Mining Reaction and Diffusion Dynamics in Social Activities

Taichi Murayama
SANKEN, Osaka University
Japan
taichi@sanken.osaka-u.ac.jp

Yasuko Matsubara
SANKEN, Osaka University
Japan
yasuko@sanken.osaka-u.ac.jp

Yasushi Sakurai
SANKEN, Osaka University
Japan
yasushi@sanken.osaka-u.ac.jp

ABSTRACT

Large quantifies of online user activity data, such as weekly web search volumes, which co-evolve with the mutual influence of several queries and locations, serve as an important social sensor. It is an important task to accurately forecast the future activity by discovering latent interactions from such data, i.e., the ecosystems between each query and the flow of influences between each area. However, this is a difficult problem in terms of data quantity and complex patterns covering the dynamics. To tackle the problem, we propose FluxCube, which is an effective mining method that forecasts large collections of co-evolving online user activity and provides good interpretability. Our model is the expansion of a combination of two mathematical models: a reaction-diffusion system provides a framework for modeling the flow of influences between local area groups and an ecological system models the latent interactions between each query. Also, by leveraging the concept of physics-informed neural networks, FluxCube achieves high interpretability obtained from the parameters and high forecasting performance, together. Extensive experiments on real datasets showed that FluxCube outperforms comparable models in terms of the forecasting accuracy, and each component in FluxCube contributes to the enhanced performance. We then show some case studies that FluxCube can extract useful latent interactions between queries and area groups.

CCS CONCEPTS

• **Information systems → Data mining**; • **Mathematics of computing → Time series analysis**.

KEYWORDS

time series, data mininig, web data, reaction-diffusion systems, neural networks

ACM Reference Format:
Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. 2022. Mining Reaction and Diffusion Dynamics in Social Activities. In *Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM '22), Oct. 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557396

1 INTRODUCTION

The increasing volume of online user activity provides vital new opportunities to measure and understand the collective behavior of social and economic evolutions such as influenza prediction [17], the impact of individual performance [16], user activity modeling [36], and other problems [28, 44, 57]. A record of online user activity can play the role of a social sensor and offers important insights into people's decision-making. In particular, accurately forecasting the volume of online user activity and unraveling hidden patterns and interactions within them have many benefits. For example, marketers want to know the future popular volume of products and the relationship between multiple products and locations in online user attention to enable appropriate advertisement and new product development. They can avoid wasting human and material resources by accurately modeling future behavior.

The modeling of online user activity remains a challenging task due to the increasing data volume, which has multiple domains and a network of time series with mutual influences. For example, when we consider online user behavior analysis using web search activities as observations, which could be in the form of 3rd-order tensor (timestamp, location, keyword), it is important to deal with the following problems: (a) ***Capturing latent interactions and diffusion between observable data***: Many time series for individual locations and keywords are not independent. For example, the search volume per keyword in the same category may compete for user resources [35]. Also, the popularity of a keyword in one location diffuses and affects the search volume in another location [40]. In other words, we consider changes in the popularity of a keyword have the properties of flux phenomena, that can be expressed as a flow of influence. (b) ***Capturing latent temporal patterns behind observable data***: Many time series contain several patterns, such as trends and seasonality. Such patterns behind time series reveal the relationship between people's activities, such as Black Friday. However, it is extremely difficult to design an appropriate model for such patterns by hand without knowing their real characteristics in advance. The modeling method should therefore be fully automatic as regards estimating the hidden pattern for understanding data structures and saving human resources. (c) ***Accurate forecast***: Accurate time series forecasts give marketers useful insights into the future trends of their keywords and avoid wasting human and material resources. The key challenge is to achieve high forecasting accuracy while addressing the two aforementioned issues related to the interpretability of observable data.

This paper focuses on an important time series modeling and forecasting task, and we design FluxCube, which is an effective mining method that forecasts large collections of co-evolving online user activity and provides good interpretability. This model solves (a) and (b) in the above problems with the combination of mathematical models: a reaction-diffusion system [19] representing the change in space and time of chemical substances and the Lotka-Volterra population model [18, 34] describing the population dynamics of prey and predator. Also, it solves (c) with the idea

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10.
https://doi.org/10.1145/3511808.3557396

(a) Modeling and forecasting tensor data of each area group (b) Latent interactions (c) Diffusion process of each area group

Figure 1: Modeling power of FluxCube for online user activity data related to six artists (World#1): (a) Given the original data (gray lines), FluxCube captures the dynamics of social activities (blue). Futhermore, FluxCube provides a multiple steps ahead forecast (red). (b) Our method detects the hidden interactions between keywords in any country. The arrows indicate the direction and the type of influence. (c) It also automatically discovers similar area groups based on their keyword interactions and then finds the flow of influence of each keyword between area groups. The figure visualizes the flow of influence in 2015.

of physics-informed neural networks [46] to conduct supervised learning tasks while respecting any given laws of physics described by general nonlinear partial differential equations. Intuitively, the problem we wish to solve is as follows.

Informal Problem 1. Given a tensor $\mathcal{X}^c$ up to the time point t_c, which consists of elements at l locations for d keywords, i.e., $\mathcal{X}^c = \{x_{tij}\}_{t,i,j=1}^{t_c,l,d}$,

- Find trends and seasonal patterns
- Find area groups of similar dynamics
- Find latent interections between keywords
- Find the flow of influence between area groups
- forecast l_f-step ahead future values, i.e., $\mathcal{X}^f = \{x_{tij}\}$ where $(t = t_c + 1, \ldots, t_c + l_f; i = 1, \ldots, l; j = 1, \ldots, d)$

1.1 Preview of our results

Figures 1 and 2 show the results obtained with FluxCube for modeling online user activity data related to six artists (World#1). Specifically, our method captures the following properties:

- **Long-term trends**: Figure 1 (a) shows tensor data that consists of weekly online search volumes in relation to artists in fifty countries. FluxCube automatically captures the trends and the hidden interactions between keywords and area groups in the modeling term (blue part) and realizes the accurate long-term forecasting over two years (red part).
- **Latent interaction between keywords**: Figure 1 (b) shows the latent interaction between each keyword. FluxCube uncovers four types of interaction (competing, commensal, parasitic, mutualistic) between the keywords, where the arrow colors and widths show the interaction type and its connection intensity, from the tensor data in each location. For example, while the relationship between maroon 5 and beyonce is competing in the United States, it is mutualistic in Japan. Also, the competing relationship between taylor swift and katy perry is common in the two countries.
- **Diffusion process of each area group**: FluxCube finds the location clustering and the flow of influence between each area group, as shown in Figure 1 (c). This figure shows that the colors of the countries on the map indicate area groups, and the colors and directions of the arrows indicate the keyword type and the influence flow, respectively. FluxCube discovers the most suitable clustering result: Europe in group B, North America in group C, and East Asia in group D. The influences of each keyword on each group, which are represented by arrows, have time-varying intensities, which FluxCube also captures, as shown in Figures 2 (a) and (b). There are constant influence flows of maroon 5 and katy perry from group C to D, while there is an influence flow of taylor swift in the early 2010s from group C to E, which has disappeared. Our model provides the flow of influence in an intuitive form.
- **Seasonality**: Figure 2 (c) shows the seasonalities extracted from the search volume data. Two kinds of annual patterns resulting from "Grammy Awards" and "New Year holidays" are found from search volumes in the United States and Brazil. It is important to discover such periodic patterns if we are to accurately forecast and model the dynamics.

2 RELATED WORK

In this section, we briefly describe investigations related to our research. The details of related work are separated into three categories: time series forecasting and modeling, web data modeling, and a physics-informed neural network.

Figure 2: Latent pattern mining of FluxCube for search volumes related to six artists in the US: (a) and (b) show that it automatically detects time-varying influence intensities related to each artist from group C. (c) shows that it automatically detects seasonality patterns about keyword beyonce in the US and Brazil.

Table 1: Symbols and definitions

Symbol	Definition
$C_2 \bullet C_5$	Modeling and forecasting time point
$X^2 \bullet X^5$	Modeling and forecasting tensor data
$;_2 \bullet ;_5$	Time interval for modeling and forecasting
$! \bullet$	Number of locations and keywords
$3_;$	Number of area groups
0_9	Intrinsic growth rate of keyword 9
1_9	Carrying capacity of keyword 9
2_{99^0}	Intra/inter-keywords interaction strength from 9^0-th keyword to 9-th keyword
$J^C_{<=:}$	Influence flow of the keyword : from location = to <
Y	Seasonality latent pattern matrix
?	Period of seasonality
U, V	Weights of regularization terms for loss function
o^8	Compression representations of item interactions in location 8
$6^1 \bullet ”””\ 6_{3_;}$	Area groups
$^{0}_{3_;}$	Parameter sets related to $6_;$

Time series forecasting and modeling: There is a lot of interest in time series forecasting and modeling. Conventional approaches to time series forecasting and modeling are based on statistical models such as auto regression (AR), state space models (SSM), Kalman filters (KF), and their extensions [9, 11, 30]. Some models combining these classical methods with dimension reduction, such as TRMF [55] and SMF [20] have shown useful results in the field of data mining. However, these methods are limited as regards the interpretability of observation data because of the dimension reduction. In recent years, many models have been based on neural networks thanks to their rapid development. These networs include convolutional neural networks (CNNs) [2, 4, 41] and recurrent neural networks (RNNs) [26, 47, 52], which capture the temporal variation. In particular, time series forecasting models using Transformer, developed in the field of a natural language preocessing [54], achieve high prediction performance [12, 31, 33]. For example, Informer [58], one of the Transformer-based models, achieves a successful outcome in long-ahead forecasting. Deep learning-based models guarantee high forecasting performance; however, they are still problematic in terms of their interpretability. We achieve high interpretability as well as a high predictive performance by our mathematical model fused with deep learning.

Web data modeling: Online user activity data, such as web search query data and social media posts, are expected to capture human movements as social sensors [1, 6, 49] and are utilized in real-world applications such as influenza forecasting [17], talent flow forecasting [56], and item popularity prediction [39]. This potential for real-world application has led to a huge interest in uncovering and modeling the movement of web data. The author of [3, 43] investigated the competing dynamics of two keywords on the web and found factors impacting final states based on the infection model. Various other modeling studies on the web have been conducted as follows: competing dynamics of membership-based websites [48], modeling the content and viewer dwell time on social media platforms [27], modeling the online activities into return and exploration across online communities [21, 29], extracting the relationships between search queries and external events [22, 23], and modeling online user behavior patterns in the knowledge community [37]. CubeCast [24] and EcoWeb [35] are focused on modeling the dynamics of web search queries. CubeCast is an online algorithm designed to capture useful time series segmentation and seasonality patterns. EcoWeb is a modeling method for capturing keyword interactions from the dynamics of web search queries based on biological mathematics. These studies revealed the interaction of search queries in a single location, but did not explore how keywords interact between locations. Our proposed model discovers the flow of influence between locations from the time series of each query.

Physics-informed neural network: Recently, neural networks have found applications in the context of numerical solutions of partial differential equations and dynamical systems because a neural network is a universal approximator [46]. These models, which are called physics-informed neural networks, can solve forward problems [45], where the approximate solutions of governing equations are obtained, as well as inverse problems, where parameters involved in the governing equation are obtained from the observation data [13]. We try to utilize the idea to represent some of the parameters in our partial differential equations flexibly with a neural network.

3 OVERVIEW

This section provides an overview of our proposed model, namely FluxCube, for mining and forecasting co-evolving online user activity data. We first introduce related notations and definitions and then describe the characteristics of FluxCube.

