Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2005 Proceedings

Americas Conference on Information Systems (AMCIS)

2005

Using Optimization-Based Classification Method for Massive Datasets

Yi Peng
University of Nebraska at Omaha, ypeng@mail.unomaha.edu

Gang Kou
University of Nebraska at Omaha, gkou@mail.unomaha.edu

Yong Shi
University of Nebraska at Omaha, yshi@mail.unomaha.edu

Zhengxin Chen
University of Nebraska at Omaha, zchen@mail.unomaha.edu

Using Optimization-Based Classification Method for Massive Datasets

Yi Peng[1]
ypeng@mail.unomaha.edu

Gang Kou[1]
gkou@mail.unomaha.edu

Yong Shi[1,2]
yshi@mail.unomaha.edu
yshi@gscas.ac.cn

Zhengxin Chen[1]
zchen@mail.unomaha.edu

[1]Peter Kiewit Institute of Information Science, Technology & Engineering, University of Nebraska, Omaha, NE 68182, , Phone number: ++1(402)5543429 or ++1(402)5543625

[2]Chinese Academy of Sciences Research Center on Data Mining and Knowledge Management, Beijing 100039, China, Phone number: ++8613651346898

ABSTRACT

Optimization-based algorithms, such as Multi-Criteria Linear programming (MCLP), have shown their effectiveness in classification. Nevertheless, due to the limitation of computation power and memory, it is difficult to apply MCLP, or similar optimization methods, to huge datasets. As the size of today's databases is continuously increasing, it is highly important that data mining algorithms are able to perform their functions regardless of dataset sizes. The objectives of this paper are: (1) to propose a new stratified random sampling and majority-vote ensemble approach, and (2) to compare this approach with the plain MCLP approach (which uses only part of the training set), and See5 (which is a decision-tree-based classification tool designed to analyze substantial datasets), on KDD99 and KDD2004 datasets. The results indicate that this new approach not only has the potential to handle arbitrary-size of datasets, but also outperforms the plain MCLP approach and achieves comparable classification accuracy to See5.

Keywords: Classification, Stratified Random Sampling, Majority vote, MCLP

INTRODUCTION

Over the years, optimization-based algorithms have shown their effectiveness in data mining classification (e.g., Bugera, Konno, and Uryasev, 2002) and Multi-Criteria Linear programming (MCLP) is one of the optimization-based classification methods (e.g., Shi, Wise, Luo, and Lin, 2001). Nevertheless, due to the limitation of computation power and memory, it is difficult to apply MCLP, or similar optimization methods, to huge datasets which may contain millions of observations. As the size of today's databases is continuously increasing, it is highly important that data mining algorithms are able to perform their functions regardless of the sizes of datasets.

Develop mining algorithms that scale to real-life massive databases is the first research challenges proposed by Bradley, Fayyad, and Mangasarian in their overview of applying mathematical programming for data mining. They also pointed out that "approaches that assume that data can fit in main memory need to be revised or redesigned (Bradley, Fayyad, and Mangasarian 1998)." MCLP is such an approach that requires the data to fit in main memory. This requirement comes from the fact that constraint matrix must be loaded into main memory in order to achieve an acceptable computation time and the size of constraint matrix is determined by the size of the training dataset. Therefore, as the size of dataset increases, the computation time increases and performance degraded.

The objectives of this paper are: (1) to propose a new stratified random sampling and majority-vote ensemble approach, and (2) to compare this approach with the plain MCLP approach (which uses only part of the training set), and See5 (which is a decision-tree-based classification tool designed to analyze substantial datasets), on KDD99 and KDD2004 datasets. The results indicate that this new approach not only has the potential to handle arbitrary-size of datasets, but also outperforms the plain MCLP approach and achieves comparable classification accuracy to See5.

The paper is organized in five parts. The first part describes the revised stratified random sampling used in this paper. The second part provides information about Multi-Criteria Linear programming (MCLP) two-group classification model formulation. The third part presents the majority-vote ensemble process. The fourth part discusses experimental results. The fifth part concludes the paper.

