Final Report: Local Structure and Evolution for Cascade Prediction

Jake Lussier (lussier1@stanford.edu), Jacob Bank (jbank@stanford.edu)

ABSTRACT

Information cascades in large social networks are complex phenomena governed by such diverse forces as the diffusion medium, user trends, and the information content itself. While these influences might be difficult to understand and model directly, the structure and evolution of the cascade can be used as a proxy for the sum effect. To this end, we study information cascades on Twitter and focus especially on the utility of local graph structure analysis for categorizing, understanding, and predicting cascade evolution. Specifically, after presenting basic statistics, we categorize cascades based on size and growth. We then count graphlet frequencies for different cascade categories in order to understand their structural differences. We also explore how these differences arise by counting graphlet frequencies at different points during the evolution process. Finally, we construct a machine learning framework for predicting cascade size.

1. INTRODUCTION

An information cascade is a phenomenon by which people influence others to acquire information or behaviors. Clear understanding and effective modeling of such cascades could have significant impact in practical domains ranging from viral marketing to crisis detection to epidemic outbreak prevention. Scientists have therefore examined these processes for decades, but it is only now, with the emergence of online social networks, that data and resources allow for large-scale study.

Previous research [7], [6] has reported large-scale properties of cascades, as well basic results pertaining to local structures. Recent studies [2] have also examined individual nodes in cascades and constructed predictive models for who will spread what contagion. We continue this line of study by conducting a more detailed analysis of local structures in order to predict eventual cascade size. We also go beyond the realm of previous studies by examining how local structures emerge and evolve. The outline for this paper is as follows:

1. We first describe in greater detail the related work in Section 2.
2. We then describe out data sources in Section 3.
3. In Section 4, we formally define our directed-acyclic graph (DAG) model for information cascades and use this model to construct cascades from our data.
4. Next, in Section 5, we count three and four-node graphlets in all cascades and plot graphlet frequencies for different cascade categories based on size and growth rate.
5. We study local structure evolution in Section 6 by computing graphlet frequencies with each node addition.
6. Finally, in Section 7, we construct a machine learning framework to predict, for a cascade of ten users, whether or not it will grow to over twenty nodes.

2. RELATED WORK

Recent work on information diffusion has explored and characterized information cascades in a variety of online network settings. In the work in [7], Leskovec et al investigate cascade patterns by looking at person-to-person product recommendations from a large online retailer. The study reports heavy-tailed cascade size distributions across all products and identifies identifying the most frequent patterns. In a study on another setting of cascades [6], Leskovec et al analyze the temporal patterns, shapes, and sizes of cascades in large blog graphs. The study finds power law size distributions as well, and a tendency for cascades to form “star” shapes.

In this work, we focus on the microblogging platform Twitter. As Twitter has grown in traffic and siginificance, it has received a significant amount of academic study. In early work in [4], Java et al provide initial analysis on the topological and geographical properties of the twitter social graph along with observations on what type of content people tweet. In the work in [3], Huberman et al perform a more detailed investigation of the social network, trying to better understand the nature of social interactions on Twitter. In another interesting line of work in [1], Cha et al develop a framework to measure and model an individual’s influence on twitter, finding that a high follower count does not necessarily lead to many retweets and mentions.

Twitter also provides a particularly rich and significant setting for the study of information diffusion, and prior work has investigated cascades from different angles. In one general study in [5], Kwak et al perform a quantitative analysis on information diffusion, analyzing trending topis and retweet dynamics. Another related study is described in [2], in which the authors analyze cascades of URLs on Twitter with the goal of predicting which users will tweet which URLs. The authors make a variety of observations of the structure of cascades, finding power law size distributions and a lognormal distribution of the diffusion delay. In the work in [8], Sadikov et al conduct another study on Twitter data in which they tackle the problem of analyzing, understanding, and correcting for missing data in information cascades.

3. DATA

For our project, we use two Twitter datasets. First, we have the Twitter social graph from July 2009, made publicly available by the authors of [5]. This data contains 41.7 million users and 1.47 billion social relations. In this paper, we construct the social graph $G = (V_G, E_G)$ where V_G is the set of all users, and an edge $(u, v) \in E_G$ if and only if u follows v. Second, we have complete tweet data from January 2011, which includes author, text, timestamp, and many other attributes. In the next section we describe how we use this data to construct our cascades.

