Accounting for Taste: Ranking Curators and Content in Social Networks

Haizi Yu[1] **Biplab Deka**[1] **Jerry O. Talton**[2] **Ranjitha Kumar**[1,2]

[1]University of Illinois at Urbana-Champaign [2]Apropose, Inc.

{haiziyu7,deka2}@illinois.edu, jerry@apropose.com, ranjitha@illinois.edu

ABSTRACT

Ranking users in social networks is a well-studied problem, typically solved by algorithms that leverage network structure to identify influential users and recommend people to follow. In the last decade, however, *curation* — users sharing and promoting *content* in a network — has become a central social activity, as platforms like Facebook, Twitter, Pinterest, and GitHub drive growth and engagement by connecting users through content and content to users. While existing algorithms reward users that are highly active with higher rankings, they fail to account for users' curatorial *taste*. This paper introduces CuRank, an algorithm for ranking users and content in social networks by explicitly modeling three characteristics of a good curator: discerning taste, high activity, and timeliness. We evaluate CuRank on datasets from two popular social networks — GitHub and Vine — and demonstrate its efficacy at ranking content and identifying good curators.

ACM Classification Keywords

H.3.1 Information Storage and Retrieval: Content Analysis and Indexing

Author Keywords

Social networks; Curation; Content; Ranking

INTRODUCTION

As social networks have evolved, they have shifted focus from directly connecting users to connecting users through *content* [4, 14]. Popular social networks are now platforms for disseminating and discovering content: users share links on Twitter, posts on Facebook, articles on LinkedIn, images on Pinterest, and code on GitHub. As a consequence, these networks are now driven by *curation*: users that consistently share and promote good content create strong network effects that benefit the community as a whole [10, 15, 3, 13].

A key problem in social networks is *ranking*. Assessing the relative import of users and content is essential to providing good search results, recommending content to consume, and suggesting users to follow [12, 22]. Most prior work in this area has been predicated on the observation that the network structure that links users and content can be leveraged to inform estimates of rank [23, 17, 19].

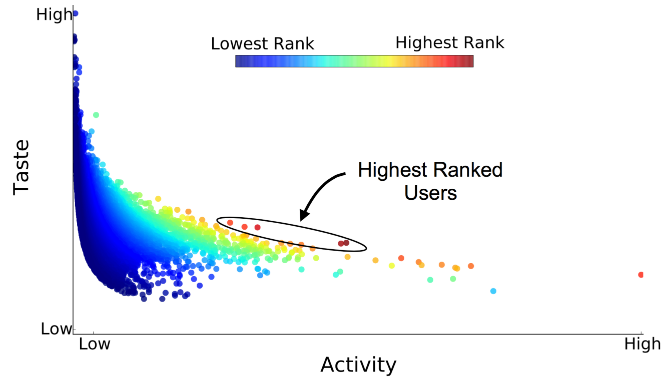


Figure 1: GitHub users ranked by CuRank based on their curatorial activities. The highest ranked users exhibit both good taste and high activity.

However, existing algorithms fail to adequately model the most important activity in modern social networks: curation. They reward highly-active users with higher rankings, but fail to account for users' *taste*. This paper introduces *CuRank*, an algorithm for ranking users and content that explicitly models users' curatorial activity by recognizing three characteristics of a *good curator*:

1. **Taste**. A good curator is discerning, and tends to promote important content over unimportant content.
2. **Activity**. A good curator is engaged in the community, and curates much important content.
3. **Timeliness**. A good curator promotes content early, often before it is validated by the community.

CuRank improves on prior work in three ways. First, the algorithm models a tradeoff between the number of items promoted by a user and the average rank of the items that are promoted: therefore, a malicious user cannot obtain a high rank by promoting every item in the network. Second, it models the inherent asymmetry in how user and content ranks should interact. While a user should be penalized for indiscriminately promoting too much content, content should not be penalized for being promoting by too many users. Third,

CHI'16, May 07 – 12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05...$15.00

