CLUSTERING BASED CAUSAL TOPIC MINING

BY

VISHAAL MOHAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Chengxiang Zhai

ABSTRACT

Events in the world generate an enormous amount of textual data like tweets and news articles. These events also manifest in the form of changes to time-series numeric data. This thesis deals with the problem of extracting these events from the timestamped document collection in the form of topics that cause a change in a time-series. We develop a conceptual framework for that can be used to analyze different causal topic mining algorithms. We also propose two novel clustering based algorithms - cCTM-CF and cCTM-CoF to generate causal topics. We evaluate these algorithms both qualitatively, and quantitatively by comparing their *coherence* and *correlation* scores to that of the baseline generative causal topic model - gCTM. We found that cCTM-CoF performs 35% and 62.5% better according to these metrics as compared to the baseline.

To my parents

ACKNOWLEDGMENTS

I thank my advisor Professor Chengxiang Zhai for guiding me through this thesis. His ideas, comments and feedback were invaluable and gave me clarity when I needed it the most. I also thank the Computer Science department for giving me this unique opportunity.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

With the abundance of electronic text data on mediums ranging from news articles that report recent important happenings to tweets on Twitter that informally express an individual's opinion, text mining as a field of study has gotten incredible attention from the academic community in recent times. One such popular task, topic mining, refers to the task of identifying groups of words that are semantically related based on their co-occurrence in a collection of documents. This problem has been extensively studied by using different probabilistic models. The two basic topic models are Probabilistic Latent Semantic Analysis (PLSA) [1] and Latent Dirichlet Allocation (LDA) [2].

Given the widespread success of these models, there has since been considerable work that incorporates external knowledge – supervised LDA or sLDA incorporates external knowledge in the topic modeling process [3] (an example the model is tested on is movie reviews and corresponding ratings); including authorship information to publications to discover topics covered by an author [4] – are examples.

1.1 Motivation

Most of these topic models operate on purely textual datasets. However, typical documents that are considered in experiments are timestamped and are generally indicative of events that are taking place during the creation of the document. These events could be short, bursty events like the rise and fall of internet trends or memes [1] or prolonged over a long period of time like technological advances by a company. These events also lead to

[1] A meme is a humorous image, video or a piece of text that is copied (often with slight variations) and spread rapidly by Internet users.

changes in many other numerical time-series data. Extending the example given before, the rise in popularity of a meme could lead to a drastic increase in the number of times a video or a picture is shared and the breakthroughs by a company could lead to an increase in the stock price of that company. This tells us that there exists information about these latent events hidden in the time series data. One way by which we could extract these events is in the form of topics by doing an integrated analysis of the time-series and accompanying documents. For example, doing a topic analysis on articles from the New York Times[2] will identify tags for news articles but adding oil price trends to the analysis could point us to important events that caused the change in oil prices.

To that point, the problem we are considering is that of *finding topics from a collection of timestamped documents such that the topics are responsible for a change in an adjoining time-series datastream.*

1.2 Overview

The contributions of this thesis are as follows

- We characterize the architectures of existing existing causal topic models.
- We propose a novel clustering based approach to generate causal topics.
- Based on existing work, this thesis proposes a baseline generative model to generate causal topics.
- Finally, we analyze the performance of these different models.

1.3 Thesis structure

The rest of this thesis is structured as follows: We start by introducing the reader to the background and discuss representative works that are related to ours. Next follows Chapter 3, where we formulate a conceptual framework to find causal topics and discuss important features and shortcomings of

[2]www.nytimes.com

each category of solution. We then formally define the problem that this thesis targets. Chapter 4 forms the core of this thesis where we propse a novel clustering-based approach to finding causal topics. In Chapter 5, we describe the baseline algorithm, which is a generative model to find causal topics. The evaluation of the proposed model and the baseline along with an analysis of the results are in Chapter 6. Finally, Chapter 7 concludes the thesis by summarizing our findings.

CHAPTER 2

BACKGROUND

This chapter descibes the background related to this thesis. We discuss how each work leads to or relates to the models we propose.

2.1 Text and non-text

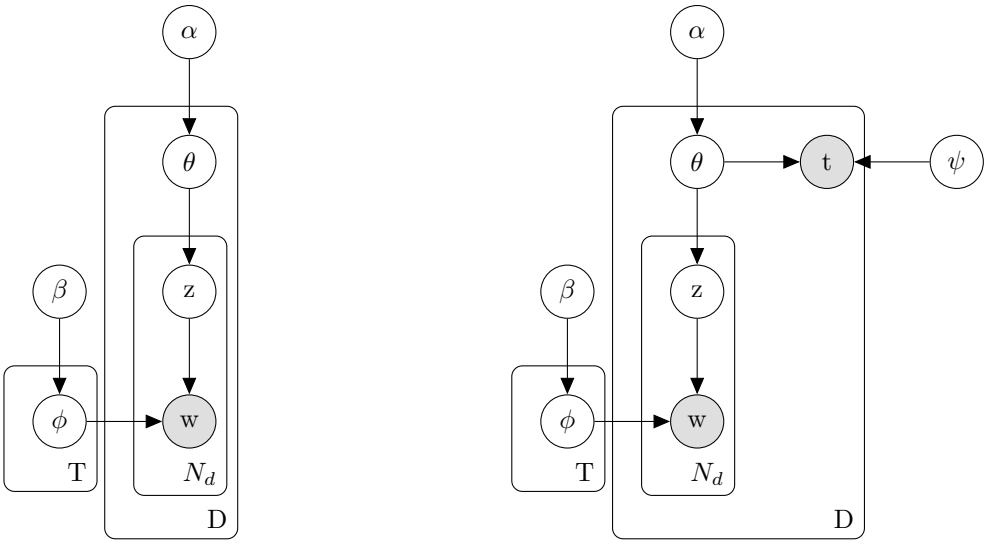
The integrated study of text and non-text data has seen interest from both the computer science and economics academic communities. In the domain of finance and economics, this has been in the form of predicting stock prices from various forms of text: news articles[5], press releases [6] and tweets on Twitter[1] [7]. There has been considerable work from the computer science community in using text for predictive tasks. An early example is predicting box office performance using blog articles [8]. A very popular class of work is using content on online media in various predictive tasks[9]: using twitter to predict football games [10], crime [11] and even the stock market [12].

