Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Tracking Idea Flows between Social Groups

Yangxin Zhong[1], **Shixia Liu**[1*], **Xiting Wang**[1], **Jiannan Xiao**[1] and **Yangqiu Song**[2]
[1]School of Software, Tsinghua University, Beijing, P.R. China
[2]Lane Department of Computer Science and Electrical Engineering, West Virginia University, United States
[1]{zhongyx12,wang-xt11,xjn11}@mails.tsinghua.edu.cn, [1]shixia@tsinghua.edu.cn, [2]yangqiu.song@mail.wvu.edu

Abstract

In many applications, ideas that are described by a set of words often flow between different groups. To facilitate users in analyzing the flow, we present a method to model the flow behaviors that aims at identifying the lead-lag relationships between word clusters of different user groups. In particular, an improved Bayesian conditional cointegration based on dynamic time warping is employed to learn links between words in different groups. A tensor-based technique is developed to cluster these linked words into different clusters (ideas) and track the flow of ideas. The main feature of the tensor representation is that we introduce two additional dimensions to represent both time and lead-lag relationships. Experiments on both synthetic and real datasets show that our method is more effective than methods based on traditional clustering techniques and achieves better accuracy. A case study was conducted to demonstrate the usefulness of our method in helping users understand the flow of ideas between different user groups on social media.

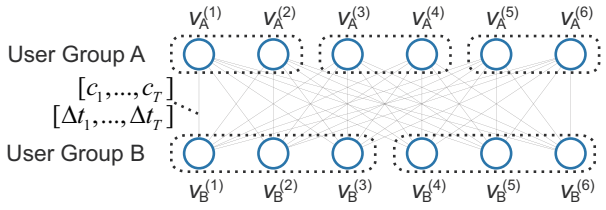
Introduction

As stated in Webster's Third Edition, an idea is "a formulated thought or opinion." Hundreds of millions of users post their ideas on bursty events, hot topics, and personal lives on social media. Users from different social groups tend to interact with each other on different ideas. For example, Democrats and Republicans often interact and communicate on ideas of interest on Twitter. One idea is about the bipartisan senate immigration bill, which can be described as "immigration, bill, act, law, bipartisan." As in this example, a set of words is often used to represent an idea (Blei and Lafferty 2006). Frequently, idea flows between different user groups on social media to reflect social interaction and influence (Pentland 2014). These flows facilitate the transfer of information, opinions, and thoughts from group to group. For instance, idea flows between Democrats and Republicans disclose their leadership and impact on different issues ranging from presidential approval ratings and immigration to health care and women's equality. For the immigration idea, Democrats led the discussion of the idea since they called for this new immigration bill. However, most Republican voters were concerned that an immigration bill might not solve border security problems. As they argued and debated with each other, the immigration idea flowed between Democrats and Republicans. For more details about this example, please refer to our case study in the evaluation section. In many applications, it is desirable to track such an idea flow and its lead-lag relationships between different groups (Liu et al. 2015; Wu et al. 2013).

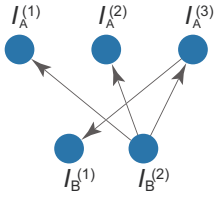
For this reason, the study of idea (influence) propagation has received a great deal of attention (Leskovec, Backstrom, and Kleinberg 2009; Myers, Zhu, and Leskovec 2012; Shaparenko and Joachims 2007; Wu et al. 2014; Yu et al. 2015). Existing methods range from link-based and content-based to a hybrid approach (Nallapati et al. 2008). Although these methods have been successful in analyzing individual users and their connections, as well as the paths the information propagates through, less attention has been paid to studying how the ideas correlate social groups and interact with each other along time.

The goal of our work is to identify an idea as a word cluster and track the lead-lag relationships between word clusters of different user groups. To achieve this goal, we first derive an augmented bipartite word graph based on the correlations and lead-lag relationships between words. Each word is represented by a time series, which encodes its term frequency change over time. Since the correlation between two words can be irregular and arbitrary over time, we provide a way to automatically identify the time period in which two words are correlated. Specifically, we use dynamic time warping (DTW) to align two time series under the monotonic and slope constraint conditions of different time points (Sakoe and Chiba 1978). Then, we employ Bayesian conditional cointegration (BCC) (Bracegirdle and Barber 2012) to discover the local correlation between two time series. After applying BCC, we can determine whether two words have a lead or lag relationship at a particular time point. Consequently, we formulate the augmented bipartite graph as a tensor representation. In contrast to traditional time dependent data analysis using tensor, which employs one additional dimension to represent time information (Sun, Tao, and Faloutsos 2006; Sun et al. 2008), we have introduced two additional dimensions to represent both time and lead-lag relationships. Moreover, we automatically discover ideas that are represented by clusters of words by factorizing the 4-order tensor. For

*S. Liu is the corresponding author.
Copyright © 2016, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



(a) Augmented Bipartite Word Graph



(b) Flow of Ideas

Figure 1: Basic idea of the idea flow model: (a) each edge is represented by a correlation vector $\mathbf{c} = [c_1, ..., c_T]$ and a lead-lag vector $\mathbf{\Delta t} = [\Delta t_1, ..., t_T]$. T is the number of time points and c_k, Δt_k are the correlation value and lead-lag time between two words for the kth time point; (b) words are aggregated into ideas and edges are aggregated into flows.

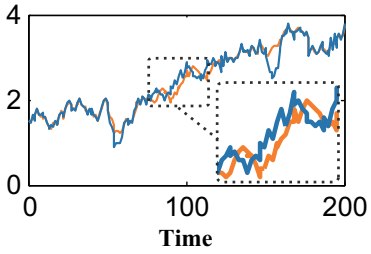
a certain pair of ideas, we further apply tensor factorization to cluster the time points, which can be used to segment the time series to identify the lead and lag period between ideas.

