Efficient Content Delivery through Popularity Forecasting on Social Media

Irene Kilanioti
Department of Computer Science,
University of Cyprus,
Nicosia, CYPRUS, 2109
Email: ekoila01@cs.ucy.ac.cy

George A. Papadopoulos
Department of Computer Science,
University of Cyprus,
Nicosia, CYPRUS, 2109
Email: george@cs.ucy.ac.cy

***Abstract*—Ubiquitous social networks have in recent years become significant for sharing of content generated in online video platforms. Our work investigates how the predictability of video sharing is associated with the underlying social network of the initial sharer of the video and the context of the media platform it was uploaded. In particular we combine user-centric data from Twitter with video-centric data from YouTube to give insights that neither dataset (social network and media service dataset) individually gives. We propose a simple model to predict future popularity of a video resource with a small and easily extracted feature set, based on the notion of influence score of a user and its fluctuation through time, as well as the distance of content interests among users of both datasets. We further demonstrate how the incorporation of our prediction model into a mechanism for content delivery results in considerable improvement of the user experience.**

***Index Terms*—Analytics; Social Networks; Social Web; Social Cascade; Social Prediction; Regression Analysis; Video Popularity; Data Mining and Knowledge Extraction;**

1. Introduction

The diffusion of video content is fostered by the ease of producing online content via media services. It mainly happens via ubiquitous Online Social Networks (OSNs), where users increasingly repost links they have received from others (social cascades). If we knew beforehand when a social cascade will happen or to what range it will evolve, we could exploit this knowledge in various ways. For example, in the area of content delivery infrastructure, where popular items would be proactively replicated, so that bandwidth could be spared. Our work focuses on video virality over an OSN and combines detailed information both of the OSN and the media service with a small and easily extracted feature set. It proposes a prediction model that performs better than methods like Support Vector Machines (SVM), Stochastic Gradient Descent (SGD) and K-Nearest Neighbours (KNN), among others. We, furthermore, proceed to incorporate it into a mechanism for content delivery achieving substantial improvement for the user experience.

The remainder of this paper is organized as follows. Section 2 reviews previous related work. Section 3 formally describes the addressed problem. Section 4 provides an outline of the methodology, followed by the preparation of the employed datasets. Our main findings are presented in Section 5, where also a validation is conducted. Section 6 investigates the incorporation of the proposed model into a content delivery mechanism. Section 7 concludes the work and discusses directions for future work.

2. Related Work

The field of predicting social virality is active ([5], [12], [6], etc.). Studies focus on the prediction of the amount of aggregate activities (e.g. aggregate daily hashtag use [11]), on the prediction of user-level behaviour (like retransmission of a specific tweet/URL [12] or on the prediction of growth of the cascade size [5]. One branch of virality research is based on study of the evolution of cascades during a specific time-window ([11], [14]), whereas other works examine the cascades continuously over their entire duration [5].

Although our work focuses solely on video sharing, we identify the following methods for virality prediction in general, with our approach falling into the first category: Feature-based methods are based on content, temporal and other features, and the learning algorithms schemes they use are based on simple regression analysis ([5]), regression trees [2], content-based methods([14]), binary classification ([7], [8]), etc. They do not focus, though, on the underlying network infrastructure, and often encounter difficulty in extracting all the necessary features due to the large volume of accommodated graphs. Time-series analysis works ([15]), on the other hand, argue that patterns of a resource's growth of popularity are indicative of its future retransmissions.

3. Problem Description

We consider a directed graph $G(t) = (V(t), E(t))$ representing a social network that evolves through time, consisting at time t of V vertices and E edges. Edges between the nodes of the graph denote friendship in case of a social network (for Twitter B is a follower of A if there is an edge between B and A pointing at A.) We want to associate the

TABLE 1. NOTATION OVERVIEW

$G(t) = (V(t), E(t))$	graph G at time t of V vertices and E edges
A_{u2v}	number of actions where u influenced v
$\widehat{A_{u2v}}$	predicted output
M	total number of predicted values
α, β, γ	coefficients of feature set variables
U	vector of YouTube interests of user u
V	vector of Twitter interests of user v
Features Set	
$Score(u,t)$	Score of node u at time t
$dScore = dScore(u,t)/dt$	derivative of Score of node u at time t
$content\_dist$	content distance

number of retransmits of a video link by a user $v \in V$ after $u \in V$ has transmitted the link. User v is a follower of u.

