Patterns of Temporal Variation in Online Media

Jaewon Yang
Stanford University
crucis@stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

Online content exhibits rich temporal dynamics, and diverse real-time user generated content further intensifies this process. However, temporal patterns by which online content grows and fades over time, and by which different pieces of content compete for attention remain largely unexplored.

We study temporal patterns associated with online content and how the content's popularity grows and fades over time. The attention that content receives on the Web varies depending on many factors and occurs on very different time scales and at different resolutions. In order to uncover the temporal dynamics of online content we formulate a time series clustering problem using a similarity metric that is invariant to scaling and shifting. We develop the K-Spectral Centroid (*K-SC*) clustering algorithm that effectively finds cluster centroids with our similarity measure. By applying an adaptive wavelet-based incremental approach to clustering, we scale *K-SC* to large data sets.

We demonstrate our approach on two massive datasets: a set of 580 million Tweets, and a set of 170 million blog posts and news media articles. We find that *K-SC* outperforms the K-means clustering algorithm in finding distinct shapes of time series. Our analysis shows that there are six main temporal shapes of attention of online content. We also present a simple model that reliably predicts the shape of attention by using information about only a small number of participants. Our analyses offer insight into common temporal patterns of the content on the Web and broaden the understanding of the dynamics of human attention.

Categories and Subject Descriptors

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

General Terms

Algorithm, Measurement

Keywords

Social Media, Time Series Clustering

WSDM'11, February 9–12, 2011, Hong Kong, China.

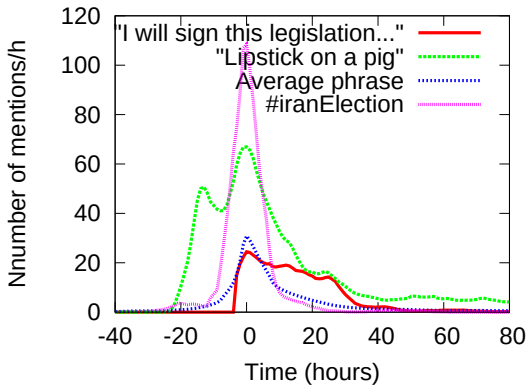


Figure 1: Short textual phrases and Twitter hashtags exhibit large variability in number of mentions over time.

1. INTRODUCTION

Online information is becoming increasingly dynamic and the emergence of online social media and rich user-generated content further intensifies this phenomena. Popularity of various pieces of content on the Web, like news articles [30], blog posts [21, 27], Videos [10], posts in online discussion forums [4] and product reviews [13], vary on very different temporal scales. For example, content on micro-blogging platforms, like Twitter [15, 34], is very volatile, and pieces of content become popular and fade away in a matter of hours. Short quoted textual phrases ("memes") rise and decay on a temporal scale of days, and represent the integral part of the "news cycle." [22] Temporal variation of named entities and general themes (like, "economy" or "Obama") exhibits variations at even larger temporal scale [3, 14, 31].

However, uncovering patterns of temporal variation on the Web is difficult because human behavior behind the temporal variation is highly unpredictable. Previous research on the timing of an individual's activity has reported that human actions range from random [26] to highly correlated [6]. Although the aggregate dynamics of individual activities tends to create seasonal trends or simple patterns, sometimes collective actions of people and the effects of personal networks result in a deviation from trends. Moreover, all individuals are not the same. For example, some act as "influentials" [33]. The overall picture of temporal activity on the Web is even more complex due to the interactions between individuals, small groups, and corporations. Bloggers and mainstream media are both producing and pushing new content into the system [16]. The content then gets adopted through personal social networks and discussed as it diffuses through the Web. Despite extensive qualitative research, there has been little work about temporal patterns by which content grows and fades over time and by which different pieces of content compete for attention during this process.

Temporal patterns of online content. Here we study what temporal patterns exist in the popularity of content in social media. The popularity of online content varies rapidly and exhibits many different temporal patterns. We aim to uncover and detect such temporal patterns of online textual content. More specifically, we focus on the propagation of the hashtags on Twitter, and the quotation of short textual phrases in the news articles and blog-posts on the Web. Such content exhibits rich temporal dynamics [21, 22, 26] and is a direct reflection of the attention that people pay to various topics. Moreover, the online media space is occupied by a wide spectrum of very distinct participants. First of all, there are many personal blogs and Twitter accounts, with a relatively small readership. Secondly, there are professional bloggers and small community-driven or professional online media sites (like, The Huffington Post) that have specialized interests and respond quickly to events. Finally, mainstream mass media, like TV stations (e.g., CNN), large newspapers (e.g., The Washington Post) and news agencies (e.g., Reuters) all produce content and push it to the other contributors mentioned above. We aim to understand what kinds of temporal variations are exhibited by online content, how different media sites shape the temporal dynamics, and what kinds of temporal patterns they produce and influence.

The approach. We analyze a set of more than 170 million news articles and blog posts over a period of one year. In addition, we examine the adoption of Twitter hashtags in a massive set of 580 million Twitter posts collected over a 8 month period. We measure the attention given to various pieces of content by tracing the number of mentions (i.e., volume) over time. We formulate a time series clustering problem and use a time series shape similarity metric that is invariant to the total volume (popularity) and the time of peak activity. To find the common temporal patterns, we develop a K-Spectral Centroid (*K-SC*) clustering algorithm that allows the efficient computation of cluster centroids under our distance metric. We find that *K-SC* is more useful in finding diverse temporal patterns than the K-means clustering algorithm [17]. We develop an incremental approach based on Haar Wavelets to improve the scalability of *K-SC* for high-dimensional time series.

Findings. We find that temporal variation of popularity of content in online social media can be accurately described by a small set of time series shapes. Surprisingly, we find that both of the adoption of hashtags in Twitter and the propagation of quoted phrases on the Web exhibit nearly identical temporal patterns. We find that such patterns are governed by a particular type of online media. Most press agency news exhibits a very rapid rise followed by a relatively slow decay. Whereas, bloggers play a very important role in determining the longevity of news on the Web. Depending on when bloggers start participating in the online discourse the news story may experience one or more rebounds in its popularity.

Moreover, we present a simple predictive model which, based on timings of only few sites or Twitter users, predicts with 75% accuracy which of the temporal patterns the popularity time series will follow. We also observe complex interactions between different types of participants in the online discourse.

Consequences and applications. More generally, our work develops scalable computational tools to further extend understanding of the roles of different participants play in the online media space. We find that the collective behavior of various participants governs how we experience new content and react to it. Our results have direct applications for predicting the overall popularity and temporal trends exhibited by the online content. Moreover, our results can be used for better placing of content to maximize clickthrough rates [5] and for finding influential blogs and Twitters [23].

2. FINDING TEMPORAL PATTERNS

In this section, we formally define the problem and then propose K-Spectral Centroid (*K-SC*) clustering algorithm.

We start by assuming that we are given a time series of mentions or interactions with a particular piece of contents. This could be a time series of clicks or plays of a popular video on YouTube, the number of times an article on a popular newspaper website was read, or the number of times that a popular hashtag in Twitter was used. Now we want to find patterns in the temporal variation of time series that are shared by many pieces of content.

We formally define this as a problem of clustering time series based on their shape. Given that online content has large variation in total popularity and occurs at very different times, we will first adopt a time series similarity metric that is invariant to scaling and shifting. Based on this metric, we develop a novel algorithm for clustering time series. Finally, we present a speed-up technique that greatly reduces the runtime and allows for scaling to large datasets.

2.1 Problem definition

We are given N items of contents and for each item i we have a set of traces of the form $(s_j, t_j)_i$, which means that site s_j mentioned item i at time t_j. From these N traces, we then construct a discrete time series $x_i(t)$ by counting the number of mentions of item i at time interval t. Simply, we create a time series of the number of mentions of item i at time t where t's measured in some time unit, e.g., hours. Intuitively, x_i measures the popularity or attention given to item i over time. For convenience let us also assume that all time series x_i have the same length, L. The shape of the time series x_i simply represents how the popularity or attention to item i changed over time. We then aim to group together items so that item i's in the same group have a similar shape of the time series x_i. This way we can infer what items have a similar temporal pattern of popularity, and we can then consider the center of each cluster as the representative common pattern of the group.

2.2 Measure of time series shape similarity

In order to perform the clustering based on the shape of the item popularity curve, we first discuss how we can measure the shape similarity of two time series.

Figure 1 shows an example of temporal variability in the number of mentions of different textual phrases and Twitter hashtags. We plot the average popularity curve of 1,000 phrases with largest overall volume (after aligning them so that they all peak at the same time). The figure shows two individual phrases. First is the quote from U.S. president Barack Obama about the stimulus bill: *"I will sign this legislation into law shortly and we'll begin making the immediate investments necessary to put people back to work doing the work America needs done."* and the second is the *"Lipstick on a pig"* phrase from the 2008 U.S. presidential election campaign. Notice the large difference among the patterns. Whereas average phrases almost symmetrically rise and fade, the "Lipstick on a pig" has two spikes with the second being higher than the first, while the stimulus bill phrase shows a long streak of moderate activity.