To demonstrate the effectiveness of our approach, two experiments and a case study were conducted. First, we used a synthetic dataset, in which the ground truth about word clusters and their lead-lag information is known, to evaluate how the algorithm performs with different noise levels. Second, we used ten time series benchmark datasets to show that our algorithm improves the time series clustering quality by introducing local lead-lag information. Third, a case study was conducted based on a set of tweets posted by 514 members of the 113th U.S. Congress in 2013. The aim is to demonstrate how the ideas from different social groups interact with each other.

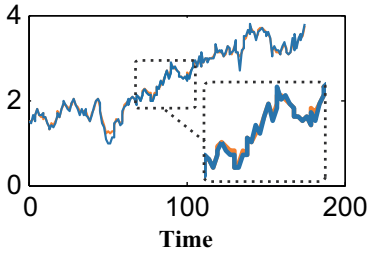
Algorithm Overview

Suppose each word is tracked by a time series that encodes its frequency change over time. The lead and lag relationships between ideas of different user groups can be derived using temporal correlations between words. As shown in Fig. 1, we first extract the correlation and lead-lag relationship between words. An augmented bipartite graph $G = (V, E)$ is used to encode all the correlations and lead-lag relationships between two groups of words. Second, we discover ideas that are represented by clusters of words as well as their flows between different user groups. The steps are detailed as below.

- **Augmented bipartite graph construction.** In the first step, we calculate correlations between words to construct the augmented bipartite graph. Generally, correlations between words can be irregular and arbitrary over time for two reasons: 1) the correlation of two words may change over time; 2) there are lead-lag relationships between the correlated parts (Fig. 2(a)). BCC is able to identify time periods in which two time series are correlated, but it is not able to detect the lead-lag relationships. To solve this problem, we developed an improved BCC algorithm that incorporates DTW to align two time series and detect the lead-lag relationships. With these correlations and lead-lag relationships, an augmented bipartite graph is built, in which $V = V_A \cup V_B$ represents the words that belong to user groups A or B. Each edge in E is represented by two vectors: 1) correlation vector $\mathbf{c} = [c_1, ..., c_T]$, where T is the number of time points and $c_k = 1$ ($c_k = 0$) means $v_A^{(i)}$ is correlated (not correlated) with $v_B^{(j)}$ at the kth time point; 2) lead-lag vector $\mathbf{\Delta t} = [\Delta t_1, ..., \Delta t_T]$, where Δt_k is the lead-lag time between words at time k.
- **Tracking idea flows.** In this step, ideas are derived by partitioning the augmented bipartite graph into word clusters. The key challenge is to partition the augmented bipartite graph, in which each edge is represented by two vectors instead of a real number. To solve this problem, we model the augmented bipartite graph as a tensor, which uses additional dimensions to represent time and lead-lag relationships. We then apply tensor factorization techniques to extract a feature vector for each word and cluster the words based on these feature vectors. According to the clustering results, words are aggregated into ideas and edges are aggregated into flows (Fig. 1(b)). For each pair of ideas, we further identify their lead and lag period by clustering the time points using tensor-based techniques.



(a) Before applying DTW



(b) After applying DTW

Figure 2: An example of applying the DTW alignment.

Augmented Bipartite Graph Construction

This section introduces our algorithm for calculating correlations and lead-lag relationships between words of different groups by combining BCC with DTW. Based on these correlations, we then derive an augmented bipartite word graph. Suppose we have two time series $x_{1:T}$ and $y_{1:T}$ for two words

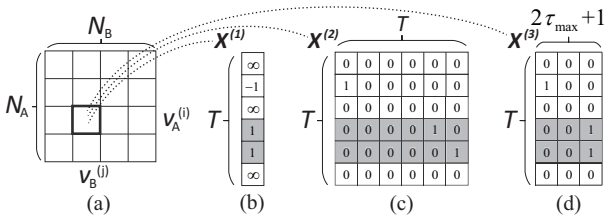


Figure 3: Three tensor representations: (a) word matrix; (b) (c) (d) three representations to encode time and lead-lag relationships.

that belong to different user groups. The goal is to extract the correlation vector $\mathbf{c} = [c_1, ..., c_T]$ and the lead-lag vector $\mathbf{\Delta t} = [\Delta t_1, ..., \Delta t_T]$. In particular, our algorithm consists of the following two steps.

First, DTW is employed to calculate $[\Delta t_1, ..., \Delta t_T]$. DTW is a widely used dynamic programming algorithm that aligns two time series under the monotonic and slope constraint conditions of different time points (Sakoe and Chiba 1978). We choose DTW because it is both flexible and efficient (Sakoe and Chiba 1978). An example of applying DTW is shown in Fig. 2. Before applying DTW, there are misalignments between correlated parts of $x_{1:T}$ and $y_{1:T}$ (Fig. 2(a)). After applying DTW, the misaligned parts are successfully aligned together (Fig. 2(b)). Δt_k is calculated based on this alignment: if x_k is aligned to y_l, we set Δt_k to $l - k$.

Second, we derive $[c_1, ..., c_T]$ by using BCC to examine the cointegration between aligned time series $x'_{1:T}$ and $y'_{1:T}$. Cointegration describes a relationship between time series where there is a stationary linear combination. Specifically, if x'_k and y'_k are cointegrated in a given time period, we have

$$y'_k = \alpha + \beta x'_k + \epsilon_k,$$

where α is a constant and β is the linear regression coefficient. $[\epsilon_1, \epsilon_2, ..., \epsilon_T]$ follows a mean-reverting, stationary process. Compared to typical measures such as the Pearson correlation and Spearman correlation (Bluman 2012), BCC has two advantages (Bracegirdle and Barber 2012).

