Prediction of stock switching points by financial news

by

Saeede Sadat Asadi Kakhki

Bachelor of Industrial Engineering, Sharif University of technology, 2015

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Masters of Applied Science

in the Program of

Mechanical and Industrial Engineering

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Prediction of stock switching points by financial news

Masters of Applied Science 2018

Saeede Sadat Asadi Kakhki

Mechanical and Industrial Engineering

Ryerson University

Abstract

The purpose of this study is to detect stock switching points from historical stock data and analyze corresponding financial news to predict upcoming stock switching points. Various change point detection methods have been investigated in the literature, such as online bayesian change point detection technique. Prediction of stock changing points using financial news has been implemented by different types of text mining techniques. In this study, online bayesian change point detection is implemented to detect stock switching points from historical stock data. Relevant news to detected change points are retrieved in the past and Latent Dirichlet Allocation technique is used to learn the hidden structures in the news data. Unseen news are then transferred to the trained topic representation. Similarity of relevant news and unseen news are used for prediction of future stock change points. Results show that stock switching points can be detected by historical stock data with better performance comparing to random guessing. It is possible to predict stock switching points by only fraction of financial news and with good result in terms of common performance metrics. According to this research, traders can take advantage of financial news to enhance prediction of future stock switching points.

Acknowledgements

Prima facea, I am grateful to the God for the good health and wellbeing that were necessary to complete this research.

I wish to express my sincere thanks to Prof. Ayse Bener, my supervisor, for providing me with all the necessary facilities for the research. I am extremely thankful and indebted to her for sharing expertise, and sincere and valuable guidance and encouragement extended to me.

I am also grateful to all Data Science Lab members, mainly Dr. Can Kavaklioglu for the continuous encouragement and help.

I take this opportunity to express gratitude to all of the Department faculty members for their help and support. I also thank my parents for the unceasing encouragement, support and attention.

Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

Introduction

Price movements in the stock exchange are important as they have direct effects on investment gains and losses [7]. Traders take positions in the market according to changes in the price signals. As it leads to early action among stock market traders and economists, detecting change points in the stock market data is crucial. Considering the efficient-market hypothesis, all types of information about a company affect its stock price [67]. In order to accurately predict a stock price, a lot of information should be processed in a limited amount of time. One of the richest sources of information is textual news data [40, 69, 81]. Although there is some debate in the literature on the value of news information, it is believed that studies which show that financial news is a good indicator of price changes in the market [4, 21, 52, 88] outweigh studies that argue the alternative point of view.

In order to detect the changes in stock price signals, it is crucial to have a clear definition of stock change points and a method to detect them. Segmenting time series is central to a wide range of applications. Many real-world data streams consist of consecutive partitions separated by an abrupt change [27, 51]. In such situations, the underlying model produces data switches multiple times among partitions. This particular issue arises in contexts ranging from speech recognition to medical monitoring [16, 24, 27, 51, 74]. In statistics, a conventional way to detect changes in point is to fit probability distributions over the partitions using past and present data to check whether there is a significant difference between these two intervals [16, 44]. In this study, OBCD is used, which defines an auxiliary variable "run length" to detect stock switching points by comparing distributions of two consecutive partitions. Run length, defined as the time elapsed since the last change point, helps to detect abrupt changes in stock prices since it diminishes to zero when a change point occurs.

News media produces a huge number of news on a daily basis. As [21] shows, there is a significant relationship between stock price movements and textual news. Therefore, using an appropriate source of news and extracting only relevant information may help to overcome the complexity of text mining for switching points detection. This study aims to identify the relevant news items that are related to a stock price switching point. In particular, this study uses one of the richest financial news sources, the Dow Jones dataset, which can be used to find news items related to a specific stock or company. Using a subset of the news items in the dataset, news relevant to switching points are detected in the training

set. In the testing phase, document similarity is measured between test news and relevant news.

The LDA model captures the underlying hidden structure of documents in natural language processing [19, 25, 48, 49, 77]. It helps to decrease the number of features to obtain a set of meaningful features, and as a result, decrease the overall complexity of text mining process. Application of the LDA for stock price movement is used in the literature [64, 89]. In this research, the performance of this method is tested in a new setting. First, stock switching points are detected from historical stock data by the OBCD technique. Second, only the news related to previously detected switching points are captured. Third, the LDA model with variational Bayesian inference is trained to the captured news items to detect their topics and words distributions. Topics distribution shows the distribution of topics for each news and words distribution shows the distribution of words for each topic. Learning these two distributions helps to represent news by the distribution of their topics rather than distribution of their words. Fourth, the distribution of words within detected topics can help to transform test news into learned topics representation. Finally, the similarity value between test and relevant news is measured by their topic representation to test the similarity value's performance as a prediction indicator of stock switching points.

Measuring the similarity of news topic representation by combining various types of similarity metrics such as cosine similarity, Kullback Leibler divergence, Jenson-Shannon divergence, and Euclidian distance helps to cover all aspects of their equality. Furthermore, evaluation of this method is completed in two phases: the performance of the change point detection technique by OBCD, and the evaluation of stock change points prediction by financial news. To have an exact evaluation, different metrics are measured to compare detected and predicted stock switching points. Detection of stock switching points by using historical stock data provides better performance compared to random guessing in terms of different measured metrics. It can be concluded by this result that it is possible to predict stock switching points by analyzing only a fraction of financial news with strong results in the defined performance metrics. Traders can take advantage of financial news to enhance their predictions of future stock switching points in real-word applications.

The rest of this dissertation is organised as follows. The next section will include a review of related works on change point detection techniques, predictions of stock exchange behaviour using different text mining approaches and the LDA method. In section 3, the methodology is discussed in detail, including algorithms implemented, dataset and experiment design. In section 4, results and threats to validity of the results are reported. Section 5 will include the results, conclusion, and suggestions for future research.

Chapter 2

Background

According to the purpose of this study as detection of stock switching points from historical stock data as well as analyzing related financial news to predict future stock switching points, it is of interest to provide some literature review about change point detection techniques and text mining approaches to predict stock switching points.

2.1 Change point detection

Change point detection techniques have been used in other studies in the literature for different fields, ranging from medical monitoring to speech recognition [16, 24, 27, 51, 74]. These techniques aim to detect one or more abrupt changes in a signal of data that segments the data into some partitions [87]. Data within each partition are homogeneous and are non-homogeneous among partitions.Figure 2.1 illustrates an example of change point detection problem which is of interest to detect timestamps t_1 and t_2. Methods available for detection of such points might vary by four different aspects.

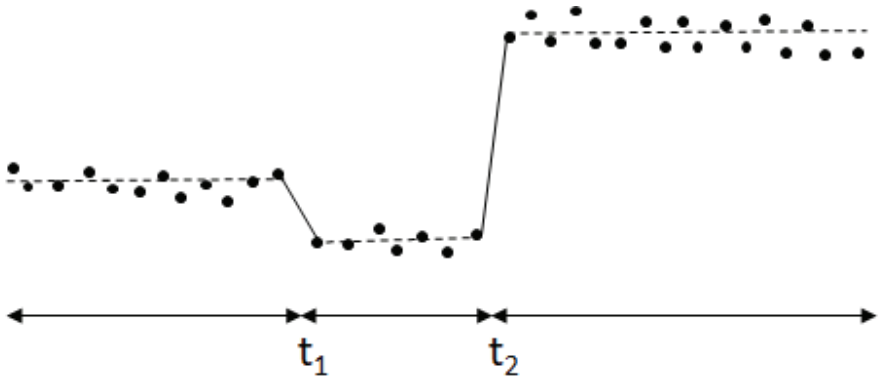

Figure 2.1: Change point detection example

First, the cost function measures the goodness of fit for each partition. Different types of cost function might be determined by assuming a piecewise constant distribution [1, 26, 27, 28, 51, 56, 66], liner model [8, 9, 10, 11, 12], kernel change point detection [5, 31, 46, 47] or mahalanobis-type metric [58, 90].

Piecewise constant distribution might detect change points through maximum likelihood estimation. It chooses a distribution for the data in each partition according to some prior knowledge about the data. More often, a Gaussian distribution [6, 57, 60] or other exponential family distributions [33, 35, 65] is assumed to be the underlying distribution for the data. Exponential family distributions is a set of distributions which can be represented by same form and have a large number of properties that make them extremely useful for statistical analysis.

Linear models are useful when there exists a linear relationship between dependent and independent variables and the coefficients among these two sets of variables change abruptly at change points, while Kernel change point detection relies on mapping the data to high dimensions by a kernel function. Mapping to high dimensions will transform the data to a piecewise constant signal. Similarly, the mahalanobis-type metric relies on mahalanobis distance which measures the distance between a data point and the mean of a distribution. Mahalanobis-type metric detect change points by going through each data point and measuring the distance of the data point and the mean of the data points in the same partition.

Another cause of differences among change point detection techniques is the assumption sometimes made regarding the number of change points. They might be different in the sense that the number of change points can be initialized as the input of the method [27, 28, 42, 54, 56], or that the method is free of this setting, which can result in a number of various detected change points [17, 22, 91]. When the number of change points is fixed, the purpose is to minimize the cost function by setting the number of detected change points. In contrast, when it is not fixed, it minimizes the cost function as well as the complexity of the segmentation with a proper penalty.

Furthermore, underlying search methods used to optimize the cost function can be different in change point detection techniques. Choosing an appropriate search method requires the number of change points whether it is set to be a fixed number or if it is unknown. When the number of change points is initialized as the input of the method, an optimal detection technique [13, 42, 53] or its fast-approximate alternatives, such as window sliding [16, 28, 55], binary segmentation [27, 56] or bottom-up segmentation [28, 36, 54] can be used. When the number of change points is unknown in the process of change point detection, search methods like the Pelt (Pruned Exact Linear Time) algorithm can be applied to use a linear penalty for controlling the trade-off between complexity and goodness-of-fit.

The last but not least aspect that can be different among change point detection techniques is the way that they receive and analyze the data for processing. Offline or retrospective algorithms try to detect change points when a particular realization of the signal is observed [16, 35, 46, 47, 56]. Common methods for offline learning are maximum likelihood estimation, regression and kernel methods. On the other hand, online algorithms processes data points one-by-one in a serial manner [31, 59, 63]. As opposed to offline algorithms, online algorithms do not need to access the entire input and update its parameters after learning from each training instance. Choosing the way the data is received and analyzed depends on the underlying question.

