A Statistical Analysis of Reddit

Jialiya Huang
Olin College of Engineering

I. Introduction

Reddit.com is a social news website with both user generated and external content [1]. This content is then curated by other users through an upvote/downvote system which, along with the time posted, determines the "hotness" ranking of a post given by the following equations.

$$\text{Score} = \text{Upvotes} - \text{Downvotes} \tag{1}$$

$$y = \begin{cases} 1 & \text{if Score} > 0 \\ 0 & \text{if Score} = 0 \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

$$z = \begin{cases} |\text{Score}| & \text{if} |\text{Score}| \geq 1 \\ 1 & \text{if} |\text{Score}| < 1 \end{cases} \tag{3}$$

$$f(t_s, y, z) = log(z)\frac{yt_s}{45000} \tag{4}$$

where t_s is the time difference in seconds between the time of posting and 7:46:43am Dec 8, 2005[3].

These rankings govern the order that posts appear on www.reddit.com, so a high ranking means more exposure to potential readers. Reddit is also divided up into smaller subreddits which focus on more specific items, such as pictures, humor, etc. These subreddits follow a similar ranking scheme except it is limited to posts within that subreddit as opposed to on the front page, which is consolidated from all the different subreddits.

Reddit is reguarly receiving more than 1 billion page views per month, and is increasingly becoming the way that people are exposed to and consume news [2]. Each page on Reddit contains 25 posts, sorted by the previously mentioned hotness rating. After 24 hours, a post is removed regardless of ranking. This paper broadly investigates the question: *What are differentiating factors between posts that reach the front page and those that do not, and is there anything a user can do to maximize a post's chances of making it to the front page?*

II. Sources

In order to collect the data necessary for this analysis, I built a Python bot to scrape Reddit.com using its JSON API. The bot collected data from the top 8 pages (200 posts) every 10 minutes. While it would have been better to collect data on posts from the time it was created until 24 hours later to when the post expires, the large amount of posts submitted to Reddit makes it unfeasible to attempt to track that much data. Instead, the top 200 posts were chosen since those posts have a high likelyhood of making it to the front page while also being far back enough that the average user does not look past the first 8 pages. I chose a time interval of 10 minutes since I wanted posts to vary in rank between scraping times. Also, it sometimes takes up to 30 seconds for one page request to complete.

I ran the bot from 9/7/2011 at 20:30 to 9/20/2011 at 09:50. Every record was given a ranking from 1-200 depending on the order the bot scraped the page, where a rank of 1 is the first post on the front page. Requests for Reddit pages timed out at several times, resuting in 361475 records collected instead of 361800. All posts which had been on the front page when the bot started/stopped running were removed from this list resulting in 339964 useable records of 5923 posts by 4798 users.

III. Posting Time

From equation 4 we can see that posting time is directly proportional to rank. Just by browsing Reddit, we can see that the turn over rate of the front page is higher during the daytime than say, 5am in the morning. This high turn over rate leads to more posts being exposed to the front page, however these times are when more users are online so more posts are made. This can be best seen through plotting the average number of unique posts that have reached the top 200 along with the average number of posts that reach the front page as shown in Fig 1.

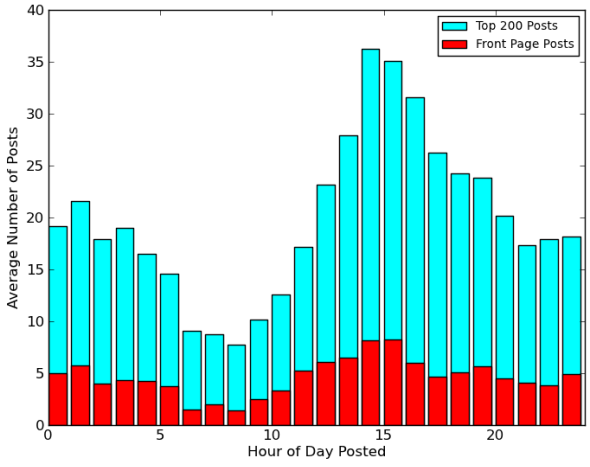

Fig. 1. Histogram of creation times (GMT)

