Study of Static Classification of Social Spam Profiles in MySpace

Danesh Irani, Steve Webb, and Calton Pu
Georgia Institute of Technology
College of Computing
Atlanta, Georgia 30332-0765
{danesh, webb, calton}@cc.gatech.edu

Abstract

Reaching hundreds of millions of users, major social networks have become important target media for spammers. Although practical techniques such as collaborative filters and behavioral analysis are able to reduce spam, they have an inherent lag (to collect sufficient data on the spammer) that also limits their effectiveness. Through an experimental study of over 1.9 million MySpace profiles, we make a case for analysis of static user profile content, possibly as soon as such profiles are created. We compare several machine learning algorithms in their ability to distinguish spam profiles from legitimate profiles. We found that a C4.5 decision tree algorithm achieves the highest accuracy (99.4%) of finding rogue profiles, while naïve Bayes achieves a lower accuracy (92.6%). We also conducted a sensitivity analysis of the algorithms w.r.t. features which may be easily removed by spammers.

1. Introduction

Social networks have grown to become an important part of social interactions for personal and business reasons. Consequently, spammers have targeted social networks as media for propagating spam. Unlike email, interactions within current social networks such as MySpace and Facebook are restricted to members of the same social network. Consequently, spam must originate from a profile inside the social network. Although the maintenance of such rogue profiles (usually called "spam profiles") is a deterrent, it has not stopped the proliferation of spam in social networks (Webb, Caverlee, and Pu 2009).

Anecdotal evidence indicates that techniques used by social network operators to detect rogue profiles in practice include collaborative filtering and behavioral analysis. Dynamic methods such as collaborative filtering (where users vote on the nature of profiles) and behavioral analysis (where logs of interactions are used to detect spamming patterns) may be eventually able to detect rogue profiles, but they require a non-trivial amount of lag time to accumulate sufficient evidence. In this paper, we study the analysis of static user profile content, which complements the dynamic methods. The main advantage of using machine learning analysis is that it can be applied as soon as the profiles are created, thus detecting suspect spam profiles before they have active propagated spam. In compensation, analysis based on machine learning depends on the quality of training data for an accurate prediction.

This paper explores the limits and potential of machine learning analysis on static profile content at creation time (or within a small time period). To test the limitations on the accuracy of such analysis, we collected a large corpus of 1.9 million MySpace profiles, including approximately 1,500 confirmed spam profiles (Webb, Caverlee, and Pu 2009). Our analysis considers two kinds of data in a social network profile:

- Categorical data - fields that can take only a limited number of values, for example: "Sex", "Age", and "Relationship Status".
- Free-form data - usually text information entered by users, for example: "About me" and "Interests".

This distinction allows appropriate machine learning algorithms to be applied to different sections of the data (e.g., decisions trees for categorical data and naïve Bayes for free-form data). To measure the discriminatory power of features and find the most important features in the identification of spam profiles, we apply the χ^2 test to measure the correlation (or lack of independence) between the features being studied and predicted class (whether a profile is rogue).

We also compare supervised machine learning techniques in their ability to detect spam profiles. On categorical data, the classifiers tested are: (1) AdaBoost algorithm (with a DecisionStump weak-classifier), (2) C4.5 decision tree, (3) Support Vector Machine, and (4) a neural network-based algorithm. In addition, we use naïve Bayes on the free-form data to classify rogue profiles.

Lastly, we perform a sensitivity analysis of the classifiers with respect to the evolution of spam profile content. Evolution of spam content is due to adversarial classification, where detection measures adopted by a social network can be countered with spammer's adaptive modification of the spam profiles to escape detection. Spam content evolution is a well known and documented phenomenon (Irani et al. 2008; Pu and Webb 2006). We evaluate the robustness of classifiers by simulating adversarial action (e.g., removing the highest discriminative features) and re-evaluating the ef-

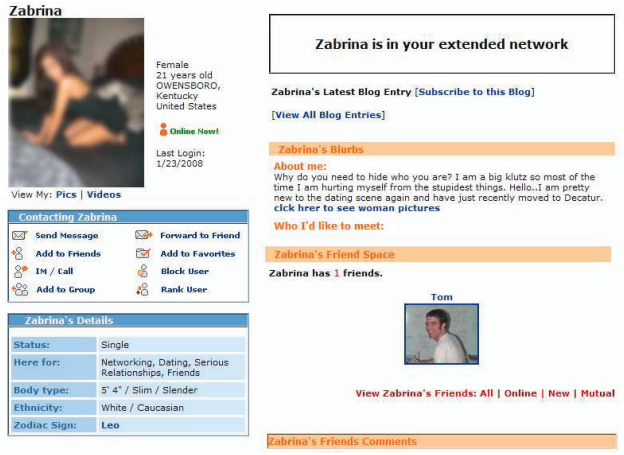


Figure 1: Example of a social spam profile

fectiveness of classifiers under the new assumptions.

