The Web as a Jungle: Non-Linear Dynamical Systems for Co-evolving Online Activities

Yasuko Matsubara
Kumamoto University
yasuko@cs.kumamoto-u.ac.jp

Yasushi Sakurai
Kumamoto University
yasushi@cs.kumamoto-u.ac.jp

Christos Faloutsos
Carnegie Mellon University
christos@cs.cmu.edu

ABSTRACT

Given a large collection of co-evolving online activities, such as searches for the keywords "Xbox", "PlayStation" and "Wii", how can we find patterns and rules? Are these keywords related? If so, are they competing against each other? Can we forecast the volume of user activity for the coming month?

We conjecture that online activities compete for user attention in the same way that species in an ecosystem compete for food. We present ECOWEB, (i.e., Ecosystem on the Web), which is an intuitive model designed as a non-linear dynamical system for mining large-scale co-evolving online activities. Our second contribution is a novel, *parameter-free*, and *scalable* fitting algorithm, ECOWEB-FIT, that estimates the parameters of ECOWEB.

Extensive experiments on real data show that ECOWEB is *effective*, in that it can capture long-range dynamics and meaningful patterns such as seasonalities, and *practical*, in that it can provide accurate long-range forecasts. ECOWEB consistently outperforms existing methods in terms of both accuracy and execution speed.

Categories and Subject Descriptors: H.2.8 [**Database management**]: Database applications–*Data mining*

Keywords: Ecosystem; Time-series; Non-linear; Parameter-free;

1. INTRODUCTION

The increasing volume of online user activity represents a vital new opportunity for data scientists and analysts to measure the collective behavior of social, economic, and other important evolutions [57, 56, 27, 13].

Given real-time, online user activity sequences, such as the search volume for the keywords "Xbox" and "PlayStation", how can we find patterns and rules to perform, e.g., sociological, behavioral, and even marketing research? If we know nothing about the sequences, we could (and should) try using Fourier, Wavelets, AR, Kalman filters and the other time series analysis tools. However, we are told that the sequences correspond to online user activity (e.g. the search volume for a keyword) — Can we do better than the existing methods?

This is exactly the idea behind our work. We conjecture that the volume per keyword/activity will behave like a species in an "ecosystem". It will compete with other species for food and also exhibit seasonal behavior. Here we propose that "food" corresponds to user resources: given a set of users and their resources (e.g., attention, time, money), the d keywords/activities compete for the user resources.

In this paper, we present an intuitive model, namely ECOWEB, which provides a good description of large collections of co-evolving online activities. [1] In short, the problem we wish to solve is as follows:

INFORMAL PROBLEM 1. *Given a large collection of co-evolving sequences $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d\}$, which consists of d keywords/activities of duration n, where each record $x_i(t)$ corresponds to a user activity (e.g., queries, time/dollars spent) for keyword i at time tick t, we want to*

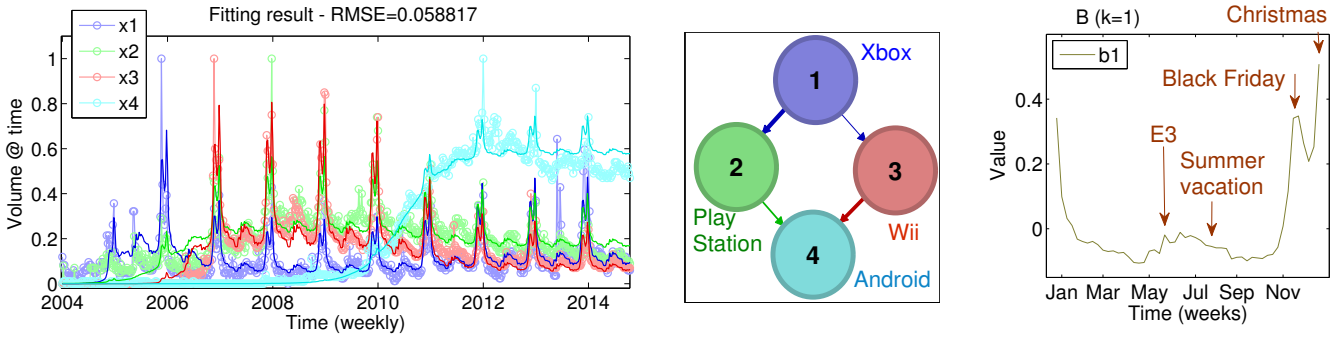
- *detect competition (e.g., "Xbox" vs. "PlayStation")*
- *find seasonal events (e.g., Christmas, summer vacations)*
- *forecast future dynamics*

Preview of our results. Figure 1 shows our discoveries related to the video game industry consisting of $d = 4$ activities, namely, the search volumes for "Xbox" ($\boldsymbol{x}_1$), "PS2, PS3" ($\boldsymbol{x}_2$), "Wii" ($\boldsymbol{x}_3$), and "Android" ($\boldsymbol{x}_4$), taken from *Google*, [2] and spanning over a decade (2004-2014), with weekly measurements. ECOWEB discovered the following important patterns:

- *Long-term fitting:* Figure 1 (a) shows the original volume of the four activities/keywords as circles, and our fitted model as solid lines. Notice that our fit is even visually very good, and it detects seasonalities and up- or down-trends: For example, our model fitted the success of "Wii" (which launched in 2006 and apparently drew attention from the competing "Xbox"). Similarly, it fitted the fall in the popularity of "Wii" in 2011, which coincided with the ascent of "Android", possibly indicating that mobile and social games attracted the attention of Wii gamers.
- *Interspecies interaction:* Recently, video games have been facing increasing competition (from online/social games), and our model automatically identifies this latent competition: Figure 1 (b) shows the interaction network that captures the interaction between the four activities/keywords. Edges indicate interaction/competition between two keywords; the thicker the edge, the stronger the interaction. For example, the red edge from "Wii" to "Android" indicates that the latter is drawing attention away from "Wii". Similarly, "Xbox" has strong connections to "PlayStation" and "Wii" (blue edges), summarizing the fact that the attention for "Xbox" was anti-correlated with "Wii" and "PlayStation", during 2007-2010.

WWW 2015, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741092.

[1] Available at
http://www.cs.kumamoto-u.ac.jp/~yasuko/software.html
[2] http://www.google.com/trends/



(a) Fitting result (solid lines) vs. original data (in circles) (b) Interaction network (c) seasonal activities

Figure 1: Modeling power of EcoWeb: (a) Our model (solid lines) fits the original data (in circles) very well, and (b) it reveals latent interaction networks, such as "Xbox" vs. "PlayStation" and "Wii" vs. "Android", as well as (c) seasonal activities (i.e., they all peak on Black Friday and at Christmas). Moreover, our fitting algorithm is *fully automatic*, requiring no user intervention.

- *Seasonal activities:* Figure 1 (c) succinctly summarizes the seasonality of all four keywords. There is a clear yearly periodicity, with peaks every November ("Black Friday") and December (Christmas); a small peak in June (coinciding with the Electronic Entertainment Expo (E3), an annual trade show for video games); and sustained, medium-level activity during the summer vacations.

Contributions. We propose EcoWeb, a succinct, yet powerful model, which is inspired by the competition between biological species, and which captures the evolution of multiple online activities. EcoWeb has the following desirable properties:

1. **Effective**: EcoWeb captures long-range dynamics, important patterns and seasonalities that agree with human intuition.
2. **Automatic**: EcoWeb-Fit requires no training set, no parameters to tune, no user intervention.
3. **Scalable**: It is carefully designed to be linear on the input size.
4. **Practical**: It can provide long-term forecasting, outperforming existing methods (Sections 6 and 7).

Outline. The rest of the paper is organized in the conventional way: Section 2 discusses related work and Section 3 describes some fundamental concepts. In Section 4 and Section 5, we describe our proposed model and algorithms. Sections 6 and 7 describe our experimental results and applications. We conclude in section 8.

2. RELATED WORK

The related work falls into the following large subgroups:

Similarity search and forecasting: There is a lot of interest in mining time series and data streams [42, 1, 20, 53, 11, 48]. Traditional approaches applied to data mining include auto-regression (AR), linear dynamical systems (LDS), Kalman filters (KF) and their variants [18, 30, 51]. Similarity search and pattern discovery in time sequences have also attracted huge interest [52, 25, 50, 48, 9].

Large-scale sequence mining: Here, TriMine [33] is a scalable method for forecasting co-evolving multiple (thousands of) sequences, while, FUNNEL [35] is a non-linear model for spatially coevolving epidemic tensors. [32] developed a fully-automatic mining algorithm for co-evolving sequences. Rakthanmanon et al. [46] proposed a similarity search algorithm for "trillions of time series" under the DTW distance. Yang et al. [55] developed a new model for mining time-evolving event sequences. As regards parameter-free mining, the work in [5, 7] focused on summarization and clustering based on the MDL principle.

Table 1: Capabilities of approaches. Only our approach meets all specifications.