- BCC is able to calculate both global and local correlations while Pearson correlation and Spearman correlation can only detect global correlations.
- BCC produces less spurious results compared with the classical cointegration testing approach.

After applying BCC, we detect the correlation (cointegration) of $x'_{1:T}$ and $y'_{1:T}$ at different times ($\mathbf{c}'$). We then assign $\mathbf{c}'$ to $\mathbf{c}$ according to the alignment relationships derived by DTW.

Tracking Idea Flows

This section introduces the tensor representation as well as tensor-based augmented bipartite graph partition and aggregation.

Tensor Representation

The key challenge of partitioning the augmented bipartite word graph is that each edge is represented by two vectors rather than a real number (Fig. 1(a)). To tackle this challenge, we model the augmented bipartite word graph as a tensor, which is able to encode time and lead-lag relationships using additional dimensions.

A straightforward tensor representation is $\mathbf{X}^{(1)} \in \mathbb{R}^{N_A \times N_B \times T}$ (Fig. 3(b)). Here N_A and N_B are the numbers of words in user groups A and B, respectively. In our implementation, user groups are already identified by two professors who majors in media and communications. $\mathbf{X}^{(1)}_{ijk}$ represents the lead-lag relationship between the ith word in A and the jth word in B at the kth time point. We set $\mathbf{X}^{(1)}_{ijk}$ to Δt_k if the two words are correlated at the kth time point and ∞ if the two words are not correlated at that time point. However, representing uncorrelated information as ∞ may not be good enough because it can also mean one word correlated with another with an infinite time lead. Moreover, $\mathbf{X}^{(1)}$ is very dense. As a result, computation based on $\mathbf{X}^{(1)}$ is very expensive.

A more reasonable solution is to extend the 3-order tensor to a 4-order representation: $\mathbf{X}^{(2)} \in \mathbb{R}^{N_A \times N_B \times T \times T}$ (Fig. 3(c)). $\mathbf{X}^{(2)}_{ijkl}$ is set to 1 if and only if the ith word in user group A is correlated to the jth word in user group B at the kth time point and $l = k + \Delta t_k$. Otherwise, $\mathbf{X}^{(2)}_{ijkl}$ is set to 0. Compared to $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$ is sparse and it clearly distinguishes between uncorrelated time points and correlated time points. However, $\mathbf{X}^{(2)}$ fails to represent lead-lag time effectively. As shown in Fig. 3, for words $v_A^{(i)}$ and $v_B^{(j)}$, the lead-lag time of the 4th time point and 5th time point are both 1. However, the feature vectors of $\mathbf{X}^{(2)}$ at the 4th time point and the 5th time point (grey rows) are different ($[0, 0, 0, 0, 1, 0]$ and $[0, 0, 0, 0, 0, 1]$). This leads to incorrect lead-lag results along the time dimension.

To solve this issue, we redesign the 4-th dimension of $\mathbf{X}^{(2)}$ to more effectively encode lead-lag time. Our design is $\mathbf{X}^{(3)} \in \mathbb{R}^{N_A \times N_B \times T \times (2\tau_{max}+1)}$ (Fig. 3(d)), where τ_{max} is the maximum allowable time deviation between two aligned time points in DTW. $\mathbf{X}^{(3)}_{ijkl}$ is set to 1 if and only if the ith word in A and the jth word in B are correlated at the kth time point and $l = \Delta t_k + \tau_{max} + 1$. Otherwise, $\mathbf{X}^{(3)}_{ijkl}$ is set to 0. As shown in Fig. 3(d), the feature vectors of $\mathbf{X}^{(3)}$ at 4th and 5th time points are the same ($[0, 0, 1]$). As a result, the 4th time point and 5th time point will be clustered into one segment and we can accurately detect the lead and lag periods.

Augmented Bipartite Graph Partition

Based on the tensor representation, our algorithm first employs tensor-based techniques to extract the feature vector for each word and then clusters the words using these features vectors. The algorithm consists of three steps.

First, we employ a tensor factorization algorithm called greedy PARAFAC (Kolda, Bader, and Kenny 2005) to factorize the tensor. This algorithm is adopted because it is efficient and is able to deal with tensors that have more than three dimensions. The algorithm yields a rank-q approximation of $\mathbf{X}^{(3)}$ in the form

$$\mathbf{X}^{(3)} \approx \sum_{m=1}^{q} \lambda^{(m)} \boldsymbol{u}^{(m)} \circ \boldsymbol{v}^{(m)} \circ \boldsymbol{w}^{(m)} \circ \boldsymbol{h}^{(m)},$$

where $\lambda^{(m)} \in \mathbb{R}$, $\boldsymbol{u}^{(m)} \in \mathbb{R}^{N_A}$, $\boldsymbol{v}^{(m)} \in \mathbb{R}^{N_B}$, $\boldsymbol{w}^{(m)} \in \mathbb{R}^{T}$, $\boldsymbol{h}^{(m)} \in \mathbb{R}^{(2\tau_{max}+1)}$. $\boldsymbol{u}^{(m)} \circ \boldsymbol{v}^{(m)} \circ \boldsymbol{w}^{(m)} \circ \boldsymbol{h}^{(m)}$ is the 4-way outer product so that $(\boldsymbol{u}^{(m)} \circ \boldsymbol{v}^{(m)} \circ \boldsymbol{w}^{(m)} \circ \boldsymbol{h}^{(m)})_{ijkl} = \boldsymbol{u}_i^{(m)} \boldsymbol{v}_j^{(m)} \boldsymbol{w}_k^{(m)} \boldsymbol{h}_l^{(m)}$. Here $\boldsymbol{u}_i^{(m)}$ is the ith entry of vector $\boldsymbol{u}^{(m)}$.

