TM-LDA: Efficient Online Modeling of Latent Topic Transitions in Social Media

Yu Wang
Emory University
yu.wang@emory.edu

Eugene Agichtein
Emory University
eugene@mathcs.emory.edu

Michele Benzi
Emory University
benzi@mathcs.emory.edu

ABSTRACT

Latent topic analysis has emerged as one of the most effective methods for classifying, clustering and retrieving textual data. However, existing models such as Latent Dirichlet Allocation (LDA) were developed for static corpora of relatively large documents. In contrast, much of the textual content on the web, and especially social media, is temporally sequenced, and comes in short fragments, including microblog posts on sites such as Twitter and Weibo, status updates on social networking sites such as Facebook and LinkedIn, or comments on content sharing sites such as YouTube. In this paper we propose a novel topic model, Temporal-LDA or TM-LDA, for efficiently mining text streams such as a sequence of posts from the same author, by modeling the topic transitions that naturally arise in these data. TM-LDA learns the transition parameters among topics by minimizing the prediction error on topic distribution in subsequent postings. After training, TM-LDA is thus able to accurately predict the expected topic distribution in future posts. To make these predictions more efficient for a realistic online setting, we develop an efficient updating algorithm to adjust the topic transition parameters, as new documents stream in. Our empirical results, over a corpus of over 30 million microblog posts, show that TM-LDA significantly outperforms state-of-the-art static LDA models for estimating the topic distribution of new documents over time. We also demonstrate that TM-LDA is able to highlight interesting variations of common topic transitions, such as the differences in the work-life rhythm of cities, and factors associated with area-specific problems and complaints.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

General Terms

Algorithm, Experimentation

Keywords

topic transition modeling, temporal language models, mining social media data

1. INTRODUCTION

Latent semantic topic analysis has emerged as one of the most effective methods for classifying, clustering, and retrieving textual data. Many latent topic modeling methods have been developed and intensively studied, such as probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA). These models are designed to analyze static collections of documents. The recent proliferation of social media textual data, such as microblog posts, Facebook status updates, or comments on YouTube, brings new challenges to these models: (1) to model and analyze latent topics in social textual data; (2) to adaptively update the models as massive amounts of social content stream in; (3) to facilitate temporal-aware applications of social media, such as predicting future trends and learning social behavioral patterns.

Semantic analysis of social content has been intensively investigated in recent years. It has been shown that information derived from social media, such as Twitter, can help event detection [17], news recommendation [1] and public health applications [14]. Yet, the models can be significantly enriched by directly considering the *temporal sequence* of the topics in the microblog post stream, instead of treating it as a "static" corpus.

Consider a microblog post stream of an author. The timestamp of each post determines the order of it along the timeline. Since microblog posts reflect activities or status of the author, the temporal order of posts reflects the time dimension of the author's behavioral patterns. Thus the temporal sequence of an author's post stream is a factor connecting post content and one's real life activities. Users' posts include a variety of topics and rich information, such as breaking news, their comments on popular events, daily life events and social interaction. Clearly, the topics of post streams will not be static, but change over time. In other words, users tend to post about different topics instead of simply repeat previous posts. This observation implies that to better model the dynamic semantics of microblog post streams, we need a temporally-sensitive model that can capture the changing pattern among topics. The implications of better modeling topic dynamics reach far beyond microblogging sites such as Twitter and Weibo, as most social textual data are naturally sequenced by time. Better understanding and modeling of the temporal dynamics of social content can benefit not only these applications, but also provide powerful analytical tools for researchers and analysts.

In this paper, we propose *Temporal Latent Dirichlet Allocation* (TM-LDA) to model topic transitions in temporally-sequenced documents. In the case of microblog post streams,

we claim that topic transitions of an author's posts follow certain social behavioral patterns. For example, people tend to talk about the topic "Drink" after "Food", which implies a certain dietary and social manner. In some cities, users complain about "Traffic" mostly after they post about "Places", which reflects poor traffic condition in those areas. Understanding these topic transition could be meaningful in several ways:

- Dynamically predicting future trends of a microblog post stream based on the previous posts.

- Providing a tool for researchers and analysts to get a more in-depth view of temporal relationships among social phenomena reflected by popular topics in microblog posts.

- Providing a signal of unusual events when topics fail to follow expected transitions.

TM-LDA is designed to learn the topic transition parameters from historical temporally-sequenced documents to predict future topic distributions of new documents over time. TM-LDA takes pairs of consecutive documents as input and finds the optimal transition parameters, which minimize the least square error between predicted topic distribution and the actual topic distribution of the new documents. Additionally, transition parameters among topics can vary over time because of the changing popularity of certain topics and external events. To adaptively update the transition parameters as new documents stream in, we propose an efficient algorithm to adjust the transition parameters by replacing outdated document pairs with new ones.

The main contribution of this paper are:

- We propose a novel temporally-aware topic language model, TM-LDA, which captures the latent topic transitions in temporally-sequenced documents. (Section 2).

- We design an efficient algorithm to update TM-LDA which enables it to operate over large-scale data. Section 3 illustrates the details of the updating algorithm, and provides the complexity analysis.

- Our evaluation of TM-LDA against the static topic modeling method (LDA) on 30 million microblog posts shows that TM-LDA consistently outperforms LDA in predicting the topic distribution of future posts. (Section 4).

2. TM-LDA MODEL AND ALGORITHMS

In this section, we mathematically define TM-LDA, and discuss the way to build TM-LDA from document streams in practice.