	LV	DWT	AR++	AutoPlait	EcoWeb
Domain knowledge	√				√
Co-evolution	√			√	√
Periodicity		√	√	√	√
Parameter free				√	√
Forecasting			√		√

Social media analysis: Analyses of social media and online user behavior has attracted considerable interest [44, 24, 22, 49, 31, 10, 28, 3]. Gruhl et al. [15] explored online "chatter" (e.g., blogging) activity, and measured the actual sales ranks on Amazon.com. Ginsberg et al. [13] examined a large number of search engine queries tracking influenza epidemics. They reported that the evolutions of search engine keywords are highly correlated with actual flu virus activity. The work reported in [8, 45, 14] studied keyword volume, to predict consumer behavior.

Spikes and propagation: The work in [34] studied the rise and fall patterns in the information diffusion process through online social media. The work in [12] investigated the effect of revisits on content popularity, while [47] focused on the daily number of active users. Prakash et al. [43] described a case where two competing products/ideas spreading over the network, and provided a theoretical analysis of the propagation model (winner takes all: WTA) for arbitrary graph topology.

Economic models: Leontief [26] developed the "input-output model", which represents an economy as d interdependent industries (i.e., sectors). This model represents an economy as a system of equations, with producer-consumer relationships (analogous to prey-predator equations).

Contrast with competitors. Table 1 illustrates the relative advantages of our method. Only our EcoWeb matches all requirements, while,

- The Lotka-Volterra (LV) model [36], the logistic function (LF) [6], the susceptible-infected (SI) model [2], and other non-linear equations [17, 38, 43, 35] incorporate domain knowledge, however, they are not intended to capture co-evolving user activities and seasonal patterns.
- Wavelets and Fourier transforms (i.e., DWT, DFT, DCT) focus on a single time sequence, and cannot detect interaction between multiple co-evolving sequences.
- The traditional AR, ARIMA and related forecasting methods including AWSOM [40], PLiF [30] and TriMine [33] are *fundamentally* unsuitable for our setting, because they are based

on linear equations, while we employ *non*-linear equations. Moreover, (a) they can not incorporate domain knowledge, and (b) most of them require parameter tuning.
- AUTOPLAIT [32], SWAB [21] and pHMM [54] have the ability to capture the dynamics of sequences and perform segmentation, however, they cannot model the long-range evolution of multiple time series.

In short, none of the existing methods focuses specifically on the automatic mining of non-linear dynamics in co-evolving online activities.

3. BACKGROUND - ECOLOGICAL CONSIDERATIONS

Let us consider a biological ecosystem by analogy with a jungle where herbivores feed on plants, carnivores feed on other animals, and so on. How many spider monkeys should we expect to have in the next time tick, given the current count of spider monkeys, bananas, squirrel monkeys, etc? This is exactly the focus of population ecology, namely, to develop mathematical models to predict the evolution of the population of each species [39, 38].

Competition between species. There are two major mechanisms that the equations try to model: (a) un-restricted growth, i.e., with infinite resources, every squirrel monkey generates r offspring in each time tick, and (b) competition, i.e., with finite resources, the "carrying capacity" K of the environment is the maximum number of squirrel monkeys it can support.

There is competition between the members of the same species (two squirrel monkeys competing for fruit), as well as between different species (e.g., squirrel monkey vs. spider monkey, all competing for fruit). This competition is what keeps the population size of a species from exploding exponentially: If an ecosystem has too many squirrel monkeys and too few fruits, competition for those resources increases, throttling the growth of the squirrel monkey population.

One of the simplest models that captures the above phenomena is the Lotka-Volterra population model of competition [37]. It describes the interaction of d species with the following non-linear differential equations:

$$\frac{dP_i}{dt} = r_i P_i \left(1 - \frac{\sum_{j=1}^{d} a_{ij} P_j}{K_i}\right), \quad (i = 1, 2, \ldots, d) \qquad (1)$$

where,

- P_i: Population size of species i.
- r_i: Intrinsic growth rate of species i, i.e., the rate of reproduction in the absence of density regulation ($r_i \geq 0$).
- K_i: Carrying capacity of species i when the other species are absent ($K_i \geq 0$).
- a_{ii}: Intraspecies competition, i.e., competition for resources between members of the same population ($a_{ii} = 1$).
- a_{ij}: Interspecies competition, i.e., competition between two different species ($a_{ij} \geq 0$).

Here, time t is considered continuous and dP_i/dt is the derivative. For each species i, the number of offspring per parent increases linearly with the size of the current population P_i, and it corresponds to the intrinsic growth rate r_i.

In the Lotka-Volterra equation, it is assumed that multiple (i.e., d) species are competing for some common resources. For example, Figure 2 (a) shows the interaction between wild animals in the jungle. [3] Assume that these species share some of the resources (e.g., fruits). The number of individuals using the resources of species i can be described as: $a_{i1}P_1 + \cdots a_{ij}P_j \cdots P_i + \cdots a_{id}P_d = \sum_{j=1}^{d} a_{ij}P_j$. Here, $a_{ij} (i \neq j)$ is called "interspecies competition", which measures the effect an individual of species j has on an individual of species i. [4]

4. PROPOSED MODEL

In this section, we present our proposed model, namely, ECOWEB. Consider that we have a collection of activity volumes X of d keywords, with duration n. That is, we have $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_i, \cdots, \boldsymbol{x}_d\}$, where $\boldsymbol{x}_i$ is a sequence of keyword i, (i.e., $\boldsymbol{x}_i = \{x_i(t)\}_{t=1}^{n}$). Given a set of co-evolving time series X, our goal is to (a) capture the evolutions of X, (b) find the hidden relationship between each sequence, and (c) forecast future dynamics.

So, how can we describe the evolutions of multiple keywords, and spot interactions between two different keywords? What exactly is the relationship, say, between Xbox and Wii, or Facebook and LinkedIn? Are there any differences or similarities? Do they compete with each other, like wild animals?

Ecosystem on the web - intuition behind our model. So, what is an ecosystem on the web? Can we find similar phenomena in virtual communities? If so, what kind of species live on the web? How does the population size of each species evolve over time? — Our answers are that: (a) there are an infinite number of "virtual species" living on the web (as in a "jungle"), and (b) they evolve naturally over time by interacting with other species.

Figure 2 (b) shows an ecosystem on the web. Similar to the biological community, which consists of multiple species (e.g., monkeys and macaws, as shown in Figure 2 (a)), there is a community of virtual species on the web (e.g., Xbox and PlayStation, as shown in Figure 2 (b)).

Here, we provide two important analogies with respect to the ecosystem on the web.

- **Keyword/activity (i.e., species)**: No keyword can survive on the web if no one is paying attention to that topic. It behaves like a living organism. The relationship between keywords and users (e.g., between Wii and kids) is similar to the relationship between species and food resources (e.g., between squirrel monkeys and fruits or between capybaras and grass). No species can survive without resources.
- **User resources (i.e., food resources)**: Similar to an ecological system, there are a finite number of users and their resources on the web. The user resources could be anything, such as user interest/attention, or an amount of the time and money they spend. Users cannot use their time/money for multiple purposes simultaneously. [5] As shown in Figure 2 (b), there are some groups of users, such as kids, teenagers and adults. For example, kids love video games, e.g., Xbox, PlayStation and Wii, while most adults prefer Android.

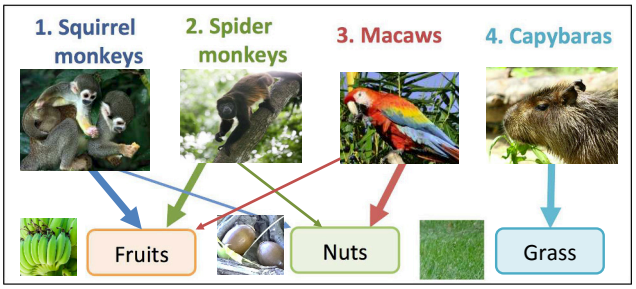
Although important, the above analogies are not immediately applicable to our setting. We need a few more concepts. Specifically, we want to describe the following three properties:

- (**G1**): Non-linear evolution of keywords/activities
- (**G2**): Interaction coefficients between keywords
- (**G3**): Seasonality of user activities

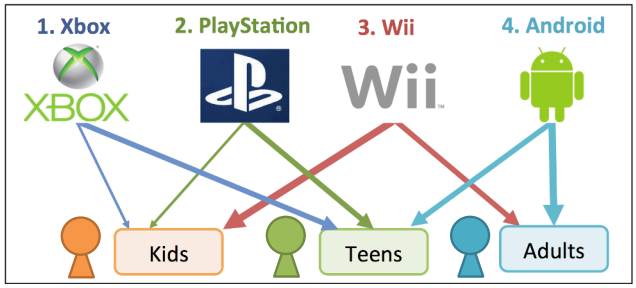
[3] Image courtesy of xura, criminalatt, David Castillo Dominici, happykanppy at FreeDigitalPhotos.net.

[4] There are several variations of the Lotka-Volterra model, e.g., the predator-prey/parasitism model. However, in this paper, we only focus on the simplest case where $a_{ij} \geq 0$ $(i \neq j)$ for all species i and j (i.e., neutralism/amensalism/competition).

[5] For example, given N users, there are $N \times 24$ hours/resources per day, or fewer, depending on the keyword and the demographic group it appeals to.



