Uncovering the Spatio-Temporal Dynamics of Memes in the Presence of Incomplete Information

Hancheng Ge[1], James Caverlee[1], Nan Zhang[2], Anna Squicciarini[3]
[1]Department of Computer Science and Engineering, Texas A&M University, College Station, TX
[2]School of Data Science, Fudan University, Shanghai, China
[3]College of Information Sciences and Technology, Pennsylvania State University, University Park, PA,
[1]{hge,caverlee}@cse.tamu.com, [2]zhangnan@fudan.edu.cn,
[3]asquicciarini@ist.psu.edu

ABSTRACT

Modeling, understanding, and predicting the spatio-temporal dynamics of online memes are important tasks, with ramifications on location-based services, social media search, targeted advertising and content delivery networks. However, the raw data revealing these dynamics are often incomplete and error-prone; for example, API limitations and data sampling policies can lead to an incomplete (and often biased) perspective on these dynamics. Hence, in this paper, we investigate new methods for uncovering the full (underlying) distribution through a novel spatio-temporal dynamics recovery framework which models the latent relationships among locations, memes, and times. By integrating these hidden relationships into a tensor-based recovery framework – called AirCP – we find that high-quality models of meme spread can be built with access to only a fraction of the full data. Experimental results on both synthetic and real-world Twitter hashtag data demonstrate the promising performance of the proposed framework: an average improvement of over 27% in recovering the spatio-temporal dynamics of hashtags versus five state-of-the-art alternatives.

1. INTRODUCTION

With the rise of mobile social media services, we are witnessing more and more GPS-enabled sharing of videos, images, blogs, and tweets that provide valuable information regarding "who", "where", "when" and "what". For instance, many mobile image sharing services such as Instagram allow users to attach their latitude-longitude coordinates to shared photographs; location sharing services such as Foursquare and Glimpse enable billions of "check-ins"; and Twitter users generate millions of geo-tagged tweets per day. In turn, these fine-grained spatio-temporal logs of user activities promise new research opportunities to uncover models of user behavior, mobility, and information sharing. Already, there have been efforts to improve location-based recommendations, targeted advertising, social media search, and event detection [5, 6, 15, 22, 29].

However, the raw data revealing these dynamics are often restricted to proprietary data warehouses (e.g., requiring privileged access to Instagram's backend photo serving services), and so researchers and practitioners typically must rely on sampling-based methods to build spatio-temporal models of user behavior. Of course, this sampling faces its own challenges – including API limitations and data sampling policies that can lead to an incomplete (and often biased) perspective on the underlying dynamics. For instance, Morstatter et al. [24] found significant differences in the quality and composition of sampled Twitter data by comparing different sampling policies over the streaming API and Twitter's Firehose. Moreover, changes to data access policies can lead to additional challenges – as demonstrated by Twitter's closing of their Firehose API in April 2015. Additionally, even a robust data sampling approach can still face errors due to missing data and errors in the data collection process. This missing data raises serious concerns. For example, Kossinets [18] found that missing data in a social network can significantly impact the estimation of structural properties of the network. Similarly, Sadikov et al. [28] pointed out that incomplete data may lead to critically different properties of information cascades in a social network. As a result, models based on mobile social media traces may be of limited usefulness and generalizability in the presence of incomplete data traces.

Hence, in this paper we explore new scalable methods for recovering the spatio-temporal dynamics of online memes – like shared images, hyperlinks, videos, or hashtags – in the presence of incomplete information. Concretely, we propose a novel tensor-based factorization approach to recover the spatio-temporal dynamics of memes. The core insight of the proposed method is to carefully take into account the latent relationships among locations, memes, and times; these relationships can then be embedded into a tensor completion framework for uncovering the approximate complete data based only on partial observations. We explore how to model and integrate this auxiliary information – here, in the form of relationships among locations, memes, and times – and show how the underlying tensor completion can be efficiently solved compared to many existing methods.

Through this proposed spatio-temporal dynamics framework – called **AirCP** that stands for **A**uxiliary **I**nformation **R**egularized **C**ANDECOMP/**P**ARAFAC completion. In Table 1, we provide an overview of the state of the art. In short, AirCP reigns, combining capability of leveraging heterogenous information as well as time efficiency, which is more feasible towards "big data". We explore research questions like: Based on an inherently limited sample, can we recover the underlying distribution of memes at a particular location? And at a particular time? What impact does the amount of sampled data have on the quality of this recovery? For example, can we build a high-quality model of meme spread with access to only 20% of actual data? Towards tackling these and related questions, the main contributions of this paper are as follows:

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983782

Table 1: Comparison between AirCP and the state-of-the-art.

	AirCP	TFAI[25]	TNCP[21]	LRCO[32]
Model				
Tensor	✓	✓	✓	✓
Coupled Tensor-Matrix	✓	✓		
Obj. Function				
Tensor Completion	✓		✓	✓
CP	✓	✓	✓	
Tucker				✓
Auxiliary Info.				
Heterogeneous Info.	✓	✓		
Regularization				
Laplacian	✓	✓		
Tikhonov	✓			
Trace Norm			✓	
Opt. Method				
ADMM	✓		✓	✓
Alternating Least Square		✓		

- First, we formally define the problem of recovering the spatio-temporal dynamics of online memes by leveraging the latent relationships among memes, locations, and times, and develop approaches for modeling these latent relationships.
- Second, we propose a novel framework for recovering spatio-temporal dynamics based on the CP tensor completion model with regularized trace of the auxiliary information from memes, locations, and times, as well as Tikhonov regularization.
- Third, we present an efficient algorithm based on the alternative direction method of multipliers (ADMM) to solve the proposed problem using less computation time than existing methods.
- Finally, we empirically evaluate the proposed framework on both synthetic and real-world Twitter hashtag datasets. We find that the proposed method achieves an average over 27% improvement in recovering missing hashtags versus state-of-the-art alternatives, while achieving significantly greater efficiency.

2. RELATED WORK

Spatio-Temporal Dynamics of Online Memes. The increasingly mobile aspects of social media services like Instagram, Facebook, and Twitter have led to a number of studies on geo-spatial characteristics of users and information sharing. For example, researchers have built models of geo-spatial properties to infer geographic information from tweets, such as spatial modeling to geolocate objects [6] and predicting user locations [5]. Other researchers have analyzed the geo-spatial properties of online memes on Facebook [1] and on YouTube based on propagation patterns [3]. On the other hand, much effort has focused on the temporal properties of online memes. Yang et al. [33] studied temporal patterns of online content including Twitter hashtags and online phrases. Matsubara et al. [23] explored temporal patterns of online information diffusion. Other researchers have focused on both spatial and temporal properties of online memes, like [15].

Estimating Missing Spatio-Temporal Data. Toward recovering missing spatio-temporal data, there have been many proposed methods adopting techniques like multivariate interpolation [30], spectrum analysis [17], and matrix factorization [4, 13]. These methods have shown good success, but typically assume a simple inter-dependence among variables of interest (e.g., memes), space, and time, resulting in a challenge to handling correlations (and complex inter-dependencies) among these different factors. In contrast, we investigate in this paper a tensor-based approach that integrates latent relationships among memes, locations, and times.

Compared to matrix factorization methods – which focus on two-way data, not multi-way data sets – tensors, as a generalization of matrices, can naturally model higher-order relationships among entities (i.e. more than two dimensions). In recent years, tensor factorization models have been studied and applied in several fields since tensors are well-suited for multi-way data analysis. There are two widely used low-rank decompositions of tensors, the CANDECOMP/PARAFAC (CP) and the Tucker decompositions [16]. Tensor completion is used to estimate missing values in tensors based on their low-rank approximations, which has been extensively studied and employed in applications such as recommendations [11, 27], user group detection [21], and link prediction [7]. However, most of these approaches focus on solving the tensor completion problem by utilizing the sampled data without considering any auxiliary information. In these cases, the recovery accuracy tends to be worse when only observing limited entries [25].

