Handling Missing Values for the CN2 Algorithm

Cuong Duc Nguyen(✉), Phuong-Tuan Tran, and Thi-Thanh-Thao Thai

HCMC University of Foreign Languages - Information Technology,
Ho Chi Minh City, Viet Nam
{cuong.nd,tuantranphuong,thao.ttt}@huflit.edu.vn

Abstract. Missing values are existed in several practical data sets. Machine Learning algorithms, such as CN2, require missing values in a data set be pre-processed. The estimated values of a missing value can be provided by Data Imputation methods. However, the data imputation can introduce unexpected information to the data set so that it can reduce the accuracy of Rule Induction algorithms. If missing values can be directly processed in Rule Induction algorithms, the overall performance can be improved. The paper studied the CN2 algorithm to propose a modified version, CN2MV, which is able to directly process missing values without preprocessing. Testing on 17 benchmarking data sets from the UCI Machine Learning Repository, CN2MV outperforms the original algorithm using data imputations.

Keywords: CN2 · Missing value · Rule induction · Data imputation

1 Introduction

Several practical data sets from research and industry have missing values (MV). MVs can be from non-response fields in surveys. Users can intentionally skip sensitive items, unintentionally bypass some difficult fields or be too busy to fill the full survey. MVs can also come from technical errors due to machine malfunction. Human errors in data inputting can also cause MVs.

The majority of Machine Learning and Data Mining classification algorithms can only process a complete data set, in which, a missing value has to be preprocessed by filling with a value, such as a mean, a pre-defined constant or a imputed value based on other available values. A research question is the difference in the criteria of missing-value preprocessing technique and the criteria of the main learning algorithm can decrease the overall performance. If the missing-value processing is integrated into the main learning algorithm, the whole efficiency may be improved.

This paper focuses on supporting CN2, a popular Rule Induction algorithm, with the capability of processing missing values during the learning process. In Sect. 2, techniques of processing missing values in data sets have been reviewed.

P. Cong Vinh and V. Alagar (Eds.): ICCASA 2018/ICTCC 2018, LNICST 266, pp. 226–234, 2019.
https://doi.org/10.1007/978-3-030-06152-4_20

CN2 is also reviewed in Sect. 2. Section 3 represents the CN2MV algorithm, the modified version of CN2. CN2MV will be tested with benchmarking data sets and compared with the CN2 in Sect. 4. Section 5 is the conclusion.

2 Related Work

2.1 Processing Missing Values by Data Imputations

Data Imputation (DI) is the well known technique to calculate an estimated value to replace for a MV. Several Machine Learning algorithms cannot process MVs directly so that DI are used to replace MVs in the pre-processing phase. The estimated value can be a constant, a mean or mode of an attribute, or the result from an estimation model. A deep review of DIs can be seen in Little and Roben's book [10].

The MV existence and the applying of DI methods has bold effects on the performance of Machine Learning algorithms. Incomplete data in either the training/test set or in both sets affects on the prediction accuracy of learned classifiers [3]. Wohlrab, Lars and Fürnkranz [16] studied possible strategies for handling missing values in separate-and-conquer rule learning algorithms, and compared them experimentally on a large number of datasets. The correlation between the data imputation methods and the classification algorithms are experimentally examined by Luengo et al. [11]. That study studies the impact of fourteen data imputation methods on three groups of classification algorithms: rule induction, approximate models and lazy learning. The experiment shows that, for each group of classification algorithms, there are different set of appropriate data imputation methods. Even in each group, a different set of data imputation methods supports a classification algorithm to achieve a good performance. Therefore, the correlation between imputation and learning models can decide the whole performance for classification methods.

Focusing on Rule Induction algorithms, this paper only studies data imputation methods, which provide the best results for Rule Induction algorithms in the Luengo research [11]. This subsection describe five imputation methods used in this paper.

