Soundcloud Spam Analysis

Pedro M. Sosa, Armin Ak, Metehan Ozten

December 5, 2016

Abstract

SoundCloud is a popular music sharing platform with over 40 million users. Our goal was to analyze the spam incidence and behavior within the platform. To achieve this we recollected around 20 million users and proposed a set of features to properly describe the dataset. We provide a more in-depth look at the behaviors of Soundcloud users and afterwards, we used a combined effort between a Batch-based K-Means clustering algorithm and manual labeling heuristics to find distinct groups of users. Furthermore, we trained a neural network to distinguish legitimate vs. illegitimate users and we published it as a web application at soundcloud.pw

1 Introduction

SoundCloud is a free sharing music platform with over 40 million users [1]. As with many other social platforms it is not uncommon to find bots or spam accounts [2]. However, because of the limited nature of this social medium, they are harder to spot and their behaviors/origins are still unstudied.

During our research our main goal was to recollect, analyze, and classify the spam found inside SoundCloud. Furthermore, we seek to provide a descriptive feature set, clustering algorithm and trained neural network to aid in the detection of this spam.

We initially created a crawler that allowed us to recollected 20 million users. Afterwards, we analyze user behaviors to better refine our definition of spam. We ultimately use both Batch K-Means and a Neural Network to find clusters of spam. Additionally, we bundled the Neural Network into an online application (http://soundcloud.space) that allows anyone to check whether a user is legitimate or not. Lastly, we analyze the origins and behaviors of the types of spam that we were able to find.

2 Data Mining

To fetch as many users from SoundCloud in the most efficient manner we had to first analyze the inner workings of the site. We found that user's IDs were given sequentially and that the upper bound of these IDs seemed to be around 200 million. The second discovery was that although SoundCloud stopped providing developers with API keys to access their API calls, they did actively use a single public "API key" in their website/app. Therefore by knowing that specific key, we were able to freely enumerate through large amount of users while not being identified.

For maximum efficiency we built a distributed crawler that could be run from many different machines at a single time. The crawler itself was built in Python and Dockerized to allow ease of deployment and maximum scalability. Furthermore, the crawler used a ticketing system from a Ruby on Rails server that would designate a random range of 1,000 users to crawl. This ensured that even if we did not obtain the complete spectrum of users, we had an evenly distributed set. Finally, the crawlers would publish their results to a centralized relational database.

Our crawler ran at approx. 1,000 users per hour and we managed to obtain approximately 20 million users.

3 User Analysis

3.1 Definition of Spam

Initially we choose a very loose definition of spam, as to not induce a subtle bias in our clustering and analysis. As such, we defined a *spam user* as a user that seem to be a duplicate of others, points to websites of untrusted nature, or otherwise seem to intentionally seek malicious purposes.

3.2 SoundCloud Spam Cleanup

As with many big internet social communities, SoundCloud must have their own private spam prevention/elimination techniques aside from simply obtaining reports from users. While sequentially obtaining the users, we found many empty IDs which could point to potential spam users who were caught and deleted. Furthermore, out of roughly 10000 spam users created (or last modified) during March 2016 - October 2016 that we obtained, we noticed that 65% of them were deleted around November 2016. This means that it took at most 8 months for some spam users to be deleted. We believe that our work could help catch these spam users faster and more accurately.

3.3 Feature Selection

Before we ran any clustering algorithms we first set out to select the features that better described the users. To accomplish this we manually probed and visualized our dataset to get a better understanding of it.

Community Engagement One of the first things we realized about the division of users in SoundCloud is that 42.0% are *dis-engaged* users. We refer to dis-engaged users as users that are not engaged in the Soundcloud community, meaning they have not commented, liked, favorited, playlisted, or published songs. It is important to point out that since we can't measure how many songs a user has listened to, it is possible that some of these dis-engaged users are still actively using the streaming services.

