

# Popularity Dynamics and Intrinsic Quality in Reddit and Hacker News

Greg Stoddard  
Northwestern University

## Abstract

In this paper we seek to understand the relationship between the online popularity of an article and its intrinsic quality. Prior experimental work suggests that the relationship between quality and popularity can be very distorted due to factors like social influence bias and inequality in visibility. We conduct a study of popularity on two different social news aggregators, Reddit and Hacker News. We define quality as the number of votes an article would have received if each article was shown, in a bias-free way, to an equal number of users. We propose a simple Poisson regression method to estimate this quality metric from time-series voting data. We validate our methods on data from Reddit and Hacker News, as well the experimental data from prior work. Using these estimates, we find that popularity on Reddit and Hacker News is a relatively strong reflection of intrinsic quality.

## 1 Introduction

One of the many narratives surrounding the growth of social media is that our systems for liking, retweeting, voting, and sharing are giving rise to a digital democracy of content. As the narrative goes, virality enabled “Gangnam Style” to dominate international audiences, helped the Ice Bucket challenge raise millions of dollars for ALS research, and we now interpret trending topics on Twitter as a signal of societal importance (Gillespie 2011). There’s a considerable amount of academic work that interrogates this narrative by delving deeply into understanding the properties of virality. For example, scholars have studied the propagation and correction of rumors (Friggeri et al. 2014), the role of influential users in spreading information (Bakshy et al. 2012), or whether information actually diffuses in a viral way at all (Goel, Watts, and Goldstein 2012). Although many papers hint at it, few papers directly address a basic question: do these systems promote the best content?

As a thought experiment, imagine polling a large population of people and asking them to rate every music video uploaded to Youtube in 2012. Would “Gangnam Style”, the most watched video on Youtube, still come out on top? Evidence from the MusicLab experiment of Salganik, Dodds, and Watts (2006; 2008) suggests that it might not. In this experiment, the authors set up a website where users could listen to and download songs from unknown artists. When visiting the site, participants were randomly assigned into

1 of 8 different worlds, and were presented a list of songs ordered by the number of downloads each song had in that world. This design let the authors observe the parallel evolution of popularity of the same set of songs across different worlds. They found that the popularity of a song could vary widely between worlds; songs with the largest share of downloads in one world went relatively ignored in another one. This variance was caused by a strong rich-get-richer effect; songs with more downloads were ranked higher in the list and were more likely to be sampled by future listeners. In the presence of such effects, the authors conclude, popularity is a noisy and distorted measure of quality.

What do these results imply about the relationship between quality and popularity in today’s socio-technical systems? Facebook and Twitter have a rich-get-richer element in their designs because posts with more likes and retweets are more visible, on average, than their less popular counterparts. Does this imply that there’s a distorted relationship between quality and popularity on these platforms? In the absence of running experiments, this question seems difficult to answer because we need to somehow estimate how popular an article could have been but only using observed popularity data.

**Present Work** In this paper we show that social news aggregators are a good setting to study the quality-popularity relationship. We conduct our study on two aggregators, Reddit and Hacker News. Reddit is a popular site where users submit links to content from around the web, and other users vote and comment on those links. Hacker News is an aggregator dedicated to programming and technology-related issues but is otherwise similar in structure. Reddit received approximately 450 million page views in December 2014, while Hacker News received approximately 3.25 million.

These aggregators have several properties that facilitate disentangling observed popularity from intrinsic article quality. The first property is that content visibility is easier to measure on Reddit and Hacker News. The interface of each site is a simple non-personalized list of links<sup>1</sup>, so the observed article ranking is (approximately) the same for all users. Due to the similarities in UI, estimating visibility on Reddit or Hacker News is very similar to estimating position

<sup>1</sup>Reddit is lightly personalized; we discuss this later in the paper.

bias in search results and search ad rankings. We exploit this similarity in our techniques. The second property is that both sites only use votes to rank articles, rather than more complex measures like impressions or social-tie strength, and these votes are publicly observable. Furthermore, each site publishes their algorithm for converting votes into a ranking.

Finally, recent empirical work shows that popularity on Reddit exhibits signs of a distorted relationship between quality and popularity (Gilbert 2013). Gilbert finds that over half of popular image submissions on Reddit are actually reposts of previous submissions. The same picture may receive no upvotes on its first submission but its second or third submission may gain thousand of upvotes.

## 1.1 Our Contributions

The main contribution of this paper is formalizing a metric for article quality and developing a method to estimate it from observed voting data. We define quality as the number of upvotes an article would have received if articles were displayed in a random order with no social signals (such as current score). This is only a hypothetical process but we show that we can estimate this counter-factual score from observed popularity data.

The key to our analysis is the use of time-series observations of voting behavior for each article. Observing the same article at different points in its life allows us to disentangle the influence of different factors on voting. We develop a simple Poisson regression model for learning parameters from observed data that factors out article qualities from biases such as position effects, time decay, and social influence. Since we lack the ability to evaluate against ground truth data from Reddit or Hacker News, we evaluate this model on data from the MusicLab experiment. We find this method is effective at recovering ground truth quality parameters, and further show that it provides a good fit for Reddit and Hacker News data.

We then examine the relationship between observed popularity and quality estimates. We find a surprisingly strong relationship between popularity and quality but with an important caveat. Many articles submitted to Reddit and Hacker News only receive a very small amount of attention and did not generate enough observations to be included in our study. Its likely that there are many high quality articles included within this ignored set of articles that our method cannot account for. However among the set of articles with a reasonable amount of attention, we conclude that popularity is a good indication of relative quality.

Finally we expand upon the study of reposting behavior on Reddit (Gilbert 2013) and show that reposters actually helps Reddit aggregate content that is popular on the rest of the web. Specifically, we show that the number of times an article is submitted to Reddit is positively correlated with its external popularity, and these reposts raise the probability that at least one becomes popular.

