Handling missing Data values in a Database Model using Random Forest

Abbas M. AL-Bakry
Collage of Computer Technology, Babylon University

Abstract

Missing values in a databases one of critical problem faced by the researchers in Data analysis and data mining. This work presents a suggested method for handling missing data values in data sets using Random Forest (RF) Technique. The use of RF present new principles to random splitting, it alters the tree growing process by narrowing its focus during split selection. For example, if the database contains numbers of columns usable for prediction, RF would begin randomly of selection number of variables and then chooses the splitter from the list of predictors. Using the suggested method we can get the actual values for the missing records entries and handling the uncertainty and outliers problem.

Key words: Random Forest, Missing value, Uncertainty.

ألخلاصة

تعتبر القيم المفقودة في قواعد البيانات واحده من التحديات ألمحرجه التي تواجه الباحثين في مجال تنقيب وتحليل البيانات. يعرض هذا العمل طريقه مقترحه لمعالجة قيم البيانات المفقودة باستخدام تقنية الغابات العشوائية (Random Forest). إن استخدام هذه ألطريقه وضح مبادئ جديدة للفصل العشوائي حيث يتعامل مع نمو الأشجار بتضييق البحث من خلال اختيار طريقة الفصل. وكمثال إذا كانت قاعدة البيانات تتكون من عدد من الأعمدة تستخدم للتنبؤ تبدأ هذه ألطريقه بالاختيار العشوائي لعدد من المتغيرات ومن ثم اختيار الفاصل الملائم من قائمة التنبؤات. إن استخدام هذه ألطريقه ألمقترحه يمكننا من ان نحصل على القيم الحقيقية لقيم القيود المفقودة ومعالجة مشاكل التشتت وعدم التأكيدية.

1- The Problem

Missing values means that there is no values in some record location make the data incomplete. The figure(1) present a record with two missing values labeled by ?

0.4	?	0.2	0.3	?	0.1

Figure (1) Data record with two missing values

There are several solutions suggested before for this problem:

1.1. The simplest solution for this problem is the reduction of the data set and elimination of all samples with missing values. This is possible when large data sets are available, and missing values occur only in a small percentage of samples.

1.2. Data miner and the domain expert, can manually examine samples that have no values and have a reasonable, probable, or expected value, based on a domain experience. This method is straightforward for small numbers of missing values and relatively small data sets. But, if there is no obvious or plausible value for each case, the miner is introducing noise into the data set by manually generating a value.

1.3 Automatic replacement of missing values with some constants:

- Replace all missing values with a single global constant.
- Replace a missing value with its mean feature.
- Replace a missing value with its mean feature for the given class.
- Replace a missing value with nearest neighborhood from top or bottom.

2. Random Forest

Random forest or random decision forest first proposed by Tin Kam Ho of Bell Labs in 1995. It is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees. The algorithm for inducing a random forest was developed by Leo Breiman[Brei01] and Adele Cutler, and "Random Forests" is their trademark

The method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Ho[HoTi95][HoTi98] and Amit and Geman[Amit97] in order to construct a collection of decision trees with controlled variation.

3- Random Forest Development

Wen Xiong et al. [WenX09] proposed a hybrid approach based on improved self adaptive ant colony optimization (ACO) and random forest(RF) to select feature from micro array gene expression data. It can capture important feature by pre selection and attain near-optimum or optimum by confining the size of ant’s solution to accelerate convergence of ant colony.

Erica Craig and Falk Huettmann., 2009[Erica09] describe the use of fast, data-mining algorithms such as TreeNet and Random Forests to identify ecologically meaningful patterns and relationships in subsets of data that carry various degrees of outliers and uncertainty.

In this work, we examine how can you develop a Random Forest Data Mining algorithm (RFDM) to pre-process the huge database that suffer from missing values problem, Random Forest(RF) [Brei01] is a collection of decision trees grown and combined using the computer code. RF models are often considerably more accurate than a single tree, accuracy achieved is often competitive with the best alternative methods. Resistance to over training (growing a large number of RF trees does not create a risk of over fitting and each tree is a completely independent random experiment). Speed (trees are grown at high speed because few variables are in use at any one time) also be used to determine how to do the best for data filter before analysis[Frei02].

4. Handling Missing values using Random Forest

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. RFs introduce a new principle to random splitting. It alters the tree growing process by narrowing its focus during split selection. For example, if the database contains 100 columns usable for prediction, RFs would begin by randomly selection the 7 variables and then selection the splitter from among that list of the 7 predictors. Once the data has been split into two subsets the process would be repeated by splitting each of the two subsets partly at random. To split one of the two subsets RFs we would select the different set of the 7 predictors at random, and would use the best of these to effect the further split. Although in RFs tree growing we always split data using the best splitter available, we first randomly limit the list of available splitters to a small number. This means that the selected splitter is the best only in the limited sense of being best in the randomly selected list.

