Supplementary Information

Virality Prediction and Community Structure in Social Networks

by Lilian Weng, Filippo Menczer & Yong-Yeol Ahn[1]

Center for Complex Networks and Systems Research,
School of Informatics and Computing,
Indiana University, Bloomingon, IN 47405, USA

Table of Contents

List of Figures

List of Tables

[1]Contact: yyahn@indiana.edu

S1 Dataset

We collected a set of English tweets using the Twitter streaming API during Mar 24 and Apr 25, 2012. We construct three networks—retweet, mention and follower networks—based on these tweets. To make the three networks comparable, we set the group of users in the giant connected component (GCC) of the follower network as the *target group*, and then extract the retweet and mention networks among the users in the target group. Then we extracted the giant connected component from the mention and follower networks, resulting in slightly smaller numbers of users in these networks. Only reciprocal communications are kept in the three networks, as bi-directional communications reflect more stable and reliable social connections. All the networks remain unweighted for community identification, to focus purely on the connection structure.

To demonstrate the results are robust across different types of community structure, we apply disjoint, *InfoMap* [1], and overlapping, *Link Clustering* [2], community detection methods to all three networks. Communities with less than three nodes are removed. Basic statistics of the three networks and communities are reported in Table S1.

S2 Meme concentration in communities

The set of communities to which user u is assigned is $C_u = \{c \mid u \in c \wedge c \in C\} \subseteq C$. For an edge $(u,v) \in E$, $u,v \in V$ are two users connected by the edge. The set of intra-community edges is defined as $E_{\circlearrowright} = \{(u,v) \mid C_u \cap C_v \neq \varnothing\}$ and inter-community edges belong to $E_{\curvearrowright} = \{(u,v) \mid C_u \cap C_v = \varnothing\}$. Similarly, sets of intra- and inter-community edges can be defined for a single community c: $E^c_{\circlearrowright} = \{(u,v) \mid (u,v) \in E_{\circlearrowright} \wedge c_u = c_v = c\}$ and $E^c_{\curvearrowright} = \{(u,v) \mid (u,v) \in E_{\curvearrowright} \wedge (c_u = c \vee c_v = c)\}$.

S2.1 Edge weight

We define the weight of an edge (u,v), $w(u,v)$, by the frequency of u retweeting ("RT") or mentioning ("@") v, noted as $w^{\mathrm{RT}}(u,v)$ or $w^{@}(u,v)$. For a community c, the average *edge weights* of intra- and inter-community links are defined as:

$$\langle w_{\circlearrowright} \rangle_c = \frac{1}{|E^c_{\circlearrowright}|} \sum_{(u,v)\in E^c_{\circlearrowright}} w(u,v), \quad \langle w_{\curvearrowright} \rangle_c = \frac{1}{|E^c_{\curvearrowright}|} \sum_{(u,v)\in E^c_{\curvearrowright}} w(u,v).$$

A common intuition that random walks on a graph tend to get trapped inside densely connected components has been employed in many community detection method [3, 1]. Even if a community does not consist of homophilous people or stronger links, the spreading of a meme can circulate more within communities, driven by dense internal connections. To estimate the structural trapping effect of communities, consider a random walker traversing the graph. The basic assumption is that if information spreads randomly through links and there is an infinite number of spreading events (treating every node and link equally), the probability that a given link is used in the transmission of information will approach the probability that a random walker traverses the link. The probability of the random walker moving from a node u to another connected node v given the walker is at u is $p_{u \to v} = 1/k(u)$, where $k(u)$ is the degree of u. We can construct a transition matrix P where $p_{u \to v} = 1/k(u)$ if u and v are connected, and $p_{u \to v} = 0$ otherwise. The stationary probability of the walker stopping at a node u is the element π_u of a vector π such that $P^T \pi = \pi$. It can be shown that $\pi_u = k(u)/\sum_v k(v)$ [4]. The expected amount of communication carried by edge (u,v), considering structural trapping but without any homophily or social reinforcement effects, can be computed by the probability $w^{\text{rw}}(u,v)$ of the random walker traveling through the edge:

$$\begin{aligned} w^{\text{rw}}(u,v) &= \pi_u p_{u \to v} + \pi_v p_{v \to u} \\ &= \frac{k(u)}{\sum_m k(m)} \frac{1}{k(u)} + \frac{k(v)}{\sum_m k(m)} \frac{1}{k(v)} \\ &= \frac{2}{\sum_m k(m)} \sim \text{const.} \end{aligned}$$

In other words, a random walker, or a random spreading event, will traverse each edge with the same probability. In contrast to the constant weights expected across every edge, we observe clear differences between intra- and inter-community links cross all cases of different networks and communities, as demonstrated in Fig. S1. The differences between intra- and inter-community measures are significant (Mann-Whitney U test [5], $p \ll 0.001$). Note that the community structure is detected from the unweighted network, capturing only topological property of the networks.

S2.2 User community focus

For each individual u in a community, we compare the average fraction of activity (retweets or mentions) with each neighbor in the same community, $f_{\circlearrowright}$, and the average fraction of activity with each neighbor in a different community, $f_{\curvearrowright}$:

$$f_{\circlearrowright}(u) = \frac{\frac{1}{k_{\circlearrowright}(u)} \sum_{(u,v) \in E_{\circlearrowright}} w(u,v)}{\frac{1}{k(u)} \sum_{(u,v) \in E} w(u,v)}, \quad f_{\curvearrowright}(u) = \frac{\frac{1}{k_{\curvearrowright}(u)} \sum_{(u,v) \in E_{\curvearrowright}} w(u,v)}{\frac{1}{k(u)} \sum_{(u,v) \in E} w(u,v)}$$

where $k_{\circlearrowright}(u)$ and $k_{\curvearrowright}(u)$ are the numbers of u's intra- and inter-community links:

$$\begin{aligned} k_{\circlearrowright}(u) &= |\{v \mid (u,v) \in E_{\circlearrowright}\}| \\ k_{\curvearrowright}(u) &= |\{v \mid (u,v) \in E_{\curvearrowright}\}| \\ k(u) &= k_{\circlearrowright}(u) + k_{\curvearrowright}(u). \end{aligned}$$

The ratios $f_{\circlearrowright}$ and $f_{\curvearrowright}$ characterize how attention is directed toward a person within the same community versus a person in another community. Using the random walk analogy, the user community focus represents the probability of a random walker from a node traveling through each of its links. By definition, the random walker does not distinguish links, and thus we always have $f_{\circlearrowright}^{\text{rw}} = f_{\curvearrowright}^{\text{rw}} = 1$. Fig. S2 shows that $f_{\circlearrowright} > 1 > f_{\curvearrowright}$ on average, indicating that people communicate more with neighbors in the same community. The results are robust across different activity measures and communities, and the differences are statistically significant (Mann-Whitney U test, $p \ll 0.001$).

