Department of Computer Science
George Mason University Technical Reports

4400 University Drive MS#4A5
Fairfax, VA 22030-4444 USA
http://cs.gmu.edu/ 703-993-1530

# Predicting Network Response Times Using Social Information

**Chen Liang**
cliang1@gmu.edu

**Sharath Hiremagalore**
shiremag@gmu.edu

**Angelos Stavrou**
astavrou@gmu.edu

**Huzefa Rangwala**
rangwala@cs.gmu.edu

Technical Report GMU-CS-TR-2010-18

## 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 what people call the "Flash Crowds" effect. Flash Crowds present a real challenge, as it is not possible to predict their intensity and occurrence time. Moreover, although increasingly capable, most present-day web hosting servers and caching systems are designed to handle a nominal load of requests before they become unresponsive. This can happen either due to the limited bandwidth or processing power allocated to the hosting site.

In this paper, we quantify the prevalence of flash crowd events for a popular social discussion board (Digg). Using PlanetLab, 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. In an effort to identify flash crowds ahead of time, we evaluate and compare traffic forecasting mechanisms. We show that predicting network traffic using network measurements has very limited success and cannot be used for large-scale prediction. However, by analyzing the content and structure of the social discussions, we were able to classify 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.

## 1 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 [23], Reddit [11], Delicious [24] 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 [34], relationships [20, 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.

Ordinarily it would be possible to forecast the load on the server by observing the trends in network latency over a period of time. However, we show that the impulse in the network traffic caused by flash crowds is difficult to predict using network measurements and, therefore, allows little time to take any preventive action, such as temporarily adding capacity or rate limiting the existing users. However, by analyzing the content and structure of the social discussions, we were able to classify 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.

For our experiments we use Digg [23], a popular social bookmarking tool. Digg allows users to share news, images and videos as stories among themselves. Early readers information helps us to predict the popularity of a story. When a top story is prominently displayed in Digg, more users read it. Stories are categorized into topics, some being more popular than others. Relationships between users in Digg are implicitly created when users read, comment or rate each other's stories. Digg associates a Digg Number with each story to keep track of the number of users who like or dislike it. Using these factors, as well as information about the users, we are able to predict the popularity of a given story. Our work demonstrates that we can effectively leverage measurements of social activity to forecast network events that will be otherwise infeasible to measure or respond to relying only on network measurements.

The rest of this paper is organized as follows. Section 2 provides the problem motivation and the traffic correlation between for external URLs using a distributed measurement infrastructure. Section 3 provides prediction results using social features. We present our experimental results on real Digg story data and show that we can predict the popularity of stories using historical story data both accurately and quickly. In Section 4 we detail the related work. We conclude the paper in Section 5 with some discussion on the potential to drive network characteristics based on data mining of social events and our plans for future work.

# 2 Correlating Popularity with Response Time

## 2.1 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.

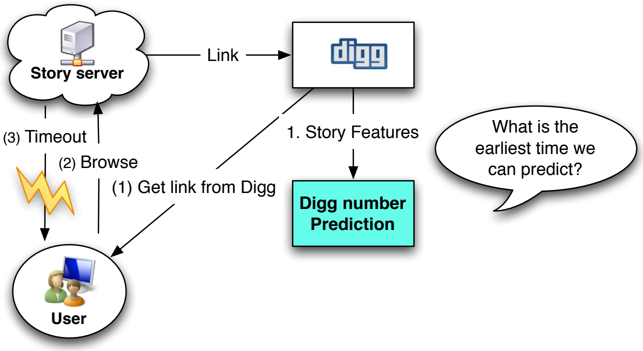


Figure 1: This figure illustrates the effects of a "Flash Crowd" event. The popularity of the social discussion board causes the externally linked website to become slow or even unresponsive.