Indeed there are more posts reaching the front page at around 15:00 GMT, which is also the period of highest average activity. However these averages could be misleading if certain days of the week had greater activity at certain times, for example higher postings during late nights on the weekends. In order to make sure this isn't the case, I plotted the total number of postings across the entire time the bot ran, as shown in figure 3. Because the sample size of posts that never make the front page (4544) is so much larger than posts that do (1379), I randomly picked 1379 data points to test out of the total sample size of 4544.

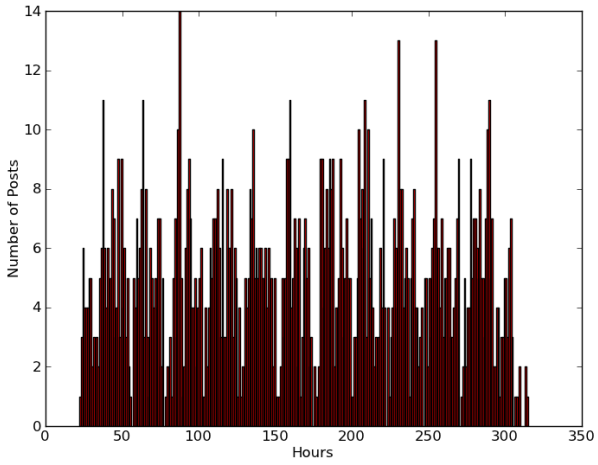

Fig. 2. Number of posts that have reached the front page by hour posting. The x axis corresponds to the number of hours since the day the bot started running.

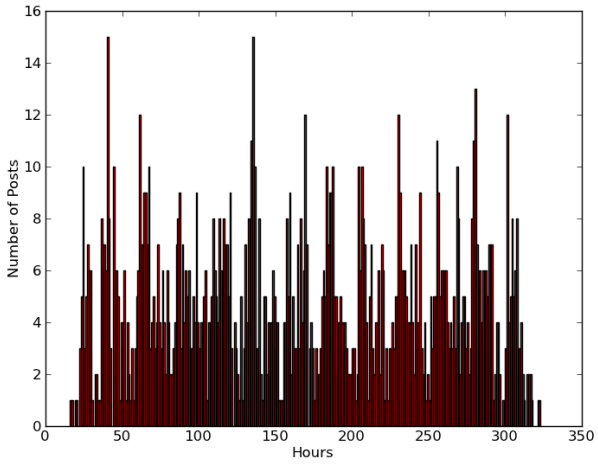

Fig. 3. Random selection of number of posts that never reached the front page by hour posting. The x axis corresponds to the number of hours since the day the bot started running.

We can see that both graphs look periodic, repeating rougly once every 24 hours. However, the number of posts that reach the front page by the hour seem to vary from day to day while posts that never reach the front page is more stable across days. I computed the autocorrelation of both posts reaching the front page and posts that did not and found an autocorrelation value of 0.736 for posts that never reach the front page and a value of 0.427 for posts that do. This indicates that while the posting times to Reddit is relatively stable across the week, the posting times for front page posts varies more.

Posts tend to gain upvotes throughout their lifetime, however at the same time newer posts are created with a higher initial score. In order to minimize this effect, a post should try to reach a high rank as soon as possible so that it is exposed to more viewers and thus has a greater chance of accumulating upvotes. We can see this effect by plotting the cumulative distribution function (CDF) of the time a post reaches max rank.