STRATIFIED RANDOM SAMPING

Since MCLP requires training datasets to fit in main memory, the size of training dataset is limited by the capacity of main memory. One possible solution is to use only part of the training dataset when the dataset size is huge. However, this approach may loss valuable information that exists in the unused part of the training dataset. In order to make the best use of the training dataset, we employ a revised stratified random sampling.

Let's briefly describe how standard stratified random sampling works. First, the dataset is partitioned into groups of data called strata. Each data belongs to one and only one stratum. Second, a sample is selected by some design within each stratum (Thompson 1992). As a sampling technique, the goal of stratified random sampling is to select a portion of a population that can be used as a "representation" of the population as a whole. While we utilize the idea of stratified random partition from stratified sampling, we believe that the classification results of using the entire training dataset to train the classification model should be better than using only a sample of the training dataset. Thus, we revise the standard stratified random sampling by repeatedly selecting stratified samples until the whole training dataset is partitioned into subsets that can fit in main memory.

The following procedure summarized the sampling process:

<u>Stratified Random Sampling Process</u>

Input: The data set A = $\{o_1, o_2, o_3, L, o_n\}$ as the population, n is the number of observation in the population and is a huge number; m is the number of subpopulations.

Output: L (L=$\lfloor 0.9 \times n / m \rfloor$**)** non-overlapping Training sets: Tr_1, Tr2, Tr3, … , $\text{Tr}_{L,}$ and the Test set Te.

Step 1 Generate the Training set Tr (90% of the population) and Test set Te (10% of the population) via Random selection from A.

Step 2 Tr is evenly partitioned into *m* subpopulations or strata by random selection.

Step 3 One random sample is drawn from every *m* subpopulations without replacement until all L training sample set has *m* samples.

END

The proportion between training dataset and test dataset in this paper is 9:1. That is, 90% of the dataset is used for training and 10% of the dataset is used for testing. One thing need to mention is that this proportion is not precise unless all the classes in a dataset have the same proportion. If different classes in a dataset have different proportion, then the number of data in the smaller class will be used to calculate the proportion between training and test sets. Since the proportions of different classes in datasets are normally different, the 9:1 proportion is only an approximation.

The value of *m* is not fixed; rather, it is flexible and can be adjusted according the size of the dataset and the capacity of main memory.

TWO-GROUP MULTI-CRITERIA LIENAR PROGRAMMING MODEL

This section describes the two-group MCLP model briefly. Since the major purpose of this paper is to propose and test the viability of applying stratified sampling and majority-vote ensemble method on massive datasets, we will use the simplest classification form of MCLP: two-group classification. For more details of two-group MCLP model formulation, please refer to (Shi, Wise, Luo, and Lin, 2001). For more general information about multiple-criteria decision making and its applications, please refer to (Yu 1985) and (Shi 2001).

Often linear classification models use a linear combination of the minimization of the sum of overlapping (represented by α_i) and maximization of the sum of distance (represented by β_i) to reduce the two criteria problem into a single criterion. A two-criterion linear programming model is stated as:

(Model 1) *Minimize* $\Sigma_i \alpha_i$ and *Maximize* $\Sigma_i \beta_i$

Subject to:

$$A_i X = b + \alpha_i - \beta_i, A_i \in G1,$$

$$A_i X = b - \alpha_i + \beta_i, A_i \in G2,$$

Where A_i are given, X and b are unrestricted, and α_i and $\beta_i \geq 0$. The advantage of this conversion is to easily utilize all techniques of LP for separation, while the disadvantage is that it may miss the scenario of trade-offs between these two separation-criteria.