Though several researchers incorporate auxiliary (external) information into the matrix factorization problem [9, 12], few studies explore the tensor completion problem with auxiliary information. Technically, it is challenging to embed auxiliary information into a factorization model, especially with many heterogeneous contexts. Bahadori et al. [2] proposed a unified low rank tensor learning framework on spatio-temporal data, under which either spatial or temporal information can be modeled, respectively. Yet, how to leverage both spatial and temporal information simultaneously was not investigated in their study. Narita et al. [25] integrated side information into tensor decomposition methods, resulting in better performance compared with ordinary tensor decomposition methods. Nevertheless, they primarily focus on general tensor decomposition with auxiliary information, but not tensor completion. Zhou et al. [34] developed a Tucker-based tensor model called the spatio-temporal tensor completion to infer missing Internet traffic data by integrating spatio-temporal constraint information as within-mode regularization. However, these models usually face some efficiency challenges since [25] requires solving the Sylvester equation with a high cost several times in each of iterations, and [34] strongly relies on solving large-scale least square problems, making them infeasible for large-scale applications. In contrast, the proposed method in this paper seeks to overcome these challenges by developing an efficient tensor-based method that integrates latent relationships from memes, locations, and times simultaneously. Additionally, the proposed approach not only inherits advantages of efficiency based on the alternating direction method of multipliers (ADMM) [10] and uniqueness of solutions enhancing the robustness, but also leads to better recovery by incorporating this auxiliary information.

3. PROBLEM STATEMENT

We assume that there exists a set of geo-temporal tagged online memes H. A meme in this case could correspond to a shared image, a hyperlink, video, or hashtag, among many other possibilities. Each meme $h \in H$ can be expressed as a tuple (h, l, t) where l is the location in which the meme is posted and t is the time at which the meme was posted. Suppose we have N unique geo-temporal tagged memes, L locations, M different timestamps and occurrences of memes O. Let $o_{lt}^{h} \in O$ be the number of occurrences of a meme h in a location l at a timestamp t. We view the spatio-temporal dynamics of geo-temporal tagged memes as a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N \times L \times M}$, in which $\boldsymbol{\mathcal{X}}(i, j, k)$ represents the count of a meme h_i in a location l_j at a timestamp t_k.[1] Due to sampling

[1]We adopt the notation of Kolda et al. [16]. A tensor is a multi-dimensional array. Formally, we can represent an N-way or Nth-order tensor as $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ where $I_i (1 \leq i \leq N)$ is the dimensionality of ith mode. Scalars are denoted by lower-case letters such as i, j, k; matrices are denoted by upper-case letters such

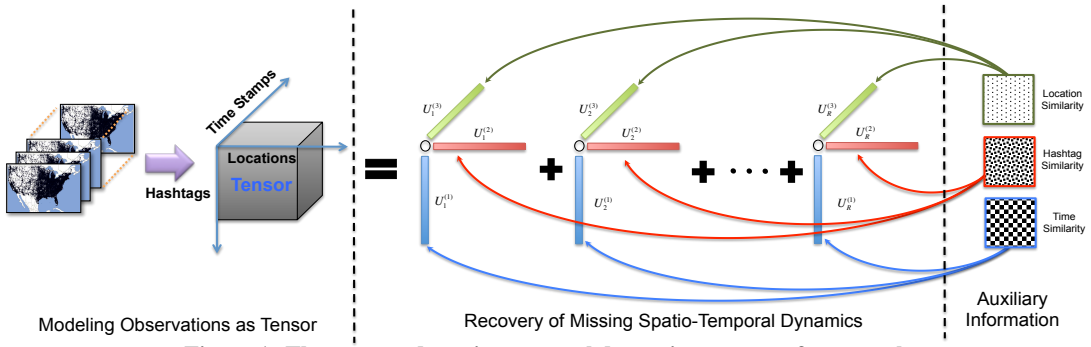


Figure 1: The proposed spatio-temporal dynamics recovery framework.

errors, corrupted data, or other external factors, we further assume that we only can observe parts of the complete dynamics $\boldsymbol{\mathcal{X}}$; we denote this partially observed tensor as $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{N \times L \times M}$, in which some elements are missing or unobservable.

Spatio-Temporal Dynamics Recovery Problem. Given a set of geo-temporal tagged memes H with only partial knowledge of their dynamics – denoted as the tensor $\boldsymbol{\mathcal{T}}$ – our goal is to learn a model to recover the missing spatio-temporal dynamics of the unobserved memes. But what entries in the partial tensor $\boldsymbol{\mathcal{T}}$ are actually missing? We investigate three common situations:

- *Scenario 1: Random Missing Observations.* This first scenario captures the straightforward case of random corruption or random data sampling errors in the dataset. We assume that some fraction of memes – that is, some of the meme, location, time counts (h_i, l_i, t_i) – are missing, and so our task is to estimate these missing counts based on the observations we do have.
- *Scenario 2: Missing Entire Memes at Some Locations.* The second scenario models the case when the data collected has some systematic errors; specifically, we assume that rather than random missing observations (as the scenario 1), there are some memes that are completely missing for some locations. For example, a data sampling strategy may target the top-k memes at a location, so some locations will be missing memes outside of this top-k. Can we recover the missing counts for these lost memes?
- *Scenario 3: Missing Entire Locations.* The final scenario corresponds to the case of total data loss for some locations. For example, a data sampling strategy may target some locations exclusively, but miss others entirely. Can we recover what memes did occur in those missing locations?

Twitter Hashtags. We ground our discussion in the rest of the paper in terms of Twitter hashtags. A Twitter hashtag is a popular type of online meme that arises on Twitter, spreads from person to person (and from place to place), resulting in a fine-grained spatio-temporal log of information sharing dynamics. Note that the methods presented here may be applied to any other dataset with meme, location, time characteristics.

4. AIRCP: AUXILIARY INFORMATION REGULARIZED CP MODEL

In this section, we propose new scalable methods for recovering the spatio-temporal dynamics of online memes. Concretely, we propose to (i) model and exploit the latent relationships among locations, memes and times; (ii) embed these latent relationships into a tensor completion framework for uncovering the approximate complete data based only on partial observations; and (iii) show how the underlying tensor factorization can be efficiently solved compared to many existing methods. The high-level outline of the proposed solution is presented in Figure 1. We model the observed data as a tensor (left), and seek to recover the missing spatio-temporal dynamics (center) by integrating auxiliary information like the relationships between locations, memes and times (right). A key aspect of the proposed approach is an iterative method to overcome the problem of incomplete auxiliary information. In the following, we introduce each part of the proposed solution in detail.

4.1 Modeling Recovery of Missing Data

We propose to model the recovery of missing hashtag data based on tensor models. Since hashtags are usually adopted in a few locations within limited life-spans [15], resulting in $\boldsymbol{\mathcal{X}}$ being sparse and low-rank, this model is built on a CP tensor completion model which can be represented by the following optimization problem:[2]

$$\underset{U^{(n)}, \boldsymbol{\mathcal{X}}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\mathcal{X}} - [\![U^{(1)}, U^{(2)}, U^{(3)}]\!]\|_F^2 + \frac{\lambda}{2}\sum_{n=1}^{3}\|U^{(n)}\|_F^2$$

$$\text{subject to} \quad \boldsymbol{\Omega} * \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{T}}, U^{(n)} \geq 0, n = 1, 2, 3.,$$

where recall that $\boldsymbol{\mathcal{X}}$ denotes the complete spatio-temporal dynamics of hashtags, $\boldsymbol{\mathcal{T}}$ denotes the observations we do have, $U^{(1)} \in \mathbb{R}^{N \times R}$, $U^{(2)} \in \mathbb{R}^{L \times R}$, and $U^{(3)} \in \mathbb{R}^{M \times R}$ are latent factor matrices for location, hashtag, and time dimensions, respectively, $R \ll min(N, L, M)$ is the number of latent factors as the rank of a tensor, $\frac{\lambda}{2}\sum_{n=1}^{N}\|U^{(n)}\|_F^2$ is a Tikhonov regularization term used to avoid overfitting and provide a unique solution, and $\boldsymbol{\Omega}$ is a non-negative weight tensor with the same size as $\boldsymbol{\mathcal{X}}$:

$$\boldsymbol{\Omega}(i,j,k) = \begin{cases} 1 & \text{if } \boldsymbol{\mathcal{X}}(i,j,k) \text{ is observed,} \\ 0 & \text{if } \boldsymbol{\mathcal{X}}(i,j,k) \text{ is unobserved.} \end{cases}$$

Our goal is to seek an estimated $\boldsymbol{\mathcal{X}}$ for recovering the missing spatio-temporal dynamics of hashtags based upon the partial data we do observe. However, as it is the case in many linear-inverse problems, there may not be sufficient information to recover $\boldsymbol{\mathcal{X}}$ only depending on the observed data. We call these *deficient linear-inverse* problems. Apparently, the case of recovering the spatio-temporal dynamics of hashtags with only observed data is a deficient linear-inverse problem, e.g., it is very difficult to estimate occurrences for the hashtag *#iphone* in San Francisco even if we know the complete dynamics of hashtags in other cities such as New York, Austin, and Los Angeles. Hence, our intuition is to leverage the spatio-hashtag-temporal relationships inherent in the observed data in order to successfully recover the missing information. For instance, if knowing that people in San Francisco tend

as X; entries in a tensor (e.g., a 3rd order tensor) is denoted by the original letters with indices such as $\boldsymbol{\mathcal{X}}(i,j,k)$. The order of a tensor is the number of dimensions known as modes N.