The major difference between this and the problem that we are focussing on is the perspective. Even though some of the models used in these publications can be modified to generate topics, the approach is far different from ours. Most importantly, they don't explicitly use the information in the time-series data to generate topics that correlate with tiem-series.

2.2 Topics and time

A class of work related to ours is to analyze the evolution of topics over time. *The reason for the similarity is because time can be thought of as a linearly increasing time-series.* The most two popular models that target this

[1] https://en.wikipedia.org/wiki/Twitter

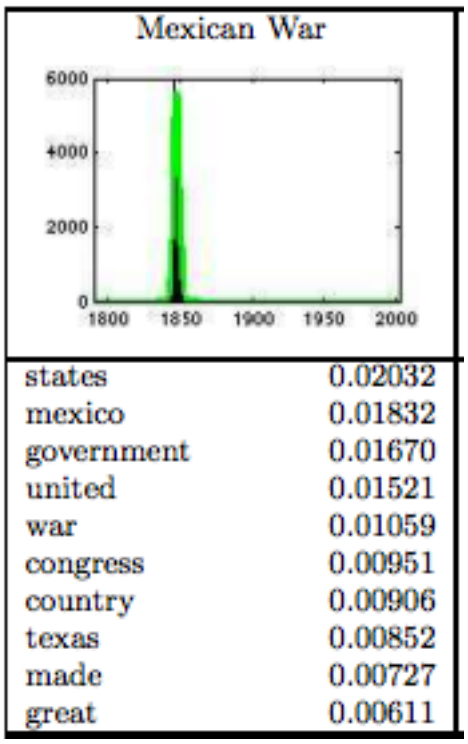


(a) Latent Dirichlet Allocation (b) Topics over Time

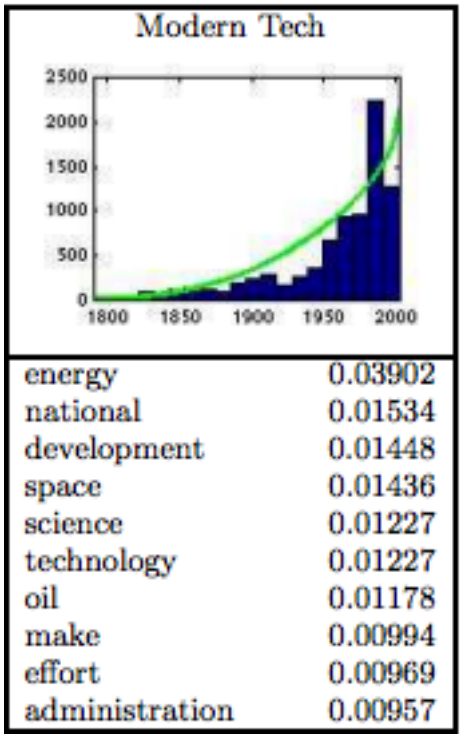
Figure 2.1: Graphical representations of related topic models

problem are Dynamic Topic Models (DTM) [13] and Topics over Time (TOT) [14]. They are both models that build on top of LDA [2] but in different ways. DTM discretizes time and finds topics by building a state space model on the natural parameters of the multinomial distributions. DTM uses variational approximations based on Kalman filters to perform inference. TOT doesn't discretize time but instead uses a Beta distribution to model the timestamp which is drawn from the topic distribution. We'll explain the generative process of TOT as the baseline algorithm we propose later in Chapter 5 relates closely to this.

To aid in the explanation of the generative process for TOT, we'll start by describing Latent Dirichlet Allocation (LDA) [2], one of the popular probabilistic topic models. LDA like many other topic models models each document as a mixture of topics. The generative process of LDA is as follows: For every document d, θ_d is the topic proportions distribution and is a multinomial over all topics that signifies the probability of a specific topic occurring in the document. For every document d, θ_d is first sampled from a Dirichlet distribution with a parameter α. Then to generate every word w_{d_i} in the document, a specific topic z_{d_i} is picked from θ_d first. For every topic z_{d_i}, $\phi_{z_{d_i}}$ which is a multinomial distribution over all the words in the vocabulary that signifies the probability of a word being chosen from a topic z_{d_i} is sampled from a Dirichlet distribution with parameter β. Every word w_{d_i} in the doc-



(a) Mexican war



(b) Modern tech

Figure 2.2: Evolution of topics as identified by the TOT model

ument d_i is generated by sampling from $\phi_{z_{d_i}}$. Refer to Figures 2.1a and 2.1b for graphical representations of both LDA and TOT.

The intuition behind TOT is that, by modeling the timestamp of a document a real valued variable generated from the topic proportions θ_d, we force the parameter estimation to find topics that correlate with the temporal information. Time is modeled as being drawn from a beta distribution $\psi_{z_{d_i}}$. The advantage that TOT has over DTM is that if there is a topic that appears for a brief period of time and disappears, TOT will create a topic with a narrow time distribution. Refer Figure 2.2 for examples[2] of two topic evolution plots. The words under each topic evolution plot are the words of the topic. Another advantage that is important in our context is that TOT is a simple model which aids not only in easy understanding and implementation but also integration into other more complex generative models.

One simple way to use either of these models described to find causal topics is to first find the time-evolving-topics and then filter the ones that correlate best with the time-series. However, such an approach would again not make

[2]Both these plots were taken from the original TOT paper [14]

use of the time-series in the topic modeling process. We'll later describe in Chapter 5, how we modify the TOT model to find causal topics.

2.3 Causal topics mining

In this section, we'll describe the works that relate closest to this thesis.

Model 1: Information Retrieval with Time Series Query (IRTSQ) [15] describes an algorithm that accepts a collection of documents and a time-series as a query and finds documents that are relevant to the time-series query. For example, given an input of Apple stock prices [3] and a collection of news articles, the algorithm would find documents that report big changes Apple's stock price. This is achieved by finding the correlation scores of all the words in the vocabulary with the time-series and scoring documents based on their content of highly correlated words. This algorithm can be modified to generate topics that correlate with the time-series rather than documents that correlate with it.

Model 2: Iterative topic modeling with time series feedback (ITMTF) [16] solves the exact same problem that this thesis focuses on. Given a collection of timestamped documents and a time-series, ITMTF finds causal topics. The steps involved in the algorithm are as follows:

1. Apply any topic model to find topics.
2. Identify significant topics that pass the causality test with the time-series data.
3. Split significant topics into two. One with words that correlate positively with the time-series data and another with those that correlate negatively. Feed these topics as prior to step 1.