Applying the techniques of MCLP and the compromise solution, we want to minimize the sum of α_i and maximize the sum of β_i simultaneously. We assume the "ideal value" of $-\Sigma_i \alpha_i$ be $\alpha^* > 0$ and the "ideal value" of $\Sigma_i \beta_i$ be $\beta^* > 0$. Then, if $-\Sigma_i \alpha_i > \alpha^*$, we define the regret measure as $-d_\alpha^+ = \Sigma_i \alpha_i + \alpha^*$; otherwise, it is 0. If $-\Sigma_i \alpha_i < \alpha^*$, the regret measure is defined as $d_\alpha^- = \alpha^* + \Sigma_i \alpha_i$; otherwise, it is 0. Thus, we have (i) $\alpha^* + \Sigma_i \alpha_i = d_\alpha^- - d_\alpha^+$, (ii) $|\alpha^* + \Sigma_i \alpha_i| = d_\alpha^- + d_\alpha^+$, and (iii) $d_\alpha^-, d_\alpha^+ \geq 0$. Similarly, we derive $\beta^* - \Sigma_i \beta_i = d_\beta^- - d_\beta^+$, $|\beta^* - \Sigma_i \beta_i| = d_\beta^- + d_\beta^+$, and $d_\beta^-, d_\beta^+ \geq 0$. A two-group MCLP model has been gradually evolved as:

(Model 2) *Minimize* $d_\alpha^- + d_\alpha^+ + d_\beta^- + d_\beta^+$

Subject to:

$$\alpha^* + \Sigma_i \alpha_i = d_\alpha^- - d_\alpha^+,$$

$$\beta^* - \Sigma_i \beta_i = d_\beta^- - d_\beta^+,$$

$$A_i X = b + \alpha_i - \beta_i, A_i \in G1,$$

$$A_i X = b - \alpha_i + \beta_i, A_i \in G2,$$

where A_i, α^*, and β^* are given, X and b are unrestricted, and $\alpha_i, \beta_i, d_\alpha^-, d_\alpha^+, d_\beta^-, d_\beta^+ \geq 0$.

Based on Model 2, the following process and a C++ program (Kou, Liu, Peng, Shi, Wise, and Xu, 2003) were developed to compute MCLP solutions. After the MCLP-based classification process is successfully executed, we will have L (the number of training samples) set of optimal solutions X_j^*. These set of optimal solutions will be used in the majority-vote ensemble process in the following section.

<u>Multi-criteria Linear Programming-based Classification Process</u>

Input: The training data set $\mathrm{Tr}_j = \{o'_1, o'_2, o'_3, \mathrm{L}, o'_m\}$, j=1, 2… L, Testing set Te, boundary b, α^*, β^*

Output: The optimal solution: $X_j^* = (x_1^*, x_2^*, x_3^*, \ldots, x_r^*)$ (r is the number of attributes of observation), the classification score $MCLP_i$

Step 1 Apply the two-group MCLP model to Tr_j (j=1, 2… L) to compute the compromise solution $X_j^* = (x_1^*, x_2^*, \ldots, x_r^*)$ as the best weights of all r attributes with given values of control parameters (b, α^*, β^*).

Step 2 The classification score $MCLP_i = o_i X^*$ of each observation has been calculated against the boundary b to check the performance measures of the classification.

Step 3 If the classification results of Step 2 is acceptable (i.e., the given performance measure is larger or equal to the given threshold), go to the next step. Otherwise, choose different values of control parameters (b, α^*, β^*) and go to Step 1.

Step 4 Use $X^* = (x_1^*, x_2^*, \ldots, x_r^*)$ to calculate the MCLP scores for all o_i in the test set Te and conduct the performance analysis. If it produces a satisfying classification result, go to the next step. Otherwise, go back to Step 1.

Step 5 Loop until L different X_j^* are generated.
END

MAJORITY-VOTE ENSEMBLE

Stratified random sampling process partitioned the original training dataset into main memory-fitted L set of training datasets and MCLP-based classification process computed L set of optimal solutions using the L set of training datasets. The next step is to generate an effective and efficient solution based on these optimal solutions.

