Lifecycle Modeling for Buzz Temporal Pattern Discovery

YI CHANG, Yahoo Research
MAKOTO YAMADA, Kyoto University
ANTONIO ORTEGA, University of Southern California
YAN LIU, University of Southern California

In social media analysis, one critical task is detecting a burst of topics or *buzz*, which is reflected by extremely frequent mentions of certain keywords in a short time interval. Detecting buzz not only provides useful insights into the information propagation mechanism, but also plays an essential role in preventing malicious rumors. However, buzz modeling is a challenging task because a buzz time-series often exhibits sudden spikes and heavy tails, where most existing time-series models fail. In this paper, we propose novel buzz modeling approaches which capture the rise and fade temporal patterns via *Product Lifecycle (PLC)* model, a classical concept in economics. More specifically, we propose to model multiple peaks in buzz time-series with PLC mixture or PLC group mixture, and develop a probabilistic graphical model (*K-MPLC*) to automatically discover inherent lifecycle patterns within a collection of buzzes. Furthermore, we effectively utilize the model parameters of PLC mixture or PLC group mixture for burst prediction. Our experimental results show that our proposed methods significantly outperform existing leading approaches on buzz clustering and buzz type prediction.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Clustering

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Time-Series Modeling, Buzz Clustering, Buzz Type Prediction

ACM Reference Format:
Yi Chang, Makoto Yamada, Antonio Ortega, Yan Liu, 2016. Lifecycle Modeling for Buzz Temporal Pattern Discovery. *ACM Trans. KDD* 11, 2, Article 20 (November 2016), 25 pages.
DOI: http://dx.doi.org/10.1145/2994605

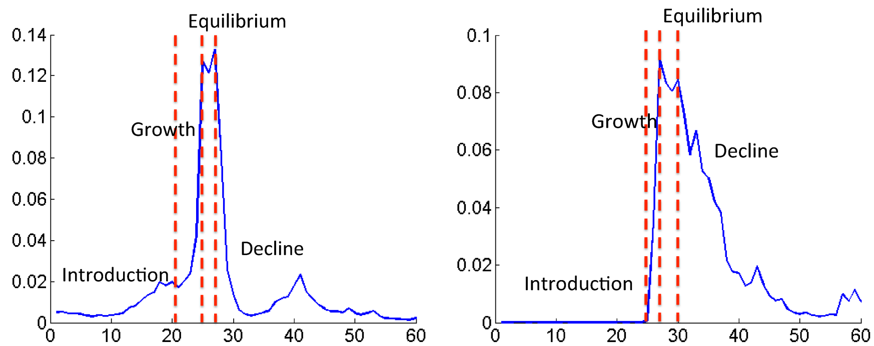
1. INTRODUCTION

Social media is growing at an explosive rate, and hundreds of millions of users generate vast amounts of contents on various social media web sites, such as Twitter and Tumblr. One critical problem in social media analysis is to effectively model temporal human bevavioral patterns over time from data [Cui et al. 2016]. More specifically, modeling buzz events, which are reflected by frequent mentions of certain key words in a short time interval, such as *new iPhone* or *Hurricane Sandy*, is in particular important problem for captureing temporal human behavioral patterns. Since the buzz events are closely related to human activities, effectively modeling buzzes could help us track mass attention, obtain public opinions, or forecast users' reactions to a particular event in the near future.

Buzz modeling is an extremely challenging problem since many buzz events are unforeseeable (e.g., earthquakes or hurricanes), while others may involve human inter-

An earlier version of this paper was published at ICDM 2014 as a short paper with the title, *Ups and Downs in Buzzes: Life Cycle Modeling for Temporal Pattern Discovery*.
Author's addresses: Y. Chang, Yahoo Research, Sunnyvale, California 94089; M. Yamada, Kyoto University, Kyoto, 611-0011, Japan; A. Ortega and Y. Liu, University of Southern California, Los Angeles, CA 90089.

DOI: http://dx.doi.org/10.1145/2994605



(a) Time-series of *Euro 2012*. (b) Time-series of *Aurora Colorado*.

Fig. 1: Buzz time-series examples.

ventions. As a result, a buzz time-series, which consists of the number of mentions for each time unit, often exhibits sudden spikes and heavy tails. Thus, it is difficult to represent a buzz time-series using traditional time-series models [Blischke and Murthy 2000]. Existing buzz modeling methods employ simple time-series models (e.g., univariate models), and explore additional feature cues (such as sentiment, spreading measures and controversialist), to predict whether a certain topic can become viral [Choi and Varian 2009; Guerini et al. 2011; Lerman and Ghosh 2010]. These methods achieve success to a certain extent, but few of them explicitly capture the inherent temporal patterns of buzzes. Effective buzz models, which not only can capture those sudden spikes and heavy tails patterns but also can differentiate true spikes from jagged noise in a time-series, are in great need.

One important challenge in pursuit of effective buzz models is how to find distinct temporal patterns within a set of buzz time-series. For example, as seen in Figure 1(a), the *Euro 2012* event attracted more and more frequent mentions when it came close to its final match day, a sudden increase of frequency happened when the final match started, and the large volume was maintained for a couple of hours during the final match, after which the mention volume dropped dramatically in just a few hours on social media, forming a sharp tail. In contrast, for the sudden breaking news of a gun shooting at *Aurora Colorado* in 2012, as seen in Figure 1(b), we observe that a huge amount of frequent mentions appeared all of a sudden, and the public's attention dropped slowly in the next 24 hours, forming a heavy tail. [1] On both sub figures, X-axis indicates the time axis in hours, and Y-axis means the density for the number of mentions on social media. Intuitively, these two events have quite distinct temporal patterns. Therefore capturing the temporal patterns and modeling the inherent structures of these temporal patterns can provide us useful insights for information propagation [Sun and Tang 2011] and time-series based prediction [Johan Bollen 2011; Conover et al. 2011; O'Connor et al. 2010; Takahashi et al. 2013] in social media.

In this paper, we propose to model rise-and-fade temporal shapes using *product lifecycle* (PLC) models [Levitt 1965]. PLC was originally introduced by economists to model the life span of a product: introduction to the market (initial sales), growth (sud-

[1]Note that on popular social media platforms, user's interest switches quickly, and there are much more users in day time than after middle night. Therefore, a asymmetrical rise-and-fade temporal shape with a sudden rise while spanning many more hours is considered following a sudden-spike-and-heavy-tail pattern.

den increase of sales), equilibrium (maturity phase defined by approximately constant sales) and decline (when the sales decrease dramatically). In Figure 1, we can observe 4 different stages, which are segmented with dotted lines. In general, for different types of buzz events, the growth and equilibrium stages might be relatively short, while the decline stage varies. Moreover, a buzz sequence often has multiple peaks, and the number of peaks is unknown in advance. To handle a time-series with multiple peaks, we propose two novel approaches based on PLC models: (i) we model each buzz time-series with a mixture of PLC models, then propose an efficient $L1$-regularized lasso [Tibshirani 1996] based approach to achieve a sparse solution for the PLC mixture model; (ii) we also model each rise-and-fade segment in time-series as a group of PLC models, and model each buzz time-series with multiple peaks as a mixture of PLC group models, then propose an efficient group lasso [Yuan and Lin 2006] based approach to achieve its sparse solution. After that, to discover the underlying patterns in a collection of buzz time-series, we develop a probabilistic graphical model, K-Mixture of Product Lifecycle (*K-MPLC*), to automatically group buzz time-series based on their PLC mixture parameters or PLC group mixture parameters. Finally, we propose a buzz type prediction approach, which uses the parameters of PLC mixture or PLC group mixture as its the feature vector. The experimental results show that our proposed approaches outperform existing leading approaches on both buzz clustering task and buzz type prediction task.

The major contributions of this paper include:

— Leveraging product lifecycle to model the temporal patterns of buzz events.
— Proposing PLC mixture model and PLC group mixture model to systematically model the possible multiple peaks in a buzz time-series and achieve sparse solutions for both models.
— Proposing a probabilistic graphical model to discover the inherent clusters within a collection of buzzes.

Preliminary version of this work appeared in [Chang et al. 2014]; here we propose the PLC Group Lasso in addition to the PLC Lasso and include a more detailed experimental section that addresses the effectiveness of the proposed method in buzz clustering. Moreover, we include the buzz type prediction tasks in Section 7.

In the rest of the paper, after introducing related work in section 2, we first formulate the problems in section 3, then propose PLC mixture model and PLC group mixture model to model each time-series in Section 4; we propose a probabilistic graphical model to cluster different time-series in Section 5; Section 6 and 7 cover buzz clustering experimental results and buzz type prediction experimental results respectively; we conclude our paper in the last section.

2. RELATED WORK

Our work is related to *time-series modeling*, *time-series clustering*, and *buzz prediction*. Below, we review some important related work in each subarea.

Time-series modeling: To model the temporal patterns of online content, Matsubara et al. [Matsubara et al. 2012] propose an algorithm to model the rise and fade patterns of influence propagation. Hong et al. [Hong et al. 2011] attempt to model the time decay of topics on Twitter with Gamma function. Furthermore, the temporal information of online forums [Gruhl et al. 2005], blogspace [Kumar et al. 2003], and online groups [Kumar et al. 2010] has been explored and mined under different scenarios, and spatiotemporal patterns of subtopics have been investigated by [Mei et al. 2006].

Another line of research would be to use survival analysis [Yu et al. 2015; Zhang et al. 2016]. The survival analysis is in particular useful for modeling individual human activity pattern which follows burstly dynamics with heavy tailed distribution. However, most of the existing methods can only deal with single spike signal. Thus, existing time-series modeling methods are not suited for buzz events with multiple spikes.

An approach for dealing with multiple spikes is to use Hawkes process [Hawkes and Oakes 1974]. Crane and Sornette propose to use a power law distribution as a function for buzz time-series [Crane and Sornette 2008]. However, this needs to set spike locations manually in Hawkes process, which is not practical in real applications.

Another useful approach is to model occurance of spikes using infinite-state automation approach [Kleinberg 2003]. However, the method uses spike locations as input, and thus it is not possible to directly apply the infinite-state automation approach for raw time-series data.

There are few previous studies to model time-series of topics with product lifecycle models, and the product lifecycle models are mainly investigated by economists [Blischke and Murthy 2000; Isaic-Maniu and Voda 2008; PETRESCU 2009]. There is some data mining work related to product lifecycle: classical PLC models are leveraged to cluster seasonality patterns for retail industry [Kumar et al. 2002]; Haider et al. [Haider et al. 2012] explore discriminative clustering for market segmentation tasks, which combines market segmentation with different group of users.