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

## 2.2 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 it was required to download their content to the Planetlab nodes. To achieve that, we employed wget [21], 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 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 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. In addition, to prevent naturally slow websites from being repeatedly probed, we setup a maximum probing window. This window enabled us to treat subsequent measurements based on the historical trends that the site has exhibited. We initially computed the maximum interval between every two adjacent measurements that we captured. Then, we set the time window at 1, 2, 4, and 8 times the maximum interval, allowing this site more time to transmit data. 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. However, as we discuss next, to correlate with the Digg score, we used a uniform representation of time where subsequent measurements were averaged.

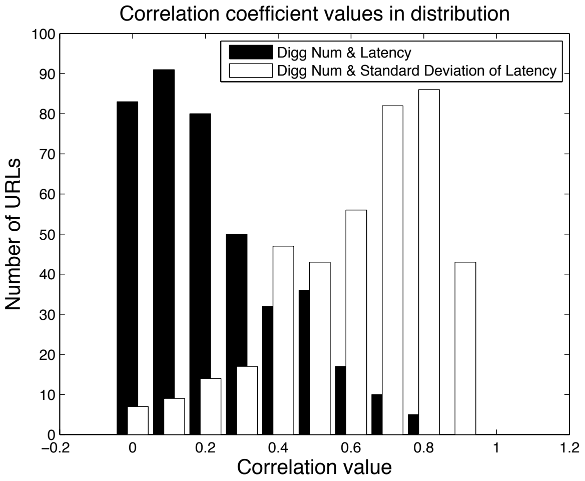


Figure 2: Comparison Between Two Correlation Coefficient Methods: 1. Correlation Value Between Digg Number and Latency; 2. Correlation Value Between Digg Number and Standard Deviation of Latency (STD of Latency)

## 2.3 Correlation Methodology

We used the 2-D Correlation Coefficient tool in MATLAB. The algorithm, which computes the correlation coefficient, is:

$$r = \frac{\sum_{k=1}^{m}\sum_{l=1}^{n}(A_{mn}-\bar{A})(B_{mn}-\bar{B})}{\sqrt{(\sum_{k=1}^{m}\sum_{l=1}^{n}(A_{mn}-\bar{A})^2)}} \times \frac{1}{\sqrt{(\sum_{k=1}^{m}\sum_{l=1}^{n}(B_{mn}-\bar{B})^2)}} \tag{1}$$

(with $\bar{A}$ and $\bar{B}$ being the means of matrix elements A and B).

This algorithm originates from Pearson's product-moment correlation coefficient, which correlates only if variables have a linear dependence upon A and B. The correlation coefficient ranges from -1 to 1. A value of 1 implies that a linear equation describes the relationship between A and B perfectly, with all data points lying on a line for which B increases as A increases. A value of -1 implies that all data points lie on a line for which A decreases as B increases. In our experiments, we expected the latency increases as Digg number increases. Hence, we only considered correlation value closes to 1 as good performance.

## 2.4 Correlating Popularity to Latency

We deployed our code on Planetlab nodes and we tracked the Digg number and response times through downloading content and measuring the resulting network latency until the completion for a series of URLs

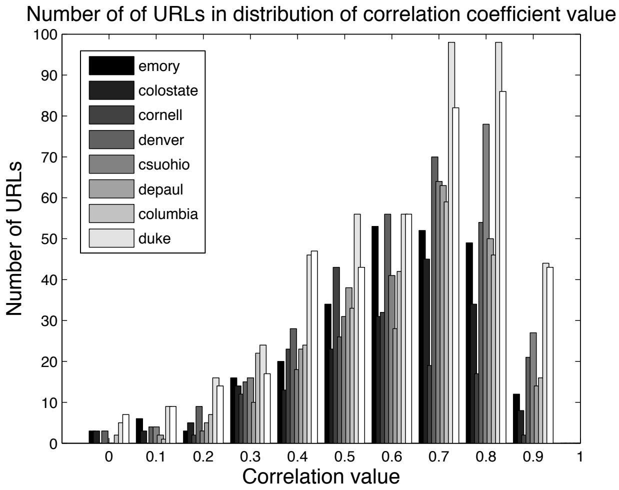


Figure 3: Comparison of Correlation Coefficient Between Eight Nodes from US-based nodes

linked in the social discussion board. These numbers represent averages over a lot of measurements. Initially, we tried to directly correlate increases in perceived network latency for the download of URLs with the increase in popularity.

