2012 International Conference on Medical Physics and Biomedical Engineering

A New Feature Selection Method for Internet Traffic Classification Using ML

Liu Zhen\textsuperscript{a}, Liu Qiong\textsuperscript{b}

\textsuperscript{a}School of computer science and engineering, South China University of Technology, Guangzhou, China
liu.zhen@mail.scut.edu.cn
\textsuperscript{b}School of software, South China University of Technology, Guangzhou, China
liuqiong@scut.edu.cn

Abstract

If 248 statistical features are used to characterize network traffic flows, the computation cost of classifier will be overlarge. The feature selection methods referenced here improve the accuracy of majority classes and meanwhile decrease the accuracy in minority classes as the cost. As a result, it brings about the multi-class imbalance problem. In this paper, main contributions include two aspects below. 1) An evaluation criterion based on information theory was proposed to assess how much do one feature bias towards one class. 2) A new feature selection method named BFS was proposed to reduce features and alleviate multi-class imbalance. BFS was compared with fast correlation-based filter (FCBF) and full feature set using Naïve Bayes and ten skewed datasets. The results show that 1) BFS is more advantage to maintain the balance of multi-class classification results than FCBF, such as the reduction of g-mean is just about 8\% using BFS, 2) classification accuracy of Naïve Bayes using BFS can achieve to 90\%.

© 2012 Published by Elsevier B.V. Selection and/or peer review under responsibility of ICMPBE International Committee.
Open access under CC BY-NC-ND license.
Keywords: machine learning; feature selection; multi-class imbalance; Internet traffic classification

1. Introduction

Valid network traffic classification is fundamental to numerous network activities, including anomaly behavior detecting, traffic engineering, and QoS deployment [1]. Simple port-based and packet payload-based classification techniques are weakened when dealing with the flows using port hopping and encryption. Machine learning (ML) techniques provide a promising alternative in classifying flows based on statistical features. And it has been a hot in traffic classification in recent years.

As we all known that feature selection method plays an important role in ML traffic classification schemes. They are expected to reduce features and improve overall classification accuracy. Even though
the classification accuracy of majority classes is improved, the accuracy of minority classes is reduced significantly. They bring about the multi-class imbalance problem. In ML Internet traffic classification, the multi-class imbalance problem could be formulated as follows: Internet traffic classifiers are overwhelmed by the majority classes and ignore the minority classes. The classification accuracy of minority classes is very small even equal to zero.

However, minority class such as ATTACK could not be ignored and is also very important for some applications such as intrusion detection system. So, it is urgent to improve traditional feature selection methods to settle the multi-class imbalance. In this paper, we devised a new filter feature selection method to focus on alleviating the multi-class imbalance problem.

We investigated on information theory and FCBF feature selection method. A measurement was proposed to evaluate the bias degree of one feature in one class. Based on this measurement, we proposed a new filter feature selection method named BFS, which was compared with the traditional filter feature selection method FCBF [2] and full feature set (Fullset) on how feature selection methods impact on the multi-class imbalance classification results.

We use g-mean, Mauc, accuracy and recall as classification performance metrics. Experimental results show that: 1) the number of features in feature subset of BFS is about ten, 2) relative the results of Fullset, BFS reduce g-mean by 8% - 9%, which is much less than the reduction of FCBF (50%), 3) Mauc and accuracy of BFS are higher than the results of FCBF in most datasets, 4) classification accuracy of Naïve Bayes using BFS can achieve to 90% on average.

The rest of this paper is organized as follows. We review related works in section 2. Then we describe our feature selection frameworks in section 3. Section 4 introduces the performance metrics. BFS is compared with FCBF feature selection method in section 5. Section 6 is the conclusion of this paper.

2. Related works

Feature selection methods are popular used in Internet traffic classification. And they are usually classified into three categories: filter methods, wrapper methods and hybrid methods.

William et al. [3] found that feature selection using correlation-based feature selection (CFS) and consistency evaluation (CON) (CFS and CON are filter feature selection methods) can greatly improve computational performance, and meanwhile the classification accuracy is not significantly degraded.

Zander et al. [4] used wrapper feature selection method namely sequential forward selection (SFS), which selects features by evaluating the performance (intra-class homogeneity metric) of classifier using datasets characterized by feature subset (started from one feature). Ultimate goal is to select an optimal feature subset for this classifier.

Moore et al. [5] used FCBF feature selection method to filter the redundancy features and evaluated the classification performance (accuracy metric) using Naïve Bayes for searching an optimal number of features. Dai L. et al. [6] proposed ChiSquared-C4.5 feature selection method for Internet traffic classification. It filters out a feature subset using Chi-squared. And then, it justify whether this feature subset is selected through evaluating classification performance (accuracy metric). These two methods are wrapper feature selection methods.