Time-series clustering: Time-series clustering has been an active research topic in recent years, and various time-series clustering algorithms are proposed such as batch mode clustering [Rakthanmanon et al. 2011; Li and Prakash 2011], incremental clustering [Lin et al. 2004], and anytime clustering [Zhu et al. 2012]. One of the important component of time-series clustering is to choose a distance metric, and Dynamic Time Warping (DTW) and Complexity-Invariant Distance (CID) are the most widely used metrics [Zhu et al. 2012; Batista et al. 2011]. Clustering algorithms have been applied to various types of time-series data, including snippets within long time-series [Rakthanmanon et al. 2011], different time-series streams [Li and Prakash 2011], and multiple time-series matrix [Jiang et al. 2012]. However, for extremely noisy buzz time-series, DTW scores tend to be inaccurate, and thus clustering performance can be degraded.

The most similar work to ours is that of Yang and Leskovec [Yang and Leskovec 2011], in which the K-spectral centroid (K-SC) algorithm is proposed to group similar time-series via aligning peaks, shifting over time-axis, and scaling over frequency-axis. The K-SC algorithm can cluster time-series that can be aligned by shifting and scaling operations, but fails for sharp or heavy-tailed temporal sequences. Moreover, if two time-series share similar peak shapes but differ in the intervals between the peaks, K-SC cannot identify these two time-series as the same cluster through simple shifting or scaling.

Buzz prediction: Buzz prediction from social media data is a challenging problem. This problem is studied for predicting stock market changes [Johan Bollen 2011], data mining for political opinions [O'Connor et al. 2010; Conover et al. 2011], and market-decision making [Takahashi et al. 2013]. In [Matsubara et al. 2012], prediction of the future burst using SpikeM was proposed. In this approach, they predict the future rise and fade pattern of burst signal from a partially observed buzz time-series. Overall, existing methods mainly focus on predicting rise and fade pattern of a time-series. In contrast, we will focus on burst prediction tasks, which aim to classifying the shape and the number of buzz (burst type) in a time-series. To the best of our knowledge, there is few burst prediction research in social media and this is the first paper to address the burst type prediction tasks.

3. PROBLEM STATEMENT

We denote a time-series $\boldsymbol{y} = [y_1, \ldots, y_T]^\top \in \mathbb{R}^T$, where T refers to the length of time-series, and $^\top$ denotes the matrix transpose. We denote a set of N time-series $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N] \in \mathbb{R}^{T\times N}$. The final goal of this paper is to model buzz time-series, and it can be divided into three sub-problems.

Buzz time-series modeling. The problem is formally stated as: *given a buzz time-series* $\boldsymbol{y}$*, we aim to model it with* L *rise and fade patterns, and each pattern is represented with a set of parameters* Θ_l*.*

Buzz clustering. The problem is formally stated as: *given a set of buzz time-series* $\boldsymbol{Y}$*, we aim to leverage its buzz modeling parameters to cluster* N *time-series into* K *clusters.*

Buzz type prediction. The problem is formally stated as: *for each buzz time-series* $\boldsymbol{y}$*, we aim to leverage its partial information* $\boldsymbol{y} = [y_1, \ldots, y_T'], T' < T$ *and corresponding buzz modeling parameters to predict* $\boldsymbol{y}$ *belongs to which of the* K *clusters.*

The buzz time-series modeling is related to feature extraction tasks from raw signal. Thus, once we extract useful features from buzz time-series, we can use the extracted features for the buzz type clustering and prediction tasks. In this paper, we focus on the two applications (clustering and buzz type prediction), but we can use the proposed buzz modeling technique for other applications in future.

4. MODELING BUZZES WITH A MIXTURE OF PRODUCT LIFECYCLE MODELS

In this section, we first introduce the concept of product lifecycle (PLC) model; next, we propose to fit each buzz time-series with a mixture of PLCs, and estimate the PLC mixture model using a lasso based method; then we propose to fit each time-series with a mixture of PLC groups, and estimate the PLC group mixture model using a group lasso based method.

4.1. Product Lifecycle Models

The concept of *Product Lifecycle* (PLC) was originally proposed in economics in the 1960s [Levitt 1965]. The classical view of PLC assumes 4 phases to cover the life span of a product: Introduction, Growth, Equilibrium and Decline. Introduction refers to low growth rate of sales as the product is newly launched in the market; Growth implies that the public gains awareness of the product and consumers come to understand its benefits and accept it, so that a company can expect a period of rapid sales growth; Equilibrium corresponds to the product reaching maturity, so that the sales growth slows and sales volume eventually peaks and stabilizes; Decline indicates that the product enters into decline, as sales and profits start to fall because the market has become saturated, the product has become obsolete, or customer tastes have changed. As an important research topic, many different versions of PLC models are summarized in [Blischke and Murthy 2000].

To model the long decay trend of the decline stage of a product, Isaic-Maniu and Voda [Isaic-Maniu and Voda 2008] proposed to use the Gamma distribution as a PLC model. The probability density function of Gamma distribution is:

$$f(t;\alpha,\beta) \quad = \frac{\beta^{\alpha}}{\Gamma(\alpha)} t^{\alpha-1} e^{-\beta t}, \tag{1}$$

where $t \geq 0$, $\alpha > 0$ is the shape parameter, $\beta > 0$ is the rate parameter, and the Gamma function $\Gamma(\alpha)$ is defined as:

$$\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} \mathrm{d}t.$$

As compared to other distribution functions, such as the Gaussian distribution, the Gamma distribution can model a sudden rise and long decay trend in the decline stage of a PLC. Note that our method is not limited to the Gamma distribution, and it can be extended with other base distribution functions, such as Alpha distribution [Isaic-Maniu and Voda 2008], Weibull distribution [Bauckhage et al. 2013], which were leveraged to model temporal dynamics. Moreover, we propose a flexible buzz fitting approach in Section 4.3, where the model does need to specify a specific distribution for a product lifecycle.

4.2. PLC Lasso

Given a buzz topic, we count its mentions on a given social media channel during a pre-determined time interval (e.g., an hour), and generate a time-series of this topic over a number of intervals during an observation window. Since a buzz sequence may consist of several obvious peaks, it could be modeled with multiple PLC models. As the number of peaks is not known in advance, we model a buzz time-series with a mixture of PLCs as:

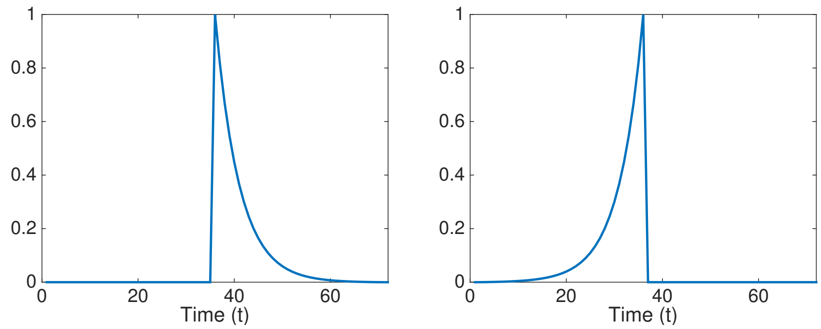
$$f(t;\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu}) = \sum_{\ell=1}^{L} w_\ell k(t;\alpha_\ell,\beta_\ell,\mu_\ell), \tag{2}$$

$$k(t;\alpha,\beta,\mu) = \begin{cases} Z^{-1}(t-\mu)^{\alpha-1}e^{-\beta(t-\mu)} & (t \geq \mu) \\ 0 & (\text{Others}) \end{cases},$$

where L is the number of PLC models, $\boldsymbol{w} = [w_1,\ldots,w_L]^\top$ denotes the weight vector, $^\top$ denotes the matrix transpose, $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_L]^\top$ and $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_L]^\top$ are the vectors of Gamma distribution parameters, $\boldsymbol{\mu} = [\mu_1,\ldots,\mu_L]^\top$ refers to locations of PLC models, and Z is the normalization factor.

Another similar approach for dealing with multiple spikes time-series is to utilize Hawkes process [Hawkes and Oakes 1974; Crane and Sornette 2008]. The difference with respect to the original Hawkes process is that the spike locations need to be set manually in the original formulation (i.e., μ_ℓ is set by human), while our model considers spike locations as model parameters to be identified. For a buzz sequence, since it is usually hard to know spike locations, our formulation is better suited than the existing Hawkes process based modeling. Nevertheless, since spike locations are unknown, the mixture PLC model estimation is more challenging than the estimation of Hawkes process parameters.

Let us denote buzz time-series as $\boldsymbol{y} = [y_1,\ldots,y_T]^\top$, where T refers to the length of time-series. Note, in this paper, we assume that time-series are normalized (i.e., $\sum_{t=1}^{T} y_t = 1$), so that it can be modeled by probability density functions. Then, the optimization problem of fitting the buzz time-series using a mixture of PLC models



(a) (b)

Fig. 2: (a) The Basis function with $\alpha = 1.0$, $\beta = 0.2$, and $\mu = 36$. (b) The Flipped Basis function with $\alpha = 1.0$, $\beta = 0.2$, and $\mu = 36$.

can be formulated as:

$$\min_{\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu}} \quad \sum_{t=1}^{T}(y_t - f(t;\boldsymbol{w},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\mu}))^2$$
$$\text{s.t.} \quad \alpha_\ell > 0, \beta_\ell > 0, w_\ell > 0, \mu_\ell > 0, \ell = 1, 2, \ldots, L.$$

We can solve this optimization problem by using gradient descent. However, since this optimization problem is *non-convex*, it tends to lead to poor locally optimal solutions.

To deal with the *non-convexity*, we first relax the problem to a *convex* optimization problem and then propose a *lasso* [Tibshirani 1996] based approach, making it possible to obtain a global solution. The idea of the lasso based approach is to select L PLC models from a large number of PLC candidates. More specifically, we first define $TM_\alpha M_\beta$ basis functions, where M_α and M_β are the number of candidate values for α and β, so that each basis function corresponds to a PLC model with fixed α and β parameters placed at μ. The total number of basis functions is $TM_\alpha M_\beta$, since a PLC model can be located at any of T positions. As we observe that many of buzz time-series on social media follow the sudden-spike-and-heavy-tail patterns, we experimentally set the shape parameter α to be 1 and the rate parameter β to be $[0.1, 0.2, \ldots, 1.0]$. Note that, we may be able to model a rise pattern by setting $\alpha > 1$. However, when setting $\alpha > 1$, the shape of the Gamma distribution becomes too smooth to model a *sudden-spike* patterns. Thus, we experimentally set $\alpha = 1$. In Figure 2(a), we show the basis function with $\alpha = 1.0$ and $\beta = 0.1$ at $\mu = 10$.