While earlier approaches have considered the problem of change point detection through maximum-likelihood estimation [43] or novelty detection [68], more recent approaches further expand the applicability of change point detection problems from a Bayesian point of view.

Bayesian methods play an important role in the change point detection literature and have been used for various applications, such as in speech recognition [78], brain imaging [2], video segmentation [76] and bioinformatics [14, 32, 33]. Bayesian inference methods are statistical inference techniques which use prior beliefs about change point location distribution and update unknown parameters as more evidence or information becomes available to deduce properties of an underlying probability distribution. In contrast, frequentist inference techniques achieve the same purpose by considering repeated sampling of a population distribution to produce datasets similar to the original datasets.

Even though earlier Bayesian approaches consider the problem of detecting change points as a retrospective segmentation problem in an offline fashion [15, 29], the pioneer work of [2] has considered the same problem in an online fashion by estimating the posterior distribution over an auxiliary variable run-length, with r_t defined as the time elapsed since the last change point. As in this paper, some researchers have also considered expanding the methodology described in [2], in various ways, such as by applying it to human-machine interaction systems [59] and geoacoustic inversions [85]. The underlying Bayesian nature of this method makes it easy to be understood and be extended for any setting. Table 2.1 summarizes the characteristics of the change point detection literature as explained.

	On\Offline	# of change points	Cost function	Search method
Ko, Chong, and Ghosh [57]	Offline	Not fixed	Piecewise constant	Bayesian
Frick, Munk, and Sieling. [35]	Offline	-	Piecewise constant	Binary segmentation
Arlot, Celisse, and Harchaoui. [5]	Offline	-	Kernel	Optimal
Harchaoui and Cappé. [46]	Offline	Fixed	Kernel	Optimal
Desobry, Davy, and Doncarli. [31]	Online	Not fixed	Kernel	Window sliding
Birgé and Massart. [17]	Offline	Not fixed	Linear model	Optimal
S. Chib. [29]	Offline	Not fixed	Piecewise constant	Bayesian
Bai and Perron. [13]	Offline	-	Linear model	Optimal
J. Bai. [8]	Offline	-	Linear model	Optimal
Basseville and Nikiforov. [16]	Offline	-	Linear model	Optimal
L.R. Rabiner. [78]	Offline	Not fixed	Linear model	Bayesian
Y.-C. Yao. [91]	Offline	Not fixed	Linear model	Pelt
Prescott Adams and MacKay. [2]	Online	Not fixed	Piecewise constant	Bayesian

Table 2.1: Summary of change point detection literature

2.2 Text mining for change point prediction

The efficient market hypothesis states that stock prices are always influenced by all available information. As a result, correlating textual news data with stock market data is a popular method to study price signal behaviours [37, 62]. Text mining is the process of extracting information from the unstructured

text documents. It is used in various domains such as natural language processing and information retrieval. Figure 2.2 gives a general overview of text mining process.

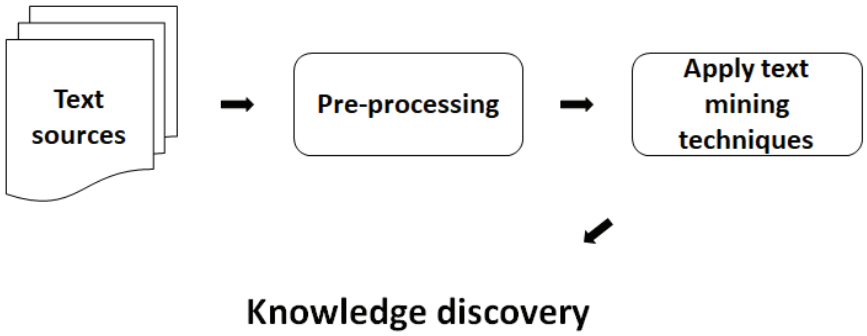

Figure 2.2: Text mining process

There are four main categories of text mining techniques which has been discussed for stock switching points prediction in the literature such as: text classification, semantic and sentiment analysis, text summarization and text similarity techniques.

Text mining techniques, such as text classification, implement machine learning algorithms on textual data using supervised learning techniques. Supervised learning analyze labeled dataset to predict class labels for unseen instances. These methods are applied for the purpose of stock switching point prediction by [40, 82, 86]. Some researchers have implemented supervised machine learning techniques such as the Naïve Bayes classifier, the Support Vector Machine, K-nearest neighbor and the Genetic Algorithm to classify labelled documents for stock change point prediction. There is no optimal classification technique to predict stock change points. The performance highly depends on the related stock exchange, news sources and the way classes are defined.

Another technique which has frequently been under investigation is semantic and sentiment analysis [62, 73]. Semantic and sentiment analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text to determine whether the writer attitude towards a particular topic is positive, negative, or neutral. Researchers have used this technique to find the subjective information behind contents to illustrate future stock price behaviour. For example in [20], two mood tracking tools, OpinionFinder and GPMOS, are used to track the mood of a Twitter user in terms of polarity (positive-negative) and degree of certain attributes, including calmness, alertness, confidence, vitality, kindness, and happiness to predict upcoming stock change points.

On the other hand, in a text corpora that contains numerous documents, it is essential to develop another text mining technique to summarize all the content. Text summarization techniques, like topic modelling, try to discover abstracts or hidden structures within the bodies of texts. These hidden structures are called topics, unobservable groups that gather specific words to create documents related to different subjects. This technique results to represent documents by their contained topics which

are quantitatively less than the total number of words in the document. A document might talk about multiple topics and words might appear in different topics. The great advantage of text summarization techniques essentially topic modeling techniques is their freedom of labeling the dataset.

The last main category of text mining techniques that has been used in the literature for the purpose of studying stock switching points is text similarity, which aims to estimate the degree of similarity between texts [3]. Evaluating the similarity among corpora, documents, paragraphs, sentences or words is important in different applications ranging from information retrieval to automatic essay scoring. There are two main types of text similarity algorithms: string-based, corpus-based and knowledge-based. String-based algorithms split into two main categories: character-based and term-based with the purpose of comparing similarity or dissimilarity among text strings [41]. On the other hand, corpus-based algorithm captures the similarity among words by analyzing large corpora. Further more, knowledge-based algorithms find the same relation by using information derived from semantic networks.

The above-mentioned techniques can be combined together to create a hybrid method. Hybrid methods use multiple techniques to improve the performance. For example, two different text mining techniques, topic modelling and sentiment analysis, are used together in the literature to predict stock market prices [71]. It is proved in [71] that combination of these two techniques helps to improve the performance of stock switching points prediction significantly comparing to using any of these two techniques individually. In this study, the topic modelling and text similarity techniques are combined to predict stock switching points by using financial news.

While there are different topic modelling techniques like latent semantic analysis [75], probabilistic latent semantic analysis [50] and latent dirichlet allocation, LDA generally works best due to its generative nature. This property leads LDA to not only have the concern of detecting the hidden structures of documents, but also understand the process of new documents generation.

Latent semantic analysis assumes that words that are close in meaning will occur in similar pieces of documents. It implements mathematical technique called singular value decomposition to capture most of the variance in a corpus on a lower dimension [75]. Singular value decomposition is a factorization of a matrix into three different matrix. LDA and probabilistic latent semantic analysis both assume topics as distribution over words and are based on mixture decomposition. However LDA takes into account an extra assumption: it considers documents as mixture of topics with dirichlet prior [19, 25, 48, 49, 77]. LDA can be assumed as a generalization of the probabilistic latent semantic analysis model under a uniform Dirichlet prior distribution. LDA assumes that documents cover only a small set of topics and that topics use only a small set of words. Table 2.2 summarized the characteristics and limitations of the above mentioned topic modeling techniques.

The Bayesian nature of LDA leads to have three different distributions to be estimated: the likelihood, prior and posterior distributions. To solve the underlying posterior distribution of hidden variables given a textual data and its contained words in LDA method, there are two main approaches: sampling methods [25, 83] and optimization methods [19, 48, 49].

In sampling methods, like Markov chain Monte Carlo (MCMC), there is no need to explicitly set the parameters and thus it is simpler in terms of implementation. In this method, the posterior distribution is estimated by repeated sampling from the probability distribution. On the other hand, optimization

Techniques	**Attributes and Constraints**
Latent semantic analysis	Attributes: • Reduce dimensionality by using singular value decomposition • Retrieve synonyms of words Constraints: • Incapable to detect the number of topics • Difficult to interpret three resulted matrices of decomposition
Probabilistic latent semantic analysis	Attributes: • Each word is generated from a single topic • Different words may be generated from different topics Constraints: • Incapable to generate new documents • Incapable to detect distribution of topics in each document
Latent dirichlet allocation	Attributes: • Provide generative model for words and topics in documents • Applicable for long-length documents Constraints: • Incapable to detect relations among topics

Table 2.2: Attributes and Constraints of topic modeling techniques [61]

methods like variational Bayesian methods are guaranteed to converge to the posterior probability by detecting a family of distributions with simpler form called a variational distribution.

As it is mentioned in [48], MCMC sampling methods like the Gibbs sampling method have higher time complexity. Gibbs sampling method tries to generate independent samples from the posterior to update document-topic and topic-word distributions. It results to not be efficient for applying on large scale datasets [18, 23]. Since sampling and optimization methods have comparable results in terms of performance, it is beneficial to use optimization methods for large scale dataset. In this study, variational Bayesian posterior estimation is applied to capture the hidden structures of the financial text news by retrieving the variational distribution and its related parameters.

Chapter 3

Methodology

3.1 Online Bayesian Change Point Detection

Defining change points as the abrupt changes in the generative parameters of a sequence of data, Online Bayesian Change point Detection (OBCD) aims to detect such points by estimating the posterior distribution over an auxiliary variable *run-length*, defined as the time elapsed since the last change point [2].