## 2 Related Work

This work is related to the large literature on popularity prediction. One implication of the MusicLab experiment is that

popularity is inherently difficult to predict at cold start (Salganik, Dodds, and Watts 2006) but the this literature generally shows that popularity can be predicted with by using early popularity as a signal. For example, the number of views that a Youtube video receives after its first month can be predicted by the pattern of views over its first week (Szabo and Huberman 2010; Pinto, Almeida, and Gonçalves 2013). Similarly a large-scale study of photo-sharing cascades on Facebook shows that temporal features related to the initial shares of a photo are effective at predicting eventual popularity (Cheng et al. 2014). On the other hand, some work shows that content features are not effective for predicting popularity. The aforementioned Facebook study and a study of Twitter show that content features add no prediction accuracy over temporal or structural features (Bakshy et al. 2011). Some scholars have proposed and tested prediction methods that only use content features (Bandari, Asur, and Huberman 2012), but a recent replication study challenges the efficacy of these “cold-start” methods.

The goal of this work has a subtle difference from the prediction literature. Our goal is estimate the popularity or rating of an article in a hypothetical unbiased world by teasing out an article’s true “quality” from biased voting data. A recent experiment shows that social influence bias can cause large distortions in comment ratings on a news site (Muchnik, Aral, and Taylor 2013), and thus demonstrates the need for better understanding social influence and developing methods to de-bias these ratings. Krumme et al use the MusicLab data to show that social influence affects a user’s decisions of which songs to sample but not to which songs to download (Krumme et al. 2012). A news aggregator experiment on Mechanical Turk shows that the effect of social influence is not as strong as the effect of a bias in attention due to positional effects (Hogg and Lerman 2014). One significant challenge in this area is to use purely observational data in studying the effects of various biases. Wang et al develop a statistical model to remove social influence bias and recover “true” product ratings from observed Amazon ratings (Wang, Wang, and Wang 2014). Other scholars have used similar statistical models to demonstrate that the helpfulness rating of Amazon reviews is affected by the rating of other reviews for the same product (Sipos, Ghosh, and Joachims 2014).

The academic study of Reddit is fairly nascent but older social news aggregators have receive a reasonable amount of attention. One line of work studies the explicit and implicit mechanisms that Slashdot’s community<sup>2</sup> uses to moderate comments, filter content, and teach new users about community standards (Lampe and Resnick 2004; Lampe and Johnston 2005). Hogg and Lerman studied the popularity dynamics on Digg<sup>3</sup> and demonstrated popularity can be forecast accurately by tailoring a statistical model to reflect the algorithm and interface that Digg used (Hogg and Lerman 2009). Finally there’s a small literature that exam-

<sup>2</sup>A technology-focused news aggregator slashdot.org.

<sup>3</sup>A general interest news aggregator digg.com. The design of the site was significantly different when these studies were conducted.

ines popularity and community behavior on Reddit. Leavitt and Clark used a mixed-methods approach to study the evolution of standards and content popularity in a community dedicated to the 2012 Hurricane Sandy event (Leavitt and Clark 2014). Lakkaraju et al study the effects of title and language on the popularity of reposts of the same image (Lakkaraju, McAuley, and Leskovec 2013). Das and Lavoie use user behavior on Reddit to train a reinforcement-learning model for how users react to community feedback (Das and Lavoie 2014). In particular, they examine how feedback influences a user’s choice of which sub-communities to join.

### 3 Data

The design of Reddit and Hacker News are quite similar. The interface of each site is an ordered list of articles, with 25 or 30 articles appearing on each page. Logged-in users of each site can upvote or downvote each article, and these votes are used to rank articles.

**Reddit** Reddit is composed of many different sub-communities called “subreddits”. For example `r/news`<sup>4</sup> is the subreddit for discussing news and current events. Within a subreddit, articles are ranked in decreasing order of their “hot score”, which is defined by<sup>5</sup>:

$$\log(u_i - d_i) - \frac{1}{750} \text{age}_i$$

Where  $u_i, d_i$  is the number of upvotes and downvotes received by article  $i$  and  $\text{age}_i$  is the number of minutes between the current time and the time the article was submitted<sup>6</sup>.

**Hacker News** Hacker News differs structurally in two ways. First, users can upvote stories but cannot downvote them. Second, there are only two different article rankings: the “new” ranking which is a chronological list of articles, and the “top ranking”. In the “top ranking”, articles are ranked by<sup>7</sup>:

$$\frac{(u_i - 1)^{.8}}{(\text{age}_i + 2)^{1.8}}$$

**Data Collection** We collected data at 10 minute intervals over a one week period from 5/26/14 to 6/1/14 from each site. For each site, we record the number of votes (upvotes and downvotes) and position of each article. We can compute the number of votes an article received in the 10 minutes between scrapes using this data. For our purposes, each observation is a tuple  $(t, i, j, v_i^t)$ , meaning that article  $i$  at time  $t$  was observed in position  $j$ , and received  $v_i^t$  upvotes in the time period  $t$  to  $t + 1$ . For Reddit, each observation is a tuple  $(t, i, j, v_i^t, s_i^t, u_i^t, d_i^t)$  where  $u_i^t$  and  $d_i^t$  are the number of upvotes and downvotes,  $v_i^t = u_i^t + d_i^t$  is the total number of votes and  $s_i^t = u_i^t - d_i^t$  is the change in score. We collect all articles that appear in the top ranking of Hacker News (which is at most 90), and the top 500 ranked articles

<sup>4</sup>by convention, “r” is prefixed to the name of a subreddit

<sup>5</sup>[github.com/reddit/reddit](https://github.com/reddit/reddit)

<sup>6</sup>There’s additional logic to handle the case where  $d_i \geq u_i$  but most of our observations have  $u_i > d_i$

<sup>7</sup>[news.ycombinator.com/item?id=1781013](https://news.ycombinator.com/item?id=1781013)

Dataset	Observations	Articles	Score
Hacker News	29K	750	66 (39)
r/todayilearned	40K	1187	125 (16)
r/videos	45K	2249	42 (2)
r/worldnews	40K	1417	39 (6)
r/news	33K	1132	38 (6)
r/pics	57K	1883	53 (5)

Table 1: Summary statistics for the data used. The last column shows the mean (and median) score for articles in the dataset.

in five different subreddits. We then filter and clean the data in a number of ways as detailed in the appendix. Summary statistics for the filtered datasets are shown in table 1.

**Terminology:** In this work, we’ll refer to *score* as the number of upvotes in the case of Hacker News, or the difference of upvotes and downvotes in the case of Reddit. We’ll also use that term to refer to an article’s score at a specific point in its life, i.e. score at time  $t$ . We will also use term *score* interchangeably with popularity.

