CS 171: Project 2

Pheme:
Birds of a Feather Tweet Together

Authors:
Carl Jackson and Peter Zhang

April 8, 2013

Contents

1 Introduction

1.1 Motivation

Most current visualizations of social data require viewers to already have a subject matter in mind; they allow viewers to search for and investigate specific keywords, hashtags, or locales. However, we think that social data is interesting primarily because it allows us to discover new topics for investigation - topics that are being generated in real time by the humans who are producing the social data. Accordingly, we want to build a visualization that guides a curious but unfocused viewer in identifying undiscovered topics/events that are currently occurring in the real world.

1.2 Goals

In this project, our goal is to investigate high-throughput social data streams, and answer two primary research questions:

1. Can we identify "events" from real-time social data?
2. If we can, what can we learn about such "events" from various social data sources?

To achieve these goals, we have built a visualization that tries to identify current events that are occurring within a user-specified geography (location + radius), using real time social data. Concretely, we pull a live stream of Twitter data being generated in the specified geography, and try to identify clusters of tweets; currently, we cluster primarily by geographical proximity, but hope to cluster using other techniques in project 3.

2 Description of Data

2.1 Source

The ultimate source of the data for our visualizations is Twitter, an online social network and microblogging service that allows users to instantaneously send and receive short messages, called tweets, on various electronic devices. Twitter is one of the most popular, if not the most popular, sites that produces real-time social data; as of 2012, it produced more than 340 million tweets per day [1]. Given our interest high-throughput social data streams,

[1] http://techcrunch.com/2012/07/30/analyst-twitter-passed-500m-users-in-june-2012-140m-of-them-in-us-jakarta-biggest-tweeting-city/

Twitter was a natural data source to focus on.

Using Twitter's API, we pulled tweets, which contained several relevant pieces of information:

1. Geographical location: Twitter users can choose to specify a geographical location to associate with their tweet. This is particularly interesting for users who tweet on their mobile phones, and choose to expose their phone's GPS data to the service.

2. Twitter Screenname: Tweets are associated with the screen name of the user publishing the tweets.

3. Tweet contents: The actual text content of the tweet. This is also available to viewers of our visualization.

4. Hashtags: Twitter users can specify hashtags, or metadata, to associate with their tweet.

5. Tweet timestamp: The time at which the tweet was published.

2.2 Scraping Method

As described above, I retrieved two types of data from Steam, and I wrote a separate script for each type of data. In both scripts, I used python with the pattern library.

2.2.1 Game Usage Over Time

I first focused on just scraping game usage over time, since I wanted to collect several days worth of data, and thus wanted to get this process started as early as possible. Originally, I had intended on scraping data for a full week, but unfortunately I wasn't able to get this script working until Monday 2/25, so I ended up only getting around 2 days worth of data. In addition, I had originally intended to scrape the game attributes over time as well, rather than just game usage numbers, but due to difficulties described in the next section and in the interest of time, I decided to implement that feature in a different script. Theoretically, it is possible that game attributes (especially price) could have changed over the time period I gathered data from; however, given the short time frame of my data, I believe it is safe to ignore this possibility when conducting analysis.

My goal for this script was to scrape the "Top games by current player

count" table at `http://store.steampowered.com/stats/` (screenshot below) for data every hour for a period of days.

However, I couldn't guarantee that my computer would be on continuously for the duration, so I decided to use Amazon's Elastic Computing service. Accordingly, I began by learning about the service, setting up a virtual server through the service, and installing pattern onto my instance.

Next, I wrote and tested my scraping script (usagescraper.py) on my local machine, which scrapes the whole table for current player, game title, and game profile url, and appends this data along with the time when the scrape was conducted to a csv file (steamusage.csv). Finally, I scped this code to my virtual server, and automatically ran the script at 10 minutes past the hour using Cron. After letting it run for around two days, I stopped the script and moved on to the second step.

2.2.2 Game Attributes

After collecting a sufficient amount of game usage data, my next goal was to collect attributes about the games that I had followed. First, I loaded the usage data in Google Refine, and got a unique list of game profile urls of all the games that I had collected data for over those two days (this was more than 100 games, since some games dropped in and out of the top 100 games by current users over time); I saved this list in urllist.csv. Next, I wrote a python script (infoscraper.py) that navigated to each of these game profile pages, and scraped price, sale price, metacritic rating, genre(s), publisher(s), release date, language(s), Entertainment Software Rating Board rating, and game feature(s) and saved them into a csv file (steaminfo.csv); screenshots a game profile are shown below.