We express this number, intuitively, as a combination of the following features: the $Score(u,t)$ of node u, $dScore(u,t)/dt$ of node u, and content distance between the content interests of the involved users both in the OSN and the media service. The validity of the predictors is analyzed in this paper. The intuition for their selection is based on the notion, that, the higher influence score a node depicts, the more influence it is expected to exert on other nodes of the social graph. Moreover, the $dScore/dt(u,t)$ expresses the popularity rise / fall of the node, and, lastly, the content distance associates the resource with the user context.

Denoting the output, the predicted output and the total number of predicted values by A_{u2v}, $\widehat{A_{u2v}}$ and M, we aim to find the values α, β, γ, so that:

$$A_{u2v} = \alpha \times Score(u,t) + \beta \times \frac{dScore(u,t)}{dt} + \gamma \times content\_dist \tag{1}$$

and

$$\sqrt{\frac{1}{M}\sum_{i=1}^{M}(\widehat{A_{u2v}} - A_{u2v})^2} \tag{2}$$

is minimum.

4. Proposed Methodology

4.1. Dataset

Interests of users were analyzed in [1] against directory information from http://wefollow.com, a website listing Twitter users for different topics, including Sports, Movies, News & Politics, Finance, Comedy, Science, Non-profits, Film, Sci-Fi/ Fantasy, Gaming, People, Travel, Autos, Music, Entertainment, Education, Howto, Pets, and Shows.

Twitter is one of the most popular OSNs centered around the idea of spreading information by word-of-mouth [13]. It provides mechanisms such as retweet, which enable users to propagate information across multiple hops in the network.

The activity of Twitter users was quantified, and a variety of features were extracted, such as the number of their tweets, the fraction of tweets that are retweets, the fraction of tweets containing URLs, etc. Aggregated features of YouTube videos shared by a user included in the dataset include the average view count, the median inter-event time between video upload and sharing, etc.

A sharing event in the dataset is defined as a tweet containing a valid YouTube video ID (with a category, Freebase topics and timestamp). We augmented the provided dataset with Tweet content information about the 15 million video sharing events included in the dataset, as well as information about the followers of the 87K Twitter users.

4.2. User Score Calculation

A user score is calculated combining the number n of its followers, reduced by a factor of 1000 to compensate the wide range of followers in the dataset from zero to more than a million, a quantity b catering for users with reciprocal followership, calculated by taking an average of number of a user's followers to the number of users he follows, as well as the effect e of a user's tweet, measured by multiplying average number of retweets with number of user's tweets and normalizing it to correspond to the total number of tweets. The distribution of these combined metrics depicts large variance and we have applied a logarithmic transformation in order to avoid the uneven leverage of extreme values.

$$Score = \log\left(n + \left(\left(\frac{b}{100}\right) \times n\right) + e\right) \tag{3}$$

4.3. Content Distance

The content distance $content\_dist$ expresses a measure of similarity of user's u YouTube and his follower's v Twitter interests. Content distance is calculated using cosine similarity between vectors of user's u YouTube and user's v Twitter video interests, as follows:

$$content\_dist = 1 - \frac{U \cdot V}{\|U\|\|V\|} \tag{4}$$

5. Experimental Evaluation

By combining user ids, followership information, user features and tweet context we build a measure of A_{u2v}, expressing the number of times a user's u tweet is retweeted by his followers v. We aim to associate the independent variables of the features set with the series depicting A_{u2v}.

The regression summary of Table 3 shows that coefficients of all predictors are significant ($P > |t|$ is significantly less than 0.05). Therefore, *Score*, *dScore* and $content\_dist$ can be considered as good predictors. We note that t here refers to $t - statistic$, denoting the quotient of the coefficient of dependent variable divided by coefficient's standard error. P refers to the $P - value$, a standard statistical method for testing an hypothesis. $P - value < 0.05$ means we can reject the hypothesis that the coefficient of the predictors is zero, in other words the examined coefficient are significant.

