

# Predicting Network Response Times Using Social Information

Chen Liang, Sharath Hiremagalore, Angelos Stavrou and Huzefa Rangwala

*Center for Secure Information Systems*

*George Mason University, Fairfax, VA 22030*

*+1-(703)-993-3772, +1-(703)-993-4776 (fax)*

*{cliang1, shiremag, astavrou, hrangwal}@gmu.edu*

**Abstract**—Social networks and discussion boards have become a significant outlet where people communicate and express their opinion freely. Although the social networks themselves are usually well-provisioned, the participating users frequently point to external links to substantiate their discussions. Unfortunately, the sudden heavy traffic load imposed on the external, linked web sites causes them to become unresponsive leading to the “Flash Crowds” effect.

In this paper, we quantify the prevalence of flash crowd events for a popular social discussion board (Digg). We measured the response times of 1289 unique popular websites. We were able to verify that 89% of the popular URLs suffered variations in their response times. By analyzing the content and structure of the social discussions, we were able to forecast accurately for 86% of the popular web sites within 5 minutes of their submission and 95% of the sites when more (5 hours) of social content became available. Our work indicates that we can effectively leverage social activity to forecast network events that will be otherwise infeasible to anticipate.

**Keywords**—Flash Crowds, Traffic Prediction, Social Networks, Website Response Time, Social Content Data Mining.

## I. INTRODUCTION

Public discussion boards have become popular over the years due to their crowd-sourcing nature. Indeed, their members have the ability to post and express their opinion anonymously on stories that are shared publicly. The popularity of these stories is voted upon by other anonymous readers who are also members of the discussion board. Over the last few years, several websites, such as Digg [20], Reddit [11], Delicious [21] offer these services. Through these sites, users organize, share and discuss interesting references to externally hosted content and other websites.

There has been a plethora of research that focuses on analyzing the discussion network structure [30], relationships [17], [2], even using the social network as an anti-spam and defense mechanism. One aspect of discussion boards that has received less research attention is the effect they have on externally hosted websites. Indeed, public discussion boards and crowd-sourcing sites can cause instantaneous popularity of a website owing to discussions in blogs or posts of other website, known as the “Flash Crowd” effect: a steep and sudden surge in the amount of traffic seen at these sites. As a result, these unanticipated flash crowds in the network traffic may cause a disruption in the existing communication infrastructure and disrupt

the services provided by the website. But how prevalent is this “Flash Crowd” phenomenon?

We show that a large portion of the websites that become popular through stories on public discussion boards suffer from the flash crowd phenomenon. These websites exhibit high latency and response time variation as they increasingly become popular. To support our hypothesis, we measured periodically and over a large period of time the download times for all the external URLs that were submitted to a social discussion board using many network vantage points. We used PlanetLab, a distributed platform that provides servers located all over the globe. The external websites’ response times were measured concurrently on several PlanetLab nodes across North America. Computing the changes in the website response time from different locations eliminates the bias introduced by observing measurements at a single location. Then, we computed the correlation values between the variation in the measured network latency with the popularity increase of website linked to a social discussion board. We were able to confirm that 89% of the popular URLs were adversely affected with 50% having correlation values above 0.7. This is a significant portion of the submitted URLs and warrants investigation into techniques to predict these sudden spikes of traffic ahead of time.

## II. CORRELATING POPULARITY WITH RESPONSE TIME

### A. Motivation

Our initial target was to assess the extent of the “Flash Crowd” effect for websites that are linked to popular stories on social discussion boards. Figure 1 illustrates the motivation for our problem. The layout of Digg home page presents users with the most popular links (story) to external web resources. A story gains popularity as users comment and “Digg up” a story, *i.e.*, click on a link to increase the Digg Number of that story. More popular stories are prominently displayed at the top of the website. This could lead to some stories becoming very popular in a short span of time increasing the load on the servers that host this story. The consequence of this is a bad user experience where the site loads very slowly or network timeouts as an effect of the flash crowd.

But how proliferate is the “Flash Crowd” phenomena for publicly accessible discussion boards?

