Journal of Computer Science 2 (9): 740-745, 2006
ISSN 1549-363

A framework to Deal with Missing Data in Data Sets

Luai Al Shalabi, Mohannad Najjar and Ahmad Al Kayed
Faculty of Computer Science and Information Technology, Applied Science University, Jordan

Abstract: Most information systems usually have some missing values due to unavailable data. Missing values minimizing the quality of classification rules generated by a data mining system. Missing vales also affecting the quantity of classification rules achieved by the data mining system. Missing values could influence the coverage percentage and number of reducts generated. Missing values lead to the difficulty of extracting useful information from that data set. Solving the problem of missing data is of a high priority in the field of data mining and knowledge discovery. Replacing missing values by a specific value should not affect the quality of the data. Four different models for dealing with missing data were studied. A framework is established that remove inconsistencies before and after filling the attributes of missing values with the new expected value as generated by one of the four models. Comparative results were discussed and recommendations were concluded.

Key words: Data mining, missing data, rules, reducts, coverage

INTRODUCTION

The growth of the size of data and number of existing databases far exceeds the ability of humans to analyze this data, which creates both a need and an opportunity to extract knowledge from databases[1]. Medical databases have accumulated large quantities of information about patients and their medical conditions. Relationships and patterns within this data could provide new medical knowledge. Analysis of medical data is often concerned with treatment of incomplete knowledge, with management of inconsistent pieces of information and with manipulation of various levels of representation of data. Existing intelligent techniques[2,3] of data analysis are mainly based on quite strong assumptions (some knowledge about dependencies, probability distributions. Large number of experiments), are unable to derive conclusions from incomplete knowledge, or can not manage inconsistent pieces of information. The most commonly intelligent techniques used in medical data analysis are neural network[4], Bayesian classifier[5], genetic algorithms[6], decision trees[7], fuzzy theory[8-10].

Data mining has come to refer to the process of analyzing data and generating new knowledge, hopefully understandable by humans, which was previously hidden and undetected. The overall goal is to create a simplified model of the domain under study. Discovery systems have been applied to real databases in medicine[11,12], astronomy[13], the stock market[14] and many other areas.

The collected data reflects the uncontrolled real world, where many different causes overlap and many patterns are likely to exist simultaneously. The patterns are likely to have uncertainty: if A, then B with uncertainty C. Many methods for deriving such patterns have been proposed, including Gaines and Shaw in[15], Quinlan in[16], Clark and Niblet in[17], Pawlak in[8] and some others.

One common problem or challenge in data mining and knowledge discovery research is a noisy data[18]. In a large database or data sets, many of the attribute values are inexact or incorrect. This may be due to an erroneous instrument measuring some property or human error when registering it. There are two forms of noise in the data as described below.

Corrupted values: sometimes some of the values in the training set are altered from what they should have been. This may result in one or more tuples in the data set conflicting with the rules already established. The system may then regard these extreme values as noise and ignore them. The problem is that one never knows if the extreme values are correct or not and the challenge is how to handle "weird" values in the best manner. More information exists in[19,20].

Missing attribute values: one or more of the attribute values may be missing both for examples in the training set and for objects which are to be classified[19,20]. Missing data might occur because the value is not relevant to a particular case, could not be recorded when the data was collected, or is ignored by users because of privacy concerns[21]. If attributes are missing in any training set, the system may either ignore this object totally, try to take it into account by, for instance, finding what is the missing attribute's most probable value, or use the value "missing", "unknown" or "NULL" as a separate value for the attribute.

Corresponding Author: Luai Al Shalabi, Faculty of Computer Science and Information Technology, Applied Science University, Jordan

The problem of missing values has been investigated since many times ago[22,23]. The simple solution is to discard the data instances with some missing values[24]. A more difficult solution is to try to determine these values[25]. Several techniques to handle missing values have been discussed in the literature[18,23,25-28].

Some popular methods are as follows: Ignore the tuple: this is usually done when the class label is missing. Also, it is recommended if the tuple contains several attributes with missing values.

Use a global constant to fill in the missing value. Replace all missing attribute values by the same constant, such as a label like "Missing", "Unknown", "-∞", or "?".
Use the attribute mean to fill in the missing value.
Use the attribute mean for all samples belong to the same class: for example, if classifying customers according to credit_risk, replace the missing value with the average income value for customers in the same credit risk category as that of the given tuple.
Use the most probable value to fill in the missing value: this technique is appropriate for sparse missing values. Difficulties arise if the tuple contains more than one missing attribute values.