### 4 Model

In this section we formalize our definition of article quality and present our method for estimating it. We define the *quality* of an article as the score an article would have received if all articles were displayed in a random order, absent of any social signals, to a large and equal number of users from the population. When computing quality quantitatively, we will scale by a constant such that the maximum quality article in a dataset is equal to 1. Intuitively, this measures the relative popularity an article would have in a hypothetical world where articles receive equal attention and user opinions are not influenced by any external factors.

The term “quality” as we use it may conflict with some natural interpretations of quality. Although some may think of a high quality article as an article on an interesting topic with good grammar and style, our use of the term “quality” is a purely democratic one. If a community wants to upvote trivial stories with terrible grammar, then we will label those stories as high quality articles. Furthermore the quality of an article is a function both the article and the community evaluating it. A well researched piece of investigative journalism may be a high quality article for `r/news` but would be a low quality article if submitted to `r/aww`, a community dedicated to pictures of cute animals.

This definition is not appropriate for all types of articles because we are removing the social aspect of article quality. Many submissions to Reddit and Hacker News are greatly enhanced by the comments, especially for discussion threads such as “What’s the happiest fact you know?” We purposely avoid these posts by excluding discussion dedicated subreddits and any post that does not redirect to an external article. With these notes in mind, we feel that this definition of quality is a reasonable one. Lastly we emphasize that this definition of quality and very similar ones have been used, implicitly and explicitly, in a number of previous works (Salganik and Watts 2008; Wang, Wang, and Wang 2014).

## 4.1 Parameter Estimation

We now describe our method for estimating article qualities from time-series observations of voting behavior. The time-series data allows us to observe the same articles in different conditions throughout its life. We use a model that separates observed voting data into confounding factors, such as position and social influence bias, and article-specific factors. After fitting this model, we use the parameters associated with each article to estimate its quality.

The largest issue is that we do not observe the number of users who may have viewed an article but decided not to vote on it. The observed Reddit data allows us to directly estimate the probability that an article will receive an upvote conditioned on it receiving a vote by taking the ratio of upvotes to total votes. However we cannot directly estimate the probability of receiving a vote versus not receiving a vote, for both Reddit and Hacker News. This problem is exacerbated by the presence of a position bias, i.e. that users are more likely to look at highly ranked articles than articles that are buried down in lower pages. This is a common problem encountered in estimating the click-through-rates of search results and ads, so we can use techniques developed in that literature (Dupret and Piwowarski 2008; Chen and Yan 2012). One model used is the *examination hypothesis* (Richardson, Dominowska, and Ragno 2007), which models the probability of a user clicking on article  $i$  in slot  $j$  as a two-step process. With probability  $p_j$  a user examines the article in position  $j$ , independent of the article in position  $j$ . If the user examines position  $j$ , they click on that article with probability  $q_i$ . The  $p$  and  $q$  parameters can then be estimated from observed clicking behavior in search logs, typically via maximum likelihood estimation.

The analogy from estimating the probability of an article receiving a click to an article receiving a vote is straightforward, but direct application of this model isn't possible because the granularity of our data is votes cast over a 10 minute interval rather than individual voting data. We must instead estimate the rate that an article receives votes. A natural model for modeling rates is a Poisson process, and recent work (Chen and Yan 2012) shows that the examination hypothesis can effectively be estimated with the following Poisson model:

$$v_i^t \sim \text{Pois}(\exp\{p_i^t + q_i\})$$

Where  $v_i^t$  is the votes received by article  $i$  at time  $t$ ,  $q_i$  is a variable representing article  $i$ , and  $p_i^t$  is the position it appeared in at time  $t$ . In words, this models the number of votes that article  $i$  receives when shown in slot  $j$  as being drawn from a Poisson distribution with mean equal to  $e^{p_j} \cdot e^{q_i}$ . This model learns a parameter  $q_i$  for each article, and a position parameter  $p_j$  for each position. The fitted  $q_i$  parameters can be used to estimate the quality of each article (described in later in this section). We emphasize that the position variables are treated as categorical variables, meaning that a position bias is estimated for each position  $j$  and there's no assumed relationship between  $p_j$  and  $p_{j'}$  for all  $j, j'$ . We expect that the position bias should be decreasing as you move towards lower positions and pages but we do not encode those constraints.

The above model accounts for position bias but there are other factors that affect voting. We first add an age factor to allow for activity on an article to decay over time. Many users may revisit the site multiple times per day and hence may see the same article many times but can only vote on it once. Next we add a factor to account for a potential social influence bias. Both Reddit and Hacker News display the current score of articles, and thus provide a signal about how other users evaluated these articles. Prior work shows that displaying such signals can cause a significant social influence on user behavior (Hogg and Lerman 2014; Muchnik, Aral, and Taylor 2013; Krumme et al. 2012; Salganik, Dodds, and Watts 2006). We add a term for score effects but first apply a log transformation to account for the large disparities in scores on Reddit and Hacker News. Our full model is:

$$v_i^t \sim \text{Pois}(\exp\{p_i^t + q_i + \beta_{age} \cdot age_i^t + \beta_{score} \cdot \log(S_i^t)\}) \quad (1)$$

In summary, the full model estimates an article quality effect  $q_i$  for each article, a position bias effect  $p_j$  for each position, a time decay effect  $\beta_{age}$ , and a score effect  $\beta_{score}$ . We fit parameters via maximum likelihood estimation, that is we find the value of parameters that maximize the probability of the observed data in the Poisson model. This is exactly equivalent to a standard Poisson regression. We use the StatsModels python module<sup>8</sup> to implement the Poisson regression, with the L-BFGS method to optimize the likelihood function (Nocedal 1980).