DOI: http://dx.doi.org/10.1145/2858036.2858219

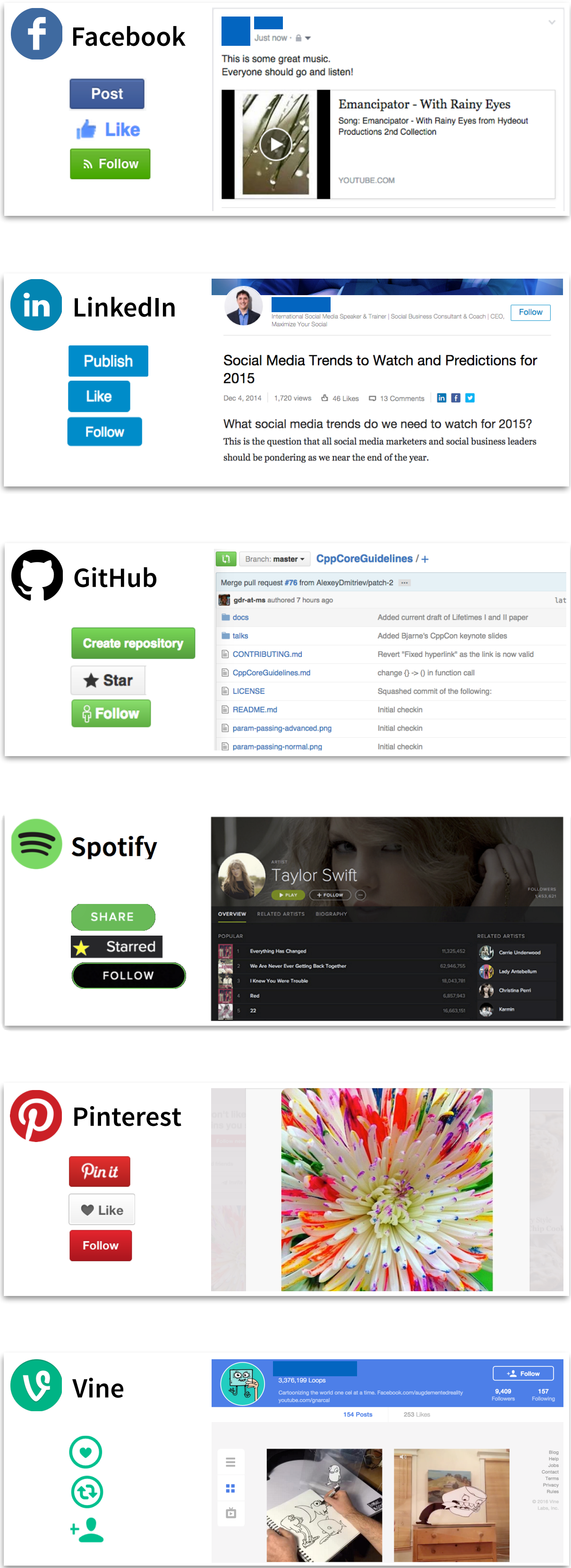


Figure 2: Popular social networks and the content-sharing mechanisms they expose.

the algorithm models the timeliness of a user by accounting for the order in which users promote content: by giving earlier promotions more weight, new, important content becomes easier to discover.

This paper describes the CuRank algorithm, demonstrates its use on a sample network, and reports results on datasets extracted from two popular real-world social networks: GitHub and Vine. We demonstrate that CuRank is able to successfully rank users based on their curation, balancing between taste and activity (Figure 1). In addition, we show how CuRank's temporal model can be used to distinguish between different types of curatorial behaviors in an algorithmic way. Finally, we illustrate an important emergent property of CuRank: that the algorithm makes it possible for new users to "break into the winner's circle" of rankings, provided they demonstrate good taste.

CONTENT AND CURATION IN SOCIAL NETWORKS

In recent years, the line between social networks and content feeds has blurred. As predominantly-social networks (like Facebook and LinkedIn) increasingly encourage the production and dissemination of content, online communities focused around content (like GitHub and Spotify) have added more and more support for social interactions [6].

Driving this convergence is a desire for faster network growth and greater user engagement. The "come for the tool, stay for the network" model popularized by platforms like Pinterest offers a compelling single-player content mode (*e.g.*, save an image to a Pinterest board) to build a critical mass of users, and then capitalizes on network effects to foster an active community [8]. For more entrenched social networks like LinkedIn, fresh content drives daily engagement amongst users in a way that better network tools cannot [21].

In this brave new social network world, *curation* is central to the distribution of content. While the type of content varies between networks, the curatorial mechanisms exposed to users follow a few standard forms (Figure 2). Users curate by posting their own work (*e.g.*, sharing code on GitHub), collecting content from external sources (*e.g.*, pinning an image on Pinterest), and sharing existing content within the network (*e.g.*, retweeting on Twitter).

The ability to identify good curators affords social networks a number of advantages. For one, reliable curators can be used to vet new content more quickly than the network as a whole. For another, giving social credit to skilled curators (*e.g.*, Spotify's "Found Them First" program [5]) can incentivize future curation and strengthen recommendations to follow users or content streams. To these ends, we introduce CuRank as a general, principled system for ranking curators and content in social networks.

