การศึกษาเปรียบเทียบเทคนิคการจัดการข้อมูลสูญหายในการทำเหมืองข้อมูลประเภทงานจำแนก*

A Comparative Study of Techniques to Handle Missing Values in the Classification Task of Data Mining

นิตยา เกิดประสพ, กิตติศักดิ์ เกิดประสพ, ยอด สายแวว, ปรีชา พุ่มรุ่งเรือง
Nittaya Kerdprasop , Kittisak Kerdprasop, Yawd Saiveaw, Preecha Pumrungreong
School of Computer Engineering, Suranaree University of Technology, 111 University Ave., Muang District, Nakorn Ratchasima 30000, Thailand; e-mail address: nittaya@ccs.sut.ac.th

บทคัดย่อ: งานวิจัยนี้เป็นการศึกษาเปรียบเทียบเทคนิคต่างๆ ที่ใช้จัดการกับกรณีข้อมูลบางส่วนสูญหายในกระบวนการวิเคราะห์ข้อมูลอัตโนมัติด้วยเทคนิคการทำเหมืองข้อมูลโดยเน้นเฉพาะงานจำแนกประเภทข้อมูล ข้อมูลที่ใช้ในการทดลองนำมาจากแหล่งข้อมูลมาตรฐานของมหาวิทยาลัยแคลิฟอร์เนียที่เออร์ไวน์ โดยเลือกมาทั้งข้อมูลประเภทจำนวนเลขและประเภทข้อความ ข้อมูลมาตรฐานถูกจำลองให้มีบางส่วนสูญหาย จากนั้นใช้เทคนิคที่แตกต่างกันสี่เทคนิคเพื่อเติมส่วนที่สูญหาย การทดสอบประสิทธิภาพของเทคนิคการเติมข้อมูลสูญหายใช้อัลกอริทึมการจำแนกประเภทข้อมูลด้วยวิธีเบย์ส์อย่างง่าย วิธีสร้างต้นไม้ตัดสินใจเชิงอุปนัย และวิธีการคำนวณระยะห่างของข้อมูล ผลการทดลองชี้ว่าถ้าข้อมูลเป็นประเภทจำนวนเลขการตัดทิ้งเรคคอร์ดที่มีข้อมูลสูญหายจะให้ประสิทธิภาพการจำแนกประเภทข้อมูลที่ดีกว่า ในขณะที่กรณีข้อมูลประเภทข้อความการเติมข้อมูลสูญหายด้วยสัญลักษณ์ “?” จะให้ประสิทธิภาพการจำแนกประเภทข้อมูลที่ดีกว่า

Abstract: We study and review the techniques for dealing with missing attribute values in data mining. Then, we conduct the experiments to observe the performance of classification algorithms on each strategy of missing-value substitution. The algorithms we used are naïve Bays, tree-based and instance-based classifiers. Four approaches of handling missing values are introduced to the numeric and nominal data sets taken from the UCI repository. The experimental results reveal the superior suggestive choice of ignoring numerical data instances with missing values, whereas replacing the unknown values with the symbol “?” produces a better classification results for the nominal data set.

Introduction: Data mining, also known as knowledge discovery in databases (KDD), is the process of extracting (or mining) useful knowledge from large volume of data. The different kinds of mined knowledge lead to the different tasks of data mining, for instance, concept description, association, classification, prediction, clustering, trend analysis. Among the diversity on mining tasks, classification is the most extensively studied one. Classification is the process of inducing a set of models from the training data. These models can describe and distinguish important data classes. The main purpose of inducing models is to use them predicting the class of the future data whose class label is unknown.

Obviously, data play an important role in the process of data mining. The quality of the collected data can directly improve the efficiency of the subsequent mining process. However, data collected in the real-world tend to be incomplete due to some values are missing. This might occur because the value is not relevant to a particular case, was not recorded when the data was collected, or is unspecified by users because of privacy concerns [1]. The incomplete data in which a large percentage of the entries are missing causes a problem since most data mining algorithms assume that the data is completely specified.

