Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2004

An Approach to Improve Classification Accuracy in Very Large Datasets

Marilyn Kletke
Oklahoma State University

Dursun Delen
Oklahoma State University

Jin-hwa Kim
Sogang University

An Approach to Improve Classification Accuracy in Very Large Datasets

Marilyn G. Kletke
Oklahoma State University
mkletke@okstate.edu

Dursun Delen
Oklahoma State University
Delen@okstate.edu

Jin-hwa Kim
Sogang University, Korea
jinhwakim@mail.sogang.ac.kr

ABSTRACT

In this paper we present a study that suggests a two-step approach, called the Iterative Refinement Algorithm (IRA), for improving the classification accuracy of inductive learning algorithms applied to very large datasets. We present the preliminary test results for IRA compared to other prediction methods including logistic regression, discriminant analysis, neural networks, C5, CART, and CHAID on a census dataset of approximately five million records. We offer IRA with the belief that it is an incremental step towards overcoming the limitations of current data mining tools as they are applied to today's massive datasets.

Keywords

Very Large Databases, Data Mining, Rule Induction, Classification Algorithm, Decision Tree, Knowledge Refinement.

INTRODUCTION

Today's businesses are faced with the problem of processing terabyte-class databases in order to extract much needed knowledge for becoming and remaining competitive in their market segments. The problems of successfully applying many current data mining algorithms to very large databases, called the scaling problem (Chan and Stolfo, 1995), have led to the recent steam of research work that focuses on scalability of data mining algorithms (Bradley, Ramakrishnan, and Srikant, 2002; Domingo, Gavalda, and Watanabe, 2002; Ganti, Gehrke, and Ramakrishnan, 1999).

The unrelenting growth in data needing to be accurately analyzed in today's organizations and the lack of scalability of many existing classification algorithms have led to the work reported in this study, which presents the approach and progress on the construction of a new scaleable, weighted classification algorithm and methodology, referred to as the Iterative Refinement Algorithm (IRA).

BACKGROUND AND LITERATURE REVIEW

The term "very large dataset" is increasingly being seen in the data mining literature (see, for example, Chan and Stolfo, 1995; Ganti et al., 1999; Heinrichs and Lim, 2003). Traditional data mining tools and techniques will become increasingly challenged as more and more companies grow increasingly massive datasets.

The data-mining tool addressed by this study is classification, often implemented through a rule induction system. Rule induction systems can be modeled using a decision tree algorithm (Quinlan, 1991). Traditional decision-tree algorithms have been "main-memory" algorithms in which all the data is stored in main memory so that it can be processed efficiently. C4.5, the implementation of Quinlan's (1993) seminal work in machine learning, provides good classification accuracy (Ruggieri, 2002) and is one of the fastest tree-construction main-memory algorithms. As data sources increase in size, however, decision trees will at some point exceed the storage capacity of main memory.

With massive datasets, random sampling becomes a practical way to begin classification efforts. However, the domain knowledge base constructed from sample subsets of the data consists of a less than perfect set of rules because a sample subset from a very large dataset, no matter how well chosen, can't completely represent the whole dataset.. Many researchers improve the quality of this set of rules is by a process called knowledge refinement, or rule refinement (see, for example, SEEK (Politakakis and Weiss, 1984), KBANN (Shavlik and Towell, 1989), and KREFS (Park, Kim, Shaw, and Piramuth, 1997). Providing domain knowledge for these kinds of systems is a highly time-consuming and expensive task. Furthermore, complete coverage of domain knowledge cannot even then be guaranteed.

To address the problems in existing approaches with refinement, we suggest an alternative decision-tree approach in which we use random sampling without replacement to generate weighted rules in two different steps: a construction step and a refinement step. According to Hand, Mannila, and Smyth (2001), a random selection strategy employed in the training dataset can be used satisfactorily for statistical inference and that in very large datasets there will be very little difference between sampling with replacement and sampling without replacement. Since we incorporate rule weights into the domain knowledge base, and since rule redundancy is a measure used in this study for weighting, we sample without replacement.