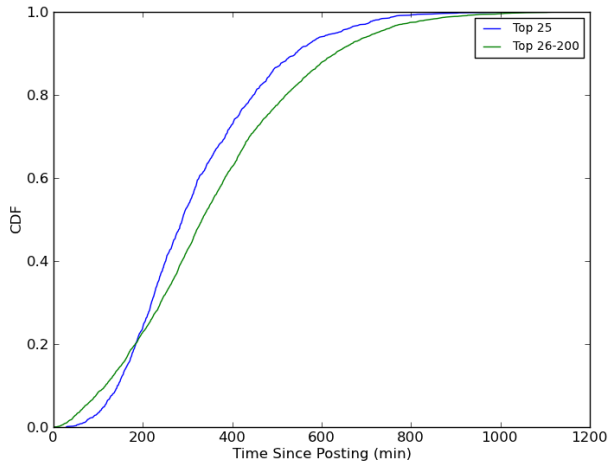

Fig. 4. CDF of posts reaching peak rank. There were a total of n=1379 posts on the front page and m=4544 other posts.

Based off of figure 4, the mean time it takes for a post to reach the highest rank if it breaks the top 25 posts is at 320 minutes while other posts lag behind at a mean of 358 minutes. The difference is similar in the median time it takes to reach the highest rank, at 289 minutes since posting for the top 25 versus 333 minutes. The difference in means could be statistically significant or the result of sampling bias. I can calculate the p-value of this difference by pooling both data sets together and randomly selecting two samples of the same size, in this case 1379 and 4544 samples, and computing the difference in means of these samples. After doing this 1000 times, I obtained a p-value of 47% which means that the difference will be at or greater than 38 minutes 47% of the time.

IV. Upvotes/Downvotes

We can also tell that newer posts will score higher than older posts given the same score from equation 4. A new post which gets upvoted immediately will rise higher than an older post with the same amount of upvotes. The higher ranking exposes the new post to more people, giving it a chance to accumulate a higher score. The voting system is logrithmically biased towards first votes, so the first 10 votes has the same weight as the next 100 votes. In order to

achieve a higher ranking, a post should accumulate upvotes as fast as possible while minimizing downvotes.

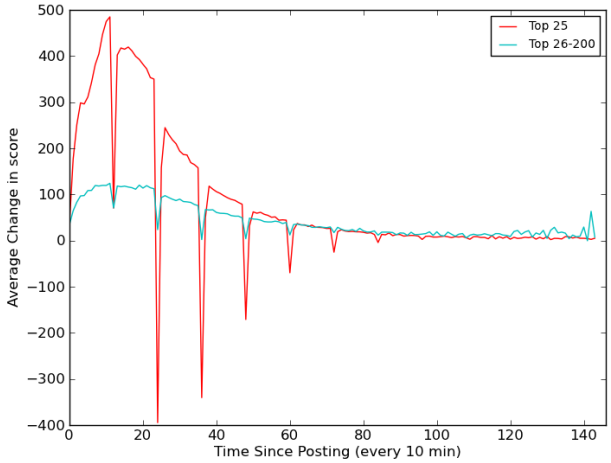

Fig. 5. Average change in score of posts. One tick on the x axis is equivalent to 10 minutes. The downward spikes in score occur when a large amount of users downvote a post over a period of time. This usually happens when a post has been too highly ranked for too long. Posts can recieve a -500 change in score over the span of 10 minutes, so these are the results of one post skewing the average.

From figure 5 we can see that while posts in both categories peak in score changes at around 100 minutes since posting time, posts reaching the front page experience, on average, a 4x greater score change than posts that never make the front page. The half life of posts that make it to the front page is at 150 minutes while the half life of posts that do not is at 250 minutes.

As posts become older, they tend to move up in score, which bumps them to a higher rank. At a higher rank these posts achieve more visibility, and have a greater chance of accumulating even more upvotes. However, as the post gets older, its score becomes smaller relative to newer posts due to equation 4. So it is in a post's best interest to obtain the highest score as soon as possible.