(a) Ecosystem in the jungle



(b) Ecosystem on the web

Figure 2: Illustration of jungle vs. web: (a) Ecosystem in the jungle (e.g., Amazon rainforest): squirrel monkeys partially share their food with spider monkeys and macaws, while capybaras are isolated (i.e., they have no competitors here); (b) Ecosystem on the web (e.g., game industry): the main targets of Xbox, PlayStation and Wii are kids and teenagers, while most adults are interested in Android games rather than Xbox and PlayStation.

Table 2: Analogy: Jungle vs. web.

Jungle	Web
Species (e.g., squirrel monkeys)	Keywords/activities (e.g., Wii/Xbox)
Food resources (e.g., fruits)	User resources (e.g., kids/adults)
Population	Popularity
Climate/season	Annual events (e.g., Christmas)

In a real ecosystem, the population of each species varies continuously over time. It depends on the reproduction rate per generation and the number of offspring produced in a lifetime by each individual. The same thing happens on the web: the popularity size of each keyword evolves over time. The popularity size corresponds to the aggregated volume of each user interest/attention. If a new product (say, Android) is attractive, the users would spend more time on it, or recommend it to their friends. Similarly, their friends would influence other users, and eventually, this would lead to an exponential growth in popularity size. To handle (**G1**), we propose using a non-linear difference equation.

For (**G2**), we assume that there are latent interactions between two different keywords. For example, in Figure 1 (a), the sequences of Xbox (i.e., $\boldsymbol{x}_1$) and PlayStation (i.e., $\boldsymbol{x}_2$) behave in opposite ways: When the volume of PlayStation increases, the volume of Xbox decreases considerably (please see Figure 1 (a) from 2007 to 2010). That is, there must be competition/interaction between these two keywords.

We should also note that online activities have certain annual patterns, i.e., seasonality (**G3**). For example, in Figure 1 (a), all the sequences have a huge spike at Christmas. This is because the users modulate their activities based on a yearly cycle. Similar behavior is observed with wild animals in that their activities may depend on climate and season.

Table 2 describes our basic analogy, namely, the jungle ecosystem applied to the web. We conjecture that users of the web behave in the same way as wild animals in the jungle in that they interact and compete with each other for resources.

Next, we introduce our model in steps of increasing complexity.

4.1 EcoWeb-individual (G1)

We begin with the simplest case, where we have a single sequence/keyword, i.e., there is no interspecies interaction/competition.

Let K be the quantity of available user resources that might be used (i.e., paid attention) as regards this keyword, and p represent the quantity of user resources that have already been used as regards this keyword at time tick $t = 0$ (i.e., initial condition).

In our model, we assume that the keyword/activity follows some very simple local rules:

Table 3: Symbols and definitions.

Symbol	Definition
d	Number of unique keywords/activities (i.e., species)
n	Duration of sequences
X	d co-evolving time sequences (i.e., $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d\}$)
$\boldsymbol{x}_i$	Sequence of keyword i (i.e., $\boldsymbol{x}_i = \{x_i(1), \ldots, x_i(n)\}$)
$x_i(t)$	Volume of keyword i at time tick t
$P_i(t)$	Popularity size of keyword i at time tick t
$C_i(t)$	Estimated volume of keyword i at time tick t
$\boldsymbol{p}$	Initial popularity size i.e., $\{p_i\}_{i=1}^{d}$
$\boldsymbol{r}$	Growth rate i.e., $\{r_i\}_{i=1}^{d}$
$\boldsymbol{K}$	Carrying capacity i.e., $\{K_i\}_{i=1}^{d}$
$\mathbf{A}$	Interaction matrix ($d \times d$) i.e., $\mathbf{A} = \{a_{ij}\}_{i,j=1}^{d,d}$
n_p	Period (i.e., 52 weeks)
k	Number of hidden seasonalities
$\mathbf{E}$	Seasonal activities ($d \times n$) i.e., $\mathbf{E} = \{e_i(t)\}_{i,t=1}^{d,n}$
$\mathbf{W}$	Participation matrix ($d \times k$) i.e., $\mathbf{W} = \{w_{ij}\}_{i,j=1}^{d,k}$
$\mathbf{B}$	Seasonality matrix ($k \times n_p$) i.e., $\mathbf{B} = \{b_j(\tau)\}_{j,\tau=1}^{k,n_p}$

- It maintains its current popularity size (i.e., user attention) unless there is intra/interspecies competition.
- For each time tick t, it obtains new user resources, and the popularity size increases by a constant percentage r.

Let $P(t)$ be the popularity size of the keyword at time tick t. The evolution of a single keyword is described by the following difference equation:

$$P(t+1) = P(t)\left[1 + r\left(1 - \frac{P(t)}{K}\right)\right], \tag{2}$$

with the initial condition $P(0) = p$, where,

- $P(t)$: Popularity size of the keyword at time tick t, i.e., the aggregated volume of user attention to the keyword. [6]
- p: Initial condition, i.e., popularity size at time tick $t = 0$.
- r: growth rate, i.e., the attractiveness/strength (i.e., impact) of the keyword.
- K: Carrying capacity, i.e., maximum popularity size of the keyword (= available user resources).

Note that the term: $\left[1 + r\left(1 - \frac{P(t)}{K}\right)\right]$ corresponds to the contribution of the current popularity to the next popularity growth, where $\left(1 - \frac{P(t)}{K}\right)$ is the percentage of available user resources for the keyword at time tick t. If the keyword runs out of user re-

[6] In this paper, we assume that $P(t)$ is the popularity density of a keyword, i.e., $0 \leq P(t) \leq 1$, however, our equations can also handle other settings, such as the actual numbers of keyword appearances.

sources (i.e., $P(t) = K$), the expanding popularity will hit a constraint. Also note that Equation 2 is a discrete version of the Lotka-Volterra differential equation, (Equation 1), when it has a single species ($d = 1$).

4.2 EcoWeb-interaction (G2)

We now move on to the next step, namely, spotting an interaction between co-evolving keywords (**G2**). In general, some keywords are competing for some common user resources. Obviously, there is some kind of competition between video game consoles, such as Xbox and PlayStation. Most users choose one of the consoles based on their preferences (e.g., price and available game titles).

MODEL 1 (ECOWEB-INTERACTION). *Let $P_i(t)$ be the popularity size of keyword i at time tick t. Our interaction model is governed by the following equations,*

$$P_i(t+1) = P_i(t)\left[1 + r_i\left(1 - \frac{\sum_{j=1}^{d} a_{ij}P_j(t)}{K_i}\right)\right], \quad (i = 1, \cdots, d), \quad (3)$$

where, $r_i > 0, K_i > 0, a_{ii} = 1, a_{ij} \geq 0$, and $P_i(0) = p_i$.

In Model 1, it is assumed that competing keywords share some of the same user resources. At time tick t, the percentage of potential (i.e., available) user resources for keyword i [7] can be described as,

$$\left(1 - \frac{\sum_{j=1}^{d} a_{ij}P_j(t)}{K_i}\right), \quad (4)$$

where, a_{ij} is the interaction coefficient, which describes the effect rate of keyword j on keyword i.

Please note that if there is no interspecies interaction/competition, (that is, $a_{ij} = 0$ $(i \neq j)$), this model is identical to Equation 2 (i.e., "neutralism"). In contrast, if $a_{ij} = a_{ji} = 1$ for keywords i, j, this means that two keywords i, j compete with each other, by sharing exactly the same user resource group. If $a_{ij} = 1, a_{ji} = 0$, the model describes an asymmetric competitive interaction, which is known as "amensalism". In this case, keyword i is strongly affected by keyword j, while keyword j is almost unaffected by keyword i.

EXAMPLE 1. *Figure 1 (b) shows the interaction between $d = 4$ keywords, where we have an interaction matrix:*

$$\mathbf{A} = \begin{bmatrix} 1 & 0.5 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Here, Xbox $\boldsymbol{x}_1$ is affected by PlayStation $\boldsymbol{x}_2$, (i.e., $a_{12} = 0.5$) and Wii $\boldsymbol{x}_3$, (i.e., $a_{13} = 0.1$), while PlayStation and Wii are affected by Android $\boldsymbol{x}_4$, (i.e., $a_{24} = 0.1$, $a_{34} = 0.3$). Xbox and Android do not interact directly with each other (i.e., $a_{14} = a_{41} = 0$).

4.3 With seasonality (G3)

Thus far, we have discussed how to describe the long-range dynamics of d co-evolving sequences. Although important, it is not sufficient to capture the real keyword evolutions. Each keyword (e.g., Xbox and Amazon) always has a certain number of users (i.e., popularity), however, the users change their behavior dynamically, according to various seasonal events (e.g., Amazon.com has many visitors on Black Friday). We can observe similar behavior in an ecological system, where activities depend on season and climate: for example, most monkeys are active during warm and sunny days, while they sleep at night. Most importantly, these activities are often correlated with other related species/keywords, e.g., the sales of most retailers including Amazon peak on Black Friday. That is, there must be some groups of "hidden" seasonal activities, (e.g., seasonal retail sales).

So how can we reflect this phenomenon in our equation? We want a powerful yet simple model that can capture seasonal patterns (**G3**) in real co-evolving sequences, as well as long-range non-linear evolutions. We provide an answer below.