[2]The subscript F here and throughout the paper indicates the *Frobenius Norm of a Tensor*. That is, given an Nth-order $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, we have: $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \boldsymbol{\mathcal{X}}^2(i_1, i_2, \ldots, i_N)}$.

to adopt similar hashtags to people in Austin, then perhaps we can estimate the dynamics of the hashtag *#iphone* in San Francisco. Hence, we turn in the following discussion to how we can model these latent relationships for integration into the overall framework. We denote spatio-hashtag-temporal relationships as Θ in the paper.

4.1.1 Modeling Spatial Relationships

We begin by considering the spatial relationships that connect different locations. Our hope is that we may be able to use location similarity with respect to adopting hashtags to infer propagations to unobserved locations. Concretely, we consider two approaches to model the spatial relationships of hashtags:

Geographical Distance. A natural first step is to treat locations that are near each other as similar in terms of the hashtags that will be adopted. Previous studies such as [15] have shown that the closer two locations are, the most likely they are to adopt the same hashtags due to factors like common language and shared culture, customs, and interests. Hence, we can encode this intuition in a measure of location similarity. Motivated by radial basis function (RBF) kernel [26] widely used as a similarity measure, we propose a unified geographic distance similarity score Θ_{GD} that captures the straightforward notion of geo-similarity, approaching 1 when two locations are physically proximate. The geographical similarity score Θ_{GD} is defined as:

$$\Theta_{GD}(l_i, l_j) = exp\left(-\frac{Dist(l_i, l_j)^2}{2\alpha^2}\right),$$

where α is a dispersion constant setting as 25 miles in this study. The score Θ_{GD} considers the *Haversine* formula to calculate the geographic distance $Dist(l_i, l_j)$ between location l_i and location l_j (based on their GPS coordinates). Compared to a straightline distance, the Haversine formula accounts for Earth's spherical shape.

Adoption Similarity. An alternative approach is to measure the "idea" similarity between two locations. That is, there may be locations that are not necessarily close in terms of geographical distance, but that are close in terms of the hashtags they do adopt. In this way, we can measure the *adoption similarity* between any two locations by considering two factors (i) shared hashtags and (ii) deviation of their occurrences under certain probabilities. We first apply the Jaccard coefficient to measure the degree of shared hashtags Θ_{SH} between two locations l_i and l_j:

$$\Theta_{SH}(l_i, l_j) = \frac{|H_{l_i} \bigcap H_{l_j}|}{|H_{l_i} \bigcup H_{l_j}|},$$

where recall that H_l is the set of unique hashtags adopted in a location l, and $|H_l|$ is the number of unique hashtags adopted in a location l. Two locations sharing all hashtags in common have a score of 1.0; those sharing no hashtags in common have a score of 0.0. Then, inspired by the work [14], we define the probability of observing a hashtag h as $P_h = (\sum_{l_i \in L} o_{l_i}^h)/(\sum_{h' \in H} \sum_{l_i \in L} o_{l_i}^{h'})$ where o_l^h is the number of occurrences for a hashtag h in a location l. P_h measures how likely a hashtag h occurs. Locations that adopt a hashtag with similar probabilities are considered more similar than locations that observe a hashtag with a very different adoption probabilities [15]. We continuously define the deviation of hashtag occurrences between two locations as:

$$\Theta_{DL}(l_i, l_j) = exp\left(-\sum_{h \in H'} (\frac{o_{l_i}^h - o_{l_j}^h}{o_{l_{max}}^h})^2 P_h\right),$$

where $H' = (H_{l_i} \bigcap H_{l_j})$ is denoted as the common hashtags for locations l_i and l_j, $o_{l_{max}}^h$ represents the maximum number of occurrence for hashtag h across all locations, which is used for normalization, and P_h yields the weighted average on the normalized squared difference of hashtag occurrences between two locations. Θ_{DL}, as a modified version of RBF kernel, indicates that two locations should be considered as similar while they have close distributions of occurrences as well as their real counts. Taking into account both of these two factors, we finally define the adoption similarity Θ_{AS} between two locations by multiplying them together:

$$\Theta_{AS}(l_i, l_j) = \Theta_{SH}(l_i, l_j)\Theta_{DL}(l_i, l_j),$$

where we assume that these two factors are independent and the values of Θ_{AS} are in the range $[0, 1]$.

Fusion of Two Properties. Naturally, we can integrate both geographical distance similarity and adoption similarity between two locations into a unified model. The intuition is that we can take advantage of both geographical and "idea" similarities between locations. We adopt a simple linear model to fuse these two properties:

$$\Theta_{FS}(l_i, l_j) = \tau\Theta_{GD}(l_i, l_j) + (1 - \tau)\Theta_{AS}(l_i, l_j),$$

where τ is a parameter used to control the contribution from the unified geographical similarity score Θ_{GD} and the adoption similarity Θ_{AS}. In this study, τ is set to 0.3 via cross-validation.

4.1.2 Modeling Hashtag Relationships

Complementary to location relationships, we can also directly model the relationships among different hashtags. Some hashtags are mainly local phenomena while others have a global footprint. Hence, we can measure the spatial footprint of different hashtags and compare them toward finding "similar" footprints by considering two factors (i) spatial spread of hashtags and (ii) deviation of their occurrences across all locations. Inspired by Tobler's hypothesis [31], we first define the similarity of spatial spreading for a pair of hashtags as:

$$\Theta_{SP}(h_i, h_j) = exp\left(-|\frac{d_{h_i} - d_{h_j}}{d_{max} - d_{min}}|\right),$$

where d_h is the average distance between all locations in which this hashtag h has been adopted, $|d_{h_i} - d_{h_j}|$ is used to measure the absolute difference of spatial spreading of two hashtags, $d_{max} = max(\{d_h, h \in H\})$, $d_{min} = min(\{d_h, h \in H\})$, and the term $d_{max} - d_{min}$ is a weight factor used for normalization. We then define the probability that a hashtag occurs in a location l as $P_l = (\sum_{h \in H} o_l^h)/(\sum_{l_i \in L} \sum_{h \in H} o_{l_i}^h)$, representing how likely a hashtag is to be adopted in location l. Following a similar fashion on modeling Θ_{DL}, we define the deviation of occurrences for a pair of hashtags across all locations adopting them as:

$$\Theta_{DH}(h_i, h_j) = exp\left(-\sum_{l \in L} (\frac{o_l^{h_i} - o_l^{h_j}}{o_l^{h_{max}}})^2 P_l\right),$$

where $o_l^{h_{max}}$ denotes the maximum number of occurrence for all hashtags in location l, which is used for normalization, and P_l yields the weighted average on the normalized squared difference of real counts between two hashtags across locations where they have been adopted. Assuming that these two factors are independent, we finally model the hashtag similarity Θ_{HS} by multiplying them together as:

$$\Theta_{HS}(h_i, h_j) = \Theta_{SP}(h_i, h_j)\Theta_{DH}(h_i, h_j),$$

The values of Θ_{HS} are in the range $[0, 1]$, implying that two hashtags adopted in the same locations with the same occurrences have a similarity score of 1; otherwise, they have a similarity score approaching 0.

4.1.3 Modeling Temporal Relationships

Finally, we consider enhancing the tensor completion by considering temporal relationships across memes. For the temporal

properties of hashtags, we posit that adoptions of hashtags in consecutive timestamps may be similar. Hence, we can define the temporal similarity matrix Θ_T, capturing the smoothness of the spatio-temporal dynamics of hashtags by using the tri-diagonal matrix:

$$\Theta_T = \begin{bmatrix} 0 & 1 & 0 & \dots \\ 1 & 0 & 1 & \dots \\ 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where Θ_T intuitively express the fact that $\boldsymbol{\mathcal{X}}$ in consecutive timestamps are often similar, which has been a common assumption in related efforts to recover missing data [19, 25, 34].