3.1 Problem definition

Table 1 lists the main symbols that we use throughout this paper. We consider online user activity data to be a 3rd-order tensor, which is denoted by $X\ 2\ R^{)\ !\ }$, where $)$ is the number of time points, and $!$ and are the numbers of locations and keywords, respectively. The element G_{C89} in X corresponds to the search volume at time C in the 8-th location of the 9-th keyword. Our overall aim is to realize the long-term forecasting of a tensor X while extracting the hidden patterns and latent interactions. We define $\mathcal{X}^c = \{x_{t,i,j}\}_{t,i,j=1}^{t_c,l,d}$ as observable tensor data, whose length is denoted by t_c. Similarly, let $\mathcal{X}^f = \{x_{t,i,j}\}_{t=t_c+1,i,j=1}^{t_f,l,d}$ denotes a partial tensor from t_c to t_f for forecasting. Here, we define t_c and l_f as a time interval for modeling and forecasting. Finally, we formally define our problem as follows.

Problem 1 (l_f-step ahead forecasting). Given a tensor $\mathcal{X}^c = \{x_{t,i,j}\}_{t,i,j=1}^{t_c,l,d}$ up to the time point t_c; Forecast l_f-step ahead future values $\hat{\mathcal{X}}^f = \{x_{t,i,j}\}_{t=t_c+1,i,j=1}^{t_f,l,d}$, where $t_f = t_c + l_f$.

3.2 Reaction-diffusion system

Our model for capturing the latent interactions between keywords and area groups is inspired by a reaction-diffusion system. A reaction-diffusion system is a mathematical model corresponding to physical phenomena such as the change in space and time of chemical substances, represented as partial differential equations. The system can also describe the dynamic processes of non-chemical areas such as biology, geology, and physics [8, 14, 15, 42]. The general form of a reaction-diffusion system can be described with the following equations:

$$\frac{\partial u}{\partial t} = f(u, t) + D\Delta u \tag{1}$$

where u represents the concentration values in a chemical substance, t represents the present time, and D is the diffusion coefficient. The first term on the right hand side, $f(u, t)$, represents the reaction term, which accounts for all local reactions, and the second term, $D\Delta u$, represents the diffusion term, which accounts for how the substance spreads to other locations.

Our model is an extended reaction-diffusion system for modeling and forecasting online user activity data. For example, the related search volumes in the data interact with each other and compete for user resources. The interaction between keywords in a location can be represented as a reaction term in the reaction-diffusion system. Also, a vogue for any keyword in a location can affect that in another location. Such an influence flow of any keyword between locations can be represented as a diffusion term. Thus, we assume that the interactions between area groups and keywords can be represented by a diffusion-reaction system.

3.3 Capabilities

To capture all the components outlined in the introduction, we present our model, namely FluxCube, which can model the latent patterns and interactions underlying online user activity 3rd-order tensor data. So, how can we build our model so that it models the user activity data while capturing the latent patterns and interactions and achieving accurate forecasting? Specifically, our model should have the following three capabilities:

- **Latent interaction and diffusion**: User activity data evolves naturally over time depending on many latent factors, such as user preferences and customs for online web-search activities. We need to capture the latent interactions of keywords and the influence flow of locations from the real-time series data. To explicitly capture such interactions, we propose using a reaction-diffusion system [19] including the Lotka-Volterra population model [34], which is expressed in differential equations.
- **Seasonality**: We should also note that online user activity has certain annual patterns. As Figure 2 (c) shows, there are annual patterns such as a huge spike in New Year holidays in online user activity data. Capturing seasonalities realizes our suitable modeling.
- **Time-varying components**: User activity data change significantly due to external events such as the influence of other locations. The variables in the differential equation cannot easily explain the relationships between locations that vary greatly with time and external events, for which we require a flexible modeling method. To achieve this, we leverage the idea of physics-informed neural networks [46]. Intuitively, our model combines a neural network with the characteristics of universal approximation [32] and differential equations with high interpretability.

Figure 3: Conceptual image of our model targeting ecommerce data (US#1) in Washington state. Our model consists of three parts: (P1) Reaction term represents the interaction between keywords, (P2) Diffusion term represents the influence flow of each keyword between locations, and (P3) Seasonal term represents the seasonality of each keyword.

4 FLUXCUBE

Here, we describe the formula of FluxCube. We represent the base form of our model applied to online user activity data, mainly web search volumes, which is inspired by the reaction-diffusion system, Eq. (1), in the following:

$$\frac{\partial x_{t,i,j}}{\partial t} = f(x_{t,i,j} \mid \mathcal{X}^c_{t,i,:}) + g(x_{t,i,j} \mid \mathcal{X}^c_{t,:,j}, t) \tag{2}$$

$$x_{t+1,i,j} = s\left(\frac{\partial x_{t,i,j}}{\partial t} + x_{t,i,j}\right) \tag{3}$$

The first term in Eq. (2) serves as the reaction term for extracting the interaction between keywords in location i (P1). The second term in Eq. (2) serves as the diffusion term for extracting the influence flow of keyword j between locations (P2). Finally, we set Eq. (3) to capture seasonality s (P3). A conceptual drawing of our modeling method is shown in Figure 3.

Next, we introduce each component of the above form of our model in steps.

4.1 Reaction term (P1)

We first introduce the reaction term in detail. The term aims to model the interaction between keywords within a location.

Observation of the search volume of each keyword on the web shows that the keywords compete for user attention. No keyword can survive on the web if no one pays attention to that topic. Besides, the number of instances of user attention on the web is finite. This relationship between keywords and user attention on the web is very similar to the relationship between species and food resources in the jungle, e.g., squirrel monkeys and spider monkeys compete for fruit and no species can survive without resources. We employ the Lotka-Volterra population model [18, 34], which represents the growth of the species considering the interaction between them in biological mathematics, as the reaction term.