## 4.2 Quality Estimation

We can estimate article qualities using the fitted parameters from the above model. Recall that quality is the expected score of an article if all articles were shown to the same number of users in a random order without displaying any social signals. If we display each article in exactly  $T$  time steps, the expected number of votes received by article  $i$  is:

$$\sum_{t=0}^T \exp\{q_i + p^t + \beta_{age} \cdot age^t\} = e^{q_i} \cdot \sum_{t=0}^T e^{p^t + \beta_{age} \cdot age^t}$$

We abuse notation slightly by letting  $p^t$  be the random variable for the position of article  $i$  in the random display order and its associated position bias. The expected value of the summation term is the same for all articles because it doesn't depend on  $i$ , so we can treat this term as a constant. Finally we scale all qualities by some constant  $\lambda$  such that the maximum quality in a dataset is equal to 1. For Hacker News, score is exactly equal to the number of votes an article receives, so we can express the quality of an article as:

$$Q_i = \lambda \cdot e^{q_i} \quad (2)$$

Reddit is slightly more complex because score is the difference between upvotes and downvotes. Recall that by observing the total upvotes and downvotes received by an article, we could estimate the probability of receiving an upvote conditional on receiving a vote but not the unconditional

<sup>8</sup><http://statsmodels.sourceforge.net>

probability. The unconditional rate of upvoting is the rate of voting times the conditional upvote probability, and the predicted growth in score is just the upvote rate minus the downvote rate. Let  $r_i^{up}$  be the observed ratio of upvotes to total votes for article  $i$  and  $r_i^{down}$  be the ratio of downvotes. The quality of Reddit article is estimated as:

$$Q_i = \lambda_{sub} \cdot e^{q_i} \cdot (r_i^{up} - r_i^{down}) \quad (3)$$

We include the subscript in the  $\lambda_{sub}$  term to emphasize that this constant is different across subreddits.

## 5 Evaluation

Ideally we would like to evaluate our quality estimates against some ground truth data from Reddit or Hacker News. Unfortunately such ground truth quality data fundamentally does not exist unless one of these aggregators runs an active experiment to randomize display order and remove social signals. Another approach is to run a controlled experiment that mimics a news aggregator, as done in (Lerman and Hogg 2014; Hogg and Lerman 2014). While this method has some advantages, it still doesn't yield ground truth quality data for Reddit or Hacker News because the recruited population is unlikely to match the relevant population of users on Hacker News or Reddit.

We instead validate the model in two ways. First we apply this model to data from the MusicLab experiment (Salganik, Dodds, and Watts 2006) and compare against the ground truth estimates from that experiment. We find that our quality estimates closely match the ground truth data. We then show that the Poisson model is a good fit for the Reddit and Hacker News voting data, even when evaluated on out-of-sample data during cross-validation.

### 5.1 MusicLab

Participants in the MusicLab experiment (Salganik, Dodds, and Watts 2006) were shown a list of unknown songs that they could listen to and download. The user interface resembles that of Reddit and Hacker News in the sense that songs were ranked vertically on the page, and users interact with content in a similar two-step way; they can choose to listen to a song (read/view an article) and/or download it (vote on it), but only downloading influences the future state of the ranking. When participants entered the site, they were assigned to 1 of 8 treatment worlds or the control world. In the treatment worlds, songs in world  $w$  were ranked by the number of downloads in  $w$ , and these download counts were shown to users. In the control world songs were displayed in a random order and download counts were not displayed.

The number of downloads that each song has in the control world is exactly our definition of article quality, and hence we can use that data to test the Poisson regression method. We use data from the treatment worlds to train the model, estimate qualities as detailed in the previous section, and compare against the observed number of downloads in the control world. We fit the following model:

$$d_i^{t,w} \sim \text{Pois}(\exp\{q_i + p_i^{t,w} + \beta_{score} \cdot \log(S_i^{t,w})\})$$

Where  $d_i^{t,w}$  is a binary variable for whether the  $t^{th}$  user in world  $w$  downloaded song  $i$ ,  $p_i^{t,w}$  was the position that

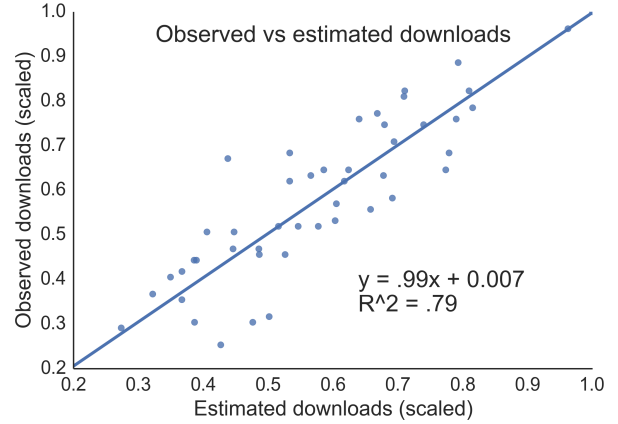


Figure 1: Observed number of downloads (scaled) in the control world versus estimated downloads (scaled) for the MusicLab experiment. Each data point represents a single song in the experiment.

song  $i$  appeared in for that user, and  $S_i^{t,w}$  is the number of downloads of song  $i$  in world  $w$  when user  $t$  visited. The age factor is dropped because most users only participated once and hence there's no temporal aspect. Unlike Reddit or Hacker News, downloads are a binary variable rather than a count variable but this is not an issue because the Poisson method was originally introduced in the CTR literature for binary data (Chen and Yan 2012). Using a logistic regression doesn't yield any significant change in results.

We then use the fitted  $q_i$  parameters and equation 2 to predict the expected number of downloads in the control world. The results are shown in figure 1 and demonstrate that estimated qualities are fairly close to the ground truth data (Pearson correlation = .88,  $\rho < 10^{-15}$ ). We have scaled such that the maximum number of downloads in both the observed and predicted values is equal to 1. The line of best fit for the unscaled values has a slope of 2.3, indicating that our raw estimates underestimate downloads by approximately 65%. This is a large underestimate for the absolute number of downloads but the good linear fit indicates that the Poisson regression accurately estimates the relative number of downloads.