RELATED WORK

Network ranking has been studied extensively in the literature, for instance in the context of academic citations [25, 7, 28], the link structure of the Web [23, 17, 19], and social networks [18, 27, 12, 15]. The common goal of this line of research is to measure the relative "import" of nodes in a network by leveraging the graph structure of the connections between them. The virtue of such approaches to ranking is that they are largely domain agnostic: one need not build features to understand the content of a scientific article to observe that it is highly cited.

Network Statistics

The most basic ranking algorithms are based on simple network statistics, like the number of backlinks to a page on the Web or the number of "followers" a user has on Twitter. These statistics are easy to implement and interpret — and are therefore widely employed — but provide no mechanism for differentiating between the relative import of connections. As a consequence, such measures are susceptible to manipulation by malicious users (*e.g.*, via link farms), and fail to preserve common-sense ranking properties: for instance, that being cited by a Nobel prize–winning scientist with thousands of publications is a better indication of importance than a citation in an undergraduate's first publication.

Rank Propagation

A class of more powerful ranking methods are predicated on the notion of *propagating rank* through the network, so that links from more important nodes convey more import in turn. Many of these algorithms such as PageRank [23], HITS [17], and SALSA [19] were developed to rank pages on the Web.

PageRank [23] computes the probability of a random surfer visiting a node in the network from the stationary distribution of a Markov chain defined by the network's link structure, and uses these probabilities as ranks. Since the PageRank of a node is defined by the quantity *and* import of the nodes that link to it, the algorithm is difficult to manipulate.

HITS [17] calculates network rankings in settings with two types of entities — *hubs* and *authorities* — by modeling a *mutually reinforcing relationship* (MRR) between them. An important hub is a node that links to many important authorities, and an important authority is a node that is linked to by many important hubs. Based on the MRR, HITS computes hub and authority rankings as the principle left and right singular vectors of the network's incidence matrix. SALSA [19], a probabilistic extension of HITS, computes hub and authority ranks as the stationary distributions of two Markov chains.

Ranking Users in Social Networks

Network-based ranking algorithms have been applied to social networks to identify influential users [18, 27] and recommend people to follow [12, 15]. These approaches leverage the *homophily* hypothesis — that similar users attract one other — to compute network-based rankings over following/follower graphs. These algorithms have been deployed at scale, including in Twitter's user recommendation service, "Who to Follow," which uses SALSA as its core ranking algorithm [12].

This paper focuses on a different, but equally-important problem: identifying users who consistently curate the best content. In the same way that HITS modeled the mutually reinforcing relationship between hubs and authorities, CuRank models the MRR between curators and content: an important curator promotes important content, and an important piece of content is promoted by important curators. This relationship is encoded in the *curation graph* of a network, comprised of users and content and the promotional links between them.

At first glance, naïvely adapting HITS or SALSA to curation seems like a promising strategy, but existing MRR algorithms suffer from a fatal flaw in this domain: they fail to account for *taste*. The rankings produced by HITS favor maximally-active users who indiscriminately promote content without regard to quality. CuRank explicitly models the tradeoff between promoting many items — activity — and promoting important items — taste.

THE CURANK ALGORITHM

Given a social network, we extract a *curation graph*

$$\mathcal{G} = (\mathsf{U}, \mathsf{C}; \mathsf{E}),$$

where U is the set of *user* nodes in the network, C is the set of *content* nodes, and $\mathsf{E} \subseteq \mathsf{U} \times \mathsf{C}$ is the set of user–content *promotions* (*e.g.*, posts, likes, stars, pins, *etc.*).

CuRank takes a curation graph as input, and outputs two *ranking* functions $r_{\mathsf{U}} : \mathsf{U} \to [0, 1]$ and $r_{\mathsf{C}} : \mathsf{C} \to [0, 1]$ for users and content, respectively. Following convention [23], these functions are normalized so that $\sum_u r_{\mathsf{U}}(u) = \sum_c r_{\mathsf{C}}(c) = 1$.

Timeliness, Activity, and Taste

To calculate $r_{\mathsf{U}}(\cdot)$ and $r_{\mathsf{C}}(\cdot)$, we formalize the notions of *timeliness*, *activity*, and *taste*. In a curation graph, taste and activity are node quantities, dually defined over users and content in a mutually-reinforcing way. Timeliness, in contrast, is an edge quantity, defined over promotions, and used to modulate the flow of rank between users and content through the network.