MODEL 2 (ECOWEB-FULL). *Let $C_i(t)$ be the estimated volume of keyword i at time tick t. Our full model captures seasonal user activities with the following equations:*

$$C_i(t) = P_i(t)\,[1 + e_i(t)] \qquad (i = 1, \cdots, d), \quad (5)$$

where $e_i(t)$ describes seasonal activities of keyword i over time.

The estimated volume $C_i(t)$ describes how many times keyword i appears at time tick t, and depends on the latent popularity size $P_i(t)$ and seasonal activities $\mathbf{E} = \{e_i(t)\}_{i,t=1}^{d,n}$. Each element in $\mathbf{E}$ describes the relative value of the potential popularity size versus the actual keyword volume, and it corresponds to seasonal events, holidays, etc. If there is no seasonal pattern in keyword i at time t, (i.e., $e_i(t) = 0$), the keyword volume is equal to the popularity size (i.e., $C_i(t) = P_i(t)$).

Compact representation of seasonality. With respect to seasonal activities $\mathbf{E}$, we need $(d \times n)$ parameters to describe the entire dataset X, and this is not feasible in our case. We want to avoid redundancy, and so it should be compressed into a small set of parameters. We are interested in capturing (a) yearly periodic patterns (e.g., Black Friday) as well as (b) hidden groups of seasonal activities (e.g., retail sales). So how can we deal with this issue? We propose *decomposing* $\mathbf{E}$, to achieve much better modeling. Specifically, we decompose $\mathbf{E}$ into two matrices, namely, seasonality matrix $\mathbf{B}$ of size $(k \times n_p)$ and participation matrix $\mathbf{W}$ of size $(d \times k)$. Here, $\mathbf{B}$ represents a set of k seasonal components of period n_p, while $\mathbf{W}$ describes the participation weight of each sequence for each seasonal component. Consequently, the seasonal activities $\mathbf{E} = \{e_i(t)\}_{i,t=1}^{d,n}$ can be described as the following function:

$$e_i(t) \simeq f(i, t|\mathbf{W}, \mathbf{B}) = \sum_{j=1}^{k} w_{ij} b_j(\tau) \quad (\tau = [t \mod n_p]) \quad (6)$$

where,

- n_p: Period (say, 52 weeks in one year).
- k: Number of latent seasonal components.
- $\mathbf{W} = \{w_{ij}\}_{i,j=1}^{d,k}$: Participation matrix, i.e., participation weight of keyword i for the j-th seasonal component.
- $\mathbf{B} = \{b_j(\tau)\}_{j,\tau=1}^{k,n_p}$: Seasonality matrix, i.e., temporal activity at time tick τ for the j-th seasonal component.

Note that the number of components k should be estimated automatically, and we will describe this in the next section.

EcoWeb: full model parameter set. Figure 3 shows our modeling framework. Given a set of d co-evolving sequences X, our goal is to find important patterns with respect to three aspects: (**G1**) individual properties, i.e., initial popularity size: $\boldsymbol{p} = \{p_i\}_{i=1}^{d}$, growth rate: $\boldsymbol{r} = \{r_i\}_{i=1}^{d}$, carrying capacity: $\boldsymbol{K} = \{K_i\}_{i=1}^{d}$; (**G2**) interaction matrix: $\mathbf{A} = \{a_{ij}\}_{i,j=1}^{d,d}$; (**G3**) a set of k seasonal activities, which consists of participation matrix $\mathbf{W}$ and seasonality matrix $\mathbf{B}$.

[7] We can also say: the amount of available user resources for keyword i with a limited size of maximum popularity size K_i is: $K_i - \sum_{j=1}^{d} a_{ij}P_j(t)$.

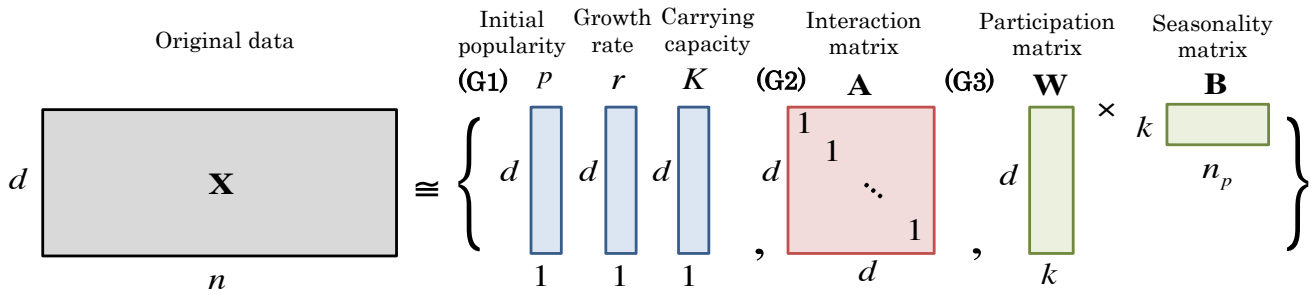


Figure 3: Illustration of EcoWeb structure. Given a set of d sequences X of length n, we extract (G1) individual properties, i.e., initial popularity size: $\boldsymbol{p}$, growth rate: $\boldsymbol{r}$, carrying capacity: $\boldsymbol{K}$, (G2) interaction matrix: $\mathbf{A}$, as well as (G3) a set of k seasonal components, i.e., participation matrix: $\mathbf{W}$ and seasonality matrix: $\mathbf{B}$.

DEFINITION 1 (COMPLETE SET OF ECOWEB). *Let $\mathcal{S}$ be a complete set of parameters (namely, $\mathcal{S} = \{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}, \mathbf{W}, \mathbf{B}\}$) that describe the individual/interactive/seasonal patterns of X.*

5. OPTIMIZATION ALGORITHM

In the previous section, we have seen how we can describe the evolutions of multiple sequences with respect to three properties that we observed with real time series data. Now, we want to figure out *how to estimate* an optimal parameter set. Specifically, we need to answer the following two questions: (1) How can we find an optimal set of seasonal components, (i.e., $\mathbf{W}, \mathbf{B}$)? (2) How can we efficiently and effectively estimate full parameter set $\mathcal{S}$ that best captures the important patterns in X? Each question is dealt with in the following subsections.

5.1 Automatic seasonal component analysis

Let us begin with the first question, namely, how to find an appropriate set of seasonal components $\mathbf{W}$ and $\mathbf{B}$. Here, we divide the question into two parts:

- *Seasonal component detection:* Find good seasonal matrices $\mathbf{W}$ and $\mathbf{B}$, when given a fixed number of components k.
- *Automatic component analysis:* Search for the best number of components among all possible k values ($k = 1, 2, \ldots$).

Seasonal component detection. Assume that we are given X, and *also* a set of base model parameters for our model, i.e., $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}\}$. According to Models 1 and 2, each element in $\mathbf{E}$ can be simply computed by:

$$e_i(t) = \frac{x_i(t) - P_i(t)}{P_i(t)} \quad (i = 1, \ldots, d; t = 1, \ldots, n). \tag{7}$$

After computing $\mathbf{E}$ of size $(d \times n)$, our next step is to decompose it into an optimal set consisting of $\mathbf{W}$ and $\mathbf{B}$.

The most straightforward solution would be to assume that there is a set of $k = d$ different temporal activities of length n for all d sequences. However, this solution requires $(d \times n)$ parameters to capture the entire sequence set X. Also, it gives a very poor representation, and cannot capture seasonal dynamics among multiple keywords.

We thus propose an efficient and effective algorithm that can find an optimal set of k distinct seasonal patterns among all sequences X. Figure 4 illustrates our approach. Given a set of seasonal activities $\mathbf{E}$ of size $(d \times n)$, our algorithm splits each sequence into non-overlapping subsequences of length n_p, and constructs a matrix $\hat{\mathbf{E}}$ of size $([d \times \lceil n/n_p \rceil] \times n_p)$. It then finds a set of k components from $\hat{\mathbf{E}}$ and creates a seasonality matrix $\mathbf{B}$ of size $(k \times n_p)$. After finding $\mathbf{B}$, it estimates a participation matrix $\mathbf{W}$ of size $(d \times k)$ so that we

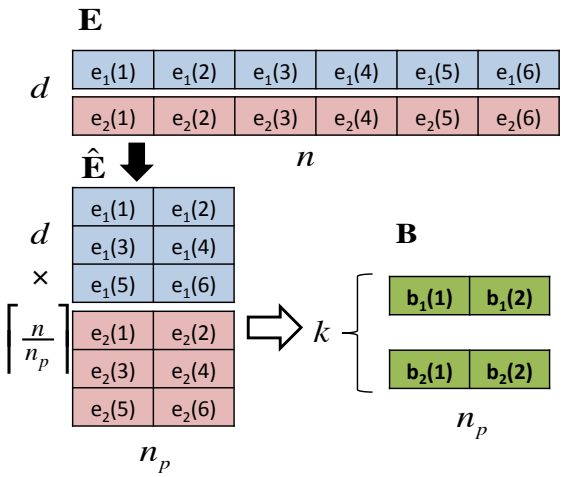


Figure 4: Illustration of seasonal component analysis (for $n_p = 2$). Given a set of seasonal activities $\mathbf{E}$ of size $(d \times n)$, it creates a matrix $\hat{\mathbf{E}}$ of $d \times \lceil n/n_p \rceil$ disjoint windows. It then finds their k major components, i.e., $\mathbf{B}$ $(k \times n_p)$.

can reconstruct the original matrix $\mathbf{E}$ as described in Equation 6, (i.e., $\mathbf{E} \simeq f(\mathbf{W}, \mathbf{B})$).

