Leveraging Social Context for Modeling Topic Evolution

Janani Kalyanam
Univ. of California, San Diego
jkalyana@ucsd.edu

Amin Mantrach
Yahoo Labs
Barcelona, Spain
amantrac@yahoo-inc.com

Diego Saez-Trumper
Yahoo Labs
Barcelona, Spain
dsaez-trumper@acm.org

Hossein Vahabi
Yahoo Labs
Barcelona, Spain
puya@yahoo-inc.com

Gert Lanckriet
Univ. California, San Diego
gert@ece.ucsd.edu

ABSTRACT

Topic discovery and evolution (TDE) has been a problem which has gained long standing interest in the research community. The goal in topic discovery is to identify groups of keywords from large corpora so that the information in those corpora are summarized succinctly. The nature of text corpora has changed dramatically in the past few years with the advent of social media. Social media services allow users to constantly share, follow and comment on posts from other users. Hence, such services have given a new dimension to the traditional text corpus. The new dimension being that today's corpora have a *social context* embedded in them in terms of the community of users interested in a particular post, their profiles etc. We wish to harness this social context that comes along with the textual content for TDE. In particular, our goal is to both qualitatively and quantitatively analyze when social context actually helps with TDE. Methodologically, we approach the problem of TDE by a proposing non-negative matrix factorization (NMF) based model that incorporates both the textual information and social context information. We perform experiments on large scale real world dataset of news articles, and use Twitter as the platform providing information about the social context of these news articles. We compare with and outperform several state-of-the-art baselines. Our conclusion is that using the social context information is most useful when faced with topics that are particularly difficult to detect.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications: Text processing

Keywords

Social networks, topic discovery, topic monitoring, topic tracking, collective factorization

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2783319.

1. INTRODUCTION

Topic discovery has been a well studied area of research since the 90s. In recent times, this area has gained renewed interest with the advent of social media [23, 3]. Social media has completely changed the dynamics of how we operate as a society, and has given each of us the power of being able to constantly produce and share content with the rest of the world. This presents several new challenges to the field of topic discovery and evolution (TDE). Firstly, the content is constantly evolving. Hence, any topic discovery algorithm needs to keep with the ever changing nature of the content. And secondly, since each one of us has the power to produce and share content, the vocabulary used to describe a particular event can be quite varied. In addition to this, it has also resulted in an enormous explosion in the size of data making it more challenging to identify which posts are indeed the important ones. Hence algorithms need to be robust to such irregularities in the textual content, and should identify the most important and relevant information from a large corpus.

Many classical TDE algorithms aim to detect the underlying latent topics from the textual content of the data alone. They are blind to the social context that comes along with the text. For example, Twitter contains information about the user, geographical location, time of post, etc, which could be very useful information as it gives a context to the textual content. We could incorporate such context information in our learning process to learn better topics. What we propose in this paper is to use the information about communities present in social media in addition to textual content to discover and track topics. We propose to use the information about the authorship of posts, shares and comments to detect communities of users. By definition, members of the same community will exhibit common interests in sharing and posting information about a particular topic. The idea here is to leverage this information (as side information) to retrieve more accurate topics.

We hypothesize that some topics are particularly difficult to discover using textual content alone either because the text present in the topic uses a widely varying vocabulary or that the text could be very volatile and could change in a very short period of time. An example is: *celebrity gossip*. For such a topic, the content varies from one celebrity to another and hence can have a widely varying vocabulary, or can take unexpected turns (celebrity break-ups, pregnancies,

etc.) and thus be volatile. But, for the same topic, we could have a very dedicated community of users who constantly share posts and comments about the topic. In this case of *celebrity gossip*, it perhaps finds its niche audience largely in the teenage demographics. Hence, leveraging the presence of community could be useful in discovering such difficult topics. We will call such topics which have a volatile and constantly changing text, but a farily dedicated community of users following the topic to be *community stable topics*. There could be some other topics which have a focused textual content that does not change much over time. For these seemingly "easier" topics (which we will call *content stable topics*), perhaps using the community of interested users as side information may not particularly improve topic discovery[1]. One cannot say with surity (yet). This is precisely the kind of question we provide an answer to in this paper.

To summarize, our aim in this paper is to study in detail if and when the presence of social context information is useful in discovering and monitoring topics. There are works in literature which have combined both content and link information, but the natural time-based evolving and changing aspects of both have not been considered. Other works track content along time, but do not accommodate for both content and link information. To the best of our knowledge, we are the first to propose an approach that exploits simultaneously both the content and the social context in a unified framework for modeling topic evolution. We build on the non-negative matrix factorization (NMF) objective, with different terms modeling the content and social context aspects of our problem. Within each modality, we model their temporal evolution to learn from the past as well. To learn both modalities at the same time, and to exploit their correlation, we rely on a *collective factorization* approach as introduced in [25]. In our context, it consists of sharing at each time step, a common variable representing both the topic and the community distributions during the learning process (for more details, see Section 3).

We perform experiments on a publicly available large scale dataset of news articles [20], and use the Twitter social networking platform as the source of social context for the news articles. We study the effects of social activity as side information when modeling topic evolution. In particular, our focus will be on the following research questions.

1. Does the presence of community as side information help in discovering those topics which have a strongly focused textual content (i.e. *content stable* topics)? One may generally not expect the presence of community to help in this case, since the strong textual content probably suffices to discover the topics. However, through our experiments, we indeed see improvements in performance in some cases.

2. Does the presence of community help in discovering those topics which do not have stable textual content, but have stable communities of members interested in them (i.e. *community stable* topics)? In this case, we observe remarkable improvements in performance when we use the community information, as opposed to only using textual content.

3. Does the presence of community help in discovering topics which have both a stable textual content and a stable community (i.e. *mixed stable* topics)? We observe improvements in performance in this case as well.

4. How does our algorithm compare to existing state of the art which model topic evolution and those which model document link structure for topic discovery? In particular, we compare our algorithm to Link-PLSA-LDA [2], a generative model that incorporates context and content (but no tracking); Collective Matrix Factorization [22, 8], an NMF-based model that incorporates context and content (but no tracking); Joint Past Present Decomposition [26], an NMF based topic tracking model, and Online-LDA, [16], a generative topic tracking model. We indeed outperform the state-of-the-art in several instances.

5. To what extent can our algorithm learn the kind of topics and communities that are at hand? That is, given an input stream of documents, how well can the algorithm figure out whether the topics that have been detected are content stable, community stable or mixed stable?

The rest of the paper is organized as follows. Section 2 compares our work to existing work highlighting its novelties and differences over them. Section 3 explains our model loss function in detail, and how we optimize it. Section 4 describes the dataset we used in our experiments. Section 5 provides a detailed explanation of the experiments and results, and we finish with the conclusion in Section 6.

2. COMPARISON TO PREVIOUS WORK

There are two families of work which we delve into to provide an overview of related papers: one is the family of work on TDE, and the other is family of work that uses some type of link structure (either derived from citation networks, or other means) for topic modeling. Works which fall under the latter family tree generally do not model the evolution of topics that are discovered, and hence do not incorporate a temporal aspect to the model they develop. To the best of our knowledge, our work is the first to combine both topic discovery and evolution with link structure. More importantly, our work is the only one which studies where the soft spot really lies. Meaning, we comprehensively study through experiments for what kind of topics does the social context of an article through user interactions really produce improvements in performance.

