

Cascade Analysis with Limited Network Data

Esther Hsu (estherh@stanford.edu)
 AJ Minich (ajminich@stanford.edu)
 Daniel Ong (danielo@cs.stanford.edu)

Abstract—Many modern network applications (epidemic containment, advertising, political campaigning) revolve around cascade behavior. To this end, many network models exist to mimic the complex interactions of nodes and edges within cascades. However, many of these models require a significant amount of data to set up, are conceptually difficult to understand, and provide more functionality than the analysis requires.

In this paper, we investigate two simple methods of using the past behavior of nodes and edges to predict future cascade behavior. Our goal is to explore several cascade analysis techniques that require almost no data about the network’s underlying structure, can be easily represented in typical network data structures, and provide high-level statistics about the network’s cascading behavior.

I. INTRODUCTION

Cascades within a network represent the spread of ideas, products, or otherwise contagious materials, to one or more connected nodes in the network. Cascade analysis has recently begun to serve practical importance for any application in which the size and extent of network effects are of interest: advertising, epidemiology, political campaigning, etc.

Analyses of these cascades have a variety of objectives:

- **Cascade Size:** determining the number of nodes that participated in or were affected by the cascade.
- **Cascade Diameter:** determining the spreadout of the cascade - that is, the longest path between the originator and affected nodes.
- **Influential Nodes:** identifying the nodes that have the greatest influence on the size of the cascade.
- **Community Identification:** segmenting the larger network into smaller communities that are typically part of the same cascades.

Some of these objectives, particularly the cascade diameter and the maximally influential nodes, require knowledge of the full underlying network. To calculate these parameters, we must either acquire the entire network structure, or otherwise infer the structure from time-based data as discussed by Gomez-Rodriguez et al [3].

However, analyzing the cascade sizes and the communities present over all the cascades do not by definition require the entire underlying graph. Suppose we only know what nodes belonged to given cascades, but not the relationships between the nodes. It is easy to see that this is the case in many situations - for example, financial transactions between individuals, biological epidemics among populations, and political campaigns among unknown constituents. In these situations, analyzing cascades and predicting future cascade behavior is crucial to taking advantage of network effects, despite the lack of insight into the underlying network.

Thus we suggest analysis methods that obviate the most intensive data need: the requirement of knowing the full underlying network. If we know high-level parameters of the network (the number of users, the approximate number of edges, the power-law factor α for the degree distribution) and have modest historical cascade data (which nodes participated in which cascades), we can generate models which behave at a high level like the actual network, despite not being based on the original network’s structure.

In this paper, we first investigate two models of predicting cascade sizes, and evaluate each model’s ability to predict the reach and spread of future cascades. In the first model, we use the participation histories of individual nodes to generate new graphs at any scale that exhibit cascade behavior similar to the original, unknown network of interest. In the second model, we infer the distribution of edges through the similarities of node tweeting; specifically, a given edge’s weight is proportional to the number of cascades shared by the two endpoint nodes. These models both exhibit scale invariance, meaning that even with an original dataset representing only a subset of the larger graph, the models can predict cascade sizes at the larger scale of the entire network.

We then turn our attention to community detection and analysis. This operation is typically performed by locating the connected components or strongly connected components of the network, and then classifying them based on the content of the tweets. Such an approach is clearly impossible without knowing the network structure, but we demonstrate that community-finding can be performed efficiently using only the cascade sets.

Finally, we end with an explanation of the limitations of these techniques. No analysis without the underlying network will perform as well as an analysis with the network, but we argue that the less intense data requirements make up for the decrease in accuracy and insight.

II. PRIOR WORK

As cascade prediction is relevant for many purposes, many models already exist for simulating cascade behavior. Although all of these models utilize the network’s structure in some way, many provide insight into modelling approaches and analysis techniques that can be helpful when our understanding of the underlying network is limited.

The earliest deterministic models are based on single-valued functions, such as Granovetter’s famous thresholding model [4]. In this model, a threshold value t_i indicates the minimum number of node i ’s neighbors that must be activated in order for node i to participate in the cascade.

Many analyses aim to identify the nodes that maximally influence the size of the cascade, as in the case of finding the optimal advertising targets or the most biologically contagious member of a population group. Such analyses (known as max-cover problems) have been shown to be NP-hard [5], and can only be estimated in polynomial-time using greedy hill-climbing techniques [1].