All of our results indicated that latency and Digg number are not directly correlated. As Digg number increases, the increasing variable is not the value of latency, but rather the variation in the measured response times and perceived latency. 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 of 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.

Figure 2 depicts the comparison between the two correlation results. It appears that Digg number has better correlation with the STD of latency rather than pure latency. This is primarily due to the fact that the sites do not become completely unresponsive but their responses exhibit heavy fluctuations clearly correlated to the increase of popularity. Although helpful in understanding the trends, the results from the two sites were not sufficient to achieve statistical significance and they were limited to two nodes. Next, we present results that are more representative.

## 2.5 Correlation Value Among PlanetLab Nodes

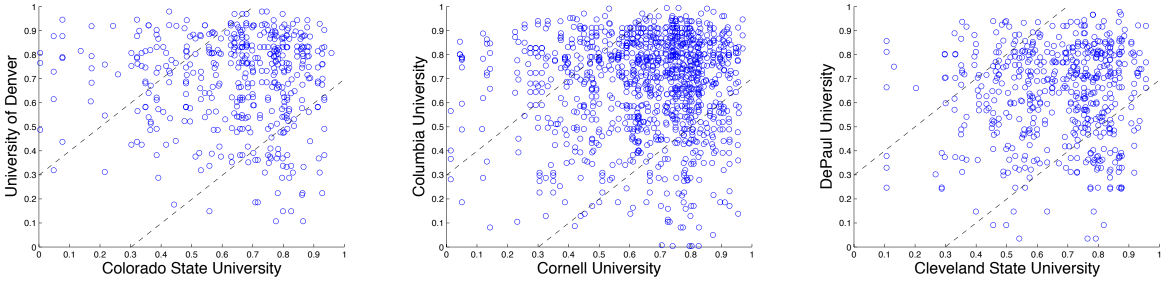
We generated results from eight US-based nodes and displayed them using distribution format in Figure 3. This figure shows the number of URLs plotted according to the correlation value between Digg number and STD of latency. Indeed, 89% of URLs have correlation value above 0.4. Meanwhile, 50% of them had a correlation value between value 0.7 to 1 which indicates very strong correlation. Moreover, to verify that this is not a localized phenomenon and that our measurements are not biased, we randomly chose three pairs of nodes from the same areas in America: Central Area, East Coast Area and Five Lake Area. Because all of the nodes were tested at the same time, they shared the same stories provided by the Digg website. We captured the CC value of these nodes and clustered the results in Figures 4(a), 4(b) and 4(c).

Two diagonal lines boundaries the areas that is tolerable for similarity of correlation values. The more points gathering within the lines, the more related the results are. All graphs show that most points gathered at top right area around [0.5, 1] for both axes. Each node has large percentage of high correlation URLs shown in figure 3, but it's good to see most high correlation value URLs are between the horizon lines that means both nodes have similar correlation value for the same URL at the same time of measurement.

There are some points locating outside the diagonal line area; this is indicative of URLs with high correlation value in one node, but low correlation value in the other node. Above all, it's generally reasonable for two nodes that are close to each other and have high percentage of similar correlation values of the same measurements. It proves that latency correlates with Digg number well regionally.

## 2.6 Geo-locating Unresponsive Servers

Latency varies for different reasons, such as network connections, user locations, and story servers. To assess the geo-location of the unresponsive servers, we analyzed the countries where the websites of the externally linked stories or URLs point to. The Geo-map Figure 5 shows the unresponsive rates for each country. In the areas with light yellow colors, are those countries that do not host any of the story servers obtained from Digg in our study. The area with shades of green colors denotes countries hosting the story servers. As the color goes dark green, the country has more unresponsive stories. It's easy to observe that the United States has the most unresponsive stories totaling 204 of the 221 observed unresponsive stories. This followed by United Kingdom, Ireland, Germany, Canada, Australian and Netherlands.