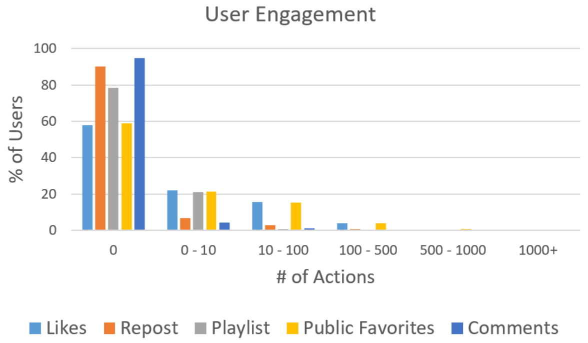

Figure 1: Breakdown of activities describing user behavior in SoundCloud.

Furthermore only 6.2% of the users have actually published tracks, and only 0.15% of users pay for a SoundCloud subscription (Pro & Pro Plus).

Followers, Following, Descriptions & URLs We found interesting clusters when analyzing the ratio of followers to following. While there was an expected large group of people that had very little followers/followings (39% of users follow no one; 45% of users are followed by no one), we found two clusters of users that followed around 250-350 users, but had little to no users following them. Manually inspecting these users, we found multiple spam accounts.

To further tighten these clusters, we tried discriminating users by whether or not they provided a Description or a URL. This proved to be beneficial as most of the spam provides both, and as seen in Figure 2 the spam clusters become more obvious.

We also found that only 0.68% of users share URLs, and only 0.004% of them use URL shorteners. As such, we manually inspected users using URL shortness and found that it was almost all exclusively spam.

Features Selection After we had manually visualized the data, understood how users were interacting with the system, and even encountered a few different types of spam, we selected a broad set of Features that could help thoroughly describe users.

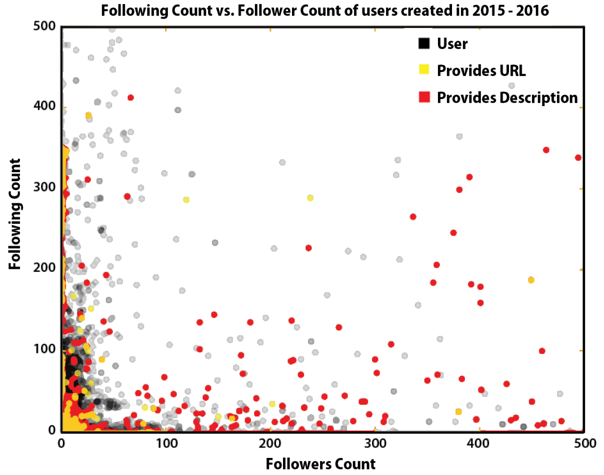

Figure 2: Finding clusters by filtering by # of Followers, # of Following, URL, & Description.

- **Followers:** # of users a user is following.
- **Following:** # of following users a user has.
- **Published:** # of following users a user has.
- **URL:** Whether user provides a link.
- **ProfDesc:** # of profanities in description.
- **ProfTitle:** # of profanities in website title.
- **DupDesc:** # of users with the same description.
- **DupURL:** # of users with the same link.
- **Activity:** Likes + Reposts + Playlists + Comments + Favorites.
- **Subscription:** Whether the user pays for SoundCloud.
- **ShortenURL:** Whether the link point to a url shortner.

4 K-Means Clustering

4.1 Methodologies

To distinguish different types of users into distinct clusters we first pre-processed our data and represented them with the features described above. We tested and pruned the features that proved irrelevant, and lastly, we ran Batched-Based K-Means on our dataset.

4.2 Feature Pruning

Given the features selected above, we used our k-means algorithm to cluster data-points in up to 11-dimensional space. Since our initial clusters generated were not very tight, we proceeded to fine-tune and prune our features.

Of the 11 initial features, we pruned "Subscription", "Duplicate Website" and "Profane Title". We determined this by observing the locations of the centroids returned by the k-means algorithm. Features which always located their center at zero proved unnecessary and simply introduce noise in our algorithms. For further verification, we ran Correlation-based Feature Subset Selection [3] on WEKA [4] which also recommended eliminating those same features.