Other methods use multiple cascade sequences to infer the underlying network, in some cases achieving a very accurate inferred graph [6]. This method requires a time-dependent function, which may not always be available, in order to calculate probabilities and maximize them with a greedy algorithm.

Previous probabilistic models, more in line with the methods used in this paper, attempt to learn infection probabilities [8] to simulate cascades. These models, however, rely on information like underlying network structure and infection times, which large data may not always have.

It is important to note that, in cases where the network is either not available (as in hidden processes) or difficult to attain in its entirety (as in Twitter networks, where the data rate is limited to 350 queries per hour), few models exist to model the cascading behavior. The purpose of this work is to leverage ideas from existing models to create a method which guesses the general networks structure well enough to behave similarly in cascade situations.

III. DATASET

The data for this project came from a single day of Twitter retweet cascades. Using the TwitterStream API, we collected 58,564 tweets over a 24 hour period starting at 8pm PST, 15th November 2011 and stored them into a CouchDB database (figure 1). We monitored specifically for tweets that contained words and phrases related to the ‘Occupy Wall Street’ movement, such as the following:

- Hashtags: e.g. #occupy, #occupywallstreet, #ows
- Keywords: occupy, wall street
- Hot topics: e.g. #nypd, zuccoti park

The time period was especially notable for the fact that at 8:30pm PST, police in New York City started using coercive force against protestors who were ‘Occupying’ Zuccoti Park. The twitterverse exploded with activity, and we monitored the stream, reading related tweets into a CouchDB database (figure 1).

The end result was over 4.3GB of data, which we then filtered selectively to find retweets related to the #occupy network. The end dataset contained a total of 58,564 retweets generated by 17,377 users.

As a first check, we rearranged these retweets into groups based on the original tweet being retweeted, and counted the number of users who were a part of each cascade. The result was a total of 12,213 cascades, with the largest cascades containing up to 800 tweets. As shown in figure 2, the sizes of the cascades follow the expected power-law distribution [2], with $\alpha = 2.46$.

We then post-processed the dataset to acquire two data structures:

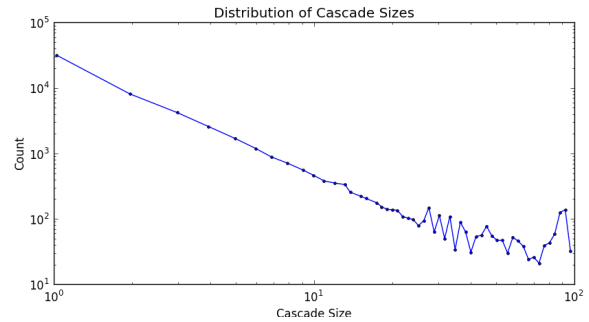


Fig. 2: The distribution of retweet cascade sizes for the #occupywallstreet Twitter datastream captured November 15-16, 2011. The distribution exhibits the expected power-law behavior [2]. The goal of the node-based and edge-based models is to exhibit the same distribution when evaluated over many simulated cascades.

1. **Users:** a map from each unique user to the list of tweets sent by that user.
2. **Cascades:** a map from the text of the retweet to the users who sent that message.

Note that none of this information contains the underlying follower network. Thus we have a dataset that includes only transactions, and not the relationships between actors in the network.

IV. FIRST MODEL: NODE-PARTICIPATION

We begin our discussion of limited-network analytical techniques by introducing a model focused on node activity. In this model, each node has a certain probability of ‘participating’ in a given cascade, and thus we refer to this model as the *Node-Participation Model*.

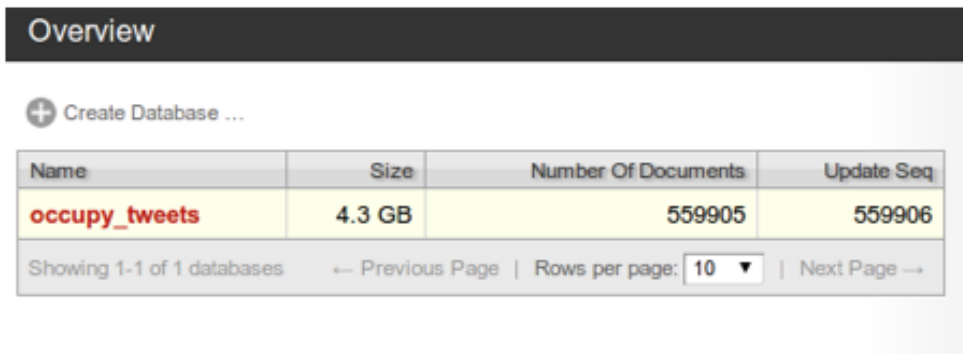
A. Model

Suppose our original graph graph G has n nodes v_1, v_2, \dots, v_n , and we have a set C of cascades. For each node i , we calculate the number of times a_i that the node retweeted a followee’s tweet - that is, the number of times the node participated in a cascade:

$$a_i = \sum_{c \in C} t_i(c)$$

where $t_i(c)$ is an indicator function equalling 1 if $v_i \in c$, 0 otherwise. Thus a is the array of participation counts for all nodes.

We would now like to use a to generate a set of probabilities q_1, q_2, \dots, q_n that each respective node will participate in a future cascade. One way to normalize the number to a probability distribution would be to divide by the total number of unique tweets coming from the user’s followees, but our fundamental goal with this model is to obviate the need for the follower graph, so this approach will not work for us. Instead, we will simply divide by the participation count of the most active user, which both guarantees probabilities between 0 and



Name	Size	Number Of Documents	Update Seq
occupy_tweets	4.3 GB	559905	559906

Showing 1-1 of 1 databases ← Previous Page | Rows per page: 10 | Next Page →



Fig. 1: CouchDB provided an easy method of catching ‘Occupy Wall Street’ tweets from the TwitterStream API and storing them for later analysis.

1 and puts all nodes on a scale irrespective of the original number of cascades in the dataset.

Thus we have the node-participation ratio

$$q_i = \frac{a_i}{\max_i a_i}.$$

Now, our fundamental proposition is that we do not know the full structure of G , but we must still generate a test graph G' . Our goal is to set up G' in such a way that it will serve as a model for cascades, with the resulting distribution of cascade sizes over multiple trials approximately matching the distribution of cascade sizes for the original dataset. Since Twitter, as an online social network, exhibits a power-law distribution on the degrees of the component nodes, we will approximate the original network G with a preferential attachment graph G' containing n' nodes.

It is important to note that we do not necessarily have $n = n'$, since part of the usefulness of G' is its cascade behavior at sizes much larger than n . An advertising agency, for example, might attain cascade data for a Twitter subnetwork G , and then wish to estimate the cascade behavior in the larger Twitter network G' . In such a case, we must have the degree distribution of G' approximately equal to that of G , but it has already been shown that the α coefficient for such a graph is approximately 2 [6], so we can create G' even without much data for the original graph G .

Now, we will treat q as a distribution from which to select node probabilities in the test graph G' . Since power-law distributions are scale-invariant, we can simply choose a probability at random from q for each node in G' . Thus we attain a model G' whose nodes behave similarly to the original nodes in G , but whose structure is completely unrelated.

B. Simulation

To perform simulations, we first implemented the Node-Participation model in Python on top of NetworkX. Each node $v_i \in G'$ possesses an attribute q that indicates whether the given node will participate in a given cascade - that is, $q_i = Pr[v_i \in c]$.

For a given cascade c , we pick a starting node at random from G' . We then perform a breadth-first search starting at the given node, in which we travel first to all the neighbors

of the starting node, then to those neighbors’ neighbors, etc. However, at each node v_i , we are not guaranteed that the search will continue: there is a probability $1 - q_i$ that the node will elect not to participate in the cascade. In such a case, we ignore the node, and continue with the search.

At the end of the simulation, we have a cascade size $\|c\|$ based on the number of nodes participating in the cascade. We run this simulation k times to determine a set $C' = \{\|c_1\|, \|c_2\|, \dots, \|c_k\|\}$ of cascade sizes. Our primary interest is verifying that the power-law distribution of C' displays some similarity to the power-law distribution of the original dataset’s cascade sizes, which we will denote as C .

C. Results

The first experiment was to perform the analysis on a graph G' with the a similar number of nodes as the original graph G , and determine whether the cascade size distribution was approximately the same. As shown in figure 3, we attain a power-law distribution with $\alpha = 2.257$, as compared with the original cascade distribution’s α of 2.46. The similarity is quite good, especially given the simplicity of the model and the inexact number of edges.

Thus we have arrived at a model that gives fairly appropriate cascade behavior, and can be scaled to arbitrary proportions.

V. SECOND MODEL: EDGE-CONDUCTION

The second model is an edge-based model where each edge has a certain probability of ‘conducting’ the given cascade from the source node to the destination node, and is thus called the *Edge-Conduction Model*.

