Reading Documents for Bayesian Online Change Point Detection

Taehoon Kim and **Jaesik Choi**
School of Electrical and Computer Engineering
Ulsan National Institute of Science and Technology
Ulsan, Korea
{taehoon,jaesik}@unist.ac.kr

Abstract

Modeling non-stationary time-series data for making predictions is challenging but important tasks. One of the key issues is to identify long-term changes accurately in time-varying data. *Bayesian Online Change Point Detection (BO-CPD)* algorithms efficiently detect long-term changes without assuming the Markov property which is vulnerable to local signal noises. We propose *Document based BO-CPD (DBO-CPD)* model which automatically detect long-term temporal changes of continuous variables based on a novel dynamic Bayesian analysis which combines a non-parametric regression, the Gaussian Process (GP), with generative models on texts such as news articles and posts on social networks. Since texts often include important clues of signal changes, DBO-CPD enables to predict long-term changes accurately. We show that our algorithm outperforms existing BO-CPDs in two real-world datasets: stock prices and movie revenues.

1 Introduction

Time series data depends on the latent dependence structure which changes over time. Thus, stationary parametric models are not appropriate to represent such dynamic non-stationary processes. Change point analysis (Smith, 1975; Stephens, 1994; Chib, 1998; Barry and Hartigan, 1993) focuses on formal frameworks to determine whether a change has taken place without assuming the Markov property which is vulnerable to local signal noises. When change points are identified, each part of the time series are approximated by specified parametric models under the stationary assumptions. Such change point detection models are successfully applied to a variety of data, such as stock markets (Chen and Gupta, 1997; Hsu, 1977; Koop and Potter, 2007), analyzing bees' behavior (Xuan and Murphy, 2007), forecasting climates (Chu and Zhao, 2004; Zhao and Chu, 2010) and physics experiments (von Toussaint, 2011). However, offline-based change points analysis suffers from slow retrospective inference which prevents real-time analysis.

Bayesian Online Change Point Detection (BO-CPD) (Adams and MacKay, 2007; Steyvers and Brown, 2005; Osborne, 2010; Gu et al., 2013) overcomes this restrictions by exploiting online efficient inference algorithms. BO-CPD algorithms efficiently detect long-term changes by analyzing continuous target values with the Gaussian Process (GP), a non-parametric regression method. The GP-based CPD model is simple and flexible. However, it is not straightforward to utilize the rich categorical data such as texts in news articles an posts in social networks.

In this paper, we propose a novel BO-CPD model that enables to improve the detection of change points in continuous signals by incorporating the rich external information implicitly written in texts on top of the long-term changes analysis of the GP. Especially, our model finds causes of signal changes in news articles in that news is one of the most influential sources of markets of interests.

Given a set of news articles extracted from the Google News service and a sequence of target, continuous values, our new model, Document-based Bayesian Online Change Point Detection (DBO-CPD) learns generative models which represents a probability of a news article given the run length (a length of consecutive observations without a change). By using the new prior, DBO-CPD models a dynamic hazard rate (h) which determines the rate at which change points occur.

In the literature, important information is extracted from news articles (Nothman et al., 2012; Schumaker and Chen, 2009; Gidófalvi and Elkan,

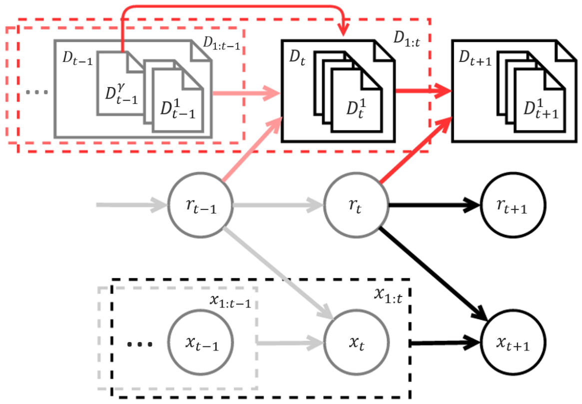


Figure 1: This figure illustrates a graphical representation of our DBO-CPD models. x_t, r_t, and D_t respectively represents a continuous variable of interest, the run length (hidden) variable, and documents. Our modeling contribution is to add texts $D_{1:t}$ to make an accurate prediction of the run length r_{t+1}.

2001; Fung et al., 2003; Fung et al., 2002; Schumaker and Chen, 2006), tweet of Twitter (Si et al., 2013; Wang et al., 2012; Bollen et al., 2011; St Louis et al., 2012), online chats (Kim et al., 2010; Gruhl et al., 2005) and blog posts (Peng et al., 2015; Mishne and Glance, 2006).

In experiments, we show that DBO-CPD can effectively distinguish whether it is a change point or not in real-world datasets (see Section 3.1). Compared to previous BO-CPD models which explain the changes by human manual mappings, our DBO-CPD automatically explains the reasons of why a change point has been occurred by connecting the numerical sequence of data and textual features of news articles.

2 Bayesian Online Change Point Detection

This section will review our research problem, the change point detection (CPD) (Barry and Hartigan, 1993), and the model, Bayesian Online Change Point Detection (BO-CPD) (Adams and MacKay, 2007).