S2.3 Quantifying meme concentration

S2.3.1 Sampling, simulation, and hashtag filtering

Our data is collected through the Twitter streaming API, which provides about 10% of all public tweets. To simulate this sampling effect, we run each simulation until we have 10 times more tweets than the empirical numbers. Then, we select 10% of the tweets at random. Every simulation is repeated 100 times and the 10%-sampling is repeated 10 times on each simulation output. Thus, the average values of the measures from our toy models are computed across 100×10 samples. We only focus on *new* memes that emerged during our observation time window. New memes are defined as those with fewer than X tweets during the previous month (March 2012). We set $X = 20$ in the following analyses. The robustness of our results with respect to hashtag filtering criteria X is tested against the prediction outcomes in Sec. S3.3.

We design a few quantities to estimate the distinguish simple and complex contagions as follows. All the measurements are computed based on the early stage of meme spreading, precisely the information of the first n tweets. We set $n = 50$ in the paper.

S2.3.2 Dominance

For each hashtag (meme) h, $t_c(h)$ tweets are generated by $u_c(h)$ adopters in community $c \in C$. There are $T(h)$ tweets and $U(h)$ adopters in total containing h in all the detected communities.

The fraction of tweets with hashtag h in community c is $r_c(h) = t_c(h)/T(h)$. The *dominant community* that produces most messages with h is $c^t(h) = \arg\max_{c \in C} r_c(h)$. The *usage dominance* of h is $r(h) = r_{c^t(h)}(h)$, quantifying the contribution to the hashtag usage from the dominant community. Similar measures can be defined with regard to adopters. The fraction of adopters in a single community is $g_c(h) = u_c(h)/U(h)$. The dominant community $c^u(h) = \arg\max_{c \in C} g_c(h)$ is the one with most adopters of hashtag h, and then we measure *adoption dominance* for h as $g(h) = g_{c^u(h)}(h)$.

The simulations of the baseline models for each meme stop when the equal number of tweets or adopters are produced. A user can adopt the hashtag h multiple times, as a user is able to generate multiple tweets with the same hashtag on Twitter. The usage dominance $r(h)$ produced by the baseline models are labelled as $r_{M_1}(h)$, $r_{M_2}(h)$, $r_{M_3}(h)$ and $r_{M_4}(h)$, respectively. Similarly, for adoption dominance, there are $g_{M_1}(h)$, $g_{M_2}(h)$, $g_{M_3}(h)$ and $g_{M_4}(h)$. The relative usage dominances, r/r_{M_1} and $g(h)/g_{M_1}(h)$, reflect the strength of meme concentration beyond random sampling.

According to Fig. S3, the empirical data displays the strongest concentration for memes of intermediate popularities, independent of types of networks and communities. Interestingly, viral (popular) memes show similar patterns as the model of simple contagion (M_2) with much weaker concentration than models of complex contagions (M_3 and M_4).

S2.3.3 Entropy

Rather than only focusing on the dominant community, the *usage entropy* describes how all the tweets of a meme h are allocated across communities, computed as $H^t(h) = -\sum_{c\in C} r_c(h)\log r_c(h)$. Similarly, the diversity of the distribution of meme adopters is quantified by the *adoption entropy* $H^u(h) = -\sum_{c\in C} g_c(h)\log g_c(h)$.

Again, we compare the concentration of usage and user engagement with random sampling (M_1) by computing the relative usage and adoption entropies, $H^t/H^t_{M_1}$ and $H^u(h)/H^u_{M_1}(h)$. In Fig. S4 we observe similar patterns on disjoint communities to dominance measures. In overlapping communities, a tweet or user can be counted multiple times when locating in the overlapping portions of the networks, which makes the measurement hard to interpret.

S2.4 Social Reinforcement

For a meme (hashtag) h, we count the number of exposures that each adopter has experienced before the adoption and compute the *average exposures* across all adopters, representing the strength of social reinforcement on h, labelled as $N(h)$. Exposures from neighbors can be measured in terms of tweets $N^t(h)$ or adopted users $N^u(h)$. Using average exposures, we can explicitly gauge the social reinforcement effect on meme adopters and thus to distinguish simple and complex contagions.

The patterns that unpopular memes are more sensitive to social reinforcement and popular memes are less so are consistent irrespective of the types of networks and communities, suggesting that unpopular memes are complex contagions while popular ones behave like simple contagions (see Fig. S5).

S3 Prediction

S3.1 Features

Our prediction model uses several features, as listed in the main paper. We present more explanations of *fraction of intra-community user interactions* as below.

Let us define $I(h)$ as the total number of pair-wise user interactions about a hashtag h, where the interactions can be either retweets or mentions, represented as $I^{\mathrm{RT}}(h)$ and $I^{@}(h)$, respectively. Then the *fraction of intra-community user interactions* is:

$$
\begin{aligned}
\frac{I^{\mathrm{RT}}_{\circlearrowleft}(h)}{I^{\mathrm{RT}}(h)} &= \frac{|\{(u,v)|\ u \text{ retweets } v \text{ about } h, C_u \cap C_v \neq \varnothing\}|}{|\{(u,v)|u \text{ retweets } v \text{ about } h\}|} \\
\frac{I^{@}_{\circlearrowleft}(h)}{I^{@}(h)} &= \frac{|\{(u,v)|u \text{ mentions } v \text{ about } h, C_u \cap C_v \neq \varnothing\}|}{|\{(u,v)|\ u \text{ mentions } v \text{ about } h\}|}.
\end{aligned}
$$

A high fraction of intra-community user interactions implies a stronger effect of homophily and social reinforcement on the meme spreading. Fig. S6 suggests that a meme with high $I_{\circlearrowleft}/I$ may have difficulty in spreading out to other communities. Interestingly, highly viral memes ($T > 10^3$) also tend to have a lot of intra-community user interactions. This suggests that viral memes circulate well inside communities while having small dominance and high entropy.

S3.2 Experiments

We predict the meme virality based on the early spreading patterns in terms of community structure. Only hashtags that started during the first two weeks in our dataset are chosen in the experiment, so that our dataset can recover the starting stage and track the meme spreading during a whole month. Prediction features are computed for each selected hashtag based on its earliest n tweets. Thus hashtags with less than n tweets are also filtered out, as insufficient early-stage information can be gathered.

The prediction task is a binary meme classification: *predict whether a meme will go viral or not.* A meme is deemed *viral* if it obtains more tweets than $\theta_T\%$ memes or more adopters than $\theta_U\%$ memes.