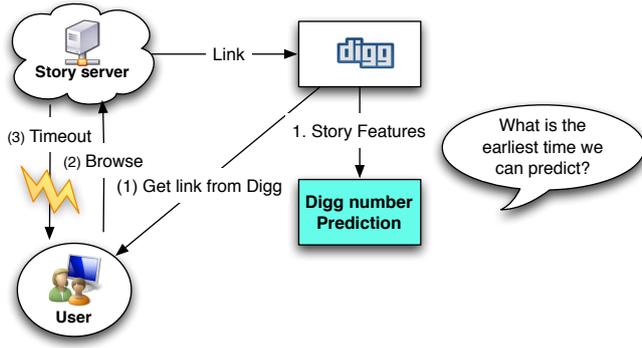


Figure 1. This figure illustrates the effects of a “Flash Crowd” event. The popularity of the social discussion board causes the externally linked website to become slow or even unresponsive.

### B. Website Response Time

The first step in estimating the prevalence of the Flash Crowd effect is to accurately measure the network download and response time of all the external web sites that are linked via the social network discussion board. This study has to be done over a large period of time and for many URLs spanning many different and geographically distributed external story websites. Moreover, to be able to perform a non-biased estimate of the web site latency, we had to perform our measurements from many geographically- and network-wise distinct network points. To that end, we deployed the latency measurement code on 30 nodes in Planetlab. Planetlab provides nodes with the same server specification, namely with 1.6Ghz, 2G memory, 40G Hard Disk. Every 10 minutes, we identified the 500 most popular stories from Digg based on their score and we stored the URLs that they point on external websites on each of the Planetlab nodes. For each of those external URLs, we computed the network latency of their hosting website by computing the amount of time that was required to download their content to the Planetlab nodes. To achieve that, we employed wget [18], a popular HTTP mirroring tool. We selected wget because of its simplicity and its capability to measure the content network dependent download time precisely and without being affected by the potential delays introduced by Javascript or other active content.

Furthermore, throughout our measurements, we downloaded first-level content and we did not follow links or received content from websites that were pointing outside the domain of the measured URL. Of course, over time new stories become popular while others are removed. We keep track of all the stories. In addition, we did not perform all our downloads simultaneously to avoid performance degradation due to network limits or bandwidth exhaustion. Instead, we only probed 20 URLs within a 5 minute window of time and with random start times. We repeated the network latency measurements every ten minutes and collected the timing results for each site.

To account for the fact that some websites may become

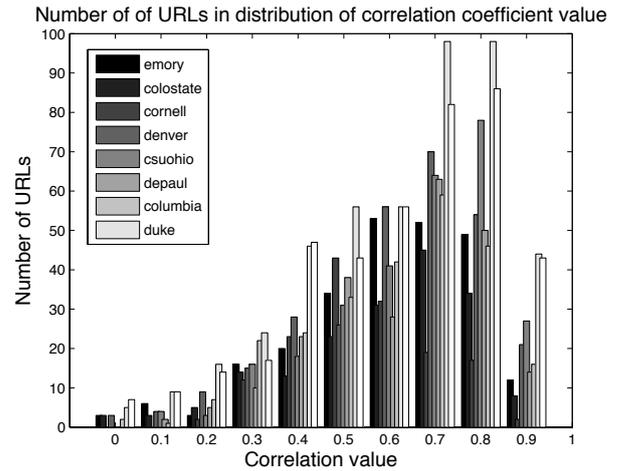


Figure 2. Comparison of Correlation Coefficient Between Eight Nodes from US-based nodes

unresponsive and lead to the stalling and accumulation of wget processes, we chose to terminate all unresponsive downloads within 2 minutes if no data has been received from the remote website. Moreover, in order to perform accurate correlations between the Digg score and the network latency trends, we had to make sure that the Planetlab hosts have accurate time within the window of measurement (2 minutes). We used a Network Time Server (NTP) to synchronize all hosts every minute. With this algorithm, we were not only able to identify websites that were slow or unresponsive, but also provide a better estimate of the time that these sites were exhibiting this behavior because we obtained more measurements.