Let $x_t \in \mathbb{R}$ be a data observation at time t. We assume that a sequence of data $(x_1, x_2, ..., x_t)$ is composed of several non-overlapping productive partitions (Barry and Hartigan, 1992). The boundaries that separate the partitions is called the change points. Let r be the random variable that denotes the *run length* which is the number of time steps since the last change point is detected. r_t is the current run at time t. $x_t^{(r_t)}$ denotes the most recent data corresponding to run r_t.

2.1 Online Recursive Detection

To make an optimal predictions of the next data x_{t+1}, one may need to consider all possible run lengths $r_t \in \mathbb{N}$ and a probability distribution over run length r_t. Given a sequence of data up to time t, $x_{1:t} = (x_1, x_2, ..., x_t)$, the run length prediction problem is formalized as computing the joint probability of random variables $P(x_{t+1}, x_{1:t})$. This distribution can be calculated in terms of the posterior distribution of run length at time t, $P(r_t|x_{1:t})$, as follows,

$$\begin{aligned} P(x_{t+1}, x_{1:t}) &= \sum_{r_t} P(x_{t+1}|r_t, x_t^{(r_t)}) P(r_t|x_{1:t}) \\ &= \sum_{r_t} P(x_{t+1}|x_t^{(r_t)}) P(r_t|x_{1:t}) \quad (1) \end{aligned}$$

The predictive distribution $P(x_{t+1}|r_t, x_t^{(r_t)})$ depends only on the most recent r_t observations $x_t^{(r_t)}$. The posterior distribution of run length $P(r_t|x_{1:t})$ can be computed recursively:

$$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{P(x_{1:t})}, \quad (2)$$

where

$$P(x_{1:t}) = \sum_{r_t} P(r_t, x_{1:t}). \quad (3)$$

The joint distribution over run length r_t and data $x_{1:t}$ can be derived by summing $P(r_t, r_{t-1}, x_{1:t})$ over r_{t-1}:

$$\begin{aligned} P(r_t, x_{1:t}) &= \sum_{r_{t-1}} P(r_t, r_{t-1}, x_{1:t}) \\ &= \sum_{r_{t-1}} P(r_t, x_t|r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(r_t|r_{t-1}) P(x_t|r_{t-1}, x_t^{(r_t)}) P(r_{t-1}, x_{1:t-1}). \end{aligned}$$

This formulation updates the posterior distribution of the run length given the prior over r_t from r_{t-1} and the predictive distribution of new data.

However, the existing BO-CPD model (Adams and MacKay, 2007) specifies the conditional prior on the change point $P(r_t|r_{t-1})$ in advance. This approach may lead to model biased predictions because the update formula highly relies on the predefined, fixed hazard rate (h). Furthermore, BO-CPD is incapable to incorporate external information that implicitly influences the observation and explains the reasons of the current change of the long-term trend.

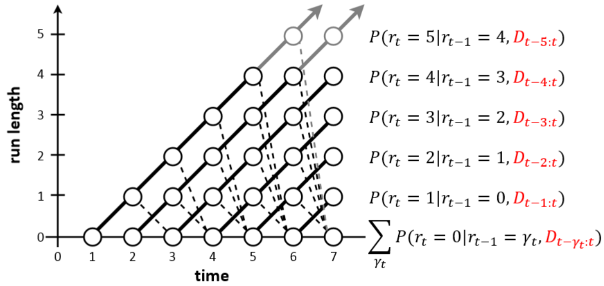


Figure 2: This figure illustrates the recursive updates of the posterior probability in DBO-CPD model.

2.2 Document-based Bayesian Online Change Point Detection

This section explains our DBO-CPD model. To represent the text documents, we add a variable D which denotes a series of text document related to the observed data as shown in Figure 1. Let D_t be a set of N_t text documents $D_t^1, D_t^2, ..., D_t^{N_t}$ that are indexed at time of publication t, where N_t is the number of documents observed at time t. Then, we can rewrite the joint probability over run length as,

$$P(r_t, x_{1:t}) = \sum_{r_{t-1}} \sum_{D_t^{(r_{t-1})}} P\left(r_t | r_{t-1}, D_t^{(r_{t-1})}\right) \cdot P\left(x_t | r_{t-1}, x_t^{(r_{t-1})}\right) P(r_{t-1}, x_{1:t-1}) \quad (4)$$

where $D_t^{(r_t)}$ ($= D_{t-r_t+1:t}$) is the set of most recent r_t documents. Figure 2 illustrates the recursive updates of posterior probability where solid lines indicate that the probability mass is passed upwards and dotted lines indicate the probability that the current run length r_t is set to zero.

Given documents $D_t^{(r_t)}$, the conditional probability is represented as follows,

$$\begin{aligned} & P\left(r_t = \gamma{+}1 | r_{t-1} = \gamma, D_t^{(\gamma)}\right) \\ &= \frac{P\left(r_{t-1} = \gamma, D_t^{(\gamma)} | r_t = \gamma{+}1\right) P(r_t = \gamma{+}1)}{\sum_{\bar{\gamma}=1}^{\gamma+1} P\left(r_{t-1} = \gamma, D_t^{(\gamma)} | r_t = \bar{\gamma}\right) P(r_t = \bar{\gamma})} \\ &= \frac{P\left(r_{t-1} = \gamma, D_t^{(\gamma)} | r_t = \gamma + 1\right) P_{gap}(\gamma{+}1)}{\sum_{\bar{\gamma}=1}^{\gamma+1} P\left(r_{t-1} = \gamma, D_t^{(\gamma)} | r_t = \bar{\gamma}\right) P_{gap}(\bar{\gamma})} \end{aligned}$$

where P_{gap} is the distribution of intervals between consecutive change-points. As the BO-CPD model (Adams and MacKay, 2007), we assume the simplest case where the probability of a change-point at every step is assumed constant where the length of a segment is modeled by a discrete exponential (geometric) distribution as

$$P_{gap}(r_t|\lambda) = \lambda exp^{-\lambda r_t}, \quad (5)$$

where $\lambda > 0$, a *rate parameter*, is the parameter of the distribution.