We use the random forest algorithm, an ensemble classifier that constructs 500 decision trees, each with 4 random features [6]. For training and testing, we employ 10-fold cross validation. We compare our results with two baselines: (i) *random guess* marks n_{viral} random memes as viral, where n_{viral} is the actual number of viral memes in the empirical data; (ii) *community-blind prediction* adopts the same learning model but without community-related features. Thus community-blind prediction only incorporates two basic features, the numbers of early adopters and of uninfected neighbors of early adopters. Precision and recall are measured for evaluation: Precision $= TP/(TP+FP)$ and Recall $= TP/(TP+FN)$, where TP is the number of true positives (correctly predicted viral memes), FP is the number of false positives (false alarms), and FN is the number false negatives (missed viral memes). We experiment with $\theta = 70, 80, 90$ and $n = 25, 50, 100$. We report results with meme popularity measured by number of tweets (Table S2) and adopters (Table S3). Our predictive model achieves the best precision and recall in *all cases*.

S3.3 Robustness of hashtag filtering

To demonstrate that the hashtag filtering method (see Sec. S2.3.1) does not bring bias into our prediction results by considering memes that were already popular in the previous month, we tested the predictive model with different filtering criteria $X = 0, 5, 10, 20, 40, 60, 80, 100$. The prediction performances do not vary much; the community-based prediction still yields the best results (see Fig. S7 and Fig. S8).

References

[1] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. USA*, 105(4):1118–1123, January 2008.

[2] Yong-Yeol Ahn, James Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[3] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *Computer and Information Sciences-ISCIS*, 3733/2005:284–293, 2005.

[4] L Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdos Is Eighty*, 2(1):1–46, 1993.

[5] Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

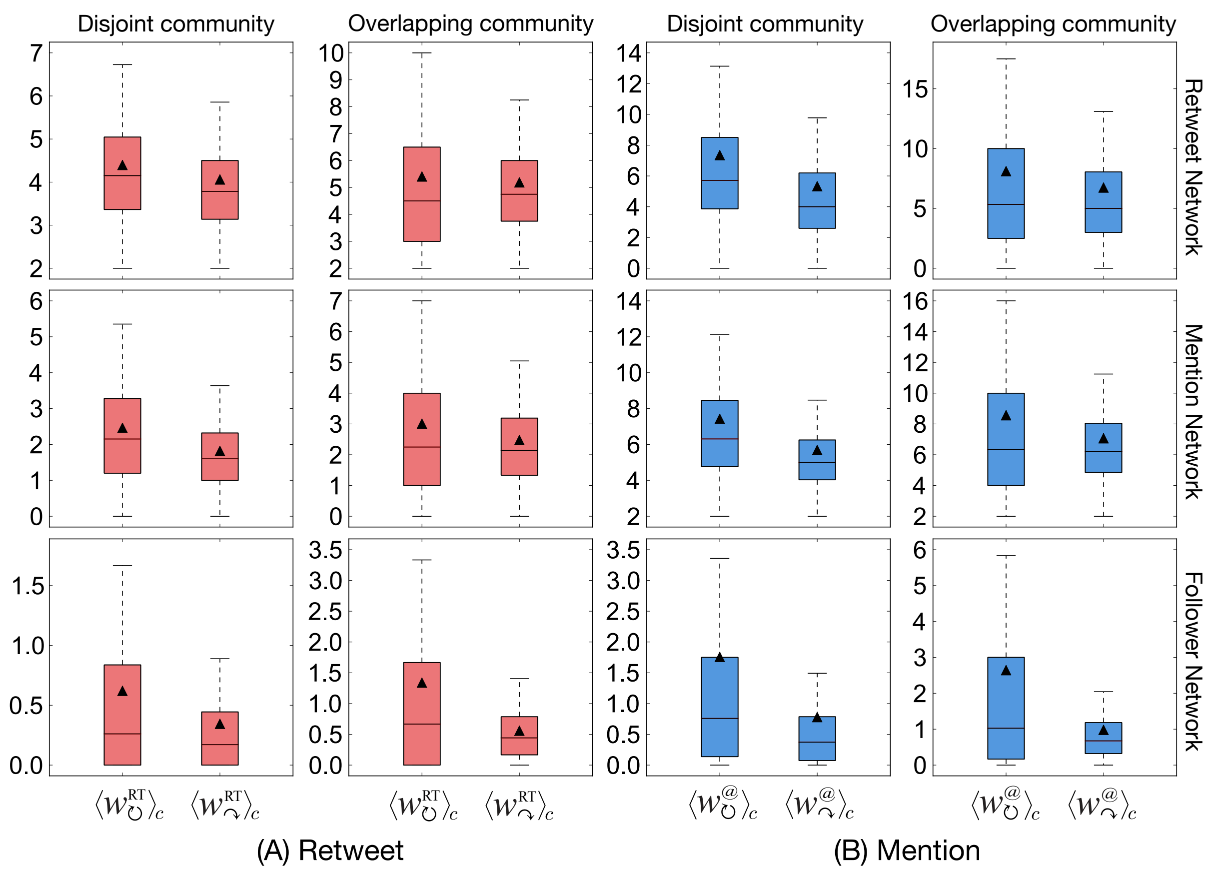

Figure S1: Boxplots of the average amount of user activities on inter- and inter-community edges for each community on three networks. (A) Measure the activity by the number of retweets, $\langle w^{\mathrm{RT}}_{\curvearrowright} \rangle_c$ and $\langle w^{\mathrm{RT}}_{\circlearrowright} \rangle_c$. (B) Measure the activity by the number of mentions, $\langle w^{@}_{\curvearrowright} \rangle_c$ and $\langle w^{@}_{\circlearrowright} \rangle_c$. In all box plots, boxes cover 50% confidence intervals, and whiskers cover 95% confidence intervals. The line and triangle in a box represent the median and mean, respectively.

