Android Malware Detection with Weak Ground Truth Data

Jordan DeLoach, Doina Caragea
Department of Computer Science
Kansas State University
{jdeloach,dcaragea}@ksu.edu

Xinming Ou
Department of Computer Science and Engineering
University of South Florida
xou@usf.edu

Abstract

For Android malware detection, precise ground truth is a rare commodity. As security knowledge evolves, what may be considered ground truth at one moment in time may change, and apps once considered benign may turn out to be malicious. The inevitable noise in data labels poses a challenge to inferring effective machine learning classifiers. Our work is focused on approaches for learning classifiers for Android malware detection in a manner that is methodologically sound with regard to the uncertain and ever-changing ground truth in the problem space. We leverage the fact that although data labels are unavoidably noisy, a malware label is much more precise than a benign label. While you can be confident that an app is malicious, you can never be certain that a benign app is really benign, or just undetected malware. Based on this insight, we leverage a modified Logistic Regression classifier that allows us to learn from only positive and unlabeled data, without making any assumptions about benign labels. We find Label Regularized Logistic Regression to perform well for noisy app datasets, as well as datasets where there is a limited amount of positive labeled data, both of which are representative of real-world situations.

Introduction

Android malware detection techniques have increasingly evolved towards machine learning. To effectively use machine learning to detect malware three elements are necessary. First, you must have a large amount of apps to learn from. Second, you need accurate labels (malware or benign) for those apps. Finally, you need an effective machine learning classifier that can accurately label new apps. Our work is motivated by the insight that, while we can be highly certain of the labels of the positive examples (malware), we can never be certain of the labels of the negative or benign examples (Kantchelian et al. 2015). Using the dataset and label preparation methods described in the following section, we verified this fact in our own dataset, where 1681 apps were reclassified by VirusTotal from benign to malware between 2015 and 2016, while only a single app was reclassified as benign after previously being considered malware.

The strong belief in the precision of the positive (malware) label and a comparatively weaker belief in the negative (benign) label present an opportunity to leverage a classifier that is not dependent on labeled negative data. Specifically, we leverage Label Regularized Logistic Regression (LR-LR) (Ritter et al. 2015), which is a modified version of Logistic Regression that learns only from labeled positive and unlabeled data. The LR-LR approach matches our problem domain well, as we are highly certain of our positive labeled data, while the rest of the data is better used as unlabeled.

Experimental Setup & Methods

Dataset & Label Preparation

We use a dataset from our previous work (Roy et al. 2015), where nearly 1 million apps are from the Google Play Store. An additional 35K malware apps come from VirusShare, and 24K from Arbor Networks. Using the app binaries, we extracted semantic features from the code using string searches. Extracted features included permissions, APIs, and intents. For further information on feature extraction, the reader can reference our previous work (Roy et al. 2015).

To identify ground truth labels for our dataset, we utilized VirusTotal, which checks an app against multiple anti-virus solutions. Using the results from VirusTotal and a multiple expert voting system, we classified each app with at least 10 A/V's marking it as malware as *high confidence malware*. We consider an app as *high confidence benign* when 0 out of 54 A/V's mark it as malware. We exclude the apps in the range 1 to 9 from the dataset, in an attempt to have a clean dataset to experiment with.

Label Regularized Logistic Regression

We utilized Label Regularized Logistic Regression to learn from positive and unlabeled data only. The LR-LR algorithm does this by using an expert-provided factor, $\tilde{p}$, to represent the proportion of positive apps in the unlabeled dataset. In addition to the terms in the optimization problem corresponding to the standard Logistic Regression algorithm, the LR-LR optimization problem also includes a label regularization term that requires the posterior class probabilities of the unlabeled data, $\hat{p}_\theta^{unlab}$, to be in line with the expert estimation of the true distribution, $\tilde{p}$. Specifically, the Kullback-Leibler (KL) divergence between the two distributions, $KL(\tilde{p}||\hat{p}_\theta^{unlab})$ needs to be minimized. We imple-

mented this algorithm as a variant of the default Spark MLLib implementation with L-BFGS optimization. The implementation is freely available on Github[1].