The update rule for the prior distribution on r_t makes the computation of the joint distribution tractable, $\sum_{\bar{\gamma}=1}^{\gamma+1} P(r_{t-1}{=}\gamma, D_t^{(\gamma)} | r_t{=}\bar{\gamma}) \cdot P_{gap}(\bar{\gamma})$. Because r_t can only be increased to $\gamma + 1$ or set to 0, the conditional probability is as follows,

$$\begin{aligned} & P(r_t = \gamma + 1 | r_{t-1} = \gamma, D_t^{(\gamma)}) \\ &= \frac{T_D(t, \gamma|\gamma{+}1)}{T_D(t, \gamma|\gamma{+}1) + T_D(t, \gamma|0)}, \end{aligned} \quad (6)$$

where the function $T_D(t, \alpha|\bar{\alpha})$ is an abbreviation of $P\left(r_{t-1}{=}\alpha, D_t^{(\alpha)} | r_t{=}\bar{\alpha}\right)$. In Equation (6), $T_D(t, \gamma|\gamma{+}1){=}P(r_{t-1}{=}\gamma, D_t^{(\gamma)} | r_t{=}\gamma{+}1)$ is the joint probability of the run length r_{t-1} and a set of documents $D_t^{(\gamma)}$ when no changed is occurred at time t and the run length becomes $\gamma + 1$. Therefore, we can simplify the equation by removing $r_{t-1}{=}\gamma$ from the condition as follows,

$$T_D(t, \gamma|\gamma{+}1) = P(D_t^{(\gamma)} | r_t{=}\gamma + 1). \quad (7)$$

We represent the distribution of words by the *bag-of-words* model. Let D_t^i be the set of M words that is part of ith document at time t, i.e. $D_t^i = \{d_t^{i,1}, d_t^{i,2}, ..., d_t^{i,M}\}$. In the model, we assume that each word $d_t^{i,j}$ is independent and identically distributed (iid) given a run length parameter r_t. In this setting, the conditional probability of the words takes the following form:

$$P\left(D_t^{(\gamma)} | r_t = \gamma{+}1\right) = \frac{1}{Z} \prod_{i,j} P\left(d_t^{i,j} | r_t = \gamma{+}1\right). \quad (8)$$

The conditional probability $P(d_t^{i,j} | r_t{=}\gamma{+}1)$ is represented by two generative models, $\phi_{\mathbf{w}}$ and $\phi_{\mathbf{emp}}$. The probability $P\left(D_t^{(\gamma)} | r_t{=}\gamma + 1\right)$ can be represented recursively as,

$$\begin{aligned} P\left(D_t^{(\gamma)} | r_t{=}\gamma{+}1\right) &= P\left(D_t^{(\gamma)} | \gamma{+}1\right) \\ \propto \quad & \phi_{\mathbf{w}}(D_t^{(\gamma)} | \gamma{+}1) \cdot \phi_{\mathbf{emp}}(D_t^{(\gamma)} | \gamma{+}1) \end{aligned}$$

$$
\begin{aligned}
&= \phi_{\mathbf{w}}(D_t|\gamma{+}1) \cdot \phi_{\mathbf{emp}}(D_t|\gamma{+}1) \\
&\quad \cdot \phi_{\mathbf{w}}(D_{t-1}^{(\gamma-1)}|r_{t-1}{=}\gamma) \cdot \phi_{\mathbf{emp}}(D_{t-1}^{(\gamma-1)}|r_{t-1}{=}\gamma) \\
&= \prod_{i,j} \phi_{\mathbf{w}}(d_t^{i,j}|\gamma{+}1) \cdot \phi_{\mathbf{emp}}(d_t^{i,j}|\gamma{+}1) \\
&\quad \cdot \phi_{\mathbf{w}}(D_{t-1}^{(\gamma-1)}|r_{t-1}{=}\gamma) \\
&\quad \cdot \phi_{\mathbf{emp}}(D_{t-1}^{(\gamma-1)}|r_{t-1}{=}\gamma) \qquad (9)
\end{aligned}
$$

where

$$\phi_{emp}(d_t^{x,y}|\gamma) = \frac{count(d_t^{x,y}, r_t = \gamma)}{\sum_{i,j} count(d_t^{i,j}, r_t = \gamma)}.$$

Here, $\phi_{\mathbf{w}}(d_t^{x,y}|\gamma)$ and $\phi_{\mathbf{emp}}(d_t^{x,y}|\gamma)$ are empirical potentials which contribute to represent $P(d_t^{i,j}|\gamma)$. $\phi_{\mathbf{w}}(\cdot)$ is explained in Section 2.3. Here, $count(E)$ is the number of event E is appeared in dataset. In Equation (9), τ_t is the time gap (difference) between t and the time when a document is generated, and $d^{i,j}$ represents a document without considering the time domain and the evidence $Z = P(d^i)$ is a scaling factor which depends only on $d^{i,1}, d^{i,1}, ...d^{i,M}$.