There is an important issue here: what is the best way of finding typical seasonal components $\mathbf{B}$ in $\hat{\mathbf{E}}$? The first idea would be to perform principal component analysis (PCA) [19] as employed in [23, 41]. However, PCA has pitfalls: it uses an orthogonal transformation. Given an input matrix $\hat{\mathbf{E}}$, it tries to find the best component that goes through $\hat{\mathbf{E}}$; and then the second best component (orthogonal to the first), and so on, until it obtains k components. That is, it cannot capture "real" activities. We thus propose employing independent component analysis (ICA) [16], which is also known as blind source separation (BSS). Unlike PCA, it finds a set of k components that are both statistically independent and non-Gaussian. That is, it seeks components that are the most independent from each other.

Automatic component analysis. As regards seasonal component analysis, we need to determine the number of components, k. We thus provide an intuitive coding scheme, which enables our algorithm to find appropriate sizes for $\mathbf{W}$ and $\mathbf{B}$, *automatically*. Our coding scheme is based on the minimum description length (MDL) principle. In short, it follows the assumption that the more we can compress the data, the more we can learn about its underlying patterns.

The description complexity of model parameter set $\mathcal{S}$ consists of the following terms: The number of dimensions d and time ticks

n require $\log^*(d) + \log^*(n)$ bits. [8] The initial popularity size, growth rate, carrying capacity i.e., $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}\}$ and the interaction matrix $\mathbf{A}$ require $d \times 3$ and $(d \times d - d)$ parameters, respectively, i.e., $Cost_M(\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}) + Cost_M(\mathbf{A}) = c_F \cdot d(3 + d - 1)$, where c_F is the floating point cost[9]. Similarly, the model description cost of k seasonal components is $Cost_M(k, \mathbf{W}, \mathbf{B}) = \log^*(k) + \log^*(n_p) + c_F(dk + kn_p)$.

Once we have decided the full parameter set $\mathcal{S}$, we can encode the original data X using Huffman coding [4], i.e., a number of bits is assigned to each value in X, which is the logarithm of the inverse of the probability (i.e., the negative log-likelihood) of the value. The encoding cost of X given $\mathcal{S}$ is computed by:

$$Cost_C(X|\mathcal{S}) = \sum_{i,t=1}^{d,n} \log_2 p^{-1}_{Gauss(\mu,\sigma^2)}(x_i(t) - C_i(t)),$$

where, $x_i(t)$ and $C_i(t)$ are the original and estimated volumes of keyword i at time tick t (i.e., Model 2). Also, μ and σ^2 are the mean and variance of the distance between the original and estimated values. [10]

The total code length for X with respect to a given parameter set $\mathcal{S}$ can be described as follows:

$$\begin{aligned} Cost_T(X; \mathcal{S}) &= \log^*(d) + \log^*(n) + Cost_M(\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}) \\ &\quad + Cost_M(\mathbf{A}) + Cost_M(k, \mathbf{W}, \mathbf{B}) + Cost_C(X|\mathcal{S}) \end{aligned} \quad (8)$$

Consequently, our algorithm automatically determines the optimal number of seasonal components k_{opt} according to the above function, i.e., $k_{opt} = \arg\min_k Cost_T(X; \mathcal{S})$.

5.2 Multi-step fitting algorithm

We have described how to find seasonal activities $\{\mathbf{W}, \mathbf{B}\}$ in X, when a set of base parameters $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}\}$ were given. Next, we tackle the most important and challenging question, namely, how to efficiently and effectively estimate a full parameter set $\mathcal{S}$. We would like to estimate (**G1**) individual parameters $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}\}$, (**G2**) interaction matrix $\mathbf{A}$ and (**G3**) seasonal activities $\{\mathbf{W}, \mathbf{B}\}$, *simultaneously*.

So how do we go about finding the optimal solution $\mathcal{S}$? The most straightforward approach would be simply to estimate all the parameters in $\mathcal{S}$ simultaneously. This approach requires us to estimate $(3d + (d^2 - d) + k(d + n_p))$ parameters for each iteration. It also requires us to compare all possible solutions for a different number k $(1 \leq k \leq d)$. This method is both extremely expensive and ineffective in that it is difficult to optimize all the parameters directly.

We thus propose an efficient algorithm, StepFit, which divides a parameter set $\mathcal{S}$ into two subsets $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}\}$, and $\{\mathbf{W}, \mathbf{B}\}$, and estimates the parameters alternately (see Algorithm 1). The first step assumes that there is no seasonality, i.e., $k = 0$, and estimates the base parameters. In the next step, the base parameters are fixed, and $\mathbf{B}$ and $\mathbf{W}$ are computed using automatic seasonal component analysis as described in subsection 5.1. Here, we use the *Levenberg-Marquardt (LM)* [29] algorithm to minimize the cost function (i.e., Equation 8). The algorithm continues to estimate the parameters until convergence.

However, StepFit still needs to update the parameters of interaction matrix $\mathbf{A}$ of size $(d \times d)$, as well as all d individual parameters i.e., $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}\}$ for every iteration. In other words, StepFit tries to find the best solution $\mathcal{S}$ among all possible combinations of d keywords. One subtle but important issue is that, compared with the linear model, it is difficult to find the optimal parameter set for non-linear equations. So, how can we efficiently and effectively estimate all the parameters $\mathcal{S}$? We want to find the optimal solution in terms of both the individual and interactive parameters.

[8] Here, $\log^*$ is the universal code length for integers.

[9] We digitize the floating number into $c_F = 8$ bits.

[10] Here, μ, σ^2 need $2c_F$ bits, but we can eliminate them because they are constant values and independent of our modeling.

Algorithm 1 StepFit (X)

Input: Co-evolving sequences X $(d \times n)$
Output: Full parameter set, i.e., $\mathcal{S} = \{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}, \mathbf{W}, \mathbf{B}\}$
$\mathbf{W} = \mathbf{B} = 0$; /* Initialize seasonal activities ($k = 0$) */
while improving the parameters **do**
  /* (I) Base parameter fitting (**G1**), (**G2**) */
  $\{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}\} = \arg\min_{\boldsymbol{p}', \boldsymbol{r}', \boldsymbol{K}', \mathbf{A}'} Cost_T(X; \boldsymbol{p}', \boldsymbol{r}', \boldsymbol{K}', \mathbf{A}', \mathbf{W}, \mathbf{B})$;
  /* (II) Seasonal parameter fitting (**G3**) */
  $\{\mathbf{W}, \mathbf{B}\} = \arg\min_{\mathbf{W}', \mathbf{B}'} Cost_T(X; \boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}, \mathbf{W}', \mathbf{B}')$;
end while
return $\mathcal{S} = \{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}, \mathbf{W}, \mathbf{B}\}$;

Algorithm 2 EcoWeb-Fit (X)

Input: Co-evolving sequences X $(d \times n)$
Output: Full parameter set, i.e., $\mathcal{S} = \{\boldsymbol{p}, \boldsymbol{r}, \boldsymbol{K}, \mathbf{A}, \mathbf{W}, \mathbf{B}\}$
$\mathbf{A} = \mathbf{I}_d$; /* Initialize $\mathbf{A}$, i.e., identity matrix of size $(d \times d)$ */
/* (I) Single fitting (**G1**), (**G3**) */
for $i = 1 : d$ **do**
  /* Estimate individual parameters of keyword i */
  $\mathcal{S}_i = \text{StepFit}(\boldsymbol{x}_i)$;
end for
/* (II) Pair fitting (**G1**), (**G2**), (**G3**) */
while improving the parameters **do**
  /* Find the most unfitted sequence $\boldsymbol{x}_i$ */
  $i = \arg\max_{1 \leq i' \leq d} Cost_T(\boldsymbol{x}_{i'}; \mathcal{S})$;
  /* Estimate parameters of pair (i, j) */
  for $j = 1 : d$ **do**
    $\mathcal{S}'_{ij} = \text{StepFit}(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
  end for
  /* Find the most affecting sequence $\boldsymbol{x}_j$ on $\boldsymbol{x}_i$ */
  $j = \arg\min_{j'} Cost_T(\boldsymbol{x}_i, \boldsymbol{x}_{j'}; \mathcal{S}'_{ij'})$;
  /* Update best pair parameters */
  Update $\mathcal{S}_{ij} = \mathcal{S}'_{ij}$;
end while
/* (III) Full fitting (**G1**), (**G2**), (**G3**) */
$\mathcal{S} = \arg\min_{\mathcal{S}'} Cost_T(X; \mathcal{S}')$;
return $\mathcal{S}$;

Algorithm - EcoWeb-Fit. We thus extend StepFit and introduce a partitioning approach for analyzing a large number of keywords, which yields a dramatic reduction in the computation cost. Algorithm 2 describes the overall procedure. The idea is that instead of fitting all the parameters of X simultaneously, it first assumes that there is no interspecies competition, (that is, it sets $\mathbf{A} = \mathbf{I}_d$, i.e., $a_{ij} = 0$ $(i \neq j)$), and estimates a model parameter set $\mathcal{S}_i = \{p_i, r_i, K_i, w_{ii}, b_i\}$ for each individual sequence x_i $(i = 1, \ldots, d)$, separately using StepFit. In the next step, it assumes that there is competition between two keywords i and j. Specifically, for each iteration, the algorithm tries to find the best pair (x_i, x_j) so that it minimizes the cost function i.e., $Cost_T(x_i, x_j|\mathcal{S}_{ij})$. It continues pair-fitting until convergence. Finally, the algorithm optimizes the full parameter set $\mathcal{S}$ using the entire sequence set X.