Timeliness

The *timeliness* of a promotion $p \in \mathsf{E}$ is given by a function $t : \mathsf{E} \to \mathbb{R}_{\geq 0}$, which measures the promotion's temporal importance. The CuRank algorithm admits a number of useful timeliness functions, and the best choice for a particular network may vary depending on the goals of the ranking.

In the simplest case, where temporal information is not available or we do not wish to differentiate between promotions, we set $t(\cdot) = 1$. When timestamps are available, we may choose any monotonically decreasing function to give lower weight to later promotions of a particular piece of content, for instance

$$t(p) = t(\langle u, c \rangle) = \exp(-\beta \cdot k(\langle u, c \rangle)),$$

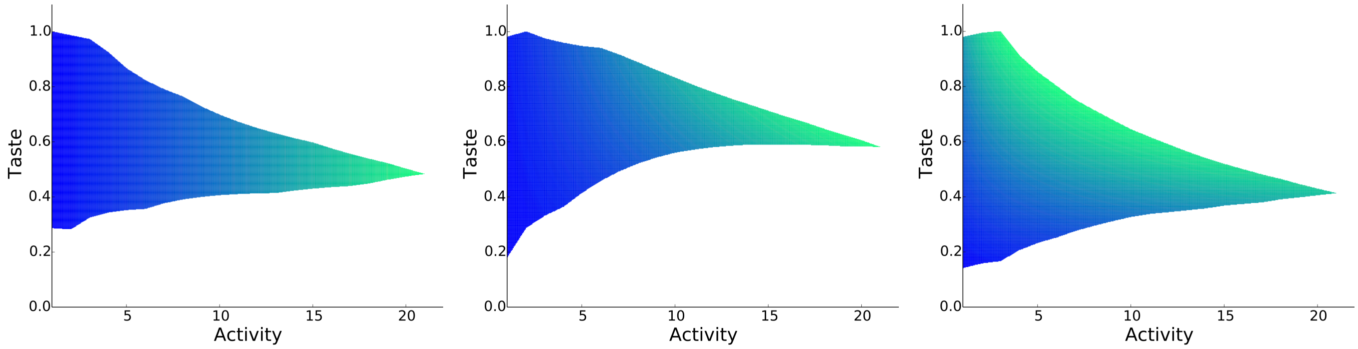
where β is a scaling constant and $k(\langle u, c \rangle)$ is a timestamp function denoting that p is the kth promotion of c.

In networks with more than one kind of promotional activity, we can use a piecewise $t(\cdot)$ to differentiate between them, for instance assigning a higher weight to promotions that introduce external content to the network than those that share content internally.

Activity

The *activity* of a user $u \in \mathsf{U}$ is a count of the user's promotions, weighted by their timeliness:

$$A_{\mathsf{U}}(u) = \sum_{\langle u, c' \rangle \in \mathsf{E}} t(\langle u, c' \rangle).$$



(a) Number of Promotions **(b)** HITS **(c)** CuRank

Figure 3: The gamut of activity and taste scores for a user entering a simple network under three ranking algorithms: number of promotions, HITS, and CuRank. Different algorithms assign higher curatorial rank (bright green) to different parts of the gamut. The rankings produced by HITS are almost perfectly correlated with activity, while CuRank assigns high rank to users who balance activity and taste.

The activity score of a piece of content $c \in \mathsf{C}$ is similarly defined:

$$A_{\mathsf{C}}(c) = \sum_{\langle u',c\rangle \in \mathsf{E}} t(\langle u', c\rangle).$$

In this way, users who promote more content will have higher activity scores, as will content that is promoted by more users.

Taste

The *taste* of a user $u \in \mathsf{U}$ is the weighted average rank of the content the user promotes:

$$T_{\mathsf{U}}(u) = \sum_{\langle u,c'\rangle \in \mathsf{E}} t(\langle u, c'\rangle)\, r_{\mathsf{C}}(c') \Big/ A_{\mathsf{U}}(u).$$

Similarly, the taste score of a piece of content $c \in \mathsf{C}$ is the weighted average rank of the users who promote it:

$$T_{\mathsf{C}}(c) = \sum_{\langle u',c\rangle \in \mathsf{E}} t(\langle u', c\rangle)\, r_{\mathsf{U}}(u') \Big/ A_{\mathsf{C}}(c).$$

Accordingly, users who promote highly-ranked content will have higher taste scores, as will content that is promoted by highly-ranked users.