Experiments

Noisy Ground Truth

Given that datasets are inherently noisy, either due to malware apps that slip into the Google Play Store, or due to the false negatives produced by VirusTotal, our assumption is that benign app datasets contain a portion of malware apps within them. We seek to find out how the one-class semi-supervised method (LR-LR) outlined in the previous section handles noise in the labeling of benign apps, by comparison with three supervised algorithms: Naïve Bayes, Logistic Regression, and Support Vector Machines (SVM). We inject synthetic noise (add malware apps to a presumably clean benign set) to study how the algorithms compare to each other in the presence of different levels of noise. As the noise level increases, we increase the expert-provided $\tilde{p}$ value in LR-LR to account for more malware in the unlabeled dataset. We test over noiseless test data using 5-fold cross-validation, and report average auPRC (area under the Precision Recall curve) values over the 5 folds.

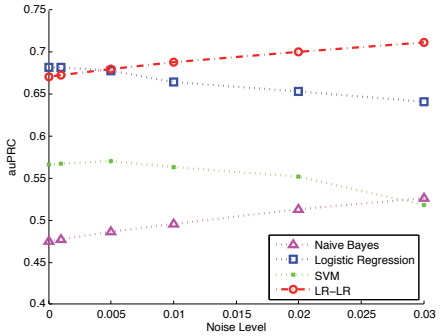

Figure 1: Performance when varying the noise level

Figure 1 shows that at low levels of noise (i.e., accurate labelings) Logistic Regression slightly outperforms LR-LR. Intuitively, this makes sense as Logistic Regression trains over both positive and negative data, while LR-LR uses only positive and unlabeled data (together with the expert-provided $\tilde{p}$ value). As the noise level increases, however, we see that LR-LR improves its performance, while other algorithms show diminishing performance, as expected. The ability of the LR-LR to handle noise can be explained by the fact that the mislabeled data is used as unlabeled and the LR-LR accounts for the noise through $\tilde{p}$. Furthermore, we should note that when we add more noise, we implicitly increase the size of the unlabeled dataset, which is likely the reason that the LR-LR method manages to improve the performance as the noise is increased.

[1] https://github.com/jdeloach/mlbSparkExperiments

Minimizing Necessary Ground Truth

Our second experiment tests which approach works best when only limited ground truth is available. We experiment with the algorithms when varying the labeled positive data, while keeping constant the noise level in the negative data.

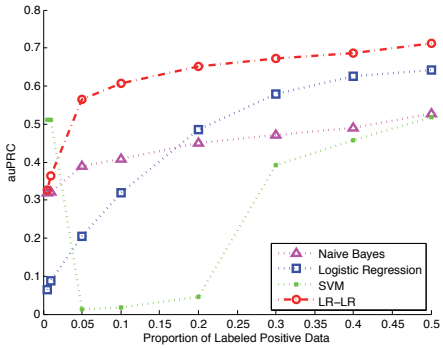

Figure 2: Performance when varying labeled positive data

As can be seen in Figure 2, LR-LR is comparatively more effective for a low amount of labeled positive data. This is to be expected as LR-LR is uniquely able to learn from the unlabeled data and grow its knowledge, even with a limited starting dataset. Also as expected, when the amount of positive labeled data is high, LR-LR and Logistic Regression have very similar performance, as the benefits offered by the inclusion of unlabeled data taper off.

Conclusions

In this work we leveraged one-class semi-supervised learning for the first time in the Android malware detection field. We carefully discussed the weaknesses in current ground truth labeling approaches in the Android space, and suggested that these weaknesses can be remedied with a one-class semi-supervised approach. Our experimental results showed that the LR-LR approach works well in the presence of noisy benign labels, leading us to believe it is a promising area of further research in Android security.

Acknowledgment

This project is partially supported by the National Science Foundation under Grant 1622402, MRI-1429316, and CC-IIE-1440548.

References

Kantchelian, A.; Tschantz, M. C.; Afroz, S.; Miller, B.; Shankar, V.; Bachwani, R.; Joseph, A. D.; and Tygar, J. D. 2015. Better malware ground truth: Techniques for weighting anti-virus vendor labels. In *Proc. of the 8th Workshop on AISec*, 45–56. ACM.

Ritter, A.; Wright, E.; Casey, W.; and Mitchell, T. 2015. Weakly supervised extraction of computer security events from Twitter. In *Proc. of the 24th WWW*, 896–905. ACM.

Roy, S.; DeLoach, J.; Li, Y.; Herndon, N.; Caragea, D.; Ou, X.; Ranganath, V. P.; Li, H.; and Guevara, N. 2015. Experimental study with real-world data for Android app security analysis using machine learning. In *Proc. of the 31st ACSAC*, 81–90. ACM.