4.1.4 Integrating Auxiliary Information

So far, we have proposed several models to capture the relationships between locations, hashtags, and times. In this section, we investigate how to take advantage of this auxiliary information into the basic CP tensor completion model. The basic idea is if two objects are similar, e.g., two cities have similar behaviors on adopting hashtags, the latent representations of these two cities should be similar. Therefore, we want to make the latent representations of two similar objects (i.e. locations, hashtags, or timestamps) as close as possible. We denote Θ as a similarity matrix encoding relationships between entities like locations, hashtags, or times. The intuition above can be formulated as minimizing the following:

$$\begin{aligned} \mathcal{F} &= \frac{1}{2}\sum_{i,j}\Theta(i,j)\|U_i^{(n)} - U_j^{(n)}\|^2 \\ &= \sum_{i,j} U_i^{(n)}\Theta(i,j){U_i^{(n)}}^T - \sum_{i,j} U_i^{(n)}\Theta(i,j){U_j^{(n)}}^T \\ &= \text{tr}({U^{(n)}}^T(D-\Theta)U^{(n)}) \\ &= \text{tr}({U^{(n)}}^T\mathcal{L}U^{(n)}), \end{aligned}$$

where $U_i^{(n)}$ is the ith row of the factor matrix $U^{(n)}$ for the nth-mode of a tensor $\boldsymbol{\mathcal{X}}, n \in \{1,2,3\}$, $tr(\cdot)$ is denoted as the matrix trace, D is a diagonal matrix with $D(i,i) = \sum_j \Theta(i,j)$, and $\mathcal{L} = D - \Theta$ is the graph Laplacian of the similarity matrix Θ which could be any of Θ_{GD}, Θ_{AS}, Θ_{FS}, Θ_{HS} and Θ_T introduced previously.

A straightforward way to integrate relationships between locations, hashtags, and times into the basic tensor completion model is as regularization terms such that we are able to regulate latent representations of two similar objects to make them as close as possible. Hence, by integrating these auxiliary information among locations, hashtags and times, we can formulate the recovery of spatio-temporal dynamics as the following objective function:

$$\begin{aligned} \underset{U^{(n)},\boldsymbol{\mathcal{X}}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{\mathcal{X}} - [\![U^{(1)},U^{(2)},U^{(3)}]\!]\|_F^2 + \frac{\lambda}{2}\sum_{n=1}^{3}\|U^{(n)}\|_F^2 \\ & + \sum_{n=1}^{3}\alpha_n \text{tr}({Z^{(n)}}^T\mathcal{L}_n Z^{(n)}) \\ \text{subject to} \quad & \boldsymbol{\Omega} * \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{T}}, U^{(n)} = Z^{(n)} \geq 0, n = 1,2,3, \end{aligned} \tag{1}$$

where α is to control the weight of auxiliary information between locations, hashtags, and time.

4.2 Optimization Algorithm

Since the objective function in Eq.(1) is not convex with respect to variables $Z^{(n)}$ and $U^{(n)}$ together, there is no closed-form solution for this optimization problem. Motivated by methods [21, 25], we now develop an efficient algorithm to find optimal solutions for the objective function above under the framework of ADMM (Alternating Direction Method of Multipliers) that can be considered as an approximation of the method of multipliers. The objective function can be firstly written in the partial augmented Lagrangian function as follows:

$$\begin{aligned} L_\eta(U^{(n)},Z^{(n)},Y^{(n)})_{n=1,2,3} &= \frac{1}{2}\|\boldsymbol{\mathcal{X}} - [\![U^{(1)},U^{(2)},U^{(3)}]\!]\|_F^2 \\ &+ \frac{\lambda}{2}\sum_{n=1}^{3}\|U^{(n)}\|_F^2 + \sum_{n=1}^{3}\frac{\alpha_n}{2}\text{tr}({Z^{(n)}}^T\mathcal{L}_n Z^{(n)}) \\ + \sum_{n=1}^{3} < Y^{(n)}, Z^{(n)} - U^{(n)} > &+ \sum_{i=1}^{3}\frac{\eta}{2}\|Z^{(n)} - U^{(n)}\|_F^2, \end{aligned} \tag{2}$$

where $Y^{(n)}$ is the matrix of Lagrange multipliers for $n = 1,2,3$, η is a penalty parameter and $< *, * >$ is an inner product of matrices.

Updating $Z^{(1)}, Z^{(2)}, Z^{(3)}$. To update $Z^{(1)}, Z^{(2)}, Z^{(3)}$, we can re-write objective function in Eq.(2) as follows:

$$\underset{Z^{(n)}}{\text{minimize}} \quad \frac{\alpha_n}{2}\text{tr}({Z^{(n)}}^T\mathcal{L}_n Z^{(n)}) + \frac{\eta_t}{2}\|Z^{(n)} - U_t^{(n)} + \frac{Y_t^{(n)}}{\eta_t}\|_F^2. \tag{3}$$

Thus, $Z^{(n)}$ can be efficiently updated by solving the optimization problem in Eq. (3) via:

$$Z_{t+1}^{(n)} = (\eta_t \boldsymbol{I} + \alpha_n\mathcal{L}_n)^{-1}(\eta_t U_t^{(n)} - Y_t^{(n)}),$$

where $\boldsymbol{I}$ is the identity matrix with the same size of $\mathcal{L}_n$. By applying the eigen-decomposition to $\mathcal{L}_n = V_n\Lambda_n V_n^T$, we can re-write the equation above as:

$$Z_{t+1}^{(n)} = V_n(\eta_t + \alpha_n\Lambda_n)^{-1}V_n^T(\eta_t U_t^{(n)} - Y_t^{(n)}), \tag{4}$$

where $\eta_t + \alpha_n\Lambda_n$ is a diagonal matrix. Since $\mathcal{L}_n$ is eigen-decomposed at the beginning of the optimization, $(\eta_t\boldsymbol{I} + \alpha_n\mathcal{L}_n)^{-1}$ can be efficiently computed by only reversing entries on the diagonal of $\eta_t + \alpha_n\Lambda_n$ instead of calculating the inverse of the whole matrix.

Updating $U^{(1)}, U^{(2)}, U^{(3)}$. To update $U^{(1)}, U^{(2)}, U^{(3)}$, the objective function in Eq.(2) can be re-written as follows:

$$\begin{aligned} \underset{U^{(n)}}{\text{minimize}} \quad & \frac{1}{2}\|X_{(n)}^t - U^{(n)}B^{(n)}\|_F^2 + \frac{\lambda}{2}\|U^{(n)}\|_F^2 \\ & + \frac{\eta_t}{2}\|Z_t^{(i)} - U_t^{(i)} + \frac{Y_t^{(i)}}{\eta_t}\|_F^2, \end{aligned} \tag{5}$$

where $B^{(n)} = (U^{(N)} \odot \cdots U^{(n+1)} \odot U^{(n-1)} \odot \cdots U^{(1)})^T|_{N=3}$, $\odot$ is Khatri-Rao product, and $X_{(n)}$ is the mode-n unfolding of the tensor $\boldsymbol{\mathcal{X}}$.[3] Then this subproblem in terms of $U^{(n)}$ is solved as follows:

$$U_{t+1}^{(n)} = (X_{(n)}^t{B^{(n)}}^T + \eta_t Z_{t+1}^{(n)} + Y_t^{(n)})(B^{(n)}{B^{(n)}}^T + \lambda\boldsymbol{I} + \eta_t\boldsymbol{I})^{-1}. \tag{6}$$

Updating $\boldsymbol{\mathcal{X}}$. To update $\boldsymbol{\mathcal{X}}$, we can have that:

$$\boldsymbol{\mathcal{X}}_{t+1} = \boldsymbol{\mathcal{T}} + \boldsymbol{\Omega}^c * [\![U_{t+1}^{(1)}, U_{t+1}^{(2)}, U_{t+1}^{(3)}]\!],$$

where $\boldsymbol{\Omega}^c$ is the complement of $\boldsymbol{\Omega}$ that is equal to $\mathbf{1} - \boldsymbol{\Omega}$.