Topic discovery and evolution has been a subject which has garnered plenty of attention for more than a decade but has gained renewed interest in recent years with the advent of the social media [23, 3]. The most effective models developed by the topic tracking community is generally built on some well-known topic discovery model (or topic model) with a temporal aspect added to it to accommodate for the incoming stream of data. This is the case with NMF (non-negative matrix factorization) [13] based models that connects along time the learned representations for the incoming stream of data [5, 14, 21, 26]. In the same spirit, other works extend generative models like latent dirichlet allocation (LDA) [6] for analyzing the evolution of topics along time [2, 28, 11, 27].

Social information for topic detection has masqueraded with many names in literature. There have been entire lines

[1]In Section 4, we will explain in detail how we obtain the content stable topics, community stable topics etc.

of works which use the link structure between documents to model topics. This link structure can be built to model a certain relationship between documents. Examples of information that can be modeled through link structure are common authors between documents, citation networks, etc. Many of these models derive inspiration from classical topic modeling algorithms, and extend them to incorporate for the new modality of information now available to them. [10] proposed the Link-LDA model which extends LDA to include citation information. It replicates the graphical model used for modeling documents and words to also model documents and citations. It enforces that the document's topic distribution and the document's citation distribution to be the same. [16] propose Link-PLSA-LDA as a scable LDA-type model for topic modeling and link prediction. Relational Topic Model (RTM) was proposed by [7] to model link between documents as a binary random variable based on the content of the document. They do not consider the community information. [19] propose the author-topic model to simultaneously model the content of the topic and the interest of the author using a shared hyperparameter. [15] propose the topic-author-recipient model to take into account the directionality of the link between the documents, and models the "who-cited-whom" information. More recently, we have works of [9] which represents documents through 'badges', which are essentially descriptive terms from the users sharing the documents. However, in our method, we model the full authorship information as a matrix and perform collective matrix factorization. We note that none of these methods that use the link structure or authorship information consider temporal aspect for monitoring topics.

3. LEARNING FROM CONTENT AND SOCIAL MEDIA ACTIVITY

In this section, we explain how we formulate and optimize the problem of topic discovery and evolution using content and social context information. Henceforth, we refer to our method as **LTECS**, an acronym for Learning Topic Evolution from Content and Social media activity. We begin with some notation. We assume a constant flow of documents. Let $\mathbf{X}^t$ be a $N_d^t \times N_f$ matrix at time t of N_d^t documents and N_f textual features. The complete data matrix $\mathbf{X}$ obtained by concatenating vertically the matrices $\mathbf{X}^t$ along the time steps is considered huge and and practically difficult to store and handle. The simplest approach to topic detection consists of directly learning from the global matrix $\mathbf{X}$. However, in the real world, we are observing evolving topics and trends [15]. Hence, using much older data to estimate current trends may lead to wrong inference. Another typical strategy, consists of directly learning topics from the current batch of data while ignoring the trend history. One is therefore faced with the tradeoff between past and present observations. While recent approaches modeling topic evolution do address this tradeoff [26, 2], they rely only on the content of the documents as their primary mode of input. In order to consider other modalities as well (e.g. the social context associated to user activities), we introduce in the remainder of the section a multimodal approach to model topic evolution. For the social context input, we have associated to each document in $\mathbf{X}^t$, a set of users who are interested in these documents. Let $\mathbf{U}^t$ be a $N_d^t \times N_u$ matrix at time t of N_d^t documents and N_u users. Here, N_u is the total number of users in the social network. In particular, we have $\mathbf{U}_{ij}^t = 1$ if document-i has been mentioned by user-j, and it is 0 otherwise.

3.1 The Objective Function

Our aim is to discover topics using both $\mathbf{X}^t$ and $\mathbf{U}^t$. We will start with the traditional objective function for NMF and build on it. The goal of non-negative matrix factorization is to decompose documents in terms of the underlying latent topics. Let us fix the number of topics to be k. We would like to decompose $\mathbf{X}^t$ so that:

$$\mathbf{X}^t \approx \mathbf{W}^t\mathbf{H}^t. \tag{1}$$

Here, the $\mathbf{H}^t$ is a $k \times N_f$ topic matrix. Each row in $\mathbf{H}^t$ represents an underlying latent topic. If the encoding features of $\mathbf{X}^t$ are the words themselves, then each entry in $\mathbf{H}^t$ represents how frequently a particular word appears in a topic. The $\mathbf{W}^t$ matrix represents how each document is decomposed in terms of the topics found in $\mathbf{H}^t$. It *explains* each document in terms of the topics discovered in the $\mathbf{H}^t$ matrix.

For each document, in addition to the textual features, we have information about which users are interested in these documents. Just as in Equation 1, where we decomposed each document in terms of the latent topics, we can think of decomposing the documents in terms of the latent communities found in the social network. That is, we have:

$$\mathbf{U}^t \approx \mathbf{W}^t\mathbf{G}^t. \tag{2}$$

The key assumption in our formulation is that we have a common decomposition matrix $\mathbf{W}^t$ for both equations 1 and 2. Our assumption is that a particular community of users will be dedicated to a particular topic. Hence, we should be able to decompose a document in terms of its topic *or* in terms of its communities in the same way. An article about Kim Kardashian can be thought of being decomposed as 90% showbiz and 10% spread across the other topics. Our postulation is that, there is a community of users who show keen interest in showbiz news, perhaps a community in teenage demographics. Hence, the same document can be equivalently decomposed in terms of the community as 90% community interested in showbiz and 10% spread across the other communities. Equations 1 and 2 form the backbone of the two different parts (namely the topic and the community part) to our objective function. The way through which we connect the two modalities is via the $\mathbf{W}^t$ matrix, making it common to both decompositions. This method is traditionally referred as *collective factorization* [25], and consists of sharing one common variable across different modalities. The same principles have also been applied in deep learning (by sharing a common hidden layer across different modalities) [18], and in probabilistic modeling (by conditioning different observed modalities on a common hidden random variable) [4].

Since we also wish to model topics' evolution over time, we make use of the topics that were discovered in the previous time steps to help in better identifying topics in the current influx of documents. We decompose the current influx of documents using the topics discovered in the previous time step as follows:

$$\mathbf{X}^t \approx \mathbf{W}^t\mathbf{M}_T^t\mathbf{H}^{t-1}. \tag{3}$$

Here, $\mathbf{H}^{t-1}$ is a matrix of topics discovered in the previous time step. The product $\mathbf{M}_T^t\mathbf{H}^{t-1}$ can be thought of

explaining the current topics $\mathbf{H}^t$ as a linear combination of the previous topics. $\mathbf{M}_T^t$ is the *topic evolution* matrix. An $\mathbf{M}_T^t$ matrix close to identity (or a permuation of it) tells us that the topics have not changed much from the previous to current time step. We delve into analyzing this matrix, and hence the stabiltiy of topics (and communities) in future sections.

We also add a component of monitoring communities over time. Similar to Equation 3, we model the current set of documents with respect to the previous communities as follows:

$$\mathbf{X}^t \approx \mathbf{W}^t \mathbf{M}_C^t \mathbf{G}^{t-1}, \tag{4}$$

where $\mathbf{M}_C^t$ is the community evolution matrix.

The crux of our loss function is formed by putting together Equations 1 through 4. Our variables are $\mathbf{W}^t$, $\mathbf{H}^t$, $\mathbf{G}^t$, $\mathbf{M}_T^t$ and $\mathbf{M}_C^t$. The optimization is performed one time step after another. Hence, $\mathbf{H}^{t-1}$ and $\mathbf{G}^{t-1}$ are *known* to us by time t. We decompose our loss function into the following components,

$$L = \mu L_T + (1-\mu) L_C + \mathrm{R}, \tag{5}$$

where L_T and L_C are the topic and community parts of the objective function and R encompasses the regularization terms. We impose l_1 regularization on $\mathbf{W}^t$, $\mathbf{H}^t$, $\mathbf{G}^t$ and both the evolution matrices $\mathbf{M}_T^t$ and $\mathbf{M}_C^t$ to promote sparsity. In order to drive the loss function more towards either topic modality or the community modality of the objective, we use a parameter $\mu \in [0,1]$. $\mu = 0$ places full weight on the community part and $\mu = 1$ places full weight on the topic part.