Our results show that analysis of static profile content based on machine learning methods has the potential to improve significantly the detection of rogue profiles in social networks with non-trivial user profile content (e.g., MySpace and Facebook). This is particularly the case for recently created rogue profiles that have not engaged in spamming activity. Since this analysis complements dynamic methods, it has good potential for practical impact.

2. Social Spam Profiles

Social networks allow users to meet new friends, keep in touch with old friends, and participate in various social interactions. A user's presence in a social networking site is represented by a *social profile*, which allows him to maintain an identity within a social network and participate in it.

Spammers looking to propagate spam through a social network need to create a social profile. Using this *social spam profile*, they send users spam using mediums offered within the community (e.g., friend requests, messaging, and commenting). Such spam has already been seen in the wild (Webb, Caverlee, and Pu 2009), and previous work (Webb, Caverlee, and Pu 2008) has focused on gathering and characterizing such profiles.

An example of a MySpace social spam profile is shown[1] in Figure 1. The profile contains a large amount of personal information, including various deceptive properties. As typical of spam profiles, this profile portrays an attractive, young, single woman with a provocative image to entice users to view them. Once the profiles have attracted visitors, they direct them to perform an action on an external website, usually by providing an alluring story in their "About me" section. For example, the profile in Figure 1 provides a link to an external website to "see woman pictures".

[1]Provocative images have been blurred to avoid offending readers.

2.1 Motivation

Once a rogue profile is created, spam can easily be sent out to other users. Early and accurate detection of such spam profiles is essential to reducing the amount of spam on social networks. Detection at the zero-minute or at profile creation is critical for long-term efficacy.

We explore the benefits and limitations of profile classification using static user content entered during profile creation (or modification) to determine whether a profile is spammy or not. Ideally, this technique would be accurate enough to allow zero-minute determination about whether a profile is spammy or not and prevent a spammer from gaining an entry point into the social network. In practice, we submit that this technique would most likely have to be used to lower or raise another technique's decision boundary, allowing it to come to a quicker decision.

Current techniques of social spam profile detection (Zinman and Donath 2007; Markines, Cattuto, and Menczer 2009), rely heavily on detecting the spamminess of artifacts created by a user profile. For example, they analyze messages, comments, and friend requests. This approach must wait for the social spam profile to impact the social network with spam. Also, depending on the discriminatory power of features based on such artifacts, a large number of artifacts may be required before a definitive decision can be made.

Some social network sites use collaborative filtering or administrator-based reporting to manually identify social spam profiles. These techniques also suffer from a lag time between profile creation and identification, which can result in spam already having impacted users. Additionally, collaborative filters require that the profiles attract enough votes from other users to be marked as spam. Higher requirements in the number of votes will result in longer certainty in the evaluation of spam with a longer time required to reach the threshold, and vice versa. The benefit to using these dynamic techniques is usually a higher accuracy.

2.2 Related Work

Heymann et al. (Heymann, Koutrika, and Garcia-Molina 2007) provide an overview of social spam fighting mechanisms, categorizing them broadly into: Detection-, prevention-, and demotion-based strategies. In the paper, they group classification approaches as detection-based strategies, likely due to the fact that previous work (Zinman and Donath 2007; Markines, Cattuto, and Menczer 2009) mostly makes use of features which require spammy behavior to be present, before classification is possible. We focus on a prevention-based strategy using machine learning, to try and identify social spam profiles before they can be used to propagate spam.

A lot of work has been done in the area of demotion [or promotion] strategies, namely trust. TrustRank (Gyongyi, Garcia-Molina, and Pedersen 2004) and SocialTrust (Caverlee, Liu, and Webb 2008) look at graph-based techniques to ranking nodes within a network. SocialTrust looks explicitly at trust in a social network, whereas TrustRank approaches the problem from a web spam perspective, both of which are modifications of the PageRank algorithm (Page et al. 1998).

A specialized look into the trust of CouchSurfing.com and Del.icio.us is done by Lauterbach et al. (Lauterbach et al. 2009) and Noll et al. (Noll et al. 2009), which use additional attributes between friends provided by the social network to facilitate trust calculations.

3. Experiment Setup

MySpace is one of the largest and most popular social networks, making it a prime target for social spam. It also features a large amount of personal user content per profile, and most of the information is publicly viewable by default.