If the sequential data is denoted by x_1, x_2, x_3, ..., x_T, the underlying change points divide the observations into partitions represented by ρ. Data within each partition is driven from a probability distribution as $P(x_t|\eta_\rho)$. Between time i and j, adjacent group of samples are denoted as $x_{i:j}$. The length of the partitions are represented by g with a prior probability as $P_{gap}(g)$. The defined variable run length at each time is denoted by r_t. Run length diminishes to zero when there is a change point detected and incrementally increases until the second change point is encountered. The set of observations in the recent partition corresponding to time t is notated by $x_t^{(r)}$. The notations can be better understood by Figure 3.1.

The assumption made in OBCD is that the data in each of the segments are independent and identically distributed. Additionally, the set of parameters η_ρ, $\rho = 1, 2, ...$ are independent and identically distributed.

The posterior probability over run length can indicate the change point positions. According to the definition of r_t as the time since last change point, detecting timestamps with run length equal to zero is our targeted change point. To achieve an online procedure, the run length posterior probability should be estimated given the observations from the start, prior to the designated timestamp.

$$P(r_t|x_{1:t}) = \frac{P(r_t, x_{1:t})}{P(x_{1:t})} \tag{3.1}$$

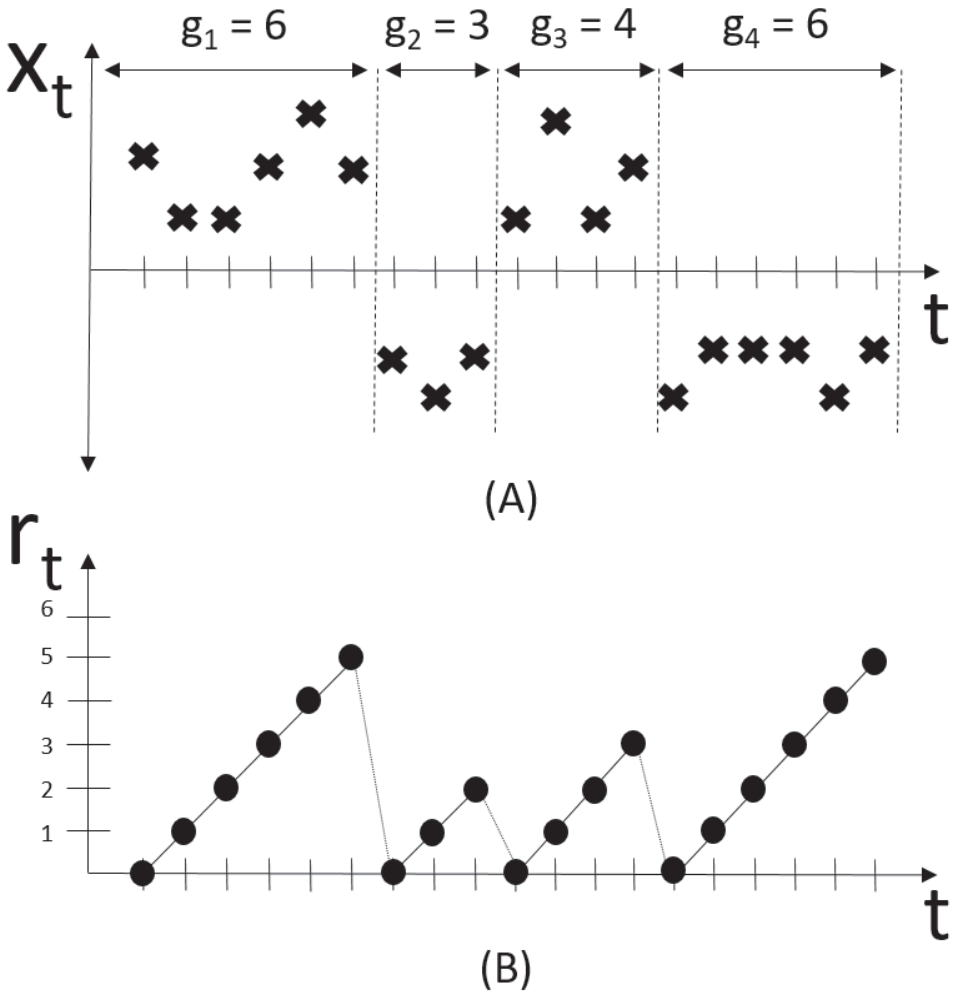

Figure 3.1: This figure illustrates how a change point model is expressed in terms of run lengths. (A) shows hypothetical univariate data divided by change points on the mean into four segments of lengths $g_1 = 6$, $g_2 = 3$, $g_3 = 4$ and $g_4 = 6$. (B) shows the run length r_t as a function of time. r_t drops to zero when a change point occurs [2].

The numerator of equation 3.1 can be estimated recursively as follows:

$$\begin{aligned} P(r_t, x_{1:t}) &= \sum_{r_{t-1}} P(r_t, r_{t-1}, x_{1:t}) = \sum_{r_{t-1}} P(r_t, x_t | r_{t-1}, x_{1:t-1}) P(r_{t-1}, x_{1:t-1}) \\ &= \sum_{r_{t-1}} P(r_t | r_{t-1}) P(x_t | r_{t-1}, x_t^{(r)}) P(r_{t-1}, x_{1:t-1}) \end{aligned} \tag{3.2}$$

According to this equation, the joint probability can be estimated recursively by knowing two statements: the prior over the current run length given the previous run length and the predictive distribution over the current datum, given the data since the recent change point. In the following estimation, these two statements are explained.

3.1.1 Run length prior

According to the definition of run length, run length incrementally increases as time passes or it might diminish to zero when a change point occurs. As a result, $P(r_t|r_{t-1})$ is not zero only if $r_t = r_{t-1} + 1$ or $r_t = 0$. On the other hand, the probability of an increase in run length given the exact previous run length is completely in line with the concept of *Hazard function*. Hazard function points to conditional density, given that the event in question has not yet occurred prior to time t [30]. Assume T as the waiting time until the occurrence of an event, *Survival function* is the probability of not having the event by time t: $S(t) = P(T > t)$. In other words, the survival function is the complement of the *Cumulative distribution function*. As a result, hazard function can be expressed as:

$$H(t) = \frac{f(t)}{S(t)} \tag{3.3}$$

Using the concept of the hazard function helps summarize the prior over the current length given the previous run length as follows:

$$P(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}+1) & r_t = 0 \\ 1 - H(r_{t-1}+1) & r_t = r_{t-1} + 1 \\ 0 & otherwise \end{cases} \tag{3.4}$$

On the other hand, it can be assumed that the length of intervals between change points (g) follows a discrete exponential (geometric) distribution. It is a valid assumption for two reasons. First, the length of interval is a discrete variable due to the nation of run length. Second, geometric probability distribution relies on a number of Bernoulli trials to have one success (or in this case, to have a change point). The waiting time until an occurrence of an event in an exponential and geometric distribution does not depend on the time that has already elapsed. This key property is called *memoryless* and can be proved as follows [34]:

$$P(T > s + t | T > s) = \frac{P(T > s + t \cap T > s)}{P(T > s)} = \frac{P(T > s + t)}{P(T > s)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} = e^{-\lambda t} = P(T > t) \tag{3.5}$$

Assume that the geometric distribution for the length of intervals between change points and its memory-less property creates a hazard function constant of H(t) = $1/\lambda$. As a result, the change point prior can be simplified as:

$$P(r_t|r_{t-1}) = \begin{cases} \frac{1}{\lambda} & r_t = 0 \\ 1 - \frac{1}{\lambda} & r_t = r_{t-1} + 1 \\ 0 & otherwise \end{cases} \tag{3.6}$$

3.1.2 Predictive Probability

The second statement of the equation 3.2 aims to measure the predictive distribution over the current data point by knowing the observations since the last change point. This part can be estimated by assuming one of the exponential family distributions for the observations. Exponential family distributions include many common distributions such as normal, exponential, Bernoulli, gamma, categorical, geometric and Poisson distributions. Two main properties of this class of distributions are [72]:

- **Sufficient statistics**: Exponential family distributions can summarize arbitrary amounts of independent and identically distributed data using a fixed number of values, which are named sufficient statistics. These statistics can be updated incrementally as data arrives.
- **Conjugate prior**: Exponential family distributions have conjugate priors. In Bayesian statistics, if a likelihood function have a conjugate prior, the posterior and prior are from same probability distribution family. It makes the inference easier since posterior at each iteration act as the prior for the next iteration.

$$P(\theta|X,\alpha) = \frac{P(X|\theta)P(\theta|\alpha)}{P(X|\alpha)} \quad : \quad Posterior = \frac{Likelihood * Prior}{Marginal \quad likelihood} \tag{3.7}$$

According to these important properties, exponential family likelihoods can be expressed in the same form:

$$P(x|\theta) = h(x)exp(\eta(\theta).T(x) - A(\theta)) \tag{3.8}$$

Where $T(x)$, $h(x)$, $\eta(\theta)$, and $A(\theta)$ are known functions for each probability distribution family and θ is the parameter of the family. The parameter of the family can be summarized and updated by sufficient statistics. Assume that the time series data follows a normal distribution, a member of exponential family distributions. It aims to estimate its parameters, the mean and variance, using Bayesian inference and conjugate priors. Depending on the question studied, any of these parameters might be unknown. It leads to have three different cases [70]:

- **Unknown mean - Unknown variance:** The conjugate prior is in the form of a **Normal-inverse-Gamma** distribution with parameters μ_0 , α_0 , β_0 and κ_0. It can be proved that the posterior predictive for a new observation follows a **T**-distribution: $t_{2\alpha_n}(x|\mu_n, \frac{\beta_n(\kappa_n+1)}{\alpha_n\kappa_n})$. Accordingly, parameters can be updated as follows [70]:

$$\alpha_{n+1} = \alpha_n + 1/2 \tag{3.9}$$

$$\kappa_{n+1} = \kappa_n + 1 \tag{3.10}$$

$$\beta_{n+1} = \beta_n + \frac{\kappa_n(x-\mu_n)^2}{2(\kappa_n+1)} \tag{3.11}$$

$$\mu_{n+1} = \frac{\kappa_n\mu_n + x}{\kappa_n + 1} \tag{3.12}$$

Figure 3.2 illustrates this case with a set of artificial data and its predefined change points.