(a)

(b)

(c)

Figure 4: Scatter Plot of Correlation Value Between Two Nodes From (a) Colorado State University and University of Denver; (b) Cornell University and Columbia University; (c) Cleveland State University and DePaul University

# 3 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 is existed. The detection time is set at the point that its latency has reached 90% of a spike. Meanwhile, a window is set as 5, 10, 15 times of capture of latency and the latency value is computed as average of the 10 captures. If the average of latency in each window is slightly increasing, we consider the URL as detectable. 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.

Using network measurements alone, the number of URLs that we can predict their increase in terms of network latency is very limited. That is primarily due to the fact that latency usually increases suddenly after the increase in popularity. Meanwhile, the diversity and regionality of correlation value shows the latency could be pretty different based on different location of PlanetLab node.

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 6 shows the two prediction mechanisms used to validate our results.

Digg provides us with an extensive and a convenient API to interact with its website. We make use of this API to obtain a latest set of popular stories posted. Using this mechanism we collect and store each stories URL. A set of 500 story 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 to predict the popularity. The following subsections describe the two prediction techniques:

The life span of each story could be quite different. If a story is just added to the group then it will have very few Digg numbers, or if a story has been "digged" for a long time before addition to our system then it will already have had a high Digg number at the beginning hours of the evaluation. We therefore focus only on the story that has been submitted in a relative equal length of time. We set up a time length, which spans from the submission time to the current time, approximately 40 to 75 hours. We separate the stories based on their Digg Number into five categories ranging from 0-50, 51-100,

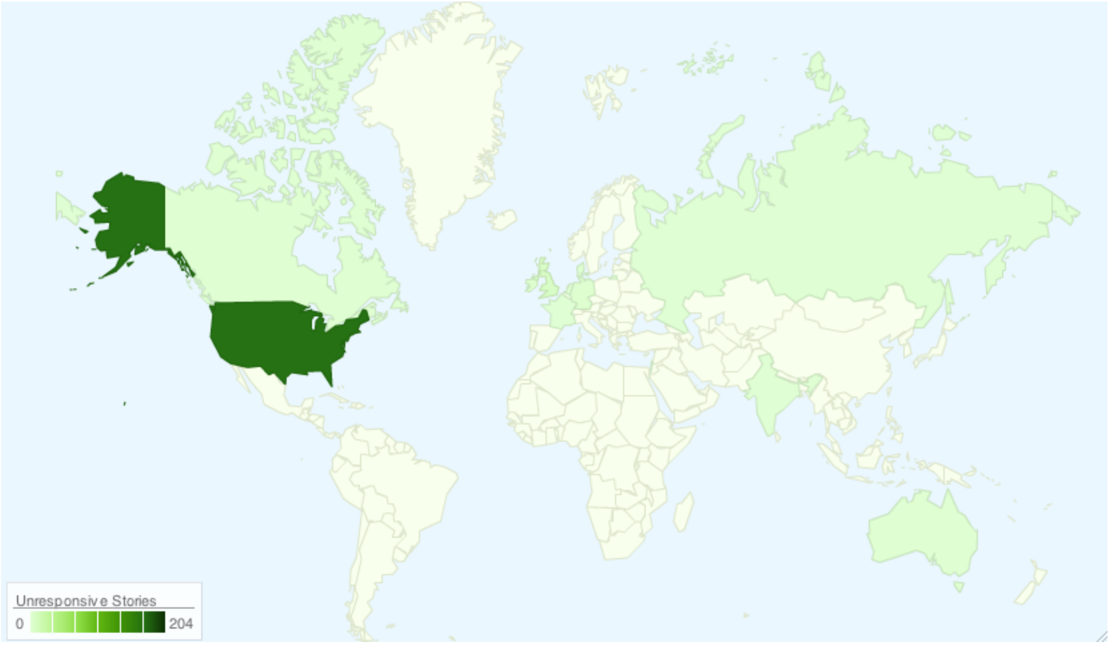


Figure 5: A Geomap of the hosting country for unresponsive websites our of the 1289 popular stories that appeared in Digg over a period of two weeks. Darker shades of green indicates higher density.

