DATA MINING, DIRTY DATA, AND COSTS

(Research-in-Progress)

Leo Pipino
University of Massachusetts Lowell
Leo_Pipino@UML.edu

David Kopcso
Babson College
Kopcso@Babson.edu

Abstract: A series of simulations examining the performance of four data mining algorithms in the face of missing data were conducted. The four algorithms were: feed-forward neural networks, logistic regression, C5.0 algorithm, and the Apriori algorithm. A credit card screening data set was used. The original data set was altered by introducing missing data, at increasingly greater levels, and the performance of the algorithms was measured.

Key Words: Missing data, feed-forward neural networks, logistic regression, C5.0 algorithm, Apriori algorithm

INTRODUCTION

In any data mining initiative a multiplicity of costs are incurred. The relationship of these costs to the results yielded by the associated data mining model have not been widely studied. The bulk of the literature deals with the algorithms used to perform data mining and with the process of data mining [1, 2, 7]. Organizations are increasingly creating data warehouses which are then used, among other things, for data mining. A closer look at costs and their relationship to benefits provided by data mining is in order.

In a data warehouse environment [5], the Extract, Transform, and Load (ETL) functions used in the data staging area would handle the cleaning and integrating of the data. Typically, however, the data resident in the data warehouse must still undergo additional preprocessing to transform it into a data set whose structure is suitable for efficient data mining. It has been estimated by some that 50% to as much as 80% of the time and effort spent on data mining is spent on preprocessing tasks. This percentage of time and effort translates into a large percentage of the costs of data mining.

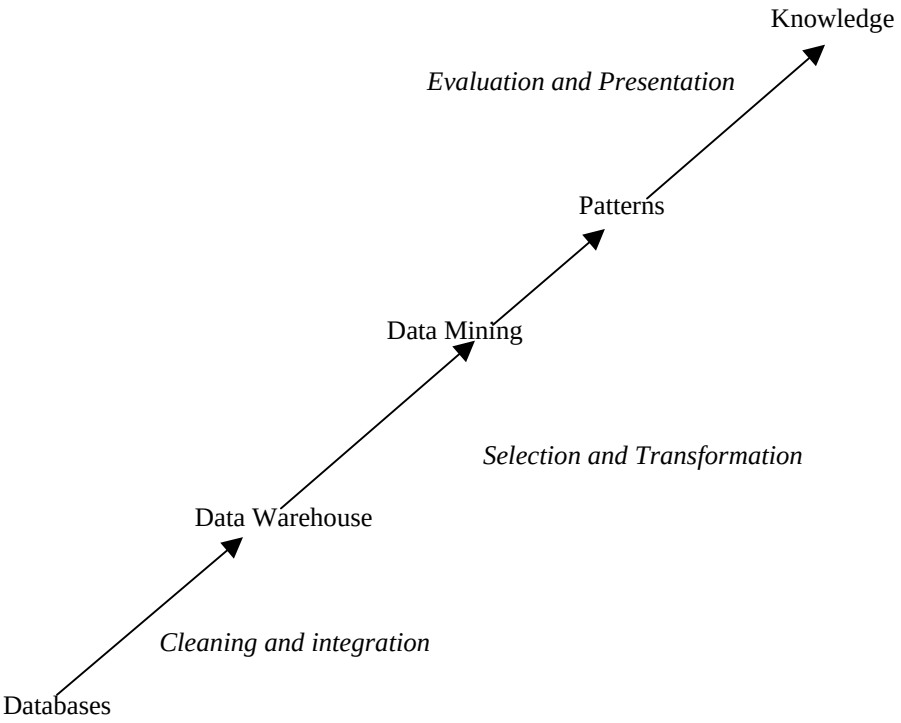
The costs incurred at different stages of the data mining process (See Figure 1) include:

(1) Costs of cleaning the data from the different sources
(2) Costs of integrating the data
(3) Costs of additional preprocessing and transforming the integrated data set into a usable data set for data mining
(4) Costs of building and running the data mining model

It should be noted that these costs are not incurred in a linear fashion, but, rather iteratively as the mining process proceeds. Not included in the above list is the cost of collecting the raw data. We assume that the collection cost would be incurred whether or not the organization undertook data mining initiatives.

A number of questions then present themselves. First, at which point do the costs of cleaning the data outweigh the benefits or returns from using the highest levels of clean, quality data? To properly study this, a second question directly follows from the first. Specifically, at what level of dirty data does the data mining technique begin to break down? Alternatively, one could ask: What is the rate of decline of a technique's accuracy as the data becomes increasingly dirty? If a clear break point emerges, an informed assessment of the benefits versus costs of improving the data can be performed. Our research is intended to address these questions. For this paper, which constitutes the initial phase of our research, we focus on the second question. We note that some work in the area of the effects of dirty data was reported in the ICIQ 2003 conference [3, 4].