3.1 Data Collection

With over 200 million profiles on MySpace, collection of all the profiles would be infeasible due to computational, storage, and network loads. We previously collected, in June to September 2006, a small subset of profiles from MySpace using two different sampling strategies:

- Top 8 - Starting with a seed list of random profiles, the top 8 most popular friends were crawled, and subsequently their top 8 most popular friends were crawled in a breath first search (BFS) manner. This resulted in a collection of 891,167 connected profiles.
- Random - Profiles were crawled by generating random userid's and retrieving the profile represented by that user. This resulted in a collection of 960,505 profiles.

More details regarding these crawls, including an analysis of demographic information and language model characterization, can be found in our previous work (Caverlee and Webb 2008).

Spam profiles from MySpace were previously collected by setting up honeypot accounts and collecting profiles that sent friend requests to these accounts. Fifty-one identical honeypot accounts portraying a fictitious, young, single male were spread over geographical locations in the United States resulting in 1,496 spam profiles collected from October 2007 to February 2008. The details of the honeypots, the exact methodology used and a study of demographics of the spam profiles are in our previous work (Webb, Caverlee, and Pu 2008).

3.2 Datasets

To use supervised learning, we need to provide spam and legitimate (non-spam) labels for a training set of profiles. We assign labels based on the method of collection with profiles collected via the honeypots being marked as spam and profiles collected via the top 8 or random sampling strategies as non-spam. As it was possible that during the top 8 or random sampling some spam profiles may have been inadvertently crawled, we ran various heuristics to detect spam profiles within these collections. For example, we used features from some of the previous work (Zinman and Donath 2007; Markines, Cattuto, and Menczer 2009) in our detection of spam profiles, and we looked for keywords in free-form text fields with external spammy-looking links. Furthermore, during our classification experiments, we paid close attention to any misclassification and manually verified some of the labels.

Field	***Field Type***	***Description***
Age	Categorical	
Gender	Categorical	
Marital Status	Categorical	
Smoke	Categorical	Does smoke?
Drink	Categorical	Does drink?
Kid	Categorical	Want kids?
Zodiac	Categorical	Zodiac sign
Education	Categorical	Level of education
Orientation	Categorical	Sexual orientation
About Me	Free-form text	

Table 1: Subset of fields parsed from a MySpace profile and a brief description of non-obvious fields.

Classifier	***Algorithm Type***
AdaBoostM1 (w/ DecisionStump)	AdaBoost
J48	C4.5 Decision Tree
SMO (w/ PolyKernel)	Support Vector Machine
Multilayer Perceptron (MLP)	Neural Network

Table 2: List of classifiers used with categorical features.

We created two datasets for classification based on the above features and labels: the *top 8 dataset* includes a random sample of 15,000 non-spam (legitimate/ham) profiles from the top 8 sampling and all 1,496 spam profiles from the honeypot profiles; the *random dataset* includes a random sample of 15,000 legitimate profiles from the random sampling and 1,496 spam profiles from the honeypot profiles. The rationale for using a subset of all the data available is that as legitimate profiles are so dominant, a majority classifier (i.e. a classifier which picks the dominant label for all profiles), would achieve over a 99.8% accuracy rate.

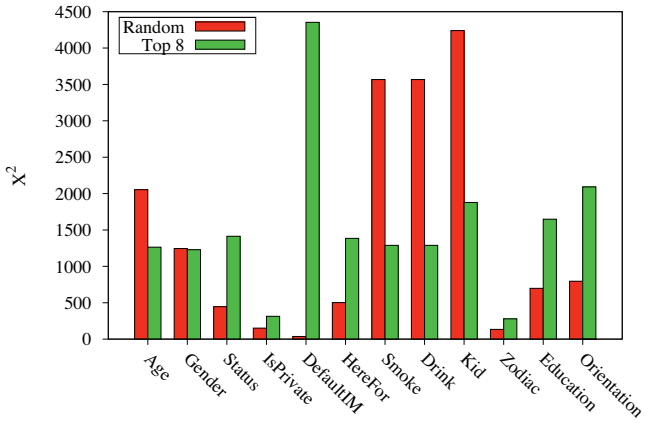
3.3 Feature extraction

Using a custom parser, we extract fields from the MySpace profiles. A subset of these fields can be found in Table 1 along with a brief description and whether we treat the field as a categorical feature or a free-form text feature. Most of the categorical data can, with minimal processing, be used as categorical features for classification. Using a bag-of-words approach, a free-form text field is broken into multiple features (with each word being a feature) after the application of stemming and the removal of stop-words. More details on the features used can be found in Section 4.