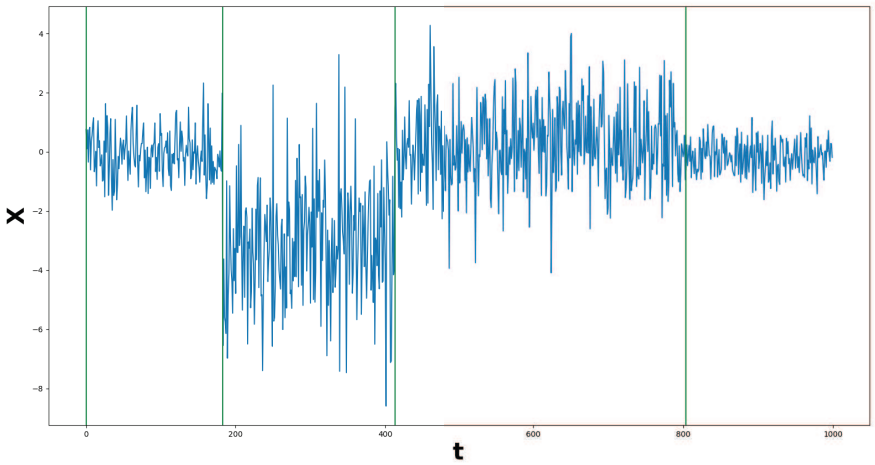

Figure 3.2: Artificial data with change in mean and/or variance. Vertical green lines point to detected change points by OBCD.

- **Known mean - Unknown variance:** The conjugate prior is in the form of a **Gamma** distribution with parameters α_0 and β_0 . It can be proved that the posterior predictive for a new observation follows a **T**-distribution: $t_{2\alpha_n}(x|\mu, \frac{\beta_n}{\alpha_n})$. Accordingly, parameters can be updated as follows :

$$\alpha_{n+1} = \alpha_n + 1/2 \tag{3.13}$$

$$\beta_{n+1} = \beta_n + \frac{(x-\mu)^2}{2} \tag{3.14}$$

Figure 3.3 illustrates this case with a set of artificial data and its predefined change points.

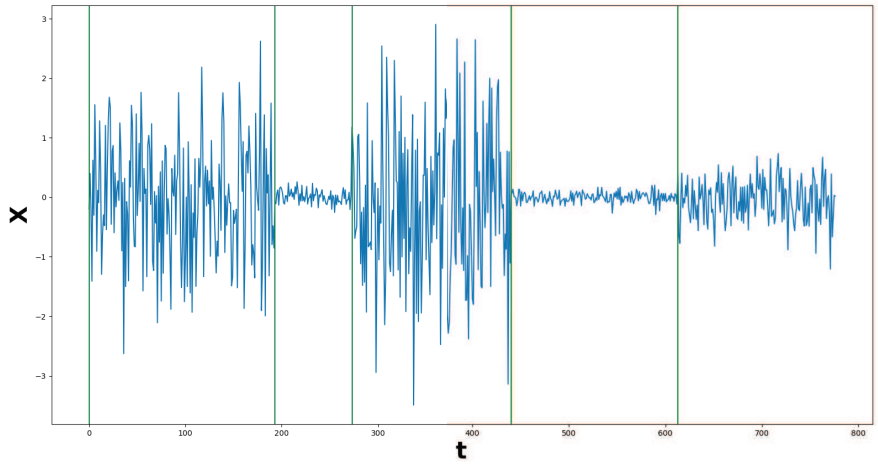

Figure 3.3: Normal distribution with change in variance and constant mean at 0. Vertical green lines point to detected change points by OBCD.

- **Unknown mean - Known variance:** The conjugate prior is in the form of a **Normal** distribution with parameters μ_0 and ${\sigma_0}^2$. It can be proved that the posterior predictive for a new observation follows a **Normal** distribution: $N(x|\mu_n, {\sigma_n}^2 + \sigma^2)$. Accordingly, parameters can be updated as follows:

$$ {\sigma_{n+1}}^2 = \frac{1}{\frac{1}{\sigma^2} + \frac{1}{{\sigma_n}^2}} \tag{3.15} $$

$$ \mu_{n+1} = \frac{\sigma^2 \mu_n + {\sigma_n}^2 x}{\sigma^2 + {\sigma_n}^2} \tag{3.16} $$

Figure 3.4 illustrates this case with a set of artificial data and its predefined change points.

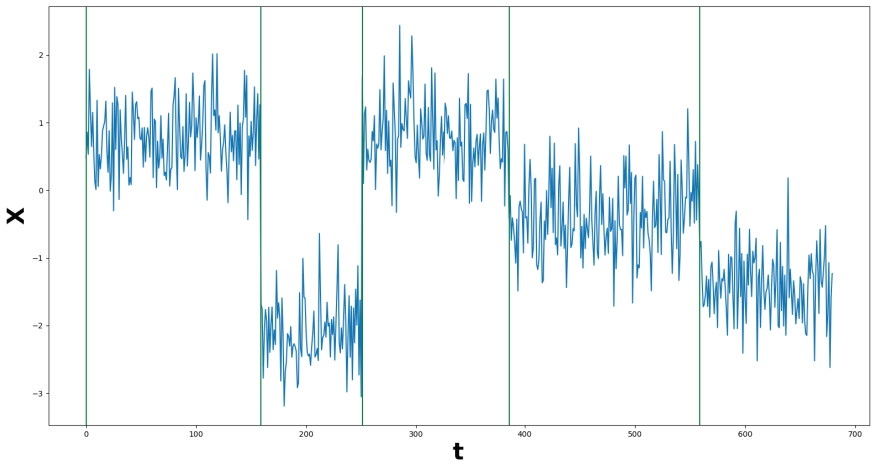

Figure 3.4: Change in mean with constant variance at 0.5. Vertical green lines point to detected change points by OBCD.

Combining the previous two sections can help provide the estimate equation 3.1. Algorithm 1 summarizes the steps toward developing an OBCD method.

Algorithm 1 Online Bayesian Change Point Detection [2]

1. Initialize

$P(r_0 = 0)$=1

$\nu_1^{(0)} = \nu_{prior}$

$\chi_1^{(0)} = \chi_{prior}$

2. Observe New Datum x_t

3. Evaluate Predictive Probability

$\pi_t^{(r)} = P(x_t|\nu_t^{(r)}, \chi_t^{(r)})$

4. Calculate Growth Probabilities

$P(r_t = r_{t-1} + 1, \boldsymbol{x}_{1:t}) = P(r_{t-1}, \boldsymbol{x}_{1:t-1})\pi_t^r(1 - H(r_{t-1}))$

5. Calculate change point Probabilities

$P(r_t = 0, \boldsymbol{x}_{1:t}) = \sum_{r_{t-1}} P(r_{t-1}, \boldsymbol{x}_{1:t})\pi_t^r H(r_{t-1})$

6. Calculate Evidence

$P(\boldsymbol{x}_{1:t}) = \sum_{r_t} P(r_t, \boldsymbol{x}_{1:t})$

7. Determine Run Length Distribution

$P(r_t|\boldsymbol{x}_{1:t}) = P(r_t, \boldsymbol{x}_{1:t})/P(\boldsymbol{x}_{1:t})$

8. Update Sufficient Statistics

$\nu_{t+1}^{(0)} = \nu_{prior}$

$\chi_{t+1}^{(0)} = \chi_{prior}$

$\nu_{t+1}^{(r+1)} = \nu_t^{(r)} + 1$

$\chi_{t+1}^{(r+1)} = \chi_t^{(r)} + \mathbf{u}(x_t)$

3.2 Latent Dirichlet Allocation

Latent Dirichlet allocation is a generative statistical topic modelling technique [19]. Topic modelling methods aim to summarize a set of documents by capturing their hidden semantic structures. Since each document is about specific topics, it is acceptable to assume that some words to occur more within a topic area. This means that documents can be represented by topics which are quantitatively less than all the words contained in each of the documents, while the essential relationships between documents remains the same. A statistical inference using an LDA method leads to a representation of a corpus of documents by distribution over topics for each document and a probability distribution over words associated with each topic. Figure 3.5 illustrates this decomposition.

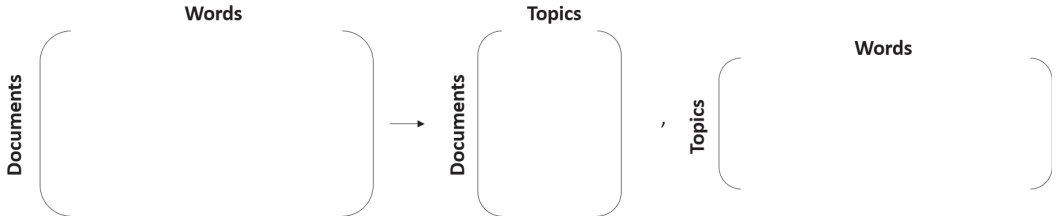

Figure 3.5: Latent dirichlet allocation decomposition

Just as other common topic modelling techniques, such as latent semantic indexing and probabilistic latent semantic indexing, LDA relies on bag-of-words assumptions that assume a document as a vessel for words. This method does not account for the grammar or order of words in a document, but focuses on the frequency and occurrence of words. The novelty of LDA is introducing a Dirichlet prior distribution over document-topic and topic-word distributions. A Dirichlet distribution makes the underlying problem of statistical inference easier because it acts as a conjugate prior for the multinomial distribution.

If the vocabulary is denoted by $\{1, 2, ..., V\}$, words are represented by a V-vector w with only one component equal to one and the remaining components equal to zero. A document with N number of words is denoted by $\mathbf{w} = (w_1, w_2, ...w_N)$ where the subscript words refer to their order within the document. And a corpus with M number of documents is denoted by $D = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M\}$.

3.2.1 Generative process

The generative process of LDA aims to investigate how the documents in a corpus are produced or how a new document can be produced according to the topics discovered from a particular corpus. Each word in a document is generated by first sampling topics from the topic distribution, then by choosing words from the topic-word distribution. The mathematical explanation of this process is as follows [19]:

1- Set the number of topics k

2- Choose $N \sim$ Poisson(ξ)

3- Choose $\theta \sim$ Dirichlet(α)

4- For each word w_n:

- Choose a topic $z_n \sim$ Multinomial(θ)

- Choose a word $w_n \sim$ Multinomial$(\beta|z_n)$

In this notation, θ and z are k-dimensional random variables. θ indicates which topics are important for a particular document (topics distribution) and z points to the assigned topic for each word. β is a $k \times V$ matrix indicates word distributions in each topic. Graphical representations of the LDA generative process is shown in Figure 3.6.