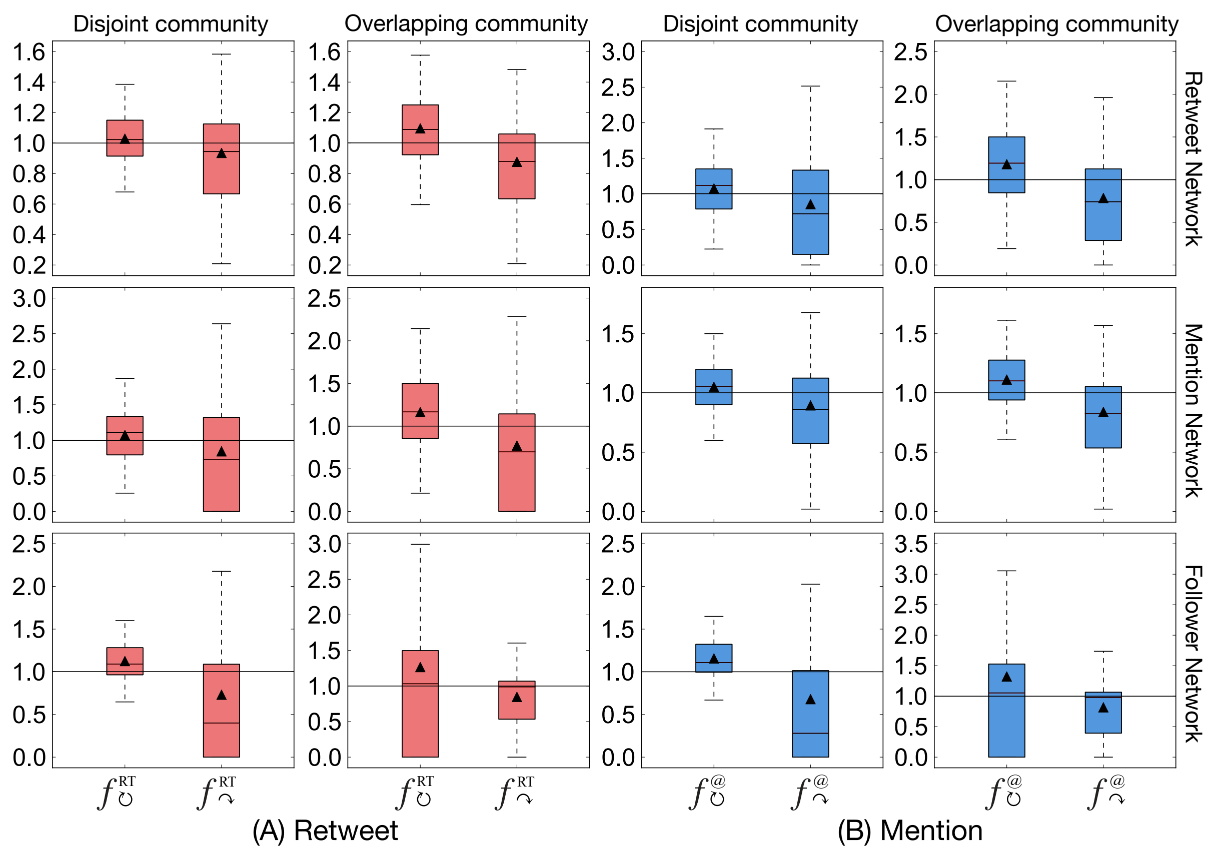

Figure S2: Boxplots of user focus on each inter- and inter-community link. (A) Focus measured by retweet activity, $f^{\mathrm{RT}}_{\curvearrowright}$ and $f^{\mathrm{RT}}_{\circlearrowleft}$. (B) Focus measured by mention activity, $f^{@}_{\curvearrowright}$ and $f^{@}_{\circlearrowleft}$.

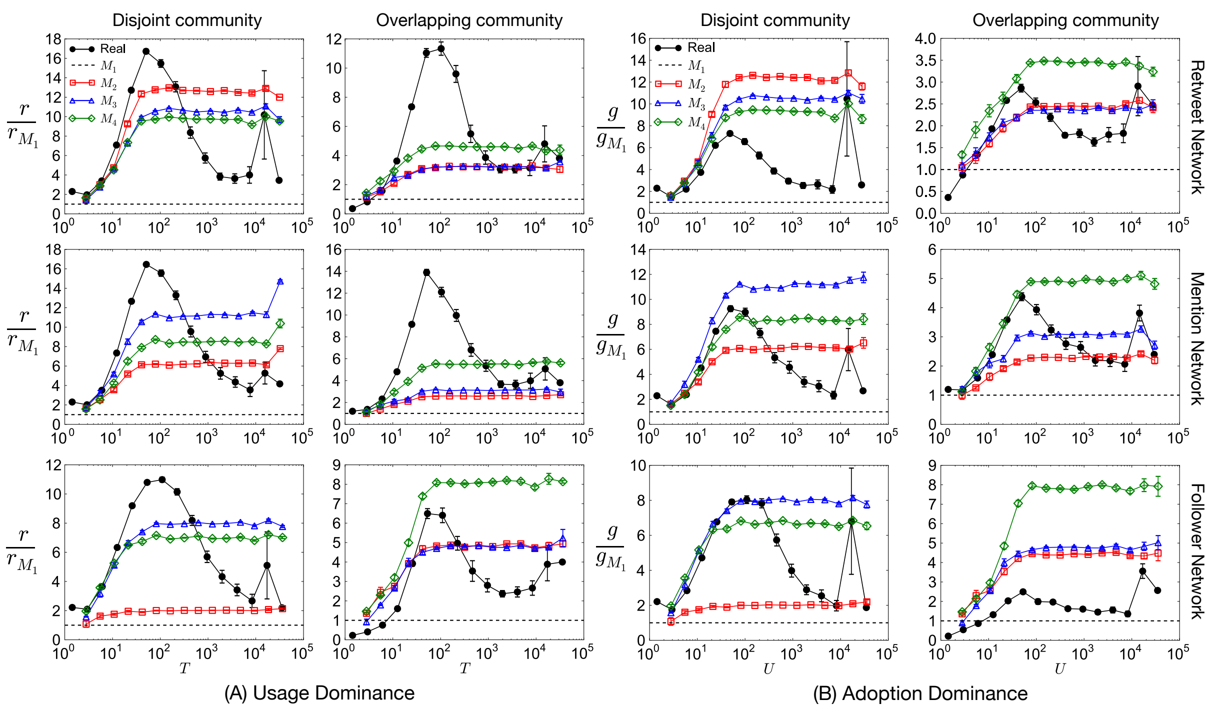

Figure S3: (A) The relative usage dominance, as the ratio of usage dominance to the same measure in M_1, $r(h)/r_{M_1}(h)$, plotted versus popularity $T(h)$. (B) The relative adoption dominance, $g(h)/g_{M_1}(h)$ versus $U(h)$. The measures are computed based on first 50 tweets.