When sampling is used in classification methods, there are various ways of assessing weights. Examples include a simple weighted average (Quinlan, 1986); a measure of marginal contribution to information about classification (gain ratio) (Quinlan, 1986); accuracy in a validation set (Chan and Stolfo, 1995); various weighed majority algorithms (Littlestone and Warmuth, 1994). It is documented that various weighting schemes for rules based on parameters such as rule accuracy, coverage, attributes and redundancy have led to improved accuracy in prediction. Weighting is also used in miscalculation management. That is, if a rule misclassifies some examples in the testing dataset, its weight can be penalized in proportion to the number of cases it misclassifies relative to the number that it classifies correctly. The new weight would then be used in a subsequent step designed to improve the prediction accuracy of the particular methodology.

METHODOLOGY AND APPROACH

Unlike existing systems that build exhaustive domain knowledge from a dataset, our approach constructs the initial domain knowledge base from a random sample of the dataset; and then in a second step uses further random samples from unused observations to iteratively improve, sharpen, and refine the domain knowledge. To date the focus of research on this algorithm and associated methodology has been to improve the prediction accuracy. At the beginning of the process the user may specify a desired predication accuracy; for example, 90%. The algorithm iterates until either this accuracy level is reached, or until there is no incremental change in the domain knowledge base for a certain period of time (or number of iterations). Depending upon the size and complexity of the data being mined, a saturation point can be reached where no incremental knowledge can be found.

Analogically, our method is like stroking with a pencil across a piece of paper laid over a coin to bring out the image engraved on the coin. The first light rubbing will give a very rough picture of the image on the coin. Successive light rubbings will bring out a somewhat less rough picture of the image on the coin. Not all the details will show, although the amount of detail will increase with each rubbing. As repetitive rubbings are conducted, the picture becomes clearer and clearer until at last the entire image shows in total clarity (see Figure 1 for a graphical illustration).

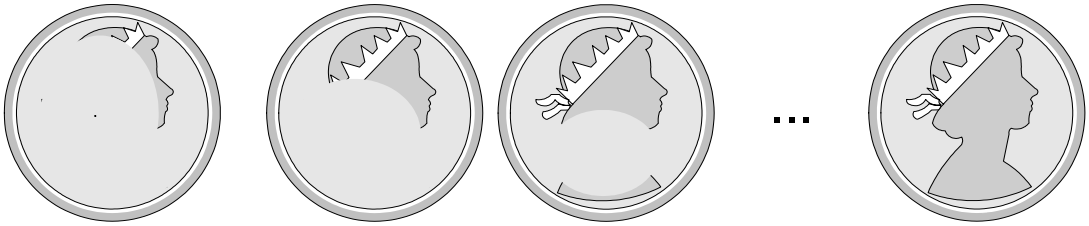

Figure 1. Revealing the perfect picture via coin scrubbing (used as an analogy to explain the process of discovering knowledge through iterative refinement algorithm)

Algorithmically, the IRA is composed of two main steps. What follows is a conceptual description of these two steps.

Step 1: Domain Knowledge Construction

The domain knowledge base is built using an iterative, weighted process that randomly selects without replacement sets of observations, called trials, from a predefined subset of the dataset (the training set). The first iteration collects a random sample from the training set and constructs a decision tree, which is converted to and stored as a weighted set of decision rules called the master domain knowledge base. The second iteration selects a new random sample from unused observations in the training set - a second trial – and extracts rules. Rules that appeared in the previous iteration are increased in weight according to frequency of appearance, and new rules are merged into the master domain knowledge base. The third iteration continues in the same manner with a new random sample from unused observations in the training set, increasing weights of

existing rules and merging new rules into the domain knowledge base. Iterations continue in this manner until the master domain knowledge base is not altered for some predetermined number of iterations. The domain knowledge base is then ready for the domain knowledge refinement step in the algorithm.