101-200, 201-500 and 501-1000. Since the Digg number of an upcoming story is relatively small, its threshold will be set only to 50, 100, and 200. Most popular stories have at least 50 Digg numbers at the time that their status become popular. Figure 7(a) shows the percentage of URLs with a Digg number range between 50, 100, 200, 500, and 1000. The total number of URLs is decreasing because we only focus on stories with a time length within the 40-75 hours range. We can therefore rule out stories that are either newly submitted or have been posted for a long time. As finished stories are removed from the Digg website, the total number of stories that we are tracking is reduced. When a story is originally submitted, it is categorized as upcoming story. Figure 7(b) illustrates the trend of Digg number in upcoming stories. Within the 10 hours of our experiment, only 60 percent the upcoming stories have a Digg number higher than 50. Of this 60 percent of stories, only 24 percent have a Digg number above 100.

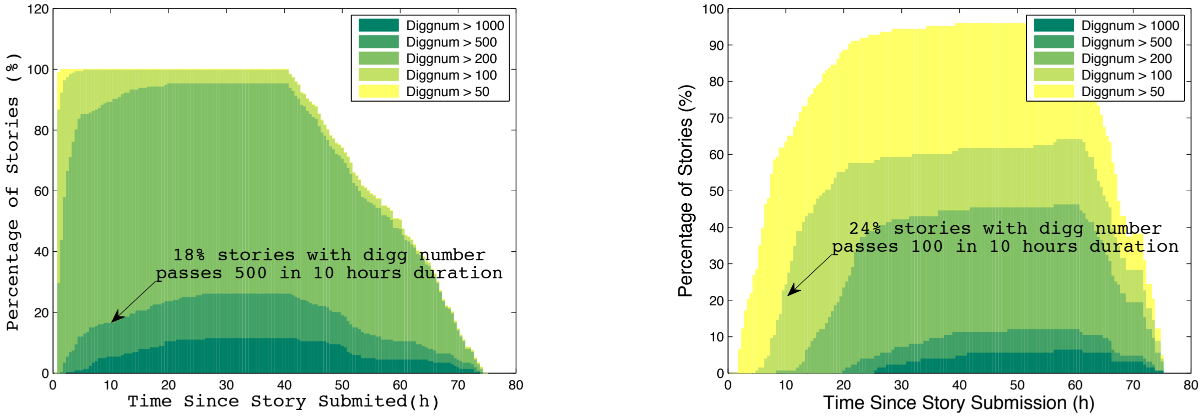
On the other hand, the popular stories increase relatively faster than the upcoming ones. When an upcoming story reaches a Digg number higher than 100, its status changes to popular. Therefore, by the time it becomes popular, its Digg number is already above 100. Figure 7(a) shows the Digg number trends of popular stories. For the first 10 hours of our experiment, approximately 98 percent of the stories already had Digg number higher than 100; 90 percent of them had Digg number above 200. It is therefore too late to predict the trends of popular stories once they have reached a high level of popularity. We need to predict the trends of stories that are still upcoming or recently submitted by a user.

## 3.1 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 the following features:

- ***Comment Statistics*** The number of comments for a posted story and the average word length of the comments are used as features. The Digg comment system forms a hierarchical tree structure where each comment is associated with its own level. A comment made directly to the story is considered a first-level comment. We count the number of comments associated with top 4 levels for each story.

- ***Digg User Feedback*** Digg users have the option to comment whether they like a story or not, but they can also rate the comments. The Digg website provides the “up” and “down” scores, which represent positive and negative feedback from users. We use the sum of up scores and sum of down scores for all comments associated with the story as two features.



(a)

(b)

Figure 7: Percentage of Stories with Digg number of bigger than 50, 100, 200, 500 and 1000 and their popularity over time. Stories in (a) Popular Queue (b) Upcoming Queue. Popular queue contains stories that are already above 100 in popularity. Notice that stories become popular not immediately but after few hours. This gives us time to analyze their content and forecast their popularity and network response fluctuations.