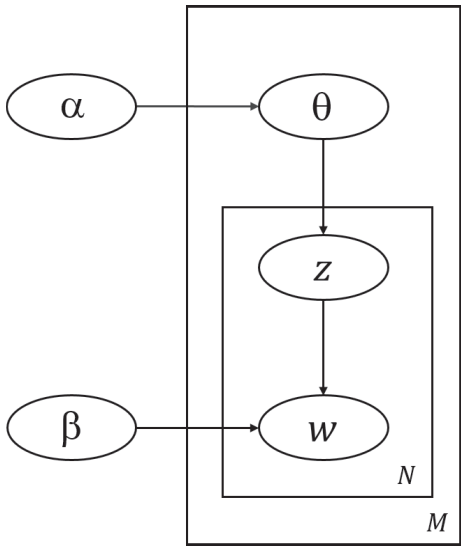

Figure 3.6: Graphical model for LDA. It shows dependencies among defined variables. Plates refer to repetition of sampling and arrows indicate conditional dependencies between variables [19, 84].

The joint distribution over topic distribution θ, assignments of topics to N words $\mathbf{z}$, and a set of N words in a document $\mathbf{w}$, given the parameters α and β, can be measured according to this process. On the right-hand side of Equation 3.17, the first statement follows a Dirichlet distribution and the last two statements refer to multinomial distributions.

$$P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = P(\theta|\alpha) \prod_{n=1}^{N} P(z_n|\theta) P(w_n|z_n, \beta) \tag{3.17}$$

3.2.2 Statistical inference

LDA tries to make sense of large collections of unstructured documents by inferring the probability distribution over topics in each document (θ) and the topic responsible for the generation of each word ($\mathbf{z}$). This inference problem can be represented in the following posterior distribution:

$$P(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{P(\mathbf{w}|\alpha, \beta)} \tag{3.18}$$

There are two main types of approximate posterior inference algorithms for this question. The first type relies on sampling techniques, such as Markov Chain Monte Carlo (MCMC), and the second

one is based on optimization techniques, mainly variational inference. According to the literature, optimization techniques have greater speed and comparable results as the sampling techniques, which makes variational inference a better solution to use when there are massive datasets.

Variational inference approximates the posterior probability in Equation 3.18 by detecting a family of distributions with simpler forms and their related free parameters. This family is called a variational distribution and can be represented as follows:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n) \tag{3.19}$$

In order to determine the free variational parameters, a Kullback Leibler (KL) divergence can be used as the cost function between the variational distribution $q(\theta, \mathbf{z}|\gamma, \phi)$ and the actual posterior $P(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$. Minimizing the Kullback Leibler divergence will lead to an update in the free variational parameters as follows:

$$\phi_{ni} \propto \beta_{iw_n} exp\{E_q\left[log(\theta_i)\,|\gamma\right]\} \tag{3.20}$$

$$E_q\left[log(\theta_i)\,|\gamma\right] = \Psi(\gamma_i) - \Psi(\sum_{j=1}^{k} \gamma_j) \tag{3.21}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^{N} \phi_{ni} \tag{3.22}$$

Ψ is computed via a Taylor approximation and is the first derivative of the log Γ function. The variational inference technique for LDA is summarized in Algorithm 2.

Algorithm 2 Variational inference for LDA [19]

1. For all i and n initialize $\phi_{ni}^0 := 1/k$
2. For all i initialize $\gamma_i := \alpha_i + N/k$
3. Repeat
4. For $n = 1$ to N
5. For $i = 1$ to k
6. $\phi_{ni}^{t+1} := \beta_i w_n exp(\Psi(\gamma_i^t))$
7. Normalize ϕ_n^{t+1}
8. $\gamma^{t+1} := \alpha + \sum_{n=1}^{N} \phi_n^{t+1}$
9. Until convergence

3.3 Experiment Design

In this study, it is of interest to detect stock switching points from historical stock data and to analyze the corresponding financial news to predict upcoming stock switching points.

For the purpose of stock switching points detection, an OBCD is implemented, which is explained in section 3.1. The input in this method is the daily stock prices related to a company whose stock switching points are of interest. Due to its online nature, the stock price data arrives as a stream and the days that correspond to detected switching points will be specified (Fig 3.7). According to Algorithm 1, the distribution of the input data should be assumed to determine the predictive probability. A normal distribution is assumed for historical stock data between each two consecutive switching points. Furthermore, according to the definition of change points as the abrupt changes in the **generative parameters** of the data, which parameters of interest must be defined: either the mean or the variance in case of a normal distribution. Contacting domain experts helps to define change points either as the changes in mean and/or the changes in the variance of the data distribution. This leads to the scenario **Unknown mean - Unknown variance** as explained in section 3.1. The controllable factor is λ, which is the general parameter in all types of probability distributions for OBCD Detection. Furthermore, assuming a normal distribution for the time series and looking for the changes in mean and/or variance causes to have the conjugate prior in the form of a **Normal-inverse-Gamma** distribution with parameters μ_0 , α_0 , β_0 and κ_0. Since changing these parameters does not affect the performance of the experiment, the parameter are set by using the original papers setting [2].

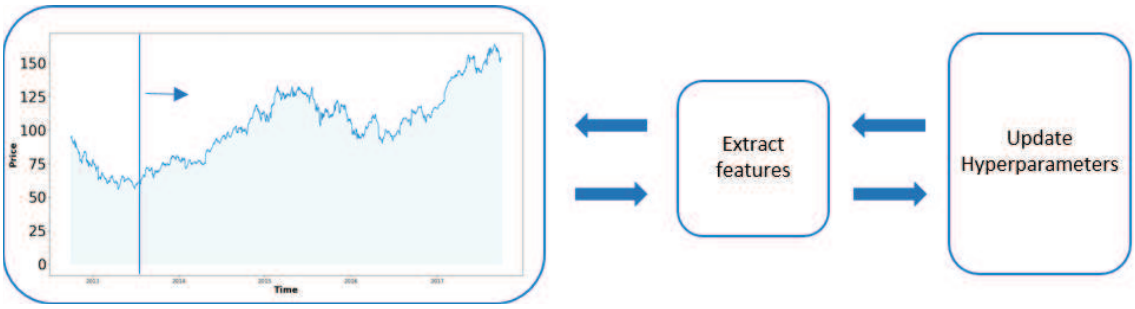

Figure 3.7: Online change point detection

Results of OBCD is the probability over run length at each time. Detecting the points where the run length diminishes to zero create the detected timestamps as change points through this algorithm. Figure 3.8 illustrates this process for artificial data. X-intercepts of the slashes that correspond to run lengths with the highest probability are detected as the output of this algorithm for change points timestamps. The performance of this method is compared with a grant truth provided by the domain experts. This grant truth is a timestamp of change points related to time period and stock companies of our study.

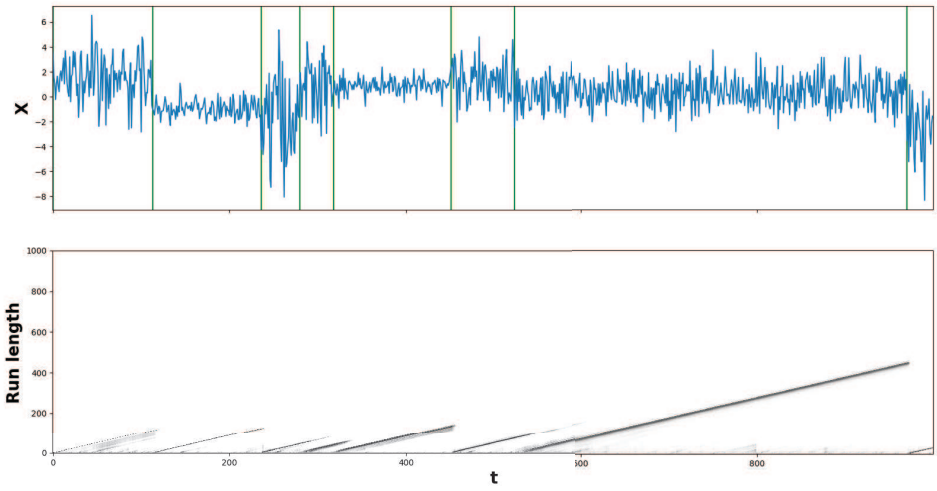

Figure 3.8: The top plot shows artificial data with defined changes in mean and/or variance as green lines. The bottom one is posterior probability of the current run $P(r_t|x_{1:t})$ at each time step, using a logarithmic color scale. Darker pixels indicate higher probability [2]. As it is shown, time steps with run length as zero matches with defined change points.

