A Peek into the Future: Predicting the Evolution of Popularity in User Generated Content

Mohamed Ahmed
NEC Laboratories Europe
Heidelberg, Germany
ahmed@neclab.eu

Stella Spagna
Dept. of Inf. Engineering
Univ. of Pisa, Italy
stella.spagna@iet.unipi.it

Felipe Huici
NEC Laboratories Europe
Heidelberg, Germany
huici@neclab.eu

Saverio Niccolini
NEC Laboratories Europe
Heidelberg, Germany
niccolini@neclab.eu

ABSTRACT

Content popularity prediction finds application in many areas, including media advertising, content caching, movie revenue estimation, traffic management and macro-economic trends forecasting, to name a few. However, predicting this popularity is difficult due to, among others, the effects of external phenomena, the influence of context such as locality and relevance to users, and the difficulty of forecasting information cascades.

In this paper we identify patterns of temporal evolution that are generalisable to distinct types of data, and show that we can (1) accurately classify content based on the evolution of its popularity over time and (2) predict the value of the content's future popularity. We verify the generality of our method by testing it on YouTube, Digg and Vimeo data sets and find our results to outperform the K-Means baseline when classifying the behaviour of content and the linear regression baseline when predicting its popularity.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Algorithm, Measurement

Keywords

Social Media, Time Series Clustering

1. INTRODUCTION

The large and growing body of user-generated content available on the Internet (e.g., blogs, videos, news postings, etc.) presents important opportunities for both understanding how users utilise the Internet and enhancing their experiences. Amongst the challenges here is the accurate forecasting of the popularity of contents to users.

Numerous proposals have been made for applications to make use of popularity information. These include media advertising [14, 22, 30], trend forecasting [2, 20, 22, 28], movie revenue estimation [5], traffic management [4, 9, 13], understanding the collective behaviour of users [11, 21, 36], election prediction [6, 32, 34] and macro economic trend forecasting [3, 35, 37].

However, predicting the popularity of content is difficult for many reasons. Among these, the effects of external phenomena (e.g., media, natural, geo-political) are difficult to incorporate into models [11, 24]. While cascades of information are difficult to forecast [10]. Finally, the underlying context of content such as locality, relevance to users, and resonance/impact, are often difficult to decipher [7].

In this work, we show that content popularity can be predicted without making too many assumptions regarding the underlying phenomena. To do so, we take a two-step approach. First, we categorise the behaviour of content overtime to reveal distinct patterns of popularity growth (over time) that generalise the variety of different behaviours displayed by large numbers of content. Second, based on this result we show that it is possible to predict the future popularity of content (with high accuracy) by tracing the types of popularity growth behaviour that contents display over time.

In this work, we define a simple and easily generalised feature space (Section 3), and show that it can be used to accurately categorise the behaviour of content (Section **??**) and predict the evolution of its popularity (Section 5). We validate our methods using three very different data sets (Section 4). Finally, we compare our results against a baseline of K-Means for the clustering and linear regression for the prediction and show that we are able to significantly outperform both at categorising and predicting the popularity growth behaviour of content.

2. RELATED WORK

Given the breadth of this topic, it is unsurprising that there have been many approaches to address the problem of content popularity prediction. This section covers other works in the field and contrasts them with ours.

Feature space: Numerous works [2, 16, 18, 26, 27, 30] study how features that describe the underlying social net-

WSDM'13, February 4–8, 2013, Rome, Italy.

work of the users that generate and use content can be leveraged to predict its popularity. The authors in [20, 24, 25, 31] study how features that take into account comments related to the content (e.g., comments found in blogs) can be used to predict popularity. Our approach differs from these in that we use a more general feature space (simply the number of hits associated with content and their timings) so as not to over-specialise for a particular application context.

Clustering: A number of works have studied the shape of the attention paid to content over time. The works in [11, 21] show that the popularity evolution of YouTube videos and Twitter messages can be categorised based on whether the stimuli that triggers them comes from within or outside the system and whether or not the users can spread news about the event. Adar et al. [1] use dynamic time warping [19] to reveal the correlation between search queries from different content sources and show that some of this information can be used to predict whether the queries at one source (e.g., blogs) trigger queries at another (e.g., web search). Laniado and Mika [23] show that the Twitter Hashtags that represent "valuable" identifiers have distinct temporal behaviours. Finally, Yang and Leskovec [36] show that the attention paid to content over time can be characterised to reveal generalisable patterns of temporal popularity evolution.

In this work we categorise the behaviour of content over time to reveal distinct patterns of popularity growth. We show that relatively a few classes of behaviour can be used to capture population growth patterns exhibited by large numbers of different contents. Unlike previous work, we build a transition model based on the classes of behaviour identified and use this information to predict the future popularity of content.

Prediction: Amongst others [9, 10, 29] show that there is a correlation in the views that content receives over time. There are however relatively few works that forecast a value for the actual popularity of content. Lee et al. [24, 25] use survival analysis to evaluate the probability that a given content will receive more than some x number of hits. Hong et al. [17] develop a coarse multi-class classifier based approach to determine whether a given Twitter Hashtag will be retweeted $x < (0, 100, 10000, \infty)$ times. Similarly, Lakkaraju and Ajmera [22] use Support Vector Machines to predict whether a given content will fall in a group that attracts $x < (10\%, 25\%, 50\%, 75\%, 100\%)$ of the "attention" in a system, while Jamali and Rangwala [18] predict the popularity of content using an entropy measure based on the "user-interest peak" and the "co-participatory network". Finally Szabo and Huberman [29] present a linear regression model based on the number of views; this method is applied by [7, 16, 26, 31] to build predictive popularity based on applying regression to different feature spaces.

In this work we show that a predictor based on modelling the transitions that content makes between being a member of different classes of behaviour (over time) outperforms the linear regression method commonly used in other papers (Section 5.1).

3. MODELLING THE TIME-EVOLUTION OF POPULARITY

The goal of content popularity prediction is to accurately estimate the number of hits associated with a given content object at some time in the future, based on the observation of its past behaviour.

In this section we formally define the content popularity problem, starting from the time series of the cumulative number of hits a given content attracts. We give the feature space based on the time series, and the similarity measure used to relate content. We then show how similar content may be clustered using the Affinity Propagation algorithm [15] and based on the results of the clustering, how the behaviour of individual content over time can be predicted.

The difficulty with defining a feature space for evaluating content behaviour is finding an acceptable trade-off between accuracy and over-specialisation to the application. Incorporating complex features such as social network variables, content context and other approaches reduces the generality of the feature space.