Step 2: Domain Knowledge Refinement

For the first iteration of Step 2, a random sample is taken from an unused subset of the database (the set of observations outside the training set). The existing rule base, the master domain knowledge base, is checked against the sample for accuracy of prediction. Any rules that result in an incorrect conclusion are assigned a penalty. A penalty value for a given rule is computed as a function of the number of records in the data, the largest rule weight for any rule at the end of Step 1, the number of samples that have been falsely classified by this rule, and the maximum penalty that currently exists for any rule. The penalty value for a rule is used to reduce the existing weight of the rule in the domain knowledge base. New rules and the penalty weights for erroneously classifying rules are merged into the existing domain knowledge base.

The second iteration of Step 2 takes a new random sample (distinct from the random sample of the first iteration) from the unused subset of the database. Again, penalties are assigned or increased for any rules that result in incorrect conclusions. The master domain knowledge base is updated with new rules and penalties. Step 2 iterations continue until either no incremental improvement is made in the domain knowledge base, or until the accuracy level specified by the user has been achieved.

THE DATA

To run preliminary tests of our algorithm, we acquired 18 regional datasets from the 1990 U.S. Census 1990 Public Use Microsample (Census, 1990)[1]. Combined, these 18 files included about 4.5 million males and 5 million females, totaling to 9.1 million records and 85 variables, requiring approximately 1.5 gigabytes of secondary storage. The prediction variable we selected for our preliminary analysis was annual personal income for U.S. citizens over 18 years of age. Initially we constructed a categorical variable to represent the continuous income variable; its value was 1 if the income was greater than the median and 0 if the income was less than the median. In a preprocessing step we eliminated from the data unusable records (non-U.S. citizens, records associated with individuals less than 18 years old, or characterized by missing or erroneous values) and ended up with 5.1 million relevant and usable records.

PRELIMINARY RESULTS

There are several standard statistical and classification tools against which the performance of a new algorithm can be compared. These are (i) popular decision tree algorithms such as CART (classification and regression tree, Breiman et al., 1984) and CHAID (Chi-squared Automatic Interaction Detector, Kass, 1980); (ii) traditional statistical classification methods such as discriminant analysis (Fisher, 1936) and logistic regression (Dunham, 2003, p. 86), and (iii) neural networks (Haykin, 1998). We chose MLP (multi- layer perceptron, Hornik et al., 1990), and radial basis function networks (Broomhead and Lowe, 1998), for our representative neural network architectures. We used as many records as each algorithm (and its implementation in the specific software tool) can handle. We randomly selected the maximum number of records, and conducted several runs for each algorithm using different training sets for each run. We then used the average prediction performance of each algorithm on the same test dataset. Our preliminary comparison results are presented in Table 1.

For this preliminary test, IRA demonstrated a higher accuracy than any of the other classification methods, although there is no statistically significant difference. Based on our experiments, the saturation point for prediction accuracy for this particular problem domain seems to be around 83%. This, then, is the point where, for this problem domain, the iterative algorithm was stopped.

[1] Census 1990 - http://www.macalester.edu/econdata/United_States/pums.html

	ALGORITHM USED FOR COMPARISON						
	CHAID	**CART**	**ANN**	**LR**	**DA**	**C5**	**IRA**
Software Tool Used	Answer Tree (SPSS)	Answer Tree (SPSS)	Neural Connection (SPSS)	Basic Statistics (SPSS)	Basic Statistics (SPSS)	See5	Coded in Java
Training Sample Size	3.24M	3.24M	10K	300K	300K	300K	300K
Accuracy (2/3 training & 1/3 testing)	80.19	80.30	76.12* 80.68**	81.10	78.30	82.30	82.70
Maximum Data Size	None	None	10K	800K	800K	None	None
Settings	Default	Default	Default	Default	Default	Default	N/A

** Radial basis function; ** Multi-layered perceptron; M: Million; K: Thousand*

Table 1. Performance results for all models included in the preliminary tests

SUMMARY AND CONCLUSIONS

In summary, IRA is a robust, stable approach to classification in massive datasets. It is scalable and does not require that all the data be held in main computer memory. It can be run for specified lengths of time, or until it can reach a saturation point of accuracy, or some prediction accuracy specified a priory by the user. It is anticipated that IRA will be useful in working with the massive datasets in problem domains including e-business web mining, fraud detection, intrusion detection, and customer relationship management. This is an ongoing research. Our current activities include (1) improving the computational efficiency of the IRA algorithm, and (2) searching for very large datasets for classification problems to better compare our algorithm to the others. We understand that it is absolutely essential for us to further test IRA in the massive datasets that are proprietary to large business organizations, the government sector, and research companies. We look to other researchers in data mining to further our ideas and our approach.

