Dillon Welch
101-93-657
11/18/12
CSC 475 001

A Comparison of Techniques for Rare Event Classification/Class Imbalance

In a data set, a rare event is an entry that occurs very infrequently to the point that standard classification algorithms will have a difficult time in properly identifying them. This difficulty will arise in different ways depending on the algorithm, for example in k-means clustering there would not be enough data to form a separate cluster and so the events would get labeled as being in the wrong cluster. The problem of rare event classification is highly related to that of class imbalance, defined as "the number of instances which represents one class is smaller than the ones with the other classes" [1]. Many methods have been proposed to attack this problem including heuristic rules and data resampling.

The problem of rare event classification is not limited to any one domain; many areas such as network traffic classification have rare events, as port scans and DoS attacks are more common than SQL injections and user-to-root escalations. The problem with standard classifiers is that they will generally have a bias towards the classes that are more represented in the data sets. As rare events do not occur frequently, they will have a small representation in the data set. When accuracy is used as the measure of a classifier's capability, this may gloss over the fact that the classifier performs poorly on a rare event, as accuracy "does not distinguish between the numbers of correctly classified examples of different classes" [1]. This means that if a classifier always classifies a rare event incorrectly it can still have a very high accuracy. For example, in the KDD99 classifier contest, it was possible to get a 73.90% accuracy rating by classifying every test entry as a 'probe' attack. This method would have achieved better results than the worst entry submitted for the contest [2]. Other methods such as ROC curves must be used to analyze the true effectiveness of a classifier dealing with rare

events.

One group of solutions to deal with the problem of rare event classification is at the algorithmic level. This group of solutions includes ideas that tweak existing algorithms to bias them towards rare events [1]. A solution in this group is the use of heuristics, also known as signature rules, to develop a rule or set of rules that can accurately classify a rare event based on its characteristics. Algorithms for automated rule generation like the C4.5 decision tree exist, but they only generate the best fit for the given data and not an overall rule. They are not very flexible regarding changes in data over time, including new features or changes in patters of existing features, so performance is always data specific. Human generated heuristic rules via signature analysis, on the other hand, can generate rules that are easier to maintain and are more general, as they are based off an understanding of the attack dynamics in a more holistic sense than just statistical analysis. In either case, heuristic rules are powerful as they can detect events in real time with comparatively little processing power with high detection rates. Unfortunately, there are many limitations and conditions to their effectiveness. First of all, a major challenge in creating a heuristic is properly defining thresholds for each feature. Inaccurate thresholds will result in weak rules with higher rates of wrong classifications. Also, a heuristic rule requires an expert level knowledge on the event in question, which may not be available or may not be easily defined as a series of rules. Finally, it is a challenge to generalize the rule such that it will work in new environments with different types of noise. The event may have multiple areas within the feature space that it occurs and the rule will have to cover each area accurately without compromising the other areas [3].

As an example, much work has been done involving heuristic rules and expert systems within the field of network traffic classification, in particular the classification of remote-to-local (R2L) attacks. R2L attacks are defined at the KDD99 website (located at [4]) as “unauthorized access from a

remote machine, e.g. guessing password" and according to M. Sabhnani and G. Serpen in [3] "offers the most diverse set of attacks in terms of attack execution, implementation, and dynamics; R2L attacks differ vastly in terms of signatures and the host against which they are executed. The diverse knowledge required to detect R2L attacks inspired many expert systems. . . " The best of these systems, EMERALD, only correctly classified 35% of R2L attacks in its data set. Various other inferior methods include a two-stage general-to-specific framework, decision forests, and non-parametric density estimation based on Parzen-window estimators with Gaussian kernels. Even the winner of the KDD99 contest only correctly classified 7.82% of R2L attacks [3].