One popular method of combining set of classifiers is ensemble method. Weingessel, Dimitriadou and Hornik (2003) list a series of ensemble-related publications (Dietterich 2000; Lam 2000; Parhami 1994; Bauer and Kohavi, 1999; Kuncheva 2000). Previous research has shown that ensemble method can help to increase classification accuracy and stability. The simplest aggregation process of ensemble method is either an average or a simple majority-vote over individual classifier/solution (Opitz and Maclin 1999; Zenobi and Cunningham 2002). In majority vote process, each solution has one vote for each data record and the final classification result is determined by the majority votes. The numbers of voters to form ensembles are randomly chosen insofar as they are odd. The following steps describe the majority-vote ensemble process used in this paper:

Majority-Vote Ensemble Process

Input: The Testing data set Te = $\{ o''_1, o''_2, o''_3, \mathrm{L}, o''_m \}$, boundary *b*, L set of optimal solutions: $X^* = (x_1^*, x_2^*, x_3^*, \ldots, x_r^*)$.

Output: The classification score $MCLP_i$, the prediction P_i

Step 1 A committee of L classifiers X^* is formed.
Step 2 The classification score $MCLP_i = A_i X^*$ of each observation has been calculated against the boundary *b* by every member of the committee. The performance measures of the classification will be decided by majorities of the committee. If more than half of the committee members find the correct classification result, then the prediction P_i for this observation is successful, otherwise, the prediction is failed.
Step 3 The accuracy for each group will be computed by the percentage of successful classification in all observations.
END

The final classification results can then be compared with MCLP results that average classification results of using only part of the training dataset and other classification method (in this paper, see5) using full set of training dataset.

EXPERIMENTAL STUDY AND RESULTS

In previous sections, we described stratified random sampling, MCLP classification computation, and majority-vote ensemble method. By combining these three methods, we are able to apply MCLP classification method on massive datasets. As explained in the stratified random sampling section, MCLP requires the training dataset to fit in main memory. Without stratified sampling and ensemble, we can only use part of the training dataset to in our earlier work (Kou, Peng, Yan, Shi, Chen, Zhu, Huff, and McCartney, 2004). Intuitively, the new approach that utilize stratified sampling and ensemble ought to perform better than the plain approach that uses only part of the training dataset since it makes the best use of the whole training dataset. The objective of the following two experiments is to investigate whether the new approach can outperform the plain one. In addition, a comparison with well-known classification method that is capable of dealing with large datasets is included for completeness. We chose see5, a well-known classification tool that is based on decision tree and is designed to analyze substantial datasets, for comparison.

Two publicly available datasets from KDD classification cup 1999 and 2004 are chosen for our experiments. These two datasets are chosen as our benchmarks because they have been analyzed by researchers from different fields using various classification methods and their sizes are large.

KDD99 Classification Cup: Intrusion Detection Dataset

The KDD-99 data set was provided by DARPA in 1998 for the competitive evaluation of intrusion detection approaches. A version of this dataset was used in 1999 KDD-CUP intrusion detection contest. There are four main categories of attacks:

denial-of-service (DOS); unauthorized access from a remote machine (R2L); unauthorized access to local root privileges (U2R); surveillance and other probing. The training dataset contains a total of 24 attack types while the testing dataset contains an additional 14 types (Stolfo, Fan, Lee, Prodromidis, and Chan, 2000). Since we are focusing on two-group classification, only one type of attack: DOS, which has a relatively large size, is selected to compare with Normal data.

Table 1 summarizes the experimental results. The table consists of two major parts: training dataset and test dataset. Both datasets has three rows: average, ensemble, and see5. Average is the average classification results of the training or test set when applying the MCLP classifiers from the 222 different training sets. That is, average represents the plain approach that uses only part of the training dataset in classification. Because a single result from using part of the training set is not representative, we use the average of a set of such results to illustrate its final performance. Ensemble is the classification results of the training or test set when applying stratified random sampling, MCLP classification, and majority-vote ensemble processes. See5 is the classification results of the training or test set when applying see5 release 1.19 for Windows (Rulequest Research 2003). The column "Correctly identified" indicates the number of correctly classified data record in the designated category or class. For example, the figure 218890 in the "Average" row and "DOS" training dataset column indicates that 218890 out of 222000 DOS data records were correctly classified using Average method. The column "Accuracy" is the percentage representation of the column "Correctly identified". For example, the figure 98.6% in the "Average" row and "DOS" training dataset column was calculated by using 222000 (total DOS data) divided by 218890 (correctly identified DOS data). In Table 1, we define "Type I Error" to be the percentage of predicted Normal records which are actually DOS records and "Type II Error" to be the percentage of predicted DOS records which are actually Normal records. In a network surveillance system, Type I Error shows rate for the "missing" alerts and Type II Error is the rate for the "false" alarms. The average of Type I error and Type II error is also reported.

