbor estimator, autoassociative neural network [6], decision tree imputation [7]. All of these are pre-replacing methods. Embedded methods include case-wise deletion [4], lazy decision tree [1], dynamic path generation [8] and some popular methods such as C4.5 and CART. Table 1 summarize our evaluation of these methods in terms of their group, computation cost, attributes types and missing value cases applicable.

3 Cluster-based algorithms to deal with missing values

3.1 Basic idea of the algorithms

For large datasets with missing values, complicated methods are not suitable because of their high computation cost. We tend to find simple methods that can reach performance as good as complicated ones.

The results and experience obtained in the previous session suggested us that mean-and-mode method can be efficient and effective for large datasets with necessary improvements. The basic idea of our method is the cluster-based filling up of missing values. Instead of using mean-and-mode on the whole dataset we use mean-and-mode in its subsets obtained by clustering. The method consists of three variants of the mean-and-mode algorithm:

1. Natural Cluster Based Mean-and-Mode algorithm (NCBMM),
2. attribute Rank Cluster Based Mean-and-Mode algorithm (RCBMM),
3. K-Means Clustering based Mean-and-Mode algorithm (KMCMM).

NCBMM uses the class attribute to divide instances into natural clusters and uses the mean or mode of each cluster to fill up missing values of instance

Table 2. Algorithm RCBMM

1. For each missing attribute a_i
2. Make a ranking of all n categorical attributes $a_{j_1}, a_{j_2}, \ldots, a_{j_n}$ in decreasing order of distance between a_i and each attribute a_{j_k}
3. Divide all instances into clusters based on the values of a_h, where a_h is the attribute, which has the highest rank among $a_{j_1}, a_{j_2}, \ldots, a_{j_n}$.
4. Replace missing value on attribute a_i of an instance by the mode of each cluster to which it belongs to.
5. Repeat steps from 2 to 4 until all missing values on attribute a_i are replaced where h changes to the next number of ranking.

belongs to the cluster depending on the attribute id numeric or categorical, respectively. This algorithm is the simplest improvement of the mean-and-mode algorithm in case of supervised data but as shown in next sections it is very efficient if applicable. RCBMM and KMCMM divide a dataset into subsets by using the relationship between attributes. The starting point of these algorithms came from the question whether the class attribute is always the key attribute for clustering an arbitrary descriptive attribute? For some attributes, some of descriptive attributes may be better for clustering than the class attribute. The remark that to cluster instances concerning one missing attribute, the key attribute selected from all attributes may give better results than NCBMM.

3.2 The Proposed Algorithms

Table 3. Algorithm KMCMM

1. For each missing attribute a_i,
2. Make a ranking of all n numeric attributes $a_{j_1}, a_{j_2}, \ldots, a_{j_n}$ in increasing order of absolute correlation coefficients between attribute a_i and each attribute a_{j_k}.
3. Divide all instances by k-means algorithm based on the values of a_h that is the attribute of highest rank among $a_{j_1}, a_{j_2}, \ldots, a_{j_n}$.
4. Replace missing value on attribute a_i by the mean of each cluster.
5. Repeat steps from 2 to 4 till all missing values on attribute a_i are replaced where h changes to the next number of ranking.

Natural Cluster Based Mean-and-Mode algorithm (NCBMM). NCBMM algorithm can be applied to supervised data where missing value attributes can be either categorical or numeric. It produces a number of clusters equal to the number of values of the class attribute. At first, the whole instances are divided into clusters, where instances of each cluster have the same value of the class attribute. Then, in each cluster, the mean value is used to fill up missing values for each numeric attribute, and the mode value is used to fill up missing values for each categorical attribute.

Attribute Rank Cluster Based Mean-and-Mode algorithm (RCBMM). RCBMM (Table 2) can be applied to filling up missing values for categorical attributes independently with the class attribute. It can be applied to both supervised and unsupervised data. Firstly, for one missing attribute, this method ranks attributes by their distance to the missing value attribute. The attribute that has smallest distance is used for clustering. Secondly, all instances are divided into clusters each contains instances having the same value of the selected attribute. Thirdly, the mode of each cluster is used to fill up missing values. This process is applied to each missing attribute.

We can have several ways to calculate distances between attributes. Our idea is to measure how two attributes have similar distributions of values. For this purpose, we used the distance proposed in [5] for two partitions P_A and P_B of n and m subsets of values of attributes A and B.