The reaction term in Eq. (2) for keyword j is represented below:

$$R(x_{t,i,j}, \mathcal{X}^2_{t,i,:}) = a_j x_{t,i,j}\left(1 - \frac{\sum_{j'} c_{jj'} x_{t,i,j'}}{b_j}\right), \quad (4)$$

where, $a_j > 0$, $b_j > 0$ and $c_{jj} = 1$. The size of each parameter depends on the number of keywords: $\mathbf{a} \in \mathbb{R}^{d_k}$, $\mathbf{b} \in \mathbb{R}^{d_k}$, and $\mathbf{C} \in \mathbb{R}^{d_k \times d_k}$. Each parameter is interpreted as:

- a_j: intrinsic growth rate of keyword j,
- b_j: carrying capacity of keyword j,
- $c_{jj'}$: intra/inter-keyword interaction strength from the j'-th keyword to the j-th keyword, which is each value in $\mathbf{C}$.

The signs of the off-diagonal elements of $\mathbf{C}$ provide an interpretation of the latent relationship between the two keywords:

- $c_{jj'} > 0, c_{j'j} > 0$: a competitive relationship
- $c_{jj'} < 0, c_{j'j} > 0$: a parasitic relationship
- $c_{jj'} < 0, c_{j'j} = 0$: a commensal relationship
- $c_{jj'} < 0, c_{j'j} < 0$: a mutualistic relationship

We expect these characteristics to be available for modeling online user activity on the web and extracting the relationships between keywords.

4.2 Diffusion term (P2)

We then introduce the diffusion term in detail. The term aims to model the influence flow of any keyword between locations.

Can partial differential equations and mathematical models adequately represent the relationships between locations in online user activity on the web? This task is hard to achieve. The reaction-diffusion system in chemistry can represent the change in space and time of chemical substances by a constant such as the diffusion coefficient, based on the observation that the spread of a chemical substance is constant without external influences. However, on the web, interactions of any keyword between locations are not constant due to external factors. To capture the time change of the interaction, a dynamic linear model with time-varying coefficients could be utilized, but it is difficult to take account of complex phenomena and rapid changes. Here, we utilize a neural network with universal approximation as part of our mathematical model to represent the changing interactions between locations over time. This corresponds to a kind of physics-informed neural network [46]. Concretely, we represent the diffusion term in Eq. (2) for keyword j at location i as below:

$$\mathbf{J}^t = \mathrm{ReLU}(\mathrm{RNN}(1:t)) \quad (5)$$

$$\mathbf{K}^t_i = \mathbf{J}^t_i \odot \mathcal{X}_{t,:} \quad (i = 1, \ldots, d_l) \quad (6)$$

$$D(x_{t,i,j}, \mathcal{X}^2_{t,:}, t) = \sum_{i'}^{d_l} K^t_{i,i',j} \quad (7)$$

where $\mathbf{J}^t \in \mathbb{R}^{d_l \times d_l \times d_k}$ and $\mathbf{K}^t \in \mathbb{R}^{d_l \times d_k}$, and $\mathbf{J}^t_{i,i,:} = 0$. $\mathbf{J}^t_i \in \mathbb{R}^{d_l \times d_k}$ and $\mathbf{K}^t_i \in \mathbb{R}^{d_k}$ represent the i-th vectors of $\mathbf{J}^t$ and $\mathbf{K}^t$, respectively. We express the flow of attention of each user between locations using a recurrent neural network (RNN) [51], which captures time-varying parameters using time $1:t$ as input in Eq. (5). The RNN enables our model to be more expressive by allowing time-varying parameters along with temporal dependencies in the covariates. The output of RNN in Eq. (5) is transformed to a tensor form: $\mathbf{J}^t$ is a form of 3rd-order tensor and $J^t_{m,n,k}$ indicates the degree to which location n contributes to the popularity of the keyword k in location m. In other words, $\mathbf{J}^t_{m,n,:}$ represents the influence intensities of the keyword from n to m, which is provided in an intuitive form for us. We show some examples of $\mathbf{J}^t (t = 1, \ldots, n_t)$ in Figure 3 (a) and (b). A total of the influences that keyword j at location i at time t receives from other locations is expressed by multiplying the observed data $\mathcal{X}$ with the output in Eq. (5) ($\mathbf{J}$), as shown in Eq. (6) and (7). By applying a neural network to a part of our mathematical model, we expect to achieve both flexible modeling and high explainability.

4.3 Seasonality (P3)

We finally consider seasonal/cyclic patterns in user activity data by extending our model. Each keyword always has certain users' attention; however, the users change their behavior dynamically, according to various seasonal events, e.g., Christmas Day and Black Friday. We need to detect hidden seasonal activities with our model. More specifically, we model the user activity value at the next step considering the seasonality pattern from our reaction-diffusion system in Eq. (3), as below:

$$\frac{\partial x_{t,i,j}}{\partial t}, x_{t,i,j} = (1 + [\mathbf{Y}]_{t \bmod p, i, j}) \odot \frac{\partial x_{t,i,j}}{\partial t}, x_{t,i,j} \quad (8)$$

where $\mathbf{Y} \in \mathbb{R}^{p \times d_l \times d_k}$, $\mathbf{Y} \geq 0$, p is period of seasonality, and $\odot$ indicates the element-wise product. $\mathbf{Y}$ acts as a projection matrix for latent seasonal patterns, such as the seasonal term in Figure 3.

4.4 Loss function

To train *FluxCube*, we use the mean squared error (MSE) loss with regularization terms as below:

$$\|\mathcal{X} - \hat{\mathcal{X}}\|^2 + \alpha \sum \|\mathbf{J}\|^2 + \beta \sum \|\mathbf{Y}\|^2 \quad (9)$$

where α and β are hyper-parameters for controling the weights of the regularization terms, and $\hat{\mathcal{X}}$ is our modeling value. The first term is MSE loss that penalizes the difference between modeling and observation values, and the second and third terms are regularization terms that encourage the suppression of the adoption of large values for two parameters: the influence between locations ($\mathbf{J}$) and the latent seasonal patterns ($\mathbf{Y}$).

5 OPTIMIZATION ALGORITHM

This section presents our optimization algorithm for learning *FluxCube* while grouping locations with similar user activity characteristics.