Furthermore, we performed mean removal and z-value scaling on our feature values to improve accuracy and found that the resulting clusters immediately became reflective of the types of users one would find within the SoundCloud (e.g. Celebrities, Super-Fans, inactive users, spammers).

4.3 Choosing # of Clusters

The Elbow Method Before running K-Means clustering, we were required to define the total number of clusters (the "k"-value). There are multiple heuristic methods that one can use to define the optimal "k"-value for a particular algorithm/dataset. In our case, we used the popular "elbow" method [5] which is performed by graphing the sum of squared errors (SSE) for all the clusters as a function of k, and selecting one of the two value following the largest drop in SSE. Notice that by having the ability to choose the value immediately following the largest drop in SSE or the subsequent value, we are able to introduce some level of non-determinism. In our case, as seen in Figure 3, we could choose k to be either 4 or 5.

Natural Clustering Another way to select a k-value is to choose a value that represents the natural amount of clusters represented in your feature space. When using k=5, our algorithm groups users into 5-

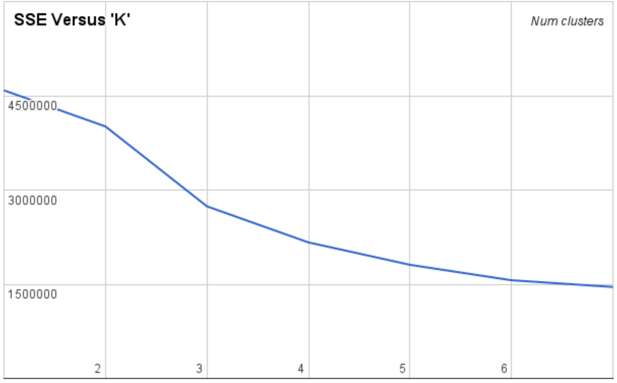

Figure 3: SSE as a function of ’k’.

groups that closely resembled the following types of SoundCloud users:

1. Regular (inactive users)
2. Popular spammers (determined by number following)
3. Unpopular spammers
4. Highly Active Users
5. Celebrities (similar to Highly Active Users except centroid Followers value > centroid Following value)

Thus it was no coincidence that the elbow method found 4 and 5 to be the appropriate number of clusters, seeing as they so appropriately adapt to the natural descriptions of users in SoundCloud.

4.4 Results

The following are the results of clustering a random set of 7 million users. We labeled each group by what seemed to be most representative within each of them, and provided the # of total users in the cluster vs. the # of spammers in the cluster.

4.4.1 Results (k=4)

When using k=4, our results were the following:

1. Regular Users (Inactive and Active)
   - Size: 564319
   - Spammers: 587 (<1%)

2. Popular Spammers
   - Size: 902
   - Spammers: 902 (100%)

3. Unpopular Spammers
   - Size: 7897
   - Spammers: 7897 (100%)

4. Celebrities
   - Size: 102
   - Spammers: 1 (<1%)

4.4.2 Results (k=5)

When using k=5, our results were the following:

1. Regular Users (Inactive)
   - Size: 562121
   - Spammers: 587 (<1%)

2. Popular Spammers
   - Size: 891
   - Spammers: 891 (100%)

3. Unpopular Spammers
   - Size: 7897
   - Spammers: 7897 (100%)

4. Celebrities
   - Total Users: 1620
   - Spammers: 12 (<1%)

5. Active Users (Super Fans)
   - Size: 691
   - Spammers: 0 (0%)

We tested the performance of our algorithm by comparing the ”spam clusters” generated by our algorithm to the groups of users that we manually identified as spam.