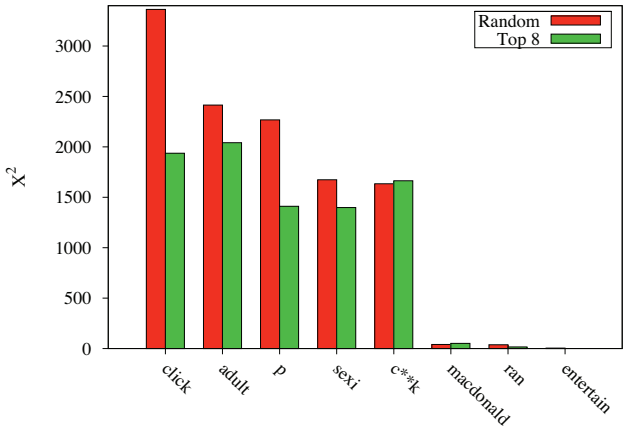
3.4 Classification Setup

A standard set of machine learning algorithms are used from Weka (Witten et al. 1999) 3.6.1. For categorical fields, we use the classifiers listed in Table 2. For free-form fields, due to our independence assumption, we use a naïve Bayes classifier. All classifiers are used with their default settings and are run with 10-fold cross-validation to avoid over-fitting.

We compare classifier performance using the false-positive (FP) errors, false-negative (FN) errors, and area under the curve (AUC) as our evaluation metrics. The FP and



(a) χ^2 test for categorical features



(b) χ^2 test for a sample of free-form text features

Figure 2: Discriminatory power of features measured by the χ^2 test for both categorical features and free-form text features.

FN represent the number of spam instances classified as non-spam and the number of non-spam instances misclassified as spam, respectively. The AUC can be intuitively thought of a measure of how good the detection trade-off between FPs and TPs. It is calculated based upon the received operator characteristic (ROC) curve, which measures the trade-off between FP-rate and TP-rate. Using a classifier which produces a probability of an instance belonging to a particular classes, the trade-off can simply be adjusted by varying a decision threshold, adjusting one's tolerance towards higher FPs and TPs or vice-versa. Examples of such classifiers are neural networks and naïve Bayes. ROC curves can also be applied to discrete classifiers (output binary decisions instead of probabilities), but this usually results in a graph with a single point. Due to the skewed number of positive (non-spam) instances versus negative (spam) instances, the AUC might not always be proportional to the total number of FPs and TPs because it relies on the rate of such FPs and TPs.

4. Feature Analysis

Before looking at classification, we take a closer look at the zero-minute features available for classification. These features based on static profile content are evaluated on their discriminative power and their robustness to adversarial classification.

Discriminative power of a feature can be seen as how strong of a signal it provides in determining the resulting class. To compare the discriminative power of features, we use the χ^2 test, which measures the lack of independence between a feature f and the class c. The χ^2 test takes into account values from a two-way contingency table between f and c, which represents the number of times f and c co-occur, the number of times f occurs without c, and so-forth. Higher values of the χ^2 test indicate a stronger discriminative power.

Robustness to adversarial classification, to a certain extent, is a subjective measure that we manually assign. It is based upon how easy or difficult it would be for an adversary to change a feature in a spam profile in order to evade a classifier. To evade a classifier, the features need to blend in with the distribution of non-spam profile values without reducing the effectiveness of the spam profile. This can be seen as how easy it is for an adversary to "switch off" a signal.

The reason we use these two characteristics to distinguish features is that a highly discriminatory feature with low robustness would yield very good classification results, but it would not be very useful over a long period of time against an adversary. This is because a feature that is not robust could be easily removed by an adversary and would likely degrade the classification results due to its high discriminatory power. Ideal features would have both a high discriminative power and high robustness to adversarial classification.

4.1 Categorical features

Categorical features are obtained by extracting values from the respective categorical fields. Categorical fields with text values such as "Gender" and "Orientation" are converted to nominal values simply by assigning each distinct text value a nominal counterpart. Categorical fields with numeric values such as "Age" are left as numeric features. Two fields which are treated in a binary fashion are "IsPrivate" and "DefaultIM" because the only possible values for those fields are true or false. Although fields like "Smoke", "Drink", and "Kid" could also be treated in this manner, we choose to make a distinction between whether or not those fields are unanswered by assigning a value of "undef" if the field is not answered. If a numeric field is unanswered, it is assigned a value of "-1".

Discriminatory power Figure 2(a) shows the results of the χ^2 test on the Categorical features for the random and top 8 datasets. Each feature is represented by a cluster of two bars on the x-axis (one bar for each dataset), and the y-axis represents the χ^2 test score.