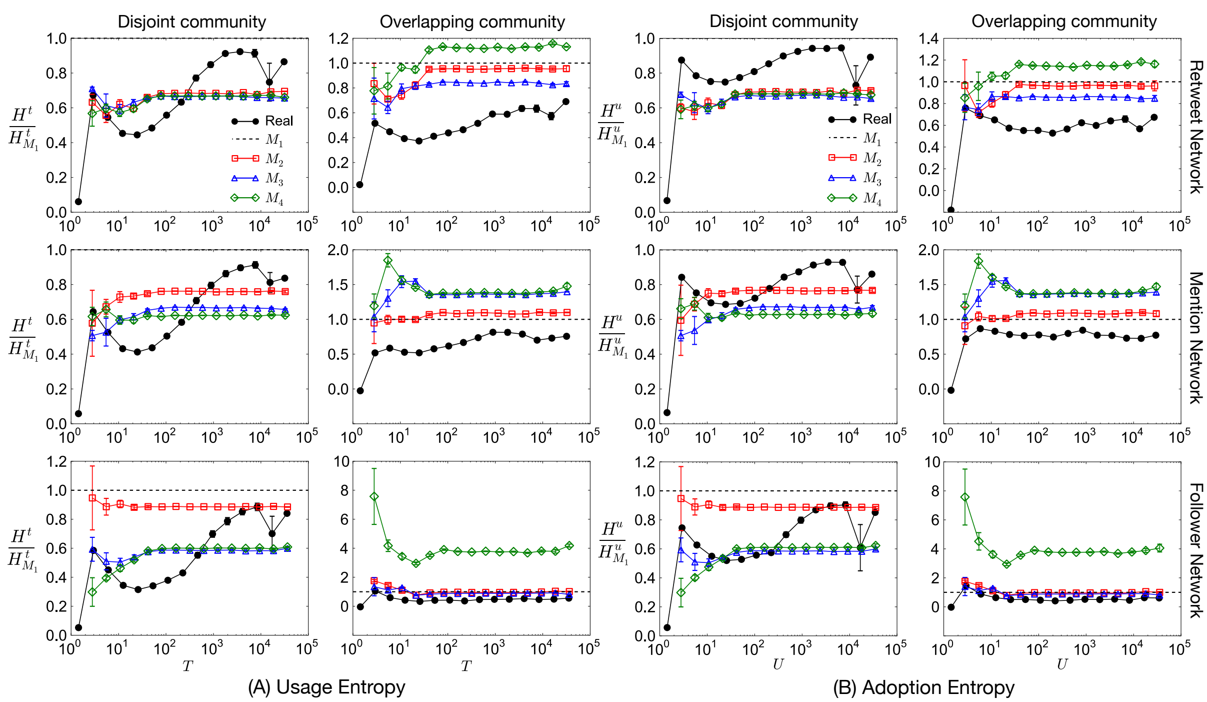

Figure S4: (A) The relative usage entropy, measured as the ratio of usage entropy to the same measure in $H^t(h)/H^t_{M_1}(h)$, plotted versus popularity $T(h)$. (B) The relative adoption entropy, $H^u(h)/H^u_{M_1}(h)$ versus $U(h)$. The measures are computed based on first 50 tweets.

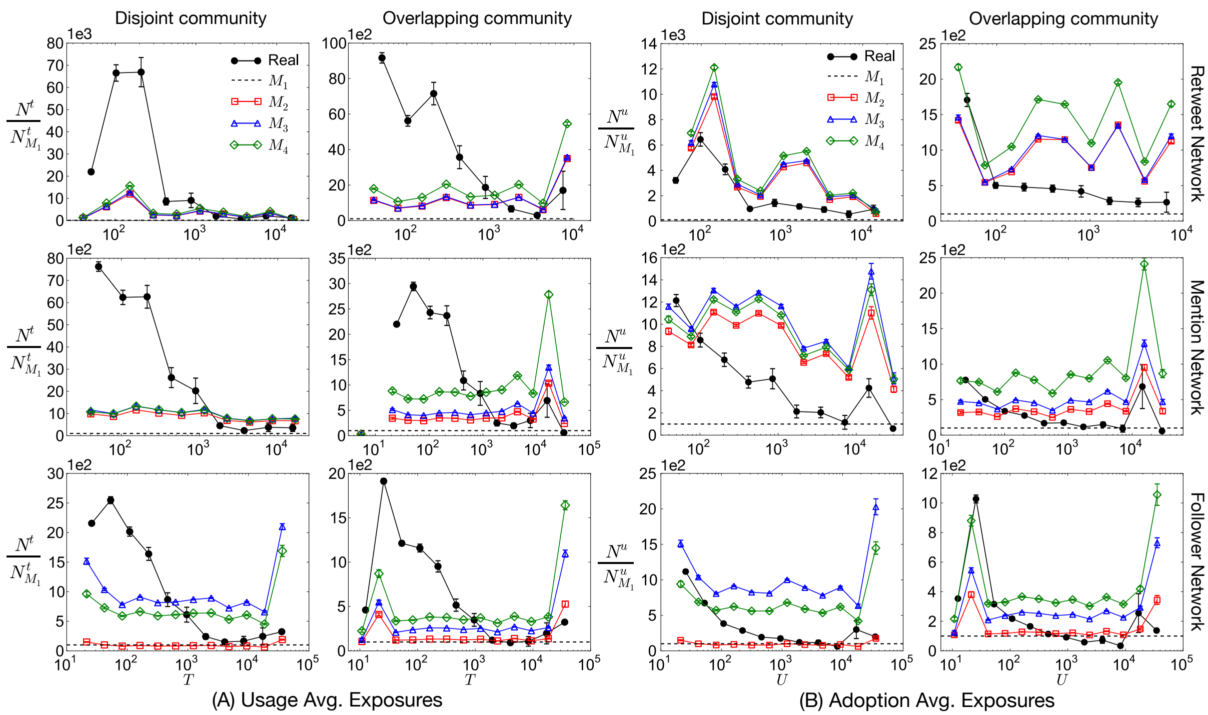

Figure S5: (A) The relative usage average exposure, measured as the ratio of usage average exposure to the same measure in M_1, $N^t(h)/N^t_{M_1}(h)$, plotted versus popularity $T(h)$. (B) The relative adoption average exposure, $N^u(h)/N^u_{M_1}(h)$ versus $U(h)$. The measures are computed based on first 50 tweets.