The selection of the above predictors comes as a result of comparing the $P - values$ of various metrics in the dataset

and the combination of those with the lowest $P-value$. The metrics included the number of distinct users retweeted, fraction of the user tweets that were retweeted, average number of friends of friends, average number of followers of friends, number of YouTube videos shared, the time the account was created, the number of views of a video, etc., among many others.

TABLE 2. REGRESSION RESULTS WITHOUT OUTLIERS (I)

Dep. Variable	A_{u2v}	R-squared	0.629
Model	OLS	Adj. R-squared	0.629
Method	Least Squares	F-statistic	3.072e+04
Prob (F-statistic)	0.00	Log-Likelihood	13947.
No. Observations	54473	AIC	-2.789e+04
Df Residuals	54470	BIC	-2.786e+04
Df Model	3	Covariance Type	nonrobust

TABLE 3. REGRESSION RESULTS WITHOUT OUTLIERS (II)

	coef	std err	t	$P>\|t\|$	95%	Conf.Int.
Score	0.1460	0.001	145.244	0.000	0.144	0.148
dScore	0.0200	0.001	25.819	0.000	0.018	0.022
con_dist	0.1656	0.003	65.690	0.000	0.161	0.171

Results of regression model on data obtained after removing outlier data points appear in Tables 2 and 3. We discover that *content_dist* feature plays an important role in video popularity prediction, suggesting high dependence of video sharing via Twitter on the video content itself. Fig. 1 plots depict an improved alignment of the path of regression line to the optimal path after the removal of outliers.

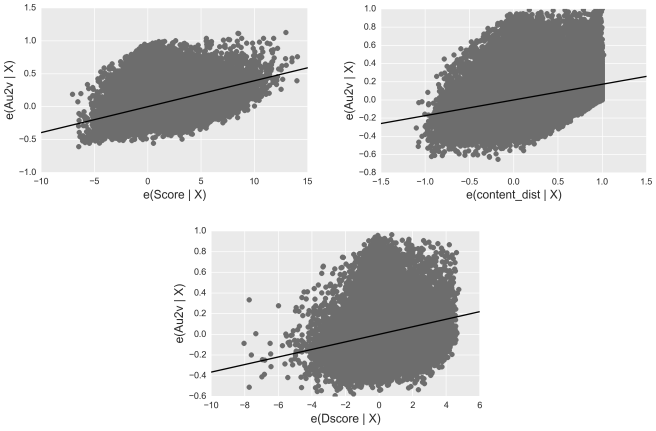

Figure 1. Regression plots for each independent variable.

5.1. 10-fold Cross-Validation

We performed a 10-fold cross validation on the dataset, fitting the regressor to 90% of the data and validating it on the rest 10% for the prediction of A_{u2v} dependent variable from *Score*, *dScore* and *content_dist* independent variables. Predictive modeling was conducted after removing outliers from the data. The results of the predictive modeling using linear regression show that we achieve a root mean squared error of 0.1873 across all folds, which means that our prediction varies by 0.1873 from the real values of A_{u2v}. Plot in Fig. 2 depicts how close our predictions are to the real values of the dependent variable.

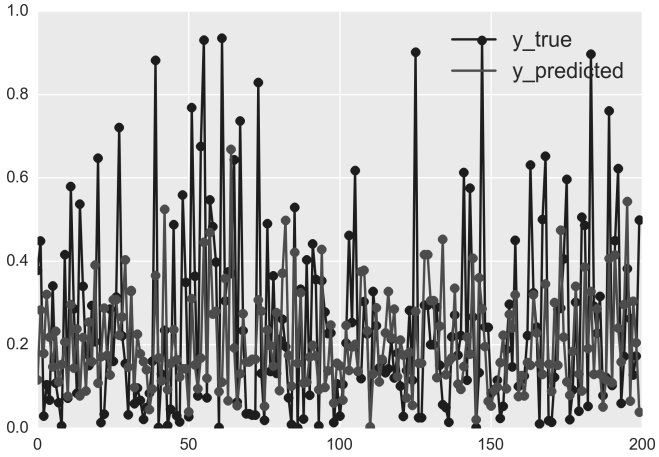

Figure 2. 10-fold Cross-Validation of A_{u2v} without outliers.