In this work we use general and publicly observable variables to predict content popularity. Specifically, we look only at the number of hits associated with content, and the timings of those hits. Based on this, we define (i) a two-variable feature space that captures the magnitude and timings of the attention content attracts from users; and (ii) a correlation-based similarity metric to group individual content based on how much alike they are in either the number of hits they attract or the timings of those hits.

To predict the popularity of content, we take a two-step approach. In the first step we cluster the data to produce a set of discretised temporal classifiers that capture the patterns of popularity evolution that contents display. In the second step, we model the probabilistic transitions that contents make between these clusters over time, and use this information to predict the future popularity of content (see Figure 1).

Our results (see Section **??** and Section 5) show that the feature space we define is generalisable to different data sets (tested on 3 distinct ones) and suitable for accurately predicting the popularity of content.

3.1 Problem Definition

Given the observations made of some N contents over a period of time T, we define $x_i(t), i \in N, t \in T$ to represent the number of hits received by content i at time t. $x_i(t)$ may be sampled to produce W windows each of length $l = \lceil \frac{T}{W} \rceil$, such that the generic element of the series is $x_i[n], n \in W$.

Given the resulting time windows, our goals are: i) to extract features that capture the particular behaviour of the content, that is, its growth in popularity relative to other observed content; ii) to use these features to derive a set of $|M| < |N|$ behaviour-sets that group together contents sharing similar properties, e.g., all contents whose popularity saturates quickly[1]; and iii) given the resulting M behaviour-sets over the W windows, to predict the future popularity of content by deriving the transition table for moving between behaviours-sets over W.

3.2 Feature Space

Because the popularity of content can be differentiated by how many hits each content attracts and when those hits arrive, we define a two-dimensional feature vector to capture this. The first feature vector is termed SA and captures the *Share of (user) Attention* a content object attracts with respect to all other observed contents at some time interval.

[1] Throughout the paper we use saturation to mean the time it takes for content to receive 90% of all their hits.

The second feature vector, termed $nROC$, captures the *normalised Rate of Change* in the attention attracted, again at some time interval.

In greater detail, the SA feature vector is a function of the number of hits some content receives, relative to the hits received by other monitored content. This is expressed as:

$$SA_i[n] = f\left(x_i[n], x_j[n], i, j \in N\right) \tag{1}$$

In the simplest case, $f(.)$ defines the ratio of $x_i[n]$ to the total number of hits ($\sum_{j\in N} x_j[n]$) in the measured interval n. However, the definition $f(.)$ is clearly subject to the application. For example, if content displays a high variance in the number of hits received, then taking a simple ratio may overestimate the impact of some contents over others. In our case, we take the ratio of the number of hits a content receives in the interval to the maximum number of hits received by contents in the interval, defined as:

$$SA_i[n] = \frac{x_i[n]}{\max\limits_{j\in N}(x_j[n])} \tag{2}$$

This measures the relative magnitude in the popularity of content i with respect to the most popular content in the set of observed contents during the interval n. Using a relative measure means that the feature is unaffected by exogenous behaviours such as diurnal patterns [8] in content use.

The $nROC$ feature, on the other hand, captures when attention is paid to content by tracking the relative change in the number of hits a content receives over time. The generic value of the $nROC$ vector is defined as:

$$nROC_i[n] = \frac{ROC_i[n]}{\max(1, \max\limits_{j\in(0,n-1)}(ROC_i[j]))} \tag{3}$$

where $ROC_i[n] = (x_i[n] - x_i[n-1])$ is the rate of change in the number of hits during the interval, and the denominator takes a range between 1 and the maximum rate of change experienced by the content thus far.

The normalisation in Equation 3 relates the number of hits a content attracts during the interval ($x_i[n]$), to the maximum it has attracted in past intervals ($\max\limits_{j\in(0,n-1)}(ROC_i[j])$), and has the following properties:

$$nROC_i[n] = \begin{cases} 0 & \text{//no hits attracted} \\ ROC_i[n] & \text{//at the first change in the \# of hits} \\ < 0 & \text{//\# hits per interval is decreasing} \\ \geq 1 & \text{//\# hits per interval is increasing} \end{cases} \tag{4}$$

3.3 Measure of similarity

The similarity between two contents at a given time interval captures to what degree they are amplitude and/or phase synchronised. In other words, it measures to what extent two contents attract similar numbers of hits (SA) or experience growth at the same time ($nROC$). For this work we define a negative similarity measure[2] such that $s_{i,j} = -1$ indicates the least similarity between the contents and $s_{i,j} = 0$ the highest.

With respect to the SA feature, the similarity between contents i, j is a measure of the distance between the magnitudes of their popularity based on their cross-correlation. Because the time series of two contents ($x_i[n], x_j[n]$, $\forall n$) form two piecewise constant waveforms, we may use their cross-correlation to measure their similarity as a function of a time-lag applied to one of them. Taking the maximum of the cross-correlation gives us a measure of maximum overlapping area between the waveforms. The result is normalised to a value between 0 and 1 by dividing by the auto-correlation. The SA similarity between contents i, j is therefore defined as:

$$sS_{i,j}(SA) = -(1 - \frac{g(SA_i, SA_j)}{max(h(SA_i), h(SA_j))}) \tag{5}$$

where the function $g(.)$ is the maximum value of the discrete convolution of i and j, and $h(.)$ is the maximum value of the auto-correlation, given by:

$$\begin{aligned} g(SA_i, SA_j) &= \max\{\textstyle\sum_k SA_i[n-k]\cdot SA_j[k]\} && (6) \\ h(SA_i) &= max\{\textstyle\sum_k SA_i[n-k]\cdot SA_i[k]\} && (7) \end{aligned}$$

With respect to the $nROC$ feature, the similarity between two contents i, j ($sR_{i,j}$) over an interval conveys the overlap in the periods where the contents experience a change in the number of hits they receive. This is defined as a negative cosine similarity, given by:

$$sR_{i,j}(n) = \quad -(1 - \frac{\sum_{k\in n} nROC^*_i(k)\cdot nROC^*_j(k)}{\|nROC^*_i\|\cdot\|nROC^*_j\|}) \tag{8}$$

where $nROC^*_i \in (0,1)$ is the normalisation of the feature vector to $\max(nROC_i[n])$, $\forall n$ and $\|nROC^*\|$ is its norm.

Given the time series of two contents, the definition in Equation 8 takes on values proportional to the time between jumps in the number of hits attracted by the content. The similarity is maximum (0) when two time series are in phase (they jump at the same time instant) and -1 when their jump phases do not overlap.

