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

Follow this and additional works at: http://aisel.aisnet.org/amcis2005

Recommended Citation
http://aisel.aisnet.org/amcis2005/110

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Using Optimization-Based Classification Method for Massive Datasets

Yi Peng
ypeng@mail.unomaha.edu

Gang Kou
gkou@mail.unomaha.edu

Yong Shi
yshi@mail.unomaha.edu
yshi@gscas.ac.cn

Zhengxin Chen
zchen@mail.unomaha.edu

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

2Chinese 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 SAMPLING**

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:

**Stratified Random Sampling Process**

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

**Output:** $L$ (where $L = \left\lfloor \frac{0.9 \times n}{m} \right\rfloor$) non-overlapping Training sets: $Tr_1, Tr_2, Tr_3, \ldots, Tr_L$, 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 LINEAR 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 $\alpha_i$) and maximization of the sum of distance (represented by $\beta_j$) to reduce the two criteria problem into a single criterion. A two-criterion linear programming model is stated as:

\textbf{(Model 1)} \quad \text{Minimize } \Sigma \alpha_i \text{ and Maximize } \Sigma \beta_i

Subject to:

\begin{align*}
A_t X &= b + \alpha \cdot \beta, A_t \in G1, \\
A_t X &= b - \alpha + \beta, A_t \in G2,
\end{align*}

Where $\alpha$ are given, $X$ and $b$ are unrestricted, and $\alpha$ and $\beta \geq 0$. 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 $\alpha_i$ and maximize the sum of $\beta_i$ simultaneously. We assume the “ideal value” of $-\Sigma \alpha_i$ be $\alpha^* > 0$ and the “ideal value” of $\Sigma \beta_i$ be $\beta^* > 0$. Then, if $-\Sigma \alpha = \alpha^*$, we define the regret measure as $-d^\alpha = \Sigma \alpha + \alpha^*$; otherwise, it is 0. If $-\Sigma \alpha < \alpha^*$, the regret measure is defined as $d^\alpha = \alpha^* - \Sigma \alpha$; otherwise, it is 0. Thus, we have (i) $\alpha^* + \Sigma \alpha = d^\alpha - d^\alpha+$, (ii) $|\alpha^* + \Sigma \alpha| = d^\alpha + d^\alpha-$, and (iii) $d^\alpha, d^\alpha+, d^\alpha- \geq 0$.

Similarly, we derive $\beta^* - \Sigma \beta = d^\beta - d^\beta+$, $|\beta^* - \Sigma \beta| = d^\beta + d^\beta-$, and $d^\beta, d^\beta+, d^\beta- \geq 0$. A two-group MCLP model has been gradually evolved as:

\textbf{(Model 2)} \quad \text{Minimize } d^\alpha - d^\alpha + d^\beta - d^\beta+

Subject to:

\begin{align*}
\alpha^* &+ \Sigma \alpha = d^\alpha - d^\alpha+, \\
\beta^* &- \Sigma \beta = d^\beta - d^\beta+, \\
A_t X &= b + \alpha \cdot \beta, A_t \in G1, \\
A_t X &= b - \alpha + \beta, A_t \in G2,
\end{align*}

where $\alpha$, $\alpha^*$, and $\beta^*$ are given, $X$ and $b$ are unrestricted, and $\alpha$, $\beta$, $d^\alpha, d^\alpha+, d^\alpha-$, $d^\beta, d^\beta+, d^\beta-$ $\geq 0$.

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.

Multi-criteria Linear Programming-based Classification Process

\textbf{Input:} The training data set $T_r = \{ O_1, O_2, O_3, ..., O_m \}, j=1, 2,..., L$, Testing set $T_e$, boundary $b$, $\alpha^*$, $\beta^*$

\textbf{Output:} The optimal solution: $X^*_j = (x_1^*, x_2^*, x_3^*, ..., x_r^*)$ ($r$ is the number of attributes of observation), the classification score $MCLP$.

\textbf{Step 1} Apply the two-group MCLP model to $T_r (j=1, 2, \ldots L)$ to compute the compromise solution $X^*_j = (x_1^*, x_2^*, ..., x_r^*)$ as the best weights of all $r$ attributes with given values of control parameters ($b$, $\alpha^*$, $\beta^*$).

\textbf{Step 2} The classification score $MCLP = o_j, X^*$ of each observation has been calculated against the boundary $b$ to check the performance measures of the classification.

\textbf{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 ($b$, $\alpha^*$, $\beta^*$) and go to Step 1.

\textbf{Step 4} Use $X^* = (x_1^*, x_2^*, ..., x_r^*)$ to calculate the MCLP scores for all $o_j$ in the test set $T_e$ 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 \( T_e = \{ o_1, o_2, o_3, L, o_m \} \), 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).

Table 1. KDD99 Results Comparison

<table>
<thead>
<tr>
<th>KDD99</th>
<th>DOS</th>
<th>NORMAL</th>
<th>Type I Error</th>
<th>Type II Error</th>
<th>Type I and II Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correctly Identified</td>
<td>Accuracy</td>
<td>Correctly Identified</td>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td>Training Dataset (222000 Dos data + 222000 Normal data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>218890</td>
<td>98.60%</td>
<td>215871</td>
<td>97.24%</td>
<td>1.420%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>221614</td>
<td>99.83%</td>
<td>220913</td>
<td>99.51%</td>
<td>0.174%</td>
</tr>
<tr>
<td>See5</td>
<td>221817</td>
<td>99.92%</td>
<td>221985</td>
<td>99.99%</td>
<td>0.082%</td>
</tr>
<tr>
<td>Test Dataset (24267 Dos data + 570813 Normal data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>23822</td>
<td>98.17%</td>
<td>551143</td>
<td>96.55%</td>
<td>0.081%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>24233</td>
<td>99.86%</td>
<td>567942</td>
<td>99.50%</td>
<td>0.006%</td>
</tr>
<tr>
<td>See5</td>
<td>24254</td>
<td><strong>99.95%</strong></td>
<td>566776</td>
<td>99.29%</td>
<td>0.002%</td>
</tr>
</tbody>
</table>

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%).
**Table 2. KDD2004 Results Comparison**

**KDD2004**

<table>
<thead>
<tr>
<th></th>
<th>Positive</th>
<th>Negative</th>
<th>Type I Error</th>
<th>Type II Error</th>
<th>AVG of Type I and II Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly Identified Accuracy</td>
<td>Correctly Identified Accuracy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Dataset (22000 Positive data + 22000 Negative data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>18070</td>
<td>82.14%</td>
<td>12597</td>
<td>57.26%</td>
<td>23.779%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>15257</td>
<td>69.35%</td>
<td>15484</td>
<td>70.38%</td>
<td>30.337%</td>
</tr>
<tr>
<td>See5</td>
<td>15756</td>
<td>71.62%</td>
<td>16122</td>
<td>73.28%</td>
<td>27.917%</td>
</tr>
<tr>
<td>Test Dataset (3139 Positive data + 2861 Negative data)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>2162</td>
<td>68.88%</td>
<td>1896</td>
<td>66.27%</td>
<td>34.006%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>2222</td>
<td><strong>70.79%</strong></td>
<td>2090</td>
<td><strong>73.05%</strong></td>
<td>30.496%</td>
</tr>
<tr>
<td>See5</td>
<td>2207</td>
<td>70.31%</td>
<td>2056</td>
<td>71.86%</td>
<td>31.191%</td>
</tr>
</tbody>
</table>

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

**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**