### C. Correlating Popularity to Latency

We deployed our code on Planetlab nodes and we tracked the Digg number and response times for downloading content and measuring the resulting network latency. All of our results indicated that, as Digg number increases, the variation in the measured response times and perceived latency increases accordingly. Therefore, we computed the correlation between Digg number and the standard deviation of latency to show that, as Digg number increases, the latency is highly volatile. Indeed, we used the standard deviation (STD) of the website response time to model the variations of the latency. We then correlated the computed latency STD values for each URL with its corresponding Digg number for all time periods. We used a fixed time window size equal to 20000 seconds, that is at least 2 times larger than average capture interval. Within each time window, we collected the average Digg number and maximum latency both for the correlation value computation and Standard deviation value of latency. Next, we present results that are more representative.

We generated results from eight US-based nodes and presented the distribution of correlated value in Figure 2. This figure shows the number of URLs plotted according to the correlation value between Digg number and STD of latency. Indeed, 89% of URLs have correlation value

above 0.4. Meanwhile, 50% of them had a correlation value between value 0.7 to 1 which indicates very strong correlation.

### III. FORECASTING LATENCY

Although we are able to identify that volatile response times can be directly attributed to the increase in popularity for a significant portion of the external URLs, it is not clear whether we can forecast either the Digg number or the spike in latency using solely network measurements. An important factor for the prediction is timing. The detection time before latency becomes large could be an important factor for our prediction. If the detection time of latency is too short, we are not able to detect the trend.

An algorithm is set up to measure the detectable URL based on whether a detection time exists. The detection time is set at the point that its latency has reached 90% of its highest spike. Due to the reason that some latency samples could be random and abnormal, we collected latency within a time window and computed the average within the time period. If the average of latency in each window is slightly increasing, we consider the URL as detectable. Window size was set up as 5, 10, 15 and 20 times captures of latency. Based on the algorithm above, we have found the highest percentage of URLs that we could predict ahead of time is merely 2.4% of all the URLs when window size is 10. That means by relying purely on network latency measurements, we do not have enough reaction time.

To address this limitation, we decided to use forecasting based on the social discussion boards content. We extracted features about early user comments using the Digg API along with Digg number. This approach provides us with early information in order to study Digg number trends. However, to compute the earliest prediction time, we first have to know the general trends in Digg number growth for both upcoming stories and popular stories. To achieve this, the Digg numbers of the top 1000 stories are captured every half hour, and we continue to update these stories. Therefore, if new stories approach the top 1000, we will add them and start to record their Digg numbers. Meanwhile, we also keep updating the prior stories until they have been removed from the Digg website. Hence, the duration of each story is different.

For an early warning system to work in the case of predicting flash crowds, we would like to arrive at the prediction results as early as possible. The earlier the prediction results are available, more time is available for the administrators to react to the flash crowd. Estimating the time required for a prediction result is an important aspect to our proposed frame work. To this end, we divide our task into two mutually exclusive requirements. Firstly, estimating the network latency of web resources (stories) posted on Digg. Secondly, predicting the popularity of Digg stories by mining social network characteristics of Digg. Figure 3 shows the two prediction mechanisms used to validate our results.

Digg provides us with an extensive and a convenient

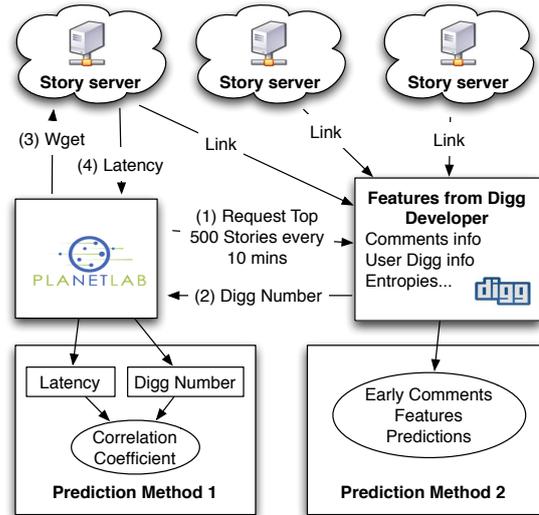


Figure 3. Digg Number Prediction Architecture. Two measurements are implemented: (1) prediction based on correlation between latency and Digg number of stories and (2) prediction based on early comments features about Digg information about users and stories

API to interact with its website. We make use of this API to obtain the latest set of popular stories. Using this mechanism we collect and store each popular URL. A set of 500 URLs and its features across all the topics available in Digg are downloaded repeatedly every ten minutes over time. The data collected is then used by the two prediction techniques to independently predict the popularity. The following subsections describe the two prediction techniques:

#### A. Prediction Methods

To achieve the earliest prediction time, we began our experiment at the time that the story was submitted. We captured the test data at different time after the story was submitted. The training data was captured within the whole time length (which is 120 hours), while testing data was captured in 5, 10, 15, 30, 60, 120, 300, 600, 900 minutes and complete time (which is 120 hours) since the story was submitted. In addition, for our prediction, we used features related to comment statistics, user feedback and community structure and membership.