$T_D(t, \gamma|0)$ is represented as follows,

$$
\begin{aligned}
&= P(r_{t-1}{=}\gamma, D_t^{(\gamma)}|r_t{=}0) \\
&= P(r_{t-1}{=}\gamma|r_t = 0)P(D_t^{(\gamma)}|r_t{=}0) \\
&= H(\gamma{+}1)P(D_t^{(\gamma)}|r_t{=}0),
\end{aligned}
$$

where $H(\tau)$ is the *hazard function* (Forbes et al., 2011),

$$H(\tau) = \frac{P_{gap}(\tau)}{\sum_{t=\tau}^{\infty} P_{gap}(t)}. \qquad (10)$$

When P_{gap} is the discrete exponential distribution, the hazard function is constant at $H(\tau) = 1/\lambda$ (Adams and MacKay, 2007).

As an illustrative example, suppose that we found a rapid change at Google stock three days ago. Today at $t = 3$, we want to know how the articles are written and whether it affects the change on tomorrow ($t = 4$). As shown in Figure 3, we can calculate what degree a word, for example *rises* or *stays*, is likely to be appeared in articles published since today, which is $P(D_t^{(\gamma)}|r_t = \gamma{+}1)$, and this probability leads us to predict run lengths from the texts. Documents for each $\tau_t = 0, 1$ and 2 are generated from the generative models with a given predicted run length through recursive calculation of the Bayesian models which enables online prediction as shown in Equation (9). This is our main contribution of this paper that enables DBO-CPD to inference online change points accurately with documents.

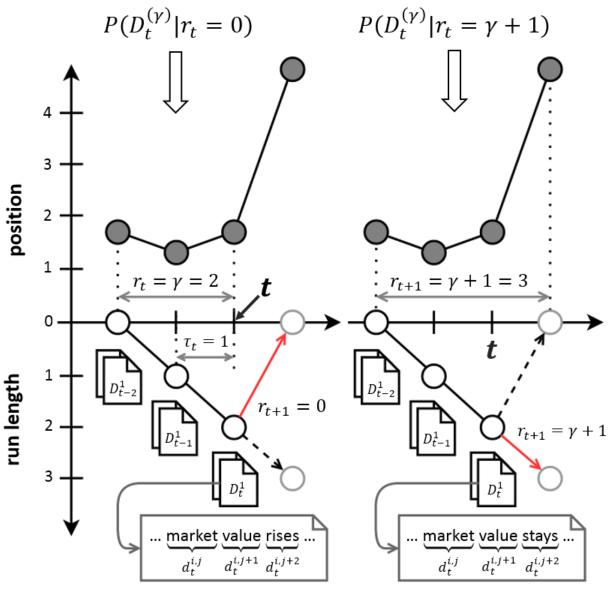


Figure 3: This figure illustrates how our Equation (9) is calculated and how it determines whether a change is occur or not.

2.3 Generative Models Trained from Regression

Let $D \in \mathbb{R}^{T \times N \times M}$ be N documents of news articles which is consisted of M vocabulary over time domain T, $D_t^i \in \mathbb{R}^M$ be the ith document of a set of documents generated at time t, and define $r \in \mathbb{R}^N$ as the corresponding set of the run length, which is a time gap between the document is generated and the next change point is occurred. Then, given a text document D_t^i, we seek to predict the value of run length r by learning a parameterized function f:

$$\hat{r} = f(D_t^i; \mathbf{w}) \qquad (11)$$

where $\mathbf{w} \in \mathbb{R}^d$ are the weights of text features for $d_t^{i,1}, d_t^{i,2}, ..., d_t^{i,M}$ which compose documents D_t^i. From a collection of N documents, we use linear regression which is trained by solving the following optimization problem:

$$\min_{\mathbf{w}, D_t^i} f(D_t^i; \mathbf{w}) \equiv \mathbf{C} \sum_{\mathbf{i=1}}^{\mathbf{N}} \xi(\mathbf{w}, \mathbf{D_t^i}, \mathbf{r_t}) + \mathbf{r}(\mathbf{w}), \qquad (12)$$

where $r(\mathbf{w})$ is the regularization term and $\xi(\mathbf{w}, \mathbf{D_t^i}, \mathbf{r_t})$ is the loss function. Parameter $C > 0$ is a user-specified constant for balancing $r(\mathbf{w})$ and the sum of losses.

Let h be a function from a document into a vector-space representation $\in \mathbb{R}^d$. In linear regres-

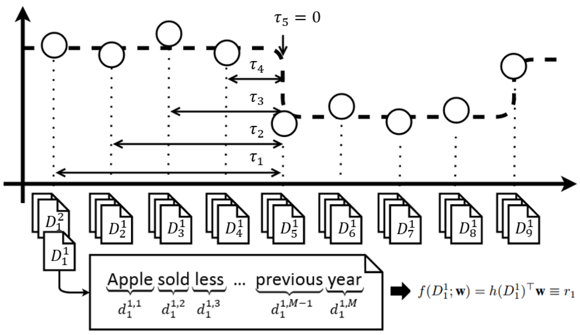


Figure 4: This figure illustrates a graphical representation of how we train generative models from regression.

sion, the function f takes the form:

$$f(D_t^i; \mathbf{w}) = h(D_t^i)^\top \mathbf{w} + \epsilon \tag{13}$$

where ϵ is Gaussian noise.