A wide range of measures of time series similarity and approaches to time series clustering have been proposed and investigated. However, the problem we are addressing here has several characteristics that make our setting somewhat different and thus common metrics such as Euclidean or Dynamic Time Warping are inappropriate in our case for at least two reasons. First, if two time series have very similar shape but different overall volume, they should still be considered similar. Thus, scaling the time series on the y-axis should not change the similarity. Second, different items appear and spike at different times. Again, even though two time series

may be shifted, they should be considered similar provided that they have similar shape. Thus, translating time series on the time axis should not change the similarity between the two time series.

Time series similarity measure. As described above we require a time series similarity measure that is invariant to scaling and translation and allows for efficient computation.

Since the time series of popularity of items on the Web typically exhibit bursty and spiky behavior [10], one could address the invariance to translation by aligning the time series to peak at the same time. Even so, many challenges remain. For example, what exactly do we mean by "peak"? Is it the time of the peak popularity? How do we measure popularity? Should we align a smoothed version of the time series? How much should we smooth?

Even if we assume that somehow peak alignment works, the overall volume of distinct time series is too diverse to be directly compared. One might normalize each time series by some (necessarily arbitrary) criteria and then apply a simple distance measure such as Euclidian norm. However, there are numerous ways to normalize and scale the time series. We could normalize so that total time series volume is 1, that the peak volume is 1, etc.

For example, Figure 2 illustrates the ambiguity of choosing a time series normalization method. Here we aim to group time series S1, . . . , S4 in two clusters, where S1 and S2 have two peaks and S3 and S4 have only one sharp peak. First we align and scale time series by their peak volume and run the K-Means algorithm using Euclidean distance (bottom figures in (B) and (C)). (We choose this time series normalization method because we found it to perform best in our experiments in Section 3.) However, the K-Means algorithm identifies wrong clusters {S2, S3, S4} and {S4}. This is because the peak normalization tends to focus on the global peak and ignores other smaller peaks (figure (B)). To tackle this problem we adopt a different time series distance measure and develop a new clustering algorithm, which does not suffer from such behavior, i.e., it groups together the two peaked time series S1 and S2, and puts single peaked time series S3 and S4 in the other cluster.

First, we adopt a distance measure that is invariant to scaling and translation of the time series [9]. Given two time series x and y, the distance $\hat{d}(x, y)$ between them is defined as follows:

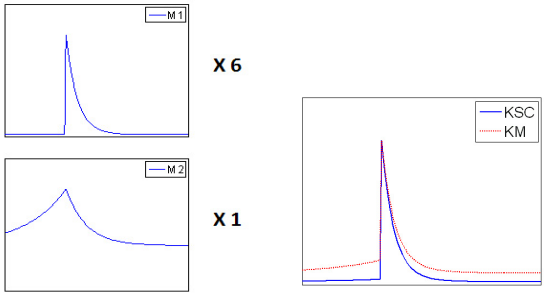
$$\hat{d}(x, y) = \min_{\alpha, q} \frac{||x - \alpha y_{(q)}||}{||x||} \quad (1)$$

where $y_{(q)}$ is the result of shifting time series y by q time units, and $||\cdot||$ is the l_2 norm. This measure finds the optimal alignment (translation q) and the scaling coefficient α for matching the shapes of the two time series. The computational complexity of this operation is reasonable since we can find a closed-form expression to compute the optimal α for fixed q. With q fixed, $\frac{||x-\alpha y_{(q)}||}{||x||}$ is a convex function of α, and therefore we can find the optimal α by setting the gradient to zero: $\alpha = \frac{x^T y_{(q)}}{||y_{(q)}||^2}$. Also, note that $\hat{d}(x, y)$ is symmetric in x and y (refer to extended version for details [1]).

Whereas one can quickly find the optimal value of α, there is no simple way to find the optimal q. In practice we first find alignment q' that makes the time series to peak at the same time and then search for optimal q around q'. In our experiments, the starting point q' is very close to the optimal since most of our time series have a very sharp peak volume, as shown in Section 3. Therefore, this heuristic finds q that is close to the optimal very quickly.

2.3 K-Spectral Centroid Clustering

Next, we present the K-Spectral Centroid (*K-SC*) clustering algorithm that finds clusters of time series that share a distinct temporal pattern. *K-SC* is an iterative algorithm similar to the classical



(a) Time Series (b) Cluster center

Figure 3: (a) A cluster of 7 single-peaked time series: 6 have the shape M1, and one has the shape M2. (b) The cluster centers found by K-Means (KM) and KSC. The KSC cluster center is less affected by the outlier and better represents the common shape of time series in the cluster.

K-means clustering algorithm [17] but enables efficient centroid computation under the scale and shift invariant distance metric that we use. K-means iterates a two step procedure, the assignment step and the refinement step. In the assignment step, K-means assigns each item to the cluster closest to it. In the refinement step the cluster centroids are then updated. By repeating these two steps, K-means minimizes the sum of the squared Euclidean distances between the members of the same cluster. Similarly, *K-SC* alternates the two steps to minimize the sum of squared distances, but the distance metric is not Euclidean but our distance metric $\hat{d}(x, y)$. As K-means simply takes the average over all items in the cluster as the cluster centroid, this is inappropriate when we use our metric $\hat{d}(x, y)$. Therefore, we develop a K-Spectral Centroid (*K-SC*) clustering algorithm which appropriately computes cluster centroids under time series distance metric $\hat{d}(x, y)$.

For example, in Figure 2, *K-SC* discovers the correct clusters (blue group in panel (A)). When $\hat{d}(x, y)$ is used to compute the distance between S2 and the other time series, $\hat{d}(x, y)$ finds the optimal scaling of other time series with respect to S2. Then, S1 is much closer to S2 than S3 and S4, as it can match the variation in the second peak of S2 with the proper scaling (panel B). Because of accurate clustering, *K-SC* computes the common shape shared by the time series in the cluster (panel C).

Moreover, even if K-means and *K-SC* find a same clustering, the cluster center found by *K-SC* is more informative. In Figure 3, we show a cluster of single-peaked time series, and try to observe the common shape of time series in the cluster by computing a cluster center by K-means and *K-SC*. Since the cluster has 6 time series of the same shape (M1) and one outlier (M2), we want the cluster center to be similar to M1. Observe that *K-SC* finds a better center than K-means. As K-means computes the average shape of time series for a cluster center, the resulting center is sensitive to outliers. Whereas, *K-SC* scales each time series differently to find a cluster center, and this scaling decreases the influence of outliers.

More formally, we are given a set of time series x_i, and the number of clusters K. The goal then is to find for each cluster k an assignment C_k of time series to the cluster, and the centroid μ_k of the cluster that minimize a function F defined as follows:

$$F = \sum_{k=1}^{K} \sum_{x_i \in C_k} \hat{d}(x_i, \mu_k)^2. \quad (2)$$

We start the *K-SC* algorithm with a random initialization of the cluster centers. In the assignment step, we assign each x_i to the closest cluster, based on $\hat{d}(x, y)$. This is identical to the assign-

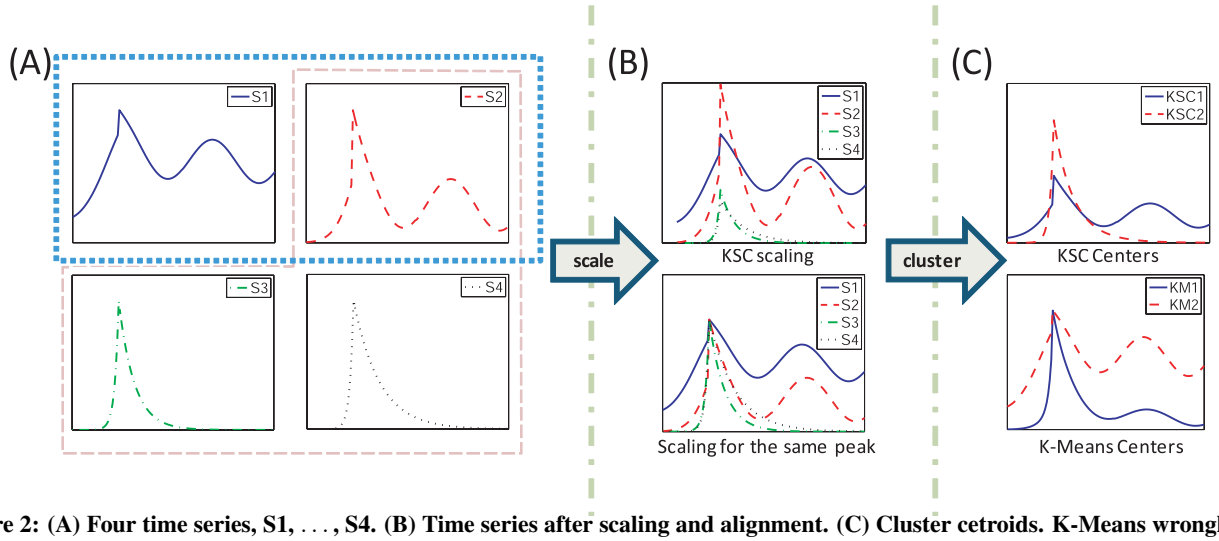


Figure 2: (A) Four time series, S1, . . . , S4. (B) Time series after scaling and alignment. (C) Cluster cetroids. K-Means wrongly puts {S1} in its own cluster and {S2, S3, S4} in the second cluster, while *K-SC* nicely identifies clusters of two vs. single peaked time series.