The authors of [3] combined the training and testing data sets of the KDD99 challenge and proposed heuristic rules for particular R2L attacks. These two attacks are the warezmaster and warezclient attacks. The warezmaster attack involves uploading warez (pirated software) to an FTP server, and the warezclient attack involves downloading warez from the FTP server. A mix of analysis and C4.5 decision tree work was used to generate the thresholds of the rules. The warezmaster attack exploits a mistake made when FTP servers have given write permissions to all users. Most public FTP servers have a guest account, which will also receive write permission. An attacker will log into the server using a guest account, create a hidden directory, and upload the warez onto the server. The relevant observable features are that an FTP session exists, large amounts of data have been uploaded to the FTP server, and new hidden directories have appeared. Based on the observations made, two rules were created based off the data. The first rule has two sub-rules, which are then combined in an or statement to make one rule. The first sub-rule, 2.1a, captures the concept of a large amount of data being transferred to the FTP server over a long duration with no data being downloaded. The second sub-rule, 2.1b, captures the concept of hidden directories being created while a guest is logged on. Based on the observations made in the KDD99 data set, this only happens when a warezmaster attack

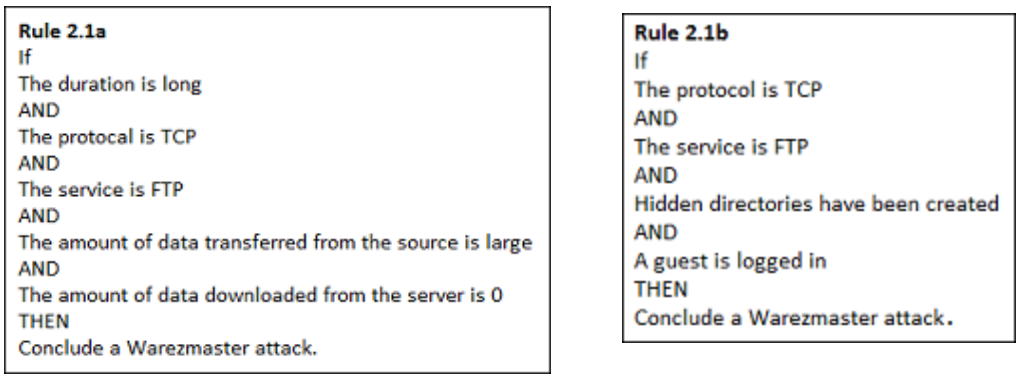

Figure 1: The two sub-rules of the first rule created for Warezmaster attacks.

has occurred. These two sub-rules are featured in Figure 1. The second rule is just a mixing of similar features from the first rule in a different way. In the combined training and testing sets, there were a total of 1622 warezmaster attacks. These rules were able to detect 1065, or 65.6597%, of them with a "false alarm" (false positive) rate of 70, or 0.005%, and a "missed alarm" (false negative) rate of 557, or more than 34.3403%. After analysis of the missed classifications, they showed that based on the data of that traffic a warezmaster attack could not have occurred, for example the FTP server showed that no data was transferred to it. The conclusion was that "these rules adequately but not necessarily precisely map the signatures of the warezmaster attacks." Warezclient rules were developed in a similar fashion to the warezmaster rules. In this case, there were 893 total attacks with a detection rate of 270, or 30.2352%, 0 false positives, and 623, or 69.7648%, false negatives. The reasoning for this was that the signature of this attack is extremely similar to regular downloading from an FTP server. There were also some potential mislabeling issues with this data. Overall, for the two attacks they achieved a 53.08% accuracy with a 0.005% false negative rate for the two attacks. This approach shows both the potential benefits and troubles of a heuristic based approach to rare event classification. If the rare event is distinct enough that precise rules can be developed for it, the approach works very well. On the other hand, if the event is very similar to other events, it is much more challenging to develop quality rules [3].