We can see from figure 6 that there are two peaks for posts that reach the front page, one at 238 minutes (4 hours) and another at 357 minutes (6 hours). These are likely posts which have made it to the front page very quickly after creation and get large rank bonuses due to their relative new-ness. However, once reaching the front page, they can no longer ascend in rank and max out prematurely. We can also see that posts that make it to the front page max out in score at longer times than posts that do not. Those are likely the posts which are still on the front page when they are removed after 24 hours from posting. Since they are still on the front page, these posts accumulate a higher score towards the end of their lifetime.

We can also look at the time that posts reach their minimum score since reaching the top 200 ranks to take downvoting into account. Posts that reach the front page have the chance to accumulate both upvotes and downvotes, as seen from figure 5, so these posts may reach their minimum

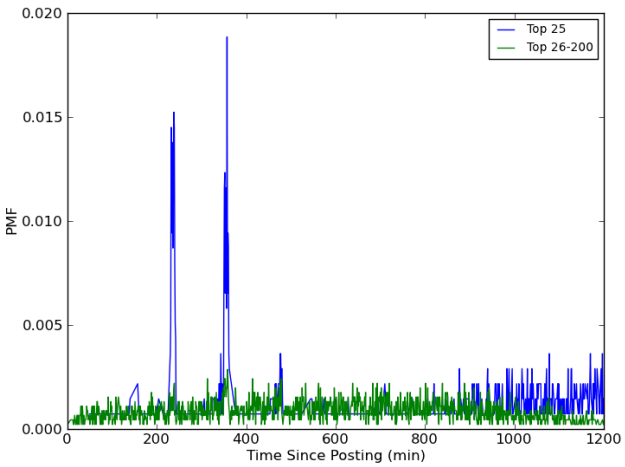

Fig. 6. PMF of posts reaching their maximum score

score later on in their lifetime. To explore this, I plotted the CDF of minimum score over time in figure 7.

We can see from figure 7 that front page posts reach their minimum score quicker with a mean of 76 minutes as opposed to other posts at a mean of 184 minutes. The median time it takes for the top 25 to reach minumum score is at 64 minutes compared to 150 minutes for other posts. The changes in mean and median time is a result of a few outliers taking a much longer time. This shows that although high ranking posts have the chance to accumulate a lot of downvotes, by the time they are on the front page the content is interesting to a majority of users. On the other hand, lower ranking posts reach a minimum score later in their lifetimes. Since I only collect data from the top 200 ranks, this could be due to them entering the top 200 ranks at a later time. For example, a post could enter the top 200 ranks a 5 hours after its creation with a low score however the bot didn't pick up the post until then and thus assigned that post a minimum score at 5 hours.

From figure 8 we can see that both sets of data are relatively linear, with the top 25 following a more exponential trend than the rest. The linear regression gives us values of -0.0086 and -0.0031 for the slopes, respectively. We can use this to model the two CDFs with the following equations:

$$\mathrm{CDF25}(x) = 1 - e^{-0.0086x} \tag{5}$$

$$\mathrm{CDF200}(x) = 1 - e^{-0.0031x} \tag{6}$$

V. Correlations

We previously looked at relationships with score and rank related to time of posting, but we can also check the correlation between scores and rank. I scanned through all the posts and plotted the score versus the rank of a post. We expect that higher ranks should have higher scores, perhaps with a linear distribution or maybe even exponential but we can see from figure 9 that there is a lot of

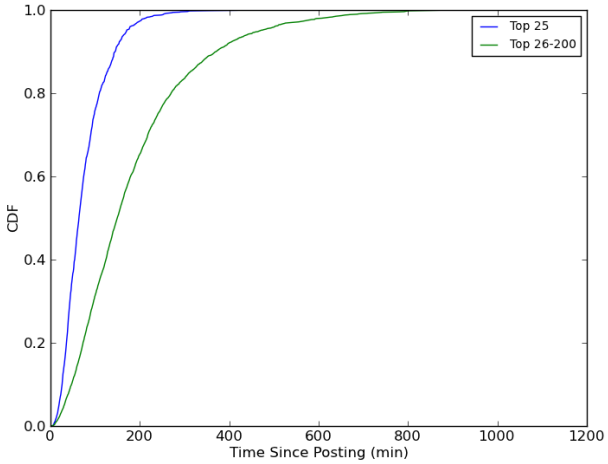

Fig. 7. CDF of time it takes to reach the minimum score since reaching the top 200 ranks.