ment step of K-means except that it uses a different distance metric. After finding the cluster membership of every time series, we update the cluster centroid. Simply updating the new center as the average of all members of the cluster is inappropriate, as this is not the minimizer of the sum of squared distances to members of the cluster (under $\hat{d}(x, y)$). The new cluster center μ_k^* should be the minimizer of the sum of $\hat{d}(x_i, \mu_k)^2$ over all $x_i \in C_k$:

$$\mu_k^* = \arg\min_{\mu} \sum_{x_i \in C_k} \hat{d}(x_i, \mu)^2. \quad (3)$$

Since *K-SC* is an iterative algorithm it needs to update the cluster centroids many times before it converges. Thus it is crucial to find an efficient way to solve the above minimization problem.

Next, we show that Eq. 3 has a unique minimizer that can be expressed in a closed form. We first combine Eqs. 1 and 3.

$$\mu_k^* = \arg\min_{\mu} \sum_{x_i \in C_k} \min_{\alpha_i, q_i} \frac{||\alpha_i x_{i(q_i)} - \mu||^2}{||\mu||^2}$$

Since we find the optimal translation q_i in the assignment step of *K-SC*, consider (without the loss of generality) that x_i is already shifted by q_i. We then replace α_i with its optimal value (Sec. 2.2):

$$\mu_k^* = \arg\min_{\mu} \frac{1}{||\mu||^2} \sum_{x_i \in C_k} ||\frac{x_i^T \mu}{||x_i||^2} x_i - \mu||^2$$

We flip the order of $x_i^T \mu x_i$ and simplify the expression:

$$\mu_k^* = \arg\min_{\mu} \frac{1}{||\mu||^2} \sum_{x_i \in C_k} ||\frac{x_i x_i^T \mu}{||x_i||^2} - \mu||^2$$

$$= \arg\min_{\mu} \frac{1}{||\mu||^2} \sum_{x_i \in C_k} ||(\frac{x_i x_i^T}{||x_i||^2} - I)\mu||^2$$

$$= \arg\min_{\mu} \frac{1}{||\mu||^2} \mu^T \sum_{x_i \in C_k} (I - \frac{x_i x_i^T}{||x_i||^2})\mu$$

Finally, substituting $\sum_{x_i \in C_k} (I - \frac{x_i x_i^T}{||x_i||^2})$ by M leads to the following minimization problem:

$$\mu_k^* = \arg\min_{\mu} \frac{\mu^T M \mu}{||\mu||^2}. \quad (4)$$

Algorithm 1 *K-SC* clustering algorithm: K-SC(**x**,**C**, K)

Require: Time series x_i, $i = 1, 2, ..., N$, The number of clusters K, Initial cluster assignments $C = \{C_1, .., C_K\}$
  repeat
    $\hat{C} \leftarrow C$
    for $j = 1$ to K **do** {Refinement step}
      $M \leftarrow \sum_{i \in C_j} (I - \frac{x_i x_i^T}{||x_i||^2})$
      $\mu_j \leftarrow$ The smallest eigenvector of M
      $C_j \leftarrow \emptyset$
    end for
    for $i = 1$ to N **do** {Assignment step}
      $j^* \leftarrow \arg\min_{j=1,..,K} \hat{d}(x_i, \mu_j)$
      $C_{j^*} \leftarrow C_{j^*} \cup \{i\}$
    end for
  until $\hat{C} = C$
  return $C, \mu_1, ..., \mu_K$

The solution of this problem is the eigenvector u_m corresponding to the smallest eigenvalue λ_m of matrix M [12]. If we transform μ by multiplying the eigenvectors of M, then $\mu^T M \mu$ is equivalent to the weighted sum of the eigenvalues of M, whose smallest element is $\lambda_m ||\mu||^2$. Therefore, the minimum of Eq. 4 is λ_m and letting $\mu = u_m$ achieves the minimum. As M is given by x_i's, we simply find the smallest eigenvector of M for the new cluster center μ_k^*. Since μ_k^* minimizes the spectral norm of M, we call μ_k^* the *Spectral Centroid*, and call the whole algorithm the K-Spectral Centroid (*K-SC*) clustering (Algorithm 1).

2.4 Incremental K-SC algorithm

Since our time series are usually quite long and go into hundreds and sometime thousands of elements, scalability of *K-SC* is important. Let us denote the number of time series by N, the number of clusters by K, and the length of the time series by L. The refinement step of *K-SC* computes M first, and then finds its eigenvectors. Computing M takes $O(L^2)$ for each x_j, and finding the eigenvectors of M takes $O(L^3)$. Thus, the runtime of the refinement step is dominated by $O(\max(NL^2, KL^3))$. However, the assignment step takes only $O(KNL)$, and therefore the complexity of one iteration of *K-SC* is $O(\max(NL^2, KL^3))$.

A cubic complexity in L is clearly an obstacle for *K-SC* to be used on large datasets. Moreover, there is another reason why ap-

Algorithm 2 Incremental K-SC

Require: Time series x_i, $i = 1, 2, ..., N$, The number of clusters K, Initial assignments $C = \{C_1, .., C_K\}$, Start level S, The length of x_i's L
  for $i = 1$ to N **do**
    $z_i \leftarrow$ Discrete Haar Wavelet Transform(x_i)
  end for
  for $j = S$ to $\log_2(L)$ **do**
    for $i = 1$ to N **do**
      $y_i \leftarrow$ Inverse Discrete Haar Wavelet Transform($z_i(1 : 2^j)$)
      $\{z_i(1 : n)$ means the first n elements of $z_i\}$
    end for
    $(C, \mu_1, ..., \mu_K) \leftarrow$ K-SC(**y**,C,K)
  end for
  return $C, \mu_1, ..., \mu_K$

plying *K-SC* directly to high dimensional data is not desirable. Like K-means, *K-SC* is a greedy hill-climbing algorithm for optimizing a non-convex objective function. Since *K-SC* starts at some initial point and then greedily optimize the objective function, the rate of convergence is very sensitive to the initialization of the cluster centers [28]. If the initial centers are poorly chosen, the algorithm may be very slow, especially if N or L are large.

We address these two problems by adopting an approach similar to Incremental K-means [28] which utilizes the multi-resolution property of the Discrete Haar Wavelet Transform (DHWT) [7]. It operates as follows: the first few coefficients of DHWT decomposition contain an approximation of the original time series at very coarse resolution, while additional coefficients show information in higher resolution. Given a set of time series x, we compute the Haar Wavelet decomposition for every time series x_i. The DHWT computation is fast, taking $O(L)$ for each time series.

By taking the first few coefficients of the Haar Wavelet decomposition of the time series, we approximate the time series at very coarse granularity. Thus, we first cluster the coarse-grained representations of the time series using the *K-SC* algorithm. In this case *K-SC* will be run very quickly and will also be robust with respect to random initialization of the cluster centers. Then, we move to the next level of resolution of the time series and use the assignments from the previous iteration of *K-SC* as the initial assignments at the current level. We repeat this procedure until we reach the full resolution of the time series, i.e., all wavelet coefficients are used. Even when we are working with full resolution time series, *K-SC* converges much faster than if we started *K-SC* from a random initialization, since we start very closely from the optimal point. Alg. 2 gives the pseudo-code of the Incremental *K-SC* algorithm.

3. EXPERIMENTAL RESULTS

Next we describe the data, experimental setup, and evaluation of the clusters we find. We describe our findings in Section 4.

3.1 Experimental setup

First we apply our algorithm to a dataset of more than 172 million news articles and blog posts collected from 1 million online sources during a one-year period from September 1 2008 to August 31 2009. We use the MemeTracker [22] methodology to identify short quoted textual phrases and extract more than 343 million short phrases. To observe the complete lifetime of a phrase, we only keep phrases that first appeared after September 5. This step removes the phrases quoted repeatedly without a reference to a certain event, such as "I love you."