5 Neural Network

In addition to our K-means clustering we wanted to build and train a simple and easy to use service to quickly distinguish spam users in SoundCloud. To do this we built a Neural Network (NN) using the Keras and Tensorflow libraries for Python, and wrapped it around a Flask web interface that allows users to do quick queries. We are currently hosting this service on soundcloud.pw and soundcloud.space

Our NN is configured as a Feed-Forward NN with 1 hidden layer of 250 ReLU nodes, and an output layer using the sigmoid function as the activation function. We trained it using manually labeled data that we obtained with our Clustering experiments. We made sure that such test cases where in fact well balanced and provided a good mix of different types of users and spam.

After training, we tested it against 20,000 randomly chosen users (evenly split between legitimate users and spam), and obtained an accuracy 85%.

6 Type of Spam and Origin

6.0.1 Pornography Related Spam

The majority of the spam we found was pornography related spam. This type of accounts tended to used URL shorteners that linked to other redirection sites before landing on pornographic websites. The two most common sites where *mrbtrack.com* and *supergoood.ru*; Both which are registered in Russia and are hosted in Russia and Germany respectively. We also found a set of links that followed a similar pattern of 9 random letters and always finished with an African top level domain (e.g *sungzkfpm.ga*). Unsurprisingly, all these domains were recently created in 2016 and used WhoisGuard.

6.0.2 "Fake" Users

We found many users that had identical or near-identical descriptions and were created within milliseconds of each other. This types of account seemed to follow a bigger pattern that would slightly randomize the descriptions and choose legitimate looking names, locations, etc. This made these types of users blend easily with real users. We are still uncertain about what purpose they serve, as they have no activity and provide no URL or description that would otherwise imply a malicious behavior.

7 Future Work

We found that to further build better and more accurate results it is advisable to crawl the latest created users. We found that since most spam is created in batches within seconds of each other, one could use these clustering algorithm or the neural network presented and through continuously feeding new information fine-tune it for better results.

Another way to improve spam detection could be to expand or generate new features that could better describe user behavior. One such method could be to focus on the relationships between following and followers. By following a similar methodology as described on previous works [6], graphing user relationships might be a good way to visualize and distinguish spam.

Another interesting followup would be to continue checking the "Fake" Users we mentioned before to see if their behavior changes, as perhaps they are being used as placeholders for further spam.

8 Conclusions

During our research we were able to crawl 20 million users from SoundCloud and found around 11000 spam accounts. We provided an analysis of user interaction and behavior within SoundCloud which could allow researchers to further understand the use cases within this platform. We proposed a descriptive set of features to describe our dataset specifically designed such that they could help distinguish spam users from legitimate users. Furthermore, we were able to tightly cluster these accounts into 4-5 groups using the Batch-Based K-means algorithm. We also coded and trained a Neural Network to distinguish user's legitimacy and published as a web app hosted at http://soundcloud.pw. We believe that using these two machine learning mechanisms we can predict and distinguish spam accounts in SoundCloud with moderately high accuracy.

Lastly, because we wish to provide other researchers with the opportunity to test their own ML algorithms and provide their own analysis we will release our dataset, crawler, neural network, and all other relevant code at https://github.com/pmsosa/CS276-Project.

References

[1] A. J. Sinclair, "Predicting music genre preferences based on online comments," 2014.

[2] M. Flores and A. Kuzmanovic, "Searching for spam: detecting fraudulent accounts via web search," in *International Conference on Passive and Active Network Measurement*, pp. 208–217, Springer, 2013.

[3] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," 1998.

[4] S. R. Garner, "Weka: The waikato environment for knowledge analysis," in *In Proc. of the New Zealand Computer Science Research Students Conference*, pp. 57–64, 1995.

[5] R. Tibshirani and G. Walther, "Cluster validation by prediction strength," *Journal of Computational and Graphical Statistics*, vol. 14, no. 3, pp. 511–528, 2005.

[6] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, and X. Li, "An empirical study of collusion behavior in the maze p2p file-sharing system," in *27th International Conference on Distributed Computing Systems (ICDCS'07)*, pp. 56–56, IEEE, 2007.