The previous feature selection methods (especially the wrapper feature selection methods) lead to the multi-class imbalance problem. Because that their performance metrics (accuracy or intra-class homogeneity) used for classifiers are dependent on the prior distribution. And, the classifiers will be overwhelmed by majority classes.

Recently, En-Najjary et al.[7] build logistic regression model for every application class, and select a feature subset for each logistic regression model using parameter estimation. They handle the class imbalance problem through transferring the multi-class classification into two-class classification. Our approach is different from them, we handle the multi-class imbalance problem straight.
3. Feature selection frameworks

3.1. Bias Coefficient Definition

Let \( A = \{A_1, A_2, \cdots, A_m\} \) as the vector of features to characterize network flows, where \( m \) is the number of features. Also, let \( C = \{C_1, C_2, \cdots, C_q\} \) be the set of traffic classes, where \( q \) is the number of classes of interest. Our goal is to select an optimal balanced feature subset from \( A \).

In designing our approach for Internet traffic classification, how to measure the bias degree of one feature in one class is a challenge. It is because that it is hard to define the positive features (bias towards minority classes) and negative features (bias towards majority classes) strictly [8] in multi-class datasets. To address this challenge, we define bias coefficient to evaluate this bias based on information theory.

In order to realize bias coefficient, we have to discretize numeric features firstly, which can be implemented by the supervised discretize method in WEKA [9]. After discretization, every numeric feature is discretized to few ranges of feature values (one range of feature values is as a feature value in the next paragraph). Definition of bias coefficient is inspired by the relative uncertainty (RU) [10]

We define \( H(X) \) as the information entropy of one variable \( X = \{x_1, x_2, \cdots, x_N\} \). \( X \) may take \( N_x \) discrete values. Suppose we observe \( X \) for \( m \) times. The \( RU \) of one random variable \( X \) is defined as:

\[
RU(X) = \frac{H(X)}{H_{max}(X)} = \frac{H(X)}{\log_2(\min\{N_x, m\})}
\]  

Consider that \( C_j \) may have \( N_{cj} \) number of samples and \( A_i \) may take \( N_{ai} \) number of values. And we define \( N_{ij} \) as the number of samples whose value of \( A_i \) equal to the \( k_{th} \) value in \( A_i \) and belong to \( C_j \). The conditional \( RU \) of \( A_i \) given \( C_j \) is calculated as:

\[
RU(A_i / C_j) = \frac{\sum_{k=1}^{N_{ai}} (-p(A_k / C_j) \log_2(p(A_k / C_j)))}{\log_2(\min\{N_C, N_x\})}
\]  

where \( p(A_k / C_j) = \frac{N_{ijk}}{N_{cj}} \)

Based on \( RU \) takes values \([0, 1]\), the definition of \( A_i \)'s bias coefficient \( B(C_j) \) in \( C_j \) is:

\[
B(A_i / C_j) = 1 - RU(A_i / C_j)
\]

According to information theory, one variable has smaller value of \( RU \), it is more certainty. If bias coefficient \( B(C_j) \) of \( A_i \) is larger, it is more certainty to \( C_j \). That means \( A_i \) bias towards \( C_j \) in higher degree.

3.2. BFS Method

In this paper, BFS focus on two goals: 1) selecting a feature subset which has balance bias degree to each class so as to alleviate the multi-class imbalance problem, 2) reducing features to improve the classification accuracy like other feature selection methods.

Firstly, we calculate the bias coefficient of each feature in each class (expressed as \( B_{array}(A_i/C_j) \), \( i = 1, \cdots, m \), and \( j = 1, \cdots, q \)). We sort all features’ bias coefficient in each class (expressed as \( B_{classArray}(C_j) = \{B_{array}(A_{i1}/C_j), B_{array}(A_{i2}/C_j), \cdots, B_{array}(A_{in}/C_j)\}, j = 1, \cdots, q \) ) by descending order. The first one in sorted \( B_{classArray}(C_j) \) bias towards \( C_j \) in the highest degree.

On the other hand, we use the symmetric uncertainty (SU) to measure discriminating ability of features. A feature has great discriminating ability if it has a high SU. The SU between \( A_i \) and \( C \) is:

\[
SU(A_i, C) = 2\left[ \frac{IG(A_i | C)}{H(A_i) + H(C)} \right]
\]
where $IG(A_i|C)$ is the conditional information gain of $A_i$ given $C$. $H(A_i)$ and $H(C)$ are the entropy of $A_i$ and $C$ respectively. We calculate $SU$ between each feature and class, and get array of $su$, which is expressed as $su(A) = \{su(A_1, C), su(A_2, C), \cdots, su(A_m, C)\}$.