Figure 4 illustrates how we trained a linear regression model on a sample article. One issue is that the run length can not be trained directly. Suppose that we train $r_5 = 0$ into regression, the weight $\mathbf{w}$ of the model will become 0 even though the words contained in $D_5^j, \forall j \in \{1, ..., T\}$ is composed of salient words which can cause a possible future change point. To solve this interpretability problem, we trained the weight in the inverse exponential domain for the predicted variable, predicting e^{-r_t} instead of r_t. In this setting, the predicted run-length takes the form:

$$e^{-\hat{r_t}} = h(D_t)^\top \mathbf{w} + \epsilon. \tag{14}$$

By this methods, regression model can give a high weight to a word which appears often close to change points. We can interpret that highly weighted words d are more closely related to an outbreak of changes than lower weighted words.

With $\mathbf{w}$, we can rewrite the probability of d, τ_t given $\mathbf{w}$ as:

$$\begin{aligned} \phi_{\mathbf{w}}(d, \tau_t) &\propto \mathbf{w}_d \cdot (\exp(-1/\mathbf{w}_d))^{\tau_t} \\ &= \mathbf{w}_d \cdot \exp(-\tau_t/\mathbf{w}_d). \end{aligned} \tag{15}$$

The potential, $\phi_{\mathbf{w}}$, can also be represented recursively as follow,

$$\phi_{\mathbf{w}}(d, \tau_{t+1}) = \phi_{\mathbf{w}}(d, \tau_t) \cdot \exp(-1/\mathbf{w}_d), \tag{16}$$

since given a word d, $\tau_{t+1} = \tau_t + 1$ holds.

3 Experiments

Now we demonstrate implementations and experiments of DBO-CPD on two real-world datasets, stock prices and movie revenues. The first case is the historical end-of-day stock prices of five information technology corporations. In the second dataset, we examine daily film revenues averaged by the number of theaters.

3.1 Datasets

In the stock price dataset, we gather data for five different companies, Apple (AAPL), Google (GOOG), IBM (IBM), Microsoft (MSFT) and Facebook (FB). These companies are selected based on the top-5 rank of market value in 2015.

We choose these technology companies because the announcement of new IT products and features and the interest of public media tend to be higher and lead to many news articles. We use the historical stock price data from Google Finance.

category	words	documents	words/doc
AAPL	15.0M	29,459	509
AAPL:N	11.0M	18,896	581
GOOG	15.0M	29,422	511
GOOG:N	8.2M	13,658	603
IBM	26.7M	45,741	583
IBM:N	3.4M	4,741	726
MSFT	20.5M	35,905	570
MSFT:N	3.5M	5,070	681
FB	18.9M	38,168	495
FB:N	4.3M	6,625	645
KNGHT	14.4M	16,874	852
INCPT	12.1M	17,155	705
AVGR	3.5M	6,476	537
FRZ	6.8M	15,021	454
INTRS	4.2M	7,846	538

Table 1: Dimensions of the datasets used in this paper, after tokenizing and filtering the news articles. "**:N**" means the articles are collected with additional "NASDAQ:" search query

The second data is a set of movie revenues averaged by the number of theaters for five months from the release date of film. We target 5 different movies, The dark knight (KNGHT), Inception (INCPT), The avengers (AVGR), Frozen (FRZ) and Interstellar (INTRS), based on the top revenue ranking in the history and recent movies. The cumulative daily revenue per