2.1 TM-LDA Algorithm

We design TM-LDA as a system which generates topic distributions of new documents by taking previous documents as input. More precisely, if we define the space of topic distributions as $X = \{x \in \mathbb{R}_+^n : ||x||_1 = 1\}$, TM-LDA can be considered as a function $f : X \to X$. Notice that n is the dimension of the space X, in other words, n is the number of topics; $|| \cdot ||_1$ is the ℓ^1 norm of vector x. Given the topic distribution vector of a historical document x, the

estimated topic distribution of a new document $\hat{y}$ is given by $\hat{y} = f(x)$. Once we know the real topic distribution of the new document y, the prediction error of the TM-LDA system would be:

$$err_f = ||\hat{y} - y||_2^2 = ||f(x) - y||_2^2.$$

Function err_f uses the ℓ^2 norm to measure the prediction error because the minimization of err_f can thus be reduced to a least squares problem, which can be efficiently solved (Section 2.1.3). The training stage of TM-LDA is to find the function f which minimizes err_f.

In our work, the function f is designed as a non-linear mapping:

$$f(x) = \frac{xT}{||xT||_1}, \tag{1}$$

where x is a row vector, $T \in \mathbb{R}^{n \times n}$. The product of x and T is also a row vector, which is the estimated new topic weighting vector (before normalization). After xT is normalized by its ℓ^1 norm, it becomes a topic distribution vector.

In our system settings, x and y are topic distribution vectors of two consecutive posts of a user, where x represents the "old" post, and y corresponds to the "new" post. TM-LDA predicts the topic distribution of y by taking historical post x as input and applies function f on it to obtain $\hat{y}$. Therefore the prediction error of TM-LDA is the difference between $\hat{y}$ and y.

2.1.1 Error Function of TM-LDA

Function (1) defines the prediction function for a single document x. The error function is therefore:

$$err_f = \left|\left| \frac{xT}{||xT||_1} - y \right|\right|_2^2. \tag{2}$$

Intuitively, this function measures the prediction error for a single pair of documents, x and y, where x represents the "old" document and y is the "new" document. Now we generalize it and define the error function for a collection of documents. Suppose we have a collection of sequenced documents D, where the number of documents is $|D| = m + 1$; the topic distribution of the i-th document is d_i, where d_i is a row vector and i indicates the temporal order of d_i. Next, we construct two matrices $D^{(1,m)}$ and $D^{(2,m+1)}$ as follows:

$$D^{(1,m)} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix}, \quad D^{(2,m+1)} = \begin{bmatrix} d_2 \\ d_3 \\ \vdots \\ d_{m+1} \end{bmatrix}.$$

Notice that both $D^{(1,m)}$ and $D^{(2,m+1)}$ are $m \times n$ matrices. The i-th rows of these two matrices are d_i and d_{i+1}, and they are sequentially adjacent in the collection D. In other words, $D^{(1,m)}$ represents the topic distribution matrix of "old" documents and $D^{(2,m+1)}$ is the matrix of "new" documents. According to the error function for a single document pair (Function (2)), the prediction error for the sequenced document collection D is defined as:

$$err_f = ||LD^{(1,m)}T - D^{(2,m+1)}||_F^2. \tag{3}$$

where $|| \cdot ||_F$ is the Frobenius matrix norm. L is a $m \times m$ diagonal matrix which normalizes each row of $D^{(1,m)}T$. The i-th diagonal entry of L is the reciprocal of the ℓ^1-norm of the i-th row in $D^{(1,m)}T$:

$$L = \begin{bmatrix} \frac{1}{||d_1 T||_1} & & & \\ & \frac{1}{||d_2 T||_1} & & \\ & & \ddots & \\ & & & \frac{1}{||d_m T||_1} \end{bmatrix}.$$

2.1.2 Iterative Minimization of the Error Function

The function err_f is a non-linear function. Numerical experiments show that function err_f is convex, which suggests using iterative methods to approach the optimal T that minimizes err_f. Each iteration updates the solution T as below:

$$T^{(j)} = (L^{(j-1)} D^{(1,m)})^\dagger D^{(2,m+1)},$$

where

$$L^{(j-1)} = \begin{bmatrix} \frac{1}{||d_1 T^{(j-1)}||_1} & & \\ & \ddots & \\ & & \frac{1}{||d_m T^{(j-1)}||_1} \end{bmatrix}.$$

Such iterative method can be initialized by

$$T^{(0)} = D^{(1,m)\dagger} D^{(2,m+1)},$$

where $D^{(1,m)\dagger}$ is the pseudo-inverse of $D^{(1,m)}$.

2.1.3 Direct Minimization of the Error Function

Iterative methods may be slow to converge and only give an approximate solution. Ideally, we would like to have a direct solution procedure for TM-LDA which could be efficiently and accurately implemented. By noticing an important property of the error function (3), we use Theorem 1 to derive a least squares characterization of the TM-LDA solution and to render an explicit form of the exact solution.

THEOREM 1. *Let* $\mathbf{e}$ *denote the* $n \times 1$ *matrix of all ones. For any* $A \in \mathbb{R}_+^{m \times n}$ *and* $B \in \mathbb{R}_+^{m \times n}$ *such that* $A\mathbf{e} = \mathbf{e}$ *and* $B\mathbf{e} = \mathbf{e}$, *it holds*

$$AA^\dagger B\mathbf{e} = \mathbf{e},$$

where $A^\dagger$ *is the pseudo-inverse of* A.