The topic part and the community part of the objective, and the regularization terms can be written as:

$$L_T = ||\mathbf{X}^t - \mathbf{W}^t\mathbf{H}^t||_F^2 + ||\mathbf{X}^t - \mathbf{W}^t\mathbf{M}_T^t\mathbf{H}^{t-1}||_F^2, \tag{6}$$

$$L_C = ||\mathbf{U}^t - \mathbf{W}^t\mathbf{G}^t||_F^2 + ||\mathbf{U}^t - \mathbf{W}^t\mathbf{M}_C^t\mathbf{G}^{t-1}||_F^2, \tag{7}$$

$$\begin{aligned}\mathrm{R} = \alpha(&||\mathbf{W}^t||_1 + ||\mathbf{H}^t||_1 + ||\mathbf{G}^t||_1 + ||\mathbf{M}_T^t||_1 \\ &+ ||\mathbf{M}_C^t||_1) + \lambda(||\mathbf{M}_T^t - I||_F^2 + ||\mathbf{M}_C^t - I||_F^2).\end{aligned} \tag{8}$$

We add a term $\lambda||\mathbf{M}^t - \mathbf{I}||_F^2$ which, depending on the value of $\lambda \in \{0, \infty\}$ controls how much importance is placed on the past and the present. A large value of λ places much weight on the past and vice versa. The role of parameters λ and μ are analyzed in detail in Section 5.

3.2 The Optimization

We minimize the loss function L as shown below:

$$\{\mathbf{W}_*^t, \mathbf{H}_*^t, \mathbf{G}_*^t, \mathbf{M}_{T,*}^t, \mathbf{M}_{C,*}^t\} = \mathbf{argmin} \underset{\mathbf{W}^t,\mathbf{H}^t,\mathbf{G}^t,\mathbf{M}_T^t,\mathbf{M}_C^t}{L}. \tag{9}$$

Note the variables with respect to which we optimize L. Of these variables, the one that is most useful for evaluation purposes is the matrix $\mathbf{H}^t$. This is a matrix of word distributions for each topic. We compare the top-10 words from each topic in $\mathbf{H}^t$ to the top-10 obtained from the groundtruth. More details about groundtruth and evaluation are provided in Section 5.

The optmization problem in Equation 9 is not convex in all the parameters simultaneously. We use multiplicative updates as in [12]. For the loss function in Equation 9, we derive the gradients with respect to each variable as:

$$\begin{aligned}\nabla_{\mathbf{W}^t} L = \mathbf{W}^t(&\mathbf{H}^t\mathbf{H}^{t^T} + \mathbf{G}^t\mathbf{G}^{t^T} \\ &\mathbf{M}_T^{t^T}\mathbf{H}^{t-1^T}\mathbf{H}^{t-1}\mathbf{M}_T^t + \mathbf{M}_C^{t^T}\mathbf{G}^{t-1^T}\mathbf{G}^{t-1}\mathbf{M}_C^t) \\ &- (\mathbf{X}^t\mathbf{H}^{t^T} + \mathbf{X}^t\mathbf{H}^{t-1^T}\mathbf{M}_T^{t^T} + \mathbf{U}^t\mathbf{G}^{t^T} \\ &+ \mathbf{U}^t\mathbf{G}^{t-1^T}\mathbf{M}_C^{t^T} - \alpha\mathbf{e}\mathbf{e}^T),\end{aligned} \tag{10}$$

$$\nabla_{\mathbf{H}^t} L = \mathbf{W}^{t^T}\mathbf{W}^t\mathbf{H}^t - (\mathbf{W}^{t^T}\mathbf{X}^t - \alpha\mathbf{e}\mathbf{e}^T), \tag{11}$$

$$\nabla_{\mathbf{G}^t} L = \mathbf{W}^{t^T}\mathbf{W}^t\mathbf{G}^t - (\mathbf{W}^{t^T}\mathbf{U}^t - \alpha\mathbf{e}\mathbf{e}^T), \tag{12}$$

$$\begin{aligned}\nabla_{\mathbf{M}_T^t} L =&(\mathbf{H}^t\mathbf{H}^{t^T})\mathbf{M}_T^{t^T}(\mathbf{W}^{t^T}\mathbf{W}^t) + \lambda\mathbf{M}_T^{t^T} \\ &- (\mathbf{H}^t\mathbf{X}^{t^T}\mathbf{W}^t + \lambda\mathbf{I} - \alpha\mathbf{e}\mathbf{e}^T),\end{aligned} \tag{13}$$

$$\begin{aligned}\nabla_{\mathbf{M}_C^t} L =&(\mathbf{G}^t\mathbf{G}^{t^T})\mathbf{M}_T^{t^T}(\mathbf{W}^{t^T}\mathbf{W}^t) + \lambda\mathbf{M}_C^{t^T} \\ &- (\mathbf{G}^t\mathbf{U}^{t^T}\mathbf{W}^t + \lambda\mathbf{I} - \alpha\mathbf{e}\mathbf{e}^T),\end{aligned} \tag{14}$$

where $\mathbf{e} = [1, 1, \ldots, 1]$. From the Karush Kuhn Tucker first order conditions, we have the primal feasibility as:

$$\mathbf{W}^t \geq \mathbf{0}, \mathbf{H}^t \geq \mathbf{0}, \mathbf{G}^t \geq \mathbf{0}, \mathbf{M}_T^t \geq \mathbf{0} \text{ and } \mathbf{M}_C^t \geq \mathbf{0}, \tag{15}$$

the stationarity condition as $L(\mathbf{W}^t, \mathbf{H}^t, \mathbf{G}^t, \mathbf{M}_T^t, \mathbf{M}_C^t) = 0$, at the minimizers, $\mathbf{W}^{t*}, \mathbf{H}^{t*}, \mathbf{G}^{t*}, \mathbf{M}_T^{t*}\mathbf{M}_C^{t*}$, and the complementary slackness:

$$\begin{aligned}\nabla_{\mathbf{G}^t} L \odot \mathbf{G}^t = \mathbf{0},\ \nabla_{\mathbf{H}^t} L \odot \mathbf{H}^t = \mathbf{0}, \\ \nabla_{\mathbf{M}_C^t} L \odot \mathbf{M}_C^t = \mathbf{0},\ \nabla_{\mathbf{M}_T^t} L \odot \mathbf{M}_T^t = \mathbf{0}, \\ \nabla_{\mathbf{W}^t} L \odot \mathbf{W}^t = \mathbf{0}.\end{aligned} \tag{16}$$