- Case deletion or Ignore Missing (IM). In this method, all instances having a missing value are omitted from the data set. This simplest method is only suitable for data sets with a small percentage of missing values.
- Most Common Attribute Value for Symbolic Attributes, and Average Value for Numerical Attributes (MC) [4]. With this method, for nominal attributes, a missing value is replaced with the most common attribute value, and for numerical values, a missing value is replaced with the average value of the corresponding attribute. This method may be the most popular imputation technique.
- Concept Most Common Attribute Value for Symbolic Attributes, and Concept Average Value for Numerical Attributes (CMC) [4]. Similar as MC, a missing value is replaced by the most repeated one if nominal or the mean

value is numerical, but considering only the instances with the same class as the reference instance. This is an advanced method of MC when considering the class attribute of the imputed data instance.
- Imputation with Fuzzy K-means Clustering (FKMI) [8]. FKMI uses the membership function, which describes the degree to which this data object belongs to a certain cluster, and the values of cluster centroids, to impute a missing value.
- Support Vector Machines Imputation (SVMI) [5]. SVMI applies an SVM regression-based algorithm to fill in missing values.

2.2 Rule Induction

CN2 [1,2] is one of the most popular Rule Induction algorithms [14]. CN2, a typical Separate-and-Conquer algorithm, induces the best rule, called ”complex” on the current training set, removes covered examples from the current training set, and repeat the process on the reduced training set until no more rules can be induced. To find the best complex, CN2 carries out a pruned general-to-specific beam search. At each stage or the search, CN2 examines the specializations of complexes in the current beam. CN2 evaluate specialized complexes by the rule's entropy (in [2]) or Laplace (in [1]). There are two versions of CN2 [1]: ordered (rules in the rule set are ordered in applying) and unordered (rules in the rule set are unordered in applying). As authors point out, the unordered rule set has much advantage than the ordered one.

The content of the unordered CN2 algorithm are shown in Fig. 1. CN2 induces rules for each class in the training set. For a class, CN2 consequently find the best complex and removes from the current training set all examples in the current class covered by the rule created by the best complex. This process is repeated until CN2 cannot find any best complex for the current class.

CN2-SD [7] is also a beam search Rule Induction algorithm, which adapts the CN2 classification rule learner to subgroup discovery. CN2-SD employs the original CN2 algorithm with the weighted relative accuracy to mine descriptive rules, so that the criterion of CN2-SD is not to maximize the predictive classification of the induced rule set. Using the same criterion with CN2-SD, DoubleBeam-SD [14] uses two separate beams and can combine various heuristics for rule refinement and selection to find rules with high descriptive capability.

Using the same classification criterion with CN2, DoubleBeam-RL [14] uses two separate beams and can combine various heuristics for rule refinement and selection, which widens the search space and allows for finding rules with improved classification capabilities.

Due to the time limit, the paper focus on showing the efficiency of integrating directly processing MVs in the learning process of CN2. The method can be applied on other CN2 versions.

```
procedure CN2unordered(allexamples,classes):
let ruleset={}
for each class in classes:
    generate rules by CN2ForOneClass(allexamples,class)
    add rules to rule set
return ruleset

procedure CN2ForOneClass(examples, class):
let rules={}
repeat
    call FindBestComplex(examples, class) to find bestcond
if bestcond is not null
then add the rule if bestcond then predict class to rules
    &  remove from examples all examples in class covered by bestcond
Until bestcond is null
Return rules

procedure FindBestComplex(examples E, class C)
Let STAR be the set containing the empty complex
Let BEST_CPX be nil
While STAR is not empty,
    Specialize all complexes in STAR as follows:
    Let NEWSTAR be the set $\{x \wedge y | x \in STAR, y \in SELECTORS\}$
    Remove all complexes in NEWSTAR that are either in STAR (i.e., the unspecialized ones)
        or null (e.g., $big = y \wedge big = n$)
    For every complex $C_i$ in NEWSTAR:
        If $C_i$ is statistically significant and better than BEST_CPX by user-defined criteria
            when tested on E,
        Then replace the current value of BEST_CPX by $C_i$
    Repeat until size of NEWSTAR $\leq$ user-defined maximum:
        Remove the worst complex from NEWSTAR
    Let STAR be NEWSTAR
Return BEST_CPX
```