### 5.2 Reddit and Hacker News

Given that our model effectively recovers ground truth data from the MusicLab experiment, we now evaluate the fit of the Poisson model to Reddit and Hacker News voting data. Rather than evaluating against the final popularity of each article, we examine the fit to the time-series data. For each observation  $v_i^t$  of the number of votes article  $i$  received at time  $t$ , our model makes a prediction of  $\hat{v}_i^t$  equal to the conditional mean of the Poisson distribution, i.e.:

$$\hat{v}_i^t = \exp\{q_i + p_i^t + \beta_{age} \cdot age_i^t + \beta_{score} \cdot \log(S_i^t)\}$$

For Reddit this only predicts the number of votes on an article, not the increase in score  $s_i^t$ . As described in section 4, we multiply the predicted number of votes by the difference

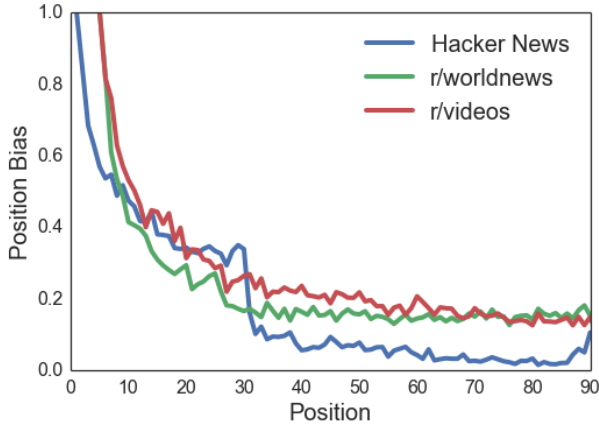


Figure 2: Estimated position bias for top 90 positions for Hacker News and select subreddits. Position biases have been normalized such that maximum position bias is 1.

in the conditional upvote and downvote probability. Recall that  $r_i^{up}, r_i^{down}$  is the observed ratio of upvotes and downvotes to total votes for article  $i$ . The predicted increase in score is:

$$\hat{s}_i^t = \hat{v}_i^t \cdot (r_i^{up} - r_i^{down})$$

### 5.3 Results

We evaluate the accuracy of the  $(v_i^t, \hat{v}_i^t)$  predictions for Hacker News and  $(s_i^t, \hat{s}_i^t)$  predictions for Reddit using coefficient of determination ( $R^2$  value), mean absolute error, and mean squared error. In addition to reporting the accuracy when we train and fit on the entire dataset, we also run a 5-fold cross validation. Specifically, after dividing each dataset into 5 equal partitions, we hold out one partition, train on the remaining 4 partitions, and then make predictions for the held-out set. We repeat this 5 times so that each partition is treated as the held-out set once. We report the average accuracy statistics over the 5 train/test splits.

The results are shown in table 2. The model performs well for both in-sample and out-of-sample prediction, capturing between 50% and 80% of the variance in the voting data. While the fit is reasonably good, we note that the variance in the dataset is significantly larger than the model assumes. The Poisson model assumes that conditional variance is equal to the conditional mean but this doesn't hold in our data. While this assumption on the variance isn't necessary for estimation of the maximum likelihood parameters, it suggests that the Poisson model can be improved upon.

The predictions in table 2 were made using the full Poisson model but we also experimented with two reduced models by removing the score and age effects. Table 3 shows the average cross-validated  $R^2$  values for the base Poisson model with just article and position factors, a model with article, position, and a time factor, and the full model. In most cases, gains in accuracy are driven primarily by the addition of the time-decay factor but the score effects do help. However score effects caused odd behaviors in some cases, as we discuss in the next section.

	In Sample Predictions			Out of Sample Predictions		
	$R^2$	MAE	MSE	$R^2$	MAE	MSE
r/pics*	0.76	1.09	7.30	0.62 (0.01)	1.14 (0.01)	8.51 (0.40)
r/videos	0.79	1.15	9.62	0.65 (0.03)	1.22 (0.01)	13.64 (2.59)
r/todayilearned	0.71	1.75	22.66	0.61 (0.03)	1.85 (0.02)	32.24 (3.74)
r/news*	0.56	1.11	3.63	0.57 (0.01)	1.14 (0.01)	3.87 (0.18)
r/worldnews	0.57	1.27	9.10	0.52 (0.01)	1.32 (0.01)	10.65 (1.17)
Hacker News	0.69	0.70	1.82	0.65 (0.01)	0.74 (0.01)	2.08 (0.11)

Table 2: Accuracy metrics for the Poisson model. In-sample values show the fit of the model to the dataset when all data is used. Out-of-sample predictions are trained on a training set and predicted for a test over 5 fold cross-validation. The mean values over 5 folds are reported (standard errors shown in parentheses).

	Base	Base + Time	Full
r/pics	0.56	0.58	0.62
r/news	0.53	0.55	0.59
r/worldnews	0.51	0.51	0.52
r/todayilearned	0.61	0.59	0.61
r/videos	0.63	0.58	0.65
Hacker News	0.51	0.63	0.65

Table 3: Average  $R^2$  values over cross-fold validation for the three models. Base model refers to the Poisson model with just quality and position effects.

## 6 Analysis

We first use these estimates to quantify position bias on Reddit and Hacker News. The relative view rate for position  $j$  is computed as  $e^{p_j}$ , where  $p_j$  is fit from the Poisson regression, and scaled so the maximum view rate in a subreddit is equal to 1. Figure 2 shows the relative view rates for the top 90 positions of Hacker News, r/worldnews, and r/news (we exclude other datasets for visualization purposes but they show similar trends). The curves for the subreddits begin at position 5 because we discard observations from the top 5 positions of each subreddit (see the appendix for the reasoning behind this). Each dataset shows an exponential decline in view rate but Hacker News has a particularly sharp drop at its page break (position 30 to 31), whereas the subreddits display a smoother decline. The general shape of position bias is consistent with estimates from other platforms (Krumme et al. 2012),(Lerman and Hogg 2014).

For two subreddits, r/news and r/pics, we observed an odd interaction between position bias estimates and effect of social influence. When we fit the full model, the resulting parameters implied that very low positions (100 to 200) received more views than the top 50 positions. Estimates for the top 50 positions seem to have been reduced because the effects were “pushed” into the social influence parameter,  $\beta_{score}$ . Although the full model was marginally more accurate, we chose to drop the score term for these two datasets because of this unintuitive behavior.

### 6.1 Quality and Popularity

We now measure the relationship between estimated quality and observed popularity. We use the  $q_i$  parameters when the model is fit on the entire dataset (no train/test splits) and

equations 2 and 3 to estimate quality on Hacker News and Reddit, respectively.