4. CASCADE CONSTRUCTION

After constructing the social network as described above, we then process two weeks of tweet data (1/5/2011 - 1/18/2011) and construct cascades for all retweets. Specifically, for each retweeted tweet in this period, we construct a cascade $C = (V_C, E_C)$, where V_C is set of users who retweeted that tweet, and edge $(u, v) \in E_C$ if and only if $(v, u) \in E_F$ and u retweeted t before v retweeted t. Thus, each cascade will be a DAG where edges represent potential influence. Note that in contrast to a simple tree representation, this *DAG cascade model* captures cases where one node retweets a tweet that was previously shared by multiple users he or she follows. Moreover, in the context of this study, this will allow us to study more complex local cascade structures.

This results in 4,926,822 cascades that follow the size (number of nodes) distribution shown in Figure 1a. As can be seen, this distribution is heavy-tailed, which agrees with observations of cascades size distributions in earlier works. However, despite the fact that our DAG cascade model allows for many *more* edges than nodes (since each user can be influenced by multiple other users), we see from the edge count distribution (not shown in the interest of space) that there are in fact many *fewer* edges. This implies that there are many disconnected nodes and likely suggests that the out-of-date social graph does not contain many of edges over which information diffuses.

We also investigate the temporal patterns of retweeting in these cascades, looking specifically at retweet *response times*, which we define as the time difference between a retweet and the latest previous retweet or tweet, and retweet *lag times*, which we define as the time difference between a retweet and the original tweet. We find that most retweets occur immediately after the original tweet and in rapid succession (less than ten second). How is it that so many retweets are so immediate? In looking at these cases, we have found that such retweets are often posted by bots, most commonly for stock trading and updates. We show the response time distribution in Figure 1b,

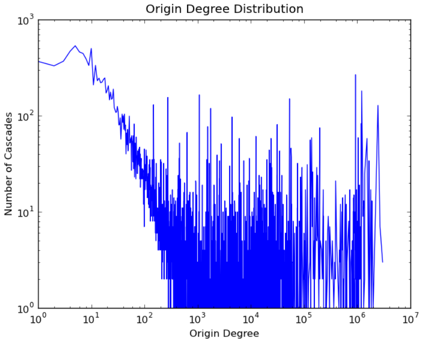


Figure 2: Cascade origin degree distribution.

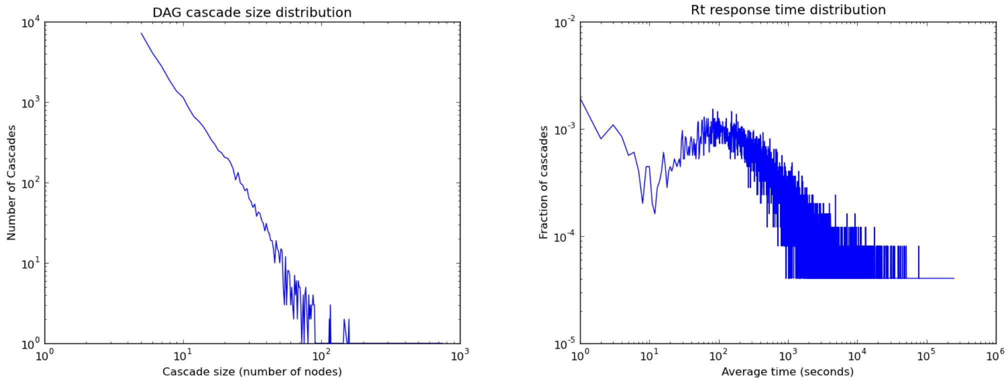
Finally, we explore the relationship between the social network and the retweet cascades. We first examine the effect of the *origin degree*, whim we define as the in-degree of the user who posted the earliest tweet / retweet in the cascade. The origin degree distribution shown in Figure 2 has a much fatter tail than previous distributions, suggesting that original tweeters tend to have high degree. We also examine the correlation between origin degree and cascade size and find no significant correlation. This might be an interesting feature of cascades on Twitter, but is more likely the result of our out-of-date social network. We do a similar analysis for *average degree* of a cascade, which we define as the average in-degree (number of followers) for all nodes in the cascade. We see a similar distribution of values and again see little to no correlation.