PROOF. Because $B\mathbf{e} = \mathbf{e}$,

$$AA^\dagger B\mathbf{e} = AA^\dagger \mathbf{e}.$$

$AA^\dagger \mathbf{e}$ is the orthogonal projection of $\mathbf{e}$ onto $Range(A)$. Since $A\mathbf{e} = \mathbf{e}$, $\mathbf{e} \in Range(A)$. Therefore $AA^\dagger \mathbf{e} = \mathbf{e}$. $\square$

The matrices $D^{(1,m-1)}$ and $D^{(2,m)}$ satisfy the properties $D^{(1,m-1)}\mathbf{e} = \mathbf{e}$ and $D^{(2,m)}\mathbf{e} = \mathbf{e}$ since each row of these two matrices is a topic distribution vector of a document and the row sum is naturally 1. By adapting the result of Theorem 1 to TM-LDA, we obtain the following result:

$$D^{(1,m)}T^{(0)}\mathbf{e} = D^{(1,m)} D^{(1,m)\dagger} D^{(2,m+1)}\mathbf{e} = \mathbf{e}.$$

In other words, $\|d_i T^{(0)}\|_1 = 1$ for any $i \in \{1, 2, \ldots, m\}$. Therefore $L^{(0)} = I$, the $m \times m$ identity matrix. Hence, $T^{(1)}$ can be written as

$$T^{(1)} = (L^{(0)} D^{(1,m)})^\dagger D^{(2,m+1)} = T^{(0)}.$$

This indicates that

$$T = D^{(1,m)\dagger} D^{(2,m+1)}$$

gives the optimal solution for minimizing err_f. Hence, computing the TM-LDA solution amounts to solving a matrix least squares problem:

$$\min_T ||D^{(1,m)}T - D^{(2,m+1)}||_F^2.$$

2.2 TM-LDA for a Document Stream

A document stream of an author consists of temporally sequenced posts. After we train LDA on the collection of documents, the topic distribution vector of each post is obtained. We can therefore construct the matrices $D^{(1,m)}$ and $D^{(2,m+1)}$. Suppose we collect 20 consecutive posts per user and the number of users is p, then the training stage of TM-LDA on such stream is illustrated in Figure 1. The left matrix is $D^{(1,m)}$ and the right matrix is $D^{(2,m+1)}$, where $m = 19 \times p$ in this case. For each user, 20 consecutive posts makes 19 pairs, they are (post 1, post 2), (post 2, post 3), ..., (post 19, post 20). Each pair is one training sample, and forms one row of the matrix $D^{(1,m)}$ and $D^{(2,m+1)}$. By multiplying the "old" post matrix $D^{(1,m-1)}$ with the transition parameter matrix T, the predicted topic distribution of "new" posts is obtained.
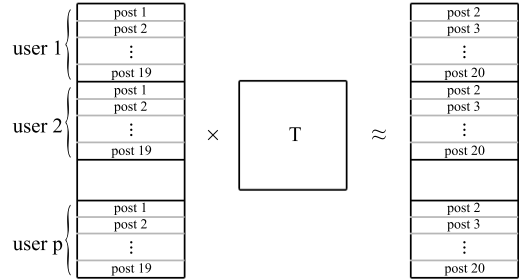

Figure 1: Constructing TM-LDA for document streams.

To simplify the notations, let $A = D^{(1,m)}$ and $B = D^{(2,m+1)}$. According to Theorem 1, TM-LDA is reduced to the following problem:

$$\min_T ||AT - B||_F^2. \quad (4)$$

Again, A is the topic distribution matrix of "old" posts and B is the topic distribution matrix of "new" posts. The training phase of TM-LDA becomes a least squares problem. When the condition number of A, $\kappa(A)$, is small and the system is overdetermined, we can effectively obtain T as:

$$T = A^\dagger B = (A'A)^{-1}A'B, \quad (5)$$

where A' denotes the transpose of A. In practice, multiplication by $(A'A)^{-1}$ is accomplished by Cholesky factorization of $A'A$ followed by forward and backward substitutions. We also observe that from Theorem 1 and the fact that $A\mathbf{e} = \mathbf{e}$ it readily follows that it is also $T\mathbf{e} = \mathbf{e}$, since A has full column rank.

3. UPDATING TOPIC TRANSITION PARAMETERS

Not only the topics of a user's posts are likely to change, but the transition weights from one topic to another could

also vary over time. Both the changing popularity of certain topics and external events will affect the transition parameters of related topics. In other words, the transition parameters have to be updated and adjusted by using recently-generated posts as training samples. One way to solve this updating problem is to compute the transition parameter matrix as the new documents emerge each time. However, re-computing transition parameters may result in lower efficiency and a less smooth parameter adjustment process. In this section, we present an efficient algorithm which can gradually and smoothly adjust transition parameters as the new posts emerge, with significantly less computation than re-computing TM-LDA from scratch.

3.1 Updating Transition Parameters

We now introduce the algorithm for updating the transition parameter matrix T. Suppose we append k rows of new document pairs, U_k and V_k, to the bottom of A and B and form $\hat{A}$ and $\hat{B}$ as below:

$$\hat{A} = \left[\begin{array}{c} A \\ U_k \end{array} \right], \quad \hat{B} = \left[\begin{array}{c} B \\ V_k \end{array} \right].$$

According to Equation (5), the new transition parameter matrix, $\hat{T}$ is:

$$\hat{T} = (\hat{A}'\hat{A})^{-1}\hat{A}'\hat{B}.$$

We apply the Sherman-Morrison-Woodbury formula [7] to $(\hat{A}'\hat{A})^{-1}$ and obtain the following result:

$$(\hat{A}'\hat{A})^{-1} = (A'A + U_k'U_k)^{-1}$$

$$= (A'A)^{-1} - (A'A)^{-1}U_k'(I + U_k(A'A)^{-1}U_k')^{-1}U_k(A'A)^{-1}.$$

Let $C = (A'A)^{-1}U_k'$, then the updated transition parameter matrix $\hat{T}$ is:

$$\hat{T} = (\hat{A}'\hat{A})^{-1}(A'B + U_k'V_k)$$

$$= T + CV_k - C(I + U_kC)^{-1}C'(A'B + U_k'V_k). \quad (6)$$

Notice that $A'A$ and $A'B$ have been computed and stored when computing T. In other words, to compute $\hat{T}$, we just need $U_k'V_k$ and C. The only possibly time-consuming part is to obtain $(I + U_kC)^{-1}C'$, which requires no more than $O(k^3)$, as shown in reference [7]. The remaining components of computing $\hat{T}$ have the complexity of $O(k)$, and even less when U_k and V_k are sparse. Therefore the overall cost for updating the transition parameter matrix is at most $O(k^3)$.