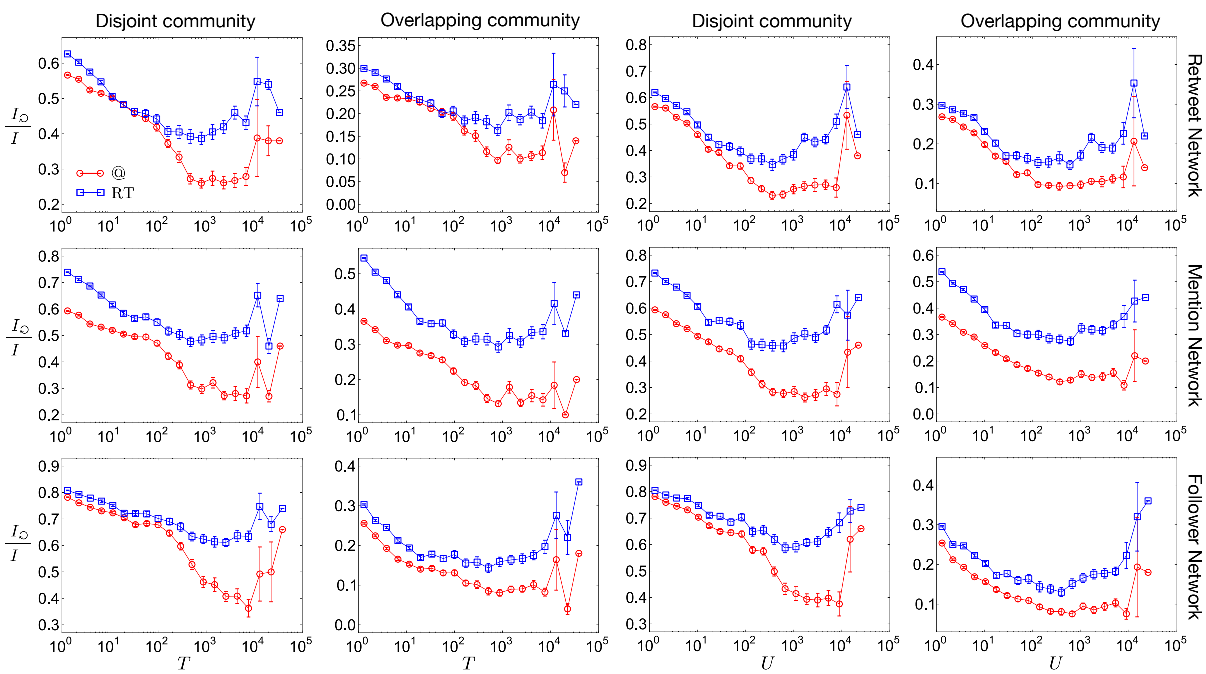

Figure S6: Fraction of intra-community user interactions by retweeting or mentioning versus meme popularity measured by T or U.

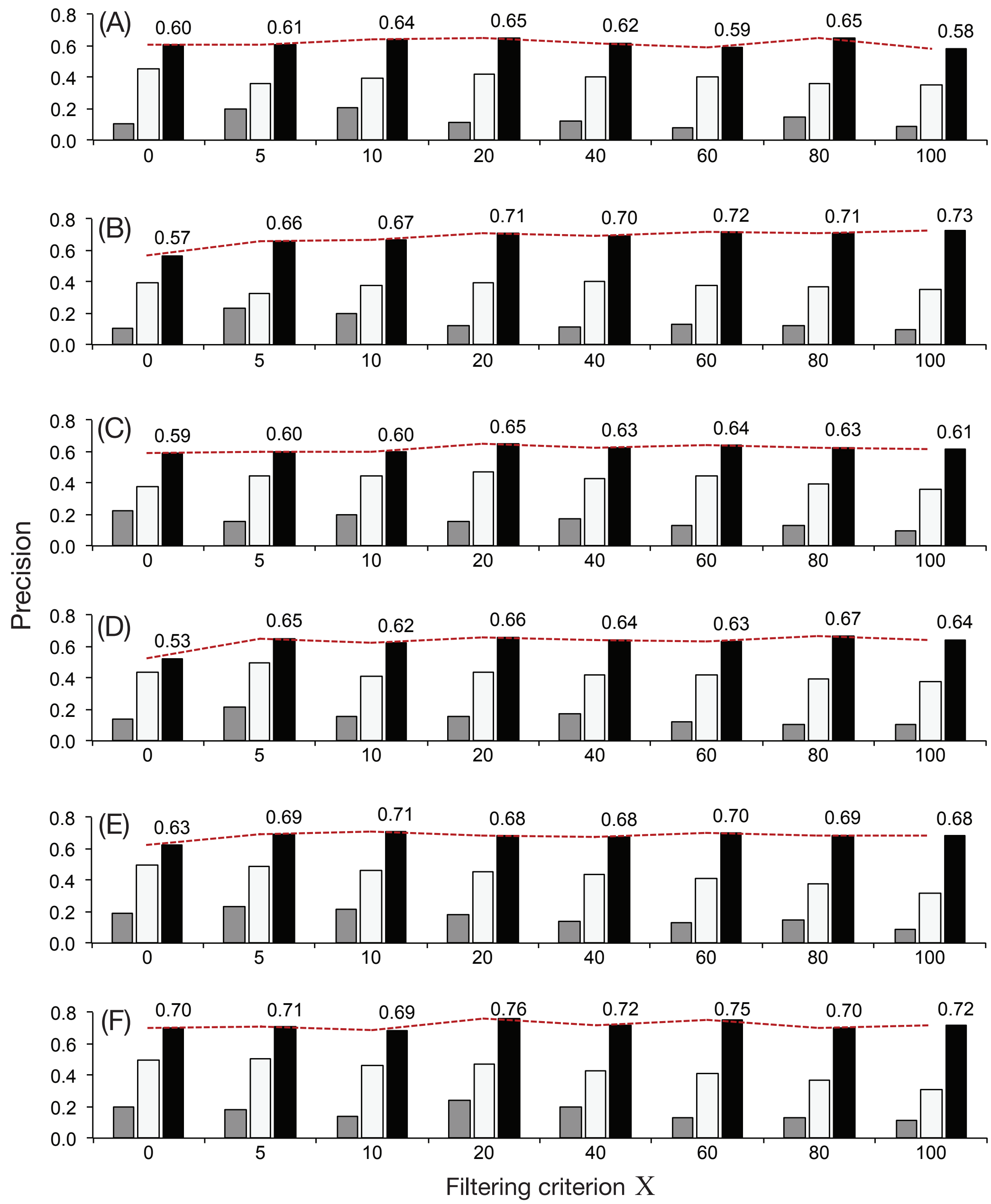

Figure S7: We experiment with $X = 0, 5, 10, 20, 40, 60, 80, 100$ as filtering criteria for fresh memes. We then run the community-based prediction method on these selected fresh memes. The figure plots the precisions of prediction results on different network and community settings: (A) Disjoint communities on retweet network. (B) Overlapping communities on retweet network. (C) Disjoint communities on mention network. (D) Overlapping communities on mention network. (E) Disjoint communities on follower network. (F) Overlapping communities on follower network.