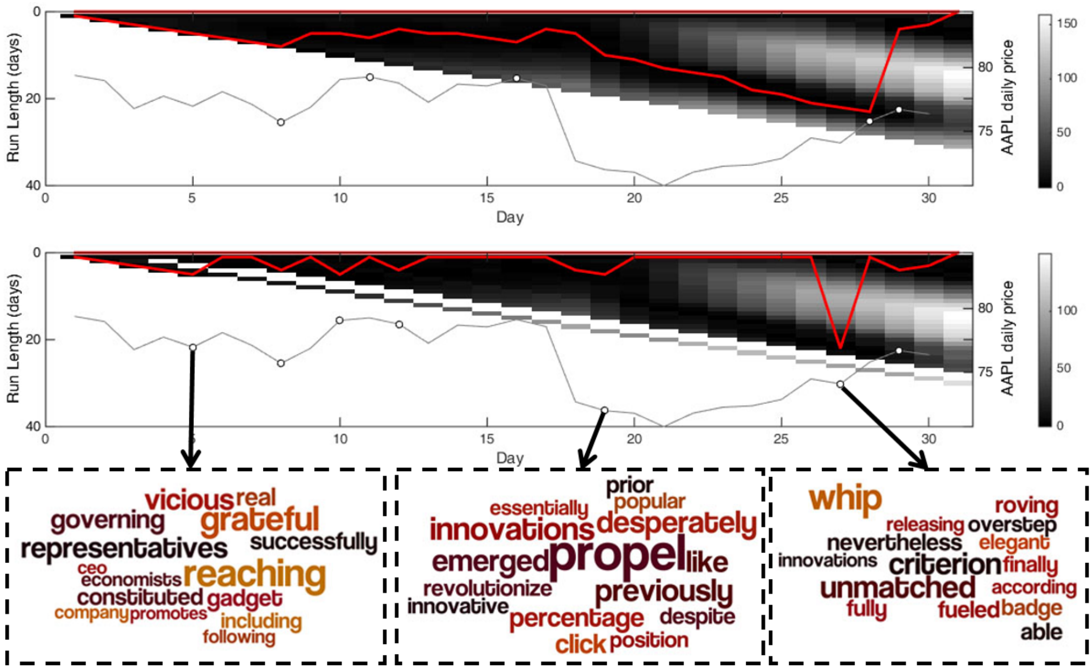


Figure 5: (a) Two plots show the results of BO-CPD (**top**) and DBO-CPD (**middle**) respectively on Apple stock prices in January 2014. The stock price is plotted in light gray, with the predictive change points drawn as small circles. The red line represents the most likely predicted run-lengths for each days. The **bottom** figures are a set of visualization of top-15 strongly weighted words which is found at selected change points in which BO-CPD is unable to predict. The size of words represents the weight of its textual features learned during the training of regression model.

theater is collected from "Box Office Mojo" (www.boxofficemojo.com).

News articles are collected from Google News and we use *Google search queries* to extract specific articles related to each dataset in a specific time period. During the online article crawling, we store not only the title of articles, HTML documents and publication date but also the number of related articles. The number of articles is used to differentiate the weight of news articles during the training of regression. In case of stock price data, we use two different queries to decrease noises. First, we search with company name such as "google". Then, we use queries specific to stock "NASDAQ:" to narrow down the content of search result relevant to stock market. In case of movie data, we search with the movie title with additional word "movie" to collect the articles only related to the target movie.

With these collected articles, we use two article extractors, *newspaper* (Ou-Yang, 2013) and *python-goose* (Grangier, 2013), to automate the text extraction of 291,057 HTML documents. After a preprocessing, we could extract successfully texts from 287,389 (98.74%) HTMLs.

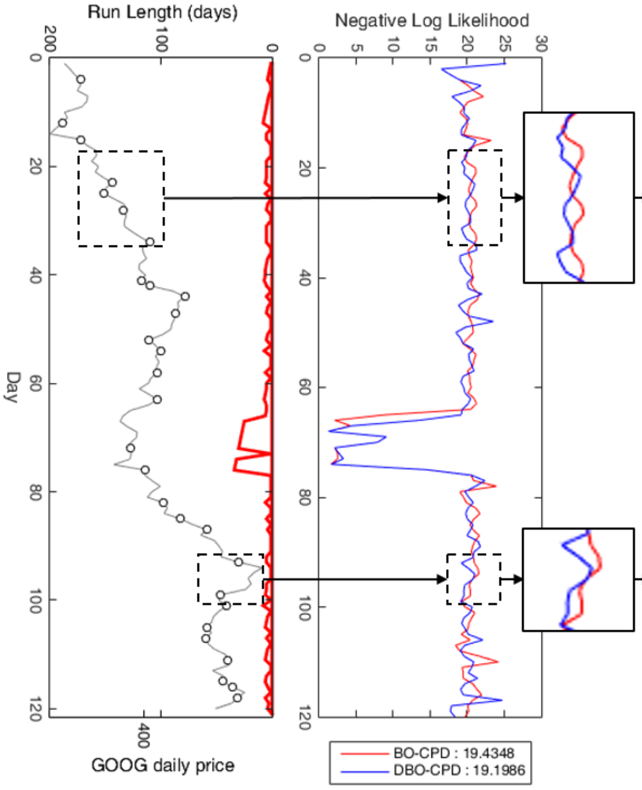
3.2 Textual Feature Representation

After the extraction of texts from HTMLs, we tokenize the texts into word tokens. We use three different tokenization methods which are downcasing the characters, punctuation removal and removing stop words of English. Table 3.1 shows the statistics on the corpora of collected news articles.

With these article corpora, we use *bag-of-words* (BoW) representation to change each words in to a vector representation where words from articles are indexed and then weighted. Using these vectors, we adopt three document representations, TF, TFIDF and LOG1P, which extends BoW representation. TF and TFIDF (Sparck Jones, 1972) calculates the importance of a word to a set of documents based on term frequency. LOG1P (Kogan et al., 2009) calculates the logarithm of the word frequencies.

3.3 Training BO-CPD

As we notice earlier, we use BO-CPD to train the regression model to learn high height for words which are more related to changes. When we



Rank	From Jan 25 to Feb 13 at 2013 from GOOG
1	The tech giant Google has been **steadily** building a portfolio of sustainability-related projects that put it squarely in the vanguard of corporate **leadership**, with a long and growing list that includes green data centers, solar power installations, open **source** alternative energy **maps** based on Google **Earth** and even amanure-to-energy project.
2	Google **announced** today that it has updated Google **Earth** for desktop with 100,000 **new tours** and a **million** photos from Panoramio, a geolocation-oriented photo share site that Google acquired in 2007.
3	Google Shopping was a **big** bet for Google because until September it was free for **retailers** to **list products** that were **searchable** in Google Shopping.

Rank	From May 14 to May 21 at 2013 from GOOG
1	Google's **director** of Google Maps, Daniel Graf, demoed the **new** version onstage and showed **new** features, **including integrated** lists from Zagat, more reviews in more places and a revamped directions and navigations experience (which now includes real-time incidence alerts and **dynamic** rerouting).
2	At Google's **developer**-focused I/O conference today, Google's Hugo Barra **announced** that Google Play has just passed 48 billion **app installs** — and had a **massive** 2.5 billion just last month.
3	Case in point: according to a **post** on the official Google Commerce **blog**, Google Wallet **support** has been baked into Gmail, so users will **soon** be able to send each other **money** by simply shooting each other emails.