Then, we solve the following optimization problem:

$$\min_{\boldsymbol{v}} \quad \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{v}\|^2 + \lambda\|\boldsymbol{v}\|_1 \tag{3}$$
$$\text{s.t.} \quad v_\ell \geq 0, \ell = 1, 2, \ldots, TM_\alpha M_\beta,$$

where $\boldsymbol{v} \in \mathbb{R}^{TM_\alpha M_\beta}$ is the model parameter and $\boldsymbol{K} \in \mathbb{R}^{T \times TM_\alpha M_\beta}$ is the pre-computed Gamma distribution functions:

$$K_{t,\mu\times i\times j} = \begin{cases} Z_{ij}^{-1}(t-\mu)^{\widetilde{\alpha}_i - 1}e^{-\widetilde{\beta}_j(t-\mu)} & (t \geq \mu) \\ 0 & (\text{Others}) \end{cases}.$$

In the above formula, $Z_{ij} = \max(\{(t-1)^{\widetilde{\alpha}_i - 1}e^{-\widetilde{\beta}_j(t-1)}\}_{t=1}^{T})$ is the normalization factor, $\widetilde{\alpha}_i,\ i = 1, \ldots, M_\alpha$ and $\widetilde{\beta}_j,\ j = 1, \ldots, M_\beta$ are pre-defined Gamma parameters, μ is the location index of a PLC model ($t - \mu$ indicates the shift of a PLC along x-axis), and $\|\boldsymbol{v}\|_1$ is the $L1$ regularizer to obtain a sparse solution. Since the optimization problem of Eq.(3) is *convex* with respect to $\boldsymbol{v}$, Eq.(3) can be solved by using a state-of-the-art lasso optimization solver, and in this paper, we employ it with the *dual augmented Lagrangian* (DAL) based approach [Tomioka et al. 2011]. Since we use the $L1$ regularizer, we can select a small number of non-overlapped basis functions.

After the lasso fitting, we select L PLC models by ranking the estimated lasso parameter $\widehat{\boldsymbol{v}}$. Since each weight $\widehat{v}_\ell$ is related to a basis Gamma distribution function, we can obtain L Gamma distribution function parameters and their location indexes without further optimization. It may be possible to choose L PLCs by controlling the regularization parameter λ, however this may require running lasso several times, which tends to be computationally expensive. Therefore, we first obtain J PLC models ($J > L$) using a small regularization parameter, and then select L PLC models from them.

After fixing individual PLC parameters with lasso, we solve the following quadratic programming (QP) problem to obtain the corresponding weight $\boldsymbol{w}$ for each PLC model:

$$\min_{\boldsymbol{w}} \quad \|\boldsymbol{y} - \widetilde{\boldsymbol{K}}\boldsymbol{w}\|^2, \quad \textbf{s.t.} \quad w_\ell \geq 0, \sum_{i=1}^{L} w_i = 1,$$

where $\widetilde{\boldsymbol{K}} \in \mathbb{R}^{T \times L}$ is the selected L basis functions from $\boldsymbol{K}$. We call this entire lasso based fitting framework as *PLC Lasso*.

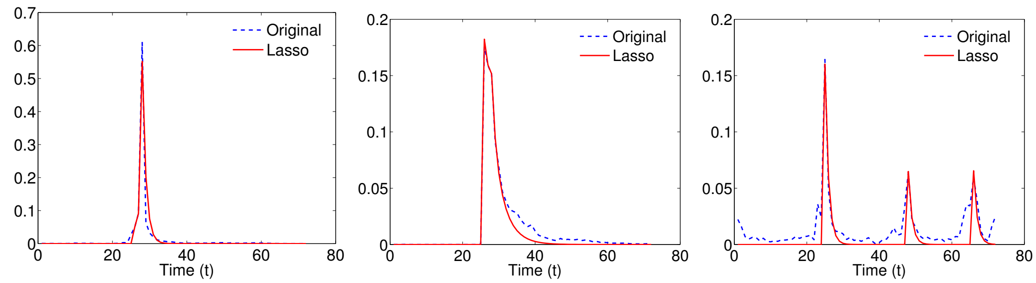
In Figures 3(a1)-(c1), we illustrate several fitted buzz time-series obtained by the PLC Lasso. We also computed root squared mean error (RSME) between original and fitted curves. As can be observed, our proposed method can successfully fit different buzz time-series. Notice in Figure 3(c1) that the fitted curve generated by the lasso based method successfully captures the true peaks of the given time-series while discarding the remaining jagged noise.

4.3. PLC Group Lasso

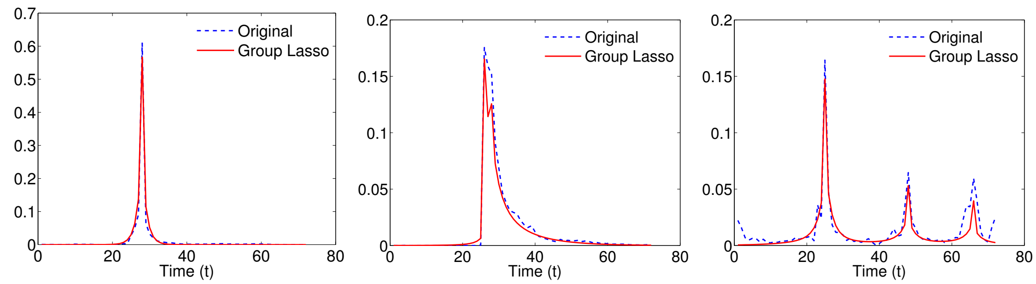
The PLC Lasso implicitly assumes that a buzz consists of a *sharp-rise-and sudden-fade* shape, and it is not suited for modeling *slow-rise patterns*. This is due to using a Gamma distribution for modeling all types of lifecycle patterns. To cope with this issue, we further propose a more general buzz modeling framework called the PLC Group Lasso. More specifically, we model each buzz by a mixture of basis functions as:

$$g(t; \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mu) = \sum_{k=1}^{b} w_k k(t; \alpha_k, \beta_k, \mu),$$

where $k(t, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mu)$ is the basis function, μ is the location of the pattern, $\boldsymbol{w}$ is the weight vector for the pattern, and $\boldsymbol{\gamma}_\ell$ is the parameters for the ℓ-th basis function. Since we represent a lifecycle pattern by a mixture of basis functions, we can represent complex lifecycle patterns.



(a1) RMSE = 0.0188. (b1) RMSE = 0.048. (c1) RMSE = 0.0116



(a2) RMSE = 0.0102. (b2) RMSE = 0.0071. (c2) RMSE = 0.0086.

Method	Figure Index	PLC Weight			PLC Shape	PLC Locations
		w_1	w_2	w_3	τ	σ
PLC Lasso	(a1)	0.0786	0.1141	0.8073	1.0	0.344
	(b1)	0.7395	0.0996	0.1608	3.31	0.4342
	(c1)	0.5505	0.2237	0.2258	1.0	9.8399
PLC Group Lasso	(a2)	0.998	0.001	0.001	1.7493	0.0001
	(b2)	0.7344	0.2646	0.001	5.4341	0.6246
	(c2)	0.6117	0.2126	0.1757	2.8211	9.3327

(d) Feature values of each time-series derived by PLC Lasso and PLC Group Lasso respectively.

Method	All time-series			Buzz time-series		
	$L=2$	$L=3$	$L=4$	$L=2$	$L=3$	$L=4$
PLC Lasso	0.0118	0.0097	0.0082	0.0126	0.0109	0.0099
PLC Group Lasso	0.0116	0.0103	0.0097	0.0107	0.0103	0.0101

(e) The RMSE of the PLC Lasso and the PLC group Lasso.

Fig. 3: Illustrative examples of lasso based fitting. We use the real buzz collected from Twitter timeline. (a1)-(c1): PLC Lasso fitting results. (a2)-(c2): PLC Group Lasso fitting results. (d): Feature Values of Corresponding Methods. (e): The RMSE of the PLC Lasso and the PLC group Lasso for the Twitter timeline data. The "All data" includes in total 534 time-series, while the "Buzz time-series" includes 58 time-series which are systematically extracted from the entire 534 time-series by checking $\max(\boldsymbol{y})-\min(\boldsymbol{y}) > 0.3$.

Then, entire time-series can be modeled as:

$$f(t; \boldsymbol{W}, \boldsymbol{\Gamma}_{\boldsymbol{\alpha}}, \boldsymbol{\Gamma}_{\boldsymbol{\beta}}, \boldsymbol{\mu}) = \sum_{\ell=1}^{L} g(t; \boldsymbol{w}_\ell, \boldsymbol{\alpha}_\ell, \boldsymbol{\beta}_\ell, \mu_\ell)$$
$$= \sum_{\ell=1}^{L} \sum_{k=1}^{b} w_{\ell,k} k(t; \alpha_{\ell,k}, \beta_{\ell,k}, \mu_\ell) \quad (4)$$

where L is the number of rise and fade patterns, μ_ℓ is the location of ℓ-th pattern, $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_L]^\top \in \mathbb{R}^{b\times L}$, $\boldsymbol{w}_\ell \in \mathbb{R}^b$, $\boldsymbol{\Gamma_\alpha}$, and $\boldsymbol{\Gamma_\beta}$ denote the set of weight vectors, the set of $\boldsymbol{\alpha}$ and the set of $\boldsymbol{\beta}$, respectively. Note, Eq.(4) is a generalization of Eq.(2), as $b = 1$, these two equations are the same. Moreover, we also use the flipped version of the Gamma basis functions to represent slow rise pattern (See Figure 2(b)).