Then we use factors $\{\boldsymbol{u}^{(m)}\}$ and $\{\boldsymbol{v}^{(m)}\}$ to extract feature vectors for each word in groups A and B, respectively. Take user group A as an example. The feature matrix $\boldsymbol{U}$ for this group is constructed as follows:

$$\boldsymbol{U} = [\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, ..., \boldsymbol{u}^{(q)}] \in \mathbb{R}^{N_A \times q}.$$

For the ith word in user group A, the feature vector is set to $[\boldsymbol{U}_{i1}, ..., \boldsymbol{U}_{iq}]$.

Finally, we utilize a clustering algorithm such as K-means, on the extracted feature vectors to detect word clusters.

Augmented Bipartite Graph Aggregation

Based on the clustering results, we aggregate words into ideas and edges into flows (Fig. 1). To identify the lead and lag periods between ideas, we segment the time series for each idea pair by further applying tensor factorization. Take ideas $I_A^{(1)}$ and $I_B^{(2)}$ as an example. First, we extract a sub-tensor from $\mathbf{X}^{(3)}$ that belongs to $I_A^{(1)}$ and $I_B^{(2)}$. Without a loss of generality, we assume that the 1st to N_1th words in A belong to $I_A^{(1)}$ and the 1st to N_2th words in B belong to $I_B^{(2)}$. Then the sub-tensor is denoted as $\mathbf{X}^{(3)}_{1:N_1,1:N_2,1:T,1:(2\tau_{max}+1)} \in \mathbb{R}^{N_1 \times N_2 \times T \times (2\tau_{max}+1)}$. After extracting the sub-tensor, we apply tensor factorization on it to derive a feature vector for each time point and cluster these feature vectors by using a clustering algorithm such as K-means. Based on the clustering results, we divide the time points into segments. Finally, we extract the correlation between $I_A^{(1)}$ and $I_B^{(2)}$ at the kth segment ($\bar{c}_k$). We average c_k for all word pairs at time points that belong to this segment. If this value is greater than a given threshold, $\bar{c}_k = 1$; otherwise, $\bar{c}_k = 0$. If $\bar{c}_k = 1$, the lead-lag time between these two ideas, $\overline{\Delta t}_k$, is computed by averaging Δt_k.

Evaluation

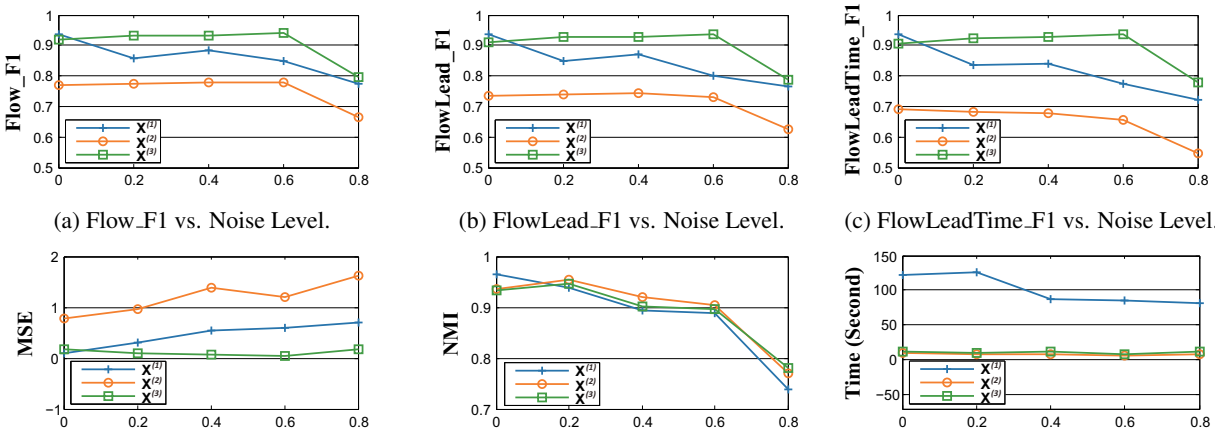
We conducted two experiments and a case study to illustrate the effectiveness and usefulness of our idea flow tracking method. All the experiments were conducted on a workstation with an Intel Xeon E52630 CPU (2.4 GHz) and 64GB of Memory.

Experiment on Synthetic Data

In this experiment, we used a synthetic dataset to show how our algorithm performs with different noise levels. We also compared the accuracy and efficiency of three tensor representations.

Experimental Settings. In this experiment, synthetic datasets with five different noise levels were generated (noise level $L \in \{0, 0.2, ..., 0.8\}$). For each L, we generated 50 synthetic datasets and averaged the results of these datasets. For each dataset, we first generated ideas and the lead-lag relationships between ideas (Fig. 1(b)), which contain $\bar{c}_k$ and $\overline{\Delta t}_k$. Then we generated the augmented bipartite word graph (Fig. 1(a)) according to noise level L. L denotes the probability that $c_k = 0$ when $\bar{c}_k = 1$. The larger the L, the more difficult it is to generate accurate idea flows by using the augmented bipartite word graph. In the synthetic datasets, the number of ideas in each user group varied from 2 to 6, the number of words in each idea varied from 10 to 30, the lead-lag time varied from -6 to 6, and the number of time points in the lead or lag periods varied from 20 to 40, all while T was set to 200. We applied our algorithm with different tensors ($\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, and $\mathbf{X}^{(3)}$) to the datasets and reported the accuracy and efficiency.