The features with the highest overall discriminatory power for both the random and top 8 datasets were "Kid", "Smoke", "Drink", and "DefaultIM". The "Kid", "Smoke", and "Drink" features are highly discriminative because over 95% of the spam profiles left this value blank (i.e., they have a value of "undef"), whereas only 15-25% of the non-spam profiles in the random dataset and 35-50% of the non-spam profiles in the top 8 dataset had these fields blank. The fewer non-spam profiles in the random dataset having this blank explains why the discriminatory power was higher as compared to the top 8 dataset. The "DefaultIM" feature had a very high discriminatory power in the top 8 dataset because 1% of the non-spam profiles left the field blank, whereas the spam profiles and the random dataset non-spam profiles had a mix of having a blank value or being filled out. It is surprising to see the "Age" feature as a good discriminator. However, we found that most non-spam profiles in the random dataset are under the age of 30 (with a peak at 17 years), and non-spam profiles in the top 8 dataset are under the age of 33 (with a peak at 18 years). The spam profiles, on the other hand, mimic profiles of young women with ages between 17 and 34 and a peak at 24 years (with over 85% between the age of 21 to 27).

Categorical features with the lowest overall discriminatory power for both the random and top 8 datasets were "Zodiac" and "IsPrivate". The reason "Zodiac" has a very low discriminatory power was that it seemed that the spam profiles had randomly chosen a zodiac sign with approximately 8% of profiles falling into each of the zodiac star signs. All the spam profiles had "IsPrivate" set as false, which coincided mostly with the non-spam profiles in the random and top 8 datasets.

Robustness of features As the categorical features are simply values of fields, most of them can be easily changed. The robustness, in this case, mainly comes from if changing a value of a field, and thereby value of a feature, would make the spam profile less effective (i.e., getting fewer users to click on a link). As mentioned in Section 2, most spam profiles portray young, single women and present other characteristics that make them more likely to be successful in spamming. Thus, the features "Age", "Gender", and "Status", would have a high robustness as they must take on certain values to be effective. Examples of low robustness features are "Kid", "Smoke", "Drink", and "DefaultIM".

4.2 Free-form text features

To convert the free-form text fields into features, we combine all the free-form text fields and use a bag-of-words approach where each word is treated as a binary feature (present or not). Before being treated as a feature, each word is stemmed and checked if it is a stop-word. Porter's stemming algorithm (Porter 1980) first reduces the words to their root, e.g., "jumping", "jumper", and "jumped" are all reduced to their root "jump". Stop-words are words commonly used in a natural language and do not contain any significant meaning. We remove these words as they can confuse classifiers due to their high occurrence. Examples of stop words are "a", "the", and "it".

We also removed words which had less than 20 occurrences (a tunable threshold) in the entire dataset, leaving us with 2,289 words and 3,535 words in the random and top 8 datasets respectively.

Not all profiles contained free-form text features because they were blank or private profiles, and a few contained unparsable free-form fields. In the random dataset, approximately 9000 non-spam profiles (55%) did not contain any text in the free-form fields; similarly, in the top 8 dataset, approximately 4300 non-spam profiles (26%) did not contain any text in the free-form fields. There were 789 spam profiles which did not contain any text in the free-form fields.

Discriminatory power Once again, we use the χ^2 test to find the most discriminative features separating the spam and non-spam classes. Figure 2(b) shows some of the words with the highest and lowest discriminatory power using the χ^2 test. As expected, the words with the highest discriminative power are spammy words because these words likely occur in most spam profile free-form text but not in legitimate profile free-form text. A few of the non-spam words that do occur in the most discriminative top 20 words are words used in conjunction with spammy terms such as [check out my] "picture", "watch" [pictures/videos], and "heard" [of a drug].

Although the individual χ^2 test scores of the most powerful discriminative free-form text features is lower than the categorical, we have over 2,000 free-form text features present for each dataset. Some features might be co-dependent, but we assume independence as this a standard assumption when dealing with text classification.

Robustness of features As previously mentioned, as most of the free-form text features with the highest discriminatory power are spammy words, an adversary could change the free-form text to not include spam words. This isn't very practical as: a) removing spam tokens from the free-form text fields would make the spam profiles less effective, and b) to be most effective, the free-form text fields must be well written, which requires spammers to manually rewrite the text. Although an adversary might attempt to replace the words with the highest discriminatory power with a synonym or another form of camouflage (e.g. replacing "click" with "c1ick"), our detection techniques could similarly be tuned to detect synonyms and adjust for this.

To be conservative, we assign 30% of the most discriminatory words a low robustness, as even with the low practicality for the adversary, these words are most likely to be changed first. Words after this are assigned a medium to high robustness.

