Data density assessment using classification techniques

Sergio Pío Alvarez[1], Adriana Marotta[1], and Libertad Tansini[1]

[1]Universidad de la República, Montevideo, Uruguay
`{sergiop,amarotta,libertad}@fing.edu.uy`

Abstract. There is general agreement among data quality researchers in that completeness is one of the most important data quality dimensions. In particular, data density can be a crucial factor in data processing and decision making tasks. Most techniques for data quality evaluation regarding density are limited to counting null values. However, density is not only about null values but also about not-null values when there should be null values, as the latter degrades the quality of data too. Besides, the existence of null values not necessarily implies a data quality problem. In this work we present a technique based on the application of data mining techniques for data quality assessment. Our proposal consists in creating a classification model from available data having null and not null values and then using that model to assess if a particular attribute of a record should or should not have a null value. This technique allows us to evaluate if a null value is an error, if it is correct, or if it is uncertain, and also we can evaluate if a not-null value is acceptable, is an error (it should be null) or is uncertain.

Keywords: Data Quality, Density, Null Values, Data Mining, Classification.

1 Introduction

The importance of *Data Quality* (DQ) in all kind of information systems is widely recognized. If data do not have the appropriate quality level the main business processes could be affected and lead to wrong decisions. DQ is a multifaceted concept, since it is defined regarding a set of dimensions [1]. There is general agreement among DQ researchers and practitioners in that *completeness* is one of the most important DQ dimensions [2]. Although there are different conceptions about what completeness means [3], it usually involves two factors: *coverage* and *density*. If the real word is composed by *entities*, each of them described by *attributes*, then coverage is about how many entities are represented in the dataset, while density is about how many attributes are known for each entity.

Once the relevant attributes for an entity are selected, density is *usually* regarded as not having *missing values* for them. In relational databases a missing value is represented with the special value '*null*'. Techniques for density assessment have been traditionally limited to counting not-null values, assuming that missing values imply data quality problems. However, it is important to understand why a value is missing for an attribute of an entity: it could be that does not apply a value for the entity or it could be really missing. We claim that, as well as a null value *could* be a density problem, a not-null value where should be a null one *is* also a density problem [4].

The purpose of *Data Mining* (DM) is to discover hidden knowledge within large amounts of data. DM spans many techniques, being classification, clustering and associative analysis the most common ones. *Classification* is a technique aimed at assigning entities into one of a set of predefined categories called *classes*. Classification algorithms build a model from a set of entities previously classified, and then use the model to classify new entities for which the class is not known.

The goal of this work is to propose a technique for density assessment using DM classification concepts.

The main contribution of this work is to put into discussion the idea that null values should not be taken always as density problems, and that not-null values could be density problems.

The rest of the document is organized as follows: in Section 2 we present related work, in Section 3 we present the proposal for assessing data density, in Section 4 we summarize some experiments, and in Section 5 we show the conclusions.

2 Related Work

Missing values are usually classified in three categories [5]: Missing Completely At Random (MCAR), which means that there is no pattern that explains why values are missing; Missing At Random (MAR), which means that a pattern relating missing values for an attribute to some other attributes can be found; and Missing Not At Random (MNAR), which means that missing values for an attribute are related to the attribute itself but not to other attributes. At first glance MNAR is similar to MCAR because looking only at data it can not be told which case it is; moreover, missing data are almost never MCAR [6]. There are many techniques that tackle the problem of density assessment, most of them focused on the MAR and MCAR scenarios [7], but most of them try to solve the *“null-value problem”*, assuming that if there is a null value then there is a problem that must be resolved, usually by means of value imputation or data deletion ([4][6][8]).