Fig. 1. The CN2 induction algorithm [1]

2.3 Processing Missing Data During Learning Process of Learning Algorithms

There is only a few attempts in processing MVs in ID3 [12]. The approaches dealing with missing values in decision tree induction can be characterized by three groups [6] as follows:

1. Evaluation of a test (in a tree node): This concerns the strategy of how to evaluate different tests when each attribute has a different amount of missing values.
2. Partitioning the training set using a test: This relates to the strategy assigning a case with missing value on an attribute considered in a test.
3. Classifying a new case with unknown value of a tested attribute: This is correspondent to the application of the learned classifier.

Quinlan [13] proposed a method to adopt the ID3 algorithm to process missing values by making changes in evaluating a decision, partitioning data subsets and classifying a new unseen data instance. In comparing the performance several modified versions of ID3, the version with changes in the learning process when processing missing values achieved the best result. In the best version, the Information Gain is reduced by the percentage of missing values in evaluating a decision, a fraction of each data instance with a missing value is assigned to data subsets and when classifying a unseen data instance with a missing value, that data object is considered on all branches of a decision. The experiments of Quinlan's research also shows the versing with modifying the learning process achieves better results than versions using missing-value pre-processing techniques, such as mean replacement or data imputation, the modified version of ID3 achieved the lowest average error.

3 Integrating Missing-Value Processing into CN2

In CN2, the four following problems have to be addressed when processing MVs:

1. Generate set SELECTORS. Procedure "FindBestComplex" in CN2 creates NEWSTAR by combining current complexes in the beam with complexes in SELECTORS. An imputed value of a MV can make the CN2 generate unwanted complex in SELECTORS.
2. Evaluation of a complex. Every complex C_i is evaluated by the Laplace measurement. If the training set has several MVs, which are imputed in the preprocessing step, the score of a complex can be incorrectly measured.
3. Remove covered positives instances by the "bestcond" in procedure "CN2ForOneClass". When a new rule is induced and added to the rule set, CN2 removes from "examples" (the current training set) all examples in "class" (the current class) covered by the rule. If an example has a MV, which is imputed in the preprocessing step, it can be accidentally covered by the rule.
4. Classify an unseen instance. The final output of CN2 is a rule set. When classifying an unseen data sample x having an imputed value for a MV, x can be wrongly classified by that imputed value.

The paper introduces CN2MV, an improved version of CN2, with four changes. The main content of CN2MV is similar to CN2 (see Fig. 1) but changes are made in its implementation to address four mentioned problems. CN2MV induces rules from a data set having MVs without data imputation. The four changes are made as follows:

1. Generate set SELECTORS: If any attribute-value pair has a MV, it is not used in generating SELECTORS. For instances, with an example as $(V_1 = A_1, V_2 =?, V_3 = A_3)$, the pair of $(V_2 =?)$ is not used in generating conditions.
2. Evaluation of a complex: When evaluating a complex, a MV can be any value in the corresponding attribute. When checking the covering of rule $(\textit{if}\ V_1 = A_1\ \textit{and}\ V_2 = A_2\ \textit{then...})$ on example $(V_1 = A_1, V_2 =?, V_3 = A_3)$, the result is true in such cases.
3. Mark covered instances by the "bestcond" in procedure "CN2ForOneClass": When checking whether a data sample is covered by a complex, a MV can be any value in the corresponding attribute (similar approach as "Evaluation of a comple").
4. Classify an unseen instance: Similar approach as in evaluating a complex is used.