For each time window in W, we can now construct two square ($|N| \times |N|$) similarity matrices representing the SA ($SA = [sS_{i,j}]_{|N|\times|N|}$) and SR ($SR = [sR_{i,j}]_{|N|\times|N|}$) similarities defined. We use these as input to the clustering algorithm defined in the next section.

3.4 Clustering

For clustering we use the Affinity Propagation (AP) algorithm [15] because it is able to handle non-linear dependencies between data and does not require us to predefine the number of clusters to seek, which helps to reduce the complexity of the training phase.

In brief, AP is a message passing algorithm which initially considers all data points, referred to as nodes, as potential centroids. Nodes exchange messages that are used to calculate their eligibility as centroids, and as the algorithm converges, nodes with low eligibility are discarded. The two main drawbacks of AP are (i) its computational cost and (ii) when using the Euclidean distance to measure similarity (as suggested in [15]) it has a tendency to produce a large number of clusters.

To address the first problem we use a sparse matrix representation, and reduce the number of messages exchanged by not considering nodes whose similarity falls below a given threshold. Fixing the similarity between data points in this way also enables us to tune the classification accuracy of

[2] The Affinity Propagation algorithm defined in Section 3.4 requires a negative similarity measure.

the algorithm; for example, we can increase the intra-cluster similarity by demanding higher correlation between similar data points. The second problem is addressed by our definition of the similarity measure, which is able to reduce the number of clusters without sacrificing the classification accuracy.

Applying this modified clustering algorithm results in a set M where each element consists of a set of clusters for a particular time window. More formally, $M = \{M_1 \dots M_W\}$, where W is the set windows, and each cluster set in $M_{i \in W} = \{m_1 \dots m_m\}$ is composed of the set of clusters identified in the ith time window.

3.5 Prediction

In order to predict the popularity of content, we model the transition probabilities between the members of the cluster set M defined in the previous section. As depicted in Figure 1, because the classification process associates each content in N with exactly one centroid in each time window, tracking the centroids that a given object is associated with over time gives the evolution of that object's popularity.

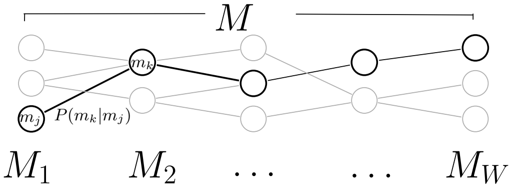

Figure 1: Simple path transition graph depicting the movement of a given content object between clusters over different time windows.

Given the generic time window w, the basic observation here is that the cluster set M forms a directed acyclic graph ($G = (V, E)$), with the vertices being the centroids ($V = M_w$) and the edges the transition probabilities between them. Because each centroid is associated with a given time window, the probability of the k'th centroid in time window w (m_k^w) is given by the ratio of the number of contents associated with the centroid ($mem(m_k^w)$) to the number of classified contents in the time window ($mem(M_w)$). This is expressed as:

$$P(m_k^w) = \frac{mem(m_k^w)}{mem(M_w)} \tag{9}$$

Therefore, the probability of any given content ending up in the centroid m_k^w is the joint probability of all paths leading to m_k^w.

More importantly, given a particular path that a content takes between time windows $1 \dots w$, ($p = \{m_j^1, \dots, m_j^w\}, j \in M_w$), we can evaluate popularity of the content at time $w + r$ by identifying the centroid at the end of the maximum likelihood path between the path (p) and all the centroids in M_{w+r} (m^*), given as:

$$m^* = \underset{j \in M_{w+r}}{\arg\max} \left(\frac{P(m_j)P(p|m_j)}{\sum_{h=1}^{w} P(m_j)|P(M_h|m_j)} \right) \tag{10}$$

where the numerator is the conditional probability of all paths going through p and ending in m_j, and the denominator the probability of all paths ending in m_j.

Evaluating Equation 10 can easily become computationally intractable. If the number of nodes involved is moderate it is possible to use exact inference to marginalise the probability of all paths between m_k^w and m_k^{w+r}, or to directly apply the Viterbi algorithm [33] to do a greedy search for the maximum likelihood path. When this is not possible, we can take advantage of Monte-Carlo methods to approximate the probabilities of the possible paths contents may take as given in Algorithm 1.

The algorithm takes as input the set of distinct paths in the graph and a path length (l). These are then used to sample the graph G and approximate the likelihood of given paths. The final output is a set of transition tables (for different path lengths) which approximate the transition probabilities of contents over time.

```
input  : graph_paths, path_length
output: path_distribution
begin
    $path\_dist = zeros(path\_length)$
    repeat
        for $i$=1 to $path\_length$ do
        $trial\_path = path\_dist[i] + random\_permutation$
        if $trial\_path \notin graph\_paths$ then
            continue
        end
        $m_j = trial\_path[path\_length])$
        $p = trial\_path[0 : path\_length - 1])$
        $all\_paths = \sum P(m_j)|P(graph\_paths|m_j)$
        $ratio = \frac{P(m_j)P(p|m_j)}{all\_paths}$
        $a = random(0, 1)$
        if $ratio > a$ then
            $path\_dist[i] + = 1$ /* trial_path accepted */
        else
            continue /* keep the old sate */
        end
        /* Viterbi path approximation given by:
        likely_path = argmax(path_dist) */
    until $n$ times;
end
```

Algorithm 1: Monte Carlo sampling for the ML path.

4. DATA SETS

This section describes the data sets used in this work, listing their key differences and then discusses the pre-processing applied to generate the feature space used later in the clustering process. Table 1 gives the three data sets used in our experiments. The YouTube and Vimeo data sets consist of the number of views contents receive, while the Digg data set consists of the number of user votes contents receive. All the data sets were collected by directly crawling the relevant websites.