The optimization problem of fitting the time-series by a mixture of lifecycle models can be given as

$$\min_{\boldsymbol{W},\boldsymbol{\Gamma_\alpha},\boldsymbol{\Gamma_\beta},\boldsymbol{\mu}} \quad \sum_{t=1}^{T}(y_t - f(t;\boldsymbol{W},\boldsymbol{\Gamma_\alpha},\boldsymbol{\Gamma_\beta},\boldsymbol{\mu}))^2$$
$$\text{s.t.} \quad w_{k,\ell} > 0, \mu_k > 0, k = 1, 2, \ldots, L.$$

Similar to the PLC Lasso, we first relax the problem to a *convex* optimization problem. Specifically, we first fix the model parameters of basis functions $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by using a prior knowledge of rise and fade patterns and represent each time-series by T mixtures ($T > L$):

$$\begin{aligned} f(t;\boldsymbol{W}) &= \sum_{\ell=1}^{T}\sum_{k=1}^{b} w_{\ell,k} k(t;\widetilde{\alpha}_k,\widetilde{\beta}_k,\mu=\ell) \\ &= \sum_{\ell=1}^{T} \boldsymbol{w}_\ell^\top \boldsymbol{k}(t;\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}},\mu=\ell), \end{aligned}$$

where $\boldsymbol{k}(t;\boldsymbol{\alpha},\boldsymbol{\beta},\mu=\ell) = [k(t;\alpha_1,\beta_1,\mu=\ell),\ldots,k(t;\alpha_b,\beta_b,\mu=\ell)]^\top$ and $\widetilde{\boldsymbol{\alpha}}$ and $\widetilde{\boldsymbol{\beta}}$ are fixed parameters. Here, we use the same parameters $\widetilde{\boldsymbol{\alpha}}$ and $\widetilde{\boldsymbol{\beta}}$ for all PLC models.

In this model, since we place each rise and fade pattern model at every time t, they are highly overlapped each others over time. However, a spiking time-series tends to be sparse, that is, only a few $\boldsymbol{w}$ parameters are non-zero. To this end, we employ the group sparsity constraint for fitting.

Then, the final optimization problem can be written as:

$$\min_{\boldsymbol{W}} \quad \sum_{t=1}^{T}(y_t - \sum_{p=1}^{T}\boldsymbol{w}_p^\top \boldsymbol{k}(t;\widetilde{\boldsymbol{\alpha}},\widetilde{\boldsymbol{\beta}},\mu=p))^2 + \lambda\sum_{p=1}^{T}\|\boldsymbol{w}_p\|_2 \tag{5}$$
$$\text{s.t.} \quad w_{\ell,p} \geq 0,\ \ell = 1,2,\ldots,b,\ p = 1,2,\ldots,T,$$

where $\sum_{p=1}^{T}\|\boldsymbol{w}_p\|_2$ is the group regularizer and λ is the regularization parameter. The group regularizer consists of L_2-regularizer for $\boldsymbol{w}$ and L_1 regularizer between groups $\|\boldsymbol{w}_1\|_2, \|\boldsymbol{w}_2\|_2, \ldots, \|\boldsymbol{w}_T\|_2$. That is, the estimated parameter $\boldsymbol{w}$ tends to be dense within the group, and only a few groups (i.e., $\boldsymbol{w}$) take non-zero values. Since the peaks are sparsely located in a temporal sequence, the group regularizer is an appropriate choice.

Since the optimization problem of Eq.(5) is *convex* with respect to $\boldsymbol{W}$, Eq.(5) can be solved by using a state-of-the-art group lasso optimization solver. In this paper, we also employ the *dual augmented Lagrangian* (DAL) [Tomioka et al. 2011]. Since we use the group regularizer, we can select a small number of rise and fade patterns. We call this entire group lasso based fitting framework as *PLC Group Lasso*.

PLC Group Lasso does not need to specify a distribution for a PLC, and fitting with a PLC group can be more accurate than fitting with single PLC model. In this paper, we set α candidates to 1 and β candidates to $[0.1, 0.2, \ldots, 1.0]$ for each PLC basis function. In Figures 3(a2)-(c2), we illustrate the same buzz time-series and their fitted results by PLC Group Lasso, together with root squared mean error (RSME). Comparing with Figures 3(a1)-(c1), we observe that PLC Group Lasso tends to outperform PLC Lasso in fitting different buzz time-series. Figures 3(d) shows the RMSE comparison between the PLC Lasso and the PLC Group Lasso. The "All time-series" setup uses in total 534 time-series, while the "Buzz time-series" setup uses 58 time-series which are systematically extracted from the entire 534 time-series by checking $\max(\boldsymbol{y}) - \min(\boldsymbol{y}) > 0.3$. For the "All time-series" setup, the performance of the PLC Lasso tends to be better than the one of the PLC Group Lasso. Since the Twitter data includes non-buzz data in the 534 time-series, the performance of the both methods tends to be random. However, overall, both the PLC Lasso and the PLC Group Lasso perform well. For the "Buzz time-series" setup, the PLC Group Lasso tends to outperform the PLC Lasso. Thus, for modeling buzz time-series, the PLC Group Lasso is more suited than the PLC Lasso.

5. CLUSTERING BUZZES USING LIFECYCLE MODELS

According to Section 4, we can successfully model each time-series either as a mixture of L PLC models using PLC Lasso, or as a mixture of PLC groups using group lasso, then represent the sequence as a set of model parameters. In this section, we first propose three types of features for characterizing buzz time-series. Then, in order to discover underlying similar patterns in a set of buzz time-series, we propose a novel probabilistic graphical model, K-Mixture of Product Lifecycle (K-MPLC), to cluster N time-series based on their PLC parameters.

5.1. Feature Extraction

We obtain estimated model parameters $\boldsymbol{w}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\mu}$ for each buzz time-series using the PLC Lasso or PLC Group Lasso method. However, some of those parameters may not be useful for buzz clustering. For example, if buzz time-series are not aligned according to their peak positions, location of each PLC or each PLC group varies, and it may not be possible to obtain meaningful clusters using absolute PLC or PLC group locations. Thus, in this paper, we propose three types of effective and robust features for characterizing buzz time-series.

PLC weight parameter: To capture the shape information of a time-series, we use the normalized weight parameters obtained by PLC Lasso or PLC Group Lasso fitting $\widehat{\boldsymbol{w}}$. The number of weight parameters depends on the number of PLC models or PLC groups.

PLC shape parameter: we propose the following feature to discriminate sharp-tailed from heavy-tailed buzz sequences. For PLC Lasso, we use:

$$\tau = \frac{1}{\beta_{i_{\max}}},$$

where $i_{\max}$ is the index of the largest PLC with respect to weight $\boldsymbol{w}$, and this feature is designed to discriminate time-series in Figures 3(a1) from 3(b1); for PLC Group Lasso, we compute:

$$\tau = \frac{1}{|S_{i_{max}}|} \sum_{\ell \in |S_{i_{max}}|} \frac{\widehat{w}_{i_{max},\ell}}{\beta_{i_{max},\ell}},$$

where $i_{\max}$ is the index of the largest PLC group with respect to weight $\boldsymbol{w}$, and S is the set of indices of non-flipped Gamma parameter which decides whether the temporal pattern is sharp-tailed or heavy-tailed.

Standard deviation of PLC locations: we propose the following feature to discriminate buzz time-series with single peak from multiple peaks. For PLC Lasso, we leverage

$$\sigma = \sqrt{\frac{1}{L}\sum_{k=1}^{L}\widehat{w}_k(\mu_k - \mu')^2},$$

where $\mu_1, \ldots, \mu_L$ are locations of PLCs and $\mu' = \sum_{k=1}^{L}\widehat{w}_k\mu_k$ is their weighted mean. Note that, since we want to focus on PLCs with large weight parameters, we use the weighted variant of standard deviation. For PLC Group Lasso, we use the same formula as above, to compute the standard deviation of all PLC group locations.

In Figures 3(d), we illustrate the extracted feature values via PLC Lasso or PLC Group Lasso approach from the same temporal sequences, assuming each time-series is modeled with a mixture of 3 PLC models or a mixture of 3 PLC groups.

5.2. Clustering Buzzes using K-MPLC

To discover the underlying similar patterns of buzz time-series, we propose a probabilistic graphical model, K-Mixture of Product Lifecycle (K-MPLC), to cluster N time-series into K groups. Throughout this paper, we assume K is known.

Suppose that we are given N buzz time-series and their corresponding parameters $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N] \in \mathbb{R}^{L\times N}$, $\boldsymbol{\tau} \in \mathbb{R}^N$, and $\boldsymbol{\sigma} \in \mathbb{R}^N$. Given distinct characteristics over different parameter vectors, we use a Dirichlet distribution to model the weight vector $\boldsymbol{w}$ as its sum needs to be 1. Since τ and σ take positive values, we use the Gamma distribution to model them separately. The graphical model we propose is shown in Figure 4.

Specifically, the probability for each instance is:

$$p(\boldsymbol{w},\tau,\sigma|\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}',\boldsymbol{b}') = \sum_{k=1}^{K}\pi_k p(\boldsymbol{w}|\boldsymbol{\theta}_k)p(\tau|a_k, b_k)p(\sigma|a'_k, b'_k).$$

Here, $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^\top$ are the mixture weights, and

$$p(\boldsymbol{w}|\boldsymbol{\theta}_k) = C(\boldsymbol{\theta}_k)\prod_{i=1}^{L} w_i^{\theta_{ki}-1},$$

$$p(\tau|a_k, b_k) = \frac{b_k^{a_k}}{\Gamma(a_k)}\tau^{a_k-1}e^{-b_k\tau},$$

$$p(\sigma|a'_k, b'_k) = \frac{{b'_k}^{a'_k}}{\Gamma(a'_k)}\sigma^{a'_k-1}e^{-b'_k\sigma}$$

are Dirichlet and Gamma distributions, $C(\boldsymbol{\theta}) = \frac{\Gamma(\sum_{i=1}^{L}\theta_i)}{\Gamma(\theta_1)\Gamma(\theta_2)\ldots\Gamma(\theta_L)}$, and $\Gamma(a) = \int_0^\infty t^{a-1}e^{-t}\mathrm{d}t$.

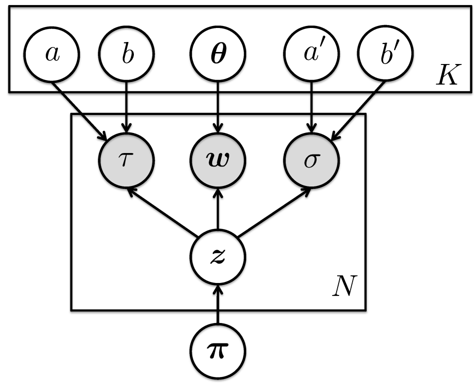


Fig. 4: Graphical Model of K-MPLC.