Thus far, we have proposed *FluxCube* for modeling and forecasting online user activity data with multiple locations. There are two challenges in fitting our model to data with many locations: (a) the learning can be inefficient and unstable because there are large candidate parameters that must be estimated and converged, and (b) the interactions between locations become more complex and results less interpretable. Here, we propose an algorithm to learn FluxCube where we cluster locations that have observed data with similar characteristics as the same area group.

5.1 Automatic location clustering

Our hypothesis for clustering online user activity data as regards location is simple. It is that locations with similar interactions between items represented by the reaction term (P1) can be considered as the same area group. The clustering method refers to [10]; according to the number of area group $3_;$, we cluster the locations, the number of which is $!$, using Uniform Manifold Approximation and Projection (UMAP) [38] and K-means: The equations are as below:

$$o^8 = *" \% \; ¹a^8 • b^8 • l\; ^8º \qquad ¹8 = 1• ”””• !º \qquad (10)$$

$$6^{1}• ”””• 6^{3_;} = \text{K-means}¹o • 3_; º \qquad (11)$$

where $o^8 \in R^2$, $o \in R^{! \; 2}$ is the compression representation of the keyword interactions and $!$ is the number of locations. We perform clustering using three parameters $(a^8 • b^8 • l^8)$ that represent the interaction of keywords at location 8 in Eq. (4) in the reaction term. These three parameters are compressed to the two-dimensional vector o^8 by UMAP. We then cluster the compression representation o by K-means into $3_;$ area groups $(6^1 • ”””• 6^{3_;})$.

5.2 Finding the appropriate number of area groups

We describe how to find the appropriate number of area groups $3_;$. The situation where our model with a small number of parameters including $3_;$ can adequately explain the observed data is the best. Here, we apply the minimum description length (MDL) principle [5, 50, 53] to find the appropriate $3_;$. The MDL principle enables us to determine the nature of good summarization by minimizing the sum of the data encoding cost and the model description cost. The cost is described as below:

$$Ÿ X^2 j \; ^0 ¡ \; ¸ \; Ÿ \; ^0_{3_;} \; ¡ \; ¸ \qquad (12)$$

where $Ÿ X^2 j \; ^0 ¡$ represnts the cost of describing the data X^2 given the model parameters ^{0}, $Ÿ \; ^0_{3_;} \; ¡$ shows the cost of describing $^0_{3_;}$, which is a parameter set that varies with $3_;$, and is a constant value, which is not affected by $3_;$. In short, it follows the assumption that the more we compress data, the more we can learn about the underlying pattern. We begin by defining the two components of the total cost more concretely.

Data encoding cost: We can encode the observation data X^2 using ^{0} based on Huffman coding [50]. The coding scheme assigns a number of bits to each value in X^2, which is the negative log-likelihood under a Gaussian distribution with mean $`$ and variance f^2, i.e.,

$$Ÿ X^2 j \; ^0 ¡ = \sum_{C•8•9=1}^{G•~~•} \log_2 ?`•f \; ¹G^2_{C•8•9} \; G^2_{C•8•9} º \qquad (13)$$

where, $G^2_{C•8•9} \; 2 \; X^2$ is the modeling value of $G^2_{C•8•9} \; 2 \; X^2$ by our proposed model.

Algorithm 1 Optimization

Input: Input tensor X^2

Output: Learned FluxCube with interpretable parameters

```
/* Initialize
$3_;$    1
$6^{3_;}$    f  1• ”””• ! g
MinCost    inf
while improving the cost do
    if $3_;$ ¡ 1 then
        /* Automatic location clustering (Section 5.1)
        $6^1 • ”””• 6^{3_;}$    ;DBC4A8=6 a$_{3_;}$ 1• b$_{3_;}$ 1• l$_{3_;}$ 1º
    end if
    /* FluxCube training (Section 4)
    FluxCube$_{3_;}$    )A08=8=6 X^2 • 6^1 • ”””• 6^{3_;} º
    /* Calculate the encoding cost (Section 5.2)
    >BC$_{3_;}$    5¹FluxCube$_{3_;}$ • X^2º •• 5¹ º : @D0C8>= 12º
    /* Update $3_;$
    if >BC$_{3_;}$ Ÿ MinCost then
        MinCost    >BC$_{3_;}$
        $3_;$    $3_;$ ¸ 1
    else
        return FluxCube$_{3_;}$
    end if
end while
```

Model description cost : The model cost is the number of bits needed to describe the model. If we use a more powerful model architecture, the total cost becomes higher. We focus only on parameters related to $3_;$, i.e., J in the diffusion term, because of our objective of selecting the appropriate $3_;$. Parameters unrelated to $3_;$ can be treated as constant value, which we can ignore in the cost calculation. Our model cost is represented as below:

$$Ÿ \; ^0_{3_;} \; ¡ = Ÿ 3_; \; ¡ \; ¸ \; Ÿ \; ¡ \qquad (14)$$

$$Ÿ 3_; \; ¡ = \log ¹3_; º \qquad (15)$$

$$Ÿ \; ¡ = jJ j \; ¹\log¹G^º \; ¸ \; 2 \; \log¹3_; º \; ¸ \; \log¹ \; º \; ¸ \; 2 \; º \; ¸ \; \log ¹jJ jº \qquad (16)$$

Here, $\log$ is the universal code length for integers, $j \; j$ describes the number of non-zero elements and 2 denotes the floating point cost (32 bits).

5.3 Optimization

Summarizing the descriptions so far, we have shown the overall procedure of learning FluxCube while grouping locations with similar user activity characteristics in Algorithm 1. The basic idea of the algorithm is that we search FluxCube to minimize the encoding cost in Eq. (12) while increasing the number of area groups $3_;$.