3.2 Updating Transition Parameters with QR-factorization

The least squares problem can also be solved by QR-factorization [7]. Suppose the QR-factorization of matrix A is $A = QR$, where $Q'Q = I$ and R is an upper triangular matrix. The solution of Formula (4) can be written as:

$$RT = Q'B.$$

Since R is upper triangular, T can be easily found.

When a new pair of documents, u and v, is added on the top of A and B, we have the updated topic distribution matrices as below:

$$\hat{A} = \left[\begin{array}{c} u \\ A \end{array} \right], \quad \hat{B} = \left[\begin{array}{c} v \\ B \end{array} \right].$$

The QR-factorization of $\hat{A}$ can be written as:

$$\hat{A} = \hat{Q}\hat{R},$$

$$\hat{R} = J_1 \ldots J_n \left[\begin{array}{c} u \\ R \end{array} \right],$$

$$\hat{Q} = \left[\begin{array}{cc} 1 & \\ & Q \end{array} \right] J_n' \ldots J_1',$$

where $J_1, \ldots, J_n$ are rotation matrices which make $\hat{R}$ upper triangular, and n is the number of columns in both A and B. Therefore, the updated transition parameter matrix $\hat{T}$ can be computed as follows:

$$\hat{R}\hat{T} = \hat{Q}'\hat{B} = J_1 \ldots J_n \left[\begin{array}{cc} 1 & \\ & Q \end{array} \right] \left[\begin{array}{c} v \\ B \end{array} \right]$$

$$= J_1 \ldots J_n \left[\begin{array}{c} v \\ Q'B \end{array} \right]. \quad (7)$$

In practice, we use the Sherman-Morrison-Woodbury formula to update the transition parameter matrix because the condition number of A is typically small, which means the document topic distribution matrix is well-conditioned. Hence, it will not easily become singular or ill-conditioned during the updating process. Also, the Sherman-Morrison-Woodbury formula provides a way to control the speed of updating by tuning the parameter k. When k is small, each update will take less time and a more fine-grained matrix changing process can be obtained. On the other hand, the updating algorithm will have similar complexity as re-computing the transition parameter matrix when k gets larger. Therefore if k is greater than the "balanced" complexity point, it is preferable to re-compute the matrix.

4. EXPERIMENTS

In this section, TM-LDA is evaluated empirically over a large crawl of Twitter data. By measuring perplexity (Section 4.2), we show that TM-LDA significantly outperforms static topic models on predicting actual word distributions of future Twitter posts (Section 4.3). Additionally, the efficiency of the algorithm for updating transition weights is assessed in Section 4.4.

4.1 Dataset

To validate TM-LDA, we collect Twitter posts (henceforth, Tweets) from more than 260,000 public user accounts, over a period of one month. The public user accounts are selected from the TREC 2011 microblog track[1] and we only keep the users with valid geo-location information. A list of 89 candidate cities are generated by taking the union of top 50 U.S. cities (in population) and the capital cities of the 50 U.S. states. After that, the users whose claimed geo-locations are one of the candidate cities will be selected.

All selected user accounts are tracked daily, generating an average of around 1.1 million new Tweets per day. However, Tweets are usually short and informal, which makes

---

[1]http://trec.nist.gov/data/tweets/

the quality of Tweets vary a lot from each other. To attempt to filter low-quality Tweets, we first filter out stopwords and the words with less than 5 occurrences in our dataset, and then keep only the Tweets with more than 3 terms after filtering. As a result, one third of the raw Tweets are excluded, with more than 20 million "high quality" Tweets remaining for analysis.

Dates	From 12-15-2011 To 1-15-2012
Number of Raw Tweets	34,150,390
Number of Valid Tweets	23,096,894
Average Length of Valid Tweets (words)	5.12
Number of Users	264,628
Number of Cities	89
Number of Valid Tweet Pairs	13,273,707

Table 1: Description of Tweet Stream Data.

4.2 Using Perplexity as Evaluation Metric

TM-LDA is designed to predict the topic distribution of future Tweets based on historical Tweets. Therefore we employ the measurement of *Perplexity* to evaluate TM-LDA against the actual word occurrences in future Tweets. Usually, perplexity is used to measure how well a language model fits the word distribution of a corpus. It is defined as:

$$Perplexity_l = 2^{-\sum_{i=1}^{N} \log_2 p_l(x_i)}. \quad (8)$$

Formula (8) dictates the perplexity of the language model l, where $p_l(x_i)$ is the probability of the occurrence of word x_i estimated by the language model l and N is the number of words in the document. Intuitively, if the language model yields higher probability for the occurrences of words in the document than words that are not in the document, the language model is more accurate and the perplexity will be lower.

4.3 Predicting Topics of Future Tweets

TM-LDA predicts the topic distribution of future Tweets by taking the "previous" Tweets as input (Formula (1)). Basically, TM-LDA will multiply the topic distribution vector by the transition parameter matrix and normalize it to form the topic distribution of a "future" Tweet. There are two key components in this process: (1) the transition parameter matrix, and (2) the topic distribution of the "previous" Tweets.