Criteria. We evaluated the accuracy of our algorithm based on five criteria. To evaluate the accuracy of our algorithm in detecting lead-lag relationships between ideas, we used three F1-measures: **Flow_F1**, **FlowLead_F1**, and **FlowLeadTime_F1**. Flow_F1 measures the accuracy of our algorithm in estimating $\bar{c}_k$. FlowLead_F1 is more rigorous than Flow_F1. It not only measures the accuracy of $\bar{c}_k$, but also measures the accuracy of our algorithm in detecting



(a) Flow_F1 vs. Noise Level. (b) FlowLead_F1 vs. Noise Level. (c) FlowLeadTime_F1 vs. Noise Level.

(d) Lead-lag Time MSE vs. Noise Level. (e) Word Clustering NMI vs. Noise Level. (f) Running Time vs. Noise Level.

Figure 4: Comparison of three tensors ($\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, and $\mathbf{X}^{(3)}$) in terms of accuracy and efficiency at different noise levels.

the lead or lag status (i.e., whether $\overline{\Delta t}_k > 0$, $\overline{\Delta t}_k = 0$, or $\overline{\Delta t}_k < 0$). FlowLeadTime_F1 is even more rigorous than FlowLead_F1, which measures both $\bar{c}_k$ and $\overline{\Delta t}_k$. A result with larger F1-measures is better. In addition to F1-measures, we used Mean Squared Error (**MSE**) to evaluate the effectiveness of our algorithm in detecting the lead-lag time $\overline{\Delta t}_k$. Smaller MSE values indicate better quality. Moreover, a widely used measure for clustering quality, Normalized Mutual Information (**NMI**) (Strehl and Ghosh 2003), was utilized to measure the accuracy of idea identification. A larger NMI value indicates more accurate idea identification.

Results. Fig. 4 compares the three tensors in terms of accuracy and efficiency at different noise levels. The following observations can be made from the results.

Accuracy. Overall, $\mathbf{X}^{(3)}$ has the best performance in terms of accuracy. This demonstrates that $\mathbf{X}^{(3)}$ is less sensitive to noises compared to $\mathbf{X}^{(1)}$ by using two additional dimensions to encode both c_k and Δt_k. Although $\mathbf{X}^{(2)}$ is comparable to $\mathbf{X}^{(3)}$ in terms of NMI, it is worse in terms of F1-measures and MSE. This indicates that $\mathbf{X}^{(2)}$ is accurate in identifying ideas, but is not very accurate in detecting lead-lag relationships between ideas. This is due to its deficiency in clustering the time points and detecting lead and lag periods.

Efficiency. As shown in Fig. 4(f), $\mathbf{X}^{(1)}$ is much slower than $\mathbf{X}^{(2)}$ and $\mathbf{X}^{(3)}$. This is because $\mathbf{X}^{(2)}$ and $\mathbf{X}^{(3)}$ are much sparser than $\mathbf{X}^{(1)}$. The computation time for $\mathbf{X}^{(2)}$ and $\mathbf{X}^{(3)}$ is comparable. Both of them are relatively efficient with a maximum computation time of 12 seconds.

Because $\mathbf{X}^{(3)}$ is both accurate and efficient, we used this tensor representation in the following experiments.

Experiment on Real Data

In this experiment, we use real-world time series benchmark datasets to demonstrate that our algorithm can improve the quality of time series clustering by incorporating local lead-lag information.

Experimental Settings. The experiment was conducted on ten real-world time series datasets in the UCR archive (Chen et al. 2015), in which the class label of each time series is known. The ten time series datasets include datasets where the shapes of the time series are quite similar as well as datasets where the shapes are different. Table 1 shows the summary statistics of the UCR time series benchmark datasets used in the experiment. NMI was utilized to measure the clustering quality. We performed a grid search on τ_{max} ($\{1, 2, ..., 10\}$) and chose the parameter with the largest NMI value ($\tau_{max} = 6$). To reduce any bias caused by initial cluster assignments, we ran each experiment 100 times with different random seeds and reported the mean and standard deviation.

Baselines. Three baselines were used in this experiment. The first baseline (**B1**) was K-means (Hartigan and Wong 1979) and the second baseline (**B2**) was spectral clustering (Chen et al. 2011). In B1, we utilized y-values of time series to form feature vectors. In B2, we employed K_{DTW} kernel (Marteau and Gibet 2015) to calculate similarities between time series. The third baseline (**B3**) was similar to our algorithm. The difference is that it does not utilize DTW to calculate local lead-lag time. Instead, it calculates a global lead-lag time by moving the time series forward and backward along the time axis to find the time shift that results in the highest correlation.

Results. Table 2 illustrates the results of comparing our tensor-based clustering algorithm with three baselines in terms of NMI. For B3, several experiments were unable to return results within 20 days. These experiments are marked with a "\." As shown in the table, our algorithm performs better than the baselines in all datasets except for one (Fish). The major difference between our algorithm and the baselines is that the baselines did not consider local lead-lag relationships during the clustering process. In particular, B1 and B2 did not employ any lead-lag measures and B3 calculated a global lead-lag time instead of local lead-lag time. This indicates that the clustering quality can be improved by using local lead-lag relationships. Next, we examined why our algorithm is worse than B1 and B2 in the Fish dataset. We found time series with similar shapes but different amplitudes are labelled to be different in this dataset. Since B1 and B2 use absolute y-values of the time series, they are more accurate in distinguishing such time series. Frequently, the time series of words that have similar shapes but different amplitudes are regarded as correlated. As a result, our algorithm is more appropriate to process textual data.