In this paper, we use maximum likelihood estimation to estimate the model parameters. Specifically, the optimization problem for the proposed model can be given as:

$$\max_{\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}',\boldsymbol{b}'} \quad \sum_{j=1}^{N} \log p(\boldsymbol{w}_j,\tau_j,\sigma_j|\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}',\boldsymbol{b}')$$

$$\text{s.t.} \sum_{k=1}^{K} \pi_k = 1, \theta_{ik} > 0, a_k > 0, b_k > 0, a'_k > 0, b'_k > 0.$$

In this paper we use the expectation-maximization (EM) algorithm to solve this problem. The complete likelihood function of the proposed model is given as:

$$\begin{aligned}
&p(\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\sigma},\boldsymbol{Z}|\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}',\boldsymbol{b}')\\
&= \prod_{k=1}^{K}\prod_{j=1}^{N}[\pi_k \cdot C(\boldsymbol{\theta}_k)\prod_{i=1}^{L} w_{ij}^{\theta_{ki}-1} \cdot \frac{{b_k}^{a_k}}{\Gamma(a_k)}\tau_j^{a_k-1}e^{-b_k\tau_j}\\
&\cdot \frac{{b'_k}^{a'_k}}{\Gamma(b'_k)}\sigma_j^{a'_k-1}e^{-b'_k\sigma_j}]^{z_{kj}},
\end{aligned}$$

where $\boldsymbol{Z}$ is the latent variable of the mixture model.

Then, the expectation of complete log-likelihood function (a.k.a., Q function) can be given as:

$$\begin{aligned}
&Q \triangleq E_{\boldsymbol{Z}}[\log p(\boldsymbol{W},\boldsymbol{\tau},\boldsymbol{\sigma},\boldsymbol{Z}|\boldsymbol{\pi},\boldsymbol{\Theta},\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}',\boldsymbol{b}')]\\
&= \sum_{k=1}^{K}\sum_{j=1}^{N}\{\gamma_{kj}\log\pi_k + \gamma_{kj}\log C(\boldsymbol{\theta}_k)\\
&+\gamma_{kj}(\theta_k - 1)\sum_{i=1}^{L}\log w_{ij} + \gamma_{kj}a_k\log b_k - \gamma_{kj}\log(\Gamma(a_k))\\
&+\gamma_{kj}(a_k-1)\log(\tau_j) - \gamma_{kj}b_k\tau_j + \gamma_{kj}a'_k\log b'_k\\
&-\gamma_{kj}\log(\Gamma(a'_k)) + \gamma_{kj}(a'_k-1)\log(z_{ij}) - \gamma_{kj}b'_k z_{ij}\},
\end{aligned} \tag{6}$$

where $\gamma_{kj} = E[z_{kj}]$ is the posterior probability.

E-step: The E-step is to compute posterior probability as:

$$\gamma_{kj} = \frac{\pi_k p(\boldsymbol{w}_j|\boldsymbol{\theta}_k)p(\tau_j|a_k, b_k)p(\sigma_j|a'_k, b'_k)}{\sum_{\ell=1}^{K} \pi_k p(\boldsymbol{w}_j|\boldsymbol{\theta}_\ell)p(\tau_j|a_\ell, b_\ell)p(\sigma_j|a'_\ell, b'_\ell)}. \quad (7)$$

M-step: M-step is to use maximum likelihood estimation to update each parameter over Q function. Due to the complexity of the Q function, the M-step contains several sub steps, if a sub step contains a close form, we will update it directly; otherwise, it requires iterations until it converges. In order to update the parameter $\boldsymbol{\theta}_k$ in the Dirichlet distribution, we use maximum likelihood estimation of Q function with respect to $\boldsymbol{\theta}_k$. However, maximum likelihood estimation of Dirichlet distribution is not available in closed-form, and thus we apply an iterative approach for parameter estimation [Minka 2000]. The basic idea is gradient ascent, which iteratively steps along positive gradient directions of Q function until convergence (Notice that $C(\boldsymbol{\theta}) = \frac{\Gamma(\sum_{i=1}^{L}\theta_i)}{\Gamma(\theta_1)\Gamma(\theta_2)\ldots\Gamma(\theta_L)}$). The gradient ascent with respect to $\boldsymbol{\theta}$ is computed as:

$$\frac{\partial Q}{\partial \theta_{ki}} = \sum_{j=1}^{N}\{\gamma_{kj}\psi(\sum_{i=1}^{L}\theta_{ki}) - \gamma_{kj}\psi(\theta_{ki}) + \gamma_{kj}\log(w_{ij})\},$$

where $\psi(w) = \frac{\partial \log\Gamma(w)}{\partial w}$ is the digamma function.

Then, we update each θ_{ki} until convergence as:

$$d_{ki} = d_{ki}^{(old)} + \eta\frac{\partial Q}{\partial d_{ki}}, \quad (8)$$

where $\eta > 0$ is the step size parameter. For choosing the step size of each gradient, we may use a line search method known as *Armijo's rule* [Patriksson 1999] .

Next, we show the maximum likelihood estimation of Gamma distribution. The gradient of the Q function with respect to a_k is given as:

$$\frac{\partial Q}{\partial a_k} = \sum_{j=1}^{N}\gamma_{kj}\left\{\log(b_k) - \frac{1}{\Gamma(a_k)}\frac{\partial\Gamma(a_k)}{\partial a_k} + \log(\tau_j)\right\}.$$

Since there is no closed form solution, we use a gradient ascent approach to update a_k and b_k as:

$$a_k = a_k^{(old)} + \eta\frac{\partial Q}{\partial a_k}, \quad b_k = \frac{\sum_{j=1}^{N}\gamma_{kj}a_k}{\sum_{j=1}^{N}\gamma_{kj}\tau_j}. \quad (9)$$

Similary, since the probability density function for σ is also Gamma, we can user a gradient ascent approach to update a'_k and b'_k as:

$$a'_k = a_k'^{(old)} + \eta\frac{\partial Q}{\partial a'_k}, \quad b'_k = \frac{\sum_{j=1}^{N}\gamma_{kj}a'_k}{\sum_{j=1}^{N}\gamma_{kj}\sigma_j}. \quad (10)$$

Next, we maximize the Q function with respect to $\boldsymbol{\pi}$. Since we need to take the sum-to-one constraint into account, we use a Lagrange multiplier λ as:

$$Q + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right).$$

Taking the derivative of the above equation with respect to π_k and equating it to zero, we have

$$\pi_k = \frac{1}{N} \sum_{k=1}^{K} \gamma_{kj}. \tag{11}$$

In summary, E-step is corresponding to Eq.(7), and M-step is corresponding to Eq.(8-11). Finally, we cluster the jth buzz time-series based on the posterior probability as:

$$\widehat{k}_j = \underset{k}{\operatorname{argmax}}\ \gamma_{kj}. \tag{12}$$

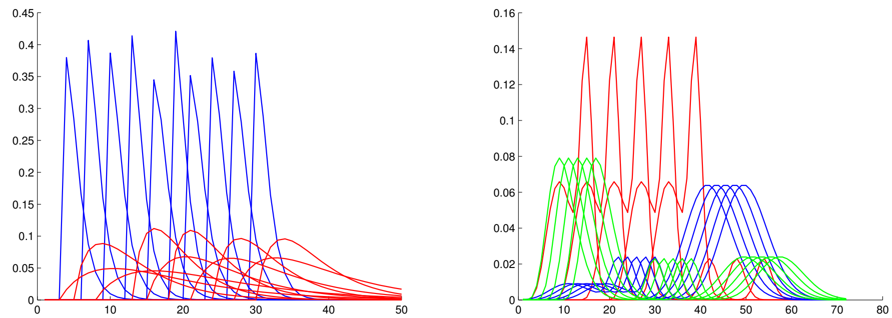
6. BUZZ CLUSTERING EXPERIMENTS

In order to cluster multiple time-series into several groups, it is natural for us to consider K-means algorithm and Gaussian mixture model (GMM) [Bishop and Nasrabadi 2006] as two baseline approaches. In addition, we compare the proposed methods with the following representative methods:

— **K-SC:** Yang and Leskovec [Yang and Leskovec 2011] proposed K-SC algorithm[2], which is the state-of-the-art algorithm for buzz time-series clustering. K-SC is proposed to group similar time-series via aligning peaks, shifting over time-axis, and scaling over frequency-axis. K-SC can cluster time-series that can be aligned by shifting and scaling operations, but fails for sharp or heavy-tailed temporal sequences. Moreover, if two time-series share similar peak shapes but differ in the intervals between the peaks, K-SC cannot identify these two time-series as the same cluster through simple shifting or scaling.
— **SpikeM:** SpikeM[3] [Matsubara et al. 2012], which is the state-of-the-art algorithm for buzz time-series modeling, is considered another strong baseline. As SpikeM does not handle time-series clustering, we concatenate all parameters extracted by SpikeM as features for GMM, and we denote it as *SpikeM+GMM*.
— **DTW+Spectral:** We also compare with a DTW-based anytime clustering algorithm[4] [Zhu et al. 2012], which is denoted as *DTW+Spectral*, whose basic idea is first to generate a DTW matrix to measure the non-metric distance between any two time-series, then apply a spectral clustering algorithm on the DTW matrix.

Given two buzz modeling approaches proposed in this paper, PLC Lasso and PLC Group Lasso, we could combine each of them with K-MPLC to handle the buzz clustering task, and note them as *PLC Lasso + K-MPLC* and *PLC Group Lasso + K-MPLC* respectively. In our experiments, for PLC Lasso approach, we set α to be 1 and β can take values of $[0.1, 0.2, \ldots, 1.0]$; similarly, for PLC Group Lasso approach, we set α to be 1 and β candidates to $[0.1, 0.2, \ldots, 1.0]$ to represent the basis PLC functions as well. In order to differentiate the contribution of buzz modeling methods (e.g. PLC Lasso or PLC Group Lasso) from K-MPLC algorithm, we also combine the extracted features with GMM for clustering, which are denoted as *PLC Lasso + GMM* and *PLC Group Lasso + GMM* respectively. All experiments in this section are based on 10-fold cross validation, and those data points with big circles on different figures indicate that this proposed method outperforms all baseline approaches statistically significantly.

[2] http://snap.stanford.edu/data/ksc.html
[3] http://www.cs.cmu.edu/~yasuko/SRC/spikeM.zip
[4] https://sites.google.com/site/anytimeclustering/



(a) Samples of synthetic dataset 1. (b) Samples of synthetic dataset 2.

Fig. 5: Illustrative examples of two synthetic datasets.