MATERIALS AND METHODS

The original (historical) data set was cleaned from all kind of inconsistencies. All tuples with same conditional attributes and different classification attribute were removed. This will clearly improve efficiency as it will remove all suspected cases.

The experiments were then designed to check the best model to fill in the missing values that generates the highest coverage of the data set. Coverage represents the ratio of classified (recognized by classifier) objects from the class to the number of all objects in the class. Number of classification rules and number of reducts were generated by each model. Minimum number of rules is recommended because it minimizes the executing classification process time of new non classified tuples. Big number of rules leads to a time consuming. Results were compared and recommendations were summarized. The experiments were run using the RSES system (Rough Set Exploration System). Number of rules and number of reducts for each data set that contains a particular replacement of missing values were observed. The HSV and the heart disease data sets were taken from the UCI repository[29]. Both data sets contain no missing data. The HSV data set contains 122 tuples while the heart disease data set consists of 270 tuples. To simulate the data with missing values, some values were removed from the original data sets. The new HSV data set contains 63 tuples with missing values varying between 1 to 9 missing values. The new heart disease data set contains 81 tuples with missing values varying between 1 and 2 missing values. Four different data sets were generated from each original one. Each data set is described as follows:

DS1: represents the data set that generates and replaces missing values with the global constant "Missing" which is a well known entry in the field of data mining. The replacement process was done through model 1 that algorithm 1 describes it. The global constant "Missing" is representing unknown values in the data set.

Algorithm 1: Prediction of missing values
Accept a decision table T=(U,C,D,V).
For each attribute with missing data do
Replace the missing data with the
value "Missing".
Enddo
T is a Data set without missing values.
The complexity of this algorithm is O(n) so it is simple algorithm.

DS2: represents the data set that generates and replaces missing values with the mean value of that attribute in the entire data set. The replacement process was performed through model 2 which is described by algorithm 2.

Algorithm 2: Prediction of missing values
Accept a decision table T=(U,C,D,V).
For each attribute with missing data do
Find the average value X
Replace the missing data with the average value
Enddo
T is a Data set without missing values.
The complexity of this algorithm is O(n) so it is also simple algorithm.

Algorithm 3: Prediction of missing values
Accept a decision table T=(U,C,D,V).
partition a decision table horizontally
into subsets:
$T_1=(U_1,C,D_1,V)$, $T_2=(U_2,C,D_2,V)$, …, $T_n=(U_n,C,D_n,V)$
where $U=(U_1,U_2,\ldots,U_n)$ and $D=(D_1,D_2, \ldots,$ and $D_n)$.
For each subset do
For each attribute with missing data
do
Find the average value X.
Replace the missing data with the average value.
Enddo.
Enddo
T1, T2, …, Tn are subsets without missing values.
The complexity of this algorithm is $O(n^2)$ so it takes longer time to be completed.

DS4: represents the data set that removes all examples that contain missing values. The deletion process is performed via model 4 that algorithm 4 represents it.