The transition parameter matrix is trained according to the algorithm introduced in Section 2. In practice, TM-LDA will use 7-day (one week) historical Tweets to train the transition parameter matrix, and then predict the Tweets generated on the 8th day. For example, if we want to predict the Tweets on the date Dec. 22, 2011, we will collect all the Tweets generated from Dec. 15, 2011 to Dec. 21, 2011 and train LDA on this one-week Tweet collection to obtain the topic distribution vectors for each single Tweet. During the training of LDA, each Tweet is treated as a document and the number of topics is set to 200. After that, we build two topic distribution matrices, "old" Tweet matrix and "future" Tweet matrix, as in Figure 1 and compute the transition parameter matrix according to Formula (5).

For the Tweets generated on the 8th day (which we want to predict), we cannot have their topic distributions from

LDA directly. Figure 2 illustrates this situation: LDA is trained on one-week Tweets but not on the Tweets a and b, which means we need to map them to the topics through the results of LDA. The topic distribution of "previous" Tweets a is inferred from the LDA model. Given the words appearing in the Tweet t, the topic distribution is inferred as:

$$p(z|t) = \sum_w p(z|w)p(w|t) = \sum_w \frac{p(w|z)p(z)}{\sum_{z'} p(w|z')p(z')}p(w|t), \quad (9)$$

where $p(w|t)$ is the normalized frequency of word w in Tweet t. Both $p(w|z)$ and $p(z)$ are the results of LDA model.

In summary, TM-LDA first trains LDA on 7-day historical Tweets and compute the transition parameter matrix accordingly. Then for each new Tweet generated on the 8th day, it predicts the topic distribution of the following Tweet. When the actual "future" Tweet b (in Figure 2) becomes available, we can measure the perplexity of TM-LDA.
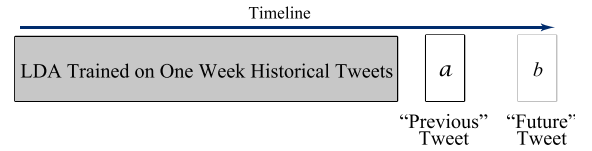

Figure 2: The Scheme for Predicting "Future" Tweets.

Figure 2 illustrates the prediction scheme of TM-LDA and other methods. All methods build LDA on one-week historical Tweet data, and for each new Tweet a, they predict the topic distribution of the "following" Tweet b. Although many dynamic topic modeling algorithms have been developed [4] [20] [16] [9], they are mainly designed to model topic trends and dynamic word distributions over time. In [4], it shows that the predictive power of static LDA and Dynamic Topic Model are very close. Therefore, we compare TM-LDA with the following methods:

1. **Estimated Topic Distributions of "Future" Tweets**: the topic distribution of the Tweet b. This is computed based on the actual words in the "future" Tweets according to Formula (9). This system approximately reflects the optimal perplexity of LDA-based models.

2. **LDA Topic Distributions of "Future" Tweets**: the inferred topic distribution of the Tweet b. They are inferred from the LDA model which is trained on the one-week historical Tweets. The inferring algorithm is introduced by Blei et al. [5]. This system knows the words appearing in the "future" Tweets, so that it shows the optimal perplexity of the original LDA [5].

3. **LDA Topic Distributions of "Previous" Tweets**: the inferred topic distribution [5] of the Tweet a. They are also inferred from the LDA trained on one-week historical Tweets. This system uses the topic distributions of "previous" Tweets as the topic distributions of the "Future" Tweets. It shows the perplexity of static prediction model built on LDA.

We test these three methods above along with our model, TM-LDA, on the Twitter corpus described in Section 4.1. In Figure 3, we can see that TM-LDA consistently provides lower perplexity than the static prediction model, reducing

Topics	Weather	Social Media	U.S. Deficit	Traffic	Job Hunting	Reading Media	Weight Loss	Presidential Election
5 Top Words	cold winter snow weather warm	social media marketing network internet	bill house budget cut obama	traffic accident bridge lane blocked	sales hiring jobs project position	daily news digest newspaper headlines	weight loss healthy fitness diet	romney paul iowa republican debate

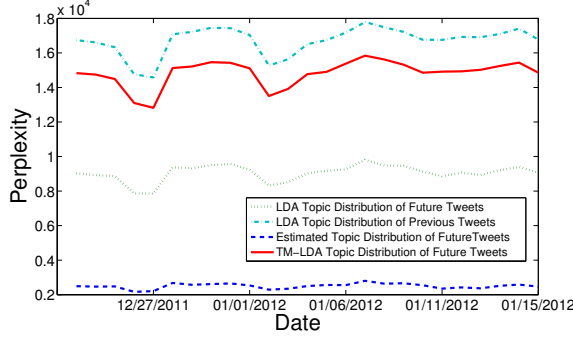Table 2: LDA Sample Topics with 5 Top Representative Words.

Figure 3: Perplexity of Different Models.

Figure 4: Time Complexity of Updating Transition Parameter Matrix based on One-Week Tweet Data.

it by 11.4% compared to the static LDA models. The improvements are statistically significant with the significance level $\alpha < 0.001$. It turns out that the performance of TM-LDA could be affected by the topic estimation of "previous" Tweets, which TM-LDA uses as input arguments.

Interestingly, we find that Tweets are easier to predict on holidays than other days. We can see that the perplexity drops on the dates of Christmas and New Year, which suggests that the topics discussed during holiday seasons are more predictable. Also note that we use the ℓ^2 norm to define the error function for TM-LDA, which enables us to efficiently optimize it by solving a least squares problem. In future work, we plan to test the use of the ℓ^1 norm of the error, which may give better results due to its being less sensitive to the presence of outliers.

4.4 Efficiency of Updating Transition Parameters

In Section 3, we introduced the Sherman-Morrison-Woodbury formula to update the transition parameter matrix. We now empirically evaluate the runtime costs of this algorithm for different values of k. Suppose we have computed a transition parameter matrix T based on one-week historical Tweet data, which consists of more than 3 million Tweet pairs. Given k new pairs of Tweets, we measure the time needed to update matrix T for different values of k.