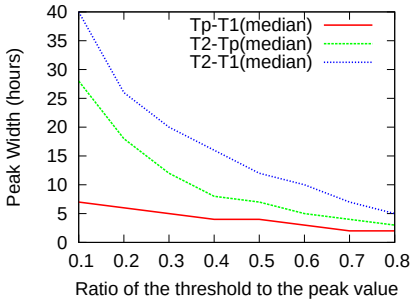


Figure 4: Width of the peak of the time series versus the fraction of the threshold we set.

After these preprocessing steps, we choose the 1,000 most frequent phrases and for each phrase create a time series of the number of mentions (i.e., volume) per unit time interval. To reduce rapid fluctuation in the time series, we apply Gaussian kernel smoothing.

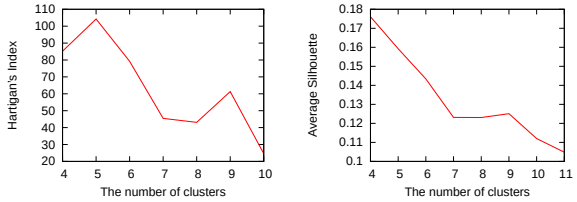
Choosing the time series length. In principle the time series of each phrase contains L =8,760 elements (i.e., the number of hours in 1 year). However, the volume of phrases tends to be concentrated around a peak [22], and thus taking such a long time series would not be a good idea. For example, we measure the similarity between two phrases that are actively quoted for one week and abandoned for the rest of the time. We would be interested mainly in the differences of them during their active one week. However, the differences in inactive periods may not be zero due to noise, and these small differences can dominate the overall similarity since they are accumulated over a long period. Therefore, we truncate the time series to focus on the "interesting" part of the time series.

To set the length of truncation, We measure how long the peak popularity spreads out: let T_p be the time when the phrase reached peak volume, and let v_p be the phrase volume at that time (i.e., number of mentions at hour T_p). For a threshold, xv_p (for $0 < x < 1$), we go back in time from T_p of a given phrases and record as $T_1(x)$ the *last* time index when the phrase's volume gets below xv_p. Next, we go forward in time from T_p and mark the *first* time index when its volume gets below threshold as $T_2(x)$. Thus, $T_1(x)$ measures the width of the peak from the left, and $T_2(x)$ measures the width of the peak from the right.

Figure 4 plots the median value of $T_p - T_1(x)$, $T_2(x) - T_p$ and $T_2(x) - T_1(x)$ as a function of x. We note that most phrases maintain nontrivial volume for a very short time. For example, it takes only 40 hours for the phrase volume to rise from 10% of the peak volume, reach the peak, and fall again below 10% of the peak volume. In general, the volume curve tends to be skewed to the right (i.e., $T_p - T_1(x)$ and $T_2(x) - T_p$ are far for small x). This means that in general the volume of phrases rather quickly reaches its peak and then slowly falls off.

Given the above results, we truncate the length of the time series to 128 hours, and shift it such that it peaks at the 1/3 of the entire length of the time series (i.e., the 43th index).

Choosing the number of clusters. The *K-SC* algorithm, like all the other variants of K-means, requires the number of clusters to be specified in advance. Although it is an open question how to choose the most appropriate number of clusters, we measure how the quality of clustering varies with the number of clusters. We ran *K-SC* with a different number of clusters, and measured Hartigan's Index and the Average Silhouette [17]. Figure 3.1 shows the values of the two measures as a function of the number of clusters. The higher the value the better the clustering. The two metrics do not



(a) Hartigan's Index (b) Average Silhouette

Figure 5: Clustering quality versus the number of clusters.

Method	F (lower is better)	$\sum \hat{d}(\mu_i, \mu_j)^2$ (higher is better)
KM-NS	122.12	2.12
KM-P	76.25	3.94
K-SC	**64.75**	**4.53**

Table 1: Cluster quality. KM-NS: K-means with peak alignment but no scaling; KM-P: K-means with alignment and scaling. Notice *K-SC* well improves over K-means in both criteria.

necessarily agree with each other, but Figure 3.1 suggests that a lower value of K gives better results. We chose $K = 6$ as the number of clusters. We also experimented with $K \in \{3, \ldots, 12\}$ and found that clusterings are quite stable. Even when $K = 12$, all of 12 clusters are essentially the variants of clusters that we find using $K = 6$ (refer to extended version of the paper [1] for details).

3.2 Performance of K-SC Algorithm

Having described the data preprocessing and *K-SC* parameter settings we evaluate the performance of our algorithm in terms of quality and speed. We compare the result of *K-SC* to that of K-means, which uses the Euclidean time series distance metric. In particular, we evaluate two variants of K-means that differ in the way we scale and align the time series. First, we align the time series to peak at the same time but do not scale them in the y-axis. In the second variant we not only align the time series but also scale them in the y-axis so that they all have the peak volume of 100.

For each algorithm we compute two performance metrics: (a) the value of the objective function F as defined in Equation 2, and (b) the sum of the squared distances between the cluster centers, $\sum \hat{d}(\mu_i, \mu_j)^2$. Function F measures the compactness of the cluster, while the distances between the cluster centers measure the diversity of the clusters. Thus, a good clustering has a low value of F and large distances between the cluster centers. Table 1 presents the results and shows that *K-SC* achieves the smallest value of F, and the biggest distance between the clusters. Note that it is not trivial that *K-SC* achieves a bigger value of $\sum \hat{d}(\mu_i, \mu_j)^2$ than K-means, because *K-SC* does not optimize $\sum \hat{d}(\mu_i, \mu_j)^2$. Manual inspection of the clusters from each algorithm also suggests that K-means clusters are harder to interpret than *K-SC* clusters, and their shapes are less diverse than those of *K-SC* clusters. We also experimented with normalizing the sum of each time series to the same value and standardization of the time series but were unable to to make K-means work well (refer to [1] for details). We also note that *K-SC* does not require any kind of normalization, and performs better than K-means with the best normalization method.

Scalability of *K-SC*. Last, we analyze the effect of the Wavelet-based incremental clustering procedure in the runtime of *K-SC*. In our data set, we use relatively short time series ($L = 128$) and *K-SC* can easily handle them. In the following experiment we show that *K-SC* is generally applicable even if time series would span for more than hundreds of time indexes.

We assume that we are dealing with much longer time series. In particular, we take $L = 512$ instead of $L = 128$ for truncating the time series, and run Incremental *K-SC* five times increasing the number of time series from 100 to 3,000. As a baseline, we perform *K-SC* without the incremental wavelet-based approach. Figure 6 shows the average values and the variances of the runtime with respect to the size of the dataset. The incremental approach reduces the runtime significantly. While the runtime of naive *K-SC* grows very quickly with the size of the data, incremental *K-SC* grows much slower. Furthermore, notice also that the error bars on the runtime of incremental *K-SC* are very small. This means that incremental *K-SC* is also much more robust to the initialization of the cluster centers than naive *K-SC* in that it takes almost the same time to perform the clustering regardless of the initial conditions.

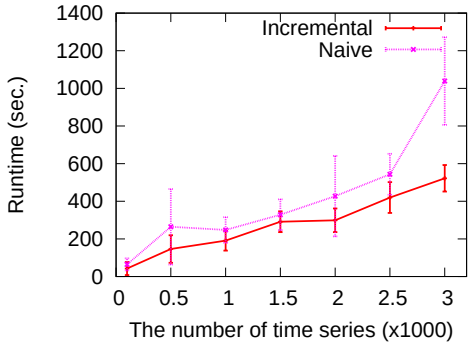


Figure 6: Runtime of Naive *K-SC* and Incremental *K-SC*.