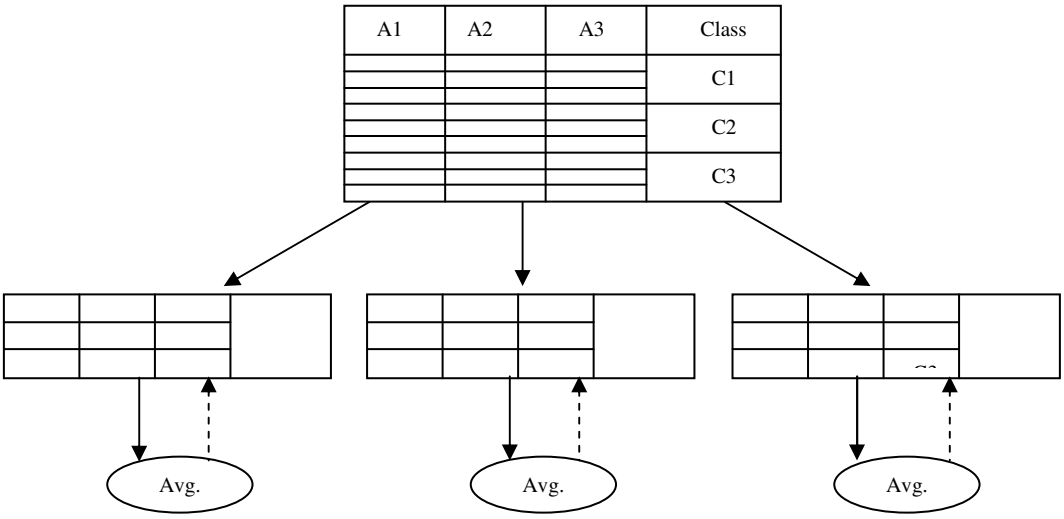

Fig. 1: Model 3

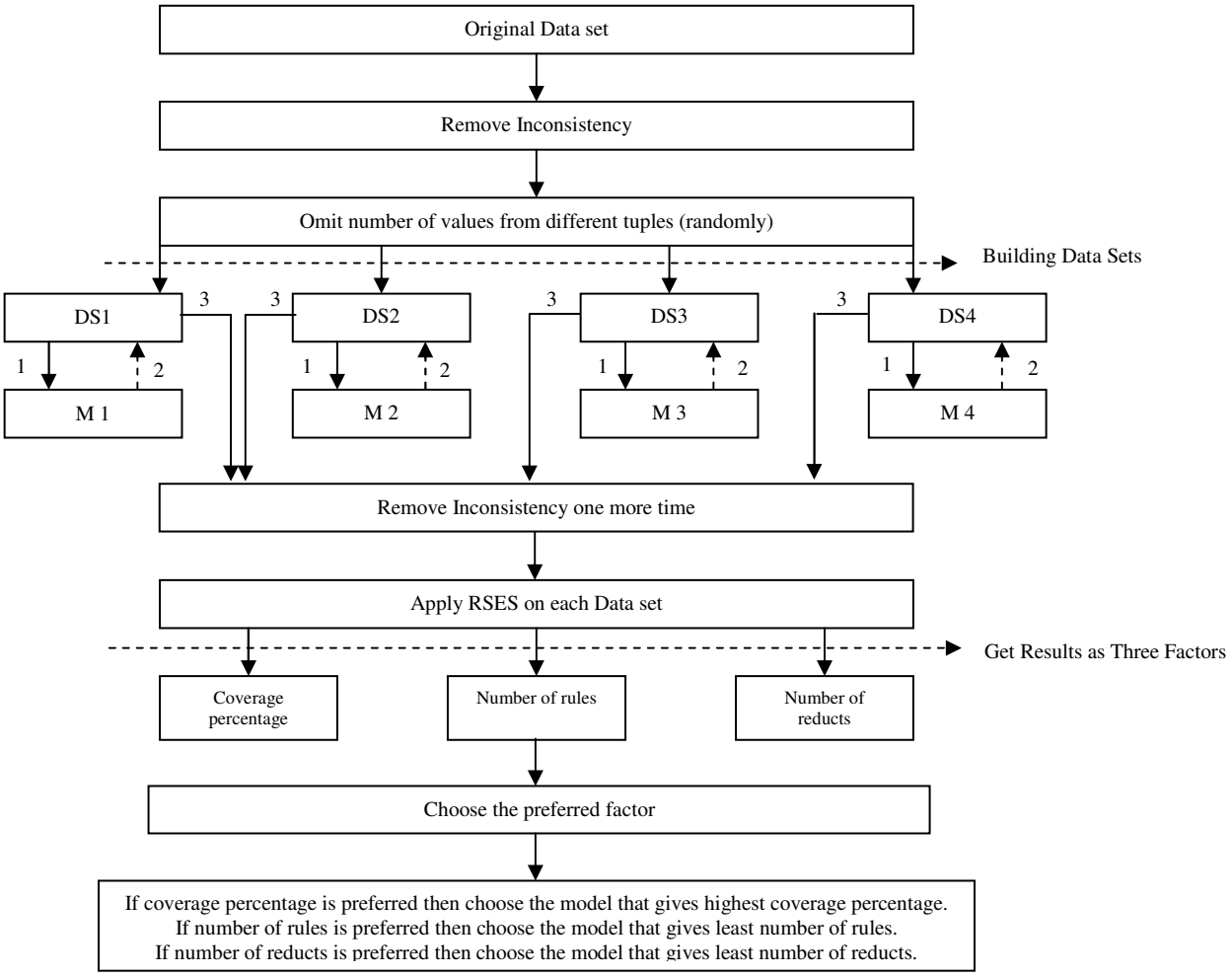

Fig. 2: The framework of guessing missing values and testing it using different factors. DS stands for Data Set and M stands for Model

Table 1: Number of rules, number of reducts, and the coverage of the HSV data set, N represents the model number which described earlier.