6. EXPERIMENTS

In this section we demonstrate the effectiveness of EcoWeb with real data. The experiments were designed to answer the following questions:

Q1 *Effectiveness*: How successful is our method in spotting meaningful patterns in given input sequences?
Q2 *Accuracy*: How well does our method match the data?
Q3 *Scalability*: How does our method scale in terms of computational time?

6.1 Q1: Effectiveness

We now demonstrate the power of our model in terms of capturing important and informative patterns of online activities. We performed experiments on sequence sets of keywords/activities from five areas on *GoogleTrend*. Note that the dataset is scaled so that each sequence has a peak volume of 1.0.

#1. Video games. The result for this area has already been presented in Figure 1 of section 1. Our method captures long-range evolving dynamics between three game consoles (i.e., Xbox, PlayStation and Wii), and the appearance of Android, as well as important annual events, e.g., Black Friday and Christmas.

#2. Programming languages. Figure 5 (a) shows our discoveries on "C", "R" and "MATLAB".

- *Long-range evolution and interaction*: Figure 5 (a-i) shows the fitting results (lines) and the original sequences (circles). Again, our method fits the real data very well. Moreover, it captures the interaction: Figure 5 (a-ii) shows the interaction network, indicating competition between the "C" programming language and the "R" statistical system, while "MATLAB" seems not to be involved. Indeed, the time sequences show that the interest in "R" has increased constantly since 2004, at the expense of "C" - possibly, due to an emphasis on big data analytics.
- *Seasonal activities*: Figure 5 (a-iii) shows the full parameter set of ECOWEB (darker gray corresponds to a higher value). With respect to the seasonal activities ($\mathbf{W}$ and $\mathbf{B}$, shown at the bottom), our method discovered, to our surprise, that there is a strong correlation with the academic calendar. For example, during the spring, summer and winter breaks, the attention paid to each keyword (especially, MATLAB) decreases significantly: Apparently, most of those issuing queries, are students (as opposed to professional programmers), and they enjoy their vacation, instead of coding.

#3. Social media. Figure 5 (b) shows the fitting result for the social media activities: "Tumblr", "Facebook" and "LinkedIn".

- *Long-range evolution and interaction*: Most social media sites have been attracting searches only recently (say, after 2008 - $\boldsymbol{p} \approx 0$, see (b-iii)). For example, Tumblr is a blog platform that was founded in 2007, and it has been attracting huge numbers of users (i.e., the growth rate r of Tumblr is steep). Figure 5 (b-ii) shows that there is competition between Tumblr and Facebook, but there is no competitor for LinkedIn.
- *Seasonal activities*: The bottom figure (b-iii) shows that there is an opposite seasonality as regards social media: during Christmas and New Year's day, the number of Facebook users increases, while the number using LinkedIn drops significantly. This is probably because the former is used for private purposes, while the latter is a business-oriented SNS.

#4. Apparel companies. Figure 5 (c) shows the result for four heavily-searched fashion-related companies: *Nordstrom* (an upscale department store); *Kohl's* (a discount retailer) *JCPenney* (a mid-range department store, with CEO problems) and *Forever21* (which focuses on young girls, and recently added a line of bigger sizes).

- *Long-range evolution and interaction*: Our method captures the competition between Kohl's and Nordstrom, and between JCPenney and Forever21. Arguably due to the recession (2008 onwards), shoppers moved away from upscale Nordstrom and towards discount-priced Kohl's (which also engaged in some brilliant marketing: offering discounts for seniors, and issuing its own credit card to encourage increased customer loyalty). Similarly, Forever21 grew significantly, probably due to their decision to add a line of bigger sizes; thus, it apparently lured attention away from JCPenney, which was damaged by poor decisions made by the new CEO, Ron Johnson, who was eventually fired.
- *Seasonal activities*: All keywords have clear patterns of annual activity. There is a huge spike on Black Friday: the biggest sale event of the year. There is also a small spike in August, which is the *"back to school"* period.

#5. Retail companies. Figure 5 (d) shows the results for the top six retail companies.

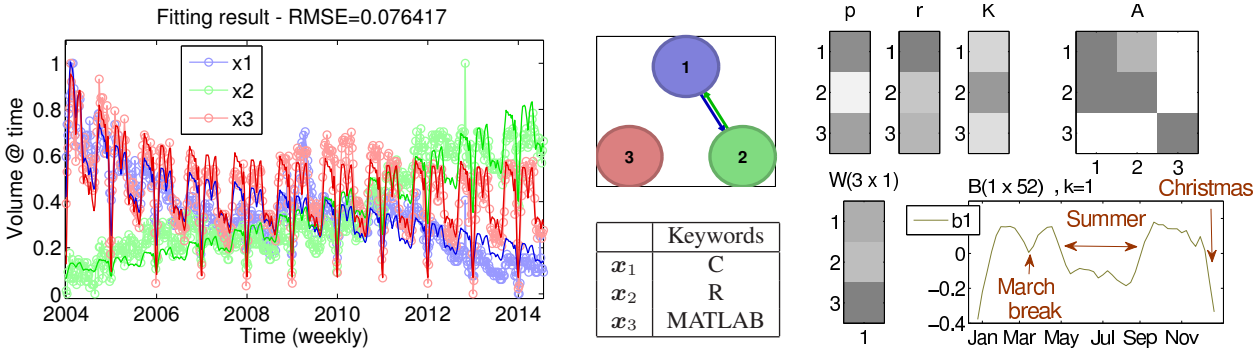
- *Long-range evolution and interaction*: Clearly, every keyword is steadily increasing, with Best Buy being the only exception (arguably suffering, due to the success of online retailers). There is no clear interaction, except between Home Depot and Lowes, which are home improvement and appliance retailers, or, *do it yourself (DIY)* stores. In Figure 5 (d-i), our method captures both the individual and interaction dynamics of retail activities.
- *Seasonal activities*: As described in (d-iii), our method *automatically* discovered *two* hidden seasonal patterns (i.e., $k = 2$) in retail companies. The first component (b_1, in light brown) corresponds to Home Depot and Lowes, and the second component (b_2, in purple) corresponds to Amazon, Walmart, Best Buy and Costco. In addition to a huge clear spike on Black Friday in both components, there are multiple spikes in Home Depot and Lowes, corresponding to the national holidays in summer (see b_1): Memorial Day (last Monday in May), Independence Day (4th of July) and Labor Day (first Monday in September).

6.2 Q2: Model accuracy

Next, we discuss the quality of our approach in terms of fitting accuracy. We compared ECOWEB-FIT with the standard *LV* model. To evaluate the effect of our efficient fitting algorithms, we also compared them with a special version of our method: ECOWEB-Plain, which uses only STEPFIT to estimate model parameters. Figure 6 shows the root mean square error (RMSE) between the original and estimated volumes for five sequence sets (#1-#5). A lower value indicates a better fitting accuracy. As shown in the figure, our approach achieved high fitting accuracy. Since the *LV* model cannot capture seasonal patterns, it was strongly affected by multiple spikes and failed to capture co-evolving dynamics. ECOWEB-Plain has the ability to capture periodic patterns, but it was not completely successful in capturing complicated dynamics and interactions between multiple sequences.

6.3 Q3: Scalability

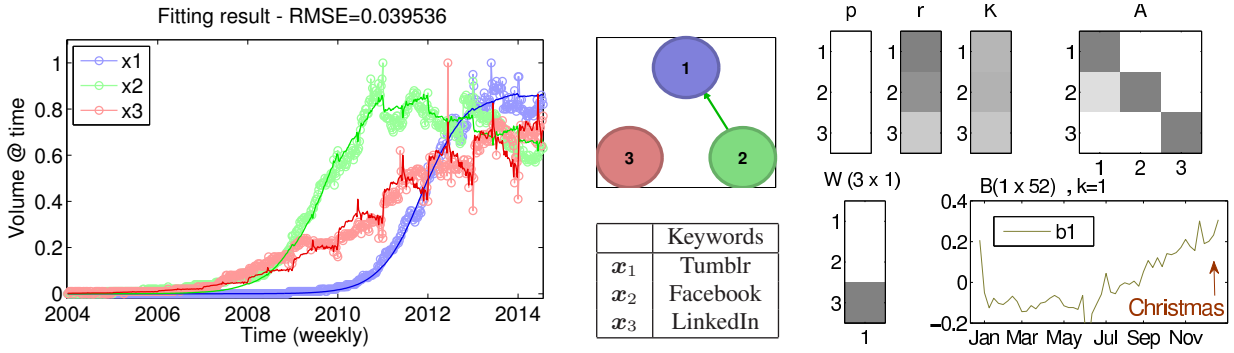
We also evaluated the scalability of our method. Figure 7 shows the average computational cost of ECOWEB-FIT. We varied the dataset size from five to ten years. Our method achieved a large reduction in terms of computation time as well as fitting error for every sequence set. We observed that ECOWEB-FIT was linear with respect to data length n, and was up to 20 times faster than ECOWEB-Plain. ECOWEB-FIT was also up to 7 times faster than the *LV* model, even though our method has the ability to capture seasonal dynamics.