As well as differing in their content, size and sampling granularity, the data sets come from applications that display different popularity evolution characteristics. For example, Digg contents show an average popularity saturation time of around 68hrs and Vimeo an average of almost 16.1 months, while YouTube shows a multi-modal pattern, with two peaks at 18 and 27.6 months. The data sets also differ in how content becomes popular. For example, the Vimeo data set displays both the smallest average number of hits per-content and smallest average rate of change (making long periods of samples redundant). At the other extreme, popular Digg contents shows the greatest average rate of change in the number of hits per time interval, with

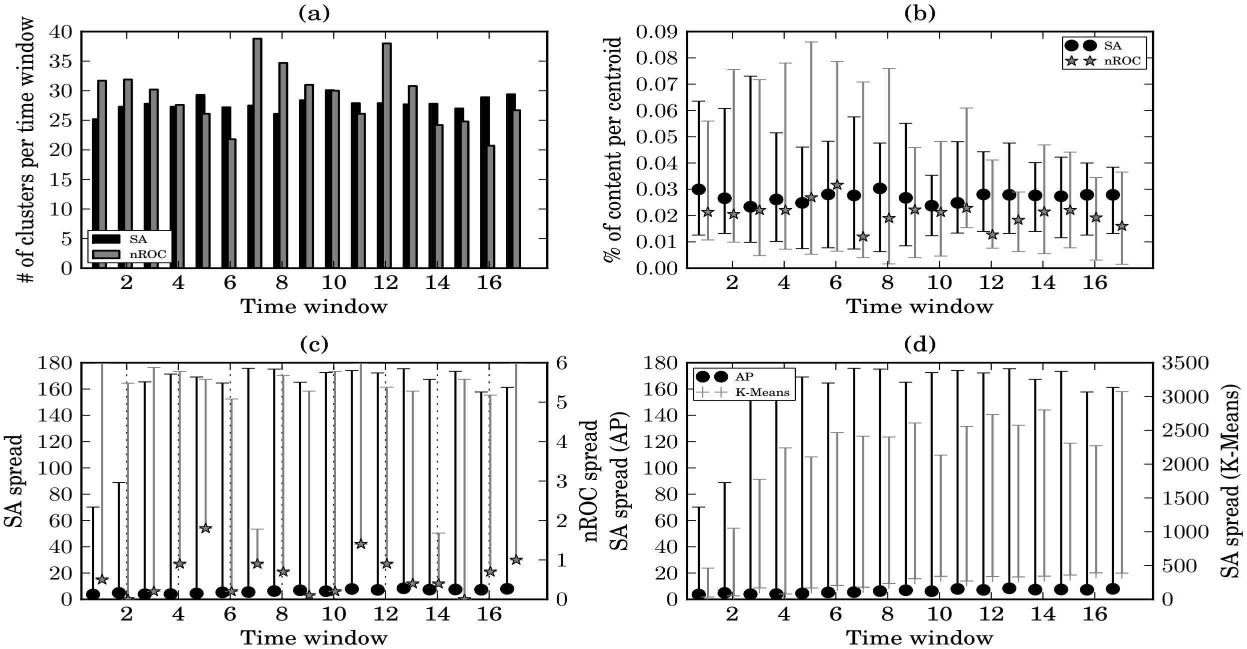

Figure 2: Digg data cluster properties. (a) The median number of clusters identified when using each feature vector. (b) The median, min and max percentage of number of contents associated with each cluster. (c) The median and max cluster spread given in terms of the number of hits (*SA*) and time shift (*nROC*). It is worth noting that 97.5% of content associated with clusters has a radius under half the max distance from the median of the cluster. Finally, (d) the K-Means cluster spread where K is equal to the number of clusters identified by the Affinity Propagation algorithm at the time window.

Name	# of contents	Dates	Granularity
Digg	87x10^3	11/'07 - 12/'07	1 / 10 mins
Vimeo	12x10^3	12/'05 - 04/'10	1 / 30 mins
YouTube	1.44x10^6	05/'06 - 11/'11	1 / 468 hrs (max)

Table 1: The experimental data sets. The data sets were chosen because they difference in their size, temporal span/granularity, and the contents' popularity growth behaviour. The sample granularity for YouTube ranges between 1/24hrs to 1/486hrs.

some contents experiencing a significant growth burst within 7hrs of submission and then saturating within 25hrs. This is in contrast to YouTube contents which display very long growth periods lasting weeks to months.

The YouTube data set is interesting because of the length of time it covers, its low temporal granularity and the content popularity growth longevity displayed by some contents.

For each content in the YouTube data set we have only 100 sample points that are uniformly distributed between the contents' upload and crawl dates. This means that depending on the age of the content each sample point summarises the number of hits received in 1 to 19 days.

4.1 Data set processing

To process the data, we first create the slotted time windows described in Section 3.1. The window sizes are chosen experimentally to either smooth the time series (Digg) or reveal interesting movements (Vimeo).[3]

Digg: For the Digg data set, we temporally aligned the data such that all sampled objects start at time zero. Although this has the effect of inflating the number of hits per given interval (since we are now considering all objects per time interval), because the sampling interval is constant, the normalisation to calculate the *SA* feature is not biased. To construct the *SA* and *nROC* feature vectors, we use 4hr windows composed of 30 minute slots. Therefore, given the original data (with 10min sample intervals), each window is composed of eight 30 minute slots.

YouTube: Because of its large temporal span and coarse granularity, the YouTube data set allows us to explore the effectiveness of identifying similar content based on the shape of its popularity growth. This means that we can look at identifying similar contents of different ages based on whether their popularity curves evolve in the same way - by comparing them at the same relative portions in their life time, e.g. the number of hits attracted when objects are at 50% of their observed lifetime. For this reason, we do not temporally align YouTube contents to compare only objects of the same age. Again based on experimentation, the YouTube feature vectors are composed of 10 sample windows.

Vimeo: Although the Vimeo data set has a high temporal granularity, the average variance of the number of views per content over time is the lowest - popularity growth is on average very slow, making it difficult to see interesting

[3] We experimented with using volatility measures [12] to determine the window size, but the results were not very convincing in helping to analytically determine the length of windows.

behaviour. For this reason, each VIMEO window contains 3428 samples, equal to two months of observations.

Finally, all experiments in the next sections are conducted by performing repeated random sub-sampling.

presenting the results of the classification process, we first look at the properties of clusters identified by the AP algorithm. This enables us to better understand the clustering results we obtain and to compare them against the performance of our baseline implementation of K-Means.

Figure 2 illustrates the properties of the clusters identified for the DIGG data set. First, we see that the number of clusters per time window ($|M|$) is four orders of magnitude less than the number of content objects (Figure 2a and Table 1). This in effect vindicates the hypothesis that a small number of behaviour patterns underlie the range of temporal popularity evolution we observe in the data. Second, the median number of objects per cluster is not skewed with respect to either feature (Figure 2b). We find that at the extreme, the most populous cluster at a given time window contains at most 30% more objects than the median in the time window (Figure 2b).

The median cluster spread reveals (Figure 2c) that the clusters identified are very tightly packed, the $nROC$ spread - which represents the phase shift between a given content and its associated centroid - reflecting the difference in time between when a given content and its associated centroid experience their biggest jumps in popularity in a given time window, is on the median $1-2$ slots ($30-60$mins) and in the worst case 6 slots (3hrs) for a few outliers ($< 97.5\%$ of centroid members). Comparing the results obtained with the $K-$Means baseline reveals the compactness of the clusters produced. Figure 2d highlights this by comparing the SA spread when using Affinity Propagation and K-Means (for the same features).