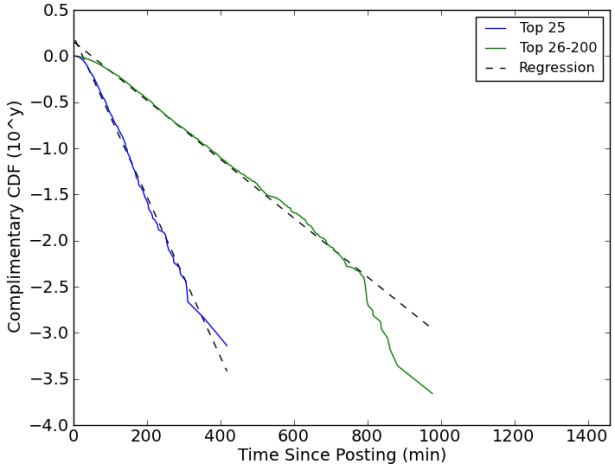

Fig. 8. CCDF of minimum score on a log-y scale, showing an exponential relationship.

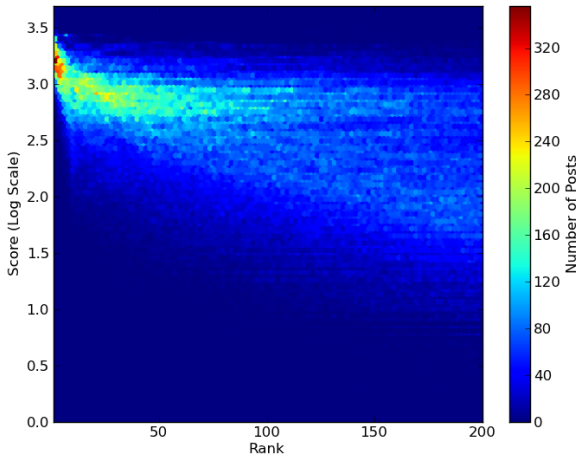

Fig. 9. Hex plot on a log scale of ranks to scores.

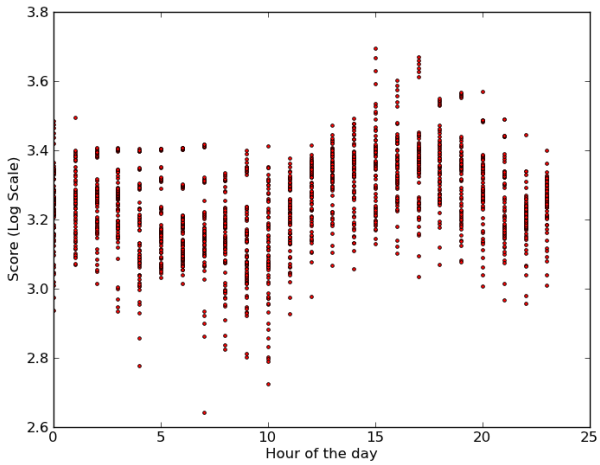

Fig. 10. Scatter plot of time versus score for posts of rank 1

variance in low scoring posts, with some of the top scoring posts in the bottom ranks having an equal score with posts in the top ranks. I computed a Pearson's coefficient of this dataset to get a value of -0.417, however since the top scores are outliers, I also computed a Spearman's coefficient and found a value of -0.451. This suggests that there is a slight correlation of ranks to score.