	ARI	NMI
K-Means	0.034	0.232
GMM	0.0111	0.0308
K-SC	0.001	0.017
PLC Lasso + K-MPLC	**1.0**	**1.0**
PLC Group Lasso + K-MPLC	**1.0**	**1.0**

Table I: Clustering results on synthetic dataset 1. (Bold font indicates the proposed methods outperform all baseline approach statistically significantly)

6.1. Datasets

Synthetic Datasets: For proof of concept, we build two synthetic datasets as follows. We first generate the first synthetic dataset to verify whether our proposed method can successfully discriminate time-series with single sharp peaks from curves with single heavy-tailed peaks. This dataset consists of two clusters: in cluster 1, we randomly generate 50 single sharp peak time-series based on the mixture of PLC models Eq.(2), whose α_1 follows a uniform distribution of [1.8,2.0], β_1 follows a uniform distribution of [0.8,1.2], and μ_1 follows a uniform distribution of [1,30]; while in cluster 2, we randomly generate 100 single heavy-tailed peak time-series, whose α_2 follows a uniform distribution of [1.8,2.0], β_2 follows a uniform distribution of [0.2,0.4], and μ_2 follows a uniform distribution of [1,30]. Figure 5 (a) shows 20 sample time-series in the synthetic dataset 1, where different color indicate corresponding cluster labels. Next, we generate the second synthetic dataset to verify whether the K-MPLC algorithm can successfully cluster time-series shifted along x-axis into the same group. The dataset contains 3 clusters, which are obtained from 3 distinct base time-series, and each time-series is generated by a mixture of 3 PLC models. We shift each base time-series along the x-axis, and generate another 9 time-series, which belong to the same cluster. In total, we generate a synthetic dataset with 30 time-series, which belong to 3 clusters. Figure 5 (b) shows 15 sample time-series of the synthetic dataset 2, and different colors indicate corresponding cluster labels.

Social Media Datasets: To evaluate the effectiveness of our proposed methods, we build two benchmark datasets from social media as follows. We first selected thousands of high frequency search queries as candidate buzz topics, and collected tweets

	ARI	NMI
K-Means	0.6678	0.6442
GMM	0.5812	0.6229
K-SC	0.7403	0.7103
PLC Lasso + K-MPLC	**1.0**	**1.0**
PLC Group Lasso + K-MPLC	**1.0**	**1.0**

Table II: Clustering results on synthetic dataset 2. (Bold font indicates the proposed methods outperform all baseline approach statistically significantly)

containing those keywords using Twitter API [5] from June 22nd to August 8th, 2012. Then, we generated a time-series for the topic within a time window, considering the number of mentions of a topic per hour. If the number of mentions at time t in a topic is 10 times higher than the average mention numbers in the past 48 hours, we regard the topic at that time as a buzz topic. According to [Kwak et al. 2010; Chang et al. 2014], hot topics on social media lose their attraction quickly, therefore, we select a time window of 72 hours, and obtain a 72-dimension time-series $\boldsymbol{y}$ for each buzz topic. Finally, we collected one general buzz dataset and one sports buzz dataset. The general buzz dataset contains 534 buzz time-series: most of them are celebrity names, such as *Michael Phelps* and *Tyler Perry*; the rest are event names, such as *Euro 2012* and *UFC 148*. The sports buzz dataset contains 124 buzz time-series, and all of buzz topics are sports celebrity names, such as *Alex Morgan*, *Larry Bird*, *Dennis Rodman*, and these 2 datasets have some overlapping. In our experiments, temporal sequences are not aligned according to their peaks. That is, the data set used in this paper is a more challenging dataset than the one used for K-SC [Yang and Leskovec 2011]. All time-series in both datasets are manually labeled into 5 clusters based on their distinct temporal shapes: time-series containing single sharp peak are labeled as 1; sequences with single heavy-tailed peak are labeled as 2; curves with double sharp peaks are labeled as 3; time-series containing double peaks (at least one is heavy-tailed) are labeled as 4; time-series with more than 3 peaks are labeled as 5. Figure 8 shows one representative time-series from each cluster.

6.2. Evaluation Metrics

In this paper, we use two widely used clustering evaluation metrics: Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI)[Kaufman and Rousseeuw 2005], and both metrics are the larger the better.

— **Adjusted Rand Index (ARI):** Rand Index (RI) is to measure the accuracy of clustering result, given the clustering ground truth:

$$RI = \frac{TP + TN}{TP + FP + TN + FN},$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. The adjusted Rand index (ARI) is the corrected-for-chance version of the Rand index, which is based on the contingency table. ARI is defined as

$$ARI = \frac{Index - ExpectedIndex}{MaxIndex - ExpectedIndex},$$

[5]http://apiwiki.twitter.com/

where it ranges from -1 to 1.

— **Normalized Mutual Information (NMI):** Let us denote C be the set of clusters obtained from the ground truth and C' be the clustering result. Their mutual information $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)},$$

where $p(c_i)$ and $p(c'_j)$ refer to the probability of a data point belonging to cluster c_i and c'_j, and where $p(c_i, c'_j)$ is the joint probability. The normalized mutual information (NMI) is defined as

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))},$$

where $H(C)$ and $H(C')$ refers to the entropy of C and C'. After normalization, the NMI score ranges from 0 to 1.

6.3. Experiments on Synthetic Datasets

To demonstrate the effectiveness of our proposed methods, we first evaluate them using synthetic datasets. Table I shows clustering results on the synthetic dataset 1. As can be observed, K-MPLC combining with either PLC Lasso or PLC Group Lasso, obtain perfect result on this dataset. K-means and GMM perform poorly, since both algorithms cannot group similar curves if they are not aligned along x-axis. For K-SC, when curves from two different clusters have different shapes yet similar shapes after re-scaling, simply shifting along x-axis or scaling along y-axis is not sufficient for K-SC to group those time-series into the right clusters.

Table II shows clustering results on the synthetic dataset 2, where K-MPLC combining with either PLC Lasso or PLC Group Lasso, achieve perfect result as well. These synthetic experimental results clearly illustrate that K-MPLC can discriminate sharp peaks from heavy-tailed peaks and group time-series with multiple peaks correctly.

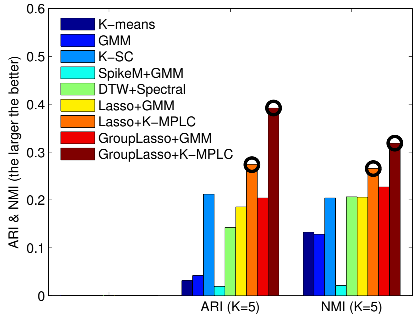
6.4. Experiments on General Buzz Dataset

Figure 6 (a) represents the clustering results of different algorithms on the general buzz dataset. It clearly shows that *PLC Group Lasso + K-MPLC* performs the best among all algorithms, while *PLC Lasso + K-MPLC* performs the second best, and both improvements are statistically significant. K-SC and *DTW+Spectral* perform well among existing methods, and their results are comparable with *PLC Lasso + GMM* or *PLC Group Lasso + GMM*, which also indicates the contribution of K-MPLC model.

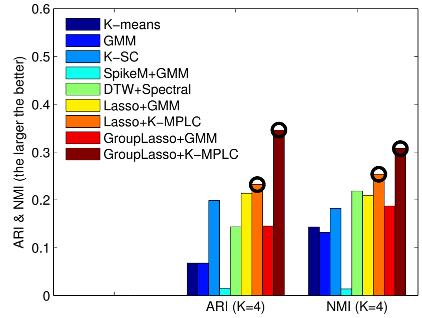
To further compare the proposed algorithm with other baseline approaches, we would like to evaluate them on more datasets. However, since building a real dataset from social media web sites is really expensive and time-consuming, we change the number of existing cluster labels to obtain more experimental results.

— **4-label clustering ground truth:** we combine the sequences originally labeled as 3 and 4 into the same cluster, while the remaining is the same.

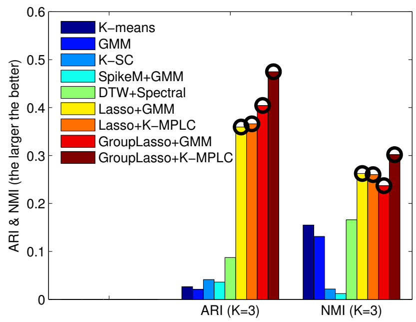
— **3-label clustering ground truth (different number of peaks):** we combine the sequences originally labeled as 1 and 2 into one cluster, combine the sequences originally labeled as 3 and 4 into another same cluster, and leave the remaining the same.



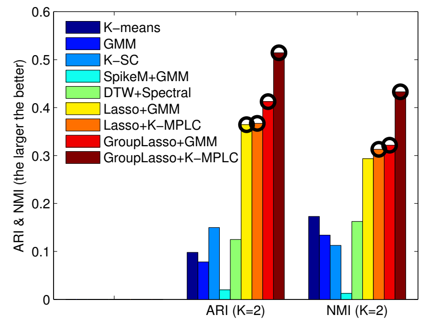
(a) Clustering results on 5-label ground truth.



(b) Clustering results on 4-label ground truth.



(c) Clustering results on 3-label ground truth.

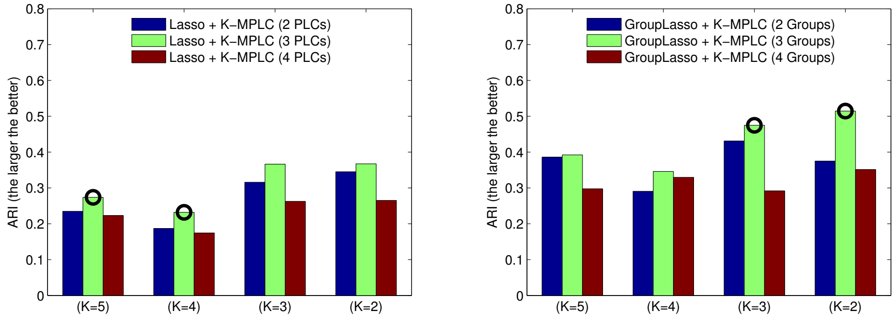


(d) Clustering results on 2-label ground truth.