6 EXPERIMENTS AND RESULTS

6.1 Experimental settings

6.1.1 Datasets. We used the two kinds of online user activity data from different target areas on GoogleTrends, which contained

Table 2: Datsets

ID	Dataset	Query
US#1	E-commerce	Amazon/Apple/BestBuy/Costco/Craigslist/Ebay/ Homedepot/Kohls/Macys/Target/Walmart
US#2	VoD	AppleTV/ESPN/HBO/Hulu/Netflix/Sling/ Vudu/YouTube
US#3	Sweets	Cake/Candy/Chocolate/Cookie/Cupcake/ Gum/Icecream/Pie/Pudding
US#4	Facilities	Aquarium/Bookstore/Gym/Library/Museum/ Theater/Zoo
World#1	Music	Beyonce/KatyPerry/LadyGaga/Maroon5/ StevieWonder/TaylorSwift
World#2	SNS	Facebook/LINE/Slack/Snapchat/Twitter/ Viber/WhatsApp
World#3	Apparel	Gap/H&M/Primark/Uniqlo/Zara

weekly web search volumes collected for about 10 years, from January 1, 2011, to December 31, 2020.

- US data includes web search volumes for 50 states of the US. Four datasets of this type were constructed:
- World data includes three kinds of web search volumes for the top 50 countries ranked by GDP score.

The queries of our datasets are described in Table 2.

6.1.2 Comparative models. We used the following comparative models, which are state-of-the-art algorithms for modeling and forecasting time series:

- EcoWeb [35], which is intended for modeling online user activity data as well as our model, is a mathematical model constructed on the basis of differential equations.
- SMF [20] is a matrix factorization model that takes into account seasonal patterns. It is mainly used for time series forecasting and anomaly detection.
- DeepAR [52] has an encoder-decoder structure that employs an auto-regression RNN modeling probabilistic distribution in the future. Here, we selected a two-stack LSTM as the structure and its suitable number of hidden units in the validation period.
- Gated Recurrent Unit (GRU) [7] is a RNN-based model for time series forecasting. We employed an encoder-decoder architecture based on the GRU for the multi-step-ahead forecast. We also applied a dropout rate of 0.5 to the connection of the output layer.
- Informer [58], which is a transformer-based model based on ProbSparse self-attention and self-attention distilling, is known for its remarkable efficiency in long time series forecasting. We select a two-layer stack for both encoder and decoder and set 26 (half of the prediction sequence length) as the token length of the decoder.

6.1.3 Setting. We forecast the online user activity data using FluxCube and other comparative models for the validation of the prediction performance.

Dataset preprocessing: We set search volumes in the datasets from January 1, 2011, to December 31, 2017, as the training and modeling period (seven years $= C_2$), then from January 1, 2018, to December 31, 2018, as the validation period (one year), and then from January 1, 2019, to December 31, 2020, as the test period (two years). We then normalized their search volumes in the range 0 to 1.

Evaluation metrics : We evaluate the predictive performance of these models at three points, 13 weeks (3 months), 26 weeks (half a year), and 52 weeks (a year) ahead. Two evaluation metrics were utilized to compare the forecast performance levels of each model: root mean squared error (RMSE) and mean absolute error (MAE). A lower value in these metrics indicates better forecasting accuracy.

Hyper-parameter : All the parameters in neural network based models were updated in the Adam update rule [25]. We set 52 as the input and forecast length and MSE loss as the loss function for DeepAR, GRU, and Informer. We then conducted a grid search for parts of parameters such as hidden layers in comparative models. In the training of FluxCube, we selected the sizes of the RNN hidden layers in Eq. (5) as (16, 32, 64) in the validation period. We also set ? in Eq. (8) as 52 to capture annual patterns from weekly datasets, the hyperparameters U and V in Eq. (9) as 0”1, and the number of epochs as 2,000 with early stopping.

6.2 Results

The experimental results are presented in Table 3. These results indicate that FluxCube outperformed most of the comparative models, confirming the benefits of our model architecture and algorithm. EcoWeb, which is a linear model that captures only interactions between items to be built with the same motivation as FluxCube, was unstable for long-ahead forecasting. SMF, which focused on the extraction of seasonal patterns, outputs a good result of forecasting online user activity data with strong seasonality, compared to EcoWeb. RNN-based models such as DeepAR and GRU were superior baseline models and exhibited almost the same performance as Informer in the 13 week forecast, but in some cases, they showed unstable results in long-ahead forecasting such as 52 weeks ahead. These results indicate that it is not easy to improve the accuracy of long-ahead forecasts. On the other hand, Informer, which is built for long-term forecasting, achieved the highest accuracy among the comparative models. Informer showed a trend whereby the accuracy with the 52 weeks ahead forecast is not much worse than that with the 13 weeks ahead forecast.

FluxCube achieved the highest accuracy in 5 out of 7 datasets for RMSE and 6 out of 7 for MAE compared to other models. Our model shows high predictive performance as well as high interpretability by explicitly modeling the online user activity data with capturing the three hidden components: keyword interaction, location diffusion, and seasonality.

Ablation study : To demonstrate the effectiveness of all the components of FluxCube, we performed ablation experiments with three ablation models, namely, Constant model, and Full model w/o P2, and w/o P3. Constant model replaces the RNN in Eq. (5) in FluxCube with time-invariant parameters. This reveals the benefit of time-varying components. Full model w/o P2 and Full model w/o P3 are the removals of the diffusion term and the seasonal term in FluxCube respectively. Figure 4 shows the RMSE performances of 52 weeks ahead forecasting in each dataset. Full model