Secondly, we select feature for every class. One feature is selected for one class $C_j$, if it has the highest value of $SU$ among these features whose bias coefficient in $C_j$ is during $DQG$ ($\alpha = 0.6$ and $\beta = 0.8$). All classes’ feature builds up an optimal feature subset, which has balanced bias coefficient in each class and great discriminating ability. The object function of selecting a feature for $C_j$ is:

$$\text{FeatureSelection}(C_j) = \arg \max_{A \in A} {su(A_i | C_j < B)}$$

4. Performance metrics

$g$-mean and $Mauc$ are used as our experimental metrics, since they are independent on prior distribution [11]. In order to illustrate the classification accuracy improvement by feature selection methods, accuracy and recall are also used to measure our classification results.

The calculating method of $g$-mean [11] is:

$$g\text{-mean} = \left( \prod_{j=1}^{q} \frac{n_{jj}}{n_{+j}} \right)^{\frac{1}{q}} \quad \text{where } n_{+j} = \sum_{i=1}^{q} n_{ij}$$

where $n_{ij}$ is the number of samples in $C_j$ and identified to be $C_j$. $n_{jj}/n_{+j}$ means the accuracy by class. $k$ is the number of classes participating in the classification. The smaller of $g$-mean, the multi-class imbalance problem is more severity.

$Mauc$ metric is the improvement of $AUC$ (area under the Receiver Operating Characteristic curve), since $AUC$ could not be used in multi-class datasets. $Mauc$ is as equation (8) [12], where $|C|$ is the number of classes participating in the classification. And, $S_i = \sum r_{ki}$, where $r_{ki}$ is the rank of $k_{th}$ sample in $C_i$ in the ranked list. $n_i$ and $n_j$ are the number of samples in $C_i$ and $C_j$ respectively.

$$Mauc = \frac{2}{|C|(|C|-1)} \sum_{(c_i,c_j) \in C \& \& c_i < c_j} AUC(c_i,c_j)$$

where $AUC(C, C_i) = [AUC(C / C_i) + AUC(C_i / C_i)]/2$  (9)

$$AUC(C_i / C_i) = \frac{S_i - n_i(n_i + 1)/2}{n_i n_j}$$

Accuracy metric is the fraction of correctly classified flows over all classes. It represents the overall classification accuracy of classifier. Let $n_{ij}$ as the number of samples classified correctly in $C_i$, and $n$ as the total number of samples. Accuracy can be calculated as:

$$accuracy = \frac{\sum_{i=1}^{q} n_{ii}}{n}$$

Recall metric is the fraction of flows of a specific class correctly classified. The recall of $C_j$ is calculated as equation (12), where $n_{jj}$ is the number of samples classified correctly in $C_j$, and $n_j$ is the number of samples of $C_j$.

$$recall = \frac{n_{jj}}{n_j}$$
5. Experiments

4.1. Datasets

In order to verify the effectiveness of our approach, we use ten skewed datasets which were generated for classification by Moore et al. [5]. They were split from the both link directions traces in a full 24 hour, week-day period using heuristic rules. All datasets are characterized by 248 flow features [13]. Table 1 is the classes of interest and the example applications. Table 2 is the content of total traffic flows.

<table>
<thead>
<tr>
<th>No.</th>
<th>Classes</th>
<th>Example application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WWW</td>
<td>www</td>
</tr>
<tr>
<td>2</td>
<td>MAIL</td>
<td>imap, pop2/3, smtp</td>
</tr>
<tr>
<td>3</td>
<td>FTP-CONTROL</td>
<td>ftp-control</td>
</tr>
<tr>
<td>4</td>
<td>FTP-PASV</td>
<td>ftp-pasv</td>
</tr>
<tr>
<td>5</td>
<td>ATTACK</td>
<td>Internet worm, virus attacks</td>
</tr>
<tr>
<td>6</td>
<td>P2P</td>
<td>KaZaA, BitTorrent, GnuTella</td>
</tr>
<tr>
<td>7</td>
<td>DATABASE</td>
<td>Postgres, sqlnet oracle, ingres</td>
</tr>
<tr>
<td>8</td>
<td>FTP-DATA</td>
<td>ftp-data</td>
</tr>
<tr>
<td>9</td>
<td>MULTIMEDIA</td>
<td>Windows media player, Real</td>
</tr>
<tr>
<td>10</td>
<td>SERVICES</td>
<td>X11, dns, ident, ldap, ntp</td>
</tr>
<tr>
<td>11</td>
<td>INTERACTIVE</td>
<td>ssh, klogin, rlogin, telnet</td>
</tr>
<tr>
<td>12</td>
<td>GAMES</td>
<td>Half-Life</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Flows</th>
<th>Data Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>377526</td>
<td>328092/28567/3054/2688/1793/2094/2648/5797/576/2099/110/8</td>
</tr>
</tbody>
</table>