We test the time complexity by running the Matlab implementation of our updating algorithm on a machine with 24 *AMD Opteron(tm) 6174* processors and 128 Gigabytes of RAM. Figure 4 shows that our updating algorithm can efficiently find T when k is not too large. Compared with re-computing T, which usually takes around 280 seconds for one-week historical Tweets in our corpus (about 7.7 million posts), our updating algorithm consumes 61.4% less time when k is 10,000, while resulting in a more smooth T.

4.5 Properties of Transition Parameters

The transition parameter matrix T is the key result of TM-LDA. It enables us to adjust the topic weights based
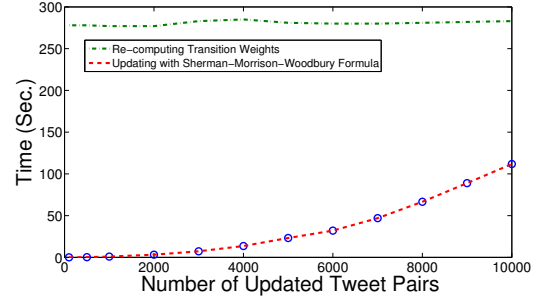
on historical Tweets and make more precise prediction. The empirical results indicate several consistent properties of matrix T.

First, T is a square matrix where the size of T is determined by the number of topics trained in LDA. Each entry in T reflects the weight that a certain topic will propagate to another topic in future Tweets. More precisely, entry t_{ij} of T shows the degree that topic i will contribute to topic j in the following Tweet. The algorithm in Section 2.1 does not enforce non-negativity of T, but it turns out that T has very few negative entries and all the negative entries in T are close to zero. Second, as already observed at the end of section 2 the row sums of T are all 1, which means that the overall weight emitted from a topic is 1, and this ensures topics will not amplify themselves and make the system unstable. Note that T is close to a row-stochastic matrix, which can be regarded as the transition matrix of a Markov chain. We can make T into a Markov chain transition matrix by making small adjustments that enforce nonnegativity in T while preserving the unit row sums property. In practice, we have not found these adjustments to be necessary.

5. APPLYING TM-LDA FOR TWITTER TREND ANALYSIS AND SENSEMAKING

The previous section shows that TM-LDA can consistently and significantly improve the prediction quality of future Tweets compared to the static model. Additionally, TM-LDA can provide a more in-depth view of temporal relationships among topics and public opinion of popular events. We now turn to discuss the analytical power of TM-LDA.

As previously mentioned, LDA is trained on the Tweet stream dataset and the number of topics is set to 200. We show the words with top weights for several topics as in Table 2. The "names" of the topics are manually labeled based on the words in each topic with highest probabilities.

5.1 Global Topic Transition Patterns

To analyze the global topic transition patterns, TM-LDA is trained on all the valid Tweet pairs. The topic transition

parameter matrix T has the size of 200×200 and the average transition weight of all 40,000 entries in T is 0.005. We visualize the matrix T in Figure 5; however, from the initial visualization it could be challenging to locate interesting topic transition patterns. We therefore pre-select "interesting points" from the raw transition matrix using a procedure described by Formula (10), and then do a case study on those "interesting transition points".
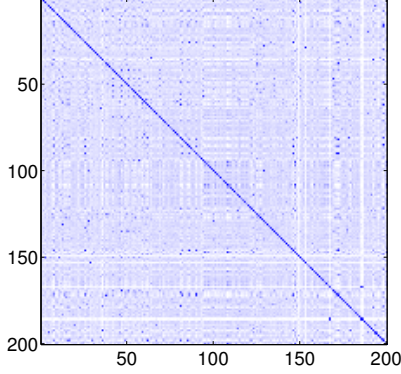

Figure 5: Visualization of Global Topic Transitions.

In Figure 5, it's clear that matrix T has relatively large diagonal entries. This provides the evidence that topics of historical Tweets do not randomly transition from one to another, but follow certain statistical rules. However, the diagonal entries of T are always less than 1. Meanwhile, the empirical average value of diagonal entries in T, $\bar{t}$ is 0.095, which shows that new Tweets usually do not simply repeat the topics of historical Tweets. The standard deviation of non-diagonal (non-self-transition) entries, σ, is 0.003. We define the threshold to be the average plus five times the standard deviation:

$$Threshold = \bar{t} + 5 \times \sigma, \qquad (10)$$

which is $0.005 + 0.003 \times 5 = 0.02$, as the bar of "interesting" points. After filtered by this threshold, a clearer transition pattern is obtained and shown in Figure 6.
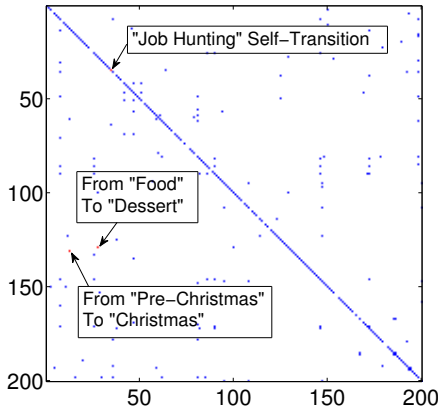

Figure 6: Interesting Transition Points.