The tree is constructed using the following process steps:

Step1: Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
Step2: We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.
Step3: Choose training set for this tree by choosing *N* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
Step4: For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.
Step5: Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

The new development of RF which differs from previous used that the prior models tend to combine more effectively into high performance aggregate models. The RFs is different in that it introduces an entirely new way of generating individual component models and do not combine more effectively into high performance aggregate models. There are two reasons for this: ***First***, combining or averaging of models does not accomplish significant accuracy improvement if the individual models are too similar to each other. The RFs have two-part randomization process makes the individual models quite different from each other. ***Second*** the other models is slow learning while in RFs because the trees are grown independently of each other with no single tree depending on any other tree for its generation, adding trees does not create a risk of over fitting. RFs can be expanded indefinitely without a performance loss. If an important pattern genuinely exists in the data portions of it will be uncovered by different trees and genuine patterns will be detected repeatedly by different trees. Figure (2) present the visual process for handling the missing values.

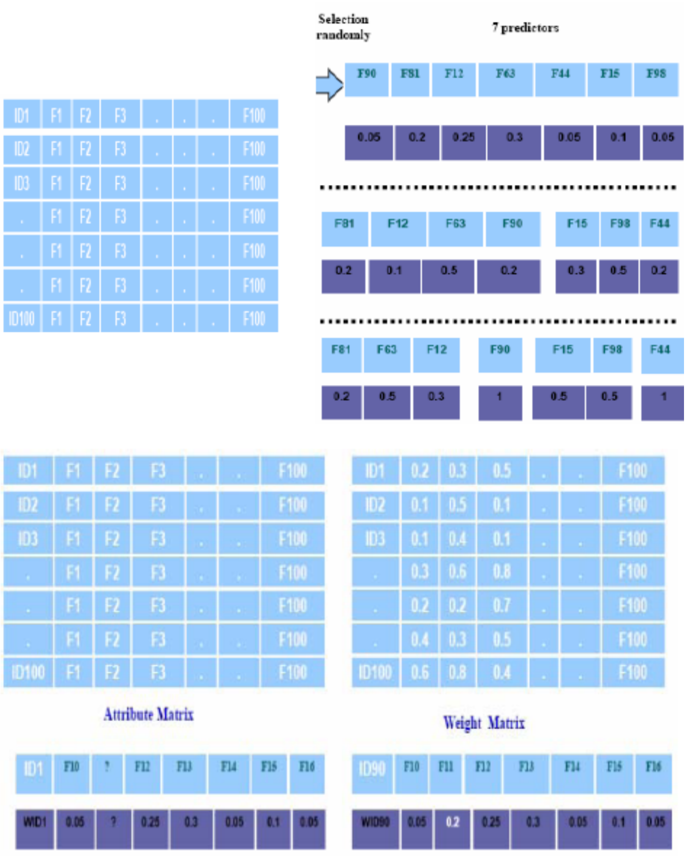
Selection randomly
7 predictors
F90 F81 F12 F63 F44 F15 F98
0.05 0.2 0.25 0.3 0.05 0.1 0.05
F81 F12 F63 F90 F15 F98 F44
0.2 0.1 0.5 0.2 0.3 0.5 0.2
F81 F63 F12 F90 F15 F98 F44
0.2 0.5 0.3 1 0.5 0.5 1
ID1 F1 F2 F3 . . . F100
ID100 F1 F2 F3 . . . F100
Attribute Matrix
Weight Matrix
ID1 F10 ? F12 F13 F14 F15 F16
WID1 0.05 ? 0.25 0.3 0.05 0.1 0.05
ID90 F10 F11 F12 F13 F14 F15 F16
WID90 0.05 0.2 0.25 0.3 0.05 0.1 0.05

Figure (2) Handling Missing Values using RF

5. Conclusions

The suggested method present that the mapping of all the weight values of the record has a missing values with all features of the record has weight values similar of that record. The mapping perform on the down level of splitter to reduce the time and number of comparing among weight of features. Using this method we can find the actual values of the missing values, and handle the uncertainty and outliers problems.

5. References

[Brei01] Breiman, Leo , "Random Forests". *Machine Learning* **45** (1): 5–32.doi:10.1023/A:1010933404324., 2001.

[HoTi95] Ho, Tin , "Random Decision Forest". 3rd Int'l Conf. on Document Analysis and Recognition. pp. 278–282. http://cm.bell-labs.com/cm/cs/who/tkh/papers/odt.pdf, 1995 .

[HoTi98] Ho, Tin , "The Random Subspace Method for Constructing Decision Forests". *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (8): 832–844. doi:10.1109/34.709601. http://cm.bell-labs.com/cm/cs/who/tkh/papers/df.pdf, 1998.

[Amit97] Amit, Y.; Geman, D. (1997). "Shape quantization and recognition with randomized trees". *Neural Computation* **9** (7): 1545–1588. doi:10.1162/neco.1997.9.7.1545.

[WenX09] Wen Xiong, Cong Wang, "A Hybrid Improved Ant Colony Optimization and Random Forests Feature Selection Method for56 v/' Microarray Data". IEEE

Computer Society , Fifth International Joint Conference on INC, IMS and IDC, 2009

[Eric09] Erica C. And Falk H., " Using “Blackbox” Algorithms Such as TreeNet and Random Forests for Data-Mining and for Finding Meaningful Patterns, Relationships, and Outliers in Complex Ecological Data", Information science reference, Hershey • New York, 2009

[Frei02]] Freitas.A. Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, 2002.