Ranking Equations

With these formulations of timeliness, activity, and taste, we can define the ranking functions for users and content. While taste and activity are defined symmetrically between users and content, the ranking functions $r_{\mathsf{U}}(\cdot)$ and $r_{\mathsf{C}}(\cdot)$ are given by

$$r_{\mathsf{U}}(u) = T_{\mathsf{U}}(u) \cdot A_{\mathsf{U}}(u)^{1-\alpha}, \tag{1}$$

$$r_{\mathsf{C}}(c) = T_{\mathsf{C}}(c) \cdot A_{\mathsf{C}}(c), \tag{2}$$

where $\alpha \in [0, 1]$. While both rankings are positively correlated with taste and activity, α penalizes users who indiscriminately promote content without regard to quality.

Substituting the definitions of T_{U}, A_{U}, T_{C}, and A_{C} in Equations (1) and (2), we see that

$$r_{\mathsf{U}}(u) = \sum_{\langle u,c'\rangle \in \mathsf{E}} t(\langle u, c'\rangle)\, r_{\mathsf{C}}(c') \Big/ A_{\mathsf{U}}(u)^{\alpha}, \tag{3}$$

$$r_{\mathsf{C}}(c) = \sum_{\langle u',c\rangle \in \mathsf{E}} t(\langle u', c\rangle)\, r_{\mathsf{U}}(u'), \tag{4}$$

which manifest the mutually-reinforcing relationships between user and content rankings. Note that when $\alpha = 0$ and $t(\cdot) = 1$, Equations (3) and (4) reduce to the update rules for HITS [17]: CuRank is a generalization of HITS.

Solving for Rankings

To solve Equations (1) and (2) and generate user and content rankings for a given curation graph, we impose an ordering on U and C, form a $|\mathsf{U}| \times |\mathsf{C}|$ *promotion matrix* P where

$$P_{ij} = \begin{cases} t\left(\langle u_i, c_j\rangle\right) & \text{if } \langle u_i, c_j\rangle \in \mathsf{E} \\ 0 & \text{otherwise} \end{cases},$$

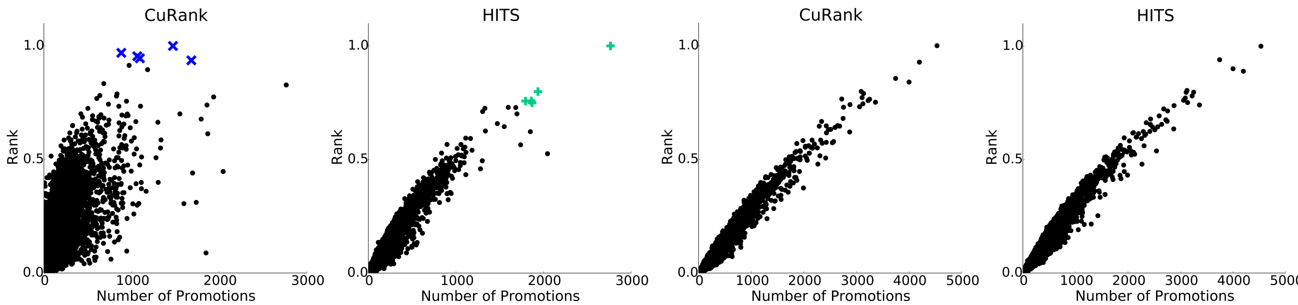
and a *penalized promotion matrix* $\overline{P} = \mathbf{diag}(P\vec{1})^{-\alpha}P$ where $\vec{1} = (1, \ldots, 1)$. Then, the principle eigenvector of $\overline{P}P^{\top}$ gives the values for $r_{\mathsf{U}}(\cdot)$, and the principle eigenvector of $P^{\top}\overline{P}$ gives values for $r_{\mathsf{C}}(\cdot)$. Both eigenvectors can be calculated efficiently using standard numerical tools [17, 11].

A SIMPLE EXAMPLE

To illustrate the properties of CuRank, we generate a random curation graph with 16 users and 21 pieces of content, and examine the gamut of possible taste and activity scores for a new user entering the network under three ranking algorithms: number of promotions, HITS, and CuRank (with $\alpha = 0.65$ and $t(\cdot) = 1$).

The shape of these gamuts, shown in Figure 3, demonstrates the tradeoff between taste and activity: a user who promotes everything in the network cannot have the most discriminating taste, and a user who promotes only important content cannot be the most active. The blue-green gradient within each gamut denotes the ranking for a particular taste/activity pair: green regions signify high curatorial ranks, and blue regions signify low ones.