Figure 6 shows three types of "interesting" transition points: (1) diagonal points: these points have high self-transition

	From Topic	To Topic	Weight
(1)	Job Hunting	Job Hunting	0.448
	Weather	Weather	0.304
	Reading Media	Reading Media	0.286
	Weight Loss	Weight Loss	0.282
(2)	Internet Company	Social Media	0.045
	U.S. Deficit	Presidential Election	0.044
	Food	Dessert	0.041
	Security and Crime	Military Action	0.039
(3)	Traffic and Accident	Rescue and Police	0.044
	Restaurant	Food	0.040
	Pre-Christmas	Christmas	0.032
	Startup Business	Social Media	0.030

Table 3: Three Kinds of Topic Transitions: (1) Self-Transition (2) Symmetric Transition (3) Non-Symmetric Transition.

weights and they are the topics people tend to keep discussing about; (2) symmetric points: both t_{ij} and t_{ji} are interesting points. These topics are highly correlated and they are usually mentioned in consecutive Tweets; (3) non-symmetric points: one and only one of t_{ij} and t_{ji} is an interesting point. These topics usually reflect strongly time-sensitive properties of certain events and scenarios. We rank the points in Figure 6 by their values and list the most representative ones in Table 3.

Table 3 shows the general topic transition patterns in our Twitter data. We can tell that certain topics are very popular according to their high self-transition weights, such topics include "Job hunting" and "Weight loss". The topic popularity provided by TM-LDA not only shows the amount of related Tweets, but also reflect the persistence of certain topics. Additionally, transition weights can also be indicators of relatedness among topics. For example, the topic "Internet company" and the topic "Social media" are very close to each other and therefore one topic could trigger users' interest in the other topic. Additionally, we can also find some "one way" transitions, which may suggest strong temporal orders or cause-effect relationships among topics. For instance, the topic "Pre-Christmas" is about the ideas and preparation of Christmas gifts; this topic always appears before the topic "Celebration of Christmas". This information could be potentially useful not only for predicting future Tweets, but also for personalization, content recommendation, and advertising.

5.2 Changing Topic Transitions Weights

Topic transitions could help dynamically model the Twitter data. We now show that topic transition parameters are also dynamic and will change over time. Figure 7 shows the changing topic transition weights for three example topics. For instance, the self-transition weight of the topic "U.S. troops withdrawal" increases before the holiday season and becomes flat after that. As another example, the transition probability between topic "Obama Government" and topic "Republican Debate" increases as the presidential election is approaching. Again, all these topics are time-sensitive and the changing transition parameters can describe how they progress over time. Figure 7 also shows how transition weights change for out-dated topics. The topic "Christmas decoration" rapidly gains popularity before the holiday, but the weights start dropping right after Christmas, which implies that users lose interest after the Christmas holiday.
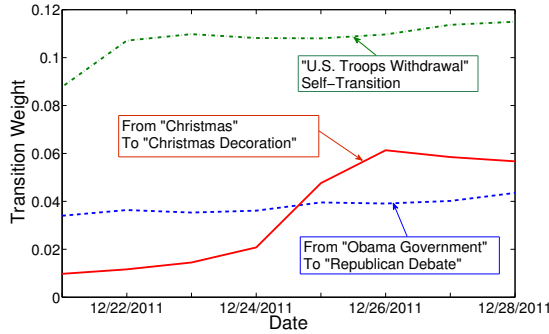
Figure 7: Topic Transition Weights Change over Time.

5.3 Varying Topic Transition Patterns by Cities

Topic transition patterns can help reveal potential social issues and identify interesting behavioral patterns in various cities. As an example, we compare the transition parameter matrices for nine major cities in the United States. Empirical results show that these cities have very different topic transition weights from each other.

The topic transitions of cities could be studied in two ways: (1) for a particular t, what topics tend to *precede* t (Table 4); and (2) what topics tend to *follow* t (Table 5). The former could indicate potential causes of a topic; the latter could help understand aftermath and potential consequences of an event.

	Traffic	Complaints	Compliments
Atlanta, GA	Airport	Smoke/Drug	Holidays
Boston, MA	Trip	Music	Love
Chicago, IL	Weather	Work Life	Pray
Los Angeles, CA	Church	Break-up	Basketball
Miami, FL	Party	Alcohol	Holidays
New York, NY	Manhattan	Break-up	Movies
San Francisco, CA	Japan/Sushi	Hate	Love
Seattle, WA	Weather	Party	Planning
Washington, D.C.	Plaza	Sleep	Dress

Table 4: Top Topics Preceding the Topics "Traffic", "Complaints" and "Compliments" (columns) for 9 U.S. Major Cities (rows).

Table 4 reports the sample topic transitions of nine major U.S. cities, and it reflects the different characteristics and problems. For example, the topics occurring before "Compliments" could potentially be pleasing, and the topics before "Complaints" might be related to social problems. The result of TM-LDA can also benefit targeted analysis. In Table 4, we show the top topics occurring *before* the topic "Traffic", which may imply the potential traffic issues in various cities. It turns out that the results align with the actual traffic conditions quite well, such as the airport in Atlanta (the busiest airport in the world), Manhattan (a busy area in New York) and Japan-town in San Francisco.

Table 5 shows the topics mentioned after "Work life" and "Dining". It provides the observation of what people tend to do or discuss after work and dinner. This can potentially be applied to advertisement and marketing to target consumers. For example, the users in Los Angeles and Boston areas tend to discuss "Beauty" topics (e.g., make-up and spas) after "Dining", which suggests a better advertising

	Work Life	Dining
Atlanta, GA	Complaint	Party
Boston, MA	Book	Beauty
Chicago, IL	Celebration	Weight Loss
Los Angeles, CA	E-shopping	Beauty
Miami, FL	Music	Shopping
New York, NY	Social Media	Weight Loss
San Francisco, CA	Weight Loss	Entertainment
Seattle, WA	Job Hunting	Weight Loss
Washington, D.C.	Presidential Election	Reading Media

Table 5: Top Topics Following "Work Life" and "Dining".

strategy in these cities. These results could also be used to better understand the common lifestyle and culture in these cities.