Figure 6: (b) The **left** plot illustrates daily stock prices of Google at 2013 from early January to late May. The black line represents the stock price, black circles indicate the predicted change points and the red line shows the predicted run length calculated by DBO-CPD. The **middle** plot shows the negative log likelihood (NLL) of BO-CPD and DBO-CPD on same data. The overall marginal NLL of DBO-CPD (19.1964) is smaller than BO-CPD (19.3438). The two zoomed intervals are the two longest interval where negative log likelihood of DBO-CPD is smaller than BO-CPD. The **right** table shows the sentences whose predicted run length of regression model (described in Section 2.3) are the highest at the two zoomed points, which means the sentences are likely to appear near feature change point. The boldface words are the top-5 most strongly-weighted terms in the regression model.

choose the parameters for the Gaussian Process of BO-CPD, we try to find the value which make the distance of intervals between predicted change points are around 1-2 weeks. This is because we assume that the information included in the articles will make an immediate effect to the data right after it is published to the public, so the external information in texts will indicate the short-term causes for a feature change.

For the reasonable comparison of BO-CPD and DBO-CPD, we use the same parameter for Gaussian Process in both models. After several experiments we found that $a = 1$ and $b = 1$ for Gaussian Process and $\lambda_{gap} = 250$ is appropriate to train BO-CPD in stock and film datasets. We separate the training and testing examples by for the cross-validation in the ratio of $2 : 1$ for each year. Then we train each models differently by year.

3.4 Learning the strength parameter w from Regression

The weight $\mathbf{w}$ of regression model gives us an intuition how much a word implies the importance to length of the current run. With the predicted run length calculated in Section 3.3, we change the run length domain $r \in \mathbb{R}$ into $0 \leq r \leq 1$ by predicting e^{r_t} rather than r_t to solve the interpretability problem. Therefore, we can think high w_i as a powerful word which change the current run length r to 0. To maintain the scalability of $\mathbf{w}$,

	2010	2011	2012	2013	2014
AAPL BO-CPD	14.93	16.33	16.24	14.44	17.63
AAPL DBO-CPD I	**14.81**	16.22	16.20	**14.21**	17.12
AAPL DBO-CPD II	15.15	**16.20**	**16.14**	14.40	**17.11**
GOOG BO-CPD	**15.03**	15.65	15.49	19.43	19.04
GOOG DBO-CPD I	15.48	15.92	**15.21**	19.24	19.07
GOOG DBO-CPD II	15.31	**15.62**	15.36	**19.20**	**19.02**
IBM BO-CPD	17.10	17.83	17.42	16.25	16.30
IBM DBO-CPD I	17.66	**17.81**	17.40	16.20	**16.04**
IBM DBO-CPD II	**17.04**	17.82	**17.38**	**16.14**	16.39
MSFT BO-CPD	12.41	11.91	14.51	15.60	17.25
MSFT DBO-CPD I	12.33	12.60	14.48	**14.92**	**16.43**
MSFT DBO-CPD II	**12.21**	**11.79**	**14.46**	15.00	16.46
FB BO-CPD	N/A	N/A	**12.32**	13.07	16.68
FB DBO-CPD I	N/A	N/A	12.34	13.00	16.24
FB DBO-CPD II	N/A	N/A	12.43	**12.98**	**16.25**

Table 2: Negative log likelihood of five stocks (Apple, Google, IBM, Microsoft and Facebook) without and with our model per year from 2010 to 2014. DBO-CPD I represents the experiments without NASDAQ as search query and DBO-CPD II is the result of articles searched with "NASDAQ:". Facebook data is not available before the year 2012.

we normalize weight into $\mathbf{w} \subset [-1, 1]$. With the word representation calculated in Section 3.2, we train the regression model by using the number of relevant articles as the importance weight of training.

3.5 Results

We evaluate the performance of BO-CPD and DBO-CPD by comparing the negative log likelihood (NLL) (Turner et al., 2009) of two models at time t as:

$$\log p(x_{1:T}|\mathbf{w}) = \sum_{t=1}^{T} \log p(x_t|x_{1:t-1}, \mathbf{w}).$$

We calculate the marginal NLL by year and the results are represented in Table 2 and Table 3, We found that most of the DBO-CPD II shows better results than DBO-CPD I and BO-CPD in most datasets due to noise reduction of texts through additional search query "NASDAQ:". Out of 23 datasets. APPL at 2010 and FB at 2012 are the only datasets where NLL of BO-CPD is smaller than the case of DBO-CPD.

One of the advantages of using a linear model is that we can investigate what the model discovers about different terms. As shown in Figure 5, we can find negative semantic words such as *vicious*, *whip* and *desperately* and words representing the status of company like *propel*, *innovations* and *grateful* are the most strongly-weighted terms in regression model. We analyze and visualize some change points where NLL of DBO-CPD is lower than NLL of BO-CPD. The results are show in Figure 6 and three sentences are the top-3 most weighted sentences in regression model for two changes with the boldface words of top-5 strongly weighted terms like the terms *big*, *money* and *steadily*. A particularly interesting case is the terms *earth* which is found between Jan. 25 and Feb. 13 in 2013. After we investigate articles where the sentence is included, we found that Google announced a new tour guide feature in Google Earth at Jan 31 and after this announcement the stock price increases. We can also found that a word *million* is also a positive term which can predicts a new change in the near feature.