The update equations are derived by substituting the gradients (Equations 10 - 14) in the first order conditions (Equation 16) as below:

$$\begin{aligned}&\mathbf{W}^t \leftarrow \mathbf{W}^t \odot \frac{N}{D}, \text{ where} \\ &N = (\mathbf{X}^t\mathbf{H}^{t^T} + \mathbf{X}^t\mathbf{H}^{t-1^T}\mathbf{M}_T^{t^T} + \mathbf{U}^t\mathbf{G}^{t^T} + \mathbf{U}^t\mathbf{G}^{t-1^T}\mathbf{M}_C^{t^T} \\ &\quad - 2\alpha\mathbf{e}\mathbf{e}^T), \\ &D = \mathbf{W}^t(\mathbf{H}^t\mathbf{H}^{t^T} + \mathbf{G}^t\mathbf{G}^{t^T} + \mathbf{M}_T^{t^T}\mathbf{H}^{t-1^T}\mathbf{H}^{t-1}\mathbf{M}_T^t \\ &\quad + \mathbf{M}_C^{t^T}\mathbf{G}^{t-1^T}\mathbf{G}^{t-1}\mathbf{M}_C^t),\end{aligned} \tag{17}$$

$$\mathbf{H}^t \leftarrow \mathbf{H}^t \odot \frac{(\mathbf{W}^{t^T}\mathbf{X}^t - \alpha\mathbf{e}\mathbf{e}^T)}{\mathbf{W}^{t^T}\mathbf{W}^t\mathbf{H}^t}, \tag{18}$$

$$\mathbf{G}^t \leftarrow \mathbf{G}^t \odot \frac{(\mathbf{W}^{t^T}\mathbf{U}^t - \alpha\mathbf{e}\mathbf{e}^T)}{\mathbf{W}^{t^T}\mathbf{W}^t\mathbf{G}^t}, \tag{19}$$

$$\mathbf{M}_T^t \leftarrow \mathbf{M}_T^t \odot \frac{(\mathbf{H}^{t-1}\mathbf{X}^{t^T}\mathbf{W}^t + \lambda\mathbf{I} - \alpha)}{(\mathbf{H}^{t-1}\mathbf{H}^{t-1^T})\mathbf{M}_T^{t^T}(\mathbf{W}^{t^T}\mathbf{W}^t) + \lambda\mathbf{M}_T^{t^T}}, \tag{20}$$

$$\mathbf{M}_C^t \leftarrow \mathbf{M}_C^t \odot \frac{(\mathbf{G}^{t-1}\mathbf{U}^{t^T}\mathbf{W}^t + \lambda\mathbf{I} - \alpha)}{(\mathbf{G}^{t-1}\mathbf{G}^{t-1^T})\mathbf{M}_C^{t^T}(\mathbf{W}^{t^T}\mathbf{W}^t) + \lambda\mathbf{M}_C^{t^T}}. \tag{21}$$

Theorem 1 *The loss function L in Equation (5) is non increasing under the update rules in Equations (17), (18), (19), (20), and (21). The loss function L is invariant under these updates if and only if $\mathbf{H}^t$, $\mathbf{G}^t$, $\mathbf{M}_T^t$ and $\mathbf{M}_C^t$ are at a stationary point of the function.* The proof for update rules on $\mathbf{H}^t$ and $\mathbf{G}^t$ comes directly from [13]. For the update rules

$\mathbf{M}_T^t$ and $\mathbf{M}_C^t$ a related formal proof is given in [26]. Finally, the proof for update rules on $\mathbf{W}^t$ follows from [13]. Due to the lack of space, its proof will be provided in an extended version of the paper.

4. DATA SET DESCRIPTION

Data: $z \in \mathbb{N}_{>0}$; $\mathbf{X}^{(t)}$,$\mathbf{X}^{(t+1)}$,..., $\mathbf{X}^{(t+z)}$; $\mathbf{U}^{(t)}$,$\mathbf{U}^{(t+1)}$, ..., $\mathbf{U}^{(t+z)}$; $\mathbf{T}^{(t)}$,$\mathbf{T}^{(t+1)}$,..., $\mathbf{T}^{(t+z)}$; We assume that we track a set of defined hashtags so that $N_h^{(t)} = N_h^{(t+1)} = \cdots = N_h^{(t+z)}$, and in particular that all the rows of $\mathbf{T}$ matrices are equal.

Result: Two vectors $\mathbf{v}_x$,$\mathbf{v}_u$ of size $N_h^{(t)}$ containing for each hashtag a stability score over the content matrix $\mathbf{X}$ and user interaction matrix $\mathbf{U}$.

forall the $i \in 1, 2, \ldots, N_h$ **do**
  $s_x = 0$;
  $s_u = 0$;
  forall the $a \in t, t+1, \ldots, t+z-1$ **do**
    s_x += $cos((\mathbf{T}^{(a)}\mathbf{X}^{(a)})\mathbf{e}_i^T, (\mathbf{T}^{(a+1)}\mathbf{X}^{(a+1)})\mathbf{e}_i^T)$;
    s_u += $cos((\mathbf{T}^{(a)}\mathbf{U}^{(a)})\mathbf{e}_i^T, (\mathbf{T}^{(a+1)}\mathbf{U}^{(a+1)})\mathbf{e}_i^T)$;
  end
  $\mathbf{v}_x[i] = \frac{s_x}{z}$;
  $\mathbf{v}_u[i] = \frac{s_u}{z}$;
end

Algorithm 1: Hashtag stability scores

In order to accurately answer the research questions posed in Section 1, and evaluate our algorithm we need a dataset where the topics persist over a period of time, and also has a social community that accompanies it. We use the public dataset which was released in 2013 [20] consisting of all the articles published from 80 international news sources such as CNN, BBC, Aljazeera during a period of 14 days. Each news article consists of the textual content of the article (via `html`) and a list of all tweets which link to that article over a period 12 hours from the article's publication. The tweets containing links to the news articles were also collected. From the tweets, two features were extracted: the author of the tweet and the hashtags present in the tweet. The information about the author of the tweet was used to detect the community information. The hashtags in the tweets were used as the groundtruth topic of the document which the tweet linked to [1].[2] Most of the articles were associated to a hashtag. We discarded the ones which did not correspond to any hashtag. Since we wish to *track* topics over a period of time, we consider only those hashtags that appear every day (and thereby pruning the number of articles even further). Moreover, to avoid data sparsity of articles, we keep only those hashtags that have at least five articles per day. After all the filtering, we end up with 33,387 articles (from an original set of 53,784) associated to 384 different hashtags. For details about data acquisition, refer to [20].

4.1 Stability of Tags

Recall that we use the hashtags as the groundtruth topic of the text document. Keeping in mind our research questions from Section 1, we wanted to detect the three different categories of hashtags for each dataset. The first category of

[2]We hereby use the words topic and hashtag somewhat interchangeably.

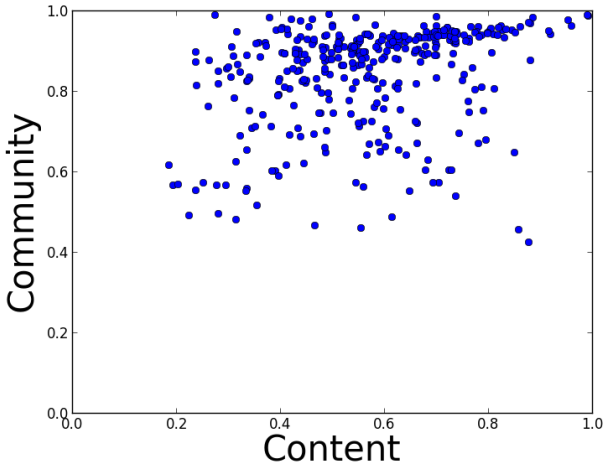


Figure 1: This figure illustrates the stability of hashtags in terms of content and community information. Each dot in the figure is a hashtag. The x-axis and y-axis represent content and community stability. The content and community stability scores are calculated according to Algorithm 1.

hashtags are those that are stable in terms of content, but relatively unstable in terms of community; meaning that the content corresponding to these hashtags does not evolve much over the period of interest, but the community of users who tweet about these hashtags evolves quite a bit. We call this set *content stable* hashtags. These are the supposedly 'easier' topics where one may expect that the presence of community may not particularly help in better topic discovery. The second category of hashtags are those that are stable in terms of their community, but the content evolves a lot; meaning that the community of users that show an interest on these tags stays relatively unchanged over a period of time, but the actual content (in terms of vocabulary) changes a lot. We call this set *community-stable* hashtags. These are the supposedly more difficult topics where using the content alone may yield only poor performance, since by definition the content is not very stable. The third category of hashtags are those that are stable in terms of both content and community, called *mixed-stable* hashtags. Intuitively, we posit that our model would work particularly well in discovering and monitoring those topics which have a stable community of active users over the period of interest, but have a content which is evolving a lot (these are community-stable hashtags).