Figures 3a and 3b show scatter plots of observed popularity versus estimated quality for Hacker News and for r/news. Hacker News has the strongest correlation between score and quality, while r/news has one of the weakest relationships. Figure 3c shows the relationship for all subreddits; in order to compress everything in one plot, observed scores are scaled such that the maximum score in each subreddit is 1, and then those scaled scores are log-transformed. The relationship between quality and popularity is consistent with expectations from the MusicLab experiments. Popularity is generally increasing with quality but articles of similar quality can have large differences in popularity. However we find that there are few instances of a mediocre quality article becoming one of the most popular articles in a subreddit, and few instances of high quality articles ending up with low scores. In general, the relationship between popularity and quality is stronger on Reddit and Hacker News than the MusicLab experiment. The first column of table 4 lists the Spearman correlation coefficients between quality and popularity. Hacker News has the strongest relationship with a correlation of .8 and r/worldnews has the weakest with a correlation of .54.

We had initially expected the quality-popularity relationship to be weaker on Hacker News than Reddit because of the lack of the downvote. Our theory was that a low quality article that made it to the front page of Hacker News could remain there for a long time and become popular because there was no ability to downvote it off. This theory is partially true; the second column in table 4 shows the relationship between quality and total views. We estimate total views by  $\sum_t e^{p_i^t}$ , i.e. the sum of position biases for the positions that article  $i$  appeared in during its lifetime. The relationship between total views and quality on Hacker News is much weaker than on Reddit, indicating that lower quality articles are being seen comparatively more often on Hacker News. However this did not translate to a weakened quality-popularity relationship as we had expected.

## 6.2 Discussion

There is one important caveat to these results. Many articles submitted to Reddit and Hacker News fail to gain any votes and quickly disappear. For example, there were 5000 articles submitted to Hacker News over the period of observation but only 1500 of them ever appeared in the top ranking. On Reddit, over half of articles were discarded because they appeared for less than an hour in the range of positions studied. These ignored articles did not generate enough observations to be included in our analysis. So when we state that the relationship between quality and popularity is fairly strong, we must interpret that as only being among a set of articles that received at least a reasonable amount of attention. In the Reddit dataset, the median article received 38 votes (upvotes plus downvotes), while the median Hacker News article received 21 votes, with a minimum of 3 votes in each case. Its likely there are a number of high quality articles that were discarded from this study because they didn't generate enough observations. Developing methods to estimate

	Score	Views
Hacker News	.80	0.49
r/todayilearned	.75	0.81
r/videos	.63	0.70
r/worldnews	.54	0.70
r/news	.59	0.75
r/pics	.63	0.77
MusicLab	.57	0.35

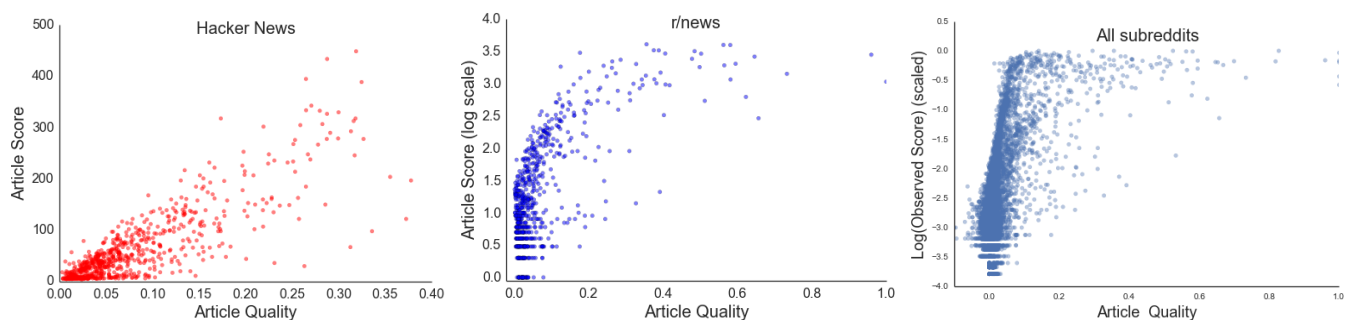
Table 4: Spearman correlation between estimated quality and observed score in the first column, and quality and estimated views in the second column. These results suggest that the relationship between quality and views is stronger on Reddit than Hacker News, despite a stronger relationship between quality and popularity on Hacker News.

properties of articles with a small number of observations is an interesting direction for future work.

## 7 Reposts

As discussed in the last section, many articles on Reddit or Hacker News go almost completely ignored. A recent estimate shows that over half of links on Reddit receive at most 1 upvote (Olson 2015). The work of Gilbert (2013) shows that it isn't because this content is necessarily bad; Gilbert finds that over half of popular images on Reddit were submitted and ignored a few times before they became popular. This seems problematic for Reddit's role as an aggregator of the most interesting and popular content on the web. However one subtle point of (Gilbert 2013) is that those images eventually became popular, even if it took a few reposts. Although Reddit's voting mechanism failed to popularize some good content, the reposting behavior of Redditors corrected this failure. In this section we briefly explore the role of reposts in popularizing content on Reddit. We find evidence that the number of reposts of an article is positively correlated with its external popularity. Unfortunately we cannot use the methods from section 4 to estimate quality because the scope of our time-series data is too limited to capture much reposting behavior. Instead we study how externally popular content, that is content whose popularity is being driven by another site, gets discovered on Reddit. We limit this study to Youtube videos submitted to Reddit and use Youtube views as the external popularity of an article.

We study all videos that were uploaded to Youtube and submitted to r/videos in 2012. We're left with a set of 61,110 unique videos after removing videos we were unable to retrieve metadata for. These videos were submitted a total of 91,841 times to Reddit; 11,297 of these videos were submitted multiple times, generating a total of 42,028 reposts to Reddit. Figure 4 shows a scatter plot of number of posts to Reddit versus Youtube views for each video. There's a strong positive relationship between views and submissions (Spearman correlation = .46,  $\rho = 0$ ), suggesting that users submit popular Youtube videos more frequently. Videos with more than 1 million views, of which there are approximately 6400, were submitted twice as of-



(a) Observed popularity versus estimated quality for Hacker News. X-axis is truncated for visualization purposes but only a few data points were omitted. (b) Observed popularity versus estimated quality on r/news. (c) Observed popularity versus estimated quality for all subreddits. Observed scores are first scaled so that the maximum score in each subreddit is equal to 1 and then log-transformed.