5. Analysis of Classification Results

We separately evaluate the effectiveness of using categorical and free-form text features, followed by an evaluation using a combined classifier. As a baseline, we compare our results to a basic majority classifier, which on both datasets results in a 90.9% accuracy with an AUC of 0.499. The majority classifier classifies all profiles as non-spam, which results in 1496 (9.1%) false-positives and 0 false-negatives.

Classifier	*Feature Set*	*AUC*		*False-Positives*		*False-Negatives*	
		Random	Top 8	Random	Top 8	Random	Top 8
Majority Classifier (Baseline)	N/A	0.499	0.499	1496	1496	0	0
AdaBoostM1 (w/DecisionStumps)	Categorical	0.991	0.982	98	1347	151	4
J48	Categorical	**0.994**	**0.994**	**24**	**23**	**55**	**85**
SMO (w/PolyKernel)	Categorical	0.986	0.984	34	34	66	136
Multilayer Perceptron (MLP)	Categorical	0.994	0.992	24	37	48	98
Naïve Bayes	Free-form Text	0.718	0.886	849	866	635	71

Table 3: Results of classification based on the different feature sets.

Table 3 shows the result of classification of social spam profiles on the random and top 8 datasets. Results in bold highlight the best results for a particular classifier and measure (averaged over both datasets).

5.1 Categorical features

We first look at the results of the classifiers that use only the categorical set of features. Based on Table 3, we see that the J48 classifier performs the best, misclassifying only 79 and 108 profiles (accuracy of 99.6% and 99.4%) on the random and top 8 dataset respectively. This is followed closely by MLPs. SMO (with PolyKernel) and AdaBoostM1 (with DecisionStumps) perform well in terms of AUC but have a lot of FPs and FNs, comparatively.

J48 performs the best as there are some very discriminatory features, which allow it to quickly detect "obvious" spam or non-spam in the early levels of the tree, followed by refinement at the lower levels. As an example, the root of the J48 tree for the random dataset simply checks if "Smoke" is "undef" and if not, detects the instance as non-spam. This correctly classifies 78% (11,734 of 15,000) non-spam profiles with an error of less than 1%. For the same reason, we expected AdaBoostM1 (with DecisionStumps) to perform much better than it did. DecisionStumps are simply single branch trees, and they should have been able to classify the spam profiles even using only 10 iterations of boosting. In fact, for the default ROC threshold used by Weka, the AdaBoostM1 (with DecisionStumps) classifier performs only marginally better than the baseline majority classifier in terms of the number of misclassified spam profiles on the top 8 dataset.

Most classifiers also tended to perform better on the random dataset than on the top 8 dataset. This can be explained by the profiles in the top 8 dataset being more similar to the spam profiles (based on the overall lower χ^2 test score), causing more misclassifications.

On investigating the misclassifications, we found that most of the false-positives were due to a particular "strain" of spam profiles, which were more complete and contained legitimate looking categorical features. Additionally, unlike other spam profiles, this strain seemed to be selling male enhancement drugs, and the profiles were in a relationship. It is likely that the classifiers attributed the strain of spam profiles as noise and avoided over-training.

Empty or partially filled out profiles resulted in a small number of false-negatives for all the classifiers, followed by additional false-negatives for some classifiers that mistook partially filled out profiles of young women as spam. Some of the empty profiles were the result of profiles "Undergoing construction" at the point of crawling.

5.2 Free-form Text features

We now explore the classification of profiles based on the free-form text-based sections of their profiles. From Table 3, we see that a naïve Bayes classifier using only free-form text features does poorly in comparison to the classifiers using only categorical features.

The large number of false-positives were due to 789 of 1496 spam profiles (53%) containing blank free-form text sections (or the free-form text sections not being parsable), which resulted in the majority non-spam label applied to such profiles. From a sample of profiles, we manually investigated and found the false-negatives were mainly due to non-spam profiles using spammy tokens in their about me section. For example, we looked at the most discriminatory spam features in a subset of non-spam profiles used in the random dataset; we found 47, 36, 457, and 90 occurrences of the spam tokens "click", "adult", "sexi", and "c**k", respectively.

5.3 Categorical and Free-form Text features

A natural question which arises is whether the independent predictions of the classifiers on the categorical and free-form text features can be combined to improve our classification results. The aim here would be to use classifiers which are best at classifying their respective feature set (and allow for co-dependence in the case of free-form text features).

To do this we experimentally examine the results of applying the 'AND' and 'OR' operators to the predictions of the classification based on the default ROC threshold. When classifying spam profiles, in the case of the 'AND' operator, only if both categorical and free-form text classifiers predicted the profile to be "spam" will the profile to be marked as spam. In the case of the 'OR' operator, either categorical or free-form text feature classifier predicting the profile as "spam" will result in the profile being marked as such. To reduce the number of false-negatives, we only considered the outcome of the free-form text classifier if the text feature set was not empty.