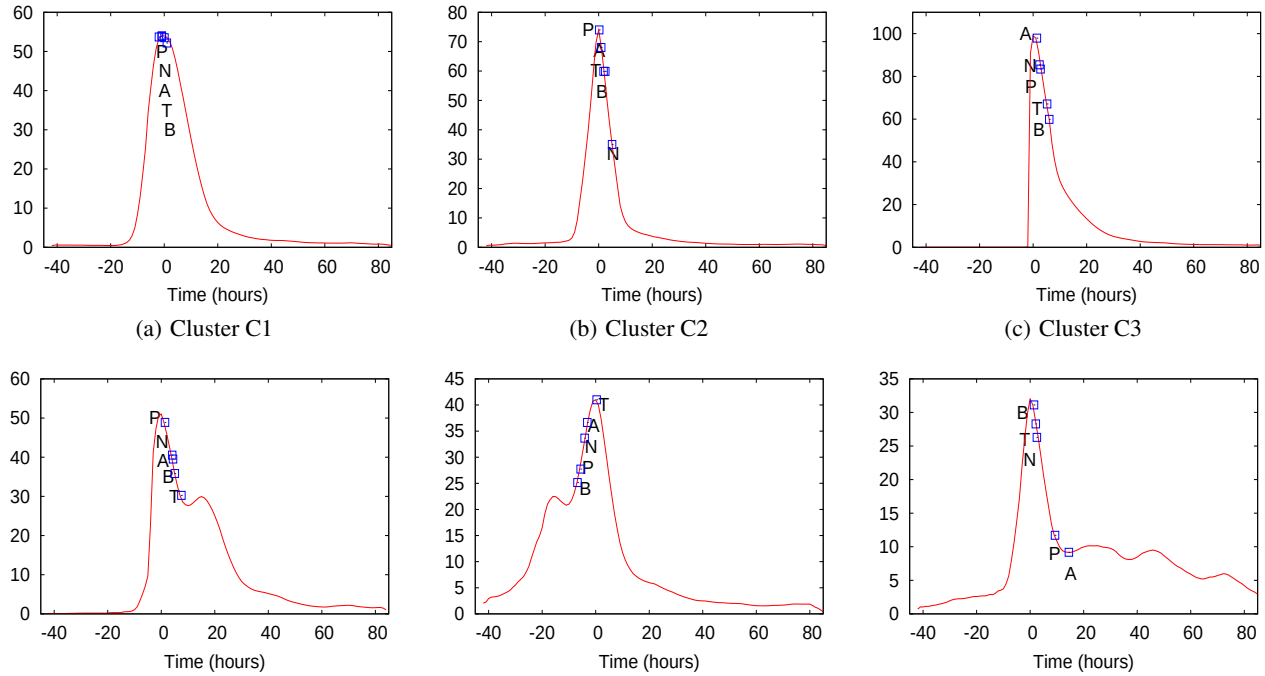
4. EXPERIMENTS ON MEMETRACKER

Cluster	C1	C2	C3	C4	C5	C6
f_c	28.7%	23.2%	18.1%	13.3%	10.3%	6,4%
V	681	704	613	677	719	800
V_{128}	463	246	528	502	466	295
V_P	54	74	99	51	41	32
V_P/V	7.9%	10.6%	16.2%	7.5%	5.7%	4.0%
L_b	1.48	0.21	1.30	1.97	1.59	-0.34
FB	33.3%	42.9%	29.1%	36.2%	45.0%	53.1%
FB_{128}:	27.4%	35.6%	28.5%	32.6%	36.5%	53.4%

Table 2: Statistics of the clusters from Figure 7. f_c: Fraction of phrases in the cluster, V: Total volume (over 1 year) V_{128}: Volume around the peak (128 hours), V_P: Volume at the peak (1 hour), V_P/V: Peak to total volume, L_b: Blog Lag (hours), FB: Fraction of blog volume over 1 year, FB_{128}: Fraction of blog volume around the peak.

Now we describe the temporal patterns of the Memetracker phrases as identified by our *K-SC* algorithm. Figure 7 shows the cluster centers for $K = 6$ clusters, and Table 2 gives further descriptive statistics for each of the six clusters. We order the clusters so that $C1$ is the largest and $C6$ is the smallest. Notice the high variability in the cluster shapes. The largest three clusters in the top row exhibit somewhat different but still very spiky temporal behavior, where the peak lasts for less than 1 day. On the other hand, in the latter three clusters the peak lasts longer than one day. Although we present the clustering of the top 1,000 most frequent phrases, more than half of the phrases lose their attention after a single day.

The biggest cluster, $C1$, is the most spread out of all the "single peak" clusters that all share the common quick rise followed by a monotone decay. Notice that C1 looks very much like the average of all the phrases in Figure 1. This is natural because the average pattern would be likely to occur in a large number of phrases.



(a) Cluster C1 (b) Cluster C2 (c) Cluster C3

(d) Cluster C4 (e) Cluster C5 (f) Cluster C6

Figure 7: Clusters identified by *K-SC*. We also plot the average of the time when a particular type of website first mentions the phrases in each cluster. The horizontal position corresponds to the average time of the first mention. P: professional blog, N: newspaper, A: news agency, T: TV station, B: blog aggregator.

Cluster $C2$ is narrower and has a quicker rise and decay than $C1$. Whereas $C1$ is not entirely symmetric, the rise and decay of $C2$ occur at around the same rate. $C3$ is characterize by a super quick rise just 1 hour before the peak and a slower decay than $C1$ and $C2$. The next two clusters, $C4$ and $C5$, experience a rebound in their popularity and have two peaks about 24 hours apart. While $C4$ experiences a big peak on the first day and a smaller peak on the second day, $C5$ does exactly the opposite. It has a small peak on the first day and a larger one on the second day. Finally, phrases in Cluster $C6$ stay popular for more than three days after the peak, with the height of the local peaks slowly declining.

Cluster statistics. We also collected statistics about the phrases in each of the clusters (Table 2). For each cluster we compute the following median statistics over all phrases in the cluster: the total phrase volume over the entire 1 year period, volume in the 128 hour period around the peak, the volume during the hour around the peak, and the ratio between the two. We also quantify the Blog Lag as follows: we use the classification of Google News and label all the sites indexed by Google News as mainstream media and all the other sites as blogs [22]. Then for each phrase we define Blog Lag as the difference between the median of the time when news media quote the phrase and the median of the time when blogs quote the phrase. Note that positive Blog Lag means that blogs trail mainstream media. At last, we compute the ratio of volume coming from the blogs to the total phrase volume for the two time horizons, a one year and 128 hours around the peak.

We find that cluster $C1$ shows moderate values in most categories, confirming that this cluster is closest to a behavior of a typical phrase. Cluster $C2$ and $C3$ have sharp peaks, but their total volume around the peak is significantly different. This difference comes from the reaction of mainstream media. Although both clusters have higher peak than other clusters, 74 and 99 respectively, the volume of $C2$ coming from mainstream media is only 30% of that of $C3$. Interestingly enough, the phrases in $C3$ have the largest volume around the peak and also far the highest peak volume. The dominant force here is the attention from news media, because $C3$ shows the smallest fraction of the blog volume. The next two clusters, $C4$ and $C5$, have two peaks and are the mirror versions of each other. They also show similar values for most categories. The only difference is that $C4$ has bigger volume from mainstream media and gets mentions from blogs for a longer time, which results in the larger value of the total volume around the peak. The last cluster, $C6$, is the most interesting one. The phrases in $C6$ have the highest overall volume, but the smallest volume around the peak. It seems that many phrases in this cluster correspond to hot topics on which the blogosphere discusses for several days. Another interesting aspect of $C6$ is that the role of blogs in the cluster. It has distinctively high fraction of the blog volume, and the only cluster where bloggers actually lead mainstream media.

Number of websites	50	100	200	300
Temporal features	76.62%	81.23%	88.73%	95.75%
Volume features	70.71%	77.05%	86.62%	95.59%
TF-IDF features	70.12%	77.05%	87.04%	94.74%

Table 3: Classification accuracy of the clusters with a different set of features. See the main text for description.

Modeling the time series shape. Our analysis so far shows that the clusters have very different characteristics as well as diverse shapes. Motivated by this result, we conduct a temporal analysis for an individual website with respect to each cluster. We hypothesize that if a certain website mentions the phrase this will create distinctive temporal signature of the phrases. For example, from Table 2 we see that blogs tend to mention the phrases in C6 earlier. If the hypothesis is true, therefore, then we should be able

to predict to which cluster a phrase belongs to solely based on the information about which websites mentioned the phrase. That is, based on which sites mentioned the phrase we would like to predict the temporal pattern the phrase will exhibit.

For each phrase we construct a feature vector by recording for each website the time when it first mentioned the phrase. If a website does not mention a phrase, we consider it as a missing data. We impute the missing time as the average of the times when the website first mentioned phrases. For comparison we also construct two other feature vectors. For each website, we first record the fraction of the phrase volume created by that website. In addition, we treat every phrase as a "document" and every site as a "word", and then compute the TF-IDF score [29] of each phrase.

Given feature vectors, we learn six separate logistic regression classifiers so that the i-th classifier predicts whether the phrase belongs to the i-th cluster or not. Moreover, we vary the length of feature vectors (i.e., the number of the sites used by the classifier), by choosing the largest websites in terms of phrase volume. We report the average classification accuracy in Table 3. By using the information from only 100 largest websites, we can predict the shape of the phrase volume over time with the accuracy of 81%. Among the three types of features, we observe that the features based on the temporal information give best performance.