Following the notation specified in Section 3, we explain how we determine the tags which fall into the content-stable, community-stable, and mixed-stable categories. Let us consider matrix $\mathbf{T}^{(t)}$ of size $N_h^{(t)} \times N_d^{(t)}$, where $N_h^{(t)}$ is the number of hashtags produced at time t and $N_d^{(t)}$ is the number of documents arriving at time t. In particular, $\mathbf{T}^{(t)}(i,j) = 1$ if the document j has been mentioned in a tweet that contains the hashtag i, and 0 otherwise. Algorithm 1 explains how we compute a stability score for each hashtag in terms of their content and community. The essence of Algorithm 1 is as follows: by simply averaging all the documents belonging to a particular hashtag, we obtain a representation for each hashtag in terms of features extracted from the documents. Following this procedure, each hashtag can consists

of a (centroid) vector of N_f entries (i.e., in a bag-of-words representation). We compare this representation with a similar representation obtained for the same hashtag at the next time step using cosine similarity. We then average all the similarities obtained across the consecutive time steps for each hashtag.

Refer to Figure 1. Note that, in such a figure, the point $(1, 0)$ represents perfect content stability, and zero community stability. To determine the set of hashtags that belong to the *content stable* set, we calculate the Euclidean distance between $(1, 0)$ and all the other hashtags, and rank them in the increasing order of their distances. Likewise for *community stable* and *mixed stable* sets (using respectively the Euclidean distances to points $(0, 1)$ and $(1, 1)$). Some examples of content stable hashtags are `#football` and `#h7n9`; community stable hashtags, `#celeb` and `#gossip` and mixed stable hashtags `#alarabiya` and `#forbes`.

5. EXPERIMENTS

Recall that our goal in this work is to gain a better understanding of when the social context surrounding the documents actually improve topic discovery. Hence, in this section, our primary focus is to provide a quantitative and qualitative answers to each of the research questions posed in the introduction. Section 5.1 provides an overview about the baseline algorithms, details how we implement them, and how we use them in our problem setting. Section 5.2 provides details about how the groundtruth topics are obtained. In addition, it also explains how the topics detected by our algorithm and each of the baselines are compared with the groundtruth topics, and what metrics are used for the evaluations. Then, each of the subsequent subsections are dedicated to answering one or more research questions.

5.1 Baselines

We evaluate our algorithm with several baselines. Our baselines can be divided into two categories; one which focuses on modeling topic evolution, and another which aims to incorporate link information into topic modeling.

Link-PLSA-LDA [16] is an algorithm which uses both the content and link information (but does not have a temporal aspect incorporated in it). The link structure is built from the citation network of the documents. The algorithm combines LDA and PLSA into a single framework and in addition, models the topical relationship between the citing document and the cited document.[3] The inference is carried out by employing mean-variation approximation of the latent variables. To implement this algorithm, we used the code developed by the authors which is available publicly.[4] As input, the algorithm requires a list of documents in a bag-of-words format, and a matrix of links between the documents. Producing bag-of-words for each document is straightforward. For the link information, we assume that a link exists between two articles if they have a common user sharing or posting the article. This information was essentially derived from the $\mathbf{U}$ matrix in Section 4. Each fresh inflow of documents is considered as a separate problem as the model was not developed to connect topics temporally.

Collective Matrix Factorization (CMF) [22, 8] Broadly speaking, the concept of collective matrix factorization has been used in several applications including recommendation systems, producing hashing functions for images, coclustering etc. In this scenario, we will use CMF to incorporate both the social and textual aspect of the objective in Equation 5 but not its temporal aspect. We compare our method to this baseline to show that using the temporal information of tracking the textual content and community helps improve performance.

Online-LDA [2] is an algorithm which monitors topic evolution, in that it utilizes the information about topics detected in the previous time steps, but does not accommodate for the link structure between the documents. We implemented Online-LDA based on the original LDA code developed by David Blei[5] (as suggested by the authors of [2]). The authors of [2] had found that using the topics detected in the previous time step produced the most improvement in performance, and suggested that using the topics from earlier time steps produced only marginal improvements. We tested this baseline in a similar setting as well and used only the topics in the previous timestep to discover current topics. In essense, implementing Online-LDA boils down to setting the prior on the topics according to the topic distribution discovered in the previous time step.

Joint Past Present Decomposition (JPP) [26] models also the topic evolution, but as Online-LDA, is blind to the social context surrounding the input documents. Our method, LTECS, reduces to JPP when $\mu = 1$. We used the code provided by the authors.[6]

5.2 Evaluation, Groundtruth and Experimental Setup

We evaluate all the algorithms by comparing a ranking of the top-10 words obtained by each algorithm, and a ranking of the top-10 words obtained by the groundtruth. It has been shown that the group of top-10 words indeed give us a good insight about the topic [24, 17]. All the algorithms considered here including the baselines and LTECS discover topics by directly producing a distribution over words. In terms of mapping the discovered topics to the groundtruth topics, we calculated the cosine similarity between each of the discovered topics and the groundtruth topics. Each discovered topic was then mapped to the most similar groundtruth topic. We borrow this procedure procedure from other state of the art experiments in topic evolution [21]. The distributions produced in each case are discrete and can be used to pick the top-10 words in each topic and to produce a ranking.

We now delve a bit more into how the groundtruth is obtained. On Twitter, hashtags are a sequence of nonwhitespace characters which follow the # sign. It is popular convention on Twitter to embed a hashtag in a tweet to give it context. And as in several studies in the past, this context is used as the groundtruth topic annotations for the news articles whose links are embedded in the tweet [1]. The hashtags for each of the three categories, content stable, community stable, and mixed stable were identified as explained in Section 4.1.

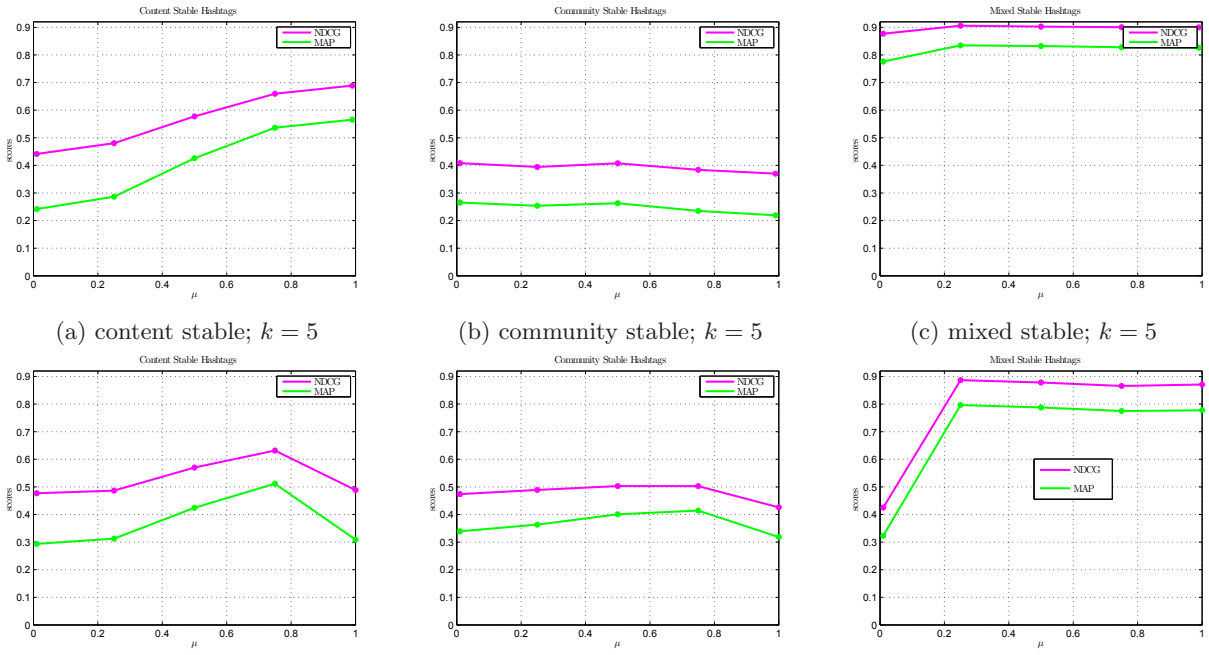
The way we calculate the actual groundtruth topic distribution is that, at each time step, the $\mathbf{T}^{(t)}$ matrix (refer to Section 4 for notation) is premultiplied by $\mathbf{X}^{(t)}$ to obtain a resulting matrix of raw word counts for each topic. Premul-