In these four changes, the first modification help CN2MV avoid generating complexes from imputed data. The second change improves the complex evaluation when treating MVs as unknown values. Similar approach in the MV treatment are used in the third and four changes. Especially, the fourth change is more natural in treating MVs in unseen data than using DI methods. Each un-seen data in the testing set independently enter the learned classifier, no information for DI methods to estimate the value for MVs. In addition, supervised DI methods, such as CMC or SVMI, cannot be used because the class/concept of a sample is unknown in the testing set.

4 Experiment

4.1 Data Sets

To evaluate new algorithms, 17 benchmarking data sets are selected from the UCI Machine Learning repository [9]. The characteristics of selected data sets are described in Table 1. These data sets are selected due to the existence of MVs. For data sets without MVs, the modifications proposed in CN2MV have no effect, so that the performances of CN2 and CN2MV are the same.

Table 1. Benchmarking data sets used in the experiments

Data set	Acro.	#Inst.	#Attr.	#Cls	%MV	%Inst. with MV
Audiology	AUD	226	71	24	1.98	98.23
Autos	AUT	205	26	6	1.11	22.44
Bands	BAN	540	40	2	4.63	48.7
Breast-cancer	BRE	286	10	2	0.31	3.15
Breast-w	BRW	699	10	2	0.23	2.29
Cleveland	CLE	303	14	5	0.14	1.98
Colic	COL	368	23	2	22.77	98.10
Credit-a	CRA	690	16	2	0.61	0.61
Dermatology	DER	365	35	6	0.06	2.19
Heart-c	HRC	303	14	5	4.73	34.09
Hepatitis	HEP	155	20	2	5.39	48.39
Labor	LAB	57	17	2	33.64	98.24
Mammographic	MAM	961	6	2	2.81	13.63
Mushroom	MUS	8124	23	2	1.33	30.53
Primary tumor	PRT	339	18	21	3.69	61.06
Soybean	SOY	307	36	19	6.44	13.36
Vote	VOT	435	17	2	5.30	46.67

4.2 Settings of Experiments

The CN2MV algorithm is implemented in the Weka framework [15]. Each numeric attributes are discretized by the ten uniform-bin method. Because Weka only has a few Data Imputation methods, 5 mentioned imputation methods are carried out in the KEEL framework [11] on the training sets. The imputed data sets will be imported to Weka to execute the CN2 algorithm.

Table 2 shows the parameters used by applied imputation methods (see Sect. 2.1). These parameters are default values set up in KEEL.

Table 2. Method Parameters

Methods	Parameters
SVMI	Kernel = RBF, C = 1.0, Epsilon = 0.001, Shrinking = No
FKMI	K = 3, Iterations = 100, Error = 100, m = 1.5
CN2 & CN2MV	Max length of complex = 2, Beam size = 5

4.3 Results

Table 3 shows the error rates of tested CN2 versions in the ten-fold cross-validation. CN2MV achieves the best result on 11 of 17 tested data sets. Specially, on data set Bands, Colic, and Labor, CN2MV has much better results than others from CN2 using data imputation methods in preprocessing MVs.

Table 3 also shows that CN2MV achieves the best average result when comparing with other methods. Method IM-CN2 has the poorest results, especially on data sets with high percentage instances with MVs (Audiology, Bands, Colic, Labor and Primary tumor) because they omits much valuable information from the training set. Method SVMI-CN2 achieves the second best average result but it has one shared best result.