Figure 3: A sample of popularity versus estimated quality plots for Hacker News and Reddit.

ten to Reddit; the mean and median number of submissions for all videos are 1.5 and 1 while the mean and median for videos with more than one million views are 3.6 and 2. A Mann-Whitney U test confirms that the distribution of reposts of videos with more than 1 million views is significantly different than videos with less than 1 million views ( $\rho = 0$ ).

These reposts are actually responsible for surfacing many Youtube videos that would have gone unnoticed on Reddit otherwise. This makes intuitive sense because the more times a video is submitted, the greater the chance it has to become popular. We define a video to be “discovered” on Reddit if it’s score was in the top 10% of scores of posts to r/videos in 2012. Given the large amount of videos with no attention, this only amounts to achieving a score of 23 or greater. We find that only 59% of videos with more 1 million views were discovered on their first submission, while 76% of videos with less than 1 million views were discovered on their first submission. This difference is likely caused by the fact that more popular videos were submitted more times; we suspect that if videos with less than 1 million views were submitted as often, then these numbers would be more equal. This conclusion, that reposts help popularize many videos, is similar to the conclusion of (Gilbert 2013) but our analysis further shows that reposts are particularly instrumental in popularizing videos that are externally popular.<sup>9</sup>

## 8 Limitations

This study is fundamentally an observational study and is accompanied with a number of limitations. Our largest limitation is the lack of ground truth data for Reddit and Hacker News. We are encouraged by our method’s ability to recover ground truth from the MusicLab experiment but we recognize that although Reddit and Hacker News are similar in some ways, they are fundamentally different.

<sup>9</sup>We cannot rule out the possibility that number of submissions to Reddit is causing a rise in Youtube views but this seems unlikely given the relative size of Reddit versus Youtube in 2012.

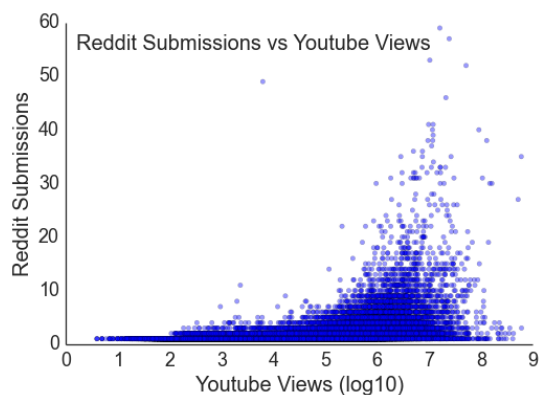


Figure 4: Number of submissions versus Youtube views for all Youtube videos submitted to r/videos in 2012.

Our statistical model makes a number of simplifying assumptions for sake of tractability. The main limitation is the implicit assumption that Hacker News and each subreddit operates as a closed system of attention. Our model cannot appropriately handle the case where a post receives significant external attention, e.g. from Twitter, and this will bias our estimates of that article’s quality. This is particularly problematic on Reddit because high-scoring posts on individual subreddits will appear on Reddit’s front page. We have attempted to reduce this issue by removing observations where a post likely appeared on or near Reddit’s front page but biases likely remain. On the other hand, only a small fraction of posts appear on Reddit’s front page. Our model also assumes that the position parameters are fixed over time. Obviously there are more people viewing Reddit on Monday mornings than Saturday nights but our model doesn’t explicitly account for this. We attempted to add time-of-day effects but found that it increased over-fitting without yielding a noticeable gain in model accuracy. Instead we limit our data to observations of Reddit and Hacker News on weekdays between 6 am and 8 pm EST. We leave it as future work to improve the model to account for such



time effects.

## 9 Conclusion and Future Work

This paper tries to understand the relationship between intrinsic article quality and popularity in two social news aggregators. The heart of the problem is developing a method to estimate the counterfactual popularity of an article in a world without bias from observed popularity data. To this end, we proposed a simple Poisson regression model whose fitted parameters allow us to estimate article quality. We found that the most popular content on Reddit and Hacker News are, for the most part, the highest quality articles amongst the set of articles that receive a moderate amount of attention.

The method presented in this paper is only an initial approach to quality estimation, and can be improved in many ways. The most immediate is expanding the model to include a richer set of temporal features, such as commenting data, and engineering the method to handle much larger data sets. Although the role of social networks is relatively small on Reddit and Hacker News, prior work demonstrates that prediction accuracy can be improved by incorporating the social networks of users who post articles (Lerman and Hogg 2010). Perhaps the most interesting future work is studying the voting dynamics when an article is first submitted. Early voters play an interesting gate-keeping role because a number of early downvotes on an article effectively buries the article and denies the broader community a chance to vote on that article. Quantifying the influence of early voters on popularity and its implications is an interesting direction for future research.