[3]While we do not explicitly compare to [10], the authors of Link-PLSA-LDA compare their own work to former and claim better performance.

[4]https://sites.google.com/site/rameshnallapati/software

[5]http://www.cs.princeton.edu/ blei/lda-c/

[6]https://github.com/amantrac/TopicDiscoveryJPP



(a) content stable; $k = 5$ (b) community stable; $k = 5$ (c) mixed stable; $k = 5$

(d) content stable; $k = 15$ (e) community stable; $k = 15$ (f) mixed stable; $k = 15$

Figure 2: This figure illustrates the effect of the importance parameter, μ on the performance. Refer to Equation 5. A high value of μ places more weight on the topic part of the objective and less weight on the community part of the objective, and vice versa.

tiplication of $\mathbf{T}^{(t)}$ by $\mathbf{X}^{(t)}$ basically yields the average word distribution within each hashtag. Once this is obtained, the highest weighted 10 words form our groundtruth ranking. We use Normalized Cumulative Discounted Gain (NDCG) metric, and the Mean Average Precision (MAP) metric to compare the rankings obtained by each algorithm to the groundtruth. We have performed experiments by considering the best $5, 10, 15$ and 20 topics for each category.

We now give details about the experimental setup. Our objective function is optimized iteratively using the multiplicative update equations (Equations 17 - 21) in Section 3. The variables $\mathbf{W}^t, \mathbf{H}^t, \mathbf{G}^t, \mathbf{M}_T^t, \mathbf{M}_C^t$ were given a random non-negative initialization. The parameters were tuned on all the data. In both datasets, the data spans for 14 days, and hence the topic discovery results that we obtained are averages of the results obtained over that time period. The $l1$ normalization parameter for CMF, the JPP model and the LTECS model were set to 0.05. The λ parameter was tuned for values of $\{10, 100, 10^3 \ldots 10^7\}$. It was consistently observed that the algorithm yielded good performance for $\lambda = 10^7$ [7]. We tuned for different values of $\mu \in \{0.01, 0.25, 0.5, 0.75, 1\}$ and picked the one which gives the best performance. We delve more into the analysis for μ in subsequent sections. For the baselines, all the parameters were tuned and set internally.

5.3 Social Information Vs Textual Content: the Trade-off

The μ parameter from Equation 5 allows to bias the objective function more towards one of the modalities, if so desired. A high value of μ biases importance to the content part of the objective and vice versa. In fact, LTECS reduces to JPP when $\mu = 1$ and hence JPP can never outperform LTECS. This section delves into investigating how the trade-off between using content and social information actually functions on both datasets.

For the case of *community stable* hashtags, the best performances were achieved for $0.01 \leq \mu \leq 0.5$ (Table 1). This implies that when a lot of importance was placed on the social context part of the objective, better were the topics that were detected. Refer to Figure 2b and 2e. These figures illustrate how the performance varies as the value of μ moves from 0 to 1. Note that highest performance is achieved when $\mu \leq 0.5$.

While considering *content stable* hashtags, we will focus on LTECS and JPP in Table 1. For $k = 5$ and 10, we observe best performances by *both* JPP and LTECS method. In other words, LTECS algorithm exhibited best performance when the objective contained only the content part with $\mu = 1$. This suggests that for topics which have a highly focused text, we need to place all the importance on content. What is more interesting is that, it implies that even if we add a little bit of the social context information to the objective, it actually *hurts* performance. Let us contrast this result to what happens when $k = 15$ and 20. In those scenarios, we observe that the best performance was obtained by LTECS, when μ was 0.75. For the purest 5 and 10 topics, it could be that the content of those documents were very well defined that the usage of side information actually detracted the objective from the correct path. However as the number of topics increases ($k = 15, 20$), there is perhaps more noise in the topics and we find that the use of community information indeed helps. This suggests that for the

[7]While it so happens that this value of lambda worked well for the Twitter dataset, it may not hold for other datasets. As a matter of fact, we explore this more in Section 5.5 where we try to assess the quality of topics by setting $\alpha = 0.5$ and $\lambda = 0$.

best performance one needs to know the accurate operating spot of the μ parameter. The important message from analyzing the μ trade-off for content stable and community stable hashtags is that, with very focused text, just using the content suffices. This is likely to be the case for the dominant topics (i.e. $k < 10$ in our study). On the other hand, if the text is a little noisy, the social context greatly helps in discovering better topics. This is likely to be the case when tracking more than just the top dominant topics ($k > 10$ in our case). As we argue in the introduction, in today's world, the text is more often than not quite noisy as topics are prone to being volatile and evolving very quickly. Refer to Figures 2(a) and 2(d). The figure illustrates that the best performance is acheived when $\mu \geq 0.5$.

By definition *mixed stable* hashtags have both stable content and stable communities. As it turns out, the best performance for LTECS is obtained when $0.25 \leq \mu \leq 0.75$. This suggests that when we have topics which have both stable content and community, it is necessary to give importance to both aspects. In addition, biasing only on content does not yield the best performance. In Figures 2(c) and 2(f), the best performance is achieved in the midregion of the plot.

5.4 Comparison with the state-of-the-art

In this section, we discuss how our algorithm performs in comparison to state of the art baselines introduced in Section 5.1. One of the main conclusions from analyzing the results is that, using the community information certainly helps with better topic tracking. This is a direct observation from Table 1 that the NDCG and MAP values for LTECS is higher, or at least as good as its competitors in most scenarios. The rest of this section will highlight where the sweet spot of the trade off between content and community lies, and why.

For *community stable topics*, as expected, good improvements were seen in the community stable hashtags. In these hashtags, LTECS algorithm consistently outperforms all the baselines. It is clear that the use of community information helps in better topic discovery. It is also interesting to note that the algorithm that exhibits second best performance is Link-PLSA-LDA. Hence, not learning from older topics does not particularly hurt Link-PLSA-LDA in its performance when compared to Online-LDA and JPP. This means that, for community stable topics, the algorithms that use some form of social context information or link information perform better than those that discover topics using content alone.