	Coffee	L7	Trace	Fish	OSU Leaf	SC	Strawberry	FA	CC	Wafer
K	2	7	4	7	6	6	2	14	3	2
N	56	143	200	350	442	600	983	2250	4307	7174
T	286	319	275	463	427	60	235	131	166	152

Table 1: Summary statistics of the 10 UCR time series datasets. Here K represents the number of classes, N is the number of time series, and T denotes the number of time points. The dataset names L7, SC, FA, and CC are abbreviations for Lightning-7, Synthetic Control, Face (All), and Chlorine Concentration, respectively.

	Coffee	L7	Trace	Fish	OSU Leaf	SC	Strawberry	FA	CC	Wafer
B1	0.00±0.00	0.43±0.02	0.53±0.02	0.29±0.03	0.22±0.02	0.78±0.03	0.12±0.00	0.37±0.02	0.01±0.00	0.00±0.00
B2	0.54±0.00	0.08±0.00	0.45±0.00	**0.53±0.02**	0.02±0.00	0.59±0.01	0.13±0.00	0.24±0.00	0.01±0.00	0.04±0.00
B3	0.48±0.00	0.22±0.03	0.54±0.00	0.12±0.00	0.13±0.00	0.46±0.06	\	\	\	\
Ours	**0.64±0.00**	**0.44±0.02**	**0.64±0.00**	0.16±0.01	**0.26±0.00**	**0.80±0.03**	**0.32±0.00**	**0.39±0.09**	**0.02±0.00**	**0.19±0.00**

Table 2: Comparison of our tensor-based clustering method with three baselines in terms of NMI on 10 datasets of the UCR time series benchmark datasets.

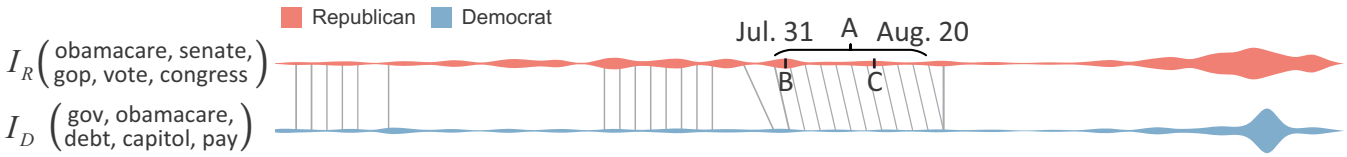


Figure 5: Local lead-lag relationships between ideas in different user groups.

1	immigration, bill, act, law, bipartisan
2	military, join, service, honor, nation
3	women, american, succeed, job, equality

Table 3: Top three ideas led by Democrats.

1	tax, economy, reform, family, icymi
2	job, plan, student, school, program
3	obamacare, senate, gop, vote, congress

Table 4: Top three ideas led by Republicans.

Case Study on Twitter Data

To demonstrate the usefulness of our algorithm in detecting idea flows, we conducted a case study with a professor who majors in media and communications. The dataset contains 156,501 tweets posted by 514 members of the 113th U.S. Congress from Apr. 9 to Oct. 13, 2013. The professor and another professor who also majors in media and communications manually divided these members into two groups: 243 Democrats and 271 Republicans. We segmented the data into 94 time points (every 2 days). Stopwords and rare words that occur less than 5 times each day by average were removed to reduce noise. We extracted 15 ideas for each group.

First, we checked the global lead-lag relationships between all ideas. We found there were 119 lead-lag relationships on pairs of ideas. Among these pairs, 62 (52%) were led by Democrats and 57 (48%) were led by Republicans. This result demonstrated that Democrats marginally outperformed Republicans in taking the lead at that time. The professor confirmed our findings. He said President Obama (Democratic) started to realize the political importance of social media after he was re-elected as President in 2012. After that, Democrats put more effort into social media. Because the president was a Democrat and the Democratic Party had a Senate majority, Democrats had an advantage over Republicans in leading public opinion.

We next examined the ideas that were led by Democrats and Republicans, respectively. We ranked the ideas by the number of time points at which they led other ideas.

The top three ideas for Democrats are shown in Table 3. The professor immediately identified that they were the bipartisan senate immigration bill, honoring military service members, and women's equality. By checking related tweets, the professor found Democrats took the lead on these ideas because they were more supportive of them. Take the "bipartisan senate immigration bill" idea as an example. Because Democrats supported this bill, they kept on posting the latest information about this bill and hosted new activities ("House leaders should allow debate & vote on Bipartisan Senate Immigration Bill. It's not perfect but we need reform. http://t.co/btKDxblbQf"). Republicans lagged because they mainly responded by disapproving of Democrats' ideas. One major concern was that the immigration bill might not solve border security problems ("The immigration bill offers false promises about border security and enforcement measures."). The professor commented that this kind of interaction is common between Democrats and Republicans.

Next, we checked the ideas led by Republicans. As shown in Table 4, the top three were tax reform, Republican's plan to create jobs, and health care vote in Congress (I_R). While Republicans were supportive of tax reform and plans to create jobs, they did not support Obama's health care plan (Obamacare). To understand why I_R was led by Republicans, we examined the correlated ideas from Democrats. Among them, the idea with the largest lag-time (I_D) was exemplified by words "gov," "obamacare," "debt," "capitol," and "pay," which all are related to Obamacare. To know why I_R led I_D, we further checked the local lead-lag relationships between the two ideas. As shown in Fig. 5, each idea is represented by a colored stripe while the x-axis represents time. The width of a stripe changes over time, encoding the temporal "hotness" of this idea. The "hotness" is measured by the number of words that talks about this idea at each time. The local lead-lag relationships are visualized by the links that connect correlated time points. Fig. 5 shows that the main lead period for idea I_R is from Jul. 31 to Aug. 20 (Fig. 5A). By checking the two peaks (Fig. 5B and Fig. 5C) on the stripe during this time period, we found Republicans led this idea because they were very active in opposing the health care plan. On Aug. 2 (Fig. 5B), they voted to repeal Obamacare for the 40th time. Around Aug. 13 (Fig. 5C), they posted many tweets criticizing the fact that a key consumer protection in the health care plan was delayed ("Yet another delay for Obamacare via @CBSNews - Key consumer protection in Obamacare delayed http://t.co/0brYxzjUj7").