The upper edge of each gamut traces the path of an "optimal" user who joins the network and promotes content in order from highest to lowest importance. The lower edge shows the path of a "least optimal" user who promotes content in the opposite order. The two paths converge at the point of maximal activity, where both users have curated all the available content.



(a) Users **(b)** Content

Figure 4: User (left) and content (right) rankings produced by CuRank and HITS on the GitHub dataset. Observe that the user rankings produced by HITS are more correlated with activity than those produced by CuRank, and on average top CuRank users (shown in blue) make fewer promotions than top HITS users (shown in green). The content rankings produced by HITS and CuRank largely agree.

Each algorithm assign higher rank to different parts of the gamut. While the rankings produced by HITS are almost perfectly correlated with activity, CuRank assigns the highest rank to the middle of the upper edge, where a user balances between activity and taste.

Another visible effect of the rankings produced by CuRank is that a new user can attain high rank with a relatively small number of promotions, provided they demonstrate exceptional taste. This ability to "break into the winner's circle" is an attractive property for social networks attempting to combat entrenchment and expand their user base.

RESULTS

To evaluate CuRank, we sampled datasets from two popular real-world social networks: GitHub [1] — a social network for sharing code — and Vine [2] — a social network for sharing short video clips. Using the Forest Fire sampling algorithm [20], we crawled a network of 24,111 users and 17,129 code repositories from GitHub, and a network of 240,990 users and 221,982 video posts from Vine.

Our CuRank implementation is written in Python, and uses the `scipy` and `numpy` libraries built with `OpenBlas` support for fast sparse matrix multiplication. All simulations are run on a MacBook Pro with a 2.5 GHz Intel Core i7 and 16 GB of RAM. For the GitHub dataset, CuRank converges in about 0.16 seconds after 17 iterations, using the eigenvector extraction method described in Kleinberg [17]. For the Vine dataset, CuRank converges in about 3.6 seconds after 25 iterations. Our reference implementation of HITS, which is similarly implemented, converges in 0.18 seconds after 15 iterations on GitHub, and in 4.0 seconds after 25 iterations on Vine.

User and Content Rankings

We compute user and content rankings over the sampled GitHub dataset under HITS and CuRank (with $\alpha = 0.5$, and an exponentially-decaying temporal function). As shown in Figure 4a, the user rankings produced by HITS are almost linearly correlated with activity. In contrast, the CuRank distribution is much wider, affording more discriminative power between users with similar activity profiles. On average, the top CuRank users (shown in blue) make fewer promotions than the top HITS users (shown in green), and have higher taste scores.

While CuRank produces user rankings that are markedly different from those produced by HITS, the content rankings generated by the two algorithms largely agree (Figure 4b). To formalize this observation, we measure the correspondence between the two rankings by computing Kendall's τ_b correlation coefficient

$$\tau_b = \frac{n_c - n_d}{\sqrt{N_1 N_2}} \in [-1, 1],$$

where n_c is the number of concordant pairs, n_d is the number of discordant pairs, and N_1 and N_2 are the numbers of untied pairs in the two rankings, respectively [16].

Under this measure, we find that the CuRank-HITS correlation coefficient for user rankings ($\tau_b = 0.6$) is much lower than the coefficient for content rankings ($\tau_b = 0.91$). This asymmetry arises from CuRank's asymmetric definitions of user and content ranks: while CuRank penalizes users for promoting low-ranked content, content is not penalized for being promoted by low-ranked users. Thus, popular content accrues high rank under CuRank just as it does in HITS.

Constant		**Exponential Decay**	
Top Ten	Average Promoter Number	Top Ten	Average Promoter Number
kingofhawks	329	Doppp	151
avinashkoyyana	363	kingofhawks	329
ArtemKulyabin	539	savage69kr	175
Doppp	151	**goshakkk**	156
chentsulin	539	filipeoliveiraa	192
hemersonvianna	807	avinashkoyyana	363
mkhoeini	304	**rimenes**	196
savage69kr	175	**dmyers**	272
hnq90	384	**nikolay**	202
filipeoliveiraa	192	**myfreeweb**	158

Figure 5: The top ten GitHub users ranked by CuRank under a constant timeliness function that weights all promotions equally (left), and an exponentially decaying function that propagates rank to early promoters of content (right). Users not common to both lists are shown in bold. This second ranking can be used to identify the *tastemakers* in a network.

Tastemakers & Content Creators

One ancillary benefit of CuRank's temporal model is that it can be tuned to promote different kinds of curatorial behaviors. For instance, by choosing a rapidly-decaying timeliness function, CuRank can be made to propagate rank disproportionately to early promoters of content in an effort to identify *tastemakers* who discover and promote content before it becomes popular.