Model 3: Supervised Dynamic Topic Models for Associative Topic Extraction with a Numerical Time Series (sDTM) [17] solves the exact same problem of finding causal topics as well. The algorithm is built on top of the Dynamic Topic Model (DTM)– by modeling every time-series datapoint as generated from α – the Dirichlet prior for the topic proportions θ. A disadvantage of this model is its need to discretize the time which could lead

[3] http://finance.yahoo.com/quote/AAPL

to missing topics that peak for a short time when the peaking time is not significant compared to the width of the time slice, or if the peak is divided between two different time slices.

Since all three models relate very closely to our propopsed solution, we'll discuss the details and critique each model in Section 3.2.

CHAPTER 3

A CONCEPTUAL FRAMEWORK FOR CAUSAL TOPIC MINING

This chapter first provides a conceptual framework for finding causal topics. We do this to make it easy to classify causal topic models and analyze their strengths and weaknesses easily. We then define the input and output to all our algorithms and then also formally define our problem. We classify the existing solutions (refer Chapter 2) as one of the three types of models.

3.1 Problem definition

Input:

- Collection of D documents and their respective timestamps $C = < (d_1, ts_1), (d_2, ts_2), \ldots, (d_D, ts_D) >$
- Numeric time-series data x with M datapoints and their respective times $x = \{(x_1, t_1), (x_2, t_2), \ldots (x_M, t_M)\} >$
- Number of causal topics required T.

Assert:

- $\text{Range}(t_1, t_2, \ldots, t_M) = \text{Range}(ts_1, ts_2, \ldots, ts_D)$
- $t_i \neq t_j \; \forall \; t_i, t_j \in \{t_1, t_2, \ldots, t_M\}$

Output:

- List of T causal topics
- Causality score of each topic

We use the same following definition for causal topics as introduced in Kim et. al. [16]

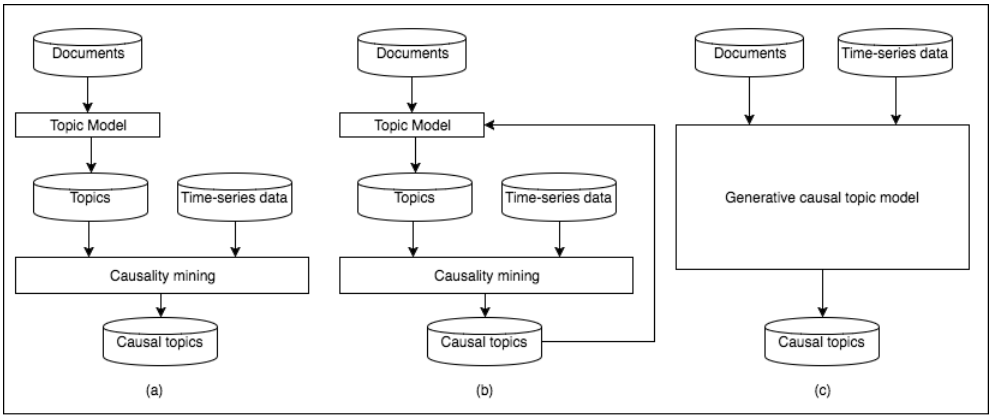


Figure 3.1: Causal topic mining model architectures

Definition 3.1.1. *Causal topics are semantically coherent topics–identified from a collection of timestamped documents–that have a strong, possibly lagged, correlation with the time-series data.*

Causal topic mining is the process of finding such causal topics. The lag that the definition discusses, could be positive or negative depending on whether the document caused the change in the time-series data or vice versa. [1]

3.2 Mining architecture

All three causal topic mining models that were described in Section 2.3 can be classified broadly as confirming to one three different types of architectures shown in Figure 3.1.

Type (a) first generates topics from the collection C and uses causality tests [18] or correlation analysis to yield causal topics. One disadvantage of this model is that any latent information in the time-series data is not utilized in the topic modeling process. That is, the topics would be the same irrespective of what time-series data is used. An advantage of this model would be its simplicity and hence time taken to run. A modified version of IRTSQ [15] would be an example of this category.

[1] Note that the word "causal" is being used loosely in this thesis as we only require correlation and never actually test for actual causality.

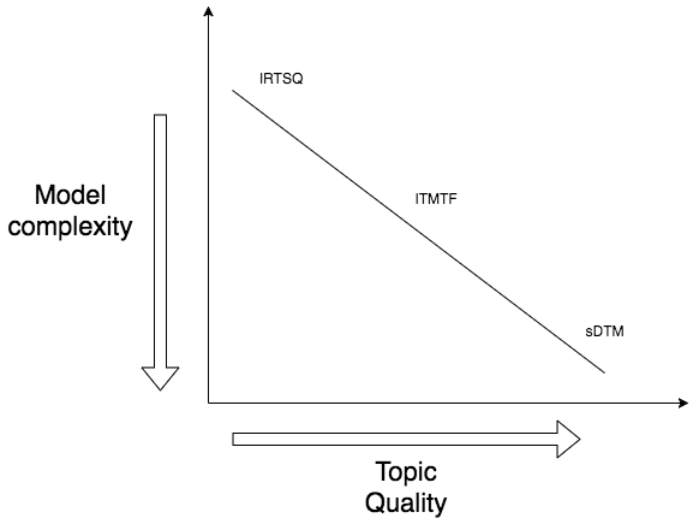


Figure 3.2: Complexity–Quality tradeoff

Type (b) adds an additional step to (a) by feeding the causal topics back to the topic model as prior and iteratively refines the topics found. This is not as fast as (a) and does not generate causal topics of as high a quality as (c). ITMTF [16] is an example of this category.

Type (c) involves feeding both the text and non-text data to a generative model that finds topics by increasing the likelihood of the observed data given the latent topic proportions. The algorithm forces the topics discovered to be correlated with the time-series. sDTM [17] is an example of a causal topic mining algorithm that falls under this category. The shortcoming of a model of this type would be its complexity therefore leading to high running times.

It can be seen that there is a tradeoff between complexity and quality of the causal topics with (a) and (c) at either extremes. Figure 3.2 is a visual depiction of this tradeoff. We now propose two causal topic mining approaches that each belongs to one of the three model types. They lie on either extremes of this tradeoff:

- Clustering-based Causal Topic Mining (cCTM) – Type (a) – We would want cCTM to remain a fast algorithm but generate topics of better quality.