For the VIMEO data set, we find that the number of clusters identified tends to be roughly between $1-6\%$ of the data set size ($100-600$ centroids). For YOUTUBE the number of clusters ranges between $0.1-0.4\%$ of the contents per time-interval. The large number of clusters here (in absolute terms) results from the low granularity of the data set, and the lack of temporal realignment to compare objects at absolute points in their lifetime.

Similar to the DIGG result, applying $K-$Means to the VIMEO and YOUTUBE data sets, gives an even larger cluster spread - reflecting the long-tail distribution of the number of hits attracted by contents. Both the median and maximum cluster spread from Affinity Propagation (AP) are always at least an order of magnitude less than the baseline when using the same data and feature space. The impact of this as we will see in the next section is that using $K-$Means results in a poorer classifier for the data set. Further, the compactness of the clusters generated by AP means that we can have a relatively small upper-bound on the prediction error when using the results to predict the popularity of content.

4.2 Accuracy of Clustering

Figure 3 gives the classification error for the clusters identified. The error here is defined as the normalised mean squared error (MSE) in terms of the distance between a classified object and its associated centroid. With regard to the SA feature, distance is the difference in the number of hits attained, while for the $nROC$ feature, the distance is the number of time window slots between when the object experiences its biggest jump and when the centroid does.

From Figure 3, we see that the median classification error for the SA feature on DIGG is between $4-15\%$ of the number of hits associated with a centroid, while for the $nROC$ feature this is between $1.3-16\%$. In practise, an average SA error of 10% in the first time window means a 35 hit difference between a content and its associated centroid. With regard to the sample maximum this is 350 hits, while in the last time window 10% translates on average to 110 hits. Regarding the $nROC$ feature, the maximum error is bounded by the number of slots in the window, and an error of 16% translates to approximately 1.5hrs difference between when content and its centroid experience their biggest jumps.

It is worth pointing out that the upward trend in the $nROC$ error for DIGG data in Figure 3 is somewhat misleading. In absolute numbers (not reported here), during early time windows, the $nROC$ values are large ($>> 1$) as contents attract the vast majority of their hits, while in the last time windows, contents attract very few hits (typically single digits over the 4hr time window). As a result, the $nROC$ values are very small ($0-0.1$) but have a high dispersion, which accounts for the high normalised error.

Regarding the YOUTUBE data set, the $nROC$ feature shows a worst case error of approximately 70%. This is equal to 7 samples, while the median values are $20-30\%$. The poor performance here is due to the lack of temporal alignment of the data. However, given that the 100 sample points associated with a given YOUTUBE content may represent anything from 100 days to over 5 years, a worst case of 70% still shows considerable accuracy in identifying the shape of the jumps in popularity that content experiences. In comparison, with temporally-aligned and higher granularity VIMEO data, we get a $nROC$ error that is bounded under 25% of the centroid's jump time for 95% of the data.

On the SA feature, for YOUTUBE we obtain an error of less than 14% in the worst case, while for the VIMEO the worst case is less than 6%. To contextualise these numbers, the cluster spread for YOUTUBE in the first time interval is $1,000$ hits for the highest outlier, $3,000$ for the second time interval and $3,500$ for the last time interval. Therefore, a worst-case error of 10% in the first time interval equates to 100 hits; a maximum error of 13% in the second time interval represents 390 hits and a maximum error of 2% in the last time interval is 70 hits. For the VIMEO data set, the cluster spread is 50 hits for the first time interval and 25 for the last, giving a maximum worst case error of 30 hits in the first time window.

In comparison with the baseline implementation of K-means, on the DIGG data set (again using the same SA feature space) we obtain a worst case median classification error of 50% and max of 200% compared with AP (16% median, 26% max). This error is even worse for the VIMEO and YOUTUBE data sets (reflecting the power law distribution in the video views). For YOUTUBE and VIMEO, we obtain a worst case median and max classification error of $\approx 150\%, 900\%$ and $\approx 250\%, 700\%$, respectively.

In contrast to related works [17, 22] which classify content to be a member of broad classes of behaviour (i.e. *not popular*, *popular*, *very popular*), these results show that we are accurate at classifying content based on the number of hits they accumulate. For example, when classifying the number of hits for a DIGG contents 8hrs after submission we obtain

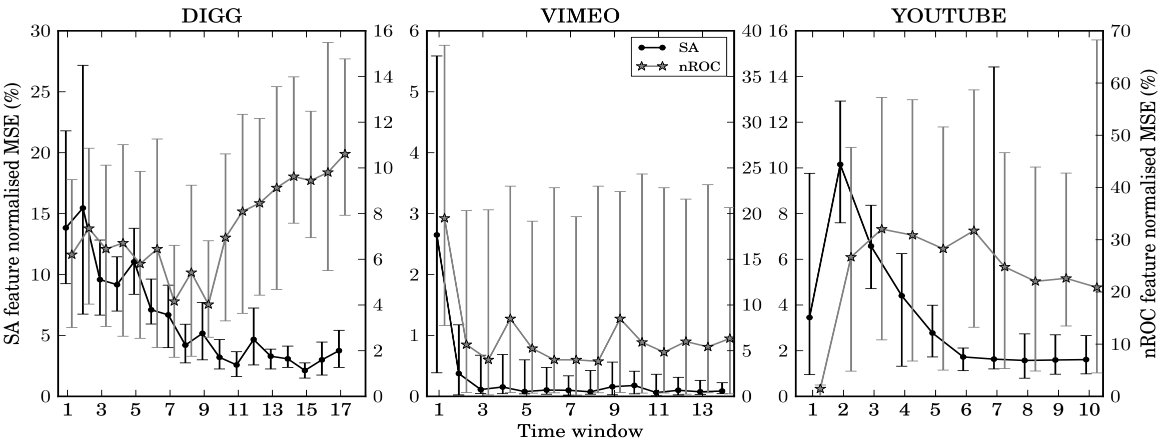

Figure 3: Classification error for the different data sets.