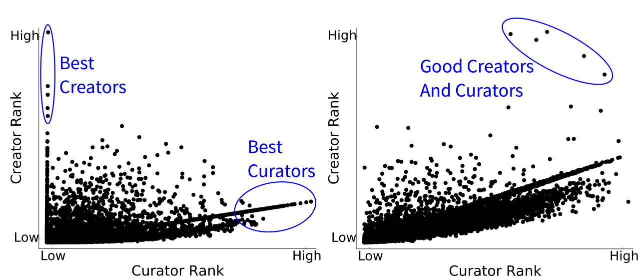
Figure 5 shows the top ten GitHub users ranked both by constant and exponentially decaying temporal functions. For each user, we compute an *average promoter number* to measure the average of the user's promoter positions over all the content she has curated. Ranking with a rapidly-decaying temporal function percolates users with lower average promoter numbers — users whose tastemaking sensibilities are ahead of the network as a whole — to the top of the list.

We can also use CuRank to distinguish between users who *introduce* high-quality content to the network, and those who predominantly promote others' work. By picking a piecewise temporal function that weights content *creation* incommensurately with content *promotion*, we can compute a *creator rank* and use it to identify the ten top creators on GitHub:

facebook jashkenas atom google nnnick
necolas angular jlevy hakimel github

Large organizations such as Facebook and Google naturally dominate this list, since they release code that is used and promoted by many other users.

Comparing this creator rank with the rankings produced by the standard CuRank exponential timeliness model illustrates an interesting dichotomy between the Vine and GitHub social networks. The large organizations that are responsible for great swaths of influential content on GitHub rarely partake in other curatorial activities (Figure 6a). In contrast, several of the most influential Vine users have high creator *and* curator ranks (Figure 6b).



(a) GitHub **(b)** Vine

Figure 6: Creator rank plotted against curator rank for the GitHub (left) and Vine (right) datasets. Top GitHub users are either strong curators *or* strong creators; the Vine network contains some highly-influential users who create original content *and* actively curate others' work.

Rankings Over Time

To understand how CuRank's user rankings evolve over time, we ran an experiment on the GitHub dataset, taking snapshots at yearlong intervals from 2008 to 2015, and computing rankings on each one. For each of the three ranking algorithms — number of promotions, HITS, and CuRank — we compare each year's hundred highest-ranked users with those from the previous year, and compute the percentage of overlap (Figure 7).

The year-to-year overlap averages 52% under CuRank, 69% under HITS, and 72% when ranking by number of promotions. This illustrates another desirable property of CuRank: users cannot rest on their laurels, and must continuously earn their place in the network.

DISCUSSION AND FUTURE WORK

This paper presents, for the first time, an algorithm for ranking curators and content in social networks in a mutually-reinforcing way. CuRank's key contribution is a mathematical model that explicitly accounts for the taste, activity, and timeliness of a user's curatorial activity.

In its current form, CuRank requires that a network's curation graph comprise a single connected component; rank cannot flow between disconnected graphs. While complex social networks usually have a "giant component" that connects most of the nodes in the network [9], future work could borrow ideas from probabilistic ranking algorithms like PageRank which add noise to make any network fully connected.

While CuRank provides a parameter to control the balance between taste and activity in rankings, finding the *ideal* balance for any given network remains an unsolved problem. Learning α in a data-driven way — perhaps by taking a set of partial rankings as input — is another avenue for future work.

The key challenge for modern social networks is to stave off stagnation by attracting new users, retaining existing ones, and driving engagement [24]. We wonder if algorithms like

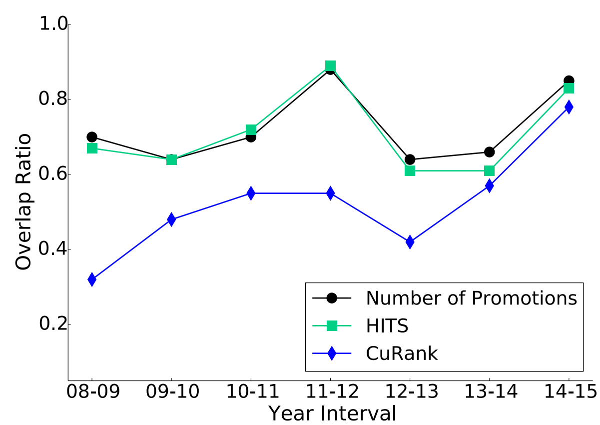


Figure 7: Year-to-year change in the hundred top-ranked GitHub users from 2008 to 2015 under number of promotions, HITS, and CuRank. CuRank exhibits the largest change (lowest overlap ratio) for each of the seven years, suggesting that the algorithm is better able to respond to changing network conditions.