Detecting the change points by OBCD leads us to the second purpose of this study, finding the relation between stock switching points and financial news. For this purpose, the proposed method is to learn about the topics from relevant news sources for 5 previous change points by LDA method using 100 topics. Relevant news related switching points is defined by comparing the display date of news and switching points time of occurrence within a three day window of the news coverage. News appearing on the same day as a switching point or the day before and after a switching point will be considered "relevant news". The experiment design has a rolling window process so that in each iteration, topics are discovered through the previous relevant news training phase and the current news will be summarized under the same topic representations using the word distributions developed from the relevant news testing phase. In the testing phase, the similarity between topic representations of current news and previous relevant news can act as an indicator for predicting stock switching points. It is assumed that news relevant to stock switching points will act as an indicator for the switching points, and as a result, news with similar topics to previous relevant news should cause a change point. The similarity of these k-dimensional vectors can be measured by different criteria such as [84]:

- **Cosine Similarity:** It measures the similarity of two non-zero vectors (e.g. $\overrightarrow{\mathrm{A}}$ and $\overrightarrow{\mathrm{B}}$) by their dot products and magnitude as follows:

$$CS(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) = 1 - \frac{\overrightarrow{\mathrm{A}}.\overrightarrow{\mathrm{B}}}{\|\overrightarrow{\mathrm{A}}\|\|\overrightarrow{\mathrm{B}}\|} \tag{3.23}$$

- **Kullback Leibler Divergence:** It measures the similarity of two non-zero vectors (e.g. $\overrightarrow{\mathrm{A}}$ and $\overrightarrow{\mathrm{B}}$) by comparing their elements as follows:

$$D(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) = \sum_{i=1}^{k} A_i \, log_2 \frac{A_i}{B_i} \tag{3.24}$$

$$KL(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) = \frac{1}{2}[\, D(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) + D(\overrightarrow{\mathrm{B}}, \overrightarrow{\mathrm{A}})] \tag{3.25}$$

- **Jenson-Shannon Divergence:** It measures the similarity of two non-zero vectors (e.g. $\overrightarrow{\mathrm{A}}$ and $\overrightarrow{\mathrm{B}}$) by comparing their elements as follows:

$$JS(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) = \frac{1}{2}[\, D(\overrightarrow{\mathrm{A}}, (\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}})/2) + D(\overrightarrow{\mathrm{B}}, (\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}})/2)] \tag{3.26}$$

- **Euclidian Distance:** It measures the similarity of two non-zero vectors (e.g. $\overrightarrow{\mathrm{A}}$ and $\overrightarrow{\mathrm{B}}$) by comparing their elements as follows:

$$d(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + ... + (A_k - B_k)^2} \tag{3.27}$$

For all of these metrics, the smaller values point to greater similarity. Each of the above-mentioned metrics measure the similarity by specific criteria. In order to include all of them, the last three metric can be normalized by the exponential cumulative probability distribution and then the average of all four metrics can be considered as the similarity value.

$$\text{Similarity}(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) = \frac{CS(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}}) + (1 - e^{-KL(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}})}) + (1 - e^{-JS(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}})}) + (1 - e^{-d(\overrightarrow{\mathrm{A}}, \overrightarrow{\mathrm{B}})})}{4} \tag{3.28}$$

As with all text mining techniques, text normalization and tokenization steps take place for each piece of news coverage before the LDA model is utilized. These steps include removing special characters like #, expanding contractions like changing *it's* to *it is* (123 contractions - Appendix 1), case conversions (convert to lower case), removing stop words like *itself* or *which* (153 words), correcting words with misspelling or repeating characters, stemming and lemmatization [80]. After text normalization and tokenization, the features of textual content are extracted to create the vector space model for each piece of news coverage. Weights for the features (words) are measured based on the frequency of words due to the nature of the VB inference technique [19]. The cleaned, vectorized news pieces are then used as the input for the LDA method.

3.3.1 Experiment Dataset

According to the design of this experiment, there is a need for two types of data: the source of historical stock prices to detect the switching points, and the source of financial news to test the predictive power of financial news for stock switching points. In the following section, these data sources are explained and appropriate exploratory analysis of them is provided.

- **Historical stock prices:** Implementation of this experiment covers three different stocks: Apple (AAPL), Microsoft (MSFT) and Facebook (FB). Yahoo finance provides historical stock prices through a free Application Programming Interface (API). Historical closing price of these companies are retrieved from the first day of October 2012 to the first day of October 2017 to provide data from over a five-year span (Fig 3.9). Statistics from these time series is reported in Figure 3.12 and a seasonal pattern is investigated in Figures 3.10 and 3.11.

Figure 3.9: Closing Prices of three tech companies Apple, Microsoft and Facebook from Oct 1st, 2012 to Oct 1st, 2017.

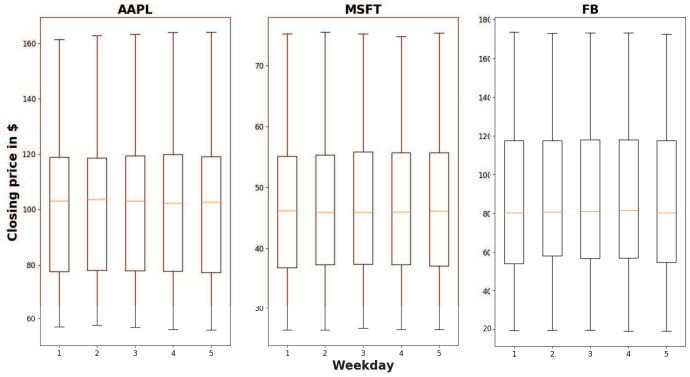

Figure 3.10: Box plot for weekday closing price for five years data. As it is shown, there is no weekly seasonality for these stocks as they have same distributions among weekdays.

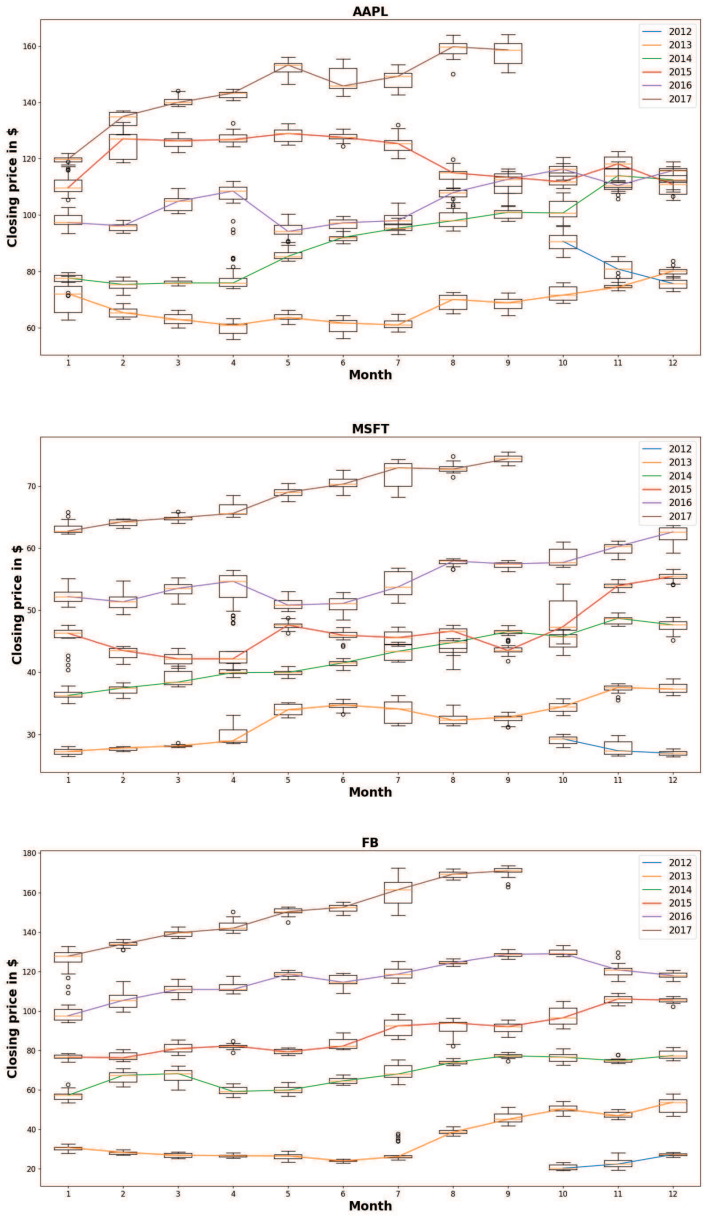

Figure 3.11: Box plot for monthly closing price for five year data. As it is shown, there is no monthly or yearly seasonality for these stocks.

	Min	Q_1	Mean	Median	Q_3	Max	Std
AAPL	55.79	77.71	102.74	103.07	119.22	164.05	26.63
MSFT	26.37	37.24	47.07	46.10	55.69	75.44	12.74
FB	18.98	55.21	85.66	80.55	117.69	173.51	40.87

Figure 3.12: Statistics of Historical Stock Closing Price Data

- **Financial news:** Dow Jones has recently begun to provide a new version of financial news. This dataset is in standard XML format (DJNML) and has valuable meta-data. The most data provided in this source are: display dates of the news (which can help to determine concurrency of switching points and news) and names of the companies related to news pieces (which makes it possible to easily filter relevant news for a given company). As a result, XML files of the news pieces from this Dow Jones dataset are parsed. From each news piece, four important elements are captured: the display date, the related companies, the headline and the text of the news. The retrieved elements are then structured in a PostgreSQL database to create a record for each news piece with four different columns to make it possible to query easily. The dataset used for the experiment belongs to the same time window as the historical stock closing price: the first day of October 2012 to the first day of October 2017. Same as historical stock price data, news pieces relevant to AAPL, MSFT and FB stock are selected. The number of news pieces related to each of these stocks is reported in Table 3.1 and an average length of the news pieces in terms of number of characters or words is summarized in Table 3.2. Figures 3.13, 3.14 and 3.15 illustrate most frequent words in news related to these stocks.

	2012	**2013**	**2014**	**2015**	**2016**	**2017**	**Sum**
AAPL	2681	7744	3463	3576	2426	3101	22991
MSFT	1885	5804	2245	1798	1176	1344	14252
FB	1856	5837	2480	2106	1693	2136	16108

Table 3.1: Number of news related to each stock from 2012/10/01 to 2017/10/01

	Avg # Chars	**Sum # Chars**	**Avg # Words**	**Sum # Words**
AAPL	2901.29	66703558	452.11	10394475
MSFT	3142.26	44783431	480.67	6850500
FB	3301.53	53181043	497.50	8013742

Table 3.2: Length of news related to each stock from 2012/10/01 to 2017/10/01

Figure 3.13: Wordcloud for Apple stock. Most frequent words in news related to AAPL. Words with larger size points to higher frequency

Figure 3.14: Wordcloud for Microsoft stock. Most frequent words in news related to MSFT. Words with larger size points to higher frequency

Figure 3.15: Wordcloud for Facebook stock. Most frequent words in news related to FB. Words with larger size points to higher frequency

3.4 Performance Measurement

To measure the achievement and applicability of an algorithm, several performance measures are used in machine learning studies. The type of measures used depend on the definition of the problem that is being explored. This study aims to test which component of a timeseries dataset is related to a stock switching point. Even though the literatures in this domain mainly reports the performance visually by picturing the concurrency of detected switching points and some micro-economic events [2, 79, 38, 39, 87], there are a few studies which assess the performance mathematically [16, 45]. To consider all aspects of the experiment, the success rate of the algorithm in this study is measured by using following metrics:

- **Compare probability distribution:** Change points are the points at which generative parameters change significantly before and after. To test the performance of the detection of such points, an appropriate statistical test can be used. Since this study focuses on the detection of timestamps with changes in mean or variance, a t-test and an f-test can be used respectively.