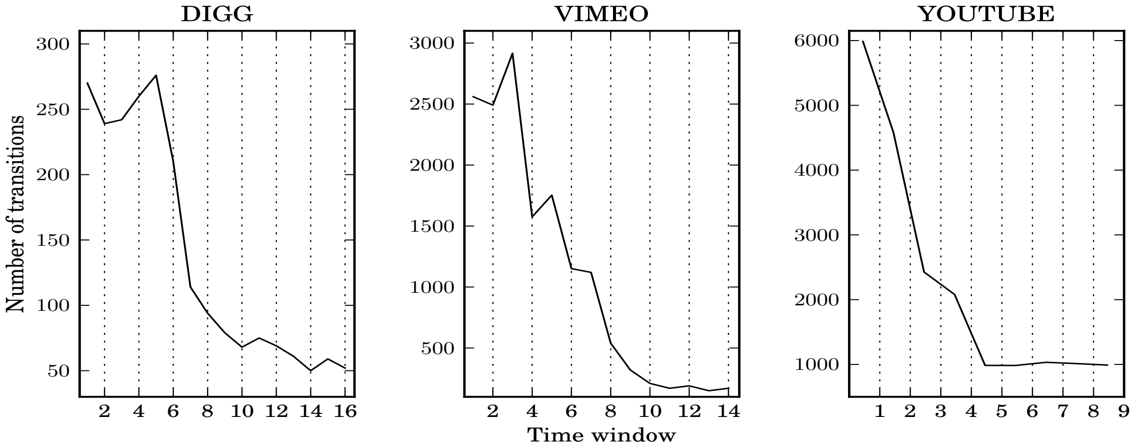

Figure 4: Number of distinct transitions per time window.

an error of 16% median and 26% max.[4] After 24hrs, this value falls to 7% median, 9% max; in contrast, the work in [16] reports an error of 40% after 24hrs.

5. POPULARITY PREDICTION

This section presents our popularity prediction results. We start by discussing the properties of the transitions that contents make when associated with different clusters over their lifetime. We then look at the prediction accuracy of applying the maximum likelihood path prediction to identify the popularity of content. Finally we compare our results against a baseline of linear regression.

As discussed in Section 3.5, the final output of the clustering process is the set of cluster sets identified for each time window ($M = \{M_1, \ldots, M_W\}$). Now, given some object x_i that at time t is classified to be a member of cluster m_j^t, our goal is to predict the most likely cluster that x_i will belong to at time $t+k$, termed m_j^{t+k}. This translates to identifying the centroid at the end of the maximum likelihood path between m_j^t and all the centroids in M_{t+k}.

Before presenting the results of the prediction process, we first look at the transition properties between clusters, that is the paths that the different contents trace as they transition from cluster to cluster over time. Figure 4 illustrates the number of distinct cluster transitions content makes over the time windows. The early stages of content popularity growth show the highest number of transitions, reflecting the high dispersion in the number of hits initially accumulated by content. As the popularity of contents saturates, the number of transitions stabilises, converging to the number of clusters identified per time window.

Figure 5 illustrates the results when using the SA feature vector, giving the error when predicting the future popularity of content based on its cluster membership. Here, the *prediction horizon* refers to how far in the future we project, and the error refers to the normalised mean squared prediction error (MSPE), which normalises the difference between the classified object's number of hits and the number of hits of the centroid it is associated with, to the number of hits at the centroid.

For instance, looking at the one-step (t_{i+1} given t_i) predictions errors in Figure 5, for Digg we obtain an error between $18.1 - 1.02\%$ (compared to $48 - 23\%$ for linear regression, see Section 5.1), where the first number refers to the first window when there is minimal information about a content's behaviour and the last number refers to the last time window.[5] Similarly, we obtain low prediction errors for the other data sets; $27 - 2.9\%$ for Vimeo, and $19.7 - 1.6\%$ for YouTube.

With regard to the *nROC* feature, using the one-step pre-

[4] Where max represents less than 5% of all content.

[5] The prediction error in the last time window is very low because it captures a point when content objects have saturated in their popularity; this is reflected in the number of cluster transitions converging as shown in Figure 4.

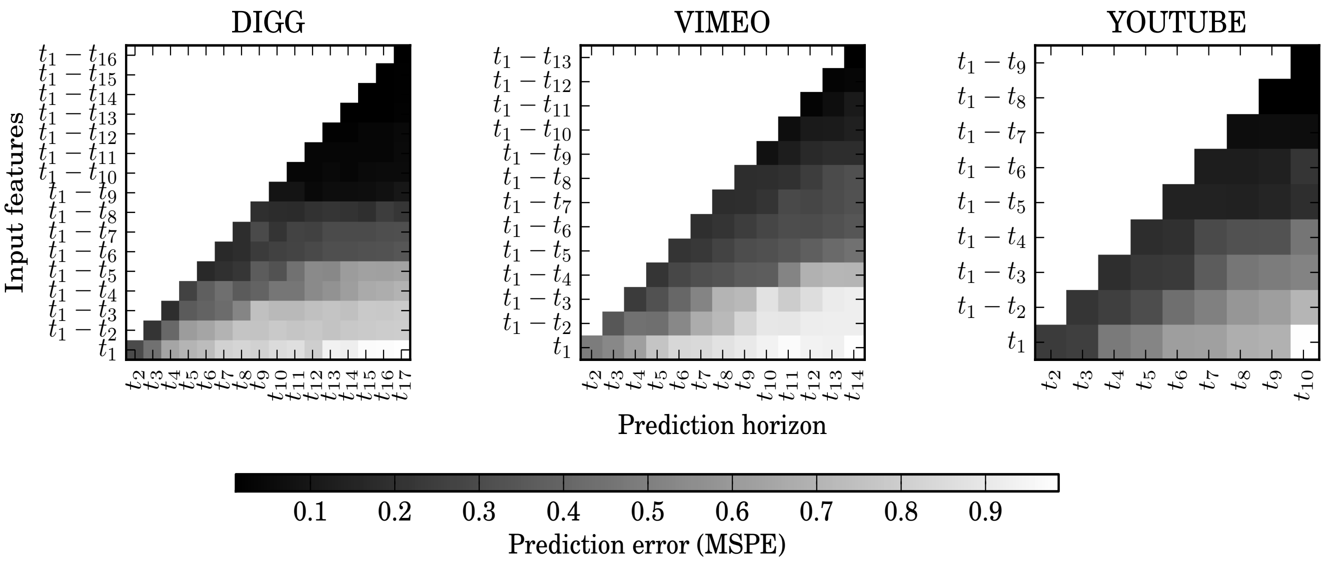

Figure 5: The SA feature prediction error. Given as input the cluster(s) that a given content is associated with from time t_i to t_j (the y-axis), the graph gives the MSPE (the x-axis) for time t_k where k is the prediction horizon.

diction, we find the same general trend in the convergence of the prediction error over time. We obtain a $20.3 - 3.2\%$ error for Digg, $35 - 6.9\%$ for Vimeo, and finally $52.6 - 8.1\%$ for YouTube.