Related Work

Our work is relevant to influence analysis between documents. Previous research can be categorized into three groups: content-based, link-based, and hybrid methods.

Content-based methods use the content of documents and their temporal changes to analyze influence (Cui et al. 2011b; Liu et al. 2012; Shaparenko and Joachims 2007; Zhang et al. 2010). Shaparenko and Joachims assumed the language model of a new document is a mixture of the language model of all previously published documents (Shaparenko and Joachims 2007). They adopted a likelihood ratio test

to detect the influence between documents. However, their method cannot model influence at the word level. To learn the influence at the word/phrase level, MemeTracker (Leskovec, Backstrom, and Kleinberg 2009) employed a graph-based method to cluster variants of phrases and derive corresponding threads in the news cycle. To link two given documents with a coherent chronological chain, Shahaf and Guestrin (Shahaf and Guestrin 2010) employed a bipartite graph between words and documents and utilized the graph to model the influence between two documents. Compared with this work, our method builds an augmented bipartite graph between words used by two user groups by employing the improved BCC technique. The bipartite link is represented by a set of correlation values and lead-lag times between two words at different times instead of a single weight. As a result, we can model the influence between groups from a global overview to local temporal details.

Link-based methods use links (e.g., citations or follower-followee relationships) in a network to analyze influence. PageRank (Brin and Page 1998) and its variations (Haveliwala 2002; Ma, Guan, and Zhao 2008) are typical examples of such methods. Researchers also studied the problem of influence maximization in a network (Feng et al. 2014; Kempe, Kleinberg, and Tardos 2003; Ohsaka et al. 2014). However, link-based methods need to utilize explicit links in the data to model influence. Such kind of links may not be available in many real-world datasets such as news articles.

To solve this issue, researchers have developed hybrid methods that use both links and content to analyze influence (Cui et al. 2011a; El-Arini and Guestrin 2011; Gomez Rodriguez, Leskovec, and Krause 2010). For example, given the topics discussed in a network, Tang et al. (Tang et al. 2009) constructed a factor graph to compute the topic level influence in a large network. To model influence as well as topics in a corpus, researchers have proposed several topic models that incorporate links between documents (Dietz, Bickel, and Scheffer 2007; Erosheva, Fienberg, and Lafferty 2004; Gerrish and Blei 2010; Guo et al. 2010; 2014; Liu et al. 2010; Nallapati et al. 2008; Nallapati and Cohen 2008). Influence can also come from outside the network. Myers et al. (Myers, Zhu, and Leskovec 2012) developed an information diffusion model to integrate both the internal influence and the external influence in a network.

The major issue of hybrid methods is that they model influence at the document, user, or topic level. In contrast, our method aims at detecting ideas at the word level and studying how the ideas correlate social groups and interact with each other along time.

Conclusions and Future Work

In this paper, we have aimed to help users understand how ideas propagate between different groups. To this end, we first derive an augmented bipartite word graph based on the correlations and lead-lag relationships between words. The major feature of the augmented bipartite graph is that its edge is represented by two vectors: a correlation vector and a lead-lag vector. The two vectors, along with the augmented bipartite graph, are formulated as a tensor representation. Next, by factorizing the tensor, we automatically discover ideas that are represented by clusters of words. For a certain pair of ideas, we further apply tensor factorization to cluster the time points, which can be used to segment the time series to identify the lead and lag period between ideas. Finally, our evaluation demonstrates that the developed approach is generally more effective than the baseline methods.

One interesting direction for future work is exploring a wider range of text data representations for better tracking idea flows on social media. Text data is typically treated as bag-of-words. In this paper, we follow this paradigm since this representation is simple and easy to be understood. There are some advanced techniques that can convert the unstructured text to a network with typed entity and relation information (Wang et al. 2015a; 2015b; 2016), which can be less ambiguous than only using bag-of-words. Integrating these techniques with our model is a very interesting topic for further study in the future. In addition, we would like to study the lead-lag relationships within each user group and combine them with the lead-lag relationships between different user groups for better modeling idea flows on social media.

Acknowledgements

We would like to thank Mengchen Liu for valuable contributions on related work and the model formulation. This research was supported by the National Key Technologies R&D Program of China (2015BAF23B03) and a Microsoft Research Fund (No. FY15-RES-OPP-112).

References

Blei, D. M., and Lafferty, J. D. 2006. Dynamic topic models. In *ICML*, 113–120.

Bluman, A. G. 2012. *Elementary statistics: A step by step approach*. McGraw-Hill.

Bracegirdle, C., and Barber, D. 2012. Bayesian conditional cointegration. In *ICML*, 1095–1102.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1):107–117.

Chen, W.-Y.; Song, Y.; Bai, H.; Lin, C.-J.; and Chang, E. Y. 2011. Parallel spectral clustering in distributed systems. *PAMI* 33(3):568–586.

Chen, Y.; Keogh, E.; Hu, B.; Begum, N.; Bagnall, A.; Mueen, A.; and Batista, G. 2015. The ucr time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/.

Cui, P.; Wang, F.; Yang, S.; and Sun, L. 2011a. Item-level social influence prediction with probabilistic hybrid factor matrix factorization. In *AAAI*.