The problem of missing values has been investigated since the last two decades [3,6]. The simple solution is to discard the data instances with some missing values [8]. A more sophisticated solution is to try to determine these values [4]. However, techniques to guess the missing values must be efficient, otherwise the replacement may introduce noise.

In this paper, we empirically study the effect on the mining performance of different techniques for dealing with missing values. Several techniques to handle missing values have been discussed in the literature [2,4,6,7]. Some popular methods are as follows:

(1) Ignore and discard the tuples with missing values: This is a simple solution, but not very effective. However, it is recommended if the tuple contains several attributes with missing values.

* ได้รับทุนสนับสนุนการวิจัยจากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ รหัสโครงการ NT-B-06-4C-19-524

(2) Replace all missing attribute values with a global constant: Some examples of a global constant are "unknown", "missing", "- ∞".
(3) Replace the missing attribute values with its attribute mean: For example, suppose the average age of employees is 37. Use the value 37 to fill in the missing value for the attribute "age".
(4) Replace the missing attribute values with the attribute mean from the same class: Suppose the mining task is to classify the "top-employee" from the typical employee and some values for the attribute age are missing. The missing value may be filled with the average age among the top-employees.
(5) Replace the missing attribute values with the most probable value: The probable value may be guessed by using regression technique, a Bayesian inference, or decision-tree induction. However, the technique is appropriate for sparse missing values. Difficulties arise if the tuple contains more than one missing attribute values.

Methodology: The experiments have been designed to test the performance of model induction from the data sets using different approaches to fill in the missing values. The induction algorithms are also selected from the different paradigms; that is, the Bayesian, the decision-tree induction, and the instance-based (or nearest neighbor) algorithms. We run our experiments on the Weka system [9] and observe the accuracy of classification on each data set that contains a particular replacement of missing values. The two data sets – Glass identification and Zoo database – are taken from the UCI repository [5]. Both data sets contain no missing value. The Glass-identification data set represents the numeric data, whereas the Zoo data set represents the nominal data. Characteristics of these two data sets are summarized as follows:

data	number of instances	numeric attributes	nominal attributes	# class
Glass identification	214	9	0	7
Zoo database	101	1	15	7

To simulate the data set with missing values, some attribute values are removed from the original two data sets. The glass identification data set contains about 21% of missing values distributed equally among the nine attributes. The zoo data set contains about 32% of missing values, distributed equally as well.

The missing values are replaced with the four approaches:
(1) Replace with a symbol "?" (the symbol normally appears in the UCI data for the unknown values).
(2) Replace with a constant value "99.9" (represents the upper bound value) for the glass identification data set, and with a constant symbol "missing" for the zoo data set.
(3) Replace with the mean value of the attribute that has missing value for the glass identification data set, and with a majority value of that attribute for the zoo data set.
(4) Simply ignore the missing values by removing all instances containing missing values.

We compare the performance of naïve Bayes classification method empirically with the decision-tree induction (J48 [14] – the re-implementation of C4.5 [9]) and the nearest neighbor method. The classification accuracy is estimated using stratified ten-fold cross-validation technique.

Results and Discussion: Table 1 and 2 summarize how the three classification methods perform on data with missing values, which are handle using different approaches. Correct classification is reported as the accuracy of each classification method.