- ***User Community Structure and Membership*** We used two entropy values computed for categories and topics as a measure knowledge associated with users. The entropy is defined as $H_1 = -\sum_i p_i log(p_i)$. With respect to the categories, $i$ iterates over eight Digg categories (World Business, Technology, Science, Gaming, Sports, Entertainment, Life Style and Offbeat) and $p_i$ denotes the probability for the user to belong to category $i$. On the other hand, with respect to the topics, the entropy is defined as $i$ iterates over 51 topics with $p_i$ denoting the probability for the user to belong to topic $i$.

Overall, our approach leverages eight features to train the prediction models, as does previous research. 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 14-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 [31] 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 [32] 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 [33] and LibSVM [6] are major tools for the popularity prediction.

## 3.2 Prediction Results

Of the seven classification algorithms, we presented the four methods in Figure 8 that have the best classification performance. Figure 8 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 [13] fluctuation range. The confidential interval range are relatively small, that means the reliability of the accuracy is in good level. Except that 8-class CVR result for the first 10 minutes is not desired, which is mostly due to its low accuracy. As time increase, all three classification accuracy increase relatively.

SVM has superior performance for most of the cases. Given by the data in Figure 8, SVM linear regression method already reaches 86% accuracy in 2-class classifi-

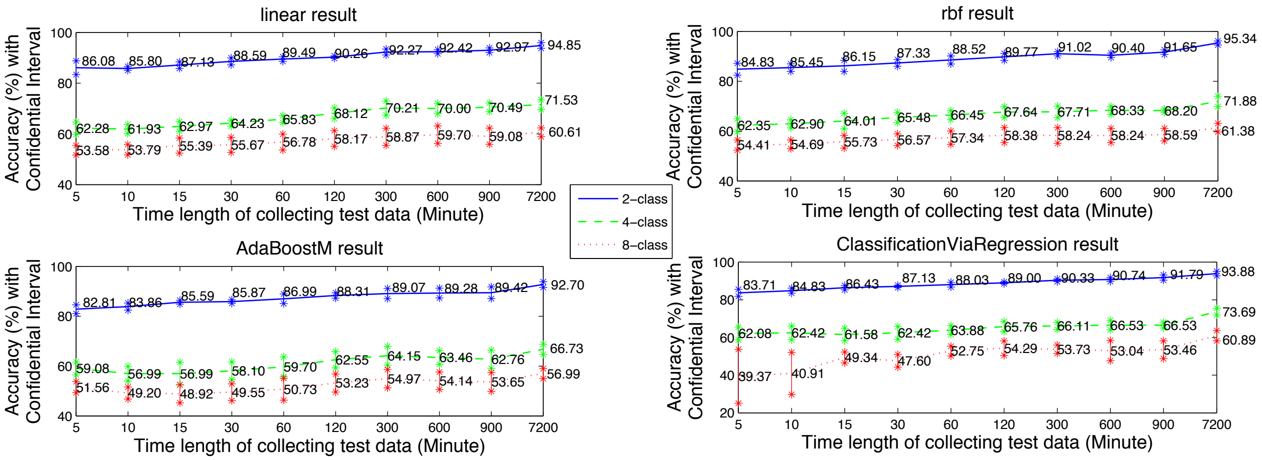


Figure 8: Multi-classification($Q_2$, $Q_4$, $Q_8$) Accuracy With Confidence Interval (Represented in Vertical Line) for (a) SVM Linear and Radial Basis Function Regression Methods; (b) Ensemble AdaBoostM1 and Classification Via Regression Methods

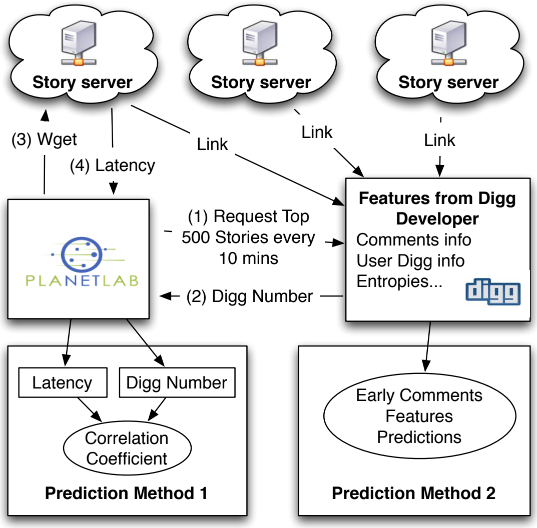


Figure 6: Digg Number Prediction Architecture. Two measurements are implemented: (1) prediction based on correlation between latency and Digg number of stories and (2) prediction based on early comments features about Digg information about users and stories

cation 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.