The performance of a classification method is judged by classification accuracies of test dataset. The results in Table 1 tell us that see5 achieves the best classification accuracy for DOS data (99.95%) and the new approach (Ensemble) achieves the best classification accuracy for Normal data (99.5%).

KDD99							
	DOS		NORMAL		Type I Error	Type II Error	AVG of Type I and II Error
	Correctly Identified	Accuracy	Correctly Identified	Accuracy			
Training Dataset (222000 Dos data + 222000 Normal data)							
Average	218890	98.60%	215871	97.24%	1.420%	2.724%	2.072%
Ensemble	221614	99.83%	220913	99.51%	0.174%	0.488%	0.331%
See5	221817	99.92%	221985	99.99%	0.082%	0.007%	0.045%
Test Dataset (24267 Dos data + 570813 Normal data)							
Average	23822	98.17%	551143	96.55%	0.081%	45.227%	22.654%
Ensemble	24233	99.86%	567942	**99.50%**	0.006%	10.593%	5.299%
See5	24254	**99.95%**	566776	99.29%	0.002%	14.270%	7.136%

Table 1. KDD99 Results Comparison

KDD2004 Supervised Classification Cup: Quantum Physics Dataset

KDD-cup 2004 (KDD-cup 2004) provided two datasets: Particle Physics Task and Protein Homology Prediction Task. For each task, a supervised training set and test set were given. The dataset we used in this experiment is the Particle Physics Task. The goal of this task is to find a classification rule that differentiates between two types of particles generated in high energy collider experiments (KDD-cup 2004). These two types are defined as Positive or Negative. The original task asks for 4 sets of predictions: accuracy, ROC area, cross entropy, and q-score. For the purpose of our experiment, only accuracy is predicted. Table 2 summarizes the experimental results. The layout of table 2 is the same as table 1. And we define "Type I Error" to be the percentage of predicted Negative records which are actually Positive records and "Type II Error" to be the percentage of predicted Positive records which are actually Negative records.

For the Positive class, ensemble method provides the best classification accuracy (70.79%), followed by see5 (70.31%). For the Negative class, ensemble method provides the best classification accuracy (73.05%), followed by see5 (71.86%).

KDD2004							
	Positive		Negative		Type I Error	Type II Error	AVG of Type I and II Error
	Correctly Identified	Accuracy	Correctly Identified	Accuracy			
Training Dataset (22000 Positive data + 22000 Negative data)							
Average	18070	82.14%	12597	57.26%	23.779%	34.226%	29.003%
Ensemble	15257	69.35%	15484	70.38%	30.337%	29.927%	30.132%
See5	15756	71.62%	16122	73.28%	27.917%	27.170%	27.544%
Test Dataset(3139 Positive data + 2861 Negative data)							
Average	2162	68.88%	1896	66.27%	34.006%	30.860%	32.433%
Ensemble	2222	**70.79%**	2090	**73.05%**	30.496%	25.760%	28.128%
See5	2207	70.31%	2056	71.86%	31.191%	26.726%	28.959%

*Average is the average classification result of the training or testing set when applying the classifiers from the 440 different training sets

Table 2. KDD2004 Results Comparison

CONCLUSION

In this paper, we proposed a new approach that combined stratified random sampling, MCLP classification, and majority-vote ensemble to handle massive dataset classification problem.