Data set name+N	Coverage percentage	Number of rules	Number of reducts
HSV1	73	1070	78
HSV2	93.4	1400	57
HSV3	95.1	1332	41
HSV4	93.7	707	39

Table 2: Number of rules, number of reducts and the coverage of the Heart disease data set, N represents the model number which described earlier

Data set name+N	Coverage percentage	Number of rules	Number of reducts
Heart_Disease1	93	4803	109
Heart_Disease2	97	5189	89
Heart_Disease3	96.7	5009	91
Heart_Disease4	98.4	3173	86

Algorithm 4: Prediction of missing values.
Accept a decision table T=(U,C,D,V).
For each tupple with missing data do
delete the tuple
Enddo
T is a Data set without missing values.
The complexity of this algorithm is O(n) so it is considered simple algorithm.

After the process of replacement, inconsistencies are checked one more time. New replacements may cause such kind of inconsistencies as it is not a perfect process. If consistencies are existing then a deletion process has to take place. All tuples with same predicting attributes and different predicted attributes are ignored. This may minimize the sample training data set one more time but on the other hand the accuracy training results will be increased. As it is a fact, training of a data set of minimum pollution will be resulted in stronger kind of knowledge than it is the case of data set with more amounts of pollutions.

The framework is represented in Fig. 2. It describes all steps that the process of guessing a suitable missing value goes throw. Testing of the models goes throw 3 different factors as exist in the framework in Fig. 2.

RESULTS AND DISCUSSIONS

Tables 1 and 2 summarize the number of rules, the number of reducts and the coverage percentage. Each of which was generated from each of the four models that have designed earlier. The first comparison was done in order to evaluate the best model that gives the highest coverage percentage. For the HSV data set, the highest coverage was for model 3. It gave the coverage ratio 95.1%. Model4 gave the next best results which was 93.7%. In the third place of priority, model2 was taken its place as it gave the coverage percentage of 93.4%. The worst model was model1 and its coverage was 73%.

As the heart disease data set was used, the coverage percentages for model1, model2, model3 and model4 were 93, 97, 96.7 and 98.4% respectively. The best coverage percentage was for model4 that removes all tuples with missing values. The worst coverage percentage was for model1 that uses "missing" global constant. Model3 was in the third place of priority while model2 was in the second place of priority.

The common conclusion is that model1 gave the worst coverage percentage in both data sets: The HSV and the heat disease. The other models were varying in their results and results were close to each other. This may conclude that in general there is no best model to solve missing values that we may always use for any data set. Choosing a suitable model for a data set depends on that data set that we want to learn.

The HSV and the heart disease data sets are two case studies. Model3 is recommended for building the best classification system for HSV data set. Model4 is recommended for building the best classification system for the heart disease data set.

In the two experiments and among the four models, model4 gave the minimum number of classification rules (707). It means that this model is the best among the four different models. The second best model was model1 that generated 1070 rules. Model3 was in the third place and it generated 1332 rules. Finally, model2 was the last choice to give the minimum number of rules and it generated 1400 rules.

The third comparison was focusing on the number of reducts that each model generates. Two different points of views were considered. First, consider that the big number of generated reducts is the most preferred because this large number of reducts will give a wide range of reducts to user. This variance of reducts will help a user to choose the most suitable reduct for the organization. If this is the case, model1 was the best model in both experiments and it generated 78 reducts from the HSV data set and 109 from the heat disease data set. Model3 was the second best choice for the heart disease data set and it generated 91 reducts. Model2 was the best second choice for the HSV data set and it generated 57 reducts. The third place of priority as the heart disease data set is used was given to model2 that generated 89 reducts. The third place of priority as the HSV data set is used was given to model3 that generated 41 reducts. The last choice in both data sets was for model4 and it generated 39 reducts from the HSV data set and 86 reducts from the heart disease data set. The last priority was given to model4 because of the new size of this data set. After

the deletion of all examples of missing values, the data set was becoming smaller in size and this may lead to have many relationships in the data set especially if many related examples were deleted.