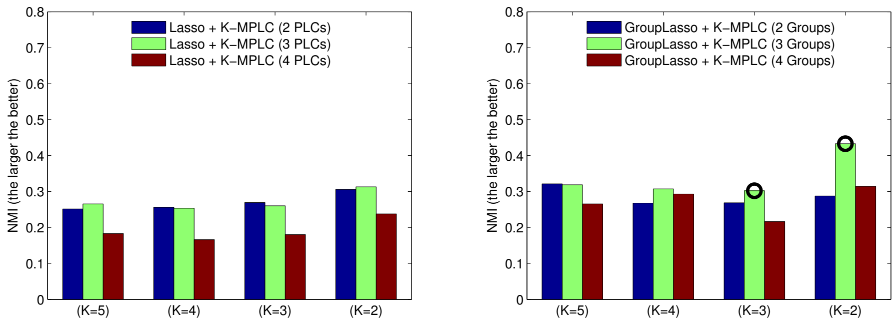
Fig. 6: Clustering results on General Buzz Dataset with different Labels. (Big circles indicate the proposed method outperforms all baseline approach statistically significantly)

— **2-label clustering ground truth (Single peak vs. multiple peaks):** we combine the sequences originally labeled as 1 and 2 into one cluster, and combine the sequences originally labeled as 3, 4 and 5 into the other cluster.

Comparing with Figure 6 (a)-(d), overall, the observations are quite consistent: *PLC Group Lasso + K-MPLC* always performs the best, and *PLC Lasso + K-MPLC* performs the 2nd best, and all improvements are statistically significant; although *PLC Lasso + GMM* and *PLC Group Lasso + GMM* also make some improvements, but not always statistically significant; *DTW+Spectral* and K-SC perform well among existing approaches, and the DTW based approach is relatively more robust than K-SC on our datasets. Notice that, with decreasing of cluster numbers K, the difficulty of clustering also decreases. In our experiment, the absolute evaluation metrics of *PLC Group Lasso + K-MPLC* is increasing (ARI is 0.3921 when $K = 5$, and ARI is 0.5145 when $K = 2$), which also shows the effectiveness of our proposed method. On the other hand, K-SC and *DTW+Spectral* perform worse when K is smaller, such as $K = 2$. One possible reason is that one of two clusters consists of single sharp peak and single fat peak sequences. More specifically, for K-SC and *DTW+Spectral*, the distance between single sharp peak and single fat peak sequences can be larger than the one between single



(a) ARI over different PLC numbers. (b) ARI over different PLC group numbers.



(d) NMI over different PLC numbers. (c) NMI over different PLC group number.

Fig. 7: Clustering Results on General Buzz Dataset with Different PLC or PLC group Numbers . (Big circles indicate the improvement is statistically significant comparing with different PLC or PLC group numbers)

sharp peak and multiple peaks sequences, and thus single peak and multiple peaks tends to be clustered together.

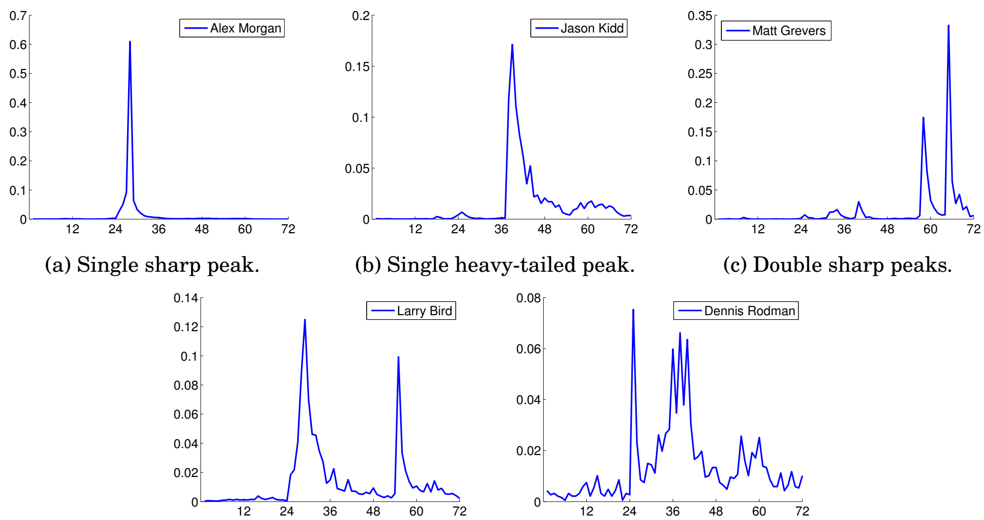
Figure 7 illustrates the clustering results with different PLC numbers L using *PLC Lasso + K-MPLC* or different PLC Group numbers using *PLC Group Lasso + K-MPLC*. Interestingly, modeling each sequence with a mixture of 3 PLC models consistently outperforms that of using 2 or 4 PLC models, although the outperforming is not always statistically significant. One possible explanation is that in our general buzz dataset, as the ground truth is to differentiate curves with 1 peak or 2 peaks from 3 or more peaks, representing each time-series as a mixture of 3 PLC models is the most appropriate, while leveraging 4 PLC models might bring in more noisy information, and leveraging 2 PLC models are least robust; however, when using PLC Group Lasso, as each rise-and-fade pattern is represented by a group of basis PLC models, and different basis PLC models in different PLC groups are more likely overlapped, which bring more noisy information for the buzz clustering task, as a result, PLC Group Lasso performs best when group number is 2.

6.5. Experiments on Sports Buzz Dataset

To investigate the relationship between clustering algorithms and buzz topical content, we evaluate K-MPLC on the sports buzz dataset, which consists of 124 sports

Table III: Buzz topic examples in each cluster.

	#Topics	Representative Sports Buzz Topics
cluster 1	52 (20)	Alex Morgan, Anderson Silva, Ben Gordon, Jason Terry, Liu Xiang, Perry Jones III, Ray Allen, Silas Redd, ...
cluster 2	29 (7)	Hope Solo, Jason Kidd, Nastia Liukin, Steve Nash, Tyler Clary, Venus Williams ...
cluster 3	9 (8)	Asafa Powell, Matt Grevers, Usain Bolt, Zara Phillips ...
cluster 4	20 (10)	David Beckham, Larry Bird, Michael Johnson, Michael Phelps, Oscar Pistorius ...
cluster 5	14 (7)	Dennis Rodman, Jeremy Lin, Kobe Bryant, Lance Armstrong, Brett Favre ...



(a) Single sharp peak. (b) Single heavy-tailed peak. (c) Double sharp peaks.

(d) Double peaks (at least one is heavy-tailed). (e) 3 or more peaks.

Fig. 8: Time-series example of each cluster.

buzz time-series, and all of them are sports celebrity names. As part of the dataset was collected during the London 2012 Olympics, i.e., there is a large number of Olympics related celebrities included. We explicitly labeled all sports buzz topics as Olympics topics or non-Olympics topics, and 52 out of 124 sports buzz topics are about Olympics celebrities. Comparing with Olympics buzz topics with non-Olympics buzz topics, 48.1% of Olympics buzz topics have 2 or more peaks, while only 25.0% of non-Olympics buzz topics have 2 or more peaks, which is due to the intensity of Olympics event: an athlete needs to compete several time (e.g., semi-final, final), and sometimes in multiple categories (e.g., different swimming events). As a result, Olympics celebrities tends to attract more mentions or discussions.

Table III shows the number of topics and samples of topics in each cluster center obtained by K-MPLC, where the number in the bracket in the second column indicates

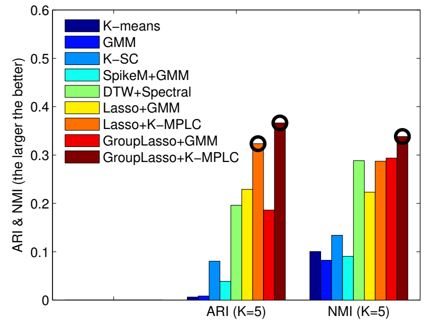
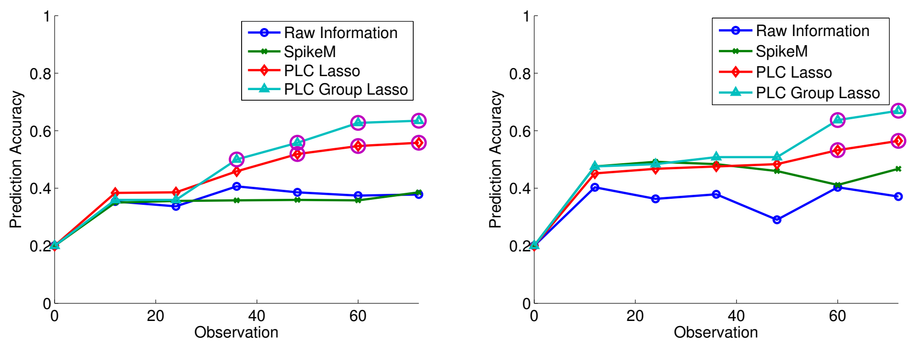


Fig. 9: Clustering results on Sports Buzz Dataset. (Big circles indicate the proposed method outperforms all baseline approach statistically significantly)



(a) Prediction on General Buzz Dataset. (b) Prediction on Sports Buzz Dataset.

Fig. 10: Prediction Experimental Results. (Big circles indicate the proposed method outperforms two baseline approach statistically significantly)

the number of Olympics topics in that cluster. As can be seen, the cluster 4 and 5 contain more superstar sports celebrities than cluster 1 or 2, which indicates the number of buzz peaks is somehow correlated with celebrity popularity.

Figure 8 shows one representative time-series from each cluster, where the topics names are also listed in each sub figure. For example, Figure 8 (e) illustrates the occurrence of *Dennis Rodman* within a 72-hour period. Figure 9 illustrates the clustering results of different algorithms on the sports buzz dataset, where the observation is generally consistent to the clustering results on the general buzz dataset.

7. BUZZ TYPE PREDICTION EXPERIMENTS

Prediction based on time-series data extracted from social media is a challenging task, which is mainly used to predict stock market changes [Johan Bollen 2011] or political opinions [O'Connor et al. 2010; Conover et al. 2011]. Unlike stock market prediction, whose focus is to precisely predict the stock price value in the next adjacent time stamp, we try to predict whether there will be a burst in the near future. Our motivation is that there exists several major temporal shapes of attention of social media content [Yang and Leskovec 2011], therefore, as long as we can successfully classify a buzz time-series belonging to which major temporal shapes based its partial segment, we could also effectively predict burst. As a consequence, we cast the buzz type prediction challenge as a classification problem: given our existing labeled datasets for buzzes clustering, we predict what is the label of each unknown buzz topics via cross validation. For example, given a partially observed time-series, if it is classified as category 1 (single sharp peak), it is less likely to attract a new burst in the next a few hours; while it is classified as category 5 (3 or more peaks), a new burst is more likely to be predicted in the near future. For those buzzing events, due to the dramatic volume change over time, most traditional time-series models, such as Auto-Regressive model [Blischke and Murthy 2000] or Birth-death process [Latouche and Ramaswami 1999] fail to generate a reliable prediction.