Figure 1



QUALITY OF DATA, DATA MINING TECHNIQUES, AND COST

We reemphasize that the research-in-progress reported in this paper is directed at the performance of various data mining techniques in the face of dirty data. For completeness, however, before describing the design of the simulation experiments we briefly address the costs involved and their relationship to the simulations. Han and Kamber [2] list the following functions under data cleansing: (a) Missing values, (b) Noisy data, and (c) Inconsistent data. These same authors list the following operations under data transformation: (a) Smoothing, (b) Aggregation, (c) Generalization, (d) Normalization, and (e) Attribute construction.

Performing each of the above operations incurs costs. In addition, other costs incurred in any data mining initiative include typical fixed costs such as those of ETL software and Data Mining software. As a first-order approximation, we can represent the total costs of the effort as a set of additive functions:

Cost of Data Mining Initiative = f_1 (degree of clean –missing data type)
+ f_2 (degree of clean – noisy data type)
+ f_3 (degree of clean – inconsistent data type)
+ f_4 (data transformation necessary)
+ f_5 (typical fixed costs)

The above rudimentary model assumes that no interaction effects among variables exist. For example, one might find interactions between the noise in the data and some inconsistencies. Also, some cleansing actions may have an effect or preclude the need to perform some data transformation operations.

The tradeoff would be whether the benefits accrued from the data mining project outweigh the costs incurred. One cannot easily assess benefits. These will depend on the objective of the project, how the results will be used, and how they will affect decisions which, in turn, result in benefits. We can state, however, that by obtaining a reasonable assessments of the costs and their relationship to goodness of results, a more informed cost/benefit tradeoff can be made.

Keep in mind that the cost model shown above is intended to illustrate the relationship between costs and the experiments to be reported in this paper. An actual cost model, on which we have begun exploratory work, would be more complex and account for potential interactions. For purposes of our research, only functions f_1, f_2, and f_3 are of immediate interest. We are interested in the effects of poor quality data on performance. If the format of the data requires transformation to be compatible with a particular data mining technique, we assume that this will be necessary whether the data is of poor or high quality. This is not to say that such costs are not important, but rather, that we assume the cost of data transformation is essentially fixed and remains constant. The same argument applies to the typical fixed costs. They will be incurred regardless of the quality of the data.

The research to be reported in this paper, then, is directly related to costs. As one attempts to achieve higher and higher levels of quality for variables of functions f_1, f_2, and f_3, the costs of achieving higher and higher levels increases. It is reasonable to expect that f_1, f_2, and f_3 are nonlinear, possibly exponential with increasing rate of change for cost, that is, $\partial f_1 / \partial(\text{clean} - \text{missing}) > 0$ and monotonically increasing, $\partial f_2 / \partial(\text{clean} - \text{noisy}) > 0$ and monotonically increasing, and $\partial f_3 / \partial(\text{clean} - \text{inconsistent}) > 0$ and monotonically increasing. Our purposes in this initial set of experiments (simulations) is not to explicitly capture these functions. Rather, we introduce these constructs in order to show the link between the results of our experiments (simulations) and costs. The estimation of these functions will be the subject of future work.

OBJECTIVE OF RESEARCH

The goal of this research is to examine the effect of corrupted data on the performance of specific data mining techniques. The intent is to consider all major data mining techniques and to make use of multiple data sets. In the long-term we intend to examine the cost implications of these effects.

Overview of Initial Experiment

In the initial phase, however, we concentrate on four major data mining techniques: feed forward neural networks, logistic regression, induction of decision tree using Quinlan's C5.0 [7] algorithm, and association rules generation (Apriori algorithm). All were applied to a credit card screening data set provided by Quinlan [9] and available in [8] or via web links at [9]. Each technique was run against this data set to establish a benchmark. The data set was then gradually corrupted. In the initial phase reported in this paper, we have focused on missing data only. Each data mining technique was tested against the corrupted data sets.

The Data Set

Although well-known and often cited, the credit card screening data set used is not large or "industrial strength". The number of cases is 690 and all attribute names and values have been changed by the source to meaningless symbols to protect confidentiality of the data. This data set, however, is interesting because it contains a mix of continuous and nominal (categorical) attribute data as well as some missing values. It was used to provide an initial proof-of-concept and to detect and correct any deficiencies in the design of the simulations. An example of one row or string of data is given below:

```
b,30.83,0,u,g,w,v,1.25,t,t,01,f,g,00202,0,+
```

Note that the first 15 elements of the string represent 15 attributes. The sixteenth element (+ or -) represents one of two classes into which each string (individual) will fall.