K-Means Clustering based Mean-and-Mode algorithm (KMCMM). KMCMM can be applied to filling up missing values for numeric attributes independently with the class attribute. Therefore, it can be applied to both supervised and unsupervised data. We describe the algorithm for KMCMM in Table 3. The *correlation coefficient* r used in KMCMM is calculated from p pairs of observations (x, y). The k-means clustering algorithm first randomly selects k of the objects, each of which initially represents a cluster mean or center. Each remaining object is assigned to the cluster who mean is most similar with it. It then computes the new mean for each cluster. This process iterates until the criterion satisfied.

3.3 Evaluation

Methodology. It consists of two phases: (1) filling up missing values on dataset by pre-replacing methods and measuring the executing time, (2) evaluating the quality of the replaced datasets in terms of error rate in classification. Each missing value dataset is filled up by suitable pre-replacing methods to yield datasets without missing values. Then the same data mining methods (See5) are applied to the non-missing value datasets as well replaced datasets and their results are compared in order to evaluate the quality of replacing methods.

Six replacing methods, NCBMM, RCBMM, KMCMM, nearest neighbor estimator, autoassociative neural network and decision tree imputation and one

Table 4. Properties of datasets used in experiments

dataset	inst.	miss inst.	atts	miss atts	class	type of miss att	miss values case
adt	22,747	1,335	13	2	2	cat	case 1&2
att	10,000	2,430	9	8	2	cat	case 1&2
ban	5,400	2,610	30	25	2	mixed	case 2&3
bcw	6,990	160	9	1	2	num	case 1
bio	2,090	150	5	2	2	num	case 3
bld	3,450	560	6	3	2	num	case 1
bos	5,060	860	13	3	3	num	case 1
bpr	3,600	75	16	3	3	cat	case 1
census	299,285	156,764	41	8	2	cat	case 1&2
cmc	14,730	2,002	9	3	3	mixed	case 1
crx	6,900	337	15	7	2	mixed	case 1&2
der	3,660	74	34	1	6	num	case 1
dna	2,372	354	60	3	3	cat	case 1
ech	1,310	230	6	5	2	num	case 3
edu	10,000	9,990	12	10	4	mixed	case 1&2
hab	3,060	562	3	3	2	num	case 3
hco	3,680	3,290	19	18	2	mixed	case 2
hea	3,030	60	13	2	2	cat	case 1
hep	1,550	750	19	15	2	mixed	case 2
hin	10,000	4,050	6	5	3	cat	case 1
hur	2,090	220	6	2	2	num	case 1
hyp	3,772	3,772	29	8	5	mixed	case 1
imp	2,050	80	22	5	5	mixed	case 1
inf	2,380	250	18	2	6	cat	case 1
lbw	1,890	240	8	3	2	mixed	case 1
led	6,000	1,770	7	7	10	cat	case 3
pima	7,680	3,750	8	4	2	num	case 1
sat	6,435	1,195	36	2	6	num	case 1
seg	23,100	3,240	11	3	7	num	case 1&3
smo	28,550	5,340	8	2	3	mixed	case 1
tae	1,510	120	5	5	3	cat	case 3
usn	11,830	10,402	27	26	3	num	case 1&2
veh	8,460	1,550	18	3	4	num	case 1
vot	4,350	420	16	1	2	cat	case 1
wav	3,600	674	21	3	3	num	case 1

embedded method, C4.5, are used for this comparative evaluation. For evaluating the quality of replaced datasets, three classification methods, C4.5, Naive Bayesian classifier, k-nearest neighbor classifiers are used. Fifteen UCI datasets were replaced and classified. The properties of each dataset are summarized in Table 4, and the error rates after treating missing values are summarized in Table 5. The experimenting result shows that NCBMM, RCBMM and KMCMM are much faster than other methods and higher accuracy than others (number in bold). We also compare the methods on the large census dataset containing 299,285 instances with 156,764 instances having missing values. This set has eight numeric and thirty-three categorical attributes except the class attribute and missing values occur on only categorical attributes. We compared the error rate of See5 obtained directly, and that of See5 obtained after replacing missing values by NCBMM and RCBMM. And we measured the execution time for preparing at each replacing method. Experiments were done on a ultra spark machine and Sun-UNIX operating system. In our experiments, See5, NCBMM and RCBMM have the same error of 4.6%, and time to replace missing values for each of our two methods is around 3 and 11 minutes respectively. These show that the proposed cluster-based algorithms could deal with missing values as good as other complicated methods and with low cost, and they can be applied to large datasets.