- Generative Causal Topic Model (gCTM), a generative topic model that we use as our baseline – Type (c) – We expect gCTM to be simpler and faster while still generating high quality causal topics.

CHAPTER 4

CLUSTERING BASED CAUSAL TOPIC MINING

In this chapter, we describe the clustering based Causal Topic Mining (cCTM), a novel way to find causal topics along with its different variants. According to Definition 3.1.1, a cluster of words has to satisfy the following two requirements to be considered a causal topic

1. The words should be semantically coherent – indicated by co-occurrence
2. The occurrence of the topic should correlate with the time-series

It has been shown that words can be clustered into groups of words that co-occur [19, 14]. Therefore, to enforce the first requirement, rather than using a topic model like most existing algorithms do, we use a clustering algorithm instead. This vastly brings down the complexity of algorithm–leading to improvements in running time, understandability and implementation – as compared to using a topic model. The second requirement is enforced using a correlation test that we define later in Section 4.2.1.

4.1 Clustering first or correlation first?

Since we only require that both requirements of causal topics are met, either clustering or correlation can be performed first, albeit leading to different topics and properties. Intuitively, performing clustering first would generate topics from the document collection and the correlation test would identify topics that are correlated. If the correlation test were to be performed first, we would find words that correlate with the time-series and then cluster them into topics. Clustering-first and correlation-first approaches will be referred to as cCTM-CF and cCTM-CoF henceforth. We'll now describe the algorithms for both the approaches.

4.2 Clustering first

While many different clustering methods can be used, we use a spectral clustering algorithm called the K-Spectral Centroid algorithm [20]. Spectral clustering is a technique that reduces the dimensionality of the data and clusters the input points in the reduced space using the similarity matrix of the data. The general way to perform spectral clustering is as follows: If the similarity between n datapoints is described by the symmetric similarity matrix S where the element S_{ij} represents a quantitative measure of the similarity between the i^{th} and j^{th} data points, use a standard clustering algorithm like K-Means [21] on the relevant eigenvectors of A to generate clusters. The following spectral method clusters the words in our vocabulary based on their occurrence trend over time. In other words, we want clusters of words that change in a similar way with time. These clusters are ranked according to their correlation with the time-series data. The topics which correlate significantly (negatively or positively) are causal topics. The causality metric called the Pearson correlation coefficient will be defined in Section 4.2.1.

We'll first describe the K-spectral centroid clustering algorithm that is used. The algorithm was formulated by Yang and Leskovec [20] as a way to analyze the dynamics of online content. The distane metric used by the algorithm is time and shift-invariant leading to clusters of datapoints with similar variational trends over time irrespective of the actual count of occurrence. The K-SC algorithm has since been used in a variety of tasks involving social media like studying the evolution of hashtags [22] and forecasting popularity of news [23]. Given two time series x and y, if $y_{(q)}$ represents y shifted by q units, the distance metric $\hat{d}(x, y)$ used by the $K - SC$ algorithm is defined as follows:

$$\hat{d}(x, y) = \min_{\alpha, q} \frac{||x - \alpha y_{(q)}||}{||x||} \tag{4.1}$$

where $|| \cdot ||$ represents the $L2$ norm. Note that this distance metric is symmetric and will not change if x and y are interchanged. α is the scale factor for y and its optimal value can be determined for a given q as follows

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \frac{||x - \alpha y_{(q)}||}{||x||} = \frac{x^T y_{(q)}}{||y_{(q)}||^2} \tag{4.2}$$

Algorithm 1 Clustering-first clustering-based Causal Topic Model

Require: Document collection with timestamps $C = \{(d_1, ts_1), (d_2, ts_2), \ldots, (d_D, ts_D)\}$, time-series data $x = \{(x_1, t_2), (x_2, t_2), \ldots, (x_M, t_M)\}$, Number of topics T.

Compute time-series $c_{w_i} \forall i \in 1, \ldots, V$. The j^{th} element in c_{w_i} is the number of times w_i occurs in any document with timestamp t_j
Assign words randomly to the K clusters $\{C_1, C_2, \ldots, C_K\}$, $K \in \mathbb{Z}^+$
repeat
    $\hat{C} \leftarrow C$
    for j = 1 to K **do**
        $M \leftarrow \sum_{x \in C_j} \left(I - \frac{x \cdot x^T}{||x||^2}\right)$
        $\mu_j \leftarrow$ Eigenvector corresponding to the smallest Eigenvalue of M
        $C_j \leftarrow \emptyset$
    end for
    for i = 1 to N **do**
        $j^* \leftarrow \arg\min_{j=1 \text{ to } K} \hat{d}(x_i, \mu_j)$
        $C_{j^*} \leftarrow C_{j^*} \bigcup \{i\}$
    end for
    this
until $\hat{C} = C$
for i = 1 to K **do**
    $score_i \leftarrow$ Pearson correlation coefficient of x and μ_i
end for
$t \leftarrow T$ clusters with maximum magnitude score
return t

The optimal value of q can only be found by a linear search. The significance of q in our topic modeling application dictates that this search has to be done only for a very small number of values. With the distance function defined, we can now use an approach similar to the K-Means algorithm–the K-SC algorithm has an assignment and an update step. Here is a brief description

- The assignment step assigns words to clusters based on the position of the centroids. This step is straightforward as we just assign every word to the closest centroid's cluster.

- The update step updates the centroids based on the new cluster assignments. Again, like the K-Means algorithm, centroids are computed by minimizing the sum of square of distances of every point to the respective centroid. This optimization leads to the following update equation – If C_k refers to the set of points assigned to cluster k, the

new centroid of the k^{th} cluster is the eigenvector corresponding to the smallest eigenvalue of the following matrix M:

$$M = \sum_{x \in C_k} \left(I - \frac{x \cdot x^T}{||x||^2} \right) \tag{4.3}$$

where I is an identity matrix of appropriate dimensions.

The topics are ranked according to the magnitiude of the Pearson correlation coefficient between the time-series and the centroid identified by the K-SC algorithm. The top T topics are returned as the causal topics. Refer Algorithm 1 for the pseudocode.

4.2.1 Pearson correlation

Pearson correlation coefficient is a metric that determines if two (time) series increase or decrease together significantly. [1] Given two random variables X, Y it is defined as

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{4.4}$$

where $cov(\cdot)$ represents the covariance function and $E[X]$ stands for the expected value of the random variable X. High magnitudes of ρ indicates high correlation.