CURRENT EXPERIMENTS AND RESULTS

We follow Quinlan [6] in dividing the data set into three parts: one of 460 records for building our models and two of 115 each for testing our models. We refer to the later two data sets as 115A and 115B. It should be noted that construction of each part was done randomly and is not necessarily the same collection of data items as in Quinlan [6]. Note also that in [6] Quinlan addresses the topic of decision tree simplification. In so doing, one of the simulation runs was performed using the credit card screening data. This gives us one "ballpark" reference point to use to verify the reasonableness of our simulation using the C5.0 algorithm.

The Simulations

We begin our examination of f_1 by instantiating our four model types (Neural Networks, Logistic Regression, C5.0, and Apriori) built on the 460 build data set. We then corrupt the 460 build data set by randomly selecting a record and then randomly omitting data for one of its 15 attributes. We do this for 5%, 10%, 15%, 20 %, 25%, 30%, 35%, and 40% of the data. Each data of the nine sets is then used to develop corresponding versions of our four models. The resultant 36 models are evaluated on each of the two test data sets. The results are given in the Table below. The benchmark runs (0% error introduced) are shown in row 1 of the table. All simulations were conducted using the Clementine 8.5 software from SPSS.

Table of Correct Classification (%)

Missing	Neural Network		Logistic Regr.		C5.0		Apriori	
	NN115	NN115	LR115	LR115	C5115	C5115		
%	A	B	A	B	A	B	AP115A	AP115B
0	80.00	85.22	74.78	82.61	80.00	92.17	20.87	18.26
5	82.61	88.70	73.04	81.47	82.61	89.57	20.87	18.26
10	79.13	87.83	72.17	80.87	86.09	90.43	20.87	18.26
15	84.35	87.83	73.91	80.00	86.09	90.43	20.87	18.26
20	80.00	87.83	73.04	80.00	86.09	90.43	20.87	18.26
25	82.61	86.96	73.04	80.87	86.09	90.43	20.87	18.26
30	81.74	90.43	73.91	81.74	86.09	90.43	20.87	18.26
35	77.39	84.35	73.91	80.87	86.09	90.43	20.87	18.26
40	77.39	87.83	74.81	82.61	86.09	90.43	20.87	18.26

Remarks

The results of our runs with C5.0 are comparable, although not identical, to the results reported by Quinlan [6] for his average error rates (with and without pruning the decision trees), namely error rates of 20% and 7.83% (our rates) versus rates ranging form 15.2% to 21% (Quinlan).

Recall that our purpose was not to compare algorithms. Clearly, the Apriori algorithm performs poorly. A requisite of any data mining exercise is to apply the correct model to the problem/data at hand. The credit screening data set was intended for developing a model to classify subjects into one of two classes. One would not choose an algorithm intended for market basket analysis for this problem. That having been said, observe that the algorithm stabilizes and performance does not deteriorate as the percentage of missing data is increased.

Note that the models generated perform better on the B test set than the A test set. We have no explanation for this. What is more intriguing, however, is the performance of the algorithms as more missing data is introduced. There is no significant deterioration in performance. Indeed, in a number of instances, the performance has improved as missing data has increased. We had anticipated that a clear breakpoint would be detected at which point the performance would degrade sharply and suddenly. This did not occur. If not a sharp deterioration in performance, one might expect a gradual deterioration of performance as more errors introduced. This also did not occur.

Again, we have no explanation at this time for this phenomenon. It may very well be that the data set is sufficiently robust to withstand a large amount of missing data and we must introduce much more than 40% missing for performance to deteriorate. Also, it is possible that in randomly introducing the missing data, the equivalent of reducing the data used, we eliminated data that otherwise might represent noisy data within the data set.

This research effort is ongoing and additional simulations and analyses are in progress.

REFERENCES

[1] Chen, Z., *Data Mining and Uncertain Reasoning*, John Wiley & Sons, New York, 2001.
[2] Han, J. and Kamber, M., *Data Mining Concepts and Techniques*, Morgan Kaufmann Publ., San Francisco, 2001.
[3] Haughton, D., Robbert, M., Seene, L., and Gada, V., "Effect of Dirty Data on Analysis Results", *Proceedings of the 8th International Conference on Information Quality*, Cambridge, MA, 2003, 64-79.
[4] Laruia, E. and Tayi, G., "A Comparative Study of Data Mining Algorithms for Network Intrusion Detection in the Presence of Poor Quality Data", *Proceedings of the 8th International Conference on Information Quality*, Cambridge, MA, 2003, 190-201.
[5] Ponniah, P., *Data Warehousing Fundamentals*, John Wiley & Sons, New York, 2001.
[6] Quinlan, J.R., "Simplifying decision trees", *International Journal of Man-Machine Studies*, 27, Dec 1987, pp. 221-234.
[7] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, Oct 1992.
[8] Roiger R. and Geatz, M., *Data Mining*, Addison-Wesley, Boston, MA 2003.
[9] http://www.ics.uci.edu/~mlearn/MLSummary.html