Abstract

With the increasing importance of online communities, discussion forums, and customer reviews, Internet “trolls” have proliferated making it difficult for information seekers to find relevant and correct information. In this paper, we consider the problem of detecting and identifying Internet trolls, almost all of which are human agents. Identifying a human agent among a human population presents significant challenges compared to detecting automated spam or computer bots.

To learn a troll’s behavior, we use contextual anomaly detection to profile each chat user. Using frequency-based clustering, we observe contextual data such as the group’s output over the past hour, the current time, and the user to classify each point as an anomaly. A user with consistent anomalies will be labeled as a troll. We have successfully trained our algorithm using methods on a dataset consisting of 58 million points from the popular Internet social site, Twitch Plays Pokémon. Using MapReduce techniques for processing and user profiling, we are able to classify trolls on 10 features extracted from a user’s lifetime history.

Introduction & Motivation

In Internet slang, a “troll” is a term that was discussed in the Internet by starting arguments or upsetting people, by posting inflammatory, extraneous, or off-topic messages in an online community (such as a forum, blog, chat room, or other discussion), either accidentally or with the deliberate intent of provoking others to respond with information that is neither relevant nor necessary to the discussion [1]. The work done in this project serves as to what is happening in gaming. For example, in online games, the players, especially in computer vision [2], can be extend to anomaly detection in the online setting. During online interactions, increasingly often an automated method or a classification system is at stake. The speaker. University courses use twitter to engage students so this project can be extend to anomaly detection during classes and interviews. We would then be able to filter out irrelevant questions. In Jiang et al. [1], context is used for mass surveillance videos. The time of day, place, day of the week, and number of people all have an impact on how people act. Modeling this as a context helps machines with some deviations regarding what is happening in gaming. See the long spike above 80% for the general period between two games when the stream is offline. Additionally, when the player défier a gym leader (major accomplishment), spam temporarily spikes.

Algorithm & Distance Measures

Anomaly Scoring

We assign a decimal value between 0 and 100 representing our confidence that the point is an outlier, or a troll. This score can be calculated in many ways. In this project, we used the following scoring technique:

1. Distance to 1 nearest neighbor (SKNN)
2. Sum of distances to 6 nearest neighbor (SKNN)
3. Relative NID
4. Local outlier factor (LOF)
5. Multidimensional scaling (MDE)

We experimented with context durations of 60 seconds, 30 seconds, and 5 seconds. Every 60, 30, and 5 seconds, the duration was increased by 60. As the context duration increases, the number of points is reduced, and the context is smoother. We chose to use a context duration of 20 seconds for the data. After applying the model, the context duration changes. The peaks at each point in time indicate which model was most popular in that context. Sometimes this is left, sometimes it is up. It is most likely that the players are attempting to navigate through the world as the frequencies of A and B are low (i.e. not in a menu).

Dimensionality Reduction

Furthermore, we used principal component analysis and selected the first three principal components: $f_1, f_2, f_3$. Since our original matrix was 6 million by 30, we used a modified version of SVD that is more computationally compact: $A \Delta \Sigma^{1/2}$ and then $A \Sigma^{-1/2}$. $A \Delta \Sigma^{1/2}$ is the first of the first three principal components is shown above. A plot of the first three principal components is shown as a comparison.

Background: Twitch Plays Pokémon

Twitch Plays Pokémon ([1]) is a “social experiment” and channel on the video streaming site TwitchTV, consisting of a crowd-sourced attempt to play Game Freak and Nintendo’s Pokemon video game by posting commands sent by users through the channel’s chat room [2]. Users can input any message into the chat but only the following commands are recognized by the bot: 1. Change the goal (such as a forum, blog, chat room, or other discussion) either accidentally or with the deliberate intent of provoking others. Anarchy and democracy refer to the “mode” of the game. In anarchy mode, inputs are executed in pseudo-FIFO order (see next paragraph for more information). In democracy mode, the bot collects user input for 20 seconds after which it executes the most frequently entered command. The bot attempts to execute commands sequentially in a pseudo-FIFO order. Many commands are shipped to the game finish and then an element of randomness is introduced, hence the name pseudo-FIFO. The uncertainty in the queue results in selecting an action at random where the frequency of user input serves as the underlying probability distribution.

Results

A lot of the work is currently ongoing. Results pertaining to the labeling of users in a streaming environment are currently under way and will be presented soon.

Related & Future Work

A lot of the work done here can be extend to anomaly detection in the online setting. During online interactions, increasingly often an automated method or a classification system is at stake. The speaker. University courses use twitter to engage students so this project can be extend to anomaly detection during classes and interviews. We would then be able to filter out irrelevant questions. In Jiang et al. [1], context is used for mass surveillance videos. The time of day, place, day of the week, and number of people all have an impact on how people act. Modeling this as a context helps machines with some deviations regarding what is happening in gaming. See the long spike above 80% for the general period between two games when the stream is offline. Additionally, when the player défier a gym leader (major accomplishment), spam temporarily spikes.

Frequently Asked Questions

Q: There’s a time delay of about 20–40 seconds when a user inputs a command and the bot parses it. How do you deal with this? A: We ignore it. We are concerned with what happens among the users rather than how it relates to the speaker. University courses use twitter to engage students so this project can be extend to anomaly detection during classes and interviews. We would then be able to filter out irrelevant questions. In Jiang et al. [1], context is used for mass surveillance videos. The time of day, place, day of the week, and number of people all have an impact on how people act. Modeling this as a context helps machines with some deviations regarding what is happening in gaming. See the long spike above 80% for the general period between two games when the stream is offline. Additionally, when the player défier a gym leader (major accomplishment), spam temporarily spikes.

Q: Did you do anything differently in democracy and anonymity modes? A: No. Democracy allows for complex inputs such as “2q” and “2i/mutantDown.” Since we don’t want to deal with this complexity, we ignored democracy and anonymity modes when labeling users. We take notes of the current mode at any given time. This is used in our features.

Q: How do you integrate the context, the buttons, users, and messages? A: We created a class diagram below the coding section. A lot of thought went into the design. As a result, we were able to solve any potential issues before the code was written. This also accelerated the development process because since most of the requirements were already specified, the design was followed closely, for the most part. With the class diagram, we were able to define the state of the game and labels. Additionally, methods are not included in the class diagram on the right.