4.3 Correlation first

In this approach, all the V words in the vocabulary are ranked according to their Pearson correlation coefficient with the time-series. All words with a correlation of greater than δ_{pos} or lesser than δ_{neg} are considered. We then cluster words based on their co-occurrence. That is, if $count(w_i, w_j)$ represents the number of times words w_i and w_j occur in the same sentence, for a word w_i we compute $\max_{w_j \in C_j} count(w_i, w_j)$ and if this value is lesser than δ_{count} we add w_i to cluster C. If not, we create a new cluster with only w_i in it. We return all the resulting clusters as the list of causal topics.

[1] https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

Algorithm 2 Correlation-first clustering-based Causal Topic Model algorithm

$\text{corr}_i \leftarrow$ Pearson correlation between word count time-series for word $w_i \forall i \in \{1, \ldots, V\}$
$S_{pos} \leftarrow \{w_i \forall i \in 1, \ldots, V \text{ if } \text{corr}_i \geq \delta_{pos}\}$
$S_{neg} \leftarrow w_i \forall i \in 1, \ldots, V \text{ if } \text{corr}_i \leq \delta_{neg}$
$\text{CT} \leftarrow \{\}$
for $S \in \{S_{pos}, S_{neg}\}$ **do**
    $\text{C} \leftarrow \{\}$
    for word $w_i \in S$ **do**
        if C is empty **then**
            $C \leftarrow C \bigcup \{\{w_i\}\}$
        else
            $\hat{C} \leftarrow \text{argmax}_{C_j \in C}\{\max_{w_j \in C_j} count(w_i, w_j)\}$
            if $\max_{w_j \in \hat{C}} count(w_i, w_j) \geq \delta_{count}$ **then**
                $\hat{C} \leftarrow \hat{C} \bigcup w_i$
            end if
        end if
    end for
    $CT \leftarrow CT \bigcup C$
end for

Intuitively, this can be thought of as an agglomerative clustering [24] of words where the proximity metric is the co-occurrence is a pair of words. Refer Algorithm 2 for a more detailed pseudocode.

CHAPTER 5

BASELINE METHOD: GENERATIVE CAUSAL TOPIC MODEL

In this chapter, we describe the baseline model that we use – generative Causal Topic Model (gCTM).

5.1 Intuition

We want to model the time-series jointly with the word co-occurrences by correlating both. This is under the assumption to the overall problem that there are latent events that affect both the documents and the time-series. As in any generative model, we want to find parameters that maximize the likelihood of the observed data. Very similar to the TOT agorithm described in Chapter 2, we model each time-series datapoint to be generated from the topic proportion θ along with the topics, thereby driving parameter estimation to find topics that "explain" the change in time-series. Similar to how TOT found topics that appeared for different periods of time, gCTM will find topics that appear corresponding to different rates of change in the time series. An example of this could be finding topics pertaining to Apple CEO Steve Job's death to occur for high drops in the company's stock prices.

5.2 Generative process

Please refer to Chapter 2 for the description of LDA as we build on top of it. Table 5.1 summarizes the notations used in this chapter.

We replace the timestamps in TOT, with the time series data, x to correlate topics with the time-series. Just like in TOT, x is a continuous variable modeled by a Beta distribution normalized to a range of 0 to 1. This is quite convenient as discretizing brings in additional questions about the bin size and width, adding an additional parameter to tune which will affect the

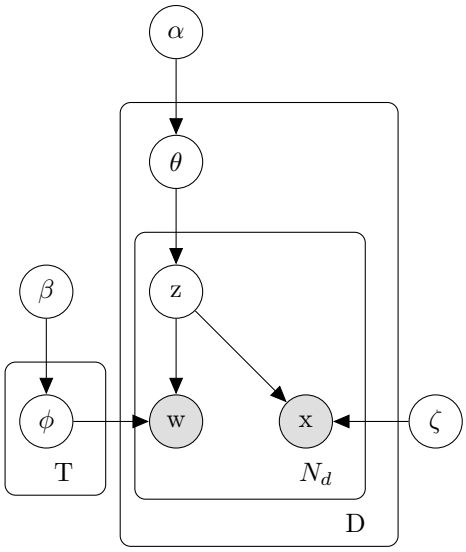


Figure 5.1: Graphical representation of gCTM

results produced. Later, in the experiments section, we describe the difference between considering the time series data, x itself or Δx – the rate of change of the time series. For now, we use the former to explain the generative process. Figure 5.1 shows the plate diagram of the graphical model we use to perform approximate inference using Gibbs sampling. It corresponds to the following generative process:

1. Draw T multinomials, ϕ_z over all the V words in the vocabulary, one for each topic. $\phi_z|\beta \sim \text{Dirichlet}(\beta)$

2. For every document d, draw a multinomial θ_d from the Dirichlet prior α: $\theta_d|\alpha \sim \text{Dirichlet}(\alpha)$ For every word w_{d_i} in the document,

   (a) From the multinomial θ_d, draw a topic $z_{d_i}|\theta_d \sim \text{Multinomial}(\theta_d)$

   (b) From $\theta_{z_{d_i}}$, draw a word $w_{d_i}|\phi_{z_{d_i}} \sim \text{Multinomial}(\phi_{z_{d_i}})$

   (c) From the beta distribution $\zeta_{z_{d_i}}$, draw a time-series value $x_{d_i}|\zeta_{z_{d_i}} \sim \text{Beta}(\zeta_{z_{d_i}})$.

Notice that we draw a time-series datapoint for every word rather than one every document. We do this to make inference possible. In our experiments we duplicate the time-series datapoint for every word in the document. This workaround comes with a cost – we lose the generative capacity of the model as generated documents will have a different timestamp for every word. The

Table 5.1: Notations for gCTM

D	Number of documents
T	Number of topics
V	Number of words in vocabulary
N_d	Number of words in document d
θ_d	Multinomial distribution over all topics for document d
ϕ_z	Multinomial distribution over all words for topic z
ζ_{d_i}	Beta distribution for the time-series for topic z_{d_i}
z_{d_i}	Topic drawn for generating the i^{th} word in document d
w_{d_i}	The i^{th} word in document d
x_{d_i}	The time-series datapoint associated with the i^{th} word in document d

Algorithm 3 Gibbs sampling inference for gCTM

```
for i = 1 to N_iter do
    for d = 1 to D do
        for w = 1 to N_d do
            draw z_dw from P(z_dw | w, t, z_-dw, α, β, ζ)
            update n_{z_dw w} and m_{d z_dw}
        end for
    end for
    for z = 1 to T do
        update ζ_z
    end for
end for
compute the posterior estimates of θ and φ
```

values of the hyperparameters α and β can be updated using the Gibbs EM algorithm [25]. For simplicity, the hyperparameters are set as follows: $\alpha = 50/T$, $\beta = 0.1$.