5.2. Classification and Comparison with other Models

We predict a user popularity as follows. If A_{u2v} crosses a threshold, e.g. 30%, i.e., if more than 30% of user's u tweets are retweeted by others users, then user u can be considered as a popular user. This can also be interpreted as "if a user u's tweets will be popular or not, given user u's *Score*, *dScore*, and *content_dist* (measure of his YouTube interests' similarity with the Twitter interests of his followers)".

Classification was conducted initially with three different methods: Linear Regression, i.e., the Predictive Model we present in this study, Random Forest and Naive Bayes methods. Area Under the Curve (AUC) is a score that computes average precision (AP) from prediction scores. This average precision score corresponds to the area under the precision-recall curve and the higher AUC represents better performance. Plots in Figure 3 correspond to computed precision-recall pairs for different probability thresholds and the AUC score computes the area under these curves. Best performance is achieved by Linear Regression (0.699), followed by Naive Bayes (AUC:0.608) and Random Forest (AUC:0.608). Complementary methods tested were Support Vector Machines (SVM), Stochastic Gradient Descent (SGD) and K-Nearest Neighbours (KNN).

SVM is a supervised learning model with associated learning algorithm that analyzes data used for classification and regression analysis. Given a set of training examples, each marked to belong to one of the two categories (popular/ non-popular user), the SVM training algorithm builds a model that assigns new examples into each of the categories, acting as a non-probabilistic binary linear classifier.

Next classification model was Stochastic Gradient Descent (SGD), a gradient descent optimization method for minimizing an objective function written as a sum of differentiable functions, and a popular algorithm for training a wide range of models in machine learning, including linear support vector machines, logistic regression and graphical models. Its use for training artificial networks is motivated by the high cost of running backpropagation algorithm over the full training set, as SGD overcomes this cost and still leads to fast convergence.

The last classifier implemented was K-Nearest Neighbours (KNN), a method classifying objects based on closest training examples in the feature space. The input consists of positive, typically small, integer -15 in our case- of closest training examples in the feature space. In KNN classification, the output is a class membership (popular/ non-popular user), whereas an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its K-Nearest Neighbours.

After plotting the results of computed precision-recall pairs for various probability thresholds we observe that best performance is noticed in the case of our Predictive Model, followed by SVM (AUC:0.608), Naive Bayes (AUC:0.608), Random Forest (AUC:0.608), KNN (AUC:0.601), and, lastly, SGD (AUC:0.580).

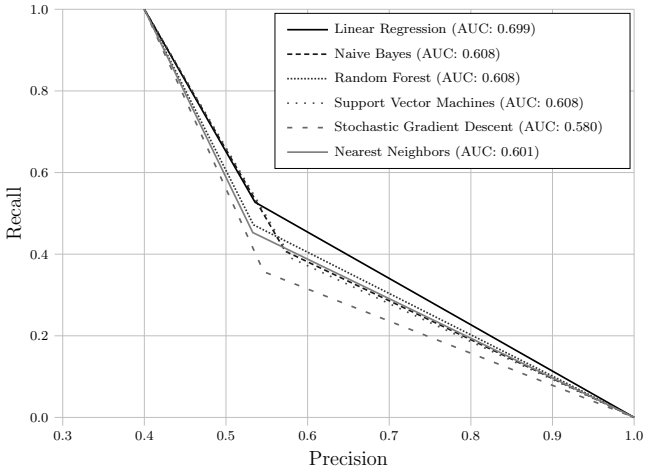

Figure 3. Comparison with other models.

6. Incorporation into Content Delivery Schemes

Content Distribution Networks (CDNs) aim at improving download of large data volumes with high availability and performance. Content generated by online media services circulates and is consumed over OSNs (with more than 400 tweets per minute including a YouTube video link [3] being published per minute) and largely contributes to internet traffic growth [4]. Consequently, CDN users can benefit from an incorporated mechanism of social-awareness over the CDN infrastructure. In [9], [10] Kilanioti and Papadopoulos introduce a dynamic mechanism of preactive copying of content to an existing validated CDN simulation tool and propose various efficient copying policies based on prediction of demand on OSNs.