4.2. Bias Coefficient Results

In this section, we focus on illustrating the bias coefficient of every feature in every class. We take “Data1” for example. Results are shown in figure1. X-axis is the sequence number of features and y-axis is the bias coefficient of every feature in every class. Because the samples of INTERACTIVE and GAMES are less than ten, their results are not listed out. The detail of bias coefficients in the range between 0.99 and 1 is also given out. Bias coefficients of some features equal to -0.1. Because that these features have missing values, such as truncated_data (client-server/server-client) etc. And, they will be ignored in the process of BFS. This result also indicates that the bias relationship between feature and class is “many to many”.
4.3. Feature Selection Results

The number of features chosen for each dataset using BFS and FCBF is demonstrated in Table 3.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>BFS</th>
<th>FCBF</th>
<th>Dataset</th>
<th>BFS</th>
<th>FCBF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data01</td>
<td>10</td>
<td>8</td>
<td>Data06</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Data02</td>
<td>10</td>
<td>7</td>
<td>Data07</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>Data03</td>
<td>10</td>
<td>5</td>
<td>Data08</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Data04</td>
<td>9</td>
<td>6</td>
<td>Data09</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Data05</td>
<td>9</td>
<td>7</td>
<td>Data10</td>
<td>10</td>
<td>7</td>
</tr>
</tbody>
</table>

The features chosen in more than five datasets using BFS and FCBF are as below. 1) BFS: server port, actual-data-pkts (client-server), min-segm-size (client-server), pushed-data-pkts (server-client), pushed-data-pkts (client-server), var-data-ip (server-client), initial-window-bytes (client-server), initial-window-bytes (server-client). 2) FCBF: server port, missed data (client-server), req-1323-ts (server-client), RTT-full-sz-smpls (client-server) [13]. All of them are related with data size, and no feature is related with time. This indicates that time related features such as duration, are not useful as the above features.

4.4. Classification Results

In order to demonstrate the ability of alleviating multi-class imbalance problem of BFS, we designed experiments to compare BFS with FCBF and Fullset.

In our experiment, FCBF and Naïve Bayes were implemented using WEKA tool [8]. The classification algorithm was trained by one dataset and tested on the other nine datasets. This process cycle of training with one dataset and testing against the other datasets was repeated once for each dataset. For every training dataset, we ignored the classes, which have less than ten samples. Because that their samples are not enough to represent them.

The classification results of ten datasets are shown in Figure 2. X-axis is the sequence number of datasets and y-axis is the classification results in Figure 2 (a), (b) and (c). Learn from Figure 2 (a), using FCBF, Naïve Bayes get zero value of $g$-mean in most datasets. Because some classes in these datasets have very few samples and the feature subset of FCBF lacks information to characterize these classes, such as AT-TACK and SERVICES. While, the classification results of Naïve Bayes using BFS are better. There is no zero value of $g$-mean and are mostly close to the results of Fullset. It is because that it consider not only the discriminating ability of features but also the balance of features.

Figure 2 (b) shows that $Mauc$ of BFS are larger than the results of FCBF and Fullset in most datasets. The classification accuracy is also improved by BFS like FCBF in Figure 2 (c). The highly classification accuracy of Naïve Bayes using BFS is achieved by not only improving classification accuracy in majority classes but also maintaining classification accuracy in minority classes. This could be illustrated by the results of recall metric in detail, which is shown in Figure 2 (d). And, x-axis represents the sequence number of eleven classes (the twelfth class is not shown because it has less than ten samples in all datasets) and y-axis represents the average recall of ten datasets.

Table 4 is the average results of the ten datasets. Average $g$-mean, $Mauc$ and accuracy of BFS are higher than the results of FCBF. Relative the results of Fullset, BFS reduce $g$-mean by 8% - 9%, while FCBF reduce $g$-mean by 50% significantly on average.

The classification results demonstrate that BFS can alleviate the multi-class imbalance problem. At the same time, it is also able to improve the classification accuracy like other feature selection methods.
5. Conclusion and future work

In this paper, we proposed a new filter feature selection method called BFS for ML Internet traffic classification. It was devised to alleviate the multi-class imbalance problem, reduce features, and improve classification accuracy. BFS selects an optimal feature subset which is balanced and has great discriminating ability. In order to realize BFS, we proposed bias coefficient to measure the bias degree of one feature in one class, and used SU to measure the discriminating ability of features. Compared with FCBF, BFS is advantage to maintain the balance of multi-class classification results. Our experimental results show that g-mean is not significantly degraded using BFS. Mauc and classification accuracy of Naïve Bayes using BFS can achieve to 93% and 90% respectively. As a future work, we intend to improve our feature selection method on reducing the computation overhead.

Acknowledgment

The authors would like to thank Moore et al. for providing their traffic flow datasets publicly for researchers.

References