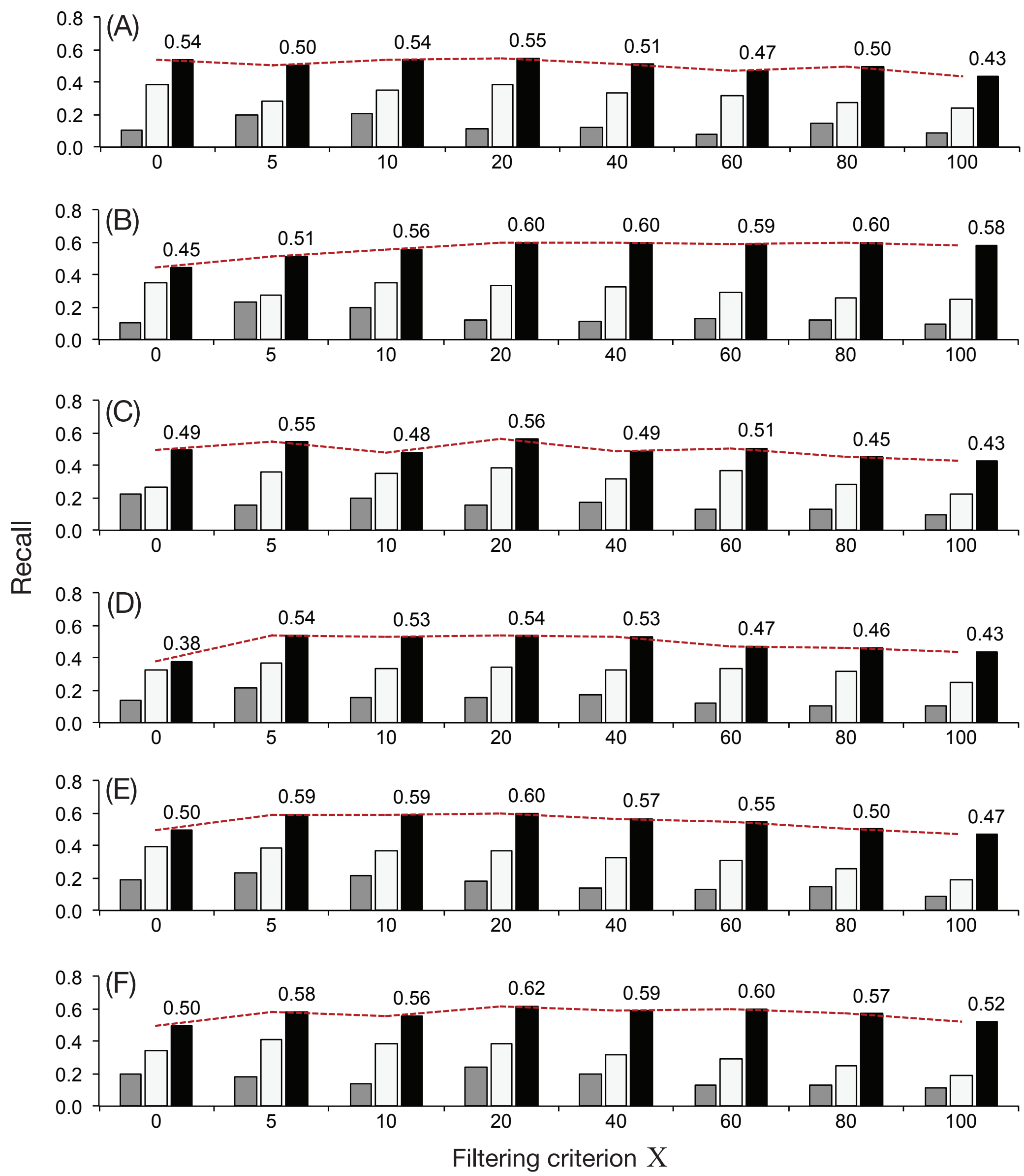

Figure S8: We experiment with $X = 0, 5, 10, 20, 40, 60, 80, 100$ as filtering criteria for fresh hashtags. The figure plots the recalls of prediction results on different network and community settings. Subfigure labels are same as in Fig. S7

Table S1: Network statistics of retweet, mention, and follower networks used in the study. Node coverage measures the proportion of nodes that belong to communities with at least three nodes.

		Network types		
		Retweet	Mention	Follower
	Number of nodes	300,197	374,829	595,460
	Number of edges	598,487	1,048,818	14,273,311
	Avg. clustering coefficient	0.0902	0.1284	0.1972
InfoMap	Number of communities	14,144	14,222	6,360
	Node coverage	99.86%	99.72%	99.72%
LinkComm	Number of communities	57,317	97,198	321,774
	Node coverage	48.42%	67.23%	47.62%

Table S2: Prediction results where the popularity is measured by the number of tweets (T). B_1 is the random guess baseline, B_2 is the community-blind baseline, and CVP is our community-based virality prediction algorithm.

θ_T	Network	Community	n	#Tags	Precision			Recall		
					B_1	B_2	CVP	B_1	B_2	CVP
70	retweet	Disjoint	25	2061	0.29	0.33	0.59	0.29	0.28	0.42
			50	907	0.31	0.46	0.67	0.31	0.45	0.53
			100	469	0.27	0.49	0.68	0.27	0.45	0.53
		Overlapping	25	2061	0.28	0.33	0.64	0.28	0.29	0.46
			50	907	0.3	0.45	0.67	0.3	0.44	0.5
			100	469	0.26	0.47	0.71	0.26	0.42	0.59
	mention	Disjoint	25	2384	0.29	0.37	0.61	0.29	0.31	0.46
			50	1037	0.29	0.34	0.65	0.29	0.3	0.49
			100	529	0.27	0.4	0.63	0.27	0.35	0.48
		Overlapping	25	2384	0.28	0.39	0.63	0.28	0.33	0.42
			50	1037	0.29	0.34	0.64	0.29	0.28	0.53
			100	529	0.26	0.37	0.65	0.26	0.34	0.5
	follower	Disjoint	25	2774	0.27	0.31	0.61	0.27	0.27	0.47
			50	1178	0.29	0.39	0.65	0.29	0.33	0.51
			100	587	0.28	0.39	0.67	0.28	0.36	0.53
		Overlapping	25	2774	0.28	0.33	0.56	0.28	0.29	0.41
			50	1178	0.3	0.41	0.69	0.3	0.35	0.48
			100	587	0.29	0.42	0.66	0.29	0.36	0.55
80	retweet	Disjoint	25	2061	0.18	0.28	0.57	0.18	0.21	0.35
			50	907	0.18	0.36	0.63	0.18	0.34	0.43
			100	469	0.18	0.34	0.65	0.18	0.25	0.42
		Overlapping	25	2061	0.19	0.28	0.65	0.19	0.21	0.43
			50	907	0.18	0.34	0.7	0.18	0.3	0.53
			100	469	0.18	0.33	0.53	0.18	0.27	0.4
	mention	Disjoint	25	2384	0.18	0.25	0.6	0.18	0.2	0.4
			50	1037	0.18	0.27	0.63	0.18	0.23	0.47
			100	529	0.18	0.29	0.58	0.18	0.22	0.35
		Overlapping	25	2384	0.18	0.23	0.64	0.18	0.18	0.39
			50	1037	0.19	0.26	0.65	0.19	0.21	0.45
			100	529	0.17	0.27	0.6	0.17	0.2	0.43
	follower	Disjoint	25	2774	0.18	0.28	0.61	0.18	0.21	0.41
			50	1178	0.19	0.33	0.63	0.19	0.28	0.47
			100	587	0.17	0.19	0.61	0.17	0.15	0.38
		Overlapping	25	2774	0.18	0.26	0.57	0.18	0.2	0.36
			50	1178	0.2	0.35	0.69	0.2	0.27	0.48
			100	587	0.19	0.24	0.59	0.19	0.16	0.46
90	retweet	Disjoint	25	2061	0.08	0.19	0.58	0.08	0.14	0.33
			50	907	0.08	0.25	0.69	0.08	0.2	0.37
			100	469	0.07	0.21	0.57	0.07	0.16	0.32
		Overlapping	25	2061	0.09	0.17	0.65	0.09	0.13	0.38
			50	907	0.07	0.2	0.47	0.07	0.15	0.33
			100	469	0.1	0.3	0.44	0.1	0.24	0.18
	mention	Disjoint	25	2384	0.09	0.15	0.54	0.09	0.12	0.3
			50	1037	0.1	0.22	0.46	0.1	0.16	0.2
			100	529	0.07	0.06	0.48	0.07	0.04	0.28
		Overlapping	25	2384	0.09	0.15	0.66	0.09	0.1	0.36
			50	1037	0.09	0.22	0.67	0.09	0.16	0.32
			100	529	0.1	0.14	0.35	0.1	0.09	0.2
	follower	Disjoint	25	2774	0.08	0.21	0.63	0.08	0.14	0.37
			50	1178	0.11	0.17	0.66	0.11	0.11	0.36
			100	587	0.07	0.06	0.5	0.07	0.04	0.15
		Overlapping	25	2774	0.09	0.19	0.6	0.09	0.13	0.38
			50	1178	0.09	0.2	0.5	0.09	0.15	0.37
			100	587	0.09	0.1	0.25	0.09	0.05	0.13