Time series shape and the types of websites. Encouraged by the above results, we further investigate how websites contribute to the shape of the phrase volume and interact each other in each cluster. For the analysis we manually chose a set of 12 representative websites. We manually classified them into five categories based on the organization or the role they play in the media space: Newspapers, Professional blogs, TV, News agencies and Blogs (refer to the full version for the used list of websites [1]).

First, we repeat the classification task from previous section but now with only the 12 websites. Surprisingly, we obtain an average classification accuracy of 75.2%. Moreover, if we choose 12 websites largest by total volume we obtain accuracy of 73.7%. By using the l_1-regularized logistic regression to select the optimal (i.e., most predictive) set of 12 websites we obtain the accuracy of 76.0%.

Second, using the classification of websites into 5 groups we compute the time when websites of that type tend to mention the phrases in particular cluster. Figure 7 shows the measured average time for each type of website. Letters correspond to the types of websites and the horizontal position of letters corresponds to the average time of the first mention. For example, it is the professional bloggers (P) that first mention the phrases in Cluster $C1$ and $C2$. For phrases in $C1$, this is followed by newspapers (N), news agencies (A), then television (T) and finally by bloggers (B). In $C2$ the order is a bit different but the point is that all types mention the phrase very close together. Interestingly, for the phrases in $C3$ news agencies (A) mention the phrase first. Notice that $C3$ has the heaviest tail among all the single-peak clusters. It is probably due to the fact that many different organizations subscribe and publish the articles from news agencies, and thus the phrases in $C3$ slowly percolates into online media. We observe the process of percolation by looking at the time values in Figure 7: starting from news agencies to newspapers and professional bloggers, and finally to TV stations and small bloggers. In $C4$ and $C5$, we note that it is the bloggers that make the difference. In $C4$ bloggers come late and create the second lower spike, while in $C5$ bloggers (both small ones and professional ones) are the earliest types. Finally, the phrases in $C6$ gain the attention mainly on the blogosphere. We already saw that this cluster has the highest proportion of the blog volume. Again, we note that bloggers mention the phrases in this cluster right at the peak popularity and later the rest of the media follows.

5. EXPERIMENTS ON TWITTER

We also analyze the temporal patterns of attention of content published on Twitter. In order to identify and trace content that appears on Twitter we focus on appearance of URLs and "hashtags". Users on Twitter often make references to interesting content by including the URL in post. Similarly, many tweets are accompanied by hashtags (e.g., ♯*ilovelifebecause*), short textual tags that get widely adopted by the Twitter community. Links and hashtags, adopted by the Twitter users, represent specific pieces of information that we can track as they get adopted across the network. Similarly as with the quoted phrases, our goal here is applying ***K-SC*** in order to identify patterns in the temporal variation of the popularity of a hashtags and URLs mentioned in tweets and to explain the patterns based on individual users' participation.

Data Preparation. We collected nearly 580 million Twitter posts from 20 million users covering a 8 month period from June 2009 to February 2010. We estimate this is about 20-30% of all posts published on Twitter during that time frame. We identified 6 million different hashtags and 144 million URLs mentioned in these posts. For each kind of items of content (i.e., separately for URLs and the hashtags) we discard items which exhibit nearly uniform volume over time. Then we order the items by their total volume and focus on 1,000 most frequently mentioned hashtags (URLs) and 100,000 users that mentioned these items most frequently.

Analysis of the results. We present the results of identifying the temporal patterns of Twitter hashtags. We note that we obtain very similar results if using URLs. For each hashtag, we build a time series describing its volume following exactly the same protocol as with quoted phrases. We use 1 hour time unit and truncate the series to 128 hours around the peak volume with the peak at occurring at 1/3 of 128 hours. We run ***K-SC*** on these time series and present the shapes of identified cluster centroids in the Figure 8.

Whereas mass media and blogs mention phrases that are related to certain pieces of news or events, most Twitter users adopt hashtags entirely by personal motivation to describe their mood or current activity. This difference appears in Figure 8 in that most hashtags maintain nonzero volume over the whole time period. This means that there always exist a certain number of users who mention a hashtag even if it is outdated or old. Nevertheless, the patterns of temporal variation in the hashtag popularity are very consistent with the clusters of temporal variation of quoted phrases identified in Figure 7. We can establish a perfect correspondence between the classes of temporal variation of these two very different types of online content, namely quoted phrases and Twitter hashtags (and URLs). We arrange the clusters in Figure 8 in the same order as in Figure 7 so that $T1$ corresponds to $C1$, $T2$ to $C2$, and so on. These results are very interesting especially considering that the motivation for people in Twitter to mention hashtags appears to be different from mechanisms that drive the adoption of quoted phrases. Although we omit the discussion of the temporal variation of URL mentions due to brevity, we note that the obtained clusters are nearly identical to the hashtag clusters (see [1]).

Table 4 gives further statistics of Twitter hashtag clusters. Comparing these statistics to characteristics of phrase clusters (Table 2) we observe several interesting differences. The largest Twitter cluster ($T2$) has more phrases than the largest phrase cluster ($C1$), while the smallest Twitter cluster has more members than smallest phrase cluster. This shows that sizes of Twitter clusters are somewhat more skewed. Moreover, we also note that Twitter clusters are less concentrated around the peak volume, with the peak volume accounting for only around 2-5% of the total volume (in phrase clusters peak accounts for 4-16% of the total volume).

Cluster	T1	T2	T3	T4	T5	T6
f_c	16.1%	35.1%	15.9%	10.9%	13.7%	8.3%
V	4083	3321	3151	3253	3972	3177
V_{128}	760	604	481	718	738	520
V_P	86	169	67	60	67	53
V_P/V	2.1%	5.1%	2.1%	1.8%	1.7%	1.7%

Table 4: Twitter hashtag cluster statistics. Table 2 gives the description of the symbols.

Number of features	50	100	200	300
Temporal features	69.53%	78.30%	88.23%	95.35%
Volume features	66.31%	71.84%	81.39%	92.36%
TF-IDF features	64.17%	70.12%	79.54%	89.93%

Table 5: Classification accuracy of the clusters in Twitter with a different set of features. Refer to the main text for description.

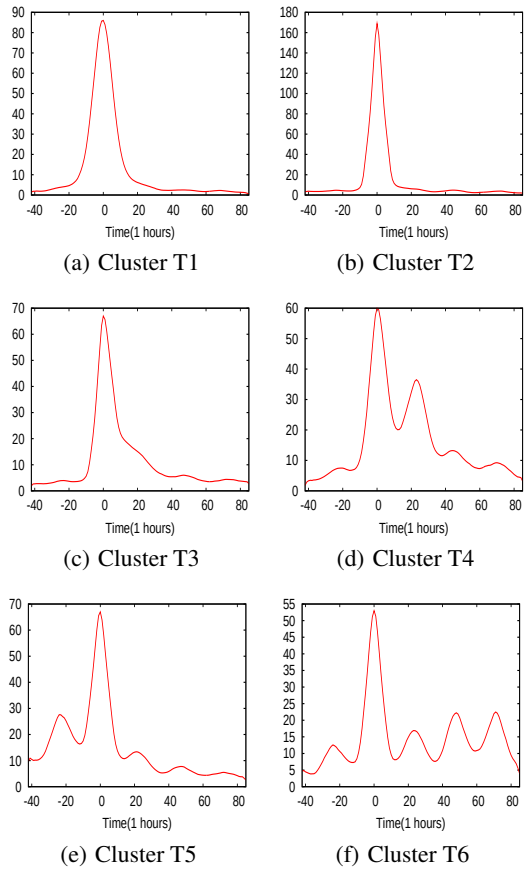
Next we also perform the predictive task of predicting the shape of volume over time curve for Twitter hashtags. Twitter data is very sparse as even the most active most users mention only about 10 to 50 different hashtags. Thus we order users by the total number of hashtags they mention, collect them into groups of 100 users, and measure the collective behavior of each group of users.

For each hashtag, we build a feature vector where i-th component stores the time of the earliest mention of the tag by any user in the group i. Similarly as with quoted phrases we construct a feature vector based on the fraction of the mentions from each group, and another feature vector based on the TF-IDF score treating hashtags as "documents" and user groups as "words". For each cluster, we perform a binary classification for a cluster against the rest using the logistic regression, and report the average accuracy over the six classification tasks in the Table 5. Again, the temporal features achieve best accuracy, suggesting that the time when a user group adopts a hashtag is an important factor in determining how the popularity of the hashtag will vary over time. We also note that the accuracies are lower than for quoted phrases (Table 3) and the gap gets larger as we choose a smaller number of features. This gap suggests that a small number of large famous media sites and blogs has a much greater influence on the adoption of news media content than the most active groups of users have on the adoption of Twitter hashtags. Even though the large scale temporal dynamics of attention of Twitter and news media content seems similar. These results hint that the adoption of quoted phrases tends to be much quicker and driven by a small number of large influential sites. On the other hand, in Twitter it appears as if the influentials are much less influential and have smaller cumulative impact on the content popularity.