6. RELATED WORK

Topic modeling of temporally-sequenced documents has been studied extensively over the last decade. Blei et al. [4] model the topic evolution over time as a discrete chain-style process and each slice is modeled by LDA. Wang et al. [20] try to model the topics continuously over time. Kawamae [10] models topic evolution by treating temporal words differently from other words. However, TM-LDA differs from these previously proposed dynamic topic models in three ways. First, TM-LDA is designed to learn the topic transition patterns from temporally-ordered documents, while dynamic topic models focus on changing word distributions of topics over time. Second, the efficient optimization algorithm of TM-LDA enables it to be applied on large scale data. Third, TM-LDA can be updated effectively as the new documents stream in.

Meanwhile, the proliferation of social media content makes large-scale temporally ordered data increasingly available. Lin et al. [12] use the social network structure to learn how topics temporally evolve within the social community. Saha and Sindhwani [16] adapt Non-negative Factorization to learn the trending topics in social media. Yang and Leskovec [21] analyze the common "shape" of popularity-time curve in social media via a time-aware clustering algorithm. TM-LDA also takes advantage of temporal information in social media data, but works differently. Rather than learning the dynamic word distributions or trends of topics over time, TM-LDA learns the *relationships* or *transitions* among topics. This allows TM-LDA to compare not only topic popularity in isolation, but also common sequences of topics over time. Additionally, TM-LDA provides trend prediction algorithm for an individual thread of documents, such as the Tweet thread of a particular user, where most of the models only focus on topic evolution over entire corpora.

Social media applications also bring challenges, such as the requirement to efficiently process large-scale data and to support online learning from social data streams, for example, Tweet streams. There has been some previous work on efficient learning from data streams [11]. Ramage et. al [15] attempt to facilitate Twitter content representation by incorporating labeled topic models. Bulut and Singh [6] provide a framework to monitor streaming data. Zhu et. al [22] propose a wavelet-tree based data structure to efficiently detect bursts in data streams. Our model, TM-LDA, is able to efficiently process Tweet stream data and dynamically update the connections among topics in real time.

130

Semantic analysis and topic modeling in social media can facilitate many applications, such as event tracking [13], trend detection [3] and popularity prediction [19]. Additionally, previous work demonstrated that understanding social media content could benefit applications in many other fields. For example, Asur and Huberman [2] show that Tweets can help predict movie revenues. More recently, Paul and Dredze[14] show that Twitter data could be useful for tracking and analyzing public health information. Similarly, TM-LDA could be potentially applied to public health or even advertising domains, for example by anticipating the future activities of a user based on common topic transition patterns.

7. CONCLUSIONS

We presented and evaluated a temporally-aware language model, TM-LDA, for efficiently modeling the topics and topic transitions that naturally arise in document streams. We have shown that our method is able to more faithfully model the word distribution of a large collection of real microblogging messages, compared to previous state-of-the-art methods. Furthermore, we introduced an efficient model updating algorithm for TM-LDA that dramatically reduces the training time needed to update the model, making our method appropriate for online operation. In a series of experiments, we demonstrated ways in which TM-LDA can be naturally applied for mining, analyzing, and exploring temporal patterns in microblogging data.

Our promising results in this paper suggest applying TM-LDA to other datasets and domains. One such natural application, to be explored in future work, is modeling topic transitions within threads on Community Questions Answering forums or social comment streams, to better analyze the evolution of the discussions and to identify valuable contributions. Additionally, TM-LDA provides a general framework of modeling topic transitions that could be applied to other probabilistic topic modeling algorithms, such as probabilistic Latent Semantic Analysis (pLSA) [8] and author-topic models such as [18]. Together, TM-LDA and the associated algorithms provide a valuable tool to improve upon and complement previous approaches to mining social media data.

8. REFERENCES

[1] F. Abel, Q. Gao, Houben, G.J., and K. Tao. Analyzing user modeling on Twitter for personalized news recommendations. In *Proceedings of the International Conference on User Modeling, Adaptation and Personalization (UMAP)*, 2011.

[2] S. Asur and B. A. Huberman. Predicting the future with social media. In *Web Intelligence*, 2010.

[3] S. Asur, B. A. Huberman, G. Szabó, and C. Wang. Trends in social media: Persistence and decay. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[4] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine learning (ICML)*, 2006.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *Journal of Machine Learning Research*, 2003.

[6] A. Bulut and A. K. Singh. A unified framework for monitoring data streams in real time. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, 2005.

[7] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[8] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

[9] S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, 2011.

[10] N. Kawamae. Trend analysis model: trend consists of temporal words, topics, and timestamps. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.

[11] J. Kleinberg. Temporal dynamics of on-line information streams. *In Data Stream Management: Processing High-Speed Data*, 2006.

[12] C. X. Lin, Q. Mei, J. Han, Y. Jiang, and M. Danilevsky. The joint inference of topic diffusion and evolution in social communities. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM)*, 2011.

[13] C. X. Lin, B. Zhao, Q. Mei, and J. Han. PET: a statistical model for popular events tracking in social communities. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.

[14] M. Paul and M. Dredze. You are what you tweet: Analyzing Twitter for public health. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[15] D. Ramage, S. T. Dumais, and D. J. Liebling. Characterizing microblogs with topic models. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[16] A. Saha and V. Sindhwani. Learning evolving and emerging topics in social media: A dynamic NMF approach with temporal regularization. In *Proceedings of the 5th International Conference on Web Search and Data Mining (WSDM)*, 2012.

[17] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World wide web (WWW)*, 2010.

[18] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2004.

[19] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Commun. ACM*, 53:80–88, Aug. 2010.

[20] X. Wang and A. McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[21] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, 2011.

[22] Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.