Another approach to rare event classification is to attack it at the data level by resampling the data in various ways. This approach avoids modifications to the overall learning algorithm and so are independent of it and therefore more versatile. There are five methods for resampling outlined in [1]. The first type of resampling is random undersampling. This method randomly eliminates examples that are common to decrease the effect they have on a classifier not learning rare events. Unfortunately, this may throw out important data from common events. The second type is random oversampling. Instead of randomly throwing out some of the common events, this method instead randomly duplicates rare events. This increases the chances of overfitting, as exact copies of events appear multiple times. The third type is synthetic minority oversampling technique (SMOTE). This is an oversampling method that combines similar rare event data points together to make new events. This is done by randomly selecting one or more of the k nearest instances of a randomly chosen instance and interpolating a point in between them. This avoids the problem of overfitting and expands the boundaries of the rare event. The fourth type is modified SMOTE, or MSMOTE. This divides the rare events into three categories by calculating the distance among each instance. The algorithm precedes similarly to SMOTE. If the randomly chosen instance is a “safe” example, then the k nearest instances are used to generate the new point. If the instance is a “border” example, then only the closest instance is used. If the instance is a “latent noise” instance, nothing is done. The fifth method is called selective preprocessing of imbalanced data, or SPIDER. It is a mix of undersampling and oversampling – difficult examples are filtered from the common events while local oversampling is done for rare events. There are other methods for resampling as well not talked about in [1]. Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani in [5] worked on the KDD'99 data set to create a new set named NSL-KDD in which data was contained inversely proportional to the difficulty of classifying it. To do this, they set up three instances of seven different classifier types. Training was done on randomly

generated collections of 50,000 events, while testing was done on the combined training and testing sets – amounting to over one million distinct events after duplicates are removed. A count of the classification success of each event was kept, ranging from 0 (no classifier got it correct) to 21 (all classifiers got it correct). These were separated into groups of 0-5 correct, 6-10 correct, 11-15 correct, 16-20 correct, and 21 correct. The new data set was created by randomly sampling from each group inversely proportional to the population of that group compared to the overall population. For example, the 0-5 group was only 0.04% of the overall data set, so 99.96% of the NSL-KDD data set contains events from the group. The accuracy results decreased by at least 10% when the NSL-KDD data set was used instead of the original to train and test. This approach shows that sometimes manipulating the process at the data level may expose weaknesses in traditional classifiers instead of increasing their effectiveness.

Other approaches exist besides these two. One approach is to use a cost sensitive learning framework, that is to add costs to correctly and incorrectly identifying certain classes of events. This is both a data level approach (adding costs) and an algorithmic level approach (modifying the algorithms to account for the cost framework). This approach can be used to bias the classifier towards rare events by assigning high misclassification costs to those events compared to common events. This approach has two significant drawbacks. Costs for misclassification need to be defined as this is not a normal feature of most data sets. Also, algorithms need to be modified to accept the cost framework or entirely new algorithms need to be used [1]. This approach does have the benefit of being able to be expanded to include a reject option. This option is used when there is not enough data about a particular event to classify it with confidence. In [6] and [7], this process is outlined using a Bayes risk minimization framework and optimizing SVMs with a double hinge loss. The approach resulted in a similar average risk but with a much higher accuracy rating than standard methods.

There are many methods available to attack the problem of rare event classification. It can be done at the algorithmic level with approaches such as heuristic signatures. This approach can be very successful, but requires an expert knowledge of the data and that the rare events are distinct enough for general rules to be distilled. It can also be done at the algorithmic level by techniques like resampling. Many resampling algorithms exist with increasing complexity and effectiveness. Resampling can also expose weaknesses in a classifier instead of increasing the efficiency. Methods such as a cost sensitive learning framework exist that blend both data and algorithmic level changes, but require significant changes to both the data and the algorithms to incorporate cost based decisions. Each type of approach can be effective in the right situation with the right type of data.

References

[1] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches," *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 42, no. 4, pp. 463-484, July 2012.

[2] C. Elkan, "Results of the KDD'99 Classifier Learning", *ACM SIGKDD* Explorations Newsletter, vol 1. no. 2, pp 63-64, January 2000.[1]

[3] M. Sabhnani, G. Serpen, "KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection", *Security and Management 2003*, pp 310-316.

[4] KDD Cup 1999: Overview, http://www.sigkdd.org/kddcup/index.php?section=1999, cited November 2012.

[5] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," *In Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications, CISDA 2009.*[2]

[6] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu, "Support Vector Machines with a Reject Option", *NIPS 21,*December 2008.

[7] P. Bartlett, M. Wegkamp, "Classification with a Reject Option using a Hinge Loss", *Journal of*

[1] Accessed through http://cseweb.ucsd.edu/~elkan/clresults.html

[2] Accessed through http://www.ee.ryerson.ca/~bagheri/papers/cisda.pdf