Updating $Y^{(n)}$. To update $Y^{(n)}$, we can have that:

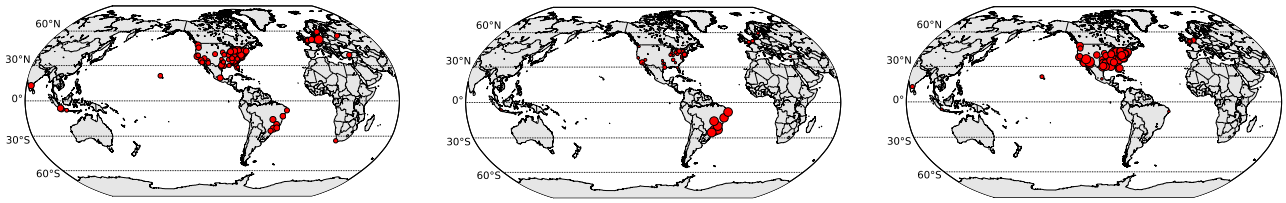
$$Y_{t+1}^{(n)} = Y_t^{(n)} + \eta_t(Z_{t+1}^{(n)} - U_{t+1}^{(n)}).$$

Updating η. We can accelerate the optimization algorithm by adaptively updating η. To update η, we can have that:

$$\eta_{t+1} = min(\rho\eta_t, \eta_{max}),$$

where ρ is a constant that we empirically set to 1.1 via cross-validation.

[3]The mode-n matrix unfolding of an order N tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ is a matrix rearranged from this tensor by fixing the dimension of the index n and multiplying other dimensions, denoted as $X_{(n)} \in \mathbb{R}^{I_n\times(\prod_{i\neq n} I_i)}$.



(a) #iphone (b) #fato (c) #healthcare

Figure 2: Distributions of three hashtags.

4.3 Recovery with Auxiliary Information

So far we have successfully solved the equation (1) by the proposed optimization algorithm based upon ADMM with leveraging auxiliary information. However, we are not able to obtain complete auxiliary information which encode similarities between locations, hashtags, and timestamps based on the sampled data. It is not reasonable to estimate missing spatio-temporal dynamics of hashtags by using auxiliary information derived from the complete data, which becomes a "Chicken-and-Egg" problem. In order to address this problem, we employ an iterative method. The initial similarity matrices derived from auxiliary information are computed based on the sampled data. And then similarity matrices will be re-calculated based on recovered spatio-temporal dynamics of hashtags. This procedure will iteratively proceed until there is no significant difference between the current and previous similarity matrices. This proposed auxiliary information regularized CP decomposition method (AirCP) is summarized in **Algorithm 1**.

Algorithm 1: Solving AirCP via ADMM

Input: $\boldsymbol{\mathcal{T}}, \boldsymbol{\Omega}, \Theta_0^{(n)}, \gamma, \lambda, \alpha_n, \rho, \eta, \eta_{max}, N$
Output: $\boldsymbol{\mathcal{X}}, U^{(n)}$
Algorithm *AirCP()*
  Initialize $U_{iter}^{(n)}$, γ, λ, α_n, ρ, η_0, η_{max}, ϵ, N, $iter = 0$
  while Not Converged and $iter \leq I_{max}$ **do**
    Construct Laplacian matrices $\mathcal{L}_n$ for matrices $\Theta_{iter}^{(n)}$
    $Optimization(\boldsymbol{\mathcal{T}}, \boldsymbol{\Omega}, U_{iter}^{(n)}, \mathcal{L}_n, \gamma, \lambda, \alpha_n, \rho, \eta, \eta_{max}, N)$
    Re-calculate similarity matrices $\Theta_{iter+1}^{(n)}$ based on $\boldsymbol{\mathcal{X}}_{iter}$
    Check the convergence:
    $\max\{\|\Theta_{iter+1}^{(n)} - \Theta_{iter}^{(n)}\|_F, n = 1, 2, \ldots, N\} < \epsilon$
    $iter = iter + 1$
  return $\boldsymbol{\mathcal{X}}_{iter}, U_{iter}^{(n)}, n = 1, 2, \ldots, N$
Procedure *Optimization*$(\boldsymbol{\mathcal{T}}, \boldsymbol{\Omega}, U_0^{(n)}, \mathcal{L}_n, \gamma, \lambda, \alpha_n, \rho, \eta_t, \eta_{max}, N)$
  Initialize $Z_0^{(n)} = Y_0^{(n)} = 0, t = 0, tol$
  while Not Converged **do**
    for $n \leftarrow 1$ ***to*** N **do**
      Update $Z_{t+1}^{(n)} \leftarrow Equation(4)$
      Update $U_{t+1}^{(n)} \leftarrow Equation(6)$
    Update $\boldsymbol{\mathcal{X}}_{t+1} = \boldsymbol{\mathcal{T}} + \boldsymbol{\Omega}^c * [\![U_{t+1}^{(1)}, U_{t+1}^{(2)}, \ldots, U_{t+1}^{(N)}]\!]$
    for $n \leftarrow 1$ ***to*** N **do**
      Update $Y_{t+1}^{(n)} = Y_t^{(n)} + \eta_t(Z_{t+1}^{(n)} - U_{t+1}^{(n)})$
    Update $\eta_{t+1} = \min(\rho\eta_t, \eta_{max})$
    Check the convergence:
    $\max\{\|U_{t+1}^{(n)} - Z_{t+1}^{(n)}\|_F, n = 1, 2, \ldots, N\} < tol$
    $t = t + 1$
  return $\boldsymbol{\mathcal{X}}, U^{(n)}, n = 1, 2, \ldots, N$

5. EXPERIMENTS

In this section, we conduct experiments to evaluate the effectiveness of the proposed AirCP framework for recovering the spatio-temporal dynamics of hashtags. Concretely, we aim to answer the following questions:

- How effective is the proposed method compared with alternative state-of-the-art methods on recovering the missing spatio-temporal dynamics of hashtags?
- What are the effects of auxiliary information – here, in the form of relationships among locations, memes, and times – on recovering the spatio-temporal dynamics of hashtags? Are some relationships more informative than others?
- How dependent on the regularization parameters is the proposed method? That is, do we need to give special care for tuning the approach, or is there a wide choice of parameters that leads to robust recovery?

We begin by introducing the Twitter dataset and the evaluation and experimental setup. Then, we compare the performance of different tensor completion methods on both synthetic and real-word hashtag data sets. At last, the effects of the different auxiliary information sources and their corresponding regularization parameters for the proposed method are investigated.

5.1 Data

Our work here focuses on an initial sample of over 55 million geo-tagged tweets via the Twitter Streaming API between February 1st and October 1st in 2013. Each tweet is tagged with a latitude and longitude indicating a location where the user posted this tweet. In this study, we first convert the GPS locations associated with tweets to corresponding cities via reverse geo-coding, and then transfer the original timestamps accurate to the second to corresponding dates. Each geo-tagged tweet can be represented by a tuple $< hashtag, city, date >$. To avoid very sparsely represented hashtags, we only consider hashtags having at least 1,000 occurrences across all cities where at least 20 unique hashtags have been adopted during the period of our data collection. Since some hashtags have appeared before the first day of the sample, we only keep those hashtags that first appear after February 1st, 2013, resulting in 4,723 unique hashtags occurring in 2,415 cities. After randomly selecting 2,000 of 4,723 hashtags, the data set consists of 2,000 hashtags occurring in 1,278 cities in the world over a span of 242 days, which we model as a tensor $\mathcal{X} \in \mathbb{R}^{2000 \times 1278 \times 242}$. For the experiments, we view this sample as if it were the true (complete) spatio-temporal dynamics of the 2,000 hashtags across these 242 days. To illustrate, Figure 2 shows the global footprint of three different hashtags (*#iphone*, *#fato*, and *#healthcare*) in the dataset.

5.2 Experimental Setup and Metrics

We evaluate the effectiveness of the proposed framework compared with alternative methods by evaluating them over the three scenarios introduced in Section 3: Scenario 1, in which we have random missing observations; Scenario 2, in which some locations

are missing entire hashtags; and Scenario 3, in which we are missing entire locations. We set the parameters in Eq.(2) through cross-validation with a separate validation dataset. We empirically set $\lambda = 0.1$ and $\alpha_1 = \alpha_2 = \alpha_3 = 0.1$ for all following experiments.