5.3 Inference

Just like in TOT, we do approximate inference using Gibbs sampling and estimate the Beta distributions ζ_z using Method of Moments estimation to aid in speed and simplicity. The conditional probability distribution can be

expressed as follows

$$
\begin{aligned}
P(z_{d_i}|\mathbf{w}, \mathbf{x}, \mathbf{z_{-d_i}}, \alpha, \beta, \zeta) &\propto (m_{dz_{d_i}} + \alpha_{z_{d_i}} - 1) \\
&\times \frac{n_{z_{d_i} w_{d_i}} + \beta_{w_{d_i}} - 1}{\sum_{v=1}^{V} (n_{z_{d_i} v} + \beta_{w_{d_i}}) - 1} \frac{(1 - x_{d_i})^{\zeta_{z_{d_i}1} - 1} (x_{d_i})^{\zeta_{z_{d_i}2} - 1}}{B(\zeta_{z_{d_i}1}, \zeta_{z_{d_i}2})}
\end{aligned} \tag{5.1}
$$

where n_{zv} is the number of tokens of word v that are assigned to topic z, m_{d_z} represents the number of tokens in document d that are assigned to topic z. If the parameters of the Beta distribution $\zeta_{z_{d_i}}$ are $\zeta_{z_{d_i}1}$ and $\zeta_{z_{d_i}2}$, the Method of Moments estimation gives us

$$
\begin{aligned}
\zeta_{z_{d_i}1} &= \bar{x_z} \Big(\frac{\bar{x_z}(1 - \bar{x_z})}{{s_z}^2} - 1 \Big) \\
\zeta_{z_{d_i}2} &= (1 - \bar{x_z}) \Big(\frac{\bar{x_z}(1 - \bar{x_z})}{{s_z}^2} - 1 \Big)
\end{aligned} \tag{5.2}
$$

The derivation of these update equations for these parameters is very similar to that in TOT [14] and hence we skip them for brevity. Refer Algorithm 3 for the inference pseudocode.

CHAPTER 6

EXPERIMENTAL EVALUATION

In this chapter, we describe the experiments we ran with the models described, the results of these experiments and its evaluation. We'll also describe the datasets that we used in these experiments and their characteristics. Our goal here is to analyze the causal topics generated by each algorithm and compare their performance. We also want to test the performance of the algorithms in downstream tasks to assess them quantitatively.

6.1 Datasets

The datasets that we use in the experiments are as follows

- NYT – Around 12,000 news articles from the New York Times for a period of 6 months, from April to September 2003[1]. We use this dataset along with historical stock prices from the same period for various companies downloaded from the Yahoo! Finance website[2].
- SOU – The State of Union address by Presidents of United States of America for 21 decades downloaded from the Gutenberg project website [3]. This dataset is studied along with the historical unemployment rates made public by the United States Department of Labor. [4]

We use the following notation for convenience – NYT-AAPL refers to using the New York Times dataset along with Apple stock prices and SOU-UNEMP refers to using the SOU text dataset with the historical unemployment rates. Figure 6.1 shows a visual representation of the number of unique words in each text dataset. The average sentence length after removing stop words is

[1]https://catalog.ldc.upenn.edu/LDC2008T19
[2]https://finance.yahoo.com/quote/AAPL/
[3]http://www.gutenberg.org/ebooks/5050
[4]https://data.bls.gov/timeseries/LNS14000000

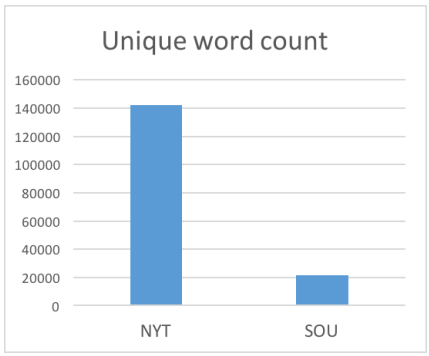


Figure 6.1: The number of unique words in each text dataset

17 words in the NYT dataset and 20 in the SOU dataset. While the NYT dataset has an overlap of 124 days with the AAPL stock data we use, the SOU-UNEMP overlap is 60 datapoints.

6.2 Setup

In this section we'll explain the experiments we run to test each of the algorithms proposed in Section 3.2 along with the datasets that we tested each algorithm on. All experiments were run on a Linux server with 24 cores and 65GB main memory. The number of topics was set at 40 for all the experiments. All iterative experiments are run for 200 epochs.

We use cCTM-CF and cCTM-CoF to denote the clustering first and correlation first methods introduce in Chapter 4 and gCTM to denote the baseline model in Chapter 5. We set the number of topics to 30 in cCTM-CF and gCTM. In cCTM-CoF we set δ_{pos} to 0.1, δ_{neg} to -0.1 and δ_{count} to 10^{-5}. We set the parameters in gCTM α and β at $50/T$ and 0.1 respectively as we can not constrain the number of topics. Note that we also do not feed the raw stock prices as the time-series to gCTM but the percentage of the stock prices. This lead to better results and can be explained by the fact that the current stock price is dependent on the previous prices and the topic proportions can only determine the percentage change and not the actual value. Note that we also restrict the size of the vocabulary to the most frequent 3000 words in the collection.

6.3 Qualitative results

Let's first qualitatively look at the causal topics generated by each algorithm.

6.3.1 Clustering first

Score = -0.23	Score = -0.24	Score = -0.16	Score = -0.15	Score = 0.12
legislature	china	vaccine	voter	captain
defendent	negative	pollution	campaign	marine
customer	september	sars	ballot	custody
pound	import	severe	district	lincoln
cash	donation	spam	abortion	headline
...	...	...	...	...