The number of comments for a posted story and the average word length of the comments are used as features. We also used the level associated with each comment, extracted from the Digg comment structure. Digg users have the option to comment whether they like a story or not, but they can also rate the comments. We used as features the positive and negative feedback from the users. Based on user interest in Digg categories (World Business, Technology, Science, Gaming, Sports, Entertainment, Life Style and Offbeat), we defined an entropy metric that assessed the scope and knowledge of user comments. We also computed entropy across 51 sub-categories. More details of these features can be found in [12].

Overall, our approach leverages eight features to train

the prediction models, as does previous research [12]. We focus on the earliest time of predicting correct Digg number. Furthermore, the number of class for presenting Digg number of story was set up as three independent multi-classification group, that is 2-class, 4-class and 8-class. For each of the 2-class, 4-class and 8-class classifiers, the bins were set in Digg number intervals of 2750, 500 and 250 respectively. For example, for 2-class problem, the bins were set as below 2750 and above 2750. On the other hand, for 8-class problem, each bin represents the Digg number with intervals of 250. The more class it has, the more difficult it takes to predict. The classification performance was evaluated as K-way classification accuracy ( $Q_K$ ) and K represents the number of class in each prediction group. In addition, we also use the area under the receive operating characteristics curve (ROC) [5] to observe the average area under the plot of true positive rate versus the false positive rate.

In this study we used various classification techniques. Firstly, we used the C4.5 decision tree [27] and Nine Nearest Neighbor Classifier [1]. For the support vector machine classifiers, we applied linear and radial basis kernel function. For the K-class classification in SVMs, we trained as one-versus-rest classifiers for each of K classes. Ensemble of classifiers have been known to outperform individual classifiers. Therefore, we use AdaBoost [8] a meta algorithm that trains successive classifiers with an emphasis on previously misclassified instances. Additionally, we also test the prediction by MultiBoost [28] also a meta algorithm, and an extension to AdaBoost algorithm. Finally, We used Classification Via Regression (CVR) [7] by applying a type of decision tree with linear regression functions at the leaves that generates more accurate classifiers. To have better performance of the classification, we performed 5-fold cross validation. ‘‘Weka’’ Toolkit [29] and LibSVM [6] are major tools for the popularity prediction.

### B. Prediction Results

Of the seven classification algorithms, we present the four methods in Figure 4 that have the best classification performance. Figure 4 also illustrates how  $Q_2$ ,  $Q_4$ ,  $Q_8$  accuracy change as time for collecting test data increases. Meanwhile, the vertical line at each point represents confidence interval fluctuation range. The confidential interval range are relatively small, that means the reliability of the accuracy is in good level. As time increase, all three classification accuracy increase relatively.

SVM has superior performance for most of the cases. Given by the data in Figure 4, SVM linear regression method already reaches 86% accuracy in 2-class classification at the first five minutes. Meanwhile, SVM Radial Basis kernel Function (RBF) gets 62% and 54% for 4-class and 8-class classification. In addition, for the whole data sets, the best performance we can reach is 95% accuracy in 2-class classification by both SVM algorithms. Other than that, SVM RBF also has the best accuracy (61%) for  $Q_8$  result, while CVR has the best  $Q_4$  result, that it reaches

73% for whole time period. In general, SVM methods have better performance of accuracy than ensemble methods for most time length of collecting test data.

## IV. RELATED WORK

Network traffic prediction has been a topic that received significant attention during the last decade [14], [3], [19]. Li [16] *et al.* proposed a method to identify network anomalies using sketch subspaces. Their work required a lot of historical data that is not feasible to obtain for externally linked websites to social discussion boards. The same hypothesis of access to historical trends holds for the work by Sengar *et al.* [22] and Fu-Ke *et al.* [9].