Rather than pushing data to all CDN surrogates, they proactively distribute it only to social connections of the user likely to consume it. The content is copied only under certain conditions (content with high viewership within the media service, copied to geographically close timezones of the geo-diversed system used where the user has mutual social connections of high influence impact). This contributes to smaller response times for the content to be consumed (for the users) and lower bandwidth costs (for the OSN provider). Herein, we incorporate the proposed Predictive Model in the suggested policy [10] and prove that it further improves its performance.

The proposed herein algorithm encompasses an algorithm for each new request arriving in the CDN and an algorithm for each new object in the surrogate server. We use LRU for handling non-homogeneous sized objects in the surrogates and prune the least recently used items first. To ensure that least recently used items are discarded, the algorithm keeps track of their usage in a scheme described in [10]. Internally, the module implementing the algorithm communicates with the module processing the requests and each addressed server separately (Fig. 4).

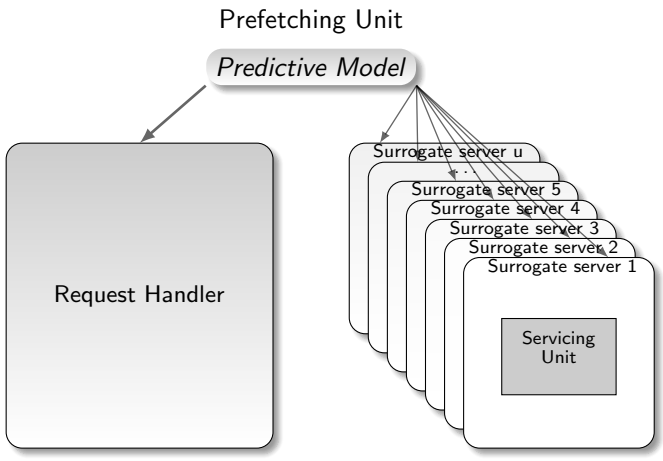

Figure 4. The social-aware CDN mechanism.

6.1. For Every New Request in the CDN

To begin with, we check whether specific time has passed after the start of cascade and, only in the case of an uncompleted cascade, define to what extent the object will be copied. We introduce the *time_threshold*, that roughly expresses the average cascade duration. Initially, we check whether it is the first appearance of the object (Fig. 5). The variable *o.timestamp* depicts the timestamp of the last appearance of the object in a request and helps in calculating the timer related to the duration of the cascade. If it is the first appearance of the object, the timer for the object

TABLE 4. CONTENT DELIVERY VERIFICATION - NOTATION OVERVIEW

$G(t) = (V(t), E(t))$	Graph representing the social network
$V(t) = \{V_1(t), \ldots, V_n(t)\}$	Nodes representing the social network users
$E(t) = \{E_{11}(t), \ldots, E_{1n}(t), \ldots, E_{nn}(t)\}$	Edges representing the social network connections, where E_{ij} stands for friendship between i and j
$R = \{r_1, r_2, \ldots, r_\tau\}$	Regions set
$N = \{n_1, n_2, \ldots, n_\upsilon\}$	The surrogate servers set. Every surrogate server belongs to a region r_i
$C_i, i \in N$	Capacity of surrogate server i in bytes
$O = \{o_1, o_2, \ldots, o_w\}$	Objects set (videos), denoting the objects users can ask for and share
S_i, $o_i \in O$	Size of object i in bytes
Π_i	Popularity of object i, $i \in O$
$q_i = \{t, V_\psi, o_x\}, 1 < x < w, 1 < \psi < n$	Request i consists of a timestamp, the id of the user that asked for the object, and the object id
$P = \{p_{12}, p_{13}, \ldots, p_{nw}\}$	User posts in the social network, where p_{ij} denotes that node i has shared object j in the social network
$pts_i, pte_i, 1 < i < \tau$	peak time start and peak time end for each region in secs
$Q = \{q_1, q_2, \ldots, q_\zeta\}$	Object requests from page containing the media objects, where q_i denotes a request for an object of set O
Q_{hit}, Q_{total}	Number of requests served from surrogate servers of the region of the user/ total number of requests
$X, Y \in R$	Closest timezones with mutual followers / with highest centrality metric values

```
1: if o.timestamp == 0 then
2:    o.timer = 0;
3:    o.timestamp = request_timestamp;
4: else if o.timestamp != 0 then
5:    o.timer = o.timer + (request_timestamp -
      o.timestamp);
6:    o.timestamp = request_timestamp;
7: end if
8: if o.timer > time_threshold then
9:    o.timer = 0;
10:   o.timestamp = 0;
11: else if o.timer < time_threshold and user.Score >
    Score_threshold then
12:   copy object o to surrogate that serves user's V_i(t)
      timezone;
13:   for all user V_y(t) that follows user V_i(t) do
14:      find surrogate server n_j that serves V_y(t)'s time-
         zone;
15:      copy object o to n_j;
16:   end for
17: else if o.timer < time_threshold then
18:   copy object o to surrogates n_j that Subpolicy decides;
19: end if
```