Table 3: Forecasting performance comparison

		13 weeks		26 weeks		52 weeks	
Dataset	Model	RMSE	MAE	RMSE	MAE	RMSE	MAE
US#1	EcoWeb	0.1470	0.0950	0.1554	0.1082	0.1654	0.1197
	SMF	0.0869	0.0620	0.0910	0.0654	0.1012	0.0674
	DeepAR	0.1003	0.0634	0.1302	0.0907	0.1385	0.1014
	GRU	0.1723	0.1175	0.1924	0.1374	0.2059	0.1525
	Informer	0.1477	0.1045	0.1375	0.0985	0.1575	0.1111
	FluxCube	0.0478	0.0257	0.0574	0.0323	0.0631	0.0365
US#2	EcoWeb	0.1440	0.1133	0.1981	0.1621	0.1920	0.1684
	SMF	0.0621	0.0445	0.0713	0.0522	0.0760	0.0529
	DeepAR	0.1471	0.1026	0.1781	0.1314	0.1906	0.1474
	GRU	0.1518	0.1171	0.1619	0.1231	0.1683	0.1440
	Informer	0.1277	0.0878	0.1292	0.0876	0.1436	0.1012
	FluxCube	0.0245	0.0130	0.0276	0.0156	0.0310	0.0181
US#3	EcoWeb	0.1555	0.1208	0.1730	0.1384	0.1754	0.1360
	SMF	0.0276	0.0186	0.0281	0.0170	0.0281	0.0190
	DeepAR	0.1107	0.0753	0.1267	0.0833	0.1309	0.0908
	GRU	0.1300	0.0869	0.1368	0.9843	0.1368	0.0939
	Informer	0.1322	0.0954	0.1311	0.0946	0.1279	0.0914
	FluxCube	0.0200	0.0121	0.0222	0.0136	0.0238	0.0148
US#4	EcoWeb	0.0847	0.0573	0.1348	0.0950	0.1511	0.1182
	SMF	0.0905	0.0762	0.1100	0.0751	0.1206	0.1070
	DeepAR	0.0927	0.0682	0.1662	0.1119	0.2168	0.1639
	GRU	0.1199	0.0872	0.1737	0.1223	0.2319	0.1764
	Informer	0.1014	0.0720	0.0992	0.0690	0.1055	0.0762
	FluxCube	0.0495	0.0256	0.0610	0.0358	0.0794	0.0503
World#1	EcoWeb	0.1259	0.0831	0.1422	0.1034	0.2101	0.1460
	SMF	0.0936	0.0783	0.0901	0.0602	0.1087	0.0787
	DeepAR	0.0900	0.0636	0.0929	0.0681	0.1395	0.0972
	GRU	0.0633	0.0452	0.0718	0.0501	0.0823	0.0572
	Informer	0.0704	0.0423	0.0719	0.0416	0.0738	0.0446
	FluxCube	0.0454	0.0274	0.0477	0.0286	0.0546	0.0331
World#2	EcoWeb	0.0908	0.0300	0.1089	0.0570	0.1353	0.0742
	SMF	0.0841	0.0436	0.0799	0.0454	0.0826	0.0480
	DeepAR	0.0374	0.0098	0.0585	0.0199	0.0643	0.0209
	GRU	0.0401	0.0159	0.0588	0.0174	0.0739	0.0254
	Informer	0.0371	0.0159	0.0595	0.0196	0.0642	0.0208
	FluxCube	0.0704	0.0271	0.0711	0.0304	0.0831	0.0351
World#3	EcoWeb	0.0523	0.0208	0.0626	0.0200	0.1080	0.0293
	SMF	0.0206	0.0111	0.0289	0.0160	0.0254	0.0195
	DeepAR	0.0211	0.0110	0.0275	0.0099	0.0613	0.0214
	GRU	0.0191	0.0090	0.0217	0.0096	0.00235	0.0115
	Informer	0.0223	0.0105	0.0226	0.0106	0.0214	0.0108
	FluxCube	0.0176	0.0085	0.0214	0.0096	0.0221	0.0100

Figure 4: Experimental ablation results for RMSE performance of 52 weeks ahead forecasting.

outperformed the other ablation models in all datasets. Full model w/o P2 and w/o P3, which exclude the diffusion term and seasonal term, respectively, exhibited significantly poorer forecasting performance. Also, Constant showed a reduced forecasting performance, suggesting the effectiveness of the time-varying parameters generated by RNN. Each component FluxCube is useful for modeling online user activity data.

7 CASE STUDY

Here, we describe the analytical results obtained with FluxCube. The result of World#1 has already been presented in Section 1 (i.e., Figure 1 and 2.) Figure 5 (a) and (b) show the FluxCube modeling and forecasting results for US#1 and World#2 tensor data, respectively. The figure contains three results obtained by FluxCube modeling each set of data: the modeling and forecasting results of the tensor data, the latent interactions between keywords for each country obtained from I, and the location clustering results and the flow of influences between each area group obtained from J.

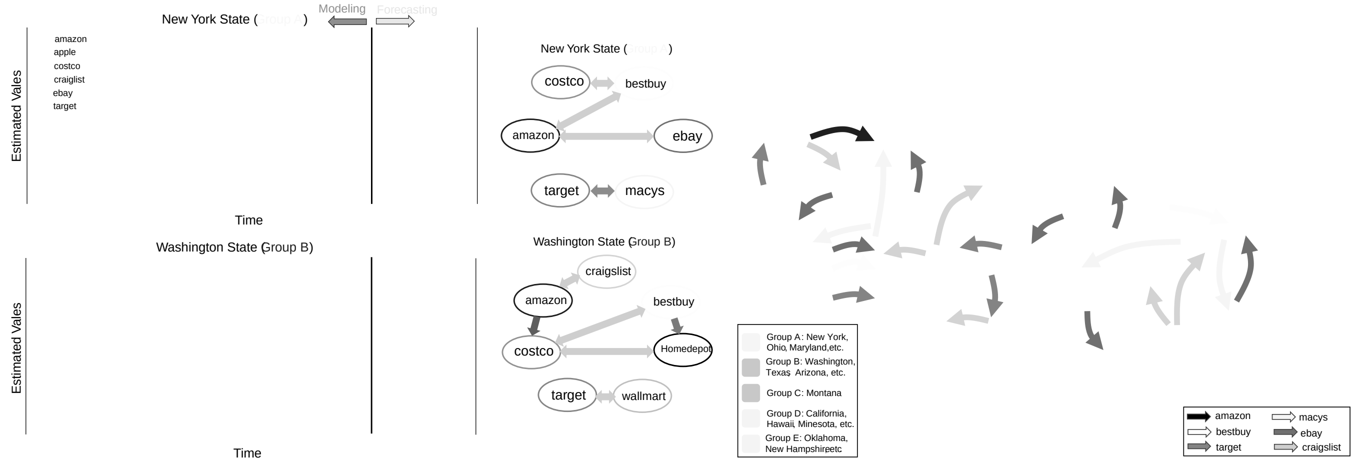
Case study of US#1: Figure 5 (a-1) shows that our proposed model successfully captures and forecasts latent dynamic patterns for multiple countries and keywords. Concretely, our proposed model can capture the seasonality patterns in amazon and apple, as well as the decreasing trends of craigslist. Figure 5 (a-2) based on the reaction term in FluxCube indicates the network graphs summarizing latent interactions between keywords with different colors. For example, in New York State, it shows that amazon competes with bestbuy and ebay, and target and macys are a mutualistic relationship. Figure 5 (a-3) shows the results of clustering each state into multiple groups. The influence of macys from group A, into which FluxCube groups the northeast region of the US, affects other areas.