	Keywords
x_1	C
x_2	R
x_3	MATLAB

(a-i) Fitting result vs. original data (a-ii) Interaction (a-iii) Full parameter set

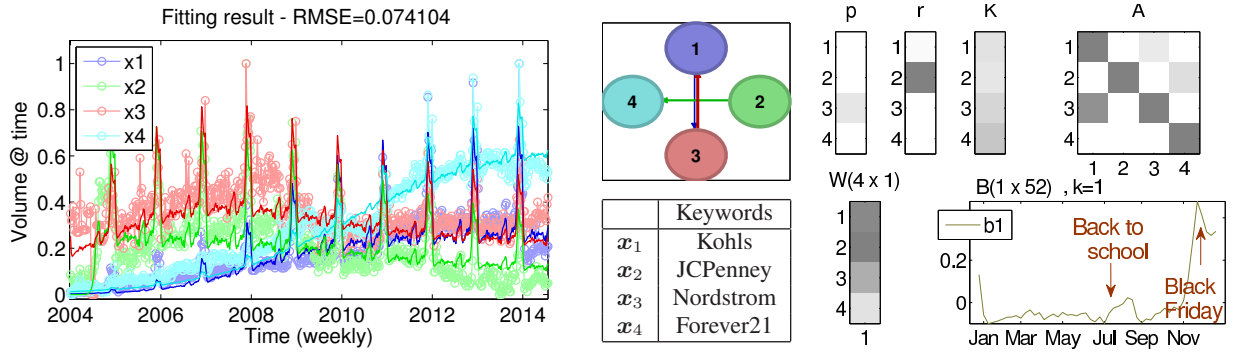
(a) Fitting result for the programming language activities.



	Keywords
x_1	Tumblr
x_2	Facebook
x_3	LinkedIn

(b-i) Fitting result vs. original data (b-ii) Interaction (b-iii) Full parameter set

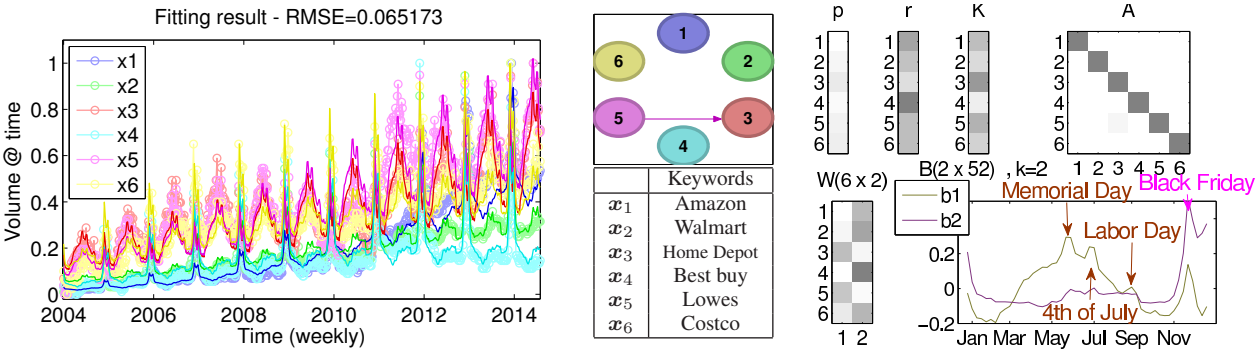
(b) Fitting result for the social media activities.



	Keywords
x_1	Kohls
x_2	JCPenney
x_3	Nordstrom
x_4	Forever21

(c-i) Fitting result vs. original data (c-ii) Interaction (c-iii) Full parameter set

(c) Fitting result for the apparel company activities.



	Keywords
x_1	Amazon
x_2	Walmart
x_3	Home Depot
x_4	Best buy
x_5	Lowes
x_6	Costco

(d-i) Fitting result vs. original data (d-ii) Interaction (d-iii) Full parameter set

(d) Fitting result for the retail companies.

Figure 5: Fitting results of EcoWeb for four areas, i.e., (a) Programming languages, (b) social media, (c) apparel and (d) retail companies. Our model (solid lines) fits the original data (in circles) very well; spots competitors (indicated by edges); and spots the strongest seasonal patterns. See text for more observations.

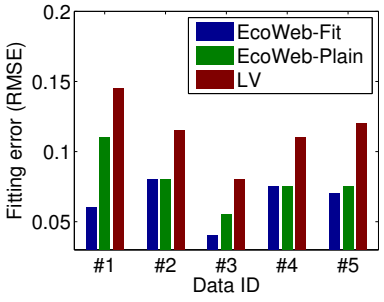


Figure 6: Accuracy of ECOWEB-FIT: Fitting error (RMSE) between original and estimated volume for five sequence sets (#1-#5) (lower is better).

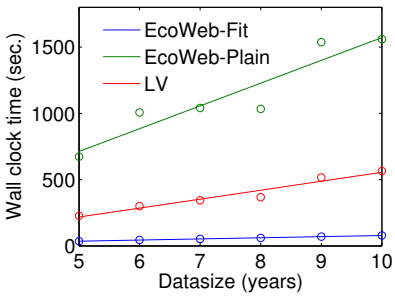


Figure 7: ECOWEB-FIT scales linearly: Wall clock time vs. dataset size (years). ECOWEB-FIT is 7 times faster than *LV* and 20 times faster than ECOWEB-Plain.

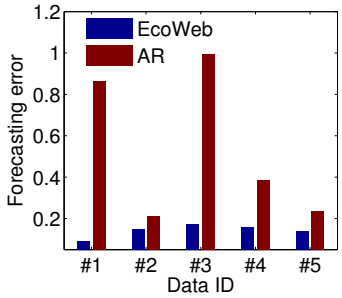
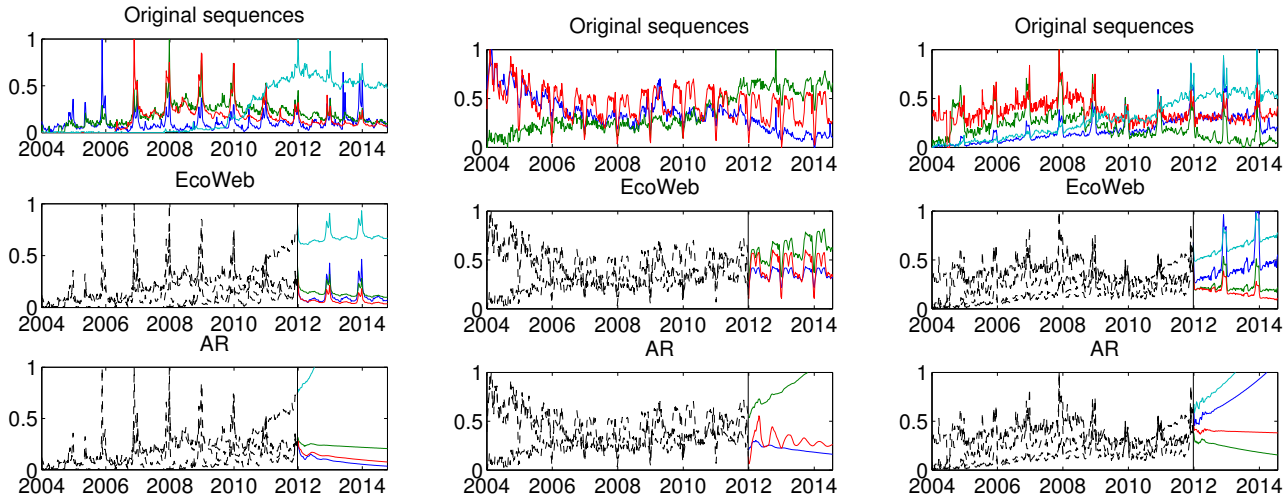


Figure 8: Forecasting error for each sequence set (#1-#5). Lower is better. Our method achieves high forecasting accuracy for every sequence set.



(a) Video games (#1) (b) Programming languages (#2) (c) Apparel companies (#4)

Figure 9: Forecasting future evolutions. Top: original sequence set. Middle and bottom: ECOWEB clearly outperforms *AR*. Both methods train the model parameters using $2/3$ of each sequence set, and then start forecasting (at the vertical line, i.e., 2012).

