Classifying Spam using URLs

Di Ai
diai@Stanford.edu

Stanford
Computer Science

Problem

Spam is a large and growing problem that is becoming increasingly difficult to tackle at scale. Search engines have limited resources and cannot crawl every page on the web. An algorithm that is able to identify a spam web page prior to crawl time would not only save precious crawl resources by allowing the crawler to bypass spam urls, but it would also remove the possibility of that spam url from later being served to user queries, thereby improving the quality of search results.

Data

40,000 manually rated urls were collected with 50% labeled as spam and 50% as not spam over the course of two months. Additional information that would have been available at crawl time was also included, such as first seen dates, whois contact data, and site traffic. Data is representative of the types of spam that appeared in Google Search in September and October 2018. All numerical features were scaled and standardized to have mean 0 and variance 1.

Naïve Bayes Feature Extraction

Each url was broken down into domain, top-level tld, and deep url sections. Each deep url was split into words by using various characters as delimiters, such as _, ., :, =, ;, and /. The full regex is shown below:

' |\.|\/|\/\/|:|-|_|%|\?|=|;|<|>|~|\$|&|\+'

Words were kept if they were more than 2 characters long, otherwise they were ignored. A multinomial event model was fitted on these words with Laplace smoothing and a dictionary limited to words that appeared with a frequency of at least 0.1% times the size of the dataset.

$$\begin{aligned}\mathcal{L}(\phi_y, \phi_{k|y=0}, \phi_{k|y=1}) &= \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}) \\ &= \prod_{i=1}^{m} \left(\prod_{j=1}^{n_i} p(x_j^{(i)} | y; \phi_{k|y=0}, \phi_{k|y=1}) \right) p(y^{(i)}; \phi_y).\end{aligned}$$

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} n_i + |V|}$$

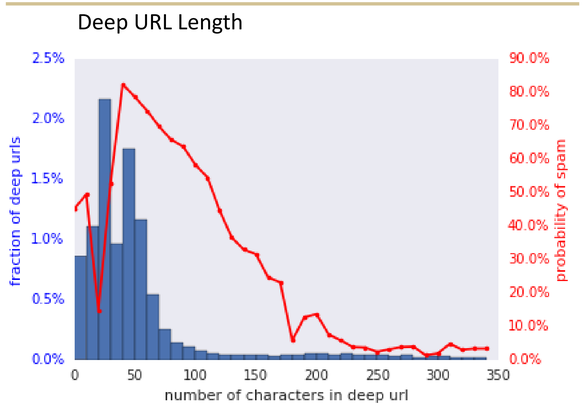
$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} n_i + |V|}.$$

https://www.example.com/buy-cheap-viagra

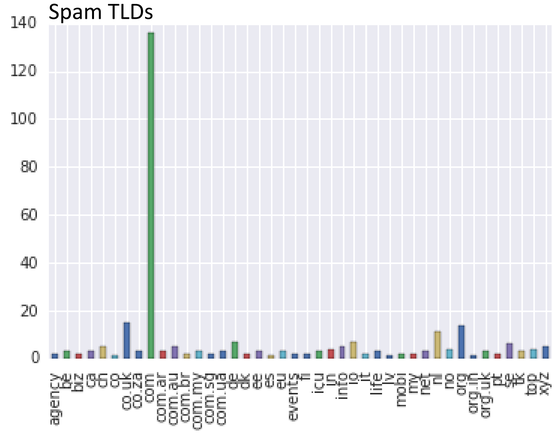
Protocol (scheme) **Sub-domain** **Domain name** **Top level domain (TLD)** Deep Url

Other Features

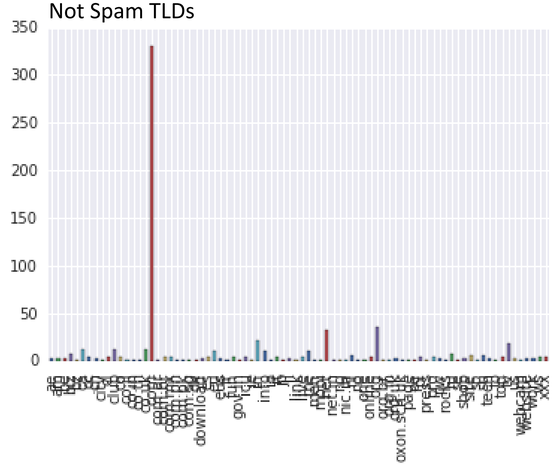
Deep URL Length



Spam TLDs



Not Spam TLDs



Models

1. Linear SVM

$$\min_{\gamma,w,b} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m$$
$$\xi_i \geq 0, \quad i = 1, \ldots, m.$$

2. RBF SVM $$K(x, z) = \exp\left(-\frac{||x - z||^2}{2\sigma^2}\right)$$

3. Random Forest $$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Results

10% of the data was set aside as the test dataset (2000 examples with roughly 50% spam/not spam). Rows in red indicate model with the best hyperparameters found via tuning.

Model	Cutoff*	Train Accuracy	Test Accuracy
Linear SVM, Cost = 1	50	88.7%	88.0%
Linear SVM, Cost = 10	50	89.4%	89.2%
Rbf SVM, Cost = 1, Gamma = 1	50	94.5%	91.2%
Rbf SVM, Cost = 1, Gamma = 0.01	50	87.4%	87.1%
Rbf SVM, Cost = 1, Gamma = 100	50	99.3%	81.5%
Rbf SVM, Cost = 10, Gamma = 1	50	96.8%	89.9%
Rbf SVM, Cost = 100, Gamma = 100	50	99.7%	81.9%
Rbf SVM, Cost = 1, Gamma = 1	5	94.4%	90.4%
Linear SVM, Cost = 1	5	89.6%	88.6%
Random Forest	50	99.7%	96.9%
Random Forest, max depth = 10	50	95.4%	95.2%
Random Forest	5	99.8%	96.5%
Random Forest, max depth = 10	5	94.4%	94.5%
Random Forest, Best	50	100.0%	97.3%

*Categorical variables were one hot encoded with a cutoff frequency threshold

Ablation Study

Model	Accuracy	Precision	Recall	AUC
SVM RBF	91.2%	91.1%	91.4%	91.1%
Remove Naive Bayes Prob Feature	87.5%	84.6%	91.2%	87.5%
Remove Url Length Feature	91.0%	91.5%	90.6%	91.0%
Remove diff between url and site first seen date	90.5%	91.0%	90.0%	90.5%
Remove ratio of url traffic to site traffic	91.2%	91.3%	91.2%	91.2%

Discussion and Future Work

Random Forest with Gini loss performed the best. To get a better idea of which variables are the most important, I also ran an ablation study on the RBF SVM, which found that the deep url naïve bayes probabilities output was the most important predictor. Looking at feature importances from random forest confirmed this as well. Since there is likely significant headroom to improve this feature, future work would involve more sophisticated feature extraction methods on the deep url portion.

References

[1] Gyongyi, Zoltan and Garcia-Molina, Hector (2005) *Web Spam Taxonomy.* In: First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005), May 10-14, 2005, Chiba, Japan.
[2] Egele, M., Kolbitsch, C. & Platzer, C. J Comput Virol (2011) 7: 51. https://doi.org/10.1007/s11416-009-0132-6
[3] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In Proceedings of the 15th international conference on World Wide Web (WWW '06). ACM, New York, NY, USA, 83-92. DOI: https://doi.org/10.1145/1135777.1135794