- **False positive rate:** Suppose that $\boldsymbol{\tau} = (\tau_1, \tau_2, ..., \tau_J)$ is the location of a detected change point in our experiment. τ_i is a true positive if there exists a true change point in a window size of h. Otherwise, it is a false positive. The false positive rate is the number of false positives divided by total number of detected change points.

- **Mean time between false alarms:** This metric measures how frequently the algorithm predicts a false change point. It can show the reliability of the system for use by traders.

- **Mean delay for detection:** The mean delay for detection can be measured by finding the closest actual change point to each predicted one and by taking the average of these distances for all change points. This metric can provide insights about detection delay of the system.

- **Probability of non-detection:** It is also important to determine how many of the actual change points are predicted by this methodology. True positives are defined as if there exists a predicted change point in a window size of h from the actual change point, all remaining actual change points are false negatives. The probability of non-detection is the number of false negatives divided by the total number of actual change points.

- **Number of predicted change points:** It is also important to compare the number of predicted and actual change points so as not to overestimate or underestimate the performance by increasing or decreasing the number of detected change points.

Algorithm 3 Experiment Pseudocode

Require: Initialize $\alpha_0, \beta_0, \kappa_0, \mu_0$

for each x_t **do** ▷ Iterate through price of each day
  $\pi_t^{(r)} = P(x_t|\alpha_0, \beta_0, \kappa_0, \mu_0)$ ▷ Predictive probability
  $P(r_t = r_{t-1} + 1, \boldsymbol{x}_{1:t}) = P(r_{t-1}, \boldsymbol{x}_{1:t-1})\pi_t^r(1 - H(r_{t-1}))$ ▷ Probability of growth for run length
  $P(r_t = 0, \boldsymbol{x}_{1:t}) = \sum_{r_{t-1}} P(r_{t-1}, \boldsymbol{x}_{1:t})\pi_t^r H(r_{t-1})$ ▷ Probability of getting zero for run length
  $P(r_t|\boldsymbol{x}_{1:t}) = P(r_t, \boldsymbol{x}_{1:t})/P(\boldsymbol{x}_{1:t})$ ▷ Run length distribution for the day
  $\alpha_{t+1} = \alpha_t + 1/2$ ▷ Update hyperparameters
  $\kappa_{t+1} = \kappa_t + 1$
  $\beta_{t+1} = \beta_t + (\kappa_t(x - \mu_t)^2)/(2(\kappa_t + 1))$
  $\mu_{n+1} = \dfrac{\kappa_n \mu_n + x}{\kappa_n + 1}$
  if $r_t : max(P(r_t)) = 0$ **then**
    change points.add(t)
  end if
end for
for each y_t **do** ▷ Iterate through news of each day
  Recent_change points = Choose(5 recent, change points)
  Relevant_News = $y_z : z \in$ Recent_change points ± 1
  Normalize and Tokenize Relevant_News and y_t
  $\widehat{y_t}$ = Transform(y_t|LDA(Relevant_News))
  if Similarity($\widehat{y_t}$, LDA(Relevant_News)) $<$ threshold **then**
    Predicted_change points.add(t)
  end if
end for

Chapter 4

Results

4.1 Change points Detection

Change points for three stocks, AAPL, MSFT and FB, are first detected by OBCD by using their historical closing price data from the first day of October 2012 to the first day of October 2017. The run length representation of this method and its detected change points are visualized in figures 4.1 and 4.2.

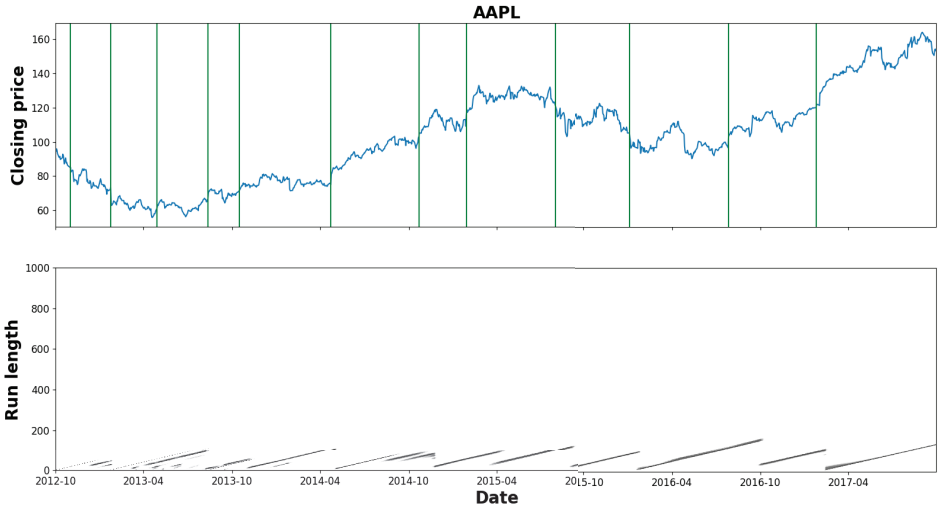

Figure 4.1: OBCD for AAPL stock.Darker pixels indicate higher probability. Run length equal to zero represent change points. Vertical green lines are timestamps related to detected change points by OBCD.

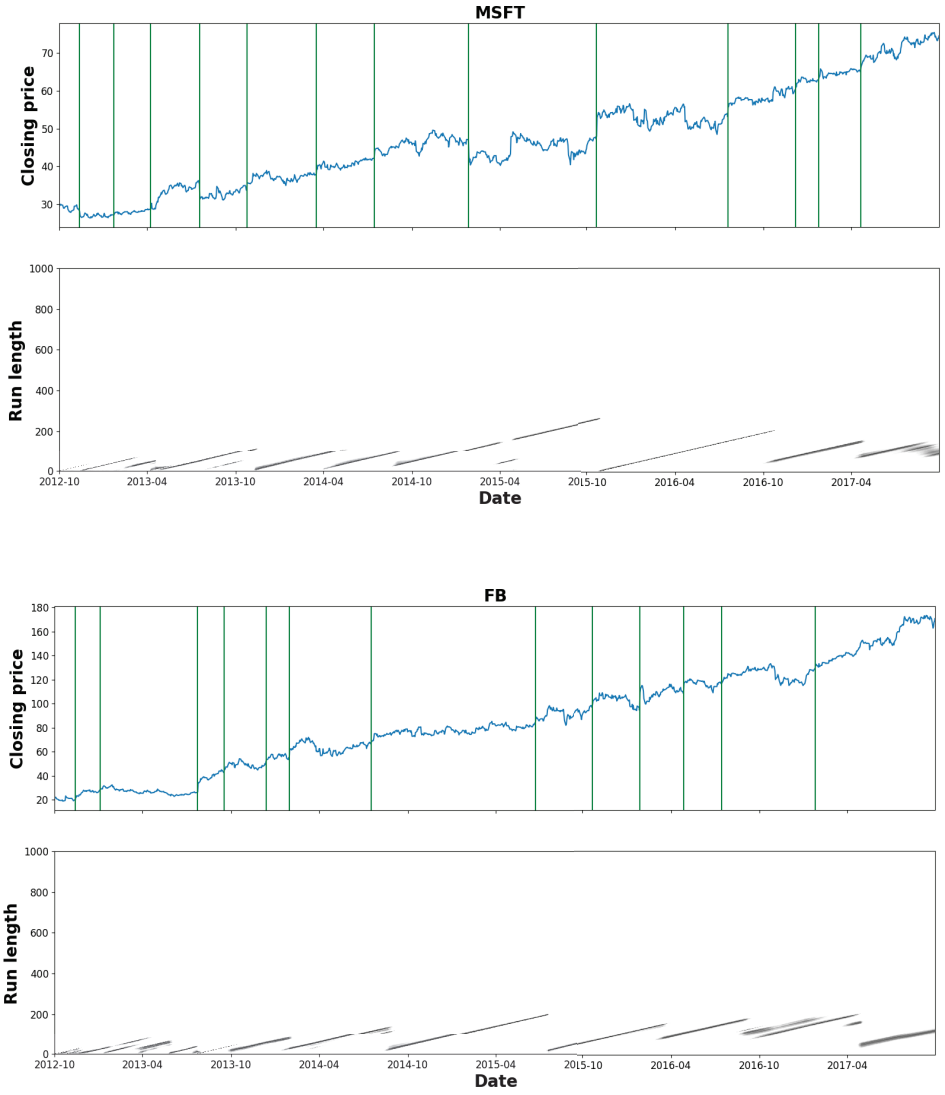

Figure 4.2: OBCD for MSFT and FB stock respectively. Darker pixels indicate higher probability. Run length equal to zero represent change points. Vertical green lines are timestamps related to detected change points by OBCD.

These visualizations help to represent the signal of closing price by the defined variable, run length. Further more, they show how the intersections of run length slopes and x-intercept are detected to retrieve the detected change points. The performance of OBCD technique can be understood by comparing the position of vertical green lines and the closing price signal.

Change points are defined as abrupt changes in the generative parameters of a sequence of data. In this study, both mean and variance are considered as the generative parameters that is of interest to detect their changes. As a result, to have a good metric of measuring the performance of the implemented technique, it is beneficial to apply appropriate statistical tests. Data points between each two consecutive change points can be considered as a partition. A T-test or an F-test can show if there is a significant difference between mean and variance of each two consecutive partitions respectively. The P-value results for T-test and F-test are reported for each three stock: AAPL, FB and MSFT and for each detected change point in table 4.1.