I faced three challenges when writing this script.

1. Steam seems to block certain user-agents from accessing their website - in particular, the default user-agent used by pattern would consistently time out when trying to download a profile page. Addressing this was relatively easy; pattern allows you to manually set your user-agent, so I set it to Mozilla firefox.

2. Steam redirects to an age verification page for games with a mature

ESRB rating which could not be easily navigated around. After playing around with the steam website for a while, and using Chrome developer tools, I noticed that one of the variables in the cookie Steam stored was called "birthtime," and I realized I needed to send a fake cookie, with that variable set appropriately, along with my requests to get around the age verification pages. Unfortunately, pattern doesn't allow you to do this, but Beautiful Soup, one of the underlying packages used by pattern, does. Accordingly, I created a new class "URL2" that inherited from the "URL" class in pattern, and overrode the "open" and "download" class functions to allow for manual setting of cookies - making use of the functionality provided by beautiful soup - and used this class in my script rather than the URL class.

3. Some of the games listed in the top 100 games by Steam do not have profile pages; the profile page url listed in the "Top games by current player count" table redirected to the Steam home page. Unfortunately, I couldn't find an alternate profile page for these games, so I just ignored any urls that redirected to the Steam home page.

2.3 Cleaning Method

After scraping all the data, I needed to clean it. First, I took the two csv files I created (steaminfo.csv and steamusage.csv) and merged them by Game Profile URL using Google Fusion Tables; in the final result (SteamData.csv), every row is a data point containing game usage at a specific time, along with all of the game's attributes.

Next, I loaded the merged data into Google Refine to clean. Along with some generic cleaning (removing excess whitespace, turning the prices and current user counts into numbers, reformatting the esrb rating, merging the date and time, etc.), I made several major modifications to the data:

1. I created a new column, titled "Number of Languages," that counted the number of languages in the languages column for each data point.

2. I created a new column, titled "Single Player or Multiplayer," that checked whether Single-player, Multi-player, or both, was listed in the Game Features column for each data point.

3. I split the genre column so that each row only contained one genre using the "Split multi-valued cells" feature in Google Refine, and then

filled down the other columns, using the method described in `http://googlerefine.blogspot.com/2012/03/fill-down-right-and-secure-way.html`.

I did all of these modifications because I was interested in examining how these variables - the number of languages the game is offered in, whether the game is single player or multiplayer, and the genre of the game - influenced game usage data over the day. Genre in particular was interesting to me, so I broke it out into "wide" format, so that I could compare individual genres against each other in Tableau (shown below).

3 Related Work

To the extent that I've used what I've learned so far in CS 171, all the readings and lecture materials are related work; when justifying my visualization choices or visualization evolution, I often refer to this material. In addition, I was inspired by one source outside of class materials. While looking for interesting way to visualize hourly data on the internet, I stumbled upon the following site:

`http://evansolomon.me/notes/data-visualization-is-itself-data/`.

I really liked his method of visualizing website pageviews by hour:

and part of my visualizations are modeled on this.

4 Design Evolution

4.1 Game Attribute Impact on Usage Statistics

Before I discuss the specifics of how my visualization designs evolved, I'd like to mention one clarification that applies to all my visualizations. Whenever I break down data by hour of the day, I'm taking the average over all data points (regardless of the date) collected at that hour of the day. Since I'm primarily interested in how the number of people playing games varies depending on what time it is in the day (rather than day of the week), taking the average across days make sense. This is further justified in the Analysis section below.

I analyzed several different attributes (genre, single or multiplayer) and their impact on usage statistics, and my design process for visualizing these variables all followed the same pattern; I use the evolution of the Genre visualizations to illustrate the pattern below.

My original conception for the design of these visualizations was a stacked curve or stacked bar chart; I've included the sketch from my project proposal for reference below:

However, I tried this, along with several variations, in Tableau and felt that the visualizations were too busy/confusing.

In the stacked bar chart (leftmost image above), it was difficult to pick out genres that have different game usage trends than other genres, or total usage trends. This is likely because it is hard to see trends in the size of bars that are not all aligned with each other (aside from the bottom-most stack, all the other stacks have varying starting points as well as ending points). Using Bertin's visual variables, we can understand why that is; we are much better at judging differences in position (which is what we need to do if all the bars are aligned - we just judge which bar has the leftmost endpoint) than differences in size, especially when the objects are not adjacent. To fix this, I next tried making a table of bar charts (middle image above), with each bar chart representing one genre. While this made trends in hourly usage over time more apparent within a, it became difficult to compare genre to genre, since comparing genre's required judging the relative size of non-adjacent bars (bars in different bar graphs). To make genres more comparable with each other, I plotted the hourly trends of all the genres on the same line graph (rightmost image above). While this did make genres more comparable, it was still difficult to perform interesting analysis, because there were too many genres; the line graph encoded genre with color, but generally humans can only distinguish around 8 colors well (again due to Bertin).