A. Model

Suppose we begin with the nodes of the original graph G . We don’t know the set of edges E in G , but we do know the nodes V . We will start by assuming G is a fully connected graph (every pair of nodes is connected). Now, for each edge $(u, v) \in E$, we will estimate the probability of an edge (u, v) conducting a cascade as:

$$P_{\text{infection}}(u, v) = \frac{\# \text{ of times } u \text{ and } v \text{ were in the same cascade}}{\# \text{ of times } u \text{ was in a cascade}}$$

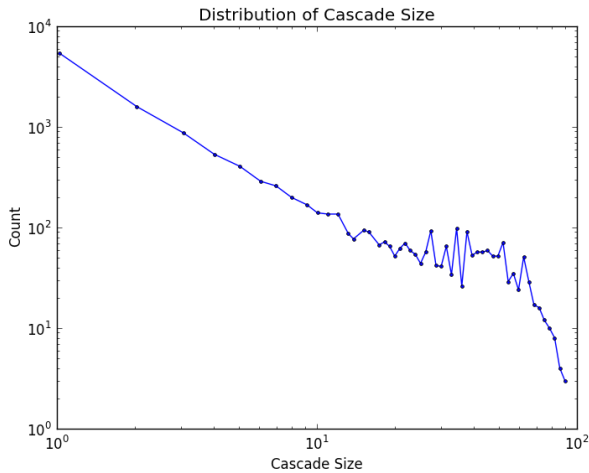


Fig. 3: The distribution of simulated cascade sizes using a Node-Participation model. The α coefficient of the distribution is 2.257, compared with 2.46 for the original distribution.

By calculating $P_{\text{infection}}$ for all edges in G , we arrive at a set of edge conductivities, as shown in figure 4. We will call this distribution r ; note that it is a probability distribution because no $P_{\text{infection}}$ value can ever be greater than 1.

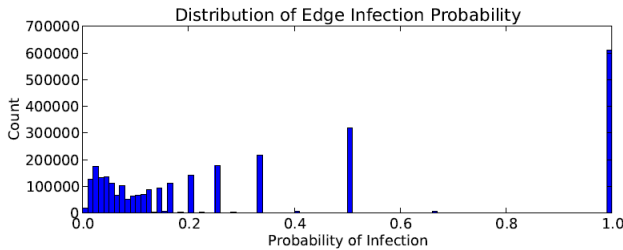


Fig. 4: Distribution of estimated edge infection probabilities

We now generate a model graph G' for simulation. As with the Node-Participation model, we use a preferential attachment graph with the same number of users/nodes as our data, and a degree distribution of approximately 2. We then assign probabilities for each edge q_{AB} by choosing at random from r . This gives us a directed graph model like the one shown in figure 5.

B. Simulation

As with the Node-Participation model, our simulation uses a Python model built on top of NetworkX. Each edge $e_i \in G'$ possesses an attribute r that indicates whether the given node will participate in a given cascade - that is, $r_i = Pr[e_i \in c]$.

For a given cascade c , we pick a starting node at random from G' . We then perform a breadth-first search starting at the given node, in which we travel first to all the neighbors of the starting node, then to those neighbors' neighbors, etc. However, for each edge e_i , we are not guaranteed that the search will continue: there is a probability $1 - r_i$ that the edge will elect not to conduct the cascade to the destination node. In

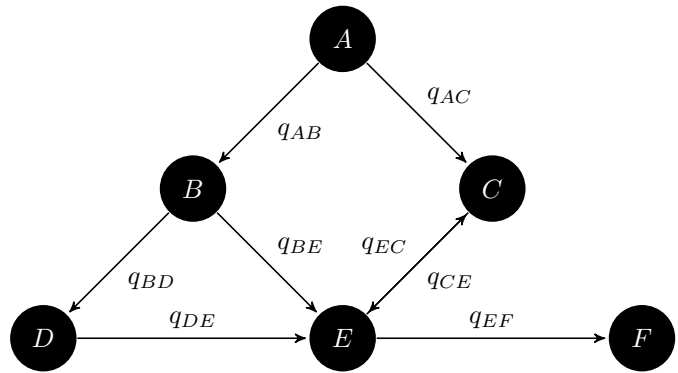


Fig. 5: In the experimental model, each directed edge possesses a $q_{n_1 n_2}$ attribute.

such a case, we ignore the destination node, and continue with the search.