Another posibility for the wide distribution of scores per rank is varience within certain times of the day. For example, a post on the front page at noon, when Reddit recieves a large amount of traffic, is more likely to have a higher score than a post with the same rank at 4am. In order to check this I plotted the score versus the time of day as shown in figure 10.

Indeed, just as we expected, we can see a bump in score from figure 10 which starts rising at around noon and peaks at around 15:00. However, with the exception of the few high scoring posts at this time, the majority of posts seem to have a similar score at around 1960. I calculated a Pearson's coefficient of 0.254, which tells us that rank correlates almost twice as much with score versus time, for posts of rank 1. The bump in score is apparent around noon, however figure 10 is only for posts of rank 1 which has the most viewership of any post on Reddit. I computed the same plot for the top post of the next page, rank 26, the results of which are shown in figure 11.

There is no visible bump in score around noon 11 when we look at posts of rank 26. This is only one page behind the top ranking post, however the distribution of scores is almost twice as low with an average of 780. Again, I computed a Pearson's coefficient of 0.024, which shows that there is almost no correlation between time and score. If we refer back to equation 4 we can see that time of posting has a much greater effect on rank compared to score, so it is unsurprising that there is such a high varience between scores of a certain rank.

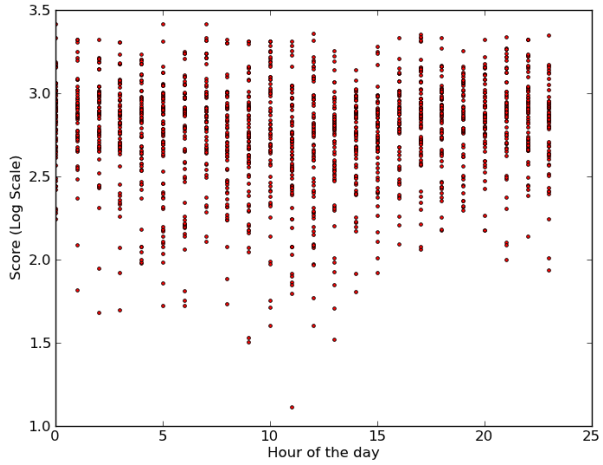

Fig. 11. Scatter plot of time versus score for posts of rank 26

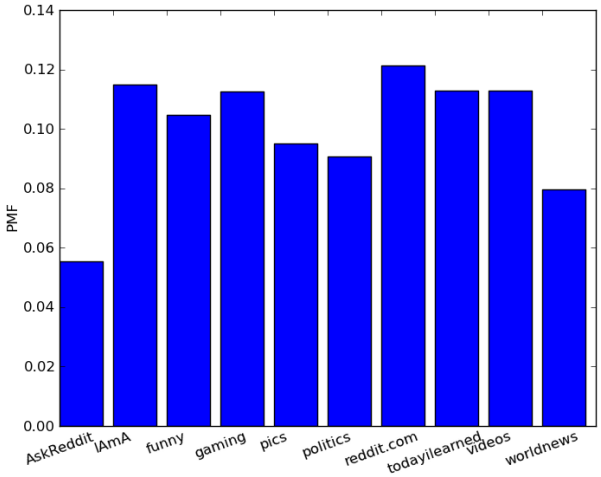

Fig. 13. PMF of reaching the front page by subreddit

VI. Subreddits

Users can submit to the main Reddit page or to one of the community supported subreddits. Subreddits vary in membership and postings, so a newer post might achieve more visibility on one subreddit versus another. Posts that made it to the top 200 were in one of 12 subreddits, including "announcements" and "blog", which are posts made from Reddit administrators about changes made to Reddit, etc. 4 posts fell into this category and were removed from the dataset since regular users cannot post to them.

Most posts were submitted to /r/pics and /r/funny, with the fewest from /r/IAmA. The chances of reaching the front page through these subreddits can be better seen in figure 13, with posts being most likely to reach the front page submitted from /r/reddit and least likely to come from /r/AskReddit. In fact, the difference is almost twice as much with a 5.67% chance from /r/AskReddit and a 11.97% chance from /r/reddit.