After playing with various visualizations, I ultimately decided that I had to do an intermediate visualization that would help me pick and choose the most interesting genres to examine. To do this, I created the following Dashboard (called Genre Overview in the Tableau file):

This Dashboard uses a similar method of visualizing hourly data as `http://evansolomon.me/notes/data-visualization-is-itself-data/` (men-

tioned in related work), by using a highlight table, where average current user count is encoded in the lightness of the highlight. This allows the viewer to easily compare trends in daily usage by genre; as is intuitive, areas on the chart that are darker indicate heavier usage, and you can easily compare the relative darkness of cells both by column and by row. It sets the overall daily usage data for all games (Daily Usage By Day of Week) right next to game usage data broken down by genre (Daily Usage By Genre) so you can easily identify genres with different usage trends than the overall game usage trend. Note that the current user number in the Daily Usage By Genre chart are percentages of the overall average for that genre, so you can see the busiest times (in terms of current users) for each Genre, since you're normalizing so that more popular Genres aren't always darker. In addition, there is a bar graph with Genre ranked by average number of current players, so you can easily identify the Genres that are most popular, and thus are more important to consider. Finally, the three charts in this Dashboard are linked. When you hover over a genre in the Most Popular Genre chart, the daily usage by day of week chart is filtered to show only data from that genre, and the relevant column in the Daily Usage By Genre chart is highlighted; this allows viewers to identify genres that are both popular, and have interesting daily usage trends. When you hover over a day of the week in the Daily Usage By Day of Week chart, both of the other charts are also filtered to only show data from that day of the week.

After analysis of this visualization (described in the Visualizations and Analysis section below); I finally picked six genres that were most interesting: Sports, Simulation, Racing, Free to Play, Indie, and Massively Multiplayer. For my final visualization, I plotted the hourly usage trends of these six Genres on a line graph (which I had settled on earlier before deciding that there were too many genres):

As before, the x-axis encodes the average number of current users for games of the specified genre, at the specified hour, as a percentage of the average number of current users of the specified genre over all hours of the day. This allows viewers to identify genres that have the most uneven and least uneven game usage patterns over the hours day. In addition, I manually set the color encoding of the genres into shades of green and shades of red, to emphasize the difference between two classes of genres. The genres that are encoded with shades of green are associated with more "intense" gamers, while the genres that are encoded with shades of red are associated with more "casual" gamers.

4.2 Release Date vs. Singleplayer or Multiplayer

One other relationship I decided to examine was the relationship between release date and whether the game was singleplayer, multiplayer, or both. At first, I tried visualizing this relationship on a line graph, shown below:

However, all I could see from this visualization was that games that were released more recently were more likely to be included in the top 100 games by current user count, regardless of whether or not they are multiplayer or singleplayer; this is neither surprising nor interesting. Instead, I decided to visualization this data as a stacked bar chart:

The reason why I chose to do this, despite the disadvantages of stacked bar charts mentioned earlier, is that I was more interested in seeing how each individual stack within a bar related to the whole bar, rather than trends in the size of individual stacks over release date. While stacked bar charts make it more difficult to see trends within a series of data, they are good for seeing the relationship between individual stacks and the whole bar. Furthermore, since there are only three categories (singleplayer, multiplayer, or both), many of the problems associated with stacked bar charts are less significant.

5 Visualizations and Analysis

5.1 Game Usage by Day of the Week

First, I examined the impact of the day of the week on hourly game usage (as measured by total number of current players):

From this graph, it appears that the hourly game usage is fairly consistent across day of the week, at least for weekdays (unfortunately, I wasn't able to collect any weekend data). In addition, if you filter by a specific genre (which you can do on the Genre Overview dashboard), you get the same result. Because of this, I felt comfortable in ignoring day of the week altogether, and just averaging hourly data across all the dates in my data. All later charts in my visualizations that contain hourly current user numbers are averaging the current user numbers for that hour across multiple days.