Table 6.1: K-SC topics for NYT-AAPL

Table 6.1 shows representative topics for the NYT-AAPL dataset that were obtained using the K-SC-Pearson method with each topic represented by 5 words. The score for each topic is the Pearson correlation coefficient between the centroid of the topic with the time-series. From a purely qualitative assessment of the words, the table suggests that the clustering algorithm works well by finding words that are related somehow. It would not be too hard to assign topic labels for each of the topics shown in the table. However, there is no clear justification for the correlation score between the topics and the time-series. For example, the third topic which seems to be about the SARS[5] disease outbreak has a negative correlation witht the stock prices of Apple. While there seems puzzling, a closer look tells us that the month of April, 2003 was when Apple's stock prices were at a relative high (due to the release of the iPod classic [6]) and this was also one of the last months of the SARS outbreak[7], which tells us that the event was in-fact correlated.

Table 6.2 shows similar representative topics for the SOU-UNEMP datset. From a first glance, it looks like the topics are more discriminative than for the previous dataset. This could be because of the size of the dataset itself. Again, the algorithm seems to find discriminative topics that can be given

[5]https://en.wikipedia.org/wiki/Severe_acute_respiratory_syndrome
[6]https://en.wikipedia.org/wiki/Timeline_of_Apple_Inc._products
[7]https://en.wikipedia.org/wiki/Timeline_of_the_SARS_outbreak

labels. An interesting trend can be noticed: Topics with words like "military" and "battle" seem to be negatively correlated with the unemployment rate while positive words like "opportunity" and "inspiration" are positively correlated. The explanation to this could be that the unemployment rate typically goes down during wars [8] and Presidents would want to use positive, reassuring words during times when unemployment rates are high.

Score = 0.13	Score = 0.12	Score = -0.16	Score = 0.09	Score = 0.08
recovery	religious	cuba	school	military
opportunities	god	battle	job	protection
inspiration	reliable	territory	technologies	service
deterrance	oldest	war	hire	refugee
recovery	hero	vietnam	computer	revitalize
...	...	...	...	...

Table 6.2: cCTM-CF topics for SOU-Unemployment rate

6.3.2 Correlation first

Before looking at topics obtained from the clustering first approach, let's first consider the words that were found to correlate with the time-series. In Table 6.3 "april" was the most highly correlated word and this is again because of the relatively high prices that Apple stocks were trading at in the month of April, 2003. Words related to the SARS disease outbreak are also highly correlated while words related to the September 11 terror attacks were negatively correlated with the stock prices. We can see that these words were rightly put in the same topic bu the clustering-first approach as well.

Positive words	Score	Negative words	score
april	0.61	sept	-0.70
respiratory	0.51	libor	-0.62
sars	0.49	oct	-0.62
acute	0.47	9/11	-0.48
disease	0.45	hurricane	-0.48

Table 6.3: Highest positive and negatively correlated words for NYT-AAPL

[8] https://www.quora.com/What-happened-to-unemployment-rates-during-the-world-war

Table 6.4 lists words that are highly correlated with the unemployment rate. Again, we see a very observation that positive words correlate positively with unemployment rate and words about war correlate negatively. These words seem to have a high co-occurrence (from the topics in clustering-first) and a high correlation which indicates that the naive clustering algorithm will give us good causal topics.

Positive words	Score	Negative words	score
ensure	0.51	june	-0.45
sector	0.51	korea	-0.44
incentive	0.51	district	-0.42
growth	0.51	men	-0.41
future	0.49	nation	-0.41
recovery	0.49	shall	-0.40
needy	0.49	measure	-0.39
victim	0.48	relate	-0.38
regulatory	0.47	upon	-0.38
spend	0.47	communist	-0.37

Table 6.4: Highest positive and negatively correlated words for SOU-UNEMP

Table 6.5 shows representative causal topics generated by the cCTM-CoF algorithm for the NYT-AAPL dataset. The score for each topic is assigned as the average correlation coefficient of all the words in the topic. We only considered the topics that had atleast 20 words. These causal topics are significantly more discriminative compared to the cCTM-CF algorithm and is also much better in terms of the correlation scores.

Score = 0.16	Score = 0.17	Score = -0.18	Score = -0.17	Score = -0.19
smoke	sars	bush	voter	denver
explode	vaccine	govern	campaign	quarterback
bomb	outbreak	afghanistan	politics	patriot
tank	disease	capitol	regulation	davis
chemical	destroy	blame	rule	transfer
...	...	...	...	...

Table 6.5: cCTM-CoF topics for NYT-AAPL

6.3.3 Baseline-gCTM

Table 6.6 shows us similar representative topics for the same NYT-AAPL dataset by the baseline gCTM model. We can see that the topics are much more discriminative than the two models we propose. It is the easiest to assign labels as well. However, it has to be noted that these are the highest correlated topics among all the causal topics and the correlation scores are much lower than both clustering methods. Eventhough such graphical models are really sensitive to the values of hyperparameters, we found that for this specific application, this behavior was unmodified by the hyperparameter values.

Score = 0.003	Score = 0.002	Score = 0.006	Score = 0.002	Score = -0.0008
hospital	coffee	govern	strike	flood
dna	meat	force	inning	air
expose	egg	troop	baseball	hurricane
virus	recipe	tank	sox	pressure
vaccine	kitchen	uniform	oakland	sky
...	...	...	...	...

Table 6.6: gCTM topics for NYT-AAPL

6.4 Quantitative results

In this section, we perform quantitative evaluation of the causal topics in order to compare algorithms. There are essentially two qualities we want to measure in causal topics: semantic coherence and time-series correlation.

6.4.1 Semantic coherence

	baseline-gCTM	cCTM-CF	cCTM-CoF
coherence	0.006	0.0054	0.0081

Table 6.7: *coherence* scores

This metric tests whether the cluster of words form a coherent meaningful topic. We use the pointwise mutual information between all pairs of words

in the topic. The pointwise mutual information between words w_i and w_j is defined as follows

$$\begin{aligned} PMI(w_i, w_j) &= \log_2 \frac{\Pr(w_i, w_j)}{\Pr(w_i)\Pr(w_j)} \\ &\propto \log_2 \left[\frac{c(w_i, w_j) / \sum_{w_i, w_j \in T} c(w_i, w_j)}{c(w_i)c(w_j) / \left(\sum_{w_i} c(w_i)\right)^2} \right] \end{aligned} \tag{6.1}$$

where $c(w_i, w_j)$ represents the number of times words w_i and w_j occur in the same sentence in the collection and $c(w_i)$ represents the number of times word w_i occurs in the collection. So, the semantic coherence score of topic T can be computed as

$$coherence(T) = \frac{\sum_{w_i, w_j \in T} PMI(w_i, w_j)}{|T||T-1|} \tag{6.2}$$

Table 6.7 shows the average coherence scores for all the topics for each of the three models we discussed.