## References

- Bakshy, E.; Hofman, J. M.; Mason, W. A.; and Watts, D. J. 2011. Everyone’s an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 65–74. ACM.
- Bakshy, E.; Rosenn, I.; Marlow, C.; and Adamic, L. 2012. The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, 519–528. ACM.
- Bandari, R.; Asur, S.; and Huberman, B. A. 2012. The pulse of news in social media: Forecasting popularity. In *ICWSM*.
- Chen, Y., and Yan, T. W. 2012. Position-normalized click prediction in search advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 795–803. ACM.
- Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J. 2014. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, 925–936. International World Wide Web Conferences Steering Committee.
- Das, S., and Lavoie, A. 2014. The effects of feedback on human behavior in social media: An inverse reinforcement learning model. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 653–660. International Foundation for Autonomous Agents and Multiagent Systems.
- Dupret, G. E., and Piwowarski, B. 2008. A user browsing model to predict search engine click data from past observations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 331–338. ACM.
- Friggeri, A.; Adamic, L. A.; Eckles, D.; and Cheng, J. 2014. Rumor cascades.
- Gilbert, E. 2013. Widespread underprovision on reddit. In *Proceedings of the 2013 conference on Computer supported cooperative work*, 803–808. ACM.
- Gillespie, T. 2011. Our misplaced faith in twitter trends. *Salon*.
- Goel, S.; Watts, D. J.; and Goldstein, D. G. 2012. The structure of online diffusion networks. In *Proceedings of the 13th ACM conference on electronic commerce*, 623–638. ACM.
- Hogg, T., and Lerman, K. 2009. Stochastic models of user-contributory web sites. *ICWSM*.
- Hogg, T., and Lerman, K. 2014. Effects of social influence in peer online recommendation. *arXiv preprint arXiv:1410.6744*.
- Krumme, C.; Cebrian, M.; Pickard, G.; and Pentland, S. 2012. Quantifying social influence in an online cultural market. *PLoS one* 7(5):e33785.
- Lakkaraju, H.; McAuley, J. J.; and Leskovec, J. 2013. What’s in a name? understanding the interplay between titles, content, and communities in social media. In *ICWSM*.
- Lampe, C., and Johnston, E. 2005. Follow the (slash) dot: effects of feedback on new members in an online community. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, 11–20. ACM.
- Lampe, C., and Resnick, P. 2004. Slash (dot) and burn: distributed moderation in a large online conversation space. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 543–550. ACM.
- Leavitt, A., and Clark, J. A. 2014. Upvoting hurricane sandy: event-based news production processes on a social news site. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, 1495–1504. ACM.
- Lerman, K., and Hogg, T. 2010. Using a model of social dynamics to predict popularity of news. In *Proceedings of the 19th international conference on World wide web*, 621–630. ACM.
- Lerman, K., and Hogg, T. 2014. Leveraging position bias to improve peer recommendation. *PLoS one* 9(6):e98914.
- Muchnik, L.; Aral, S.; and Taylor, S. J. 2013. Social influence bias: A randomized experiment. *Science* 341(6146):647–651.
- Nocedal, J. 1980. Updating quasi-newton matrices with limited storage. *Mathematics of computation* 35(151):773–782.
- Olson, R. 2015. Over half of all reddit posts go completely ignored.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Pinto, H.; Almeida, J. M.; and Gonçalves, M. A. 2013. Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, 365–374. ACM.

Richardson, M.; Dominowska, E.; and Ragno, R. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, 521–530. ACM.

Salganik, M. J., and Watts, D. J. 2008. Leading the herd astray: An experimental study of self-fulfilling prophecies in an artificial cultural market. *Social Psychology Quarterly* 71(4):338–355.

Salganik, M. J.; Dodds, P. S.; and Watts, D. J. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science* 311(5762):854–856.

Sipos, R.; Ghosh, A.; and Joachims, T. 2014. Was this review helpful to you?: it depends! context and voting patterns in online content. In *Proceedings of the 23rd international conference on World wide web*, 337–348. International World Wide Web Conferences Steering Committee.

Szabo, G., and Huberman, B. A. 2010. Predicting the popularity of online content. *Communications of the ACM* 53(8):80–88.

Wang, T.; Wang, D.; and Wang, F. 2014. Quantifying herding effects in crowd wisdom. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1087–1096. ACM.

## A Data Issues

**Observation Inclusion Criteria** As with any study, we only study a subset of the data. Here’s the list of our criteria for including observations.

1. Data must have been observed between 6am and 8pm EST on a weekday.
2. For Reddit, we limit observations to only include positions in a certain range of  $[p_{min}, p_{max}]$ .  $p_{min}$  is defined to be 5 for all subreddits, except for r/pics where  $p_{min}$  is 15. We do this to avoid observations of an article that also appeared on or near the front page of Reddit. We define  $p_{max}$  to be median of the distribution of article’s initial positions within a subreddit.
3. We discard observations of articles when they are older than 12 hours. Since our model accounts for time decay, this is primarily to reduce the size of the dataset. After 12 hours, over 95% of articles have received over 90% of votes that they will ever receive.
4. After removing data according to the above criteria, we finally discard any article that we don’t have at least 5 observations for.

## Vote Fuzzing

During the period of observation, Reddit used a practice called “vote fuzzing”. Reddit displayed the upvotes, downvotes, and score (difference between upvotes and downvotes) but a (semi-random) constant would be added to displayed upvotes and downvotes. This kept the score accurate but changed the ratio of upvotes to total votes. As of June 18, 2014 this process was stopped<sup>10</sup>. Reddit no longer displays the individual number of upvotes and downvotes, and instead displays the score and the ratio of upvotes to total votes for each article. They claim the ratio and score are fairly accurate.

Our data was primarily collected in the periods before the change but we were able to use the change in policy to retroactively “de-fuzz” the observed upvotes and downvotes. Since Reddit is now displaying the true score,  $s^{true}$  and true ratio  $\tau^{true}$ , one can easily recover the true number of upvotes and downvotes. However we cannot recompute the true values for our time-series data because we cannot retrieve the  $s^{true}$  and  $\tau^{true}$  for articles at some arbitrary point in the past. Instead, we take advantage of the fact that articles on Reddit receive almost zero activity after they are a few days old. Thus the state of an article in our collected data after 48 hours is very close to the state of the article as it would be a few months later. In August 2014, we retrieved the current  $s^{true}$ ,  $\tau^{true}$  for these articles and used those values to calculate  $u^{true}$  and  $d^{true}$ .

We used this data to train a random forest regressor<sup>11</sup> to predict on the following to predict  $u^{true}$  using  $u^{obs}$ ,  $s^{obs}$ ,  $r^{obs}$  as features, where  $(u^{obs}, s^{obs}, r^{obs})$  are the observed upvotes, score, and upvote ratio at the time we scraped the data. This method is quite accurate (average  $r^2 = .96$  with 10 fold cross validation). We then use this regressor to generate the “true” ups and down for all data we collected. We emphasize that while this is not the “true” data, this method is far more accurate than using the “fuzzed” votes Reddit displayed prior to this change. Vote fuzzing appears to have inflated the number of votes observed at the upper tail of the distribution. This observation is consistent with anecdotal evidence from Reddit users, moderators, and administrators.

As a final note, collecting voting data at frequent intervals is now considerably more difficult because Reddit has since changed their API. The ratio of upvotes to total votes isn’t available when retrieving information in batch, only when retrieving the information for a single article. So instead of retrieving information about 1000 articles 1 API call, it now requires 1000 API calls. Collecting that information at regular intervals is impossible to do while respecting their rate limits.

<sup>10</sup><http://www.reddit.com/28hjga>

<sup>11</sup>We used the implementation from the scikit-learn Python module (Pedregosa et al. 2011).