At the end of multiple simulations, we again have a set $C' = \{\|c_1\|, \|c_2\|, \dots, \|c_k\|\}$ of cascade sizes. Our primary interest is verifying that the power-law distribution of C' displays some similarity to the power-law distribution of the original dataset's cascade sizes, which we will denote as C .

C. Results

The distribution of cascade sizes that resulted followed a power-law distribution, with an MLE estimate of $\alpha = 2.2$ (compared to the actual data's $\alpha = 2.46$), as seen in figure 6.

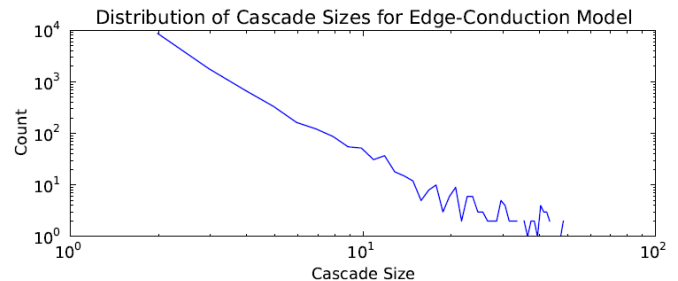


Fig. 6: Distribution of cascade sizes as a result of the Edge-Conduction model

Thus although the Node-Participation model gave better absolute performance in terms of α correlation, the Edge-Conduction model also provides fair accuracy. We suggest that this model would outperform the Node-Participation model in cases where relationships between users are highly bivariate - for example, in the case of bipartite political discourse.

VI. THIRD MODEL: COMMUNITY-SET

The third model seeks to detect communities using only the knowledge of which nodes are in a cascade, without full knowledge of the cascade's path, and incomplete network data. We found this model useful due to Twitter's retweet format (which does not preserve cascade paths), and also due to their API Rate Limiting, that makes complete network data difficult (and possibly expensive) to purchase.

A. Model

The model approaches the problem from a ‘naive’ perspective. Given that a retweet can be seen as an ‘endorsement’ of a tweet’s message, a given user retweeting another indicates a certain agreement of values. We extrapolate this to indicate that they belong to the same ‘community’. Furthermore, the model assumes that a user’s values are constant across their retweets. The corollary of these two assumptions is that a nodes in two different cascades are all part of the same community, if there is a common node in both cascades.

The model puts each cascade’s participants in a set, and then joins any two sets that have a intersection > 0 . It does so till every set is disjoint. Each resultant set is a ‘community’.

B. Results

The resultant communities featured one large ‘community’, and many small ‘communities’, as shown in figure 7. This is unsurprising given the ‘naive’ nature of the algorithm, and given the cascade distributions.

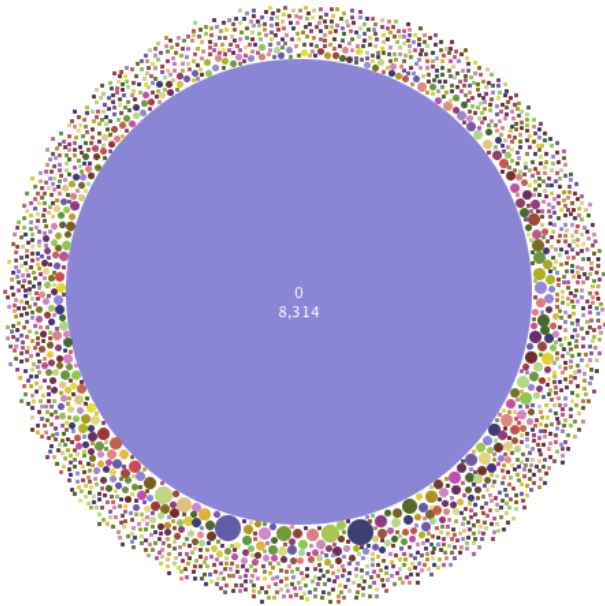


Fig. 7: A bubble representation of the sizes of the communities detected by the Community-Set model. The largest community by far is the Occupy Wall Street supporters, representing well over 90% of the total users.

To ascertain the reason for the single huge community, we sampled from each community’s actual tweet texts. The content of the tweets is highly indicative of the communities’ attitudes toward the Occupy Wall Street movement.