Data Quality Mining (DQM) is defined as the application of DM techniques to measure and improve DQ. The underlying concept is that modelling of different behaviours within data can not only be used to understand the data but also to detect anomalies, hence pointing out possible quality problems [9,10,11,12,13,14,15]. Roughly, DQM consists of two phases: in the first phase a model capturing the characteristics of data is induced from a training dataset, and in the second phase the model is used to assess the quality of another dataset to detect deviations. Data which deviates from the model are candidates to show some kind of data quality problem.

3 Data Density Assessment through Classification Techniques

As we pointed in the previous section there are many techniques for solving the problem of null values. We propose to take a step back and evaluate first if a null-value is really a density problem, as well as if a not-null value could also be a density problem. Our method takes a dataset and estimates the probability of each value to be correct (whether null or not). For this task we use a classification technique that marks each value as 'probably null', 'probably not null' or 'uncertain' based on other values in the dataset.

Let *D* be a dataset and *A* an attribute (not a key). For each record *r* of *D* it can happen that the value for *A* is either *null* or *not-null*. The algorithm is as follows:

1. Drop all keys from the dataset. This step is important because most classification algorithms will generate single classification rules in the form *'IF key=X then A=[null|not-null]'* for each value *X* of the key, and those rules are trivial and possibly wrong.
2. Among the remaining attributes, those that should be used to assess the attribute *A* must be identified; this task could be challenging as it constitutes a whole working area named *feature learning* [16].
3. Discretize non discrete values following the next guidelines: replace text values with '*SOMEVALUE*' as it does not matter which value the attribute has but it is only important if it is null or not, and for all other non-discrete values can be used as-is but most classification algorithms work better when all attributes are discrete. If the selected classification algorithm requires attributes to be discrete then some discretization technique should be applied [17].
4. Replace the value of the attribute *A* for each record as follow: if the value of the attribute *A* is not-null then replace that value by the text '*NOTNULL*' otherwise replace the null value by the text '*NULL*'. This defines two classes: *null* and *notnull*, and each record will have assigned one of them.
5. Apply some classification algorithm to build a classification model *M* using the attribute *A* as the class. Any classification technique can be used, but decision-tree based algorithms are easier to interpret. Usually the model is built using a clean dataset and then is applied to another dataset to verify how well the model predicts the class. In our case both, the training dataset and the test dataset, are the whole input dataset because we assume that density problems are exceptions and so are lost in the wideness while the model is being built.
6. Apply the classification model built in the previous step to the dataset. For each record the classification model will output a prediction (either '*NULL*' or '*NOTNULL*') and a decimal value in the range [0,1] which is the confidence of the prediction.
7. Evaluate each record from the dataset again as follows:
   - If the classification confidence for the record is above a predefined threshold then the assigned class is accepted as correct, leading to two scenarios:
     - If the assigned class matches the value of the attribute for the record then the record does not present a density problem.
     - Otherwise it can be taken for sure that the record is wrong and there is a density problem.

- Otherwise is an uncertain scenario because it can not be told if there should or should not be a null or not-null value for the attribute for the record, and it is a candidate for manual revision.

Although the threshold is defined beforehand it can be adjusted based on the results of the evaluation of the whole dataset. We usually set the threshold to 0.66 and if there are too many records that fall above the threshold then it can be increased while if there are too few then it can be decreased.

4 Experiments

We applied the proposed method to a laboratory case, with an extensive combination of attributes: attributes functionally dependant on others, attributes not functionally dependants but related to others, and attributes not related in any form to other attributes. Some attributes had null values, some of which were real density problems but others were not as the null value was the correct one (for example, non deceased people should have a null value for the date-of-decease attribute), and conversely some other attributes had not null values where it should have null (there were alive people with not null date-of-decease values), which were also data density problems.

We used the Weka software [18,19], and chose the Random Tree algorithm because with the default configuration it produces good decision trees.