	NLL
KNGHT BO-CPD	39.76
KNGHT DBO-CPD I	**39.54**
INCPT BO-CPD	55.60
INCPT DBO-CPD I	**55.54**
AVGR BO-CPD	32.12
AVGR DBO-CPD I	**32.10**
FRZ BO-CPD	51.25
FRZ DBO-CPD I	**51.04**
INT BO-CPD	38.49
INT DBO-CPD I	**38.31**

Table 3: Negative log likelihood (NLL) of five movies (The dark knight, Inception, Avengers, Frozen and Interstellar) without and with our model for 1 year from the release date of movie.

4 Conclusion

In this paper, we propose a novel generative model for online inference to find change points from non-stationary time-series data. Unlike previous approaches, our model can incorporate external information in texts which includes the causes of signal changes. The main contribution of our approach is to combine the generative models of documents and regression model that learned the weights of words. Thus, our model makes it possible to accurately infer the conditional prior on the change point and automatically explains the reasons of a change by connecting the numerical sequence of data and textual features of news articles.

References

Ryan Prescott Adams and David JC MacKay. 2007. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*.

Daniel Barry and John A Hartigan. 1992. Product partition models for change point problems. *The Annals of Statistics*, pages 260–279.

Daniel Barry and John A Hartigan. 1993. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

Jie Chen and AK Gupta. 1997. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92(438):739–747.

Siddhartha Chib. 1998. Estimation and comparison of multiple change-point models. *Journal of econometrics*, 86(2):221–241.

Pao-Shin Chu and Xin Zhao. 2004. Bayesian change-point analysis of tropical cyclone activity: The central north pacific case. *Journal of Climate*, 17(24):4893–4901.

Catherine Forbes, Merran Evans, Nicholas Hastings, and Brian Peacock. 2011. *Statistical distributions*. John Wiley & Sons.

Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Wai Lam. 2002. News sensitive stock trend prediction. In *Advances in knowledge discovery and data mining*, pages 481–493. Springer.

Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Wai Lam. 2003. Stock prediction: Integrating text mining approach using real-time news. In *IEEE International Conference on Computational Intelligence for Financial Engineering*, pages 395–402.

Gyozo Gidófalvi and Charles Elkan. 2001. Using news articles to predict stock price movements. *Department of Computer Science and Engineering, University of California, San Diego*.

Xavier Grangier. 2013. Python-goose - article extractor.

Daniel Gruhl, Ramanathan Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. 2005. The predictive power of online chatter. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 78–87.

William Gu, Jaesik Choi, Ming Gu, Horst Simon, and Kesheng Wu. 2013. Fast change point detection for electricity market analysis. In *IEEE International Conference on Big Data*, pages 50–57.

Der-Ann Hsu. 1977. Tests for variance shift at an unknown time point. *Applied Statistics*, pages 279–284.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialogue acts in one-on-one live chats. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 862–871.

Shimon Kogan, Dimitry Levin, Bryan R Routledge, Jacob S Sagi, and Noah A Smith. 2009. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association for Computational Linguistics*, pages 272–280.

Gary Koop and Simon M. Potter. 2007. Estimation and forecasting in models with multiple breaks. *The Review of Economic Studies*, 74(3):pp. 763–789.

Gilad Mishne and Natalie S Glance. 2006. Predicting movie sales from blogger sentiment. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 155–158.

Joel Nothman, Matthew Honnibal, Ben Hachey, and James R Curran. 2012. Event linking: Grounding event reference in a news archive. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 228–232.

Michael Osborne. 2010. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. Ph.D. thesis, Oxford University New College.

Lucas Ou-Yang. 2013. newspaper - news, full-text, and article metadata extraction.

Baolin Peng, Jing Li, Junwen Chen, Xu Han, Ruifeng Xu, and Kam-Fai Wong. 2015. Trending sentiment-topic detection on twitter. In *Computational Linguistics and Intelligent Text Processing*, pages 66–77. Springer.

Robert Schumaker and Hsinchun Chen. 2006. Textual analysis of stock market prediction using financial news articles. *AMCIS 2006 Proceedings*, page 185.

Robert P Schumaker and Hsinchun Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12.

Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. 2013. Exploiting topic based twitter sentiment for stock prediction.

AFM Smith. 1975. A bayesian approach to inference about a change-point in a sequence of random variables. *Biometrika*, 62(2):407–416.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Connie St Louis, Gozde Zorlu, et al. 2012. Can twitter predict disease outbreaks? *BMJ*, 344.

DA Stephens. 1994. Bayesian retrospective multiple-changepoint identification. *Applied Statistics*, pages 159–178.

Mark Steyvers and Scott Brown. 2005. Prediction and change detection. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1281–1288.

Ryan Turner, Yunus Saatci, and Carl Edward Rasmussen. 2009. Adaptive sequential bayesian change point detection.

Udo von Toussaint. 2011. Bayesian inference in physics. *Reviews of Modern Physics*, 83(3):943.

Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120. Association for Computational Linguistics.

Xiang Xuan and Kevin Murphy. 2007. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 1055–1062.

Xin Zhao and Pao-Shin Chu. 2010. Bayesian changepoint analysis for extreme events (typhoons, heavy rainfall, and heat waves): An rjmcmc approach. *Journal of Climate*, 23(5):1034–1046.