Intuitively, the 'AND' operator should reduce the number of false-negatives as both classifiers will have to predict a "spam" classification before one is assigned. The 'OR' operator on the other hand should reduce the number of false-positives as either classifier predicting "spam" will cause

Classifier	*Feature Set*	*AUC*		*False-Positives*		*False-Negatives*	
		Random	Top 8	Random	Top 8	Random	Top 8
Majority Classifier (Baseline)	N/A	0.499	0.499	1496	1496	0	0
AdaBoostM1 (w/DecisionStumps)	Categorical	0.916	0.980	1496	1496	0	0
J48	Categorical	**0.942**	**0.987**	**219**	**39**	**953**	**225**
SMO (w/PolyKernel)	Categorical	0.500	0.965	1496	76	0	285
Multilayer Perceptron (MLP)	Categorical	0.912	0.988	950	53	807	237
Naïve Bayes	Free-form Text	0.718	0.886	848	863	612	67

Table 4: Results of classification under assumption of an adversary. Features with high discriminatory power and low robustness are removed.

"spam" to be assigned. On the flip-side, the 'AND' operator will increase the number of false-positives due to more "spam" being classified as "non-spam" and similarly the 'OR' operator will increase the number of false-negatives.

We do not show results here because, as intuitively explained, applying the above operators to predictions simply reduced either false-negatives or false-positives but not both simultaneously. We hypothesize that building a new classifier over the combined categorical and free-form text features would alleviate this problem and leave investigating this as future work.

6. Analysis of Adversarial Classification Results

Previous evolutionary studies (Irani et al. 2008; Pu and Webb 2006) have shown that once techniques are developed to counter particular types of spam, spammers evolve and deploy new types of spam to bypass those techniques. Assuming the presence of an adversary with an ability to probe our classifier for most discriminatory features, we evaluate the effectiveness of our classifiers with the low robustness and high discriminatory power features removed—as adversaries are likely to remove most spammy easy-to-change (low-impact on spam) features first. To emulate this, we use the χ^2 test score of the features over both datasets and remove the highest discriminatory features with low robustness.

As a baseline, we again use a basic majority classifier, which on both datasets results in a 90.9% accuracy with an AUC of 0.499. The majority classifier classifies all profiles as non-spam, which results in 1496 (9.1%) false-positives and 0 false-negatives—the results remain the same as we do not change the number of profiles, only reduce the set of features.

Table 4 shows the result of classification of social spam profiles, assuming an adversary, on the random and top 8 datasets. Results in bold highlight the best results for a particular classifier and measure (averaged over datasets).

6.1 Categorical features

For the categorical set of features, the four features with the strongest discriminatory power and low robustness are "DefaultIM", "Smoke", "Drink", and "Kid". Based on Table 4, we see that the J48 classifier performs the best, misclassifying 1172 and 264 profiles (accuracy of 92.9% and 98.4%) on the random and top 8 dataset respectively. AdaBoostM1 (with DecisionStumps) and MLPs, also do well based on the resulting AUC, but AdaBoostM1 (with DecisionStumps) has a significantly higher number of misclassifications. SMO (with PolyKernel) performs the poorest, with the classification on the random dataset being similar to that of the baseline majority classifier.

J48 performs the best, for reasons similar to it doing the best at regular classification as well—it is able to use the highest remaining discriminatory features to detect "obvious" spam or non-spam in the early levels of the tree. As the most discriminatory features have been removed, the J48 decision tree resulting from this classification is more complex with the number of leaves in the tree growing by about 45%. Even with the best classification results, the J48 classifier misclassifies over 1400 profiles (4.3%) on both advesarial datasets, as compared to a misclassification of less than 200 profiles (0.6%) on both non-advesarial datasets. Although the lack of expressiveness of the PolyKernel is the likely reason for SMO with PolyKernel classifier performing badly on the random dataset, we are investigating this fruther.

On investigating the misclassifications we found that in-addition to the false-positives incurred by the non-adversarial classifier, profiles of women around the age of 30 started being marked as legitimate. Additional false-negatives incurred by the adversarial classifier were due to partially complete profiles of women which had filled in certain categorical fields which previously would have identified them as non-spam, but which are no-longer considered.

6.2 Free-form Text features

To pick which free-form text features to remove we averaged the discriminatory power of the free-form text features between datasets and chose features with the strongest discriminatory power and lowest robustness. We set a limit of disregarding 900 features per dataset, which left us with 1,389 features and 2,635 features in the random and top 8 datasets respectively. The results of running the classifiers over the new set of features, gives us the results shown in Table 4.