In the case of *content stable topics*, one may expect the baselines which focus only on the content of the documents to exhibit best performance. This is partially true. For $k = 5, 10$, JPP and LTECS outperform Link-PLSA-LDA and Online-LDA. Note that in both cases LTECS achieves the best performance only when $\mu = 1$ implying that adding community information does not help. On the other hand, for $k = 15$ and 20, LTECS achieves best performance over all the baselines. In these cases note that the value for $\mu = 0.75$. This implies that adding the community information actually helps. We already discussed this behavior in Section 5.3. Another thing to note here is that Link-PLSA-LDA some what consistently ranks last. This is because Link-PLSA-LDA, unlike LTECS lacks the tuning parameter μ which can seamlessly shift the focus of the objective between topic and community. It perhaps places equal weight on both, and hence fails to make the appropriate tradeoff.

For *mixed stable hashtags*, the performance is in generally very good for all the hashtags. This becomes clear when we compare the performance metrics of content stable and community stable to mixed stable. There is a noticeable jump in the average NDCG and MAP scores. This implies that, if a hashtag has both well focused content, and a dedicated community of users, detecting those topics are much easier.

5.5 Learning Stability of Topics

In this section, we investigate to what extent can our algorithm learn the type of topics present in the documents; i.e., are the documents more content stable, community stable or mixed stable. And we certainly do not want to be able to bias the objective function more towards one of the modalities. Hence, for all experiments in this section, we set $\mu = 0.5$. Recall that our loss function (Equation 5) is built such that $\mathbf{H}^t \approx \mathbf{M}\mathbf{H}^{t-1}$. The proposed model encourages for stability of topics and communities by regularizing the evolution matrices $\mathbf{M}_T^t$ and $\mathbf{M}_C^t$ through $\lambda(||\mathbf{M}_T^t - \mathbf{I}||_F^2 + ||\mathbf{M}_C^t - \mathbf{I}||_F^2)$. A high value of λ pushes the evolution matrices close to $\mathbf{I}$ which enforces the topics (and communities) to evolve very little over time. So far, we demonstrated the effectiveness of using side social information in order to discover topics on large scale dataset based on Twitter. In this section, we aim to assess the extent to which our algorithm is able to recover correctly the evolution patterns exhibited by the data by studying the evolution matrices $\mathbf{M}_T^t$ and $\mathbf{M}_C^t$ across consecutive time steps.

This raises the question of how to set λ. In presence of a groundtruth this parameter can be tuned by cross validation as we did in the previous section (where hashtags where used as proxy to build the groundtruth). However, in the real world, topical annotations are rarely available. In this context, the user can decide to use the model in an "agnostic mode" by not placing any form of prior on the evolution matrices. This is achieved by setting λ to 0. In this real world scenario, we may wonder if the model, without the help of any prior, will be able to recover the correct evolution patterns. In other words, we propose to test the extent to which the retrieved evolution matrices are close to the "real ones". To do so, we make use of the group of hashtags previously identified (Section 4) as stable and unstable at the topic and community level. To validate that the retrieved $\mathbf{M}$ matrices exhibit a temporal stability pattern which is indeed present in the data, we test if the matrix retrieved from the stable group of hashtags is closer to the identity than the one retrieved from the unstable group (for both topics and communities). In other words, for topic stable hashtags, we want $\mathbf{M}_T^t$ to exhibit more stability than $\mathbf{M}_C^t$ and for community stable hashtags, we want $\mathbf{M}_C^t$ to exhibit more stability than $\mathbf{M}_T^t$.

For the purpose of the experiments, we need to measure how close an evolution matrix $\mathbf{M}$ is to the identity $\mathbf{I}$. Or, in other words how stable is the evolution exhibited by $\mathbf{M}$. Now, we will quantify this closeness. An important point to remember now is that, many distance or similarity measures will fail to capture the notion that we are after. For example, quantifying the stability of $\mathbf{M}$ by simply calculating a cosine similarity between $\mathbf{M}$ and $\mathbf{I}$ will not work because $\mathbf{M}$ is prone to topical (and community) permutations over time. Since we are working in an unsupervised setting, what was 'topic-1' at time-t could have been discovered as 'topic-5' at time-$(t+1)$. Hence, we must make sure that the result-

Category of Topic	Metric	Model	k = 5	k = 10	k = 15	k = 20
Community	NDCG	LTECS	0.4081	**0.4800**	**0.5029**	0.5129
			$\mu = 0.01$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$
		JPP	0.3699	0.4496	0.4608	0.4138
Stable		Online-LDA	0.3903	0.4138	0.4446	0.5667
		Link-PLSA-LDA	0.3943	0.4608	0.4761	0.4925
		CMF	0.3454	0.4338	0.4771	0.4827
	MAP	LTECS	0.2653	0.3637	**0.4007**	**0.4173**
			$\mu = 0.01$	$\mu = 0.5$	$\mu = 0.5$	$\mu = 0.5$
Hashtags		JPP	0.2191	0.3596	0.3462	0.3420
		Online-LDA	0.2628	0.3160	0.3489	0.3835
		Link-PLSA-LDA	0.2704	0.3364	0.3658	0.3937
		CMF	0.2044	0.3190	0.3757	0.3665
Content	NDCG	LTECS	0.6888	0.6055	**0.6317**	**0.6623**
			$\mu = 1$	$\mu = 1$	$\mu = 0.75$	$\mu = 0.75$
		JPP	0.6888	0.6055	0.4885	0.6504
Stable		Online-LDA	0.6815	0.5988	0.6166	0.6684
		Link-PLSA-LDA	0.6574	0.5862	0.6087	0.6401
		CMF	0.5846	0.4919	0.4455	0.4327
	MAP	LTECS	**0.5655**	**0.4784**	**0.5115**	**0.5559**
			$\mu = 1$	$\mu = 1$	$\mu = 0.75$	$\mu = 0.75$
Hashtags		JPP	0.5655	0.4784	0.3089	0.5411
		Online-LDA	0.5175	0.4083	0.4555	0.5443
		Link-PLSA-LDA	0.4890	0.3817	0.4434	0.5053
		CMF	0.4423	0.3207	0.2556	0.2557
Mixed	NDCG	LTECS	0.9005	0.8868	**0.9249**	0.9089
			$\mu = 0.25$	$\mu = 0.75$	$\mu = 0.25$	$\mu = 0.25$
		JPP	0.8771	0.8762	0.4251	0.4580
Stable		Online-LDA	**0.9564**	0.9168	0.9111	0.5967
		Link-PLSA-LDA	0.8944	0.9159	0.8392	0.8975
		CMF	0.6712	0.8768	0.8905	0.8753
	MAP	LTECS	0.7783	0.7965	**0.8964**	0.8845
			$\mu = 0.25$	$\mu = 0.75$	$\mu = 0.5$	$\mu = 0.25$
Hashtags		JPP	0.7762	0. 7783	0.3232	0.3644
		Online-LDA	**0.9208**	**0.8804**	0.8841	0.4308
		Link-PLSA-LDA	0.8787	0.8379	0.7452	**0.8982**
		CMF	0.5329	0.8223	0.8499	0.8337

Table 1: Topic discovery evaluation using Normalized Cumulative Discounted Gain and Mean Average Precision metrics for all three categories of hashtags. k stands for the number of topics. λ was set to 10^7, and α was set to 0.05 for LTECS model. All the values in bold represent significant improvement in performance (using Student-t test, $p < 0.05$).

ing definition of stability is invariant to such topical (and community) permutations.