Two publicly available datasets, KDD99 and KDD 2004, were used to test the viability of this new approach. The experimental results indicate that the new approach outperforms the plain approach, which uses only part of training dataset, in both sets. Also, the results of the new approach are comparable with see5 when applying to large datasets. The KDD99 training set has the size of 444,000, but our proposed approach should be scalable to even larger size of data sets.

REFERENCES

1. Bauer, E. and Kohavi, R. (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, *Machine Learning*, 36, 105–139.
2. Bradley, P.S., Fayyad, U.M., Mangasarian, O.L. (1999) Mathematical programming for data mining: Formulations and challenges. *INFORMS Journal on Computing* 11 217-238.
3. Bugera, V., Konno, H. and Uryasev, S. (2002) Credit Cards Scoring with Quadratic Utility Function. *Journal of Multi-Criteria Decision Analysis* **11** 197-211.
4. Dietterich, T. (2000) Ensemble methods in machine learning. In Kittler and Roli (eds.), *Multiple Classifier Systems*, vol. 1857 of Lecture Notes in Pattern Recognition, pp. 1–15. Springer.
5. KDD-Cup 2004, available at: http://kodiak.cs.cornell.edu/kddcup/.
6. Kou, G., Liu, X., Peng, Y., Shi, Y., Wise, M., and Xu, W. (2003) Multiple Criteria Linear Programming to Data Mining: Models, Algorithm Designs and Software Developments, *Optimization Methods and Software*, Vol. 18, 453-473.
7. Kou, G., Peng, Y., Yan, N., Shi, Y., Chen, Z., Zhu, Q., Huff, J. and McCartney, S. (2004) Network Intrusion Detection by Using Multiple-Criteria Linear Programming, *2004 International Conference on Service Systems and Service Management*, July 19 to 21,Beijing, China.
8. Kuncheva, L. I. (2000) Clustering-and-selection model for classifier combination, *Proceeding of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies* (KES'2000).
9. Lam, L. (2000) Classifier combinations: Implementations and theoretical issues. In Kittler and Roli (eds.), *Multiple Classifier Systems*, vol. 1857 of Lecture Notes in Pattern Recognition, pp. 78–86. Springer.
10. Opitz D. and Maclin R. (1999) "Popular ensemble methods: an empirical study", *Journal of Artificial Intelligence Research 11*, 169-198.
11. Parhami, B. (1994) Voting algorithms, *IEEE Transactions on Reliability*, 43(4), 617–629.
12. Rulequest Research (2003), available at: http://www.rulequest.com/see5-info.html.
13. Shi, Y. (2001) Multiple Criteria and Multiple Constraint Levels Linear Programming: concepts, techniques, and applications, World Scientific publication, New Jersey.

14. Shi, Y., Wise, M., Luo, M., and Lin, Y. (2001) Data Mining in Credit Card Portfolio Management: A Multiple Criteria Decision Making Approach. *Multiple Criteria Decision Making in the New Millennium*, In: M. Koksalan and S. Zionts (eds.), Springer, Berlin, 427-436.
15. Stolfo, S.J., Fan, Wei., Lee, W., Prodromidis, A. and Chan, P.K. (2000) Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project, *Proceeding of DARPA information survivability conference and exposition 2000*.
16. Thompson, S.K. (1992) Sampling, A Wiley-Interscience Publication, New York.
17. Weingessel, A., Dimitriadou, E., and Hornik, K. (2003) An Ensemble Method for Clustering, working paper of *the 3rd International Workshop on Distributed Statistical Computing*, March 20-22, Vienna, Austria, Vienna University of Technology.
18. Yu, P.L. (1985) Multiple-Criteria Decision Making: concepts, techniques, and extensions, Plenum press, New York and London.
19. Zenobi, G. and Cunningham, P. (2002) An Approach to Aggregating Ensembles of Lazy Learners That Supports Explanation, *Lecture Notes in Computer Science*, Vol. 2416, p. 436-447,.