6.4.2 Time-series correlation

	baseline-gCTM	cCTM-CF	cCTM-CoF
$correlation_{pos}$	0.001	0.00062	0.002
$correlation_{neg}$	-0.0008	-0.00098	-0.0013

Table 6.8: *correlation* scores

The measure of time-series correlation of a topic is just given by the average of time-series correlation measures of all the words. We simply find the avearge correlation of all the words in the topic with the time-series and use their mean as the metric for the topic. For a topic T and time-series x

$$correlation(T) = \frac{\sum_{w \in T} corr(w, x)}{|T|} \tag{6.3}$$

where $corr(\cdot)$ represents the Pearson correlation coefficient (refer Section 4.2.1). We compute $correlation_{pos}$ as the average of correlation all positively correlated topics and $correlation_{neg}$ as the average over negatively correlated topics. Ideally, we want both their magnitudes to be as close to 1 as possible.

Table 6.8 shows the average positive and negative correlation scores over all the topics for each of the three models we discussed.

6.5 Analysis

Let's now analyze the results of these experiments in detail. While the respresentative topics are not very easy to parse and draw conclusions from, due to the sheer size of topics, a quick glance suggested that gCTM generated the most discriminative topics, closely followed by cCTM-CoF. The topics generated by cCTM-CF are not very easy to classify into topics and the reason for this could be the large amount of noise in the count of word occurrences over time. The clustering algorithm works well only for starkly different trends over a long period of time.

The coherence and correlation scores gave us a much more deeper analysis of the algorithms. Surprisingly, we found that cCTM-CoF had the highest coherence score of all the three algorithms eventhough the topics suggested otherwise. One contributing reason to this could be the fact that we restrict the number of topics for gCTM and cCTM-CF to 30 which forces the algorithms to cluster words that are not that semantically related into the same topic. cCTM-CoF does not have this restriction as it assigns words that are very different from others to a separate cluster leading to higher semantic scores. cCTM-CF had the least semantic score which came as less of a surprise given the quality of the topics. cCTM-CoF got the highest correlation score, followed by cCTM-CF and finally gCTM. This stems from the fact that cCTM-CoF essentially restricts the vocabulary to a set of words that are highly correlated with the time-series. It is also noteworthy that cCTM-CoF was by far the fastest algorithm as both the other algorithms are iterative and have high running times. Another disadvantage of both iterative algorithms is that they try to optimize a global loss function which is too expensive and could lead to non-deterministic behavior as there could be multiple local maxima.

CHAPTER 7

CONCLUSION

We presented a novel clustering based algorithm and a baseline generative model for generating causal topics from a collection of documents and time-series. We evaluated these models and found that the cCTM-CoF model generated the best causal topics and achieved a 35% improvement on the *coherence* score and 62.5% improvement on the *correlation* score as compared to the baseline model. We also formulated a conceptual framework for causal topic models thereby making it easier to classify and analyze such algorithms.

CHAPTER 8

REFERENCES

[1] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003. [Online]. Available: http://www.jmlr.org/papers/v3/blei03a.html

[3] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in neural information processing systems*, 2008, pp. 121–128.

[4] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, "The author-topic model for authors and documents," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 2004, pp. 487–494.

[5] G. Gidofalvi and C. Elkan, "Using news articles to predict stock price movements," *Department of Computer Science and Engineering, University of California, San Diego*, 2001.

[6] M. A. Mittermayer, "Forecasting intraday stock price trends with text mining techniques," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, Jan 2004, pp. 10 pp.–.

[7] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of computational science*, vol. 2, no. 1, pp. 1–8, 2011.

[8] G. Mishne and N. Glance, "Predicting movie sales from blogger sentiment," in *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, 2006, pp. 155–158.

[9] S. Asur and B. A. Huberman, "Predicting the future with social media," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, ser. WI-IAT '10. Washington, DC, USA: IEEE Computer Society, 2010. [Online]. Available: http://dx.doi.org/10.1109/WI-IAT.2010.63 pp. 492–499.

[10] S. Sinha, C. Dyer, K. Gimpel, and N. A. Smith, "Predicting the NFL using twitter," *CoRR*, vol. abs/1310.6998, 2013. [Online]. Available: http://arxiv.org/abs/1310.6998

[11] X. Chen, Y. Cho, and S. Y. Jang, "Crime prediction using twitter sentiment and weather," in *2015 Systems and Information Engineering Design Symposium*, April 2015, pp. 63–68.

[12] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis."

[13] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 2006. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143859 pp. 113–120.

[14] 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*. ACM, 2006, pp. 424–433.

[15] H. D. Kim, D. Nikitin, C. Zhai, M. Castellanos, and M. Hsu, "Information retrieval with time series query," in *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ser. ICTIR '13. New York, NY, USA: ACM, 2013. [Online]. Available: http://doi.acm.org/10.1145/2499178.2499195 pp. 14:56–14:63.

[16] H. D. Kim, M. Castellanos, M. Hsu, C. Zhai, T. Rietz, and D. Diermeier, "Mining causal topics in text data: iterative topic modeling with time series feedback," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, ser. CIKM '13. New York, NY, USA: ACM, 2013. [Online]. Available: http://doi.acm.org/10.1145/2505515.2505612 pp. 885–890.

[17] S. Park, W. Lee, and I.-C. Moon, "Supervised dynamic topic models for associative topic extraction with a numerical time series," in *Proceedings of the 2015 Workshop on Topic Models: Post-Processing and Applications*, ser. TM '15. New York, NY, USA: ACM, 2015. [Online]. Available: http://doi.acm.org/10.1145/2809936.2809938 pp. 49–54.

[18] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969. [Online]. Available: http://www.jstor.org/stable/1912791

[19] C. A. Bejan and A. Harabagiu, "Using clustering methods for discovering event structures."

[20] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011. [Online]. Available: http://doi.acm.org/10.1145/1935826.1935863 pp. 177–186.

[21] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979. [Online]. Available: http://www.jstor.org/stable/2346830

[22] J. Lehmann, B. Gonçalves, J. J. Ramasco, and C. Cattuto, "Dynamical classes of collective attention in twitter," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187871 pp. 251–260.

[23] R. Bandari, S. Asur, and B. A. Huberman, "The pulse of news in social media: Forecasting popularity," *arXiv preprint arXiv:1202.0332*, 2012.

[24] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[25] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1, pp. 5–43, 2003.