The naïve Bayes classifier performs similarly over the full feature set. This indicates that although the features with the strongest discriminatory power were removed, there is a large enough set of weaker features available for classification.

Once again, a large number of the false-positives were due to spam profiles not containing a free-form text section,

and the false-negatives due to spammy tokens being used in legitimate profiles.

7. Conclusion and Future Work

In this paper, we present an exploration of the limits of classifying social spam profiles on MySpace. Specifically, we focus on zero-minute fields or static fields present when a profile is created to try and determine whether such a technique would be feasible in preventing spammers from gaining a foothold inside a social network to send spam. We start with an analysis of features that includes a statistical analysis to determine the discriminatory power of a feature as well as the robustness of features to adversarial attack. Then we compare the effectiveness of classifiers that were built with these features.

Our classification results show that a standard C4.5 decision tree performs the best with an AUC of 0.994 and an average of approximately 24 false-positives and 70 false-negatives. The number of false-positives is acceptable as most of the non-spam profiles misclassified were partially filled out profiles. A subset of the false-positives are a new "strain" of spam profiles that expose one of the weaknesses of our classification technique—namely, a weakness in detecting new types of spam profiles without sufficient training examples. To further explore the potential of new types of spam profiles evading our classifiers, we use our earlier statistical analysis to disregard features that are most likely to be camouflaged by an adversary and analyze the effects this has on the classifier's performance.

We believe that the classification results are positive enough to justify building a system in which a classifier can automatically detect most spam profiles with a high confidence, and mark others as gray profiles. Collaborative filtering (with a lower threshold) or administrator reporting, can be used to confirm the correct class label for these gray profiles, which in turn could be used in a feedback loop to improve future classification results.

Acknowledgements

This research has been partially funded by National Science Foundation grants ENG/EEC-0335622, CISE/CNS-0716484, CISE/CNS-0855067, AFOSR grant FA9550-06-1-0201, NIH grant U54 RR 024380-01, Wipro Technologies, Fujitsu Labs, and Georgia Tech Foundation through the John P. Imlay, Jr. Chair endowment. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding agencies and companies mentioned above.

References

Caverlee, J., and Webb, S. 2008. A large-scale study of MySpace: Observations and implications for online social networks. *Proceedings of ICWSM*.

Caverlee, J.; Liu, L.; and Webb, S. 2008. Socialtrust: tamper-resilient trust establishment in online communities. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, 104–114.

Gyongyi, Z.; Garcia-Molina, H.; and Pedersen, J. 2004. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 587.

Heymann, P.; Koutrika, G.; and Garcia-Molina, H. 2007. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing* 36–45.

Irani, D.; Webb, S.; Giffin, J.; and Pu, C. 2008. Evolutionary study of phishing. *eCrime Researchers Summit, 2008* 1–10.

Lauterbach, D.; Truong, H.; Shah, T.; and Adamic, L. 2009. Surfing a web of trust: Reputation and reciprocity on couchsurfing.com. *Computational Science and Engineering, IEEE International Conference on* 4:346–353.

Markines, B.; Cattuto, C.; and Menczer, F. 2009. Social spam detection. In *Proceedings of the Fifth International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2009)*.

Noll, M.; Yeung, C.; Gibbins, N.; Meinel, C.; and Shadbolt, N. 2009. Telling experts from spammers: expertise ranking in folksonomies. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 612–619.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

Porter, M. 1980. An algorithm for suffix stripping. *Program* 14(3):130–137.

Pu, C., and Webb, S. 2006. Observed trends in spam construction techniques: a case study of spam evolution. In *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*.

Webb, S.; Caverlee, J.; and Pu, C. 2008. Social Honeypots: Making Friends With A Spammer Near You. In *Proceedings of the Fifth Conference on Email and Anti-Spam (CEAS 2008)*.

Webb, S.; Caverlee, J.; and Pu, C. 2009. Granular computing system vulnerabilities: Exploring the dark side of social networking communities. In *Encyclopedia of Complexity and Systems Science*. Springer. 4367–4378.

Witten, I.; Frank, E.; Trigg, L.; Hall, M.; Holmes, G.; and Cunningham, S. 1999. Weka: Practical machine learning tools and techniques with Java implementations. In *ICONIP/ANZIIS/ANNES*, volume 99, 192–196.

Zinman, A., and Donath, J. 2007. Is Britney Spears spam? In *Proceedings of the Fourth Conference on Email and Anti-Spam (CEAS 2007)*.