Table 5. Error rates of datasets after treating missing values

dataset	See5	NCBMM NCBMM	NCBMM RCBMM	KMCMM NCBMM	KMCMM RCBMM	NN NCBMM	NN RCBMM	decision tree	ANN
adt	**15.1**	*15.2*	**15.1**					*15.2*	
att	*2.2*	1.2	1.2					*2.3*	**1.1**
ban	*3.0*	2.5	2.5	2.6	2.5			2.5	
bcw	*0.9*	**0.7**		0.8		0.8		0.8	0.8
bio	*2.2*	1.8		1.8		1.8		1.8	1.8
bld	*4.7*	**3.3**		3.5		3.7		3.7	3.7
bos	*3.8*	2.9		2.9		2.9		2.9	2.9
bpr	3.8	**3.7**	3.8					3.8	*3.9*
cmc	6.7	5.2	5.5	5.2	5.5			*11.2*	**4.7**
crx	*2.5*	2.4	2.3	2.4	2.3	2.3	2.3	2.3	2.3
der	1.4	1.4						1.4	
dna	4.7	4.7	4.7					**4.6**	
ech	*5.4*	**3.5**		**3.5**				4.0	3.9
edu	5.2	**0.0**	**0.0**	**0.0**	**0.0**			7.7	1.8
hab	4.3	**2.9**		**2.9**				*7.6*	3.8
hco	0.9	**0.0**	**0.0**	**0.0**	**0.0**			*2.8*	*2.7*
hea	3.0	3.0	3.0					3.0	3.0
hep	*3.8*	**1.3**	**1.3**	**1.3**	**1.3**	2.5	2.6	2.2	2.2
hin	*12.9*	**9.0**	10.6					10.4	11.3
hur	*2.8*	**2.0**		**2.0**		2.2		2.2	2.1
hyp	0.4	0.4	0.4	0.4	0.4			0.4	0.4
imp	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	
inf	2.4	**2.3**	**2.3**					2.4	2.4
lbw	5.1	**3.9**	**3.9**	**3.9**	**3.9**	5.2	5.2	*8.6*	4.7
led	29.1	**25.1**	**25.1**					25.2	*32.0*
pima	*3.7*	**1.5**		**1.5**		3.0		2.9	2.9
sat	13.1	13.6		*14.1*		13.5		**9.0**	13.1
seg	*1.2*	**0.6**		**0.6**		0.7		0.7	0.7
smo	2.2	**1.3**	**1.3**	**1.3**	**1.3**	1.9	1.9	*7.3*	1.6
tae	**2.8**	**2.8**	4.0					*11.2*	
usn	*4.2*	**1.3**		2.4				2.8	2.9
veh	*3.0*	**2.7**		**2.7**		**2.7**		2.8	2.8
vot	*1.9*	**1.0**	**1.0**					1.1	1.4
wav	*24.5*	24.0		23.5		*24.6*		**13.1**	24.4

References

1. Friedman, J. H., Khavi, R., Yun, Y.: Lazy Decision Trees. Proceedings of the 13th National Conference on Artificial Intelligence, 717-724, AAAI Pres/MIT Press, 1996.
2. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.
3. Kononenko, I., Bratko, I., Roskar, E.: Experiments in automatic learning of medical diagnostic rules. Technical Report. Jozef Stefan Institute, Ljubjana, Yogoslavia, 1984.
4. Liu, W.Z., White, A.P., and Thompson S.G., Bramer M.A.: Techniques for Dealing with Missing Values in Classification. In IDAf97, Vol.1280 of Lecture notes, 527-536, 1997.
5. Mantaras, R. L.: A Distance-Based Attribute Selection Measure for Decision Tree Induction. Machine Learning, 6, 81-92, 1991.
6. Pyle, D.: Data Preparation for Data Mining. Morgan Kaufmann Publishers, Inc, 1999.
7. Quinlan, J.R.: Induction of decision trees. Machine Learning, 1, 81-106, 1986.
8. White, A.P.: Probabilistic induction by dynamic path generation in virtual trees. In Research and Development in Expert Systems III, edited by M.A. Bramer, pp. 35-46. Cambridge: Cambridge University Press, 1987.