The second point of view is to consider that the best model is the one which gives the smaller number of reducts. This point of view considers a time as an important factor. The computing time of generating reducts is minimized. In other word, the number of computations and comparisons in the data set is

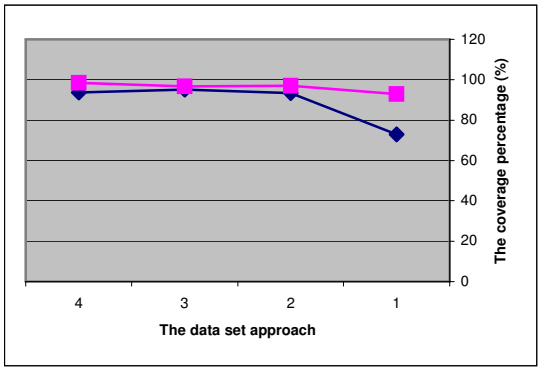

Fig. 3: The relationship between the four types of data sets and the coverage percentages.

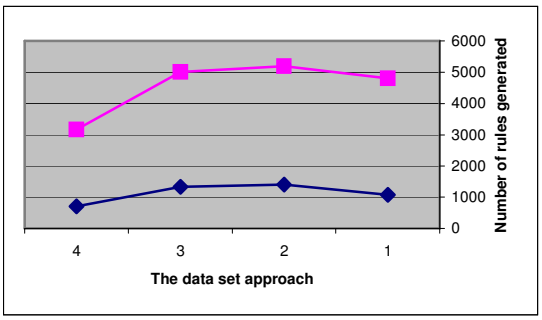

Fig. 4: The relationship between the four types of data sets and the number of rules generated

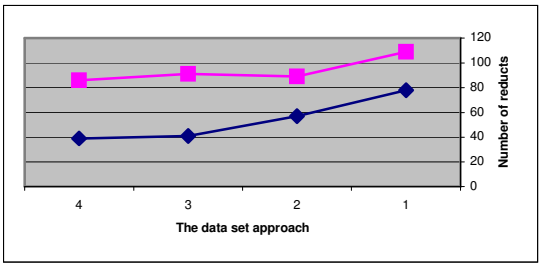

Fig. 5: The relationship between the four types of data sets and the number of reducts generated.

minimized. Results of this point of view were exactly the inverse of the first point of view. Model4 was the best model in both data sets followed by model2 of the heart disease data set and model3 of the HSV data set. The third priority was for model3 of the heart disease data set. While the third priority was given to model 2 of the HSV data set. The last choice was for model1 of both data sets: the HSV and the heart disease.

The relationships between different Data sets and factors of evaluations are shown in Fig. 3-5.

Figure 3 shows the relationship between the Data sets and the coverage percentages. Whereas Fig. 4 shows the relationship between the different Data sets and the number of rules generated from each one. The relationship between the four different types of Data sets and the number of reducts is shown in Fig. 5.

The pink line represents the heart disease dataset while the black line represents the HSV data set. The distance between lines or the distance between corresponding points on the line represents the difference of measurements values between the different factors; the coverage percentage as in Fig. 3, the number of rules as in Fig. 4 and the number of reducts as in Fig. 5.

CONCLUSION

Four different models of dealing with missing values were studied. When applying data mining to the real world, learning from the incomplete data is an inevitable situation. Trying to complete missing values is one obvious solution. However, techniques to guess the missing values must not introduce noise. Two experiments were designed to test the effect of different data replacement strategies on the coverage percentage, the number of rules and the number of reducts that were generated from each data set.

Coverage percentage was the best when model 3 was used to learn the HSV data set. It was the best when model4 was used to learn the heart disease data set. This leads to say that different data sets may need to use different models to get the best results. Filling in missing values is a complex strategy and needs a careful research. So, it is not a general case that one model can suite all data sets.

The experimental results suggested that the best model of generating minimum number of classification rules was the model of removing the examples that contain missing values regardless the size of the data set.

If the priority is given to the minimum number of reducts, then the best model was also for the model of removing the examples that contain missing values. And if the priority is given to the maximum number of reducts, then replacing the missing values with the global constant “missing” was the best chosen model.

This study concludes that the best model of dealing with missing values is a task-dependent. As the two case studies were considered in this paper, recommendations were concluded as earlier.