5. GRAPHLET COUNTING

In the previous section, we examined large-scale properties of retweet cascades on Twitter. In this section we now study local cascade structures by computing graphlet frequencies for all three and four node graphlets. Since the subgraph isomorphism problem is NP-Hard, previous works have used various techniques, such as the multi-level hashing method described in [9]. For our study, we utilized the subgraph counting code implemented in SNAP to count all three and four node graphlets for all cascades. We then describe each cascade C in terms of a graphlet frequency vector v_C, where $v_C^{(i)}$ is the number of times the i^{th} graphlet appeared in cascade C. Moreover, for a set S of cascades, we can compute the average graphlet frequency vector $\hat{v}_S$, where the i^{th} entry is given as follows:

$$\hat{v}_S^{(i)} = \frac{\sum\limits_{C \in S} v_C^{(i)}}{|S|}$$

With this, we can now compare local structures for small and large cascades by choosing some size threshold β, partitioning cascades based on whether the number of nodes is less than or greater than or equal to β, and computing the



(a) `Cascade size (number of nodes).`

(b) `Response time.`

Figure 1: Cascade distributions.

average graphlet frequency vectors for both sets. We do this for β values of 10 and 40, and show the results for $\beta = 10$ in Figures 3a. In this plot, small and large cascades follow the same general pattern of peaks, with the only notable difference being that graphlet 0 (a 3-node chain) is far more prevalent in small cascades, and graphlet 4 (a 3-node star) has a much higher peak in the large cascades. Although not shown, the plot for $\beta = 40$ is different in that large cascades look very different with a giant peak at 4, a small peak at 0, and a flat line elsewhere. This seems to indicate that our largest cascades are made up of many tiny stars.

We also compare local structures for slow-growing and fast-growing cascades by choosing some growth rate threshold α, and doing similarly as above. We choose $\alpha = 3600$ seconds (1 hour), and compare for cascades of about the same size. Results for cascades between sizes 10 and 20 are shown in Figure 3b. As can be seen from the first plot, of cascades sized 10-20 nodes, fast cascades have many more instances of graphlet 8 while fast cascades have many more instances of graphlets 4 and 31. Although not shown, in the plot for cascades of size 20-30, graphlet 0 is very prevalent in fast cascades, while more complex structures, like graphlets 21, 25, 31, and 36 appear much more in slow cascades. Across both of these plots, we can observe this same interesting effect that the more complex graphlets appear more in slow cascades, whereas simple chains and stars characterize fast cascades.

6. CASCADE EVOLUTION

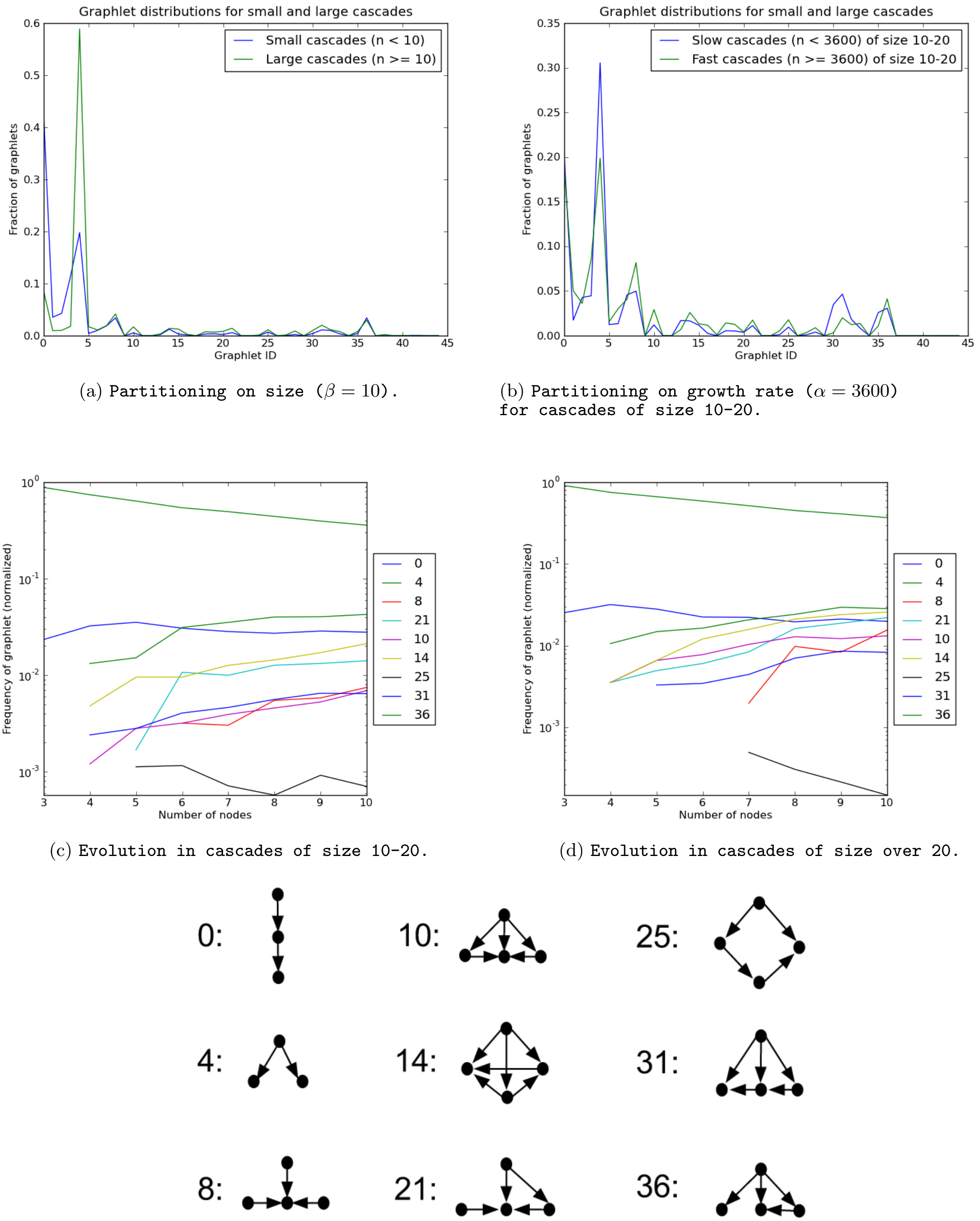
In the previous section, we constructed a cascade for each retweeted tweet, added all nodes and corresponding edges that we observe during the two weeks of data, and then computed graphlet frequencies for these final complete cascades. In this section, we compute the graphlet frequency vector with each node addition so that we might be able to understand the process by which cascades grow.