6. RELATED WORK

There are two distinct lines of work related to the topics presented here: work on temporal dynamics of human activity, and research on the general time series clustering.

Temporal dynamics of human activity. Patterns of human attention [34, 35], popularity [24, 30] and response dynamics [6, 10] have been extensively studied. Research investigated temporal patterns of activity of news articles [5, 30], blogposts [3, 14, 21, 27], Videos [10] and online discussion forums [4]. Our work here is different as we are not trying to find a unifying global model of temporal variation but rather explore techniques that allow us to quantify what kinds of temporal variations exist on the Web. In this light, our work aligns with the researches on Web search queries



(a) Cluster T1 (b) Cluster T2

(c) Cluster T3 (d) Cluster T4

(e) Cluster T5 (f) Cluster T6

Figure 8: Shapes of attention of Twitter hashtags.

that find temporal correlation between social media [2] or queries whose temporal variations are similar each other [8]. After temporal patterns are identified, one can then focus on optimizing media content placement to maximize clickthrough rates [5], predicting the popularity of news [30] or finding topic intensities streams [19].

Time series clustering. Two key components of time series clustering are a distance measure [11], and a clustering algorithm [32]. While the Euclidean distance is a classical time series distance metric, more sophisticated measures such as the Dynamic Time Warping and the Longest Common Subsequence [18] have also been proposed. Among clustering algorithms, the agglomerative hierarchical [20] and the K-means clustering [28] are frequently used. Due to its simplicity and scalability, K-means inspired many variants such as k-medoids[17], fuzzy K-means [17], and the Expectation Maximization based variant [28]. To address the issues caused by the high dimensionality of time series data, transforms such as Discrete Fourier Transform, Discrete Haar Wavelet Transform [7], Principal Component Analysis and Symbolic Aggregate Approximation [25] have also been applied.

7. CONCLUSION

We explored temporal patterns arising in the popularity of online content. First we formulated a time series clustering problem and motivated a measure of time series similarity. We then developed *K-SC*, a novel algorithm for time series clustering that efficiently computes the cluster centroids under our distance metric. Finally, we improved the scalability of *K-SC* by using a wavelet-based incremental approach.

We investigated the dynamics of attention in two domains. A

massive dataset of 170 million news documents and a set of 580 million Twitter posts. The proposed *K-SC* achieves better clustering than K-means in terms of intra-cluster homogeneity and inter-cluster diversity. We also found that there are six different shapes that popularity of online content exhibits. Interestingly, the shapes are consistent across the two very different domains of study, namely, the short textual phrases arising in news media and the hashtags on Twitter. We showed how different participants in online media space shape the dynamics of attention the content receives. And perhaps surprisingly based on observing a small number of adopters of online content we can reliably predict the overall dynamics of content popularity over time.

All in all, our work provides means to study common temporal patterns in popularity and the attention of online content, by identifying the patterns from massive amounts of real world data. Our results have direct application to the optimal placement of online content [5]. Another application of our work is the discovery of the roles of websites which can improve the identification of influential websites or Twitter users [23]. We believe that our approach offers a useful starting point for understanding the dynamics in the online media and how the dynamics of attention evolves over time.

Acknowledgment

We thank Spinn3r for resources that facilitated the research, and reviewers for helpful suggestions. Jaewon Yang is supported by Samsung Scholarship. The research was supported in part by NSF grants CNS-1010921, IIS-1016909, Albert Yu & Mary Bechmann Foundation, IBM, Lightspeed, Microsoft and Yahoo.

8. REFERENCES

[1] Extended version of the paper. Patterns of temporal variation in online media. Technical Report, Stanford Infolab, 2010.

[2] E. Adar, D. Weld, B. Bershad, and S. Gribble Why We Search: Visualizing and Predicting User Behavior. In *WWW '07*, 2007.

[3] E. Adar, L. Zhang, L. A. Adamic, and R. M. Lukose. Implicit structure and the dynamics of blogspace. In *Workshop on the Weblogging Ecosystem*, 2004.

[4] C. Aperjis, B. A. Huberman, and F. Wu. Harvesting collective intelligence: Temporal behavior in yahoo answers. *ArXiv e-prints*, Jan 2010.

[5] L. Backstrom, J. Kleinberg, and R. Kumar. Optimizing web traffic via the media scheduling problem. In *KDD '09*, 2009.

[6] A.-L. Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005.

[7] F. K.-P. Chan, A. W. chee Fu, and C. Yu. Haar wavelets for efficient similarity search of time-series: With and without time warping. *IEEE TKDE*, 15(3):686–705, 2003.

[8] S. Chien and N. Immorlica. Semantic Similarity between Search Engine Queries Using Temporal Correlation. In *WWW '05*, 2005.

[9] K. K. W. Chu and M. H. Wong. Fast time-series searching with scaling and shifting. In *PODS '99*, 237–248, 1999.

[10] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *PNAS*, 105(41):15649–15653, October 2008.

[11] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *VLDB.*, 1(2):1542–1552, 2008.

[12] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, 1996.

[13] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *KDD '05*, 2005.

[14] D. Gruhl, D. Liben-Nowell, R. V. Guha, and A. Tomkins. Information diffusion through blogspace. In *WWW*, 2004.

[15] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD workshop*, pages 56–65. 2007.

[16] E. Katz and P. Lazarsfeld. *Personal influence: The part played by people in the flow of mass communications*. Free Press, 1955.

[17] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, March 2005.

[18] E. Keogh and C. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[19] A. Krause, J. Leskovec, and C. Guestrin. Data association for topic intensity tracking. In *ICML '06*, 2006.

[20] M. Kumar, N. R. Patel, and J. Woo. Clustering seasonality patterns in the presence of errors. In *KDD '02*, 2002.

[21] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW '02*, 2003.

[22] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09*, 2009.

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

[24] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM '07*, 2007.

[25] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *SIGMOD '03*, 2003.

[26] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. A. N. Amaral. A poissonian explanation for heavy tails in e-mail communication. *PNAS*, 105(47):18153–18158, 2008.

[27] Q. Mei, C. Liu, H. Su, and C. Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW '06*, 2006.

[28] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos. Iterative incremental clustering of time series. In *EDBT*, 2004.

[29] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1986.

[30] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *ArXiv e-prints*, Nov 2008.

[31] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD '07*, page 793, 2007.

[32] T. Warren Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[33] D. J. Watts and P. S. Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34(4):441–458, December 2007.

[34] F. Wu and B. A. Huberman. Novelty and collective attention. *PNAS*, 104(45):17599–17601, 2007.

[35] S. Yardi, S. A. Golder, and M. J. Brzozowski. Blogging at work and the corporate attention economy. In *CHI '09*, 2009.

APPENDIX

A. APPENDIX

A.1 Comparison to the clusters by k-means

Here, we compare the clusters we find using *K-SC* to the clusters by K-means. To begin with, we normalize the peak of all time series to the same value because, in our experiment, this normalization method lead to more diverse looking clusters than other normalization with K-means such as normalizing sum of time series to the same, and normalizing the standard deviation of time series to the same. Then, we apply K-means to the Memetracker data set using the same settings as we do in Section 3, and plot the clusters in Figure 9. The clusters do not have as diverse shape as *K-SC* clusters do in Figure 7, and they are not well interpretable as all media types appear at about the same time in most clusters.

A.2 The consistency of clusters in Memetracker

We present that our observations in Memetracker are quite robust and thus the observed phenomena seem genuine. Figure 10 shows the clusters that we perform *K-SC* over the Memetracker data set with $k = 12$. Notice the cluster shapes are practically same as with 6 clusters in Figure 7.