In our analysis so far we have used the entire span of sample points in our data sets. But are there certain sample points or time windows that have more impact in achieving small prediction errors? To answer this question, we began by dividing a content's popularity evolution into three phases: *initialisation*, during which content achieves $0 - 10\%$ of its hits; *growth*, where content gains $10 - 90\%$ of its hits; and *saturation*, where it receives 90%+ of hits.

In brief, having observed that fewer transitions (convergence) results in lower prediction errors, we would like to examine the relationship between the number of transitions and the phases just defined. To highlight this relationship, Figure 6 plots the distribution of the time it takes for content to achieve 10% of its lifetime views (i.e., enter the growth phase)[6] alongside the number of distinct transitions between centroids that contents experience over time.

From Figure 6, we see that the end of the distribution (at 24hrs) coincides with the highest rate of change in the number of cluster transitions as the majority of contents enter the growth phase, meaning that after this point the number of transitions starts to converge and the prediction error decreases. In effect the greatest improvement in the prediction accuracy is observed at point where contents start to attract hits - reflecting the entry into growth phase.

This result leads us to understand that in order to accurately predict the future popularity of content, it is not necessary to use all the points in the feature space - i.e., provide a long history of observations to the prediction process, but rather to choose the most informative subset of points. Specifically, feature points that correspond to when content enters its popularity growth phase have the biggest impact in reducing the prediction error.

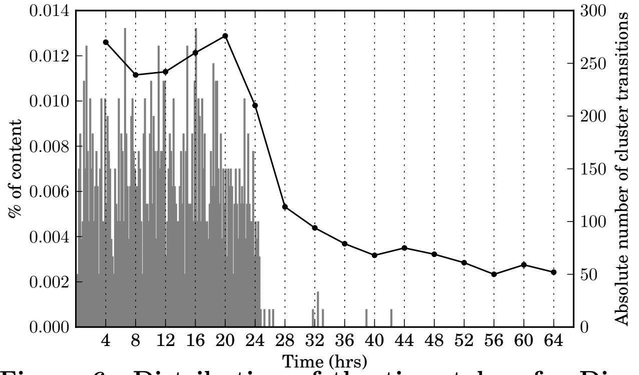

Figure 6: Distribution of the time taken for Digg content to achieve 10% of its lifetime hits (enter the growth phase) alongside the number of centroid transitions.

To better understand this, Figure 7 shows how the prediction error drops as content starts to enter the growth phase (the y-axis gives the MSPE from a given time window t all the way to the last window in each data set). From the figure, we see that the biggest change in the prediction error comes during the initial part of the growth phase (the shaded area).

The practical side effect is, contents that attract a high volume of hits in a short time are identified quickly. Further, this result has the potential to affect how the inputs used to predict popularity are selected, since the points in the feature space corresponding to the beginning of the growth phase contribute most to reducing the prediction error. It is also worth pointing out that identifying the beginning of this growth phase is relatively easy since it corresponds to the point where the rate of change in the number of hits starts to increase the fastest.

[6] Not including all contents with less than 10 hits in their life time.

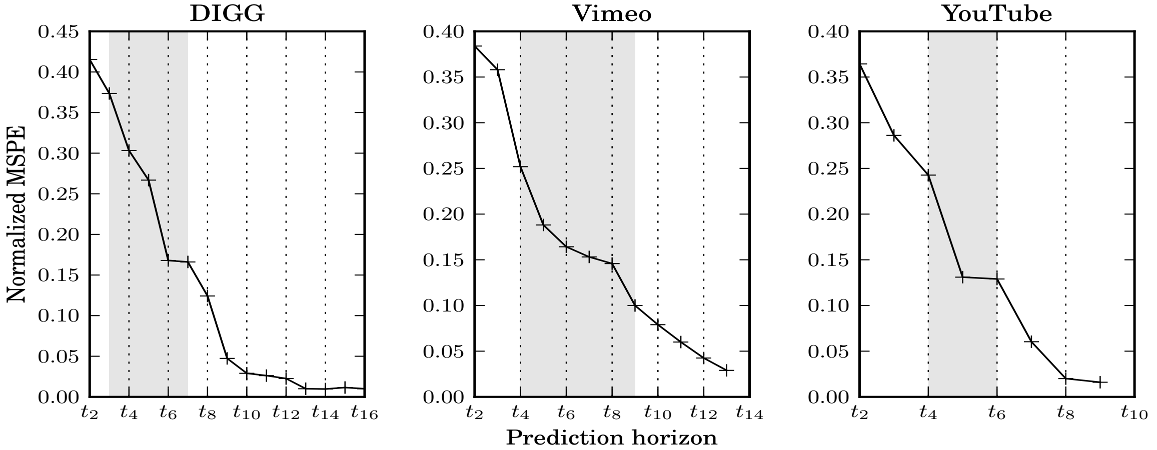

Figure 7: The prediction errors during the growth phase. The solid line gives the normalised MSPE from window t_i to the last window, while the shaded area gives the length of the growth phase (the time taken to accumulate 10% of hits) for 85% of contents.

5.1 Linear Regression Comparison

In this section we compare our results against the baseline implementation of linear regression, the most popular regression technique applied to content popularity prediction (see [7, 16, 26, 29, 31] amongst others).

Figure 8 gives the prediction error obtained when applying linear regression to the DIGG data as suggested by [29], alongside our results (labelled maximum likelihood). Following the author's suggestion, given the number of hits at time t, the linear regression predicts the number of hits that the content expects to achieve $t + k$ steps ahead.

We find that after 4hrs of observations, linear regression obtains a median error of 49% (70% max), after 8hrs the error falls to 29%, and continues to fall, converging slowly to a median error of 23% after 24hrs. Our results, on the other hand start at 18% median error at 4hrs (25% max), after 8hrs (representing two time windows) we have an error of 13%, and after 24hrs this falls to 10%. At the end (after 68hrs) the regression error converges to a minimum of 17%, while our error at this point (the 16th time window) is at 1.0%.

These results are not particular to the DIGG data set: when applied to the VIMEO and YOUTUBE data sets, the median prediction error converges to 24.2% and 29.7% respectively. In comparison, for both data sets our method converges to a median error of under 3.5%. Looking more closely at the result, we believe that the relative poor performance of linear regression is best explained by looking at the coefficient of variation of the data ($\frac{\sigma}{\mu}$) as depicted in Figure 8. This shows that linear regression simply follows the dispersion of the data samples, while in contrast our method takes advantage of the correlations identified by the clustering process.