The plots in Figure 3c-d show the relative frequencies of common graphlets in the early evolution of cascades. The x-axis shows the number of nodes currently in the cascade, and the y-axis shows the proportion of all graphlets. Each line is a different graphlet (the graphlet key is shown in Figure 3e). Figure 3c shows the early evolution of cascades that reached a size between 10 and 20 nodes, and Figure 3d shows the early evolution of cascades that reached a size over 20. Both plots show the same general trend where graphlet 4 (a 3-node star) dominates throughout the early evolution, but its proportion decreases quickly. The more complex 4-node graphlets – such as 10, 14, 21, and 36 – gain a larger and larger share as time progresses. Comparing across the plots, we notice that graphlets 10, 14, 21, and 31 (4-node graphlets with many edges) are more represented in Figure 3d, indicating that they appear more in the early stages of large cascades than small cascades.

7. CASCADE SIZE PREDICTION

In this section, we frame cascade size prediction as a supervised classification problem in which we observe the cascade up to size 10 and then aim to predict whether or not its eventual size will exceed 20. In the previous sections, we studied temporal retweet trends, the relationship between the social graph and cascade size, and graphlet frequencies for different kinds of cascades. We therefore define three kinds of features:

- Temporal features: these features aim to capture the temporal dynamics of the first 10 retweets. Here we include the *average lag time* and the *average response time*, as defined in Section 4.
- Follower features: these features aim to capture nodes' connectivities in the social graph. To this end, we include the *origin degree* and the *average degree*, as defined in Section 4
- Graphlet features: these features aim to capture the frequently appearing local graph structures that appear in the cascade up to 10 nodes. For this, we include each entry in a cascade's graphlet degree vector as a feature. Note that we could include each entry in a cascade's vector at each point in its evolution, but given the weak signal from the out-of-date social graph, we choose to only include the entries from the vector for the cascade of size 10.



(a) Partitioning on size ($\beta = 10$).

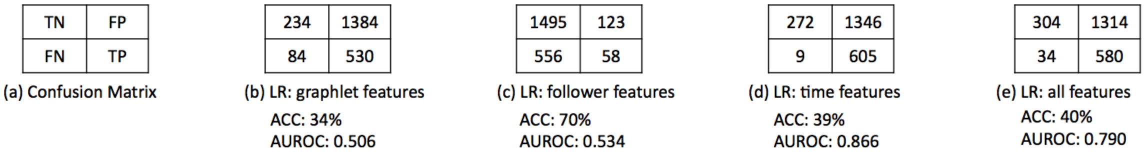
(b) Partitioning on growth rate ($\alpha = 3600$) for cascades of size 10-20.