Figure 5. Algorithm for every new request ($timestamp$, $V_i(t)$, o) in the CDN.

```
1: find the Y timezones that depict the highest average
   values of Predictive Model(Score, dScore, content_dist)
   for user V_i(t);
2: for all timezones that belong to Y do
3:    find surrogate server n_j that serves timezone;
4:    copy object o to n_j;
5: end for
```

Figure 6. Subpolicy.

cascade is initialized and *o.timestamp* takes the value of the timestamp of the request. If the cascade is not yet complete (its timer has not surpassed a threshold), we check the importance of the user applying its Score.

For users with Score surpassing a threshold (average value: 1.2943 in the dataset), we copy the object to all surrogate servers of the user's timezone and to the surrogate servers serving the timezones of all followers of the user. Otherwise, selective copying includes only the surrogates that the subpolicy decides. Subpolicy (Fig. 6) checks the Y timezones with the highest value of the combined feature set (Predictive Model(*Score*, *dScore*, *content_dist*)) for the user $V_i(t)$ as an average. Copying is performed to the surrogate servers that serve the Y timezones of highest combined feature set value, according to the coefficients derived from our analysis. We note here that variations of the subpolicy include the replacement of the timezones depicting the highest average values of Predictive Model(*Score*, *dScore*, *content_dist*), with those derived from the application of Naive Bayes, Random Forest, SVM, SGD, and KNN schemes.

The realistic network depiction for the simulation of nodes representing the surrogate servers, the origin server, and the users requesting the objects is analyzed in detail in [9], along with information about the dataset and the configuration settings of the simulations.

We examine Mean Response Time (MRT), the most significant client-side metric associated with CDN user experience. MRT indicates how fast a client is satisfied. For the most representative case of time thresholds covering all the examined requests of our dataset we observe a trade-off between the reduction of the response time and the cost of copying in servers. This is expressed for all schemes used (Linear Regression, Naive Bayes, Random Forest, SVM, SGD, KNN) with a decrease of the MRT as the timezones increase and a point after which the MRT starts to increase again (Fig. 7). For the scheme augmented with our Predictive Model, namely the Linear Regression, this shift occurs with approximately 7 timezones out of the 10 used. After this point the slight increase in the MRT is attributed to the delay for copying content to surrogate servers. The cost for every copy is related to the number of

hops among the client asking for it and the server where copying is likely to take place. We observe that Linear Regression outperforms all the other schemes, depicting MRTs smaller than their respective. The proposed model performs better than the algorithm suggested in [9], and its variations [10], depicting an average MRT of 1.0647 msec. We note here that timezones with highest average values for each scheme, that Subpolicy defines, are pre-calculated, in order to reduce computational burden in the simulations.

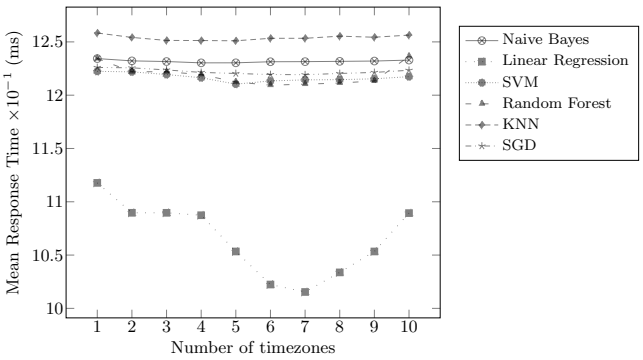

Figure 7. Effect of timezones used as Y on Mean Response Time for various schemes.