Table 1. Classification accuracy on numerical data with missing values

Data set	Accuracy		
	naïve Bayes	decision-tree induction	nearest neighbor
Glass[1]	47.19632 %	65.4206 %	70.0935 %
Glass_M1[2]	50 %	72.8972 %	61.215 %
Glass_M2[3]	11.215 %	68.6916 %	61.6822 %
Glass_M3[4]	47.1963 %	71.9629 %	65.8879 %
Glass_M4[5]	50.2959 %	72.1893 %	68.6391 %

[1] original data set – no missing value
[2] generate and replace missing values with "?"
[3] generate and replace missing values with a global constant numeric value
[4] generate and replace missing values with attribute's mean value
[5] remove instances that contain missing values

Table 2. Classification accuracy on nominal data with missing values

Data set	Accuracy		
	naïve Bayes	decision-tree induction	nearest neighbor
Zoo[1]	93.0693 %	92.0792 %	98.0198 %
Zoo_M1[2]	94.0594 %	91.0891 %	99.0099 %
Zoo_M2[3]	91.0891 %	85.1485 %	91.0891 %
Zoo_M3[4]	92.0792 %	87.1287 %	91.0891 %
Zoo_M4[5]	89.5522 %	88.0597 %	94.0299 %

[1] original data set – no missing value

[2] generate and replace missing values with "?"

[3] generate and replace missing values with a global constant

[4] generate and replace missing values with attribute's majority value

[5] remove instances that contain missing values

When introducing the missing values and then replacing them with a global maximal value, the classification performance of naïve Bayesian classifier degrades dramatically (comparing to other missing-value handling approaches – as shown in Table 1). The replaced constant value may interfere the computation of mean value on the course of deriving probability. Among the four approaches on dealing with missing values, the strategy of removing all instances with missing values gives the best prediction accuracy.

Note that replacing the missing values with the symbol "?" (representing unknown values) works equally well on the Naïve Bayesian and decision tree induction classifiers. The introduction of "?" to the unknown values is a common practice appeared in the UCI data sets. Therefore, many classification algorithms implement a module to handle the case of reading the unknown value "?", whereas the other kinds of replacement (such as a global constant) are treated as another attribute value. This can mislead the classification process.

For the classification on nominal data (Table 2), replacing the missing values with the unknown-value symbol "?" produces the best accuracy on all three classifiers. Removing instances with missing values can be the second choice for the nearest neighbor and decision-tree induction classifiers.

Conclusion: We study the different approaches of dealing with missing values. 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 bias the classification method or introduce noise. We thus design the experiments to test the effect of different data replacement strategies on the accuracy of classifying numeric and nominal data sets.

The experimental results either suggest replacing the missing values with the unknown-value symbol "?", or removing the instances with missing values. For the naïve Bayesian classifier, if the instances are so important that ignoring them may affect the result, replacing the missing values with a mean (for numeric data) or majority (for nominal data) value is another tempting strategy.

The understanding of classifier's behavior on different missing-value replacement strategies is useful for the decision of how to prepare data that best support the classifier.

References:

[1] A. Agrawal and R. Srikant, "Privacy preserving data mining", ACM SIGMOD, 2000.

[2] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2001.

[3] R.J.A. Little and D.B. Rubin, *Statistical Analysis with Missing Data*, John Wiley and Sons, 1987.

[4] W.Z. Liu, A.P. White, S.G. Thompson, and M.A. Bramer, "Techniques for dealing with missing values in classification", Second International Symposium on Intelligent Data Analysis, 1997.

[5] C.J. Merz and P.M. Murphy, UCI Repository of machine learning databases, 1996. [http://www.ics.uci.edu/ ~mlearn/MLRepository.html]

[6] J.R. Quinlan, "Unknown attribute values in induction", Proceedings of the Sixth International Workshop on Machine Learning, 1989, 164-168.

[7] A. Ragel and B. Cremilleux, "MVC: A preprocessing method to deal with missing values", Knowledge-Based Systems Journal, 1999, 285-291.

[8] A.P. White, "Probabilistic induction by dynamic path generation in virtual trees", M.A. Bramer (editor), *Research and Development in Expert Systems III*, Cambridge University Press, 1987, 35-46.

[9] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations,* Morgan Kaufmann, San Francisco, 2000. [software accessible via the URL http://www.cs. waikato.ac.nz/ml/weka]