(c) Evolution in cascades of size 10-20.

(d) Evolution in cascades of size over 20.

(e) Visualizations of Common Graphlets.

Figure 3: Average graphlet frequency vectors after partitioning on cascade size (plots (a) and (b)) and cascade growth rate (Plots (c) and (d)). Visualizations for frequent graphlets are shown in (e).



TN	FP
FN	TP

(a) Confusion Matrix

234	1384
84	530

(b) LR: graphlet features
ACC: 34%
AUROC: 0.506

1495	123
556	58

(c) LR: follower features
ACC: 70%
AUROC: 0.534

272	1346
9	605

(d) LR: time features
ACC: 39%
AUROC: 0.866

304	1314
34	580

(e) LR: all features
ACC: 40%
AUROC: 0.790

Figure 4: Prediction confusion matrices, accuracies (ACC), and areas under the receiver operating curve (AUROC) for different experiments. (a) defines the entries of the confusion matrix.

With this, we define four datasets that include temporal features, follower features, graphlet features, and all features, respectively. All of these datasets have imbalanced class distributions (5262 cascade do not grow to 20 nodes, while only 2175 reach 20 nodes), so we perform a 70-30 split (70% train, 30% test) and undersample the training set to equality. After doing so, we are left with 5205 training examples (1561 negatives, 1561 positives), and 1618 testing examples with (1618 negatives, 614 positives) for each dataset. Note that since we only undersample the training set, the testing set is still representative of the original distribution. Moreover, each dataset has the same cascades in the training and testing sets, so we can evaluate performances directly.

For our experiments, we train four logistic regression classifiers on the four training sets, and then evaluate each of them on the corresponding testing set. The results are shown in Figures 4b-e. Note that accuracy is a poor performance measure for these imbalanced classification problems since a simple classifier that predicted the negative class for all examples could achieve 72% accuracy, so we also include area under the receiver operating curve (AUROC). Looking the AUROC values, we see that graphlet features give only a marginal lift over random, as do follower features. Logistic regression with just the time features achieves a significantly higher AUROC of 0.866, and even outperforms the classifier with all features. This suggests that the time features are by far the strongest and that other features provide such weak signals that they essentially add noise to the problem.

8. CONCLUSION

In this paper we presented a detailed analysis of information cascades on Twitter and explained a supervised classification framework for predicting cascade growth. In Section 4, we defined the structure of our cascades, described temporal retweeting trends, and explored the relationship between the follower graph and cascades. In particular, we showed that most retweets follow shortly behind the original tweet, and found little relation between degrees of nodes in a cascade and the size of that cascade. Next, in Section 5, we studied local structure by counting graphlets and noting those that occur most frequently in different kinds of cascades (small and large, slow-growing and fast-growing). We then proceeded in Section 6 to examine cascade evolution by conducting similar experiments at each point during a cascade's growth. Finally, in Section 7, we used our previous analyses to extract features from cascades when they had ten nodes, and used these to predict whether or not at least ten additional nodes would retweet the tweet. We found that time features were by far the most salient, and that other features gave little to no signal.

For future work, we would like to study the effect of the content itself on diffusion. For example, do news items diffuse differently than entertainment items? We might also examine the effect of different sources. Previous work has shown that most retweets on Twitter originate with "celebrities," so we might incorporate this into our framework.

Finally, throughout this work, we found the social graph resulted in inclusive or noisy results. Since the social graph is from two years before the tweet data, we hypothesize that this is because the social graph is out-of-date. We hope to obtain a more recent social graph in the future and rerun the above experiments with clean data. This might allow for more novel insights and better prediction results.

9. REFERENCES

[1] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM)*. AAAI, 2010.

[2] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers-predicting information cascades in microblogs. In *Proceedings of the 3rd conference on Online social networks*. USENIX Association, 2010.

[3] B. Huberman, D. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. *First Monday*, 2009.

[4] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM, 2007.

[5] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*. ACM, 2010.

[6] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. *Arxiv preprint*, 2007.

[7] J. Leskovec, A. Singh, and J. Kleinberg. Patterns of influence in a recommendation network. *Advances in Knowledge Discovery and Data Mining*, 2006.

[8] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011.

[9] S. Wernicke. Efficient detection of network motifs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2006.