To quantify stability, first note that, through the primal feasibility conditions (Equation 15), we have $\mathbf{M}_T^t \geq \mathbf{0}$, and $\mathbf{M}_C^t \geq \mathbf{0}$. Therefore, when we apply $l1$ normalization to the row or column of the $\mathbf{M}$ matrices, we obtain stochastic matrices. Also, recall that the largest eigenvalue of a stochastic matrix is 1. We now define the stability score for the evolution matrix $\mathbf{M}$ as follows:

Definition 1. *Let M be a stochastic matrix obtained after $l1$ normalization of the evolution matrix $\mathbf{M}$. The* stability *of M is defined as:*

$$stability(M) := \frac{\sum_i abs(\gamma_i)}{n}, \tag{22}$$

where $\{\gamma_i\}$s are the eigenvalues of M, and n is the number of rows (and of columns) of M.

We make some observations about this definition. The $stability(M)$ takes value between $[0, 1]$, since none of the individual $abs(\gamma_i)$s can exceed 1. The matrix representing perfect stability would be $\mathbf{I}$ or a permutation of $\mathbf{I}$ (due to possible topical shifts between two consecutive time steps). A matrix M has $stability(M) = 1 \iff M = \mathbf{I}$ or a permutation of $\mathbf{I}$.

Through Definition 1, we investigate if the model can recover the temporal topic and community stability patterns. We evaluate this for each category of hashtags by calculating the stability(M) for the $\mathbf{M}_T^t$ and $\mathbf{M}_C^t$ calculated in each time step, and producing an average value. $stability(M)$ is calculated through two ways: by calculating the left and right eigen values of the $\mathbf{M}$ matrices. We confirm that the model can recover a more stable temporal matrix for topics than for communities when processing hashtags with topical stability (Figure 3, left). While when processing hashtags that are community stable, the model recovers a more stable temporal community matrix (Figure 3, right). We are thereby able to see that using such stability analysis of the evolution matrices, one can study the nature of the text corpora when there is no prior knowledge. This will actually help the user determine a value for μ.

6. CONCLUSION

The goal of our work was to gain a better understanding of when social context helps in modeling topic evolution. In order to achieve this, we proposed a matrix factorization based approach which takes into account both the content of the documents and their social context. We found that, depending on the kind of topic, there is a clear trade off between the content and community. The content of the document suffices if the text of the topic is very focused, and evolves little over time. As we begin to move away from this scenario to consider documents that have a richer and more variable vocabulary, we find that the use of social context begins to help greatly. We were also able to show that our model can learn the kind of topics at hand; i.e., whether they are content stable, community stable, or both.

This work predominantly considered the user interactions of the documents as the social context. In the same spirit, one could explore what it means to consider other types of contexts like geographical location of the user (or docu-

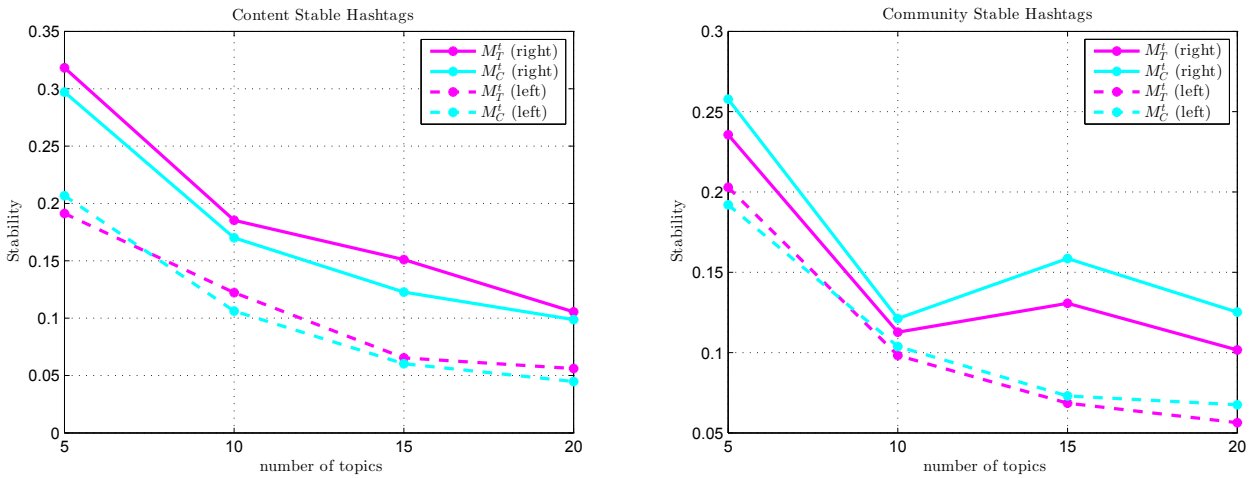


Figure 3: This figure plots the stability of $\mathbf{M}_C^t$ and $\mathbf{M}_T^t$. We note that in (a) $\mathbf{M}_T^t$ matrix shows higher stability than the $\mathbf{M}_C^t$ matrix, and in (b) $\mathbf{M}_C^t$ shows higher stability than $\mathbf{M}_C^t$, thus confirming that we are indeed able to learn the stability through our algorithm.

ment), and also perhaps delve more into the user profiles and incorporate information about age, gender and demographics to give a well rounded view of the social context. We hope to be able to work on these aspects in the future.

7. ACKNOWLEDGEMENTS

J.K. and G.R.G.L. acknowledge support from Yahoo! Inc., and the NSF (grants CCF-0830535 and IIS-1054960). G.R.G.L. acknowledges support from the Alfred P. Sloan Foundation. D.S is funded by the EC SUPER (FP7-606853) project.

8. REFERENCES

[1] What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *WSDM*, pages 643–652. ACM, 2012.

[2] Loulwah AlSumait, Daniel BarbarÃą, and Carlotta Domeniconi. Online-lda. ICDM '08, 2008.

[3] Hila Becker, Mor Naaman, and Luis Gravano. Event identification in social media. In *WebDB*, 2009.

[4] David M. Blei and Michael I. Jordan. Modeling annotated data. SIGIR '03, 2003.

[5] David M. Blei and John D. Lafferty. Dynamic topic models. ICML '06, 2006.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[7] Jonathan Chang and David M Blei. Relational topic models for document networks. In *AISTATS*, 2009.

[8] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR 2014*, pages 2083–2090, June 2014.

[9] Khalid El-Arini, Min Xu, Emily B. Fox, and Carlos Guestrin. Representing documents through their readers. KDD '13, 2013.

[10] Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5220–5227, 2004.

[11] Noriaki Kawamae. Trend analysis model: Trend consists of temporal words, topics, and timestamps. WSDM '11, 2011.

[12] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[13] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2000.

[14] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, March 2010.

[15] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Int. Res.*, 30(1):249–272, October 2007.

[16] Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. KDD '08, 2008.

[17] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Association for Computational Linguistics*, 2010.

[18] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML-11*, 2011.

[19] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. UAI '04, 2004.

[20] Diego Saez-Trumper, Carlos Castillo, and Mounia Lalmas. Social media news communities: Gatekeeping, coverage, and statement bias. CIKM '13, 2013.

[21] Ankan Saha and Vikas Sindhwani. Learning evolving and emerging topics in social media: A dynamic nmf approach with temporal regularization. WSDM '12, 2012.

[22] Martin Saveski and Amin Mantrach. Item cold-start recommendations: Learning local collective embeddings. RecSys '14, 2014.

[23] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In *ICWSM*, 2009.

[24] Y. Sekiguchi, H. Kawashima, H. Okuda, and M. Oku. In *MDM 2006*.

[25] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. KDD '08, 2008.

[26] Carmen K. Vaca, Amin Mantrach, Alejandro Jaimes, and Marco Saerens. A time-based collective factorization for topic discovery and monitoring in news. WWW '14, 2014.

[27] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. KDD '06, 2006.

[28] Yu Wang, Eugene Agichtein, and Michele Benzi. Tm-lda: Efficient online modeling of latent topic transitions in social media. KDD '12, 2012.