7. Conclusions

Circulation via OSNs further intensifies the problem of HTTP traffic caused by online generated content. We conclude herein that video sharings over an OSN platform can be predicted with a small set of features combined from both the platform and the media service. Despite the focused scope of this work and the limitations of its conduction merely with Twitter and YouTube data, the scale of the medium allows us to make assumptions for generalization across different OSNs and microblogging platforms, which we plan to extensively analyze in the future. Our results have direct application to the optimal placement of online video content. We believe that our approach can serve as a useful starting point for extensive experimentation with social-aware content delivery schemes.

References

[1] Abisheva, A., Garimella, V.R.K., Garcia, D., Weber, I.: Who watches (and shares) what on YouTube? And when?: Using Twitter to understand Youtube viewership. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014, pp. 593–602 (2014). DOI 10.1145/2556195.2566588.

[2] Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on Twitter. In: Proceedings of the 4th International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011, pp. 65–74 (2011). DOI 10.1145/1935826.1935845.

[3] Brodersen, A., Scellato, S., Wattenhofer, M.: YouTube around the world: geographic popularity of videos. In: Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012, pp. 241–250 (2012). DOI 10.1145/2187836.2187870.

[4] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y., Moon, S.B.: I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC 2007, San Diego, California, USA, October 24-26, 2007, pp. 1–14 (2007). DOI 10.1145/1298306.1298309.

[5] Cheng, J., Adamic, L.A., Dow, P.A., Kleinberg, J.M., Leskovec, J.: Can cascades be predicted? In: Proceedings of the 23rd International World Wide Web Conference, WWW 2014, Seoul, Republic of Korea, April 7-11, 2014, pp. 925–936 (2014). DOI 10.1145/2566486.2567997.

[6] Dow, P.A., Adamic, L.A., Friggeri, A.: The anatomy of large Facebook cascades. In: Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013. (2013).

[7] Hong, L., Dan, O., Davison, B.D.: Predicting popular messages in Twitter. In: Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (Companion Volume), pp. 57–58 (2011). DOI 10.1145/1963192.1963222.

[8] Jenders, M., Kasneci, G., Naumann, F.: Analyzing and predicting viral tweets. In: Proceedings of the 22nd International World Wide Web Conference, WWW 2013, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume, pp. 657–664 (2013).

[9] Kilanioti, I.: Improving multimedia content delivery via augmentation with social information. The Social Prefetcher approach. IEEE Transactions on Multimedia **17**(9), 1460–1470 (2015). DOI 10.1109/TMM.2015.2459658

[10] Kilanioti, I., Papadopoulos, George A.: Socially-aware multimedia content delivery for the cloud. Proceedings of the 8th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2015, Limassol, Cyprus, December 7-10, 2015, pp. 300–309 (2015). DOI 10.1109/UCC.2015.48

[11] Ma, Z., Sun, A., Cong, G.: On predicting the popularity of newly emerging hashtags in Twitter. JASIST **64**(7), 1399–1410 (2013). DOI 10.1002/asi.22844.

[12] Petrovic, S., Osborne, M., Lavrenko, V.: RT to win! Predicting message propagation in Twitter. In: Proceedings of the 5th International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011 (2011).

[13] Rodrigues, T., Benevenuto, F., Cha, M., Gummadi, P.K., Almeida, V.A.F.: On word-of-mouth based discovery of the web. In: Proceedings of the 11th ACM SIGCOMM Conference on Internet Measurement, IMC 2011, Berlin, Germany, November 2-, 2011, pp. 381–396 (2011). DOI 10.1145/2068816.2068852.

[14] Tsur, O., Rappoport, A.: What's in a hashtag?: Content based prediction of the spread of ideas in microblogging communities. In: Proceedings of the 5th International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012, pp. 643–652 (2012). DOI 10.1145/2124295.2124320.

[15] Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the 4th International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011, pp. 177–186 (2011). DOI 10.1145/1935826.1935863.