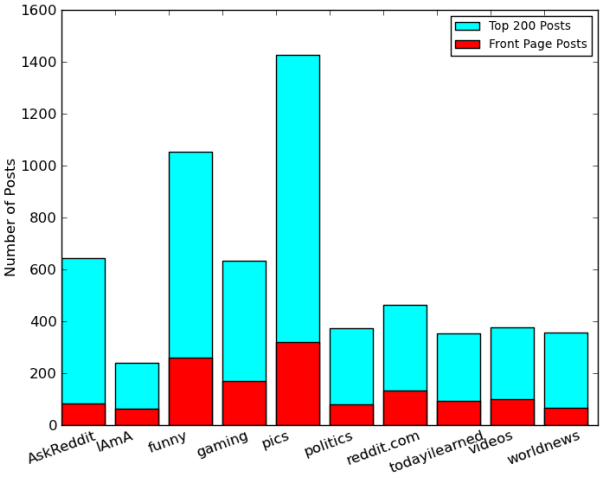

Fig. 12. Histogram of posts by subreddit

However, since subreddits have specific functions specified by "reddiquette", the code of conduct that is enforced by Reddit users, it is not likely that a post would fit well in both /r/reddit and /r/AskReddit, since the latter is specifically reserved for questions. A better comparison would be between /r/funny, /r/pics, and /r/reddit, which can overlap from, for example, a post of a funny picture. We can calculate the probabilities using Baye's theorem using equation 7.

$$P(H|E) = \frac{P(H)P(E|H)}{P(E)} \tag{7}$$

In equation 7 H is our hypothesis (that a post will reach the front page) and E is the evidence (that a post has been submitted to X subreddit at Y time). Since I only collected data from the top 200 posts, we must add the additional condition that the post is already in the top 200 ranks for both the hypothesis and the evidence. Plotting these probabilities for the 3 subreddits gives us figure 14.

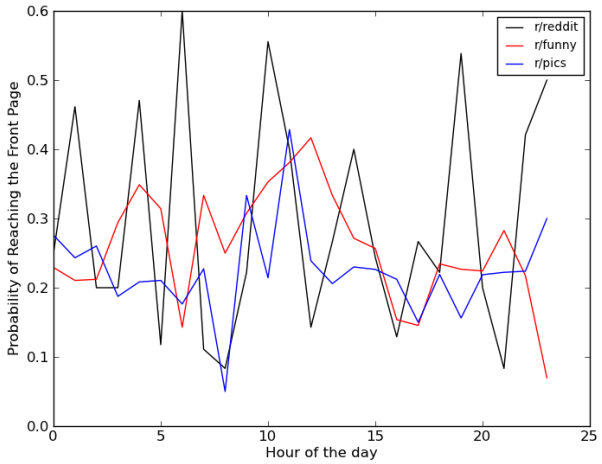

Fig. 14. Probability of reaching the front page by subreddit

We can see that /r/reddit.com displays a higher variance than either /r/funny or /r/pics; at times it is much more likely to get on the front page compared to the other subreddits. On the other hand, the probability of getting to the front page from /r/pics is almost always smaller than /r/funny.

VII. Comments

A user who has commented on a post is more likely to vote on it, so the number of comments may be an early predictor of whether or not a post will reach the front page. In order to test this theory, I made a plot of the number of comments on a post when it first appeared at the lowest rank of the top 200. I cut off posts which first appeared at ranks higher than 175 for the data used in this analysis.