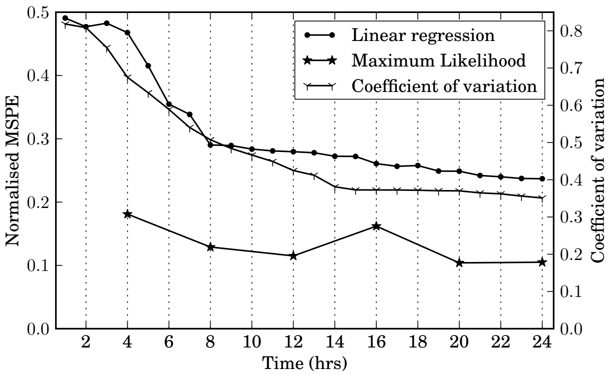

Figure 8: Normalised MSPE when using linear regression (dotted line), along side our method (starred line) for 1-step ahead prediction. From the figure we see that the linear regression tracks the coefficient of variation of the data (crossed line).

6. CONCLUSION AND FURTHER WORK

In this work we presented a novel method to characterise and predict the time evolution of popularity in user generated content. We defined an application agnostic feature space to capture the patterns of behaviour that contents display overtime. Our results show that: (i) we can accurately and in a fine-grained manner classify the behaviour of content based only on the number of hits it receives and (ii) based on our classification results we can construct accurate and again fine-grained predictive models. We have applied our method to three distinct data sets and shown that it is capable of performing better than the baselines of K-Means and linear regression as well as comparable work in the area.

Accurate classification and early prediction have direct relevance to a large number of application areas, including content caching [13], where network caching strategies could be improved by making them proactive; and advertising [22], where advertising strategies can be devised to react to the expected popularity of content (e.g., place a cheap ad on the page of a video that will become popular later).

As future work we are looking into how to extend our method to handle variable-length time windows and heterogeneous data mixtures. We are looking to move away from the offline training method presented in this paper to an online method that can dynamically build classifiers as new data comes in. More specifically, we are looking to replace the Affinity Propagation algorithm with mixture models in order to be able to represent centroids as PDFs and support online updating.

7. ACKNOWLEDGEMENTS

This publication is based on work performed in the framework of the Project COAST-ICT-248036, which is partially funded by the European Community. We would like to thank Christian Wallenta and Nishanth Sastry for sharing data sets with us and the reviewers for their helpful suggestions.

References

[1] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble. Why we search: visualizing and predicting user behavior. In *WWW'07*, 2007.

[2] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the influential bloggers in a community. In *WSDM '08*, 2008.

[3] W. Antweiler and M. Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *The Journal of Finance*, 59(3):1259–1294, Jun 2004.

[4] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan. Optimal content placement for a large-scale vod system. In *Co-NEXT '10*, 2010.

[5] S. Asur and B. A. Huberman. Predicting the future with social media. In *WI-IAT '10*, 2010.

[6] R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *ICWSM'10*, 2010.

[7] R. Bandari, S. Asur, and B. A. Huberman. The pulse of news in social media: Forecasting popularity. *CoRR*, abs/1202.0332, 2012.

[8] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. Characterizing and modeling popularity evolution of user-generated videos. In *IFIP Performace*, 2011.

[9] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC'07*, 2007.

[10] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *WWW '09*, 2009.

[11] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 105(15469), 2008.

[12] R. Cumby, S. Figlewski, and J. Hasbrouck. Forecasting Volatility and Correlations with EGARCH models. *Journal of Derivatives*, 1:51–63, 1993.

[13] J. Famaey, T. Wauters, and F. De Turck. On the merits of popularity prediction in multimedia content caching. In *Integrated Network Management*, 2011.

[14] F. Figueiredo, F. Benevenuto, and J. Almeida. The tube over time: Characterizing popularity growth of youtube videos. In *WSDM'11*, February 2011.

[15] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[16] T. Hogg and K. Lerman. Social dynamics of digg. *CoRR*, abs/1202.0031, 2012.

[17] L. Hong, O. Dan, and B. D. Davison. Predicting popular messages in twitter. In *WWW (Companion Volume)*, 2011.

[18] S. Jamali and H. Rangwala. Digging digg: Comment mining, popularity prediction, and social network analysis. In *WISM '09*, 2009.

[19] E. J. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *KAIS*, 7(3):358–386, 2005.

[20] S.-D. Kim, S.-H. Kim, and H.-G. Cho. Predicting the virtual temperature of web-blog articles as a measurement tool for online popularity. In *CIT '11*, 2011.

[21] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW '10*, 2010.

[22] H. Lakkaraju and J. Ajmera. Attention prediction on social media brand pages. In *CIKM '11*, 2011.

[23] D. Laniado and P. Mika. Making sense of twitter. In *ISWC'10*, 2010.

[24] J. G. Lee, S. Moon, and K. Salamatian. An approach to model and predict the popularity of online contents with explanatory factors. In *WI-IAT*, 2010.

[25] J. G. Lee, S. Moon, and K. Salamatian. Modeling and predicting the popularity of online contents with cox proportional hazard regression model. *Neurocomputing*, 76(1):134–145, 2012.

[26] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. *CoRR*, abs/1004.5354, 2010.

[27] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *SIGKDD'07*, 2007.

[28] D. A. Shamma, J. Yew, L. Kennedy, and E. F. Churchill. Viral actions: Predicting video view counts using synchronous sharing behaviors. In *ICWSM*, 2011.

[29] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, August 2010.

[30] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida. Predicting the popularity of online articles based on user comments. In *WIMS '11*, 2011.

[31] M. Tsagkias, W. Weerkamp, and M. de Rijke. News comments: exploring, modeling, and online prediction. In *ECIR'2010*, 2010.

[32] T. O. S. Tumasjan, P. G. Sandner, and I. M. Welpe. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *ICWSM'10*, 2010.

[33] Wikipedia. Viterbi algorithm, August 2012. [Online; accessed 30-Nov-2012].

[34] C. Williams and G. Gulati. What is a social network worth? Facebook and vote share in the 2008 presidential primaries. *Annual Meeting of the American Political Science Association*, 2008.

[35] H. F. X. Zhang and P. A. Gloor. Predicting asset value through twitter buzz. In *Advances in Collective Intelligence*, 2011.

[36] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM '11*, 2011.

[37] X. Zhang, H. Fuehres, and P. Gloor. Predicting Stock Market Indicators Through Twitter: "I hope it is not as bad as I fear". *Procedia - Social and Behavioral Sciences*, 26, Oct 2011.