To investigate the quality of the proposed framework, we adopt *Relative Error*, and *Accuracy@k* as evaluation metrics. Relative Error is defined as $RelativeError = \|X - Y\|_F / \|Y\|_F$ where X is the recovered tensor and Y is the ground-truth tensor. Accuracy@k represents the percentage of correctly predicted popular hashtags out of the top-k popular hashtags. Formally, if we denote S_l as the real top-k popular hashtags at a location l and $\hat{S}_l$ as the set of popular hashtags selected by a recovery method at a city l, we have $Accuracy@k = (\hat{S}_l^t \cap S_l^t)/k$ which is in the range [0,1]. In the following experiments, we evaluate k at 1, 5, and 10.

5.3 Baseline Methods

Previous research has shown that tensor-based completion methods outperform matrix-based ones [20, 32, 34]. Hence, we focus our evaluation here on tensor-based state-of-the-art methods:

- *Tensor Factorization with Auxiliary Information (TFAI)*: The first baseline is a tensor analysis method introduced in [25] that integrates auxiliary information. We adopt the within-mode auxiliary information method that performs better than the cross-mode method according to the results.
- *Trace Norm-based CP Decomposition (TNCP)*: The second baseline method regularizes the trace norm in the CP tensor decomposition method based upon alternating direction method of multipliers (ADMM) [21]. We choose parameters $\lambda = 10$, and $\alpha = 0.33$ that result in the best performance for this method.
- *Low-Rank Tensor via Convex Optimization (LRCO)*:The third baseline is a low-rank tensor factorization method with a trace norm regularization [32]. We adopt the "mixture" version, which models the tensor as a mixture of K sub-tensors. We set the initial step-size $\eta_0 = 0.1$ and $\lambda = 0$.
- *Weighted Tucker Decomposition (WTucker)*: The fourth baseline is a weighted Tucker decomposition [8] that is similar in spirit to PARAFAC [16] method with missing data.
- *Fast Low-Rank Tensor Completion (FaLRTC)*: Finally, we consider the fast low-rank tensor completion method [20], which estimates missing data based on the smoothed trace norm.

5.4 Evaluating AirCP over Synthetic Data

As a first step toward evaluating the effectiveness and efficiency of the AirCP method, we first test over a synthetic dataset before moving on to the real hashtag data. We generate a low-rank (10,10,10) tensor $M \in \mathbb{R}^{100\times100\times100}$ with correlated objects as the ground truth data. The factor matrices $U^{(1)} \in 100 \times 10$, $U^{(2)} \in 100 \times 10$, and $U^{(3)} \in 100 \times 10$ are generated by the following linear formulae [25]:

$$U^{(1)}(i,r) = i\varepsilon_r + \varepsilon', \quad i = 1,2,\ldots,100, r = 1,2,\ldots,10$$
$$U^{(2)}(j,r) = j\zeta_r + \zeta', \quad j = 1,2,\ldots,100, r = 1,2,\ldots,10$$
$$U^{(3)}(k,r) = k\eta_r + \eta', \quad k = 1,2,\ldots,100, r = 1,2,\ldots,10$$

where $\{\varepsilon_r, \varepsilon'_r, \zeta_r, \zeta'_r, \eta_r, \eta'_r\}_{r=1,2,\ldots,10}$ are constants generated by the standard Gaussian distribution $N(0,1)$. Then the synthetic tensor M is calculated as $M = \mathcal{J} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$ where $\mathcal{J} \in \mathbb{R}^{10\times10\times10}$ is a unit tensor with all of its super-diagonal elements being 1 and the other elements being 0 and $\times_i$ means the tensor-matrix operation for the dimension-i of tensor. Since each factor matrix is generated by linear functions mentioned above column by column, the consecutive rows are similar to each other.

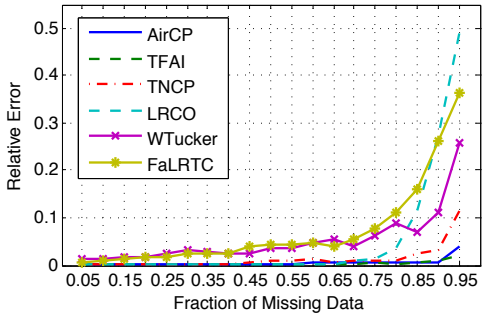


Figure 3: Comparison of recovery results over synthetic data.

Therefore, we generate the similar matrix for the ith mode as the following tri-diagonal matrix:

$$\Theta_i = \begin{bmatrix} 0 & 1 & 0 & \ldots \\ 1 & 0 & 1 & \ldots \\ 0 & 1 & 0 & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (7)$$

Table 2: Computation time (in seconds) over synthetic data as the fraction of missing data (FM) varies.

FM	AirCP	TFAI	TNCP	LRCO	WTucker	FaLRTC
20%	**3.97**	46.36	4.51	58.64	195.22	5.47
40%	4.27	49.87	**3.94**	49.61	186.13	5.62
60%	**3.89**	35.98	4.08	51.30	202.58	4.78
80%	4.76	43.75	**4.16**	60.04	183.98	4.46

We randomly sample entries from M and recover the complete tensor by varying the fraction of unobserved entries from 5% to 95%. We set the tolerance of error as 10^{-5} and the maximal number of iteration as 1,000 for all methods we tested here.

We show in Figure 3 the relative error for all methods, averaged over 10 independent runs. At a moderate fraction of missing data, most of the methods perform comparably, with only WTucker and FaLRTC performing clearly worse. But in cases when there is a large fraction of missing data (i.e. greater than 75%), we see that AirCP and TFAI achieve the lowest relative error in all cases and that this error is objectively low. This result is encouraging since it indicates that the proposed framework for spatio-temporal dynamics recovery can achieve robust recovery in realistic scenarios where only a small fraction of data is available.

While AirCP and TFAI achieve relatively lower error rates, what about their comparative efficiency? We present the average computation time (in seconds) of all tested approaches in Table 2. We can observe that AirCP is an order of magnitude faster than TFAI and that it is on par with both TNCP and FaLRTC, which both demonstrate higher relative errors (as shown in Figure 3). Hence, these experiments over synthetic data show the potential of the proposed AirCP method to achieve low error rates while also being more appropriate for large-scale data.

5.5 Evaluating AirCP over Hashtag Data

Given these encouraging results, we now turn to an examination of AirCP over the real hashtag data. We consider the three missing data scenarios introduced previously. For all cases, we set the rank of the tensor to 10. For Scenario 1 (*Random Missing Observations*), we randomly select a fraction of all hashtag-location-time counts and assume that these are unobservable (that is, missing). We report results by varying the fraction from 25% to 55% to 85%. For Scenario 2 (*Missing Entire Memes at Some Locations*), we ran-

Table 3: Relative errors for recovering missing hashtags as the fraction of missing data varies from 25% to 55% to 85%. We observe that AirCP is an order of magnitude faster than TFAI.

Method	Scenario 1			Scenario 2			Scenario 3			Avg. Improvement
	25%	55%	85%	25%	55%	85%	25%	55%	85%	(comparing with AirCP)
AirCP(FS+HS+T)	0.1059	**0.3330**	**0.5079**	**0.2032**	**0.4615**	**0.6017**	0.2830	**0.5772**	**0.7680**	N/A
TFAI	**0.0999**	0.3565	0.5314	0.2282	0.4870	0.6336	**0.2828**	0.5863	0.7861	3.34%
TNCP	0.1829	0.4198	0.5762	0.2846	0.5937	0.7378	0.3674	0.5919	0.8272	19.62%
LRCO	0.2307	0.4753	0.6851	0.3245	0.6261	0.798	0.4097	0.6288	0.9026	28.02%
WTucker	0.4859	0.778	1.1602	0.7427	1.0155	1.3256	1.1852	1.3914	1.7383	62.65%
FaLRTC	0.2417	0.4543	0.6548	0.3156	0.5621	0.8069	0.3592	0.6243	0.8802	25.09%

Table 4: Relative errors for recovering appearances of hashtags as the fraction of missing data varies from 25% to 55% to 85%. We witness that AirCP is an order of magnitude faster than TFAI.