Table S3: Prediction results where the popularity is measured by the number of adopters (U). B_1 is the random guess baseline, B_2 is the community-blind baseline, and CVP is our community-based virality prediction algorithm.

θ_U	Network	Community	n	#Tags	Precision			Recall		
					B_1	B_2	CVP	B_1	B_2	CVP
70	retweet	Disjoint	25	2061	0.25	0.5	0.68	0.25	0.48	0.57
			50	907	0.25	0.52	0.69	0.25	0.55	0.68
			100	469	0.23	0.56	0.71	0.23	0.54	0.64
		Overlapping	25	2061	0.26	0.51	0.69	0.26	0.48	0.57
			50	907	0.25	0.53	0.73	0.25	0.55	0.65
			100	469	0.22	0.56	0.77	0.22	0.51	0.75
	mention	Disjoint	25	2384	0.24	0.51	0.69	0.24	0.49	0.59
			50	1037	0.27	0.54	0.69	0.27	0.54	0.69
			100	529	0.26	0.44	0.68	0.26	0.45	0.64
		Overlapping	25	2384	0.25	0.51	0.67	0.25	0.5	0.57
			50	1037	0.26	0.5	0.72	0.26	0.49	0.67
			100	529	0.25	0.46	0.67	0.25	0.45	0.63
	follower	Disjoint	25	2774	0.25	0.51	0.67	0.25	0.5	0.58
			50	1178	0.27	0.52	0.72	0.27	0.5	0.68
			100	587	0.26	0.45	0.69	0.26	0.42	0.64
		Overlapping	25	2774	0.24	0.5	0.65	0.24	0.48	0.55
			50	1178	0.27	0.5	0.75	0.27	0.5	0.68
			100	587	0.25	0.46	0.73	0.25	0.43	0.66
80	retweet	Disjoint	25	2061	0.16	0.36	0.68	0.16	0.32	0.54
			50	907	0.18	0.41	0.63	0.18	0.39	0.59
			100	469	0.18	0.46	0.67	0.18	0.44	0.49
		Overlapping	25	2061	0.16	0.37	0.69	0.16	0.33	0.58
			50	907	0.17	0.4	0.7	0.17	0.41	0.63
			100	469	0.17	0.43	0.65	0.17	0.38	0.58
	mention	Disjoint	25	2384	0.16	0.35	0.68	0.16	0.32	0.56
			50	1037	0.16	0.35	0.67	0.16	0.3	0.6
			100	529	0.16	0.31	0.66	0.16	0.26	0.53
		Overlapping	25	2384	0.16	0.36	0.66	0.16	0.32	0.53
			50	1037	0.18	0.39	0.68	0.18	0.32	0.57
			100	529	0.17	0.3	0.6	0.17	0.27	0.48
	follower	Disjoint	25	2774	0.16	0.36	0.68	0.16	0.32	0.53
			50	1178	0.17	0.45	0.64	0.17	0.4	0.6
			100	587	0.18	0.4	0.66	0.18	0.33	0.5
		Overlapping	25	2774	0.17	0.38	0.63	0.17	0.32	0.45
			50	1178	0.18	0.44	0.74	0.18	0.4	0.63
			100	587	0.17	0.38	0.67	0.17	0.31	0.63
90	retweet	Disjoint	25	2061	0.08	0.26	0.59	0.08	0.2	0.39
			50	907	0.09	0.17	0.58	0.09	0.14	0.38
			100	469	0.11	0.31	0.54	0.11	0.22	0.38
		Overlapping	25	2061	0.09	0.23	0.67	0.09	0.18	0.47
			50	907	0.09	0.19	0.58	0.09	0.16	0.49
			100	469	0.09	0.28	0.41	0.09	0.22	0.19
	mention	Disjoint	25	2384	0.08	0.16	0.58	0.08	0.13	0.43
			50	1037	0.09	0.2	0.58	0.09	0.13	0.36
			100	529	0.06	0.07	0.52	0.06	0.05	0.3
		Overlapping	25	2384	0.08	0.15	0.61	0.08	0.12	0.4
			50	1037	0.08	0.13	0.72	0.08	0.09	0.37
			100	529	0.09	0.06	0.33	0.09	0.05	0.19
	follower	Disjoint	25	2774	0.1	0.3	0.64	0.1	0.23	0.44
			50	1178	0.09	0.17	0.62	0.09	0.13	0.42
			100	587	0.09	0.15	0.68	0.09	0.1	0.37
		Overlapping	25	2774	0.08	0.3	0.63	0.08	0.25	0.43
			50	1178	0.09	0.18	0.58	0.09	0.13	0.5
			100	587	0.1	0.19	0.4	0.1	0.12	0.27