REFERENCES

1. Agrawal, R., Imielinski, T. and Swami, A. (1993) Database mining: A performance perspective, IEEE Transactions on Knowledge and Data Engineering, 5, 6, 914 –925.
2. Bradley, P., Gehrke, J., Ramakrishnan, R. and Srikant, R. (2002) Scaling mining algorithms to large databases, Communications of the ACM, 45(8) 38-43.
3. Breiman, L., Friedman, J, Olshen, R. and Stone, C. (1984) Classification and Regression Trees, Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software.
4. Broomhead, D. and Lowe, D. (1988) Multivariable functional interpolation and adaptive networks, Complex Systems, 2, 321-355.
5. Census (1990) - http://www.macalester.edu/econdata/United_States/pums.html
6. Chan, P. and Stolfo, S. (1995) A comparative evaluation of voting and meta-learning on partitioned data, Proc. Twelfth Intl. Conf. on Machine Learning, 90-98.
7. Chan, P. and Stolfo, S. (1995) Learning arbiter and combiner trees from partitioned data for scaling machine learning, Proc. Intl. Conf. on Knowledge Discovery and Data Mining. 39-44.
8. Domingo, C., Gavalda, R. and Watanabe, O. (2002) Adaptive sampling methods for scaling up knowledge discovery algorithms, Data Mining and Knowledge Discovery, 6, 131-152.
9. Dunham, M. (2003) Data Mining Introductory and Advanced Topics, Prentice Hall/Pearson Education Inc., New Jersey.
10. Fisher, R.A. (1936) The use of multiple measurements in taxonomic problems, Eugen., 179-188.
11. Ganti, V., Gehrke, J. and Ramakrishnan, R. (1999) Mining very large databases. Computer. 32(8) 38-45.
12. Hand, D., Mannila, H. and Smyth, P. (2001) Principles of data mining, MIT Press.
13. Heinrichs, J. and Lim, J. (2003) Integrating web-based data mining tools with business models for knowledge management, Decision Support Systems, 35, 1, 103-112.
14. Hornik, K., Stinchcombe, M. and White, H. (1990) Universal approximation of an unknown mapping and its derivatives using multilayer feedforward network, Neural Networks, 3, 359-366.
15. Haykin, S. (1998) Neural Networks: A Comprehensive Foundation, New Jersey: Prentice Hall.
16. Kass, G. (1980) An exploratory technique for investigating large quantities of categorical data. Applied Statistics. 29 119-127.
17. Littlestone, N. and Warmuth, M. (1994) The weighted majority algorithm, *Information and Computation*, 108, 2, 212-261.
18. Park, S., Kim, J., Shaw, S. and Piramuth, S. (1997) A comparative study on rule refinement in expert systems, Proceedings of International Society of DSS 1997 Conference.
19. Politakakis, P. and Weiss, S. (1984) Using empirical analysis to refine system knowledge bases, Artificial Intelligence, 22, 23-48.
20. Quinlan, J. (1986) Induction of decision trees, Machine Learning, 1, 81 – 106.
21. Quinlan, J. (1991) Knowledge acquisition from structured data, IEEE Expert, 6, 6, 32-37.
22. Quinlan, J. (1993) C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, California.
23. Ruggieri, S. (2002) Efficient C4.5, IEE Transactions on Knowledge and Data Engineering, 14,2, 430-444.
24. Shavlik, J. and Towell, G. (1989) An approach to combining explanation-based and neural learning algorithms, Connection Science, 1, 233-255.