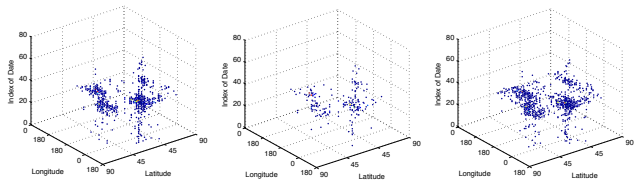
Method	Scenario 1			Scenario 2			Scenario 3			Avg. Improvement
	25%	55%	85%	25%	55%	85%	25%	55%	85%	(comparing with AirCP)
AirCP(FS+HS+T)	**0.2032**	0.4675	**0.7017**	**0.2405**	**0.3653**	**0.4719**	**0.3083**	**0.5965**	**0.7451**	N/A
TFAI	0.2082	**0.4470**	0.7236	0.2668	0.3976	0.4970	0.3246	0.6281	0.7900	4.40%
TNCP	0.2846	0.5937	0.7878	0.3148	0.4644	0.5799	0.4071	0.6599	0.8541	18.98%
LRCO	0.3245	0.6261	0.7980	0.6303	0.6142	0.6428	0.4380	0.6605	0.9315	29.23%
WTucker	0.7427	1.0155	1.3256	0.8484	0.8651	0.7403	1.2191	1.4297	1.7844	61.66%
FaLRTC	0.3156	0.5621	0.8069	0.4767	0.599	0.6313	0.4154	0.6660	0.9089	26.03%

domly select for each location some fraction of hashtags that are unobservable. Again, we report results for 25%, 55%, and 85%. Finally, for Scenario 3 (*Missing Entire Locations*), we randomly select a fraction of the 1,278 locations and assume that these locations are completely unobservable (missing) across the whole collection period. We evaluate all methods using a fraction of missing locations of 25%, 55%, and 85%.

We present in Table 3 the relative error for all three scenarios across all approaches for three levels of missing data (25%, 55%, and 85%). Reinforcing our observations from the synthetic data experiment, we witness that AirCP achieves better performance than TFAI with an average improvement of 3.34% over real hashtag data. In practice, again, TFAI still takes around an order of magnitude longer to calculate than the proposed AirCP method. We see that AirCP gives an average improvement of 27.8% in terms of relative error over other alternative methods. And as the sparser the observed tensor is (that is, the smaller the number of actual observed hashtags), we see that AirCP gives an even greater improvement versus the alternatives. To illustrate, Figure 4 shows an example recovery for the hashtag *#mtvema* for Scenario 1 when 85% of the data is missing. We can see that the proposed method can successfully recover the sampled data based upon limited sample data (15% of the complete data).

Returning to Table 3, for both Scenarios 2 and 3 in which either a portion of all hashtags for a location are missing or the entire location is unobserved (which places great burden on the recovery framework, since there are not even partial observations for those hashtags in those locations as in Scenario 1), we see that integrating the latent relationships among locations, hashtags, and time (e.g., distances between two cities, similarities between hashtags, same hashtags adopted by two cities) can lead to a significant improvement through the tensor factorization. These relationships can alleviate the problem of sparsity to some extent and provide valuable information for the tensor factorization to obtain more interpretable low-rank representations.

Moreover, after varying the rank of tensor from 5 to 20, we found that the proposed method has a better consistency than other alternative state-of-the-art methods as the rank of tensors increases, implying that the proposed method is more robust on predicting missing diffusion dynamics of hashtags. The details are omitted here due to the limited space.



(a) Original (b) Sampled (c) Recovered

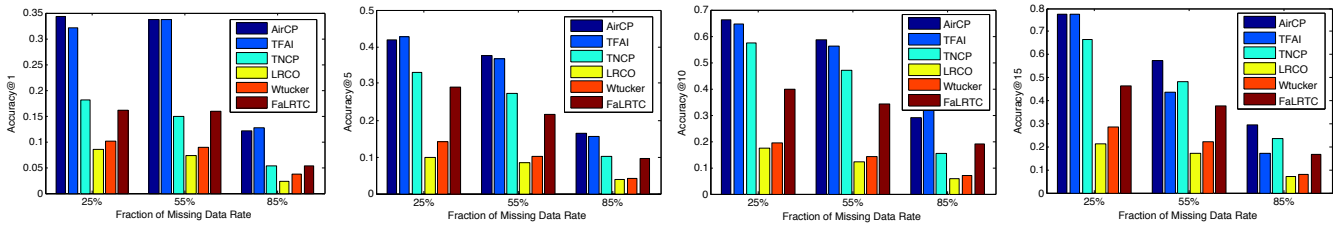
Figure 4: An example recovery for the hashtag #mtvema when 85% of the data is missing (Scenario 1).

5.6 Recovery Under Constraints

While the previous experiment examined whether we could recover the count of the number of hashtags in a location at a particular time, we now turn to two more constrained situations that could arise in practice.

Appearance of Hashtags. In the first situation, we consider the task of determining whether or not a hashtag has appeared at a location at a particular timestamp. By considering only this binary information (rather than count information), we can explore the quality of the proposed approach at identifying rare hashtags, rather than emphasizing on hashtag counts as in the previous experiments. In this way, we can determine how well the approaches recover the appearance information of hashtags. For this experiment, for a hashtag h, the corresponding cell $\boldsymbol{\mathcal{X}}(h, l, t)$ will be set as 1 if that hashtag appears in a city l at a date t. Otherwise, it will be assigned to 0. For the recovered tensor $\boldsymbol{\mathcal{X}}$, the cell $\boldsymbol{\mathcal{X}}(h, l, t)$ will be set as 1 if the value of that cell is larger or equal to a threshold; otherwise, it will be set to 0. The threshold is empirically set to 0.3 via cross-validation. In this experiment, we vary the fraction of missing data in [25%, 55%, 85%] for all three scenarios, meaning that the observations we randomly selected based on fractions of missing data can be considered as training data, and the remainder as the test data. We empirically set the rank of tensor to 10. The experimental results in terms of relative errors are illustrated in Table 4.

We can see that in general, the proposed AirCP consistently outperforms other alternative methods including TFAI, TNCP, LRCO, WTucker, and FaLRTC in all three scenarios. Specifically, it gives an average improvement of 28.01% on the relative error over other alternative methods. This indicates that auxiliary information among locations, hashtags, and times can help predict whether a hashtag



(a) Top-1 popular hashtag (b) Top-5 popular hashtags (c) Top-10 popular hashtags (d) Top-15 popular hashtags

Figure 5: Recovering hashtag popularity: Accuracy@1, 5, 10, and 15 as the fraction of missing data varies from 25%, 55% to 85%. Though AirCP only achieves slightly better performance than TFAI, it has much better time efficiency for the computation with around an order of magnitude faster speed.

has appeared in a location at a time or not. Similar to the previous experiment, TFAI has comparable performance with the proposed method, which performs worse in scenarios 1 and 2 and slightly better in scenario 3. However, as TFAI requires longer computation time, the proposed AirCP method is more efficient at this task.

Popularity of Hashtags. In the second situation, we consider the task of determining the top-k hashtags at a location at a particular timestamp. In this way, we can explore the quality of the proposed approach at identifying the popularity of hashtags. We evaluate the performance of the AirCP method in the recovery of top-k popular hashtags in 1,278 cities by varying the value of k as the fraction of missing values varies. For a hashtag h, the corresponding cell $\boldsymbol{\mathcal{X}}(h, l, t)$ will be set as 1 if the total number of occurrences of that hashtag is one of the top-k largest among all hashtags occurred in that city l after the date t; otherwise, it will be assigned to 0. For this problem, only Scenario 3 is reasonable since for Scenarios 1 and 2, we are not likely to accurately mark the top-k popular hashtags in a city only depending on observed sampled data. It indicates that we cannot know the top-k popular hashtags in a location l after a date t while only observing partial data. For instance, we mark hashtags *#cjbbq* and *#subway* as two of top-3 popular hashtags in Houston after the date t only based on observed sample data in the period of the data collection. Nevertheless, once we can retrieve the whole diffusion data of hashtags in Houston, the top-3 popular hashtags are *#northgate*, *#rockets* and *#texian* as we do not observe or partially observe their diffusion data. Therefore, Scenarios 1 and 2 would never happen on this problem. In the implementation of the proposed method, we empirically set the rank of tensor to 10.