Sample of 5 tweets from large ‘community’:

- **RT zunguzungu:** I love it when occupy people hum the imperial theme from Star Wars at riot cops. They were doing it at Cal on Wednesday, ...
- **RT zuccottipark:** American homes destroyed by Al Qaeda last year = 0. Destroyed by banks = 1,200,000. Who are the real terrorists? #ows ...

- **RT zoeschlanger:** RT RDevro: The NYPD now tearing, literally ripping and tearing, the safer spaces tent. The tent intended to make wome ...
- **RT zorganizr Curtis:** sitting in #blackroots11 neworganizing on poc media organizing while liberty square is being raided #ows #reallife ...
- **RT zedshaw:** Your nightly dose of irony: The OccupyOakland protests will cost the city of Oakland \$2.4mil of 99%-er taxes.

Sample of 6 tweets from small ‘communities’:

- **RT LoniLove:** I’m gonna occupy a restaurant on wall street. #yeswecaneat
- **RT ClaudeKelly:** Occupy Bed
- **RT Votekick:** OWS Go Home! (live at <http://t.co/vLY0hQ4w>)
- **RT kmflett:** Given the alarming attack on democratic rights in New York I’m expecting Cameron to announce a UN resolution & air strikes ...
- **RT LibyaLiberty:** A tweet I wish I’d written: tomgara: Every time you seriously compare Occupy protests to the Arab uprisings, God kill ...
- **Thezog:** Police are raiding Occupy Wall St right now!! News is being blocked from reporting. Make your voice heard. <http://t.co/dG2gVxsm>

This brief, random sampling from the communities suggests that the large community can be characterized as the ‘pro-occupy’ community, while the smaller communities can be characterized as either ‘anti-occupy’, ‘jokes’ or ‘pro-occupy’ communities that are isolated from the larger community, with the former two being more common. This is interesting because it indicates the ‘anti-occupy’ movement, while possibly sizeable in number (approximately $\frac{1}{2}$ of all users) is very fragmented. The largest small community consists of 28 nodes, and the majority of communities have less than 5 nodes.

C. Further Research

The ‘naive’ community detection model discussed above is deterministic based only on whether two cascades have at least one user in common. In reality, users from different communities may retweet the same tweet, even if they have differing viewpoints on the content being cascaded. A probabilistic representation would more accurately capture the relative connections between separate users, such that communities would have a core of high-probability connections in the core, and lower-probability relationships on the periphery. Future work may expand on this concept of non-deterministic connections between users for the purpose of community analysis.

A second improvement involves better datasets regarding the content of the communities. In the example above, we sampled liberally from various communities to understand user commonalities. A more rigorous experiment would involve creating a training set by tagging users with demographic features, and then ‘learning’ how these users typically associate themselves with various communities.

VII. CONCLUSIONS

Analysis using the full underlying network certainly remains the most accurate method of characterizing cascade behavior,

but for cascade analyses on datasets that do not include the actual network, the above models approximate the behavior of cascades to sufficient accuracy. It also allows the characterization of network communities using a simple approximation of behavior. We have demonstrated the ability to answer several specific high-level questions about current and future performance of the network in cascade scenarios. We foresee this work being extended and applied to studies in which the underlying network is unavailable or very difficult to ascertain, such as word-of-mouth advertising, epidemic containment, and political campaigning.

REFERENCES

- [1] W. Chen, Y. Wang, S. Yang. Efficient Influence Maximization in Social Networks. In Proc. KDD, 2009.
- [2] W. Galuba, D. Chakraborty, Z. Despotovic. Outtweeting the Twitterers - Predicting Information Cascades in Microblogs. In Proceedings of the 3rd conference on Online Social Networks, WOSN 10, 2010.
- [3] M. Gomez-Rodriguez, J. Leskovec, A. Krause. Inferring Networks of Diffusion and Influence. In Proc. KDD 2010.
- [4] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology* 83(6):1420-1443, 1978.
- [5] D. Kempe, J. Kleinberg, E. Tardos. Maximizing the Spread of Influence through a Social Network. In Proc. KDD 2003.
- [6] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, M. Hurst. Cascading Behavior in Large Blog Graphs. In Proc. SIAM International Conference on Data Mining, 2007.
- [7] S. Morris. Contagion. *Review of Economic Studies* 67, 57-78, 2000.
- [8] A. Goyal, F. Bonchi, L.V.S. Lakshmanan. Learning influence probabilities in social networks. In Proc. WSDM, 2010.