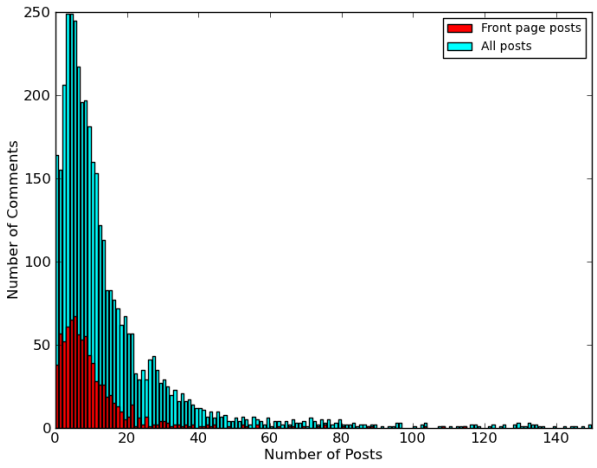

Fig. 15. Number of posts with a given number of comments the first time it reached ranks 175-200. Posts which first appeared higher thank rank 175 were ignored, giving a sample size of 840 for front page posts and 3924 for other posts.

From figure 15 I calculated the mean and variance of front page posts versus other posts and obtained 10.3, 189.3, 16.0, and 613.5, respectively. From this, it seems that a lower number of comments could be a predictor for a higher rank instead of what we initially thought. We can explore this further by running a Chi-squared test with bins for a low number of comments (between 0 and 5), a normal number (6-20), and a high number (21 and above). The Chi-squared statistic generated from this definition is 848. If we run multiple simulations with randomly chosen data, we obtain a p value of 21%, which means this effect occurs randomly 21% of the time. A value this high most likely means that the effect we are seeing could be from random chance.

VIII. Reposts

A repost is a post in which a user submits the same content as another user, usually within a short span of time. Users receive "karma" based off on the up/downvotes of their submissions, so seeing a repost which gets a higher ranking than an original post is frustrating. Of the 5923 posts analyzed, only 15 were reposts with the same external url, with 14 posts having 1 repost and 1 post having 2 reposts. Self posts (posts made on reddit which do not link to external content) were not counted. This study only analyzed posts within the top 200 ranks; many reposts were missed since the obvious reposts made within a few minutes of each other would have gotten filtered out.

Orig Rank	Repost Rank	Δ Time (min)
1	140	74
2	115	5
6	177	457
7	51	48
13	102	1694
22	166	986
25	41, 171	108, 75
28	114	88
43	164	6
99	1	.5
110	47	161
111	45	118
135	121	466
194	190	22

Fig. 16. Table showing peak rank reached by original posts versus reposts along with the difference in creation time of the posts.

Once a post has reached the top 200, original postings tend to do better than reposts with the exception of one repost making it to rank 1. This could be due to it being posted only half a minute behind the original posting. After a post has reached the front page, subsequent reposts do not perform as well, however the opposite is true of lower ranking posts, where reposts seem to rank higher than the original.

IX. Conclusion

Based off of the observations and results in this paper, I suggest the following scheme to maximize a post's rank:

1. Time of posting doesn't matter much, however the upvotes right after creation does, so if there is region centric information in the post, post at a time where those readers will be actively browsing Reddit. This seems to be around noon in US EST.
2. Post to /r/reddit.com or a subreddit which has a lot of subscribers but doesn't get completely flooded with new posts like /r/pics.
3. If the intial post doesn't do well, repost after a few hours.

Overall, the differences that a user can do outside of improving the quality of a post is minimal since the probability of reaching the front page is so low even among posts that reach the top 200. The easiest way to bump a post up in ranking would be to create multiple fake accounts and have these accounts upvote the post as soon as it is made. Since the weight in score is logrithmic, these first votes will have a much larger impact on ranking than anything else a user can do.

References

[1] Reddit faq. http://www.reddit.com/help/faq.
[2] Ben Parr. Reddit surpasses 1 billion monthly pageviews. http://mashable.com/2011/02/02/reddit-surpasses-1-billion-monthly-pageviews/, Feb 2011.
[3] Amir Salihefendic. How reddit ranking algorithms work. http://amix.dk/blog/post/19588, Nov 2010.