The performance comparison is presented in Figure 5. As we can see, overall, the proposed AirCP generally gives the best performance in terms of accuracy for different k with a maximal accuracy of 77.7%. Since the distribution of hashtags at a location usually follows a Zipfian distribution, it could be harder for the problem of predicting top-k hashtags at a location at a date as the value of k decreases. This hypothesis is confirmed in the performance of our method. We observe that all these compared methods as well as the AirCP method perform better with higher values of k. In addition, we can see that the AirCP method performs consistently when the fraction of missing data is less then 85%. All these results illustrate that the proposed method can successfully predict top-k popular hashtags for a city after a date with limited observations.

5.7 Effects of Auxiliary Information

We next turn to the relative impact of the different sources of auxiliary information. Recall that we consider different relationships among locations, hashtags, and time. Here, we are interested to explore whether some of these relationships are more informative than others. To that end, we narrow our focus on the problem of recovering hashtag counts under Scenario 1 (where a random fraction of hashtags are missing), and empirically fix the rank of tensor

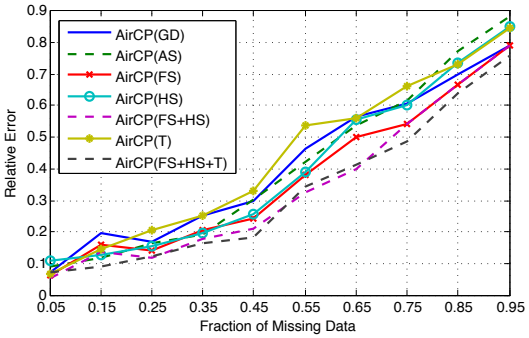


Figure 6: Relative errors for different combinations of auxiliary information.

to 10. We evaluate the proposed AirCP method with different combinations of auxiliary information. In Figure 7, we show how auxiliary information affects the performance of the proposed method in terms of the relative error by varying the fractions of missing data. We can see that, in general, the proposed AirCP method integrating all three types of auxiliary information (modeled in Section 4) achieves the best performance than those only integrating part of the auxiliary information, indicating that the proposed method successfully makes use of all useful information sources to perform effective recovery for the spatio-temporal diffusion dynamics of hashtags. For the spatial information, we find that AirCP with the fusion of geographical distance similarity (GD) and adoption similarity (AS) performs better than ones with either GD or AS solely. This result implies that the integration of these two types of information yields complementary evidence of hashtag adoption. We also observe that AirCP with only the hashtag similarity (HS) has comparable performance with one integrating GD and AS (denoted as FS). In summary, the use of all three types of auxiliary information can help enhance the performance of the proposed AirCP.

5.8 Effects of Regularization Parameters

Finally, we explore the impact of the regularization parameters on the quality of hashtag recovery. Recall that these parameters control the contributions of the relationships between locations, hashtags and time on the recovery framework. In order to better understand the effect of different choices of these parameters, we vary their values in the range [0.001 0.01, 0.1, 1, 10], and then evaluate the AirCP method for the scenario in which some hashtags are missing (Scenario 1) with a fraction of missing data of 55%. The rank of the tensor is set to 10 and other settings are the same as we set in Section 5.5. We observe in Figure 7 that the proposed AirCP method achieves relatively stable performance when the parameters are in the range [0.01, 0.1, 1]. This result indicates that the proposed framework is fairly robust to reasonable choices of these parameters. Specifically, comparing all results for parameters, we find that setting $\alpha_1(city) = 0.1$, $\alpha_2(hashtag) = 0.1$

and $\alpha_3(date) = 0.1$ achieves the best performance with a relative error of 0.3330, and that parameter settings of $\alpha_1(city) > 0.1$, $\alpha_2(hashtag) > 0.1$ and $\alpha_3(date) > 0.1$ lead to fairly stable relative errors. These results indicate the stability of the proposed AirCP to these regularization parameters. Similar results can be found when we set the fraction of missing data to 25% and 85%.

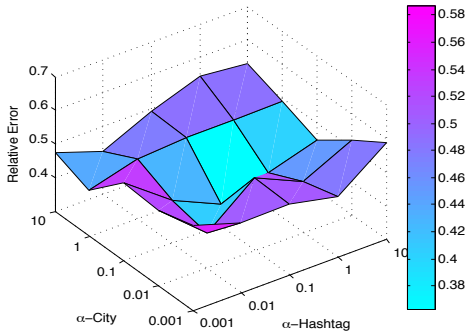


Figure 7: Relative errors for different parameter settings: $\alpha_3(date) = 0.1$ with 55% of the fraction of missing data.

6. CONCLUSION

In this paper, we have tackled the critical problem of recovering spatio-temporal dynamics of memes. Concretely, we have proposed a tensor-based spatio-temporal dynamics recovery framework that leverages auxiliary information among locations, hashtags, and times with better time efficiency. Through experimental evaluation on both synthetic and real-world Twitter hashtag data, we see that the proposed framework outperforms alternative state-of-the-art methods with an average improvement of over 27%, and find that the integration of auxiliary information among locations, hashtags, and times are crucial factors in the performance of the proposed framework. In our future work, we are interested in exploring opportunities to add more external information such as textual, and social contexts into the proposed framework with a better performance in recovering spatio-temporal dynamics of memes.

7. ACKNOWLEDGEMENTS

This work was supported in part by AFOSR grant FA9550-15-1-0149 and NSF grant IIS-1149383.

8. REFERENCES

[1] L. Backstrom, E. Sun, and C. Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *WWW*, 2010.

[2] M. T. Bahadori, Q. R. Yu, and Y. Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *NIPS*, 2014.

[3] A. Brodersen, S. Scellato, and M. Wattenhofer. Youtube around the world: geographic popularity of videos. In *WWW*, 2012.

[4] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM*, 2010.

[5] M. Clements, P. Serdyukov, A. P. De Vries, and M. J. Reinders. Using flickr geotags to predict user travel behaviour. In *SIGIR*, 2010.

[6] N. Dalvi, R. Kumar, and B. Pang. Object matching in tweets with spatial models. In *WSDM*, 2012.

[7] D. M. Dunlavy, T. G. Kolda, and E. Acar. Temporal link prediction using matrix and tensor factorizations. *TKDD*, 2011.

[8] M. Filipović and A. Jukić. Tucker factorization with missing data with application to low-n-rank tensor completion. *MSSP*, 2014.

[9] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat. Matrix factorization with explicit trust and distrust side information for improved social recommendation. *TOIS*, 2014.

[10] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 1976.

[11] H. Ge, J. Caverlee, and H. Lu. Taper: A contextual tensor-based approach for personalized expert recommendation. In *RecSys*, 2016.

[12] X. Hu, J. Tang, H. Gao, and H. Liu. Social spammer detection with sentiment information. In *ICDM*, 2014.

[13] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He. Fast and accurate matrix completion via truncated nuclear norm regularization. *TPAMI*, 2013.

[14] K. Y. Kamath and J. Caverlee. Spatio-temporal meme prediction: learning what hashtags will be popular where. In *CIKM*, 2013.

[15] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng. Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In *WWW*, 2013.

[16] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM*, 2009.

[17] D. Kondrashov and M. Ghil. Spatio-temporal filling of missing points in geophysical data sets. *NPG*, 2006.

[18] G. Kossinets. Effects of missing data in social networks. *Social Networks*, 2006.

[19] H. Lamba, V. Nagarajan, K. Shin, and N. Shajarisales. Incorporating side information in tensor completion. In *Companion on WWW*, 2016.

[20] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *TPAMI*, 2013.

[21] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng. Trace norm regularized candecomp/parafac decomposition with missing data. *Cybernetics*, 2014.

[22] Z. Lu, D. Agarwal, and I. S. Dhillon. A spatio-temporal approach to collaborative filtering. In *RecSys*, 2009.

[23] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *SIGKDD*, 2012.

[24] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. *ICWSM*, 2013.

[25] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima. Tensor factorization using auxiliary information. *ECML PKDD*, 2012.

[26] C. E. Rasmussen. Gaussian processes for machine learning. 2006.

[27] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *WSDM*, 2010.

[28] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. In *WSDM*, 2011.

[29] T. Sakaki, M. Okazaki, and Y. Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *TKDE*, 2013.

[30] M. Scheuerer, R. Schaback, and M. Schlather. Interpolation of spatial data–a stochastic or a deterministic problem? *European Journal of Applied Mathematics*, 2013.

[31] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 1970.

[32] R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

[33] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, 2011.

[34] H. Zhou, D. Zhang, K. Xie, and Y. Chen. Spatio-temporal tensor completion for imputing missing internet traffic data. In *IPCCC*, 2015.