Cui, W.; Liu, S.; Tan, L.; Shi, C.; Song, Y.; Gao, Z.; Qu, H.; and Tong, X. 2011b. Textflow: Towards better understanding of evolving topics in text. *TVCG* 17(12):2412–2421.

Dietz, L.; Bickel, S.; and Scheffer, T. 2007. Unsupervised prediction of citation influences. In *ICML*, 233–240.

El-Arini, K., and Guestrin, C. 2011. Beyond keyword search: Discovering relevant scientific literature. In *KDD*, 439–447.

Erosheva, E.; Fienberg, S.; and Lafferty, J. 2004. Mixed-membership models of scientific publications. *PNAS* 101(suppl 1):5220–5227.

Feng, S.; Chen, X.; Cong, G.; Zeng, Y.; Chee, Y. M.; and Xiang, Y. 2014. Influence maximization with novelty decay in social networks. In *AAAI*.

Gerrish, S., and Blei, D. M. 2010. A language-based approach to measuring scholarly impact. In *ICML*, 375–382.

Gomez Rodriguez, M.; Leskovec, J.; and Krause, A. 2010. Inferring networks of diffusion and influence. In *KDD*, 1019–1028.

Guo, Z.; Zhu, S.; Zhang, Z.; Chi, Y.; and Gong, Y. 2010. A topic model for linked documents and update rules for its estimation. In *AAAI*.

Guo, Z.; Zhang, Z. M.; Zhu, S.; Chi, Y.; and Gong, Y. 2014. A two-level topic model towards knowledge discovery from citation networks. *TKDE* 26(4):780–794.

Hartigan, J. A., and Wong, M. A. 1979. Algorithm as 136: A k-means clustering algorithm. *Applied statistics* 100–108.

Haveliwala, T. H. 2002. Topic-sensitive pagerank. In *WWW*, 517–526.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *KDD*, 137–146.

Kolda, T.; Bader, B.; and Kenny, J. 2005. Higher-order web link analysis using multilinear algebra. In *ICDM*, 242–249.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Meme-tracking and the dynamics of the news cycle. In *KDD*, 497–506.

Liu, L.; Tang, J.; Han, J.; Jiang, M.; and Yang, S. 2010. Mining topic-level influence in heterogeneous networks. In *CIKM*, 199–208.

Liu, S.; Zhou, M. X.; Pan, S.; Song, Y.; Qian, W.; Cai, W.; and Lian, X. 2012. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM TIST* 3(2).

Liu, S.; Chen, Y.; Wei, H.; Yang, J.; Zhou, K.; and Drucker, S. M. 2015. Exploring topical lead-lag across corpora. *IEEE TKDE* 27(1):115–129.

Ma, N.; Guan, J.; and Zhao, Y. 2008. Bringing pagerank to the citation analysis. *Information Processing & Management* 44(2):800–810.

Marteau, P.-F., and Gibet, S. 2015. On recursive edit distance kernels with application to time series classification. *TNNLS* 26(6):1121–1133.

Myers, S. A.; Zhu, C.; and Leskovec, J. 2012. Information diffusion and external influence in networks. In *KDD*, 33–41.

Nallapati, R. M., and Cohen, W. W. 2008. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In *ICWSM*.

Nallapati, R. M.; Ahmed, A.; Xing, E. P.; and Cohen, W. W. 2008. Joint latent topic models for text and citations. In *KDD*, 542–550.

Ohsaka, N.; Akiba, T.; Yoshida, Y.; and Kawarabayashi, K.-i. 2014. Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In *AAAI*.

Pentland, A. P. 2014. *Social Physics: How Good Ideas Spread-The Lessons from a New Science*. Penguin Press.

Sakoe, H., and Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing* 26:43–49.

Shahaf, D., and Guestrin, C. 2010. Connecting the dots between news articles. In *KDD*, 623–632.

Shaparenko, B., and Joachims, T. 2007. Information genealogy: Uncovering the flow of ideas in non-hyperlinked document databases. In *KDD*, 619–628.

Strehl, A., and Ghosh, J. 2003. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *JMLR* 3:583–617.

Sun, J.; Tao, D.; Papadimitriou, S.; Yu, P. S.; and Faloutsos, C. 2008. Incremental tensor analysis: Theory and applications. *TKDD* 2(3).

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

Tang, J.; Sun, J.; Wang, C.; and Yang, Z. 2009. Social influence analysis in large-scale networks. In *KDD*, 807–816.

Wang, C.; Song, Y.; El-Kishky, A.; Roth, D.; Zhang, M.; and Han, J. 2015a. Incorporating world knowledge to document clustering via heterogeneous information networks. In *KDD*, 1215–1224.

Wang, C.; Song, Y.; Li, H.; Zhang, M.; and Han, J. 2015b. Knowsim: A document similarity measure on structured heterogeneous information networks. In *ICDM*.

Wang, C.; Song, Y.; Li, H.; Zhang, M.; and Han, J. 2016. Text classification with heterogeneous information network kernels. In *AAAI*.

Wu, F.; Song, Y.; Liu, S.; Huang, Y.; and Liu, Z. 2013. Lead-lag analysis via sparse co-projection in correlated text streams. In *CIKM*, 2069–2078.

Wu, Y.; Liu, S.; Yan, K.; Liu, M.; and Wu, F. 2014. Opinionflow: Visual analysis of opinion diffusion on social media. *IEEE TVCG* 20(12):1763–1772.

Yu, L.; Cui, P.; Wang, F.; Song, C.; and Yang, S. 2015. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics. *arXiv preprint arXiv:1505.07193*.

Zhang, J.; Song, Y.; Zhang, C.; and Liu, S. 2010. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *KDD*, 1079–1088.