In this section, we try to solve the buzz type prediction task within a supervised learning framework. Given the whole sequence of each time-series data, we divide each sequence into 2 segments: *past segment* and *future segment*. For training data, both past and future segments are observed, while for testing data, only past segment is observed. We treat the past segment as a vector of features, the ground truth of the training data as the learning target, and we train a model; in testing, given an observed segment, the supervised learning model can be used to make a prediction. In this paper, we use logistic regression [Bishop and Nasrabadi 2006] as the supervised learning algorithm. As a baseline, we use each time-series as a feature vector; we also use SpikeM parameters extracted from each past segment as a feature vector; for our proposed method, we use the PLC Lasso parameters or PLC Group Lasso parameters extracted from each past segment as a feature vector.

Figure 10 illustrates the buzz type prediction results on both datasets using leave-1-out cross validation, and those data points with big circles indicate that our proposed method outperforms both baseline approaches statistically significantly. It is clear the PLC Group Lasso approach performances the best. When the full time-series (72 hours) is observed, the prediction accuracy on General Buzz Dataset is 0.6348, while the prediction accuracy on Sports Buzz Dataset is 0.6694; when the 60 hours partial time-series are observed, the prediction accuracy on General Buzz Dataset is 0.6273, while the prediction accuracy on Sports Buzz Dataset is 0.6371; when the 36 hours partial time-series are observed, the prediction accuracy on General Buzz Dataset is 0.5000, while the prediction accuracy on Sports Buzz Dataset is 0.5081. In the other words, although different buzz events might be independent each other, their time-series follow several patterns and such patterns can be predicted via supervised learning approach. Given partial observed testing data, the overall accuracy is close to the accuracy with fully observed testing data. As the General Buzz Dataset contains more temporal sequences than the Sports Buzz Dataset, more data points based on General Buzz Dataset are statistically significantly better than baseline approaches. Notice in Figure 10(b), the prediction result using PLC Lasso is not monotonic increasing. One possible reason is due to the noisy data, and we will improve the buzz prediction in future work.

8. CONCLUSION AND FUTURE WORK

In this paper, we presented our study to model the sudden spikes and heavy-tailed patterns of buzz events on social media with product lifecycle models (PLC). Specifically, we model a time-series with PLC mixture model or PLC group mixture model, and propose an efficient lasso or group lasso based optimization approach for parameter estimation. Then, we proposed a novel probabilistic graphical model to cluster buzz time-series with similar temporal patterns into the same group based on the features obtained from PLC mixture model or PLC group mixture model. The novelty of our proposed approaches is that it can effectively distinguish time-series with sharp peaks from time-series with heavy-tailed peaks, which is in practice a favorable property in buzz time-series clustering. Through evaluation on two buzz datasets, our proposed method significantly outperforms the current state-of-the-art algorithms for both buzz clustering and buzz type prediction tasks.

Our work can be extended as follows. First, in this paper, we only deal with a segment of time-series within a fix size of time window. Second, we will leverage PLC models to split a long time-series into different segments, and combine with social media content to mine more interesting findings.

ACKNOWLEDGMENTS

This research was supported by the NSF research grants IIS-1134990, IIS-1254206, and U.S. Defense Advanced Research Projects Agency (DARPA) under Social Media in Strategic Communication (SMISC) program, Agreement Number W911NF-12-1-0034. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government. We also acknowledge the research grant MEXT KAKENHI 16K16114 from Japan.

REFERENCES

BATISTA, G., WANG, X., AND KEOGH, E. J. 2011. A complexity-invariant distance measure for time series. In *Proceedings of SDM*.

BAUCKHAGE, C., KERSTING, K., AND HADIJI, F. 2013. Mathematical models of fads explain the temporal dynamics of internet memes. In *Proceedings of EMNLP*.

BISHOP, C. M. AND NASRABADI, N. M. 2006. *Pattern recognition and machine learning*. springer New York.

BLISCHKE, W. R. AND MURTHY, D. N. P. 2000. *Reliability Modeling, Prediction and Optimization*. John Wiley and Sons Inc.

CHANG, Y., TANG, L., INAGAKI, Y., AND LIU, Y. 2014. What is tumblr: A statistical overview and comparison. *SIGKDD Explorations 16,* 1, 21–29.

CHANG, Y., YAMADA, M., ORTEGA, A., AND LIU, Y. 2014. Ups and downs in buzzes: Life cycle modeling for temporal pattern discovery. In *Proceedings of ICDM*.

CHOI, H. AND VARIAN, H. 2009. Predicting the present with google trends. In *Technical report, Google*.

CONOVER, M., RATKIEWICZ, J., FRANCISCO, M., GONCALVES, B., MENCZER, F., AND FLAMMINI, A. 2011. Political polarization on twitter. In *Proceedings of ICWSM*.

CRANE, R. AND SORNETTE, D. 2008. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences 105,* 41, 15649–15653.

CUI, P., LIU, H., AGGARWAL, C., AND WANG, F. 2016. Uncovering and predicting human behaviors. *IEEE Intelligent Systems 31,* 2, 77–88.

GRUHL, D., GUHA, R. V., KUMAR, R., NOVAK, J., AND TOMKINS, A. 2005. The predictive power of online chatter. In *Proceedings of KDD*.

GUERINI, M., STRAPPARAVA, C., AND OZBAL, G. 2011. Exploring text virality in social networks. In *Proceedings of ICWSM*.

HAIDER, P., CHIARANDINI, L., AND BREFELD, U. 2012. Discriminative clustering for market segmentation. In *Proceedings of KDD*.

HAWKES, A. AND OAKES, D. 1974. A cluster representation of a self-exciting process. *Journal of Applied Probability 11*, 493–503.

HONG, L., DOM, B., GURUMURTHY, S., AND TSIOUTSIOULIKLIS., K. 2011. A time-dependent topic model for multiple text streams. In *Proceedings of KDD*.

ISAIC-MANIU, A. AND VODA, V. G. 2008. On a model regarding the product life-cycle. *Management and Marketing 3,* 3, 87–96.

JIANG, S., FERREIRA, J., AND GONZALEZ, M. C. 2012. Clustering daily patterns of human activities in the city. In *Data Mining and Knowledge Discovery*. Volume 25, Number 1.

JOHAN BOLLEN, HUINA MAO, X.-J. Z. 2011. Twitter mood predicts the stock market. In *Journal of Computational Science*. 1–8.

KAUFMAN, L. AND ROUSSEEUW, P. J. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience.

KLEINBERG, J. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery 7,* 4, 373–397.

KUMAR, M., PATEL, N. R., AND WOO, J. 2002. Clustering seasonality patterns in the presence of errors. In *Proceedings of KDD*.

KUMAR, R., MAHDIAN, M., AND MCGLOHON, M. 2010. Dynamics of conversations. In *Proceedings of KDD*.

KUMAR, R., NOVAK, J., RAGHAVAN, P., AND TOMKINS, A. 2003. On the bursty evolution of blogspace. In *Proceedings of WWW*.

KWAK, H., LEE, C., PARK, H., AND MOON, S. B. 2010. What is twitter, a social network or a news media. In *Proceedings of WWW*.

LATOUCHE, G. AND RAMASWAMI, V. 1999. *Quasi-Birth-and-Death Processes*, 1st edition ed. Introduction to Matrix Analytic Methods in Stochastic Modelling. ASA SIAM.

LERMAN, K. AND GHOSH, R. 2010. Information contagion: an empirical study of the spread of news on digg and twitter social networks. In *Proceedings of ICWSM*.

LEVITT, T. 1965. Exploit the product life cycle. *Harvard Business Review 43*, 81–94.

LI, L. AND PRAKASH, B. A. 2011. Time series clustering: Complex is simpler. In *Proceedings of ICML*.

LIN, J., VLACHOS, M., KEOGH, E. J., AND GUNOPULOS, D. 2004. Iterative incremental clustering of time series. In *Proceedings of EDBT*.

MATSUBARA, Y., SAKURAI, Y., PRAKASH, B. A., LI, L., AND FALOUTSOS, C. 2012. Rise and fall patterns of information diffusion: Model and implications. In *Proceedings of KDD*.

MEI, Q., LIU, C., SU, H., AND ZHAI, C. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proceedings of WWW*.

MINKA, T. 2000. Estimating a dirichlet distribution. In *Technical report, MIT*.

O'CONNOR, B., BALASUBRAMANYAN, R., ROUTLEDGE, B. R., AND SMITH, N. A. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of ICWSM*.

PATRIKSSON, M. 1999. *Nonlinear Programming and Variational Inequality Problems:: A Unified Approach*. Vol. 23. Springer.

PETRESCU, E. 2009. A statistical distribution useful in product life-cycle modeling. *Management and Marketing 4,* 2, 165–170.

RAKTHANMANON, T., KEOGH, E. J., LONARDI, S., AND EVANS, S. 2011. Time series epenthesis: Clustering time series streams requires ignoring some data. In *Proceedings of ICDM*.

SUN, J. AND TANG, J. 2011. A survey of models and algorithms for social influence analysis. *Social Network Data Analysis (Springer)*, 177–214.

TAKAHASHI, R., MIZUTA, H., ABE, N., KENNEDY, R., JEFFS, V. J., SHAH, R., AND CRITES, R. H. 2013. Collective response spike prediction for mutually interacting consumers. In *Proceedings of ICDM*.

TIBSHIRANI, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. 58*, 267–288.

TOMIOKA, R., SUZUKI, T., AND SUGIYAMA, M. 2011. Super-linear convergence of dual augmented lagrangian algorithm for sparsity regularized estimation. *JMLR 12*, 1537–1586.

YANG, J. AND LESKOVEC, J. 2011. Patterns of temporal variation in online media. In *Proceedings of WSDM*.

YU, L., CUI, P., WANG, F., SONG, C., AND YANG, S. 2015. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. In *Proceedings of ICDM*.

YUAN, M. AND LIN, Y. 2006. Model selection and estimation in regression with grouped variables. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 68*, 49–67.

ZHANG, T., CUI, P., SONG, C., ZHU, W., AND YANG, S. 2016. A multiscale survival process for modeling human activity patterns. *PLOS ONE 11,* 3.

ZHU, Q., BATISTA, G., RAKTHANMANON, T., AND KEOGH, E. 2012. A novel approximation to dynamic time warping allows anytime clustering of massive time series datasets. In *Proceedings of SDM*.