One other interesting thing to note is that the data from Wednesday at

1pm EST seems to be missing; I assume my Amazon virtual server instance experienced some downtime around them.

5.2 Impact of Genre on Hourly Game Usage

As mentioned in the design evolution section above, I made two visualizations to examine the impact of the genre of a game on it's hourly current user count. First, I used a Dashboard (the tab labeled "Genre Overiew" in the Tableau file) to identify the most interesting genres:

First, note that Sports, Simulation, and Free to Play are definitely the most popular genres (as measured by average current user count) on Steam, as seen in the Most Popular Genre's table. Next, if you look at the Daily Usage by Genre chart, both Indie and Massively Multiplayer games have relatively smooth daily usage statistics as compared to other genres (hourly usage barely rises above 150% and never falls below 60% of average usage). Finally, again looking at the Daily Usage by Genre chart, Racing has an extremely uneven daily usage trend - players play racing games only between 5am and 5pm. This is somewhat exaggerated by the fact that we're only collecting usage data from the top 100 games; racing games probability still have some current players before 5am and after 5pm, but they have dropped out of the top 100 games by current player, and we thus don't collect any data on these games.

After identifying these six genres as particularly interesting, I graphed them on a line chart (the tab labeled "Genre Focus" in the Tableau file), as shown below:

I also imposed another layer of categorization in the genres based on my own knowledge of games; generally genres like Sports, Racing, and Simulation are associated with more casual, everyday gamers, while genres like Indie and Massively Multiplayer are associated with more intense, "nerder" gamers. The Free to Play genre is a mixed bag, but I categorized it in the latter group in this graph. Not surprisingly, casual genres have spikey hourly usage data (since causal players always play at more reasonable times of day, like between noon and 10pm), while intense genres have smoother hourly usage data (since intense gamers are more willing to play at odd hours of the night).

5.3 Impact of Singleplayer or Multiplayer on Hourly Game Usage

Unlike genres, which were too numerous to compare effectively, there are only three relevant categories to compare here: singleplayer, multiplayer, or both. As before, I visualized this data as a highlight table (the tab labeled "Daily Usage By Single or Multiplayer" in the Tableau file), shown below:

There are two interesting observations to make about this data. First, notice that singleplayer games have marginally smoother hourly usage data than multiplayer or both single and multiplayer games; while this is probably too small a difference to be significant, it may be explained by the fact that multiplayer games are most fun when other people are online, so there is a natural spikiness to hourly usage data for multiplayer games. Second, notice that multiplayer game usage peaks between 7am and 5pm, whereas singleplayer and both singleplayer and multiplayer game usage peaks between 10am and 10pm; on average multiplayer gamers are playing their games earlier in the day. I don't have a hypothesis as to why this is the case, but it is interesting to see this distinction; it is possible that one type of game is more popular in a certain time zone or with a certain age group (older people stay up later).

5.4 Game Release Date vs. Single or Multiplayer

I visualized the relationship between Game Release Date and Single or Multiplayer in a stacked bar chart (the tab labeled "Release Date vs. Single or Multiplayer"), shown below:

The y-axis encodes the number of game titles released, the x-axis encodes the release year, and color encodes the type of game (singleplayer, multiplayer, or both). This visualization can be read to have one of two conclusions. Either games that are both multiplayer and singleplayer have more staying power (in terms of popularity), or that publishers have been more willing to release games that are exclusively multiplayer or exclusively singleplayer in recent years. Since the data we collected consists of only the top 100 or so games (in terms of current user count), it is hard to tell which of these two conclusions to draw; in reality, the trend we see in the data (that games released more recently are more likely to be exclusively singleplayer or multiplayer) is likely caused by a combination of these two effects.

6 Conclusion

Through this process of Exploratory Data Analysis, I've identified several interesting relationships between game usage statistics and various game attributes, as well as relationships between different game attributes. That being said, if I had more time, I would be interested in further investigating this data in the following ways:

1. Examine the relationship between hourly game usage statistics and price, language count, ESRB rating, metacritic rating, etc.

2. Identify any differences between hourly game usage statistics on week and weekend days.

3. Scrape data for games other than games in the top 100 games by current user count, to get a richer picture of how various game attributes affect to each other.

Furthermore, number 3 on this list emphasizes a general weakness with the data I used: I only collected data for the top 100 games by current user count at the current date and time. This data collection method likely introduced some artifacts into the data, like the dropoff of current users playing racing games to zero at certain times of day, and thus all results presented above should be regarded with caution; further investigation is needed to confirm these trends.