REFERENCES

1. Cios, K., W. Pedryez and R. Swiniarski, 1988. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, 1998.
2. Lavrajc, N., E. Keravnou and B. Zupan, 1997. Intelligent Data Analysis in Medicine and Pharmacology. Kluwer Academic Publishers.
3. Wolf, S., H. Oliver, S. Herbert and M. Michael, 2000. Intelligent data mining for medical quality management. Proc. Fifth Intl. Workshop on Intelligent data Analysis in Medicine and Pharmacology (IDAMP-2000), Berlin, Gennany.
4. Setiono, R., 2000. Generating concise and accurate classification rules for breast cancer diagnosis. Artif. Intell. Med., 3: 205-219.
5. Cheesemen, P. and J. Stutz, 1996. Bayesian classification (AutoClass): theory and results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthunsamy (Eds.), Advances in knowledge discovery and Data Mining. AAAI Press/MIT Press.
6. Grzymala-Busse, J., Z. Pawlak, R. Slowinski and W. Ziarko, 1999. Rough sets. Communications of the ACM, 11: 38.
7. Hassanien, A.E., 2003. Classification and feature selection of breast cancer data based on decision tree algorithm. Intl. J. Studies in Information and Control J., 1: 33-39.
8. Pawlak, Z., 1982. Rough Sets. Intl J. Computer and Information Sci., 11: 341-356.
9. Pawlak, Z., 1991. Rough Sets-Theoretical Aspect of Reasoning About Data. Kluwer Academic Publishers.
10. Pawlak, Z., J. Grzymala-Busse, R. Slowinski and W.Ziarko, 1995. Rough sets. Communications of the ACM, 11: 38, 89-95.
11. Al-shalabi, L., 2000. New learning models for generating classification rules based on rough set approach. Ph. D Thesis, Universiti Putra Malaysia.
12. Blum, R., 1982. Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project. Lecture Notes in Medical Informatics 19, New York: Spring-Verlag.
13. Cheesman, P., J. Kelly, M. Self, J. Stutz, W. Taylor and D. Freeman, 1988. AutoClass: A baysian classification system. Proc. Fifth Intl. Conf. on Machine Learning San Mateo, Calif.: Morgan Kaufmann, pp: 54-64.
14. Beard, P., 1989. Automated arbitrage expert system developed. It Outperformed S&P's 500 in First Quarter, AI Week 6, 13: 1-3.
15. Gaines, B.R. and M.L.G. Shaw, 1986. Introduction of inference rules for expert systems. Fuzzy Set and Syustems, 18: 315-328.
16. Quinlan, J.R., 1987. Generating production rules from decision trees. Proc. Tenth Intl. Joint Conf. Artificial Intelligence, pp: 304 –307, Menlo Park, Calif.
17. Clark, P. and T. Niblett, 1989. The CN2 induction algorithm. Machine Learning, 3: 261-283.
18. Al-shalabi, L., R. Mahmod., A. Abdulghani and M. Yazid, 1999. Data mining: An overview. World Engineering Congress (WEC'99), Kuala Lumpur, Malaysia.
19. Elmasri, R.. and B.N. Shamkant, 2003. Database Management Systems, The Benjamin, Redwood City, CA.
20. Pieter, A. and Z. Dolf, 1996. Data Mining. Harlow, England.
21. Agrawal, A. and R. Srikant, 2000. Privacy Preserving Data Mining. ACM SIGMOD.
22. Little, R.J.A. and D.B. Rubin, 1987. Statistical Analysis with Missing Data. John Wiley and Sons.
23. Quinlan, J.R., 1989. Unknown attribute values in induction. Proc. Sixth Intl. Workshop on Machine Learning, pp: 164-168.
24. White, A.P., 1987. Probabilistic Induction by Dynamic Path Generation in Vertual Trees. M.A. Bramer (Ed.), Research and Development in Expert Systems III, Cambridge University Press, pp: 35-46.
25. Liu, W.Z., A.P. White, S. G. Thompson and M.A. Bramer, 1997. Techniques for dealing with missing values in classification. Second Intl. Symp. Intelligent Data Analysis.
26. Han, J. and M. Kamber, 2001. Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco.
27. Kerdprasop, N., K. Kerdprasop, Y. Saiveaw and P. Pumrungreong, 2003. A comparative study of techniques to handle missing values in the classification task of data mining. 29th Congress on Science and Technology of Thailand, Khon Kaen University, Thailand.
28. Ragel, A. and B. Cremilleux, 1999. MVC: A preprocessing method to deal with missing values. Knowledge-Based Systems J., pp: 285-291.
29. Merz, C.J. and P.M. Murphy, 1996. UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html.