Case study of World#2: Figure 5 (b-1) shows that FluxCube captures the increasing trend of slack and snapchat in Spain and the decreasing trend of facebook in Japan. On the other hand, it failed to adequately capture the trend of slack, which increased rapidly in popularity in Japan after 2018. As shown in Figure 5 (b-2), FluxCube discovers many commensal and mutualistic relationships between social media platforms: e.g., in Spain, facebook has mutualistic relationships with several platforms such as line, viber, and twitter. We find that many social media tend to be symbiotic rather than compete for online users. Figure 5 (b-3) shows that our model groups countries such as Japan and Saudi Arabia, where twitter is mainly used, into group B and countries such as the United States and the United Kingdom, where facebook is mainly used, into group F. Our results about the influence interactions between area groups show that the influence of facebook flows from group F to other area groups. Also, the influence of viber founded in Israel flows from group E, including countries in the Middle East, to other area groups.

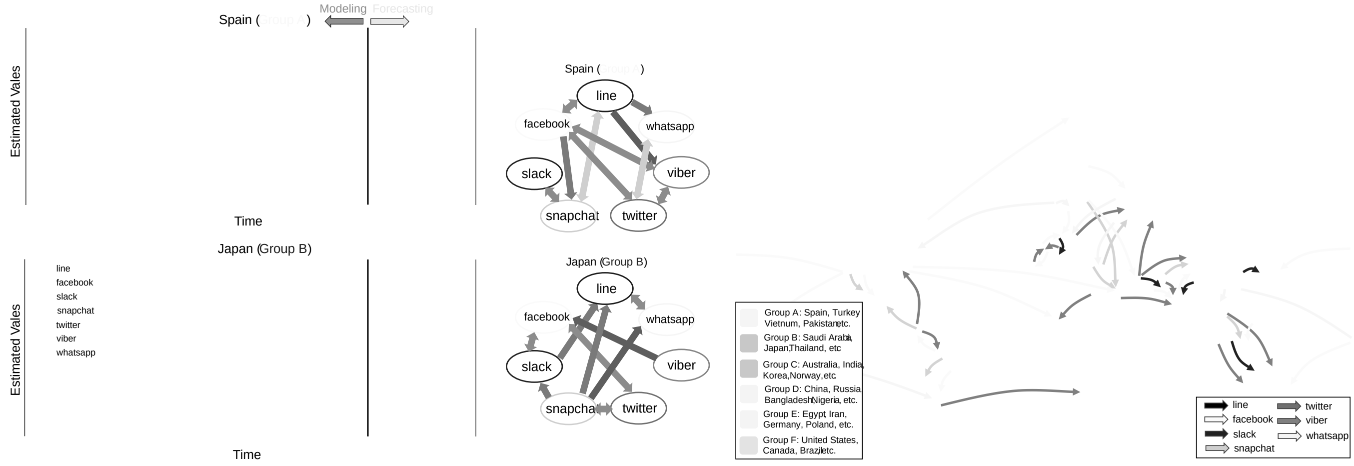
As our case studies show, FluxCube provides the ability to describe complex interactions that can reveal underlying relationships among keywords and locations, and is suitable for modeling and forecasting online user activity data.

8 CONCLUSION

In this study, we attempted to model and forecast large time-evolving online user activity data such as web search volumes. To achieve this, we proposed an effective modeling and forecasting method,



(a–1) Modeling and forecasting results **(a–2) Interactions** **(a–3) Diffusion process of each area group**



(b–1) Modeling and forecasting results **(b–2) Interactions** **(b–3) Diffusion process of each area group**

Figure 5: FluxCube modeling and forecasting results for US#1 (a) and World#2 (b) data: (a/b-1) shows the modeling and forecasting results of FluxCube given the observational data. It represents observational data in gray and the modeling and forecasting results for each keyword in various colors. (a/b-2) shows the network graphs summarizing four latent interactions between keywords in each country. FluxCube uncovers four types of interaction, each of which is represented by arrows. Competing and mutualistic relationships are bidirectional, while commensal and parasitic relationships are one-directional: source species benefit from destination species’ growth. (a/b-3) shows the clustering results with countries/states divided into multiple area groups and the flow of influences between each area group.

namely FluxCube based on reaction-diffusion and ecological systems. Our method can recognize trends, seasonality and interactions in input observations by extracting their latent dynamic systems. Throughout the search volume forecasting experiments in multiple GoogleTrends datasets, the proposed model achieved the highest performance by capturing the latent dynamics.

In addition to its predictive performance, FluxCube provides the latent interactions and the influence flows hidden behind observational data in a human-interpretable form. For example, in Washington State, costco competes with bestbuy and homedepot for online users, as shown by our case studies. Uncovering such latent interactions behind the observational data assists to our decision-making.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research Number JP20H00585, JP21H03446, NICT 21481014, MIC/SCOPE 192107004, JST-AIP JPMJCR21U4, and ERCA-Environment Research and Technology Development Fund JPMEERF20201R02.