In some cases we found that with a threshold of 0.66 then 2/3 of the records were classified with a high confidence, 1/4 of which were assigned a class different from the real value of the assessed attribute. This means that at least 1/6 of the whole dataset presented a density problem regarding the assessed attribute (there are null values where there should not be or conversely). On the other hand, 3/4 of the records classified with confidence above the threshold were assigned the class matching the assessed attribute, so is almost certain that those records did not present any kind of density problem regarding the attribute. There was 1/3 of the records for which the algorithm could not determine if the assessed attribute should have a null value or not, these record are candidates for further inspection.

5 Conclusions

We believe that data density should not only be fighting null values, since the presence of not-null values where there should be null values is also a data quality problem. Moreover, the presence of a null value should not be considered a density problem when there should be a null value in that place. In this sense we propose a simple approach for data density assessment using classification techniques. It is oriented to evaluating when null values and not-null values could imply data quality problems. The presented method helps in two complementary ways to achieve a high density database: it can detect when a null-value or not-null-value may be wrong, and the model built can be used to prevent the degradation of the dataset quality by checking data before inserting it into the dataset.

6 References

1. Batini, C., Scannapieco, M.: Data and Information Quality, Dimensions, Principles and Techniques. Springer International Publishing (2016).
2. Mendes Sampaio, S. de F., Dong, C., Sampaio, P.: DQ2S, A framework for data qualityaware information management. Exp. Systems with Applications, Vol. 42, No. 21, (2015).
3. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Computing Surveys, Vol. 41, No. 3, Article 16 (2009).
4. Horton, N.J., Kleinman, K.P.: Much Ado About Nothing: A Comparison of Missing Data Methods and Software to Fit Incomplete Data Regression Models. The American Statistician, Vol. 61, No. 1 (2007).
5. Little, R., Rubin, D.: Statistical Analysis with Missing Data. John Wiley & Sons, New York (1987).
6. Newman, D.A.: Missing Data: Five Practical Guidelines. Organizational Research Methods, Vol. 17 No. 4 (2014).
7. Charini Tremblay, M., Dutta, K., VanderMeer, D.: Using Data Mining Techniques to Discover Bias Patterns in Missing Data. ACM Journal of Data Information Quality, Vol. 2, No. 1 (2010).
8. Sessions, V., Gieves, J., Perrine, S.: A Technique for Incorporating Data Missing Not at Random (MNAR) into Bayesian Networks. Int. Conf. on Information Quality (2016).
9. Hipp, J., Güntzer, U., Grimmer, U.: Data quality mining, making a virtue of necessity. Proceedings of the 6th ACM SIGMOD workshop on research issues in data mining and knowledge discovery (2001).
10. Grüning, F.: Data quality mining: employing classifiers for assuring consistent datasets. Proceedings of the 3rd International ICSC Symposium (2007).
11. Grimmer, U., Hinrichs, H.: A methodological approach to data quality management supported by data mining. Proc. of Int. Conference on Information Quality (2001).
12. Farzi, S., Baraani Dastjerdi, A.: Data quality measurement using data mining. International Journal of Computer Theory and Engineering, Vol. 2, No. 1 (2010).
13. Luebbers, D., Grimmer, U., Jarke, M.: Systematic development of data mining-based data quality tools. Proceedings of the 29th VLDB Conference (2003).
14. Vázquez Soler, S., Yankelevich, D.: Quality mining: a data mining based method for data quality evaluation. International Conference on Information Quality (2003).
15. Dasu, T., Johnson, T.: Hunting of the snark: finding data glitches using data mining methods. Proceedings of Int. Conference on Information Quality (1999).
16. Bengio, Y., Courville, A., Vincent, P.: Unsupervised Feature Learning and Deep Learning: A Review and New Perspectives. arXiv:1206.5538v3 (2012).
17. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised Discretization of Continuos Features. Machine Learning: Proc. of the 12th. International Conference (1995).
18. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update - SIGKDD Explorations, Vol. 11, No. 1 (2009).
19. Machine Learning Group at the University of Waikato. Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/~ml/weka/ (2015) (last access: 2017/03/24)