Table 3. Average error rates of tested methods

Data set	CN2MV (%)	IM-CN2 (%)	MC-CN2 (%)	CMC-CN2 (%)	FKMI-CN2 (%)	SVMI-CN2 (%)
AUD	**33.10**	98.26	35.36	39.35	35.36	38.50
AUT	**22.93**	29.17	23.86	26.74	23.86	26.31
BAN	**28.56**	38.02	39.68	38.20	39.68	37.10
BRE	**31.81**	32.52	34.25	34.25	34.25	34.25
BRW	6.58	**5.44**	5.87	5.73	5.87	5.87
CLE	43.27	42.26	41.30	**41.29**	41.30	41.97
COL	**16.52**	36.95	19.27	24.73	19.27	20.35
CRA	**13.33**	15.07	13.91	13.91	13.91	14.49
DER	**6.00**	**6.00**	6.82	6.55	6.82	6.82
HRC	22.35	22.38	21.40	22.04	**21.40**	23.05
HEP	18.75	18.63	18.79	**14.92**	18.79	16.17
LAB	**17.67**	64.67	37.33	33.00	37.33	24.67
MAM	18.00	18.21	**17.69**	18.21	17.69	18.21
MUS	**0.00**	16.94	0.18	0.18	0.18	0.18
PRT	**54.88**	59.57	56.35	59.29	56.35	56.66
SOY	**17.87**	27.24	18.74	19.47	18.74	18.58
VOT	**4.14**	5.05	4.37	4.37	4.37	**4.14**
Avg	**19.97**	30.03	22.17	22.60	22.17	21.77

5 Conclusion

The paper proposed the CN2MV algorithm, which is able to directly process missing values without data imputation in preprocessing. Four main changes has been proposed to efficiently process missing values during the rule inducing process. Testing on 17 benchmarking data sets from the UCI Machine Learning Repository, the modified versions outperformed the original algorithms using data imputation techniques to pre-process missing values.

References

1. Clark, P., Boswell, R.: Rule induction with CN2: some recent improvements. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 151–163. Springer, Heidelberg (1991). https://doi.org/10.1007/BFb0017011
2. Clark, P., Niblett, T.: The CN2 induction algorithm. Mach. Learn. **3**(4), 261–283 (1989)
3. Gheyas, I.A., Smith, L.S.: A neural network-based framework for the reconstruction of incomplete data sets. Neurocomputing **73**(16), 3039–3065 (2010)

4. Grzymala-Busse, J.W., Goodwin, L.K., Grzymala-Busse, W.J., Zheng, X.: Handling missing attribute values in preterm birth data sets. In: Ślęzak, D., Yao, J.T., Peters, J.F., Ziarko, W., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 342–351. Springer, Heidelberg (2005). https://doi.org/10.1007/11548706_36
5. Honghai, F., Guoshun, C., Cheng, Y., Bingru, Y., Yumei, C.: A SVM regression based approach to filling in missing values. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3683, pp. 581–587. Springer, Heidelberg (2005). https://doi.org/10.1007/11553939_83
6. Latkowski, R.: High computational complexity of the decision tree induction with many missing attribute values. In: Proceedings of Concurrency, Specification and Programming, CS&P 22, pp. 318–325 (2003)
7. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup discovery with CN2-SD. J. Mach. Learn. Res. **5**, 153–188 (2004)
8. Li, D., Deogun, J., Spaulding, W., Shuart, B.: Towards missing data imputation: a study of fuzzy k-means clustering method. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS (LNAI), vol. 3066, pp. 573–579. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25929-9_70
9. Lichman, M.: UCI machine learning repository (2013). http://archive.ics.uci.edu/ml
10. Little, R.J., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley, Chicester (2002)
11. Luengo, J., García, S., Herrera, F.: On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowl. Inf. Syst. **32**(1), 77–108 (2012)
12. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
13. Quinlan, J.R.: Unknown attribute values in induction. In: Proceedings of the International Machine Learning Workshop, pp. 164–168 (1989)
14. Valmarska, A., Lavrač, N., Fürnkranz, J., Robnik-Šikonja, M.: Refinement and selection heuristics in subgroup discovery and classification rule learning. Expert Syst. Appl. **81**, 147–162 (2017)
15. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques, 4th edn. Morgan Kaufmann, San Mateo (2016)
16. Wohlrab, L., Fürnkranz, J.: A review and comparison of strategies for handling missing values in separate-and-conquer rule learning. J. Intell. Inf. Syst. **36**(1), 73–98 (2011)