The ROC results for the four methods are presented in Figure 9. We can see that CVR methods have superior ROC results than others, except for its unstable results for the first 10 minutes in 8-class classification. For the first five minutes, even if SVM have the highest accuracy for 2-class, 4-class and 8-class methods, the CVR method in ensemble has much higher ROC value in 2-class and 4-class classification and its accuracy is the same level of SVM or a little bit less than SVM. Same as the complete hours (120 hours) that CVR have the highest ROC for all three multi-classification and its accuracy results are also the top level. Hence we can see that CVR method are actually more reliable than other methods.

Summarizing from results above, given enough time, the best accuracy we can get is 95% by SVM RBF method in $Q_2$. Moreover, the shortest time that we are able to predict the $Q_2$ accuracy that is good enough (above 85% accuracy and 0.8 ROC) is first 15 minutes with ensemble CVR algorithm.

## 4 Related Work

Network traffic prediction has been a topic that received significant attention during the last decade [16, 3, 22]. Li [19] *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.* [25] 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

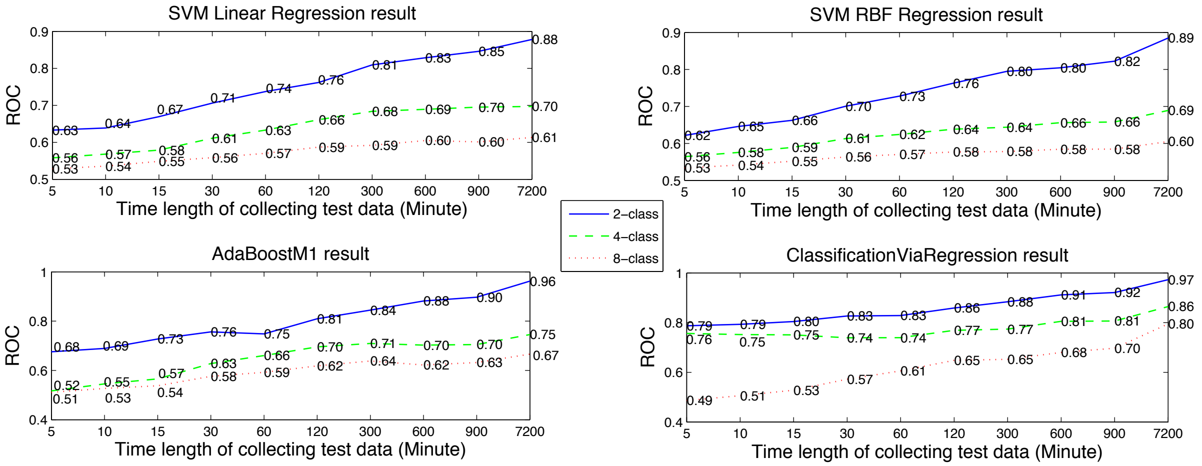


Figure 9: Multi-classification(2-class, 4-class and 8-class) Results About the Area Under the Receiver Operating Characteristics Curve (ROC) for SVM Methods: Linear and Radial Basis Function Regression and Ensemble Methods: AdaBoostM1 and Classification Via Regression

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 [26]. Jung *et al.* [14] 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. Furthermore, they discuss techniques to predict the flash crowds using network load statistics. The above work does not look into the advantage of using social networks and data mining techniques to predict traffic volatility.