7. EcoWeb AT WORK - FORECASTING

Here, we describe the most important application of ECOWEB, namely, forecasting the future dynamics of co-evolving activities. Figure 8 shows the forecasting accuracy of five sequence sets (i.e., #1-#5) and Figure 9 shows results of our forecasting in relation to three sequence sets: (#1, #2 and #4). We trained the model parameters by using the $2/3$ values for each sequence set (black lines in Figure 9), and then forecasted the following years (colored lines, from 2012). We compared ECOWEB with the auto regressive (*AR*) model. For a fair comparison, we used coefficients that were the same size as our model parameters. In Figure 9, the top, middle and bottom rows show the original sequences, and the forecast results of ECOWEB and *AR*, respectively. As shown in Figure 9, our method successfully forecasted the long-range evolution of each sequence, as well as seasonal spikes, while *AR* failed to capture the non-linear evolutions.

The forecasting error (RMSE) between the original and the forecasted volume of each dataset is shown in Figure 8. A lower value indicates a better forecasting accuracy. Unlike *AR*, our method achieves high forecasting accuracy for every sequence set.

8. CONCLUSIONS

We presented ECOWEB, an intuitive model for mining large scale co-evolving online activities. Our main idea is that online activities behave like species in an ecological system in that they compete for resources (such as user attention), and they evolve over time according to a non-linear dynamical system. Our proposed method has the following appealing properties:

1. **Effective**: it detects important patterns, hidden interactions and seasonalities that match human intuition.
2. **Automatic**: it needs no parameter tuning, thanks to our coding scheme.
3. **Scalable**: it is linear on the input size.
4. **Practical**: it can undertake long-range forecasting and outperforms existing methods (Section 7).

Acknowledgement. The authors would like to thank Christina Cowan for her help with interpreting the patterns of apparel companies. This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research Number 26730060, 24500138, 26280112, 25·7946. This material is based upon work supported by the National Science Foundation under Grants No. CNS-1314632 and IIS-1408924; and by the Army Research Laboratory (ARL) under Cooperative Agreement Number W911NF-09-2-0053; and by a Google Focused Research Award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, ARL, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

9. REFERENCES

[1] C. C. Aggarwal. The setwise stream classification problem. In *KDD*, pages 432–441, 2014.

[2] R. M. Anderson and R. M. May. *Infectious Diseases of Humans Dynamics and Control*. Oxford University Press, 1992.

[3] A. Beutel, B. A. Prakash, R. Rosenfeld, and C. Faloutsos. Interacting viruses in networks: can both survive? In *KDD*, pages 426–434, 2012.

[4] C. Böhm, C. Faloutsos, J.-Y. Pan, and C. Plant. Ric: Parameter-free noise-robust clustering. *TKDD*, 1(3), 2007.

[5] C. Böhm, C. Faloutsos, and C. Plant. Outlier-robust clustering using independent components. In *SIGMOD*, pages 185–198, 2008.

[6] F. Brauer and C. Castillo-Chavez. *Mathematical models in population biology and epidemiology*, volume 40. Springer Verlag, New York, 2001.

[7] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos. Fully automatic cross-associations. In *KDD*, pages 79–88, 2004.

[8] H. Choi and H. R. Varian. Predicting the present with google trends. *The Economic Record*, 88(s1):2–9, 2012.

[9] I. N. Davidson, S. Gilpin, O. T. Carmichael, and P. B. Walker. Network discovery via constrained tensor analysis of fmri data. In *KDD*, pages 194–202, 2013.

[10] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *ACM Trans. Internet Techn.*, 3(1):1–27, 2003.

[11] J. Ferlez, C. Faloutsos, J. Leskovec, D. Mladenic, and M. Grobelnik. Monitoring network evolution using MDL. In *ICDE*, pages 1328–1330.

[12] F. Figueiredo, J. M. Almeida, Y. Matsubara, B. Ribeiro, and C. Faloutsos. Revisit behavior in social media: The phoenix-r model and discoveries. In *PKDD*, pages 386–401, 2014.

[13] J. Ginsberg, M. Mohebbi, R. Patel, L. Brammer, M. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457:1012–1014, 2009.

[14] S. Goel, J. Hofman, S. Lahaie, D. Pennock, and D. Watts. Predicting consumer behavior with web search. *PNAS*, 2010.

[15] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *KDD*, pages 78–87, 2005.

[16] A. Hyvärinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Netw.*, 13(4-5):411–430, 2000.

[17] E. Jackson. *Perspectives of Nonlinear Dynamics:*. Cambridge University Press, 1992.

[18] A. Jain, E. Y. Chang, and Y.-F. Wang. Adaptive stream resource management using kalman filters. In *SIGMOD*, pages 11–22, 2004.

[19] I. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986.

[20] R. L. Jr., A. Veloso, A. M. Pereira, W. M. Jr., R. Ferreira, and S. Parthasarathy. Economically-efficient sentiment stream analysis. In *SIGIR*, pages 637–646, 2014.

[21] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani. An online algorithm for segmenting time series. In *ICDM*, pages 289–296, 2001.

[22] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[23] F. Korn, H. V. Jagadish, and C. Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In *SIGMOD 1997*, pages 289–300, 1997.

[24] R. Kumar, M. Mahdian, and M. McGlohon. Dynamics of conversations. In *KDD*, pages 553–562, 2010.

[25] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD*, pages 593–604, 2007.

[26] W. Leontief. *Input-output economics*. Oxford University Press, 1986.

[27] J. Leskovec, L. Backstrom, and J. M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506, 2009.

[28] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD*, pages 462–470, 2008.

[29] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.

[30] L. Li, B. A. Prakash, and C. Faloutsos. Parsimonious linear fingerprinting for time series. *PVLDB*, 3(1):385–396, 2010.

[31] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *WWW*, pages 691–700, 2010.

[32] Y. Matsubara, Y. Sakurai, and C. Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *SIGMOD*, 2014.

[33] Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, and M. Yoshikawa. Fast mining and forecasting of complex time-stamped events. In *KDD*, pages 271–279, 2012.

[34] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *KDD*, pages 6–14, 2012.

[35] Y. Matsubara, Y. Sakurai, W. G. van Panhuis, and C. Faloutsos. FUNNEL: automatic mining of spatially coevolving epidemics. In *KDD*, pages 105–114, 2014.

[36] R. M. May. Qualitative stability in model ecosystems. *Ecology*, 54(3):638–641, 1973.

[37] J. Murray. *Mathematical Biology II: Spatial Models and Biomedical Applications*. Intercisciplinary Applied Mathematics: Mathematical Biology. Springer, 2003.

[38] M. Nowak. *Evolutionary Dynamics*. Harvard University Press, 2006.

[39] E. Odum and G. Barrett. *Fundamentals of Ecology*. Thomson Brooks/Cole, 2005.

[40] S. Papadimitriou, A. Brockwell, and C. Faloutsos. Adaptive, hands-off stream mining. In *VLDB*, pages 560–571, 2003.

[41] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *VLDB*, pages 697–708, 2005.

[42] S. Papadimitriou and P. S. Yu. Optimal multi-scale patterns in time series streams. In *SIGMOD*, pages 647–658, 2006.

[43] B. A. Prakash, A. Beutel, R. Rosenfeld, and C. Faloutsos. Winner takes all: competing viruses or ideas on fair-play networks. In *WWW*, pages 1037–1046, 2012.

[44] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos. Threshold conditions for arbitrary cascade models on arbitrary networks. In *ICDM*, pages 537–546, 2011.

[45] T. Preis, H. S. Moat, and H. E. Stanley. Quantifying trading behavior in financial markets using google trends. *Sci. Rep.*, 3, 04 2013.

[46] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, pages 262–270, 2012.

[47] B. Ribeiro. Modeling and predicting the growth and death of membership-based websites. In *WWW*, pages 653–664, 2014.

[48] Y. Sakurai, S. Papadimitriou, and C. Faloutsos. Braid: Stream mining through group lag correlations. In *SIGMOD*, pages 599–610, 2005.

[49] E. Shmueli, A. Kagian, Y. Koren, and R. Lempel. Care to comment?: recommendations for commenting on news stories. In *WWW*, pages 429–438, 2012.

[50] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD*, pages 374–383, 2006.

[51] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *SIGMOD*, pages 611–622, 2004.

[52] M. Vlachos, D. Gunopulos, and G. Kollios. Discovering similar multidimensional trajectories. In *ICDE*, pages 673–684, 2002.

[53] H. Wang, J. Yin, J. Pei, P. S. Yu, and J. X. Yu. Suppressing model overfitting in mining concept-drifting data streams. In *KDD*, pages 736–741, 2006.

[54] P. Wang, H. Wang, and W. Wang. Finding semantics in time series. In *SIGMOD Conference*, pages 385–396, 2011.

[55] J. Yang, J. J. McAuley, J. Leskovec, P. LePendu, and N. Shah. Finding progression stages in time-evolving event sequences. In *WWW*, pages 783–794, 2014.

[56] R. Zafarani and H. Liu. Connecting users across social media sites: a behavioral-modeling approach. In *KDD*, pages 41–49, 2013.

[57] Y. Zhao, N. Sundaresan, Z. Shen, and P. S. Yu. Anatomy of a web-scale resale market: a data mining approach. In *WWW*, pages 1533–1544, 2013.