A.3 Proof of the symmetry of our similarity measure

We prove that the similarity measure proposed in Section 2 is symmetric. Let us define $g(x, y)$ as follows:

$$g(x,y) = \min_{\alpha} \frac{||x - \alpha y||}{||x||}$$

. Then, $\hat{d}(x,y) = \min_{\alpha,k} g(x, y_k)$. If we can show that g^2 is symmetric, then $\hat{d}$ is also symmetric:

$$\hat{d^2}(x,y) = \min_k g^2(x, y_k) = \min_k g^2(x_{-k}, y) = \min_k g^2(y, x_{-k}) = \hat{d^2}(y,x)$$

So, we will show that g^2 is symmetric by the following algebra. We will use $\alpha = \frac{x^T y}{||y||^2}$ that we showed in Section 2.

$$\begin{aligned}
g(x,y)^2 &= \min_{\alpha} \frac{||x - \alpha y||^2}{||x||^2} \\
&= \frac{1}{||x||^2} ||x - \frac{x^T y}{||y||^2} y||^2 \\
&= \frac{1}{||x||^2} ||x - \frac{yy^T}{||y||^2} x||^2 \\
&= \frac{1}{||x||^2} ||(I - \frac{yy^T}{||y||^2})x||^2 \\
&= \frac{1}{||x||^2} x^T (I - \frac{yy^T}{||y||^2})^2 x \\
&= \frac{1}{||x||^2} x^T (I + \frac{yy^T yy^T}{||y||^4} - 2\frac{yy^T}{||y||^2})x \\
&= \frac{1}{||x||^2} x^T (I - \frac{yy^T}{||y||^2})x \\
&= \frac{1}{||x||^2} (x^T x - \frac{x^T yy^T x}{||y||^2}) \\
&= \frac{1}{||x||^2} (x^T x - \frac{(x^T y)^2}{||y||^2}) \\
&= 1 - \frac{(x^T y)^2}{||x||^2 ||y||^2}
\end{aligned}$$

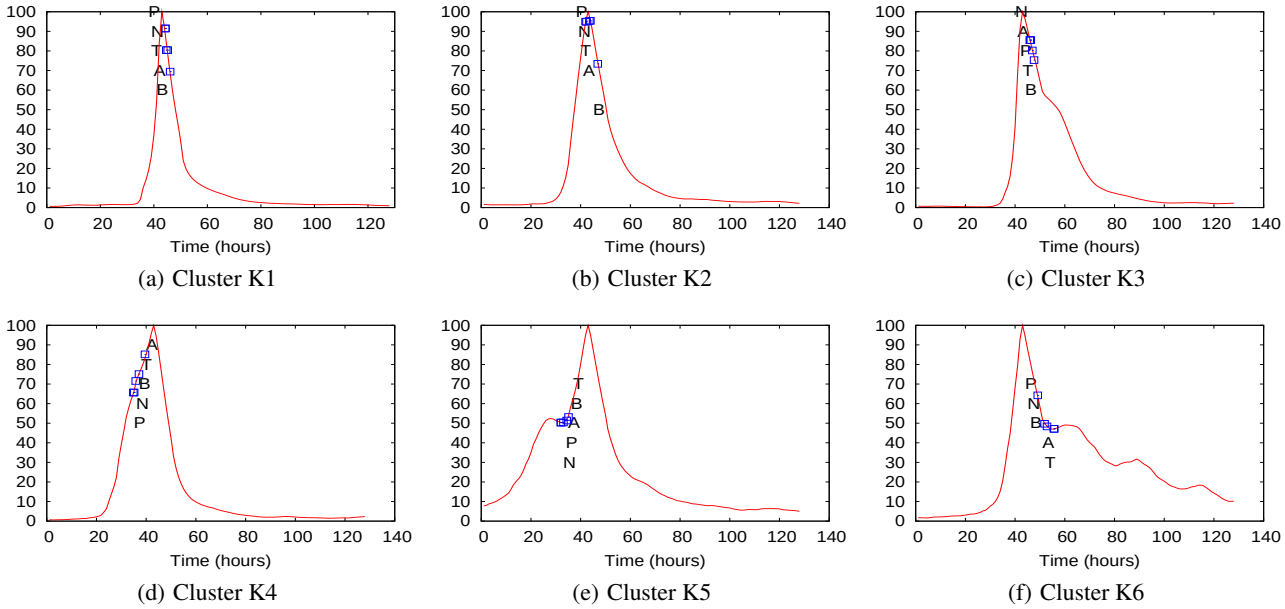
The last form is symmetric in x and y, and thus g^2 is symmetric.

Type	Website
Newspaper	startribune.com washingtonpost.com boston.com
Professional blog	huffingtonpost.com salon.com
TV	cnn.com abcnews.go.com cbsnews.com
News **A**gency	reuters.com hosted.ap.org
Blog (aggregator)	freerepublic.com wikio.com

Table 6: Five groups of websites.

A.4 Five groups of websites

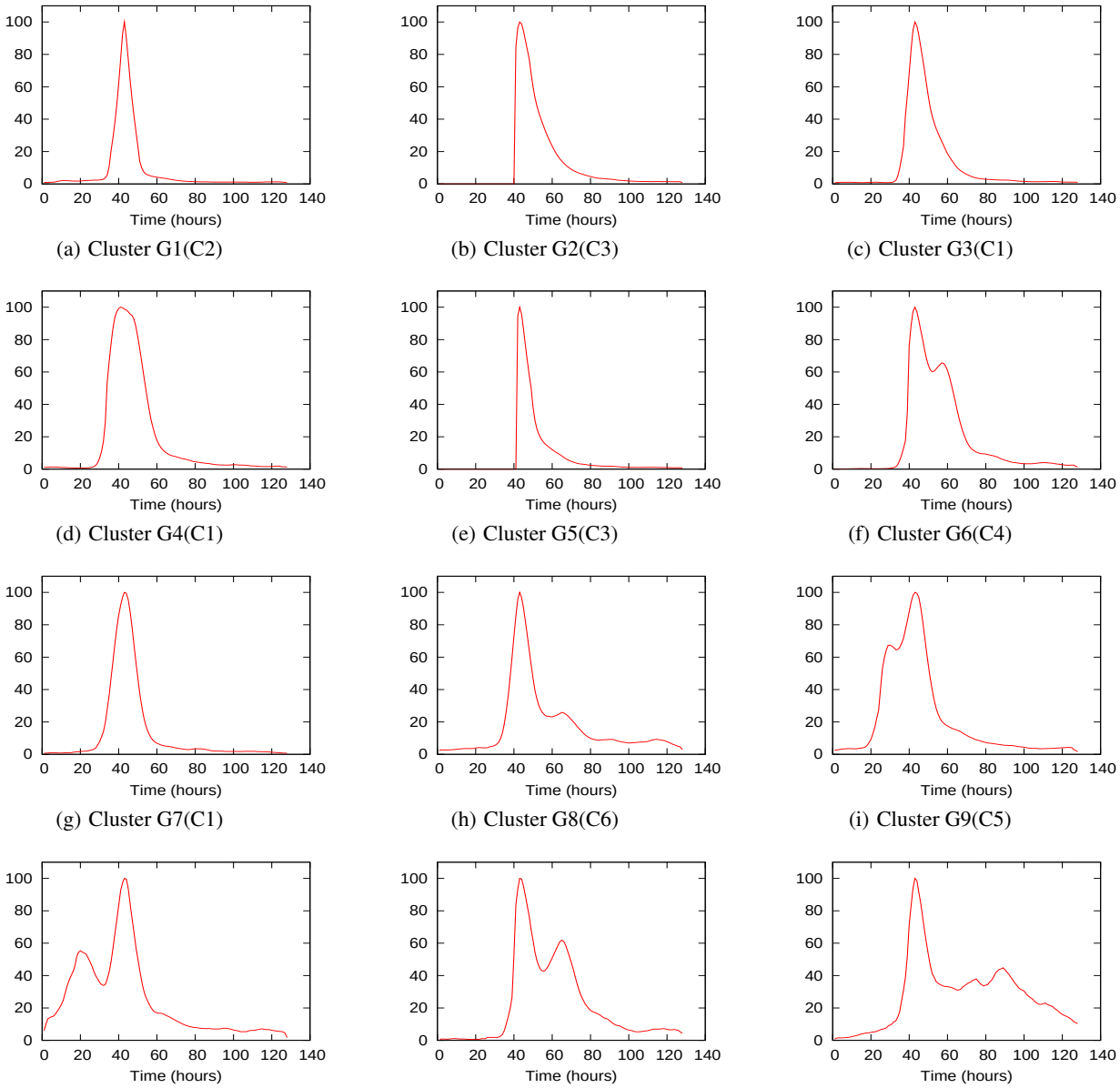
In Table 6 we show the five groups of the websites that used for modelling the time series shape in Section 4.



(a) Cluster K1 (b) Cluster K2 (c) Cluster K3

(d) Cluster K4 (e) Cluster K5 (f) Cluster K6

Figure 9: K-means clusters. K1,K2,and K4 do not show significant difference. Also, K-means fails to find C3 in the figure 7. C3 is unique in sudden emergence led by News Agencies ("A" in the figure). No cluster is led by agencies.



(a) Cluster G1(C2) (b) Cluster G2(C3) (c) Cluster G3(C1)

(d) Cluster G4(C1) (e) Cluster G5(C3) (f) Cluster G6(C4)

(g) Cluster G7(C1) (h) Cluster G8(C6) (i) Cluster G9(C5)

(j) Cluster G10(C5) (k) Cluster G11(C4) (l) Cluster G12(C6)

Figure 10: 12 clusters by KSC. Each of them can be represented as one of clusters in the figure 7. In parentheses, we denote which cluster in the figure 7 can represent each cluster.