CuRank can help. Of the top 20 Twitter users by number of followers five years ago, 13 of them remain in the top 100 today [18, 26]. Could encouraging more dynamic ranking methodologies and developing mechanisms to reward curation incentivize engagement and new user growth?

ACKNOWLEDGMENTS

We thank Devin Ho, Forrest Huang, Arbishek Harish, Kedan Li, Dingcheng Yue, and Wenzhu Tong for collecting and preprocessing the social network data; Hari Sundaram, Dan Roth, Jiawei Han, the UIUC HCI Group, and the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

1. 2015. GitHub. (2015). https://github.com/.
2. 2015. Vine. (2015). https://vine.co/.
3. Shuo Chang, Vikas Kumar, Eric Gilbert, and Loren G. Terveen. 2014a. Specialization, homophily, and gender in a social curation site: findings from Pinterest. In *Proc. CSCW*. 674–686.
4. Yi Chang, Lei Tang, Yoshiyuki Inagaki, and Yan Liu. 2014b. What is Tumblr: a statistical overview and comparison. *ACM SIGKDD Explorations Newsletter* 16, 1 (2014), 21–29.
5. Jordan Crook. 2015. Spotify introduces 'Found Them First' so you know how cool you are. (2015). http://techcrunch.com/2015/09/09/spotify-introduces-found-them-first-so-you-know-how-cool-you-are/.
6. Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proc. CSCW*. 1277–1286.
7. Ying Ding, Erjia Yan, Arthur Frazho, and James Caverlee. 2009. PageRank for ranking authors in co-citation networks. *Journal of the American Society for Information Science and Technology* 60, 11 (2009), 2229–2243.
8. Chris Dixon. 2015. Come for the tool, stay for the network. (2015). http://cdixon.org/2015/01/31/come-for-the-tool-stay-for-the-network/.
9. David Easley and Jon Kleinberg. 2010. *Networks, crowds, and markets: reasoning about a highly connected world.* Cambridge University Press.
10. Eric Gilbert, Saeideh Bakhshi, Shuo Chang, and Loren Terveen. 2013. I need to try this: a statistical overview of Pinterest. In *Proc. CHI*. 2427–2436.
11. Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations* (3rd ed.). Johns Hopkins University Press.
12. Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. WTF: the Who to Follow service at Twitter. In *Proc. WWW*. 505–514.
13. Catherine Hall and Michael Zarro. 2012. Social curation on the website Pinterest.com. *Proc. ASIS&T* 49, 1 (2012), 1–9.
14. Kyungsik Han, Jin Yea Jang, and Dongwon Lee. 2015. Exploring tag-based like networks. In *Extended Abstracts of CHI*. 1941–1946.
15. Yuchen Jing, Lifang Wu, Xiuzhen Zhang, Dan Wang, and Changwen Chen. 2015. Recommending users to follow based on user taste homophily for content curation social networks. In *Proc. SRS*.
16. M. G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (1938), 81–93.
17. Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM* 46, 5 (1999), 604–632.
18. Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media. In *Proc. WWW*. 591–600.
19. Ronny Lempel and Shlomo Moran. 2000. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks* 33, 1 (2000), 387–401.
20. Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proc. KDD*. 631–636.
21. Ari Levy. 2011. LinkedIn acquires Pulse mobile-news reading tool for $90 million. (2011). http://www.bloomberg.com/news/articles/2013-04-11/linkedin-acquires-pulse-mobile-news-reading-tool-for-90-million.
22. Rinkesh Nagmoti, Ankur Teredesai, and Martine De Cock. 2010. Ranking approaches for microblog search. In *Proc. WI-IAT*. 153–157.
23. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: bringing order to the Web*. Technical Report. Stanford InfoLab.
24. Chris Sacca. 2015. What Twitter can be. (2015). http://lowercasecapital.com/2015/06/03/what-twitter-can-be-2/.
25. Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proc. KDD*. 797–806.
26. Twitter. 2015. Twitter top 100 most followers. (2015). http://twittercounter.com/pages/100.
27. Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twitterrank: finding topic-sensitive influential twitterers. In *Proc. WSDM*. 261–270.
28. Ding Zhou, Sergey Orshanskiy, Hongyuan Zha, and C Lee Giles. 2007. Co-ranking authors and documents in a heterogeneous network. In *Proc. ICDM*. 739–744.