Flash crowds are defined as the phenomenon where there is an acute increase in the volume of network traffic and are difficult predict. The flash crowd effect has been referred to with different names such as hot-spot and slashdot effect [10]. Most of the previous research attempts to provide a reactive approach to solving the flash crowd problem by proposing replication of services [23]. Jung *et al.* [13] attempt to prevent flash crowds by blocking requests from malicious clients. They achieve this by distinguishing between the characteristics of a flash crowd and a DDoS attack. Baryshnikov *et al.* [4] argue that it is possible to design a framework to predict flash crowds in web traffic.

In terms of social network popularity prediction, the work of Szabo *et al.* [24] introduced a regression model to predict the popularity of posts on the Digg network using the popularity ratings at an earlier time interval. In contrast, the method introduced here predicts the popularity of posts using different features, and models user participation explicitly in the form of comments. Another document recommendation model [15] captures users reading certain posts and explicit relationships between friends. Jamali *et al.* [12] have shown that it is possible to predict the popularity of a story by mining Digg. However, their approach was geared towards long term analysis and requires up to ten hours of historical data to obtain a prediction result. We present results that start the evaluation as early as 2 minutes. Another recent thread of research focused on collective behavior prediction using extraction of the social dimension [26], [25].

## V. CONCLUSIONS

Our initial goal was to quantify the effects of social discussion boards on popular externally linked stories and websites in terms of network response time. To that end, we measured the download times of the websites hosting popular stories for 1289 distinct URLs over a period of two weeks. By correlating the variation of the measured latency with the increase in popularity, we were able to show that the network response times of 89% of the popular URLs were affected. This includes over 50% of the stories having correlation values greater than 0.7.

Furthermore, using features extracted from the content and structure of the social discussions, we were able to successfully classify as popular approximately 86% of the

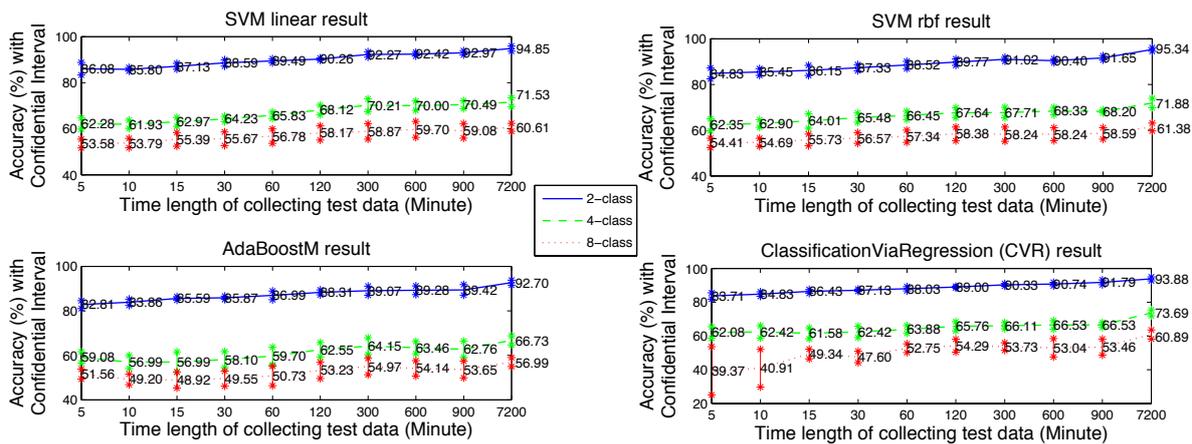


Figure 4. Multi-classification( $Q_2$ ,  $Q_4$ ,  $Q_8$ ) Accuracy With Confidence Interval (Represented in Vertical Line) for (a) SVM Linear and Radial Basis Function Regression Methods; (b) Ensemble AdaBoostM1 and Classification Via Regression Methods

stories within just five minutes of their submission. This number further increases to 95% when we collect five hours of online discussions. Our study shows that there is clear benefit in using information derived from social activities to predict potentially abrupt increase in demand that can cause delays or become debilitating for the underlying network infrastructure. We will consider more discussion boards rather than Digg because the same story could be published at the same time at other discussion boards. In the future, We will design a system to predict network response time by using our correlation results.