In terms of social network popularity prediction, the work of Szabo *et al.* [27] 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 [17] 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 [29, 28]. In addition, Kunegis *et al.* [15] proposed a mechanism to mine social networks with negative edges. Finally, Tsagkias *et al.* [30] presented techniques to predict the volume of the online news stories while Lerman *et al.* [18] focused on the analysis of the social voting patterns for Digg. Analyzing collective behavior and voting patterns can potentially assist to predict future story popularity for social discussion boards and news sites. We do not, however, make use of them in this paper.

## 5 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, knowing that there is a direct correlation between popularity and network response times, we tried to investigate mechanisms to forecast the popularity of the externally hosted stories. Indeed, using features extracted from the content and structure of the social discussions, we were able to successfully classify as popular approximately 86% of the 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 can cause delays or become debilitating for the underlying network infrastructure.

# References

[1] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, January 1991.

[2] N. Ali-Hasan and L. 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 measurment*, pages 71–82. ACM, 2002.

[4] Y. Baryshnikov, E. Coffman, G. Pierre, D. Rubenstein, M. Squillante, and T. 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] A. 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] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] E. Frank, Y. Wang, S. Inglis, G. Holmes, and I. H. Witten. Using model trees for classification. *Mach. Learn.*, 32:63–76, July 1998.

[8] Y. Freund and R. 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. JCAI'09. International Joint Conference on*, pages 432–435. IEEE, 2009.

[10] A. Halavais. *The Slashdot Effect: Analysis of a large-scale public conversation on the world wide web*. University of Washington, 2001.

[11] S. Huffman and A. Ohanian. reddit. `http://www.reddit.com/`.

[12] S. Jamali and H. Rangwala. Digging digg : Comment mining, popularity prediction, and social network analysis. In *WISM'09-AICI'09*, pages 6–6, Shanghai University of Electic Power, Shanghai, China, 2009. EI Compendex and ISTP.

[13] V. J.Easton and J. H. McColl. confidence interval. In *Statistics Glossary v1.1*, 1997.

[14] J. Jung, B. Krishnamurthy, and M. 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.

[15] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The Slashdot Zoo: Mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750. ACM, 2009.

[16] 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.

[17] K. Lerman. Social information processing in news aggregation. *IEEE Internet Computing*, 11(6):16–28, 2007.

[18] K. Lerman and A. Galstyan. Analysis of social voting patterns on digg. In *Proceedings of the first workshop on Online social networks*, pages 7–12. ACM, 2008.

[19] 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.

[20] G. Mishne and N. Glance. Leave a reply: An analysis of weblog comments. In *In Third annual workshop on the Weblogging ecosystem*, 2006.

[21] H. Niksic. GNU wget, 1996.

[22] K. Papagiannaki, N. Taft, Z. Zhang, and C. Diot. Long-term forecasting of Internet backbone traffic. *Neural Networks, IEEE Transactions on*, 16(5):1110–1124, 2005.

[23] K. Rose. Digg. `http://digg.com/news`.

[24] J. Schachter. Delicious. `http://www.delicious.com/`.

[25] 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.

[26] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. Steen. Replication for web hosting systems. *ACM Computing Surveys (CSUR)*, 36(3):291–334, 2004.

[27] G. Szabo and B. Huberman. Predicting the popularity of online content. *Technical Report HP Labs*, pages 1–6, 2008.

[28] 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.

[29] L. Tang and H. Liu. Toward collective behavior prediction via social dimension extraction. *IEEE Intelligent Systems*, 2010.

[30] M. Tsagkias, W. Weerkamp, and M. De Rijke. Predicting the volume of comments on online news stories. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1765–1768. ACM, 2009.

[31] G. Webb. Decision tree grafting. In *In IJCAI-97: Fifteen International Joint Conference on Artificial Intelligence*, pages 846–851. Morgan Kaufmann, 1997.

[32] G. I. Webb. Multiboosting: A technique for combining boosting and wagging. *Mach. Learn.*, 40:159–196, August 2000.

[33] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann Pub, 2005.

[34] K. Zhongbao and Z. Changshui. Reply networks on a bulletin board system. *Phys. Rev. E*, 67(3):036117, Mar 2003.