## REFERENCES

- [1] David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, January 1991.
- [2] Noor Ali-Hasan and Lada A. Adamic. Expressing social relationships on the blog through links and comments. In *International Conference on Weblogs and Social Media (ICWSM)*, 2007.
- [3] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 71–82. ACM, 2002.
- [4] Yuliy Baryshnikov, Ed Coffman, Guillaume Pierre, Dan Rubenstein, Mark Squillante, and Teddy Yimwadsana. Predictability of web-server traffic congestion. In *Proceedings of the 10th International Workshop on Web Content Caching and Distribution*, pages 97–103, Washington, DC, USA, 2005. IEEE Computer Society.
- [5] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.
- [6] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [7] Eibe Frank, Yong Wang, Stuart Inglis, Geoffrey Holmes, and Ian H. Witten. Using model trees for classification. *Mach. Learn.*, 32:63–76, July 1998.
- [8] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55:119–139, August 1997.
- [9] S. Fu-Ke, Z. Wei, and C. Pan. An Engineering Approach to Prediction of Network Traffic Based on Time-Series Model. In *Artificial Intelligence, 2009. ICAI'09. International Joint Conference on*, pages 432–435. IEEE, 2009.
- [10] A.M.C. Halavais. *The Slashdot Effect: Analysis of a large-scale public conversation on the world wide web*. University of Washington, 2001.
- [11] Steve Huffman and Alexis Ohanian. reddit. <http://www.reddit.com/>.
- [12] Salman Jamali and Huzefa Rangwala. Digging digg : Comment mining, popularity prediction, and social network analysis. In *WISM'09-AICI'09*, pages 6–6. Shanghai University of Electric Power, Shanghai, China, 2009. EI Compendex and ISTP.
- [13] Jaeyeon Jung, Balachander Krishnamurthy, and Michael Rabinovich. Flash crowds and denial of service attacks: characterization and implications for cdns and web sites. In *Proceedings of the 11th international conference on World Wide Web, WWW '02*, pages 293–304, New York, NY, USA, 2002. ACM.
- [14] A. Lakhina, M. Crovella, and C. Diot. Characterization of network-wide anomalies in traffic flows. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 201–206. ACM, 2004.
- [15] Kristina Lerman. Social information processing in news aggregation. *IEEE Internet Computing*, 11(6):16–28, 2007.
- [16] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identification of network anomalies using sketch subspaces. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 147–152. ACM, 2006.
- [17] Gilad Mishne and Natalie Glance. Leave a reply: An analysis of weblog comments. In *In Third annual workshop on the Weblogging ecosystem*, 2006.
- [18] H. Niksic. GNU wget, 1996.
- [19] K. Papagiannaki, N. Taft, Z.L. Zhang, and C. Diot. Long-term forecasting of Internet backbone traffic. *Neural Networks, IEEE Transactions on*, 16(5):1110–1124, 2005.
- [20] Kevin Rose. Digg. <http://digg.com/news>.
- [21] Joshua Schachter. Delicious. <http://www.delicious.com/>.
- [22] H. Sengar, X. Wang, H. Wang, D. Wijesekera, and S. Jajodia. Online detection of network traffic anomalies using behavioral distance. In *Quality of Service, 2009. IWQoS. 17th International Workshop on*, pages 1–9. IEEE, 2009.
- [23] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. Steen. Replication for web hosting systems. *ACM Computing Surveys (CSUR)*, 36(3):291–334, 2004.
- [24] G. Szabo and B. Huberman. Predicting the popularity of online content. *Technical Report HP Labs*, pages 1–6, 2008.
- [25] L. Tang and H. Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009.
- [26] L. Tang and H. Liu. Toward collective behavior prediction via social dimension extraction. *IEEE Intelligent Systems*, 2010.
- [27] Geoffrey Webb. Decision tree grafting. In *In IJCAI-97: Fifteen International Joint Conference on Artificial Intelligence*, pages 846–851. Morgan Kaufmann, 1997.
- [28] Geoffrey I. Webb. Multiboosting: A technique for combining boosting and wagging. *Mach. Learn.*, 40:159–196, August 2000.
- [29] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005.
- [30] Kou Zhongbao and Zhang Changshui. Reply networks on a bulletin board system. *Phys. Rev. E*, 67(3):036117, Mar 2003.