#	AAPL,T_test	AAPL,F_test	FB,T_test	FB,F_test	MSFT,T_test	MSFT,F_test
1	6e−23	9e−2	2e−21	2e−2	5e−29	1e−4
2	9e−41	1e−2	5e−15	3e−5	2e−24	4e−1
3	6e−14	1e−1	8e−71	5e−1	3e−38	2e−25
4	7e−1	2e−2	1e−85	1e−14	1e−2	0
5	9e−17	5e−2	3e−32	0	3e−64	5e−2
6	2e−111	1e−21	2e−93	0	7e−62	5e−1
7	7e−77	2e−5	8e−142	0	3e−72	2e−9
8	5e−139	1e−1	2e−49	0	3e−9	1e−4
9	1e−26	2e−3	5e−47	0	8e−127	1e−2
10	5e−4	1e−2	5e−43	0	2e−71	4e−4
11	1e−21	3e−2	4e−39	0	3e−34	1e−6
12	3e−163	5e−18	3e−87	0	8e−24	2e−1
13	-	-	6e−151	0	5e−52	4e−19

Table 4.1: T_test and F_test P_values for datapoints before and after each change point. Considering α_level as 0.05, for all of the change points at least one of the mean or the variance has changed. Red cells are p_values larger than α_level which point to not a significant abrupt change in terms of the corresponding parameter.

The results show that at each detected change point, at least one of the generative parameters: mean or variance are changed significantly. It points that OBCD was totally capable to detect change points in this experiment.

The last type of results are related to the comparison of the detected change points of OBCD and the grant truth values provided by domain experts. To improve the comparisons, the same metrics are measured for random change point detection. The random change point detection multiplied by 1000 with same number of detected change points as OBCD is implemented and the average performance is measured. Different metrics are explained in section 3.4 and reported in tables 4.2, 4.3, 4.4, 4.5 and 4.6.

	Numebr of Actual change points	**Number of detected change points by OBCD**
AAPL	16	12
MSFT	13	13
FB	16	13

Table 4.2: Comparing number of detected change points in grant truth and OBCD

	h=10 days	h=20 days	h=35 days
AAPL - OBCD	0.75	0.58	0.25
AAPL - Random	0.81	0.66	0.42
MSFT - OBCD	0.62	0.62	0.46
MSFT - Random	0.85	0.73	0.57
FB - OBCD	0.70	0.46	0.15
FB - Random	0.82	0.66	0.50

Table 4.3: Comparing false positive rate of OBCD and random guessing with different window sizes (in days).

	h=10 days	h=20 days	h=35 days
AAPL - OBCD	170.5	213.5	640.5
AAPL - Random	172.1	209.54	303.36
MSFT - OBCD	231.43	231.43	306.6
MSFT - Random	157.18	185.9	250.16
FB - OBCD	185.25	296.4	1210
FB - Random	160.35	200.29	256.56

Table 4.4: Mean time between false alarms for OBCD and random guessing with different window sizes (in days).

	OBCD	**Random**
AAPL	54	75.38
MSFT	27.1	65.64
FB	34.8	68.54

Table 4.5: Mean delay for detection for OBCD and random guessing.

	h=10 days	h=20 days	h=35 days
AAPL - OBCD	0.81	0.69	0.38
AAPL - Random	0.87	0.76	0.61
MSFT - OBCD	0.62	0.62	0.31
MSFT - Random	0.86	0.74	0.59
FB - OBCD	0.75	0.65	0.25
FB - Random	0.86	0.75	0.59

Table 4.6: Probability of non-detection for OBCD and random guessing with different window sizes (in days).

In table 4.2 comparison between total number of detected change points and grant truth change points is reported. It shows that these two numbers are in same range for the three investigated stocks. The highest difference belongs to AAPL with four less number of detected change points and the lowest difference belongs to MSFT with same number of detected change points as the actual number in the grant truth. According to this metric, OBCD perform enough good for all of the three stocks.

Table 4.3 is reporting the performance of OBCD by measuring the false positive rate. False positive change points are the detected change points with no actual change point in h days before or after their time of occurrence. False positive rate refers to the proportion of false positive instances to the total number of detected change points. Due the dependency of this metric to the window size, h, it is beneficial to measure it with different setting. Further more, to have a good comparison, 1000 times of random guessing is implemented and the false positive rate is measured. As reported in table 4.3, increasing the window size helps to have a better false positive rate for all stocks and window sizes. Performance of OBCD is better comparing to random guessing by having lower false positive rate for all three stocks in this study.

In addition to comparing number of change points and false positive rate, it is informative to measure the time distance between actual and detected change points. Table 4.4 shows the average time between each two consecutive false positive detected change points by the above mentioned definition of false positive instances. Same as false positive rate, increasing the window size improve the performance. In addition, mean time between false alarms is larger for OBCD technique comparing to random guessing. It points out that OBCD outperform random guessing by taking mean time between false alarms as the metric.

Table 4.5 is reporting the delay of the system for detection of change points. It is beneficial to capture how fast the system can detect the change points. Comparing the suggested technique, OBCD, and random guessing can helps to illustrate the capability of OBCD to detect change points. For all the stocks which are investigated in this study, OBCD has less delay to detect the change points comparing to random guessing.

The last metric which is measured to understand the performance of OBCD is to find the probability of non-detecting the change points. This metric tries to report the reliability of the system to detect the change points. For each actual change point, there are two possibilities, even there is a detected change point in an h days window, or there is no detected change point related to this actual change point. This metric is in line with the above mentioned metrics and prove the power of OBCD comparing to random guessing.

4.2 Change points Prediction

Change points for the three stocks AAPL, MSFT and FB are predicted by using historical stock data as well as news related to each stock. The performance of the whole experiment is reported in tables 4.7, 4.8, 4.9 and 4.10. These sets of results points to the performance of change point prediction by detecting the last five change points from historical stock data through OBCD and then learn the topics of news relevant to detected change points through LDA. Similarity of news related to past detected change points and the test news is acted as an indicator of future stock switching points. The similarity threshold is set to 10^{-6}. If the similarity between a current news and relevant news is larger than the specified threshold, the system is predicting a new change point. The predicted and detected change points are compared to evaluate the performance of the whole experiment.

	h=10 days	h=20 days	h=35 days
AAPL	0.85	0.675	0.475
MSFT	0.75	0.61	0.5
FB	0.79	0.61	0.46

Table 4.7: False positive rate of stock switching points prediction by financial news. It is reported by assuming different window sizes (in days). The similarity threshold is set to 10^{-6}.

	h=10 days	h=20 days	h=35 days
AAPL	21.57	27.38	38.72
MSFT	35.81	44.04	53.47
FB	20.63	23.26	30.48

Table 4.8: Mean time between false alarms for prediction of stock switching points by financial news with different window sizes (in days). The similarity threshold is set to 10^{-6}.

	AAPL	**MSFT**	**FB**
Mean delay for detection	108.83	82.31	80.15

Table 4.9: Mean delay for detection in days for stock switching points prediction by financial news. The similarity threshold is set to 10^{-6}.

	h=10 days	h=20 days	h=35 days
AAPL	0.67	0.5	0.5
MSFT	0.46	0.38	0.38
FB	0.38	0.31	0.31

Table 4.10: Probability of non-detection for stock switching points prediction by financial news with different window sizes (in days). The similarity threshold is set to 10^{-6}.

Table 4.7 is reporting the performance of the change point prediction by measuring the false positive rate. False positive change points are the predicted change points with no detected change point in h days before or after their time of occurrence. False positive rate refers to the proportion of false positive instances to the total number of predicted change points. Due the dependency of this metric to the window size, h, it is beneficial to measure it with different setting. As reported in table 4.7, increasing the window size helps to have a better false positive rate for all stocks and window sizes. The values of this table are reported by setting the similarity threshold to 10^{-6}.

As a next metric to measure the performance of the proposed technique, it is informative to measure the time distance between predicted and detected change points. Table 4.8 shows the average time between each two consecutive false positive predicted change points by the above mentioned definition of false positive instances. Same as false positive rate, increasing the window size improve the performance. According to this metric, the proposed method has been able to best predict change points for Fb stock and then AAPL.

Table 4.9 is reporting the delay of the system for prediction of change points. It is beneficial to capture how fast the system can detect the change points. Comparing the result of proposed method for the three investigated stocks: AAPL, MSFT and FB shows the proposed method outperform for FB by having less delay to predict the change points comparing to AAPL and MSFT stocks.

The last metric which is measured to understand the performance of the implemented techniques is to find the probability of non-predicting the change points. This metric tries to report the reliability of the system to detect the change points. For each detected change point, there are two possibilities, even there is a predicted change point in an h days window, or there is no predicted change point related to this detected change point. This metric is in line with the above mentioned metrics and shows better performance results for FB stock.

4.3 Threats to Validity

Threats to validity of this research involves internal, external, construct and statistical conclusion threats:

- **Internal Validity:**

  Internal validity concerns about the causal relationships that are concluded in an experiment. The measures which are used, the research setting, and the whole research design effects on the internal validity. In this study, a complete method for stock switching points prediction first requires a

change point detection technique that is applied using historical stock data to develop a clear definition about change points. Detected change points help ensure that only relevant and informative news, not irrelevant news, will be retrieved. A topic modelling technique is applied on relevant news pieces during the training phase to uncover the topics of interest that will be used in the model. The testing phase will assess whether the current news pieces discuss the same topics as the training news pieces. The degree of similarity between the topic representation of training news and the testing news is measured based on an ensemble similarity technique which combine various similarity techniques, such as cosine similarity, Kullback Leibler divergence, Jenson-Shannon divergence and Euclidian distance. This similarity technique helps to cover all aspects of similarity and to ensure confidence about the degree of similarity. Furthermore, performance is measured not only by one criteria, but by different metrics and time windows that might be of interest for different stakeholders having different purposes. The effects of extraneous variables have also been taken into consideration in this study. For example, concerns about seasonal patterns in historical stock data is mitigated by performing exploratory data analysis.

- **External Validity:**

  This type of validity refers to the degree to which the results of an experiment can be generalized to the population. While this study explores three stocks, AAPL, FB and MSFT, the methods have the capability to be implemented to study other companies as well. The LDA implementation is designed in two phases. The first phase trains the model to discover which topics will be used in the test. The second phase transforms unseen news under the same topic headings discovered in the test phase. Applying LDA for test news and measuring the performance based on unseen news, make this model and its results generalizable. But as visible in the results, the performance of this method may be different from stock to stock.

- **Construct Validity:**

  Construct validity is the degree to which an experiment measures what it claims to be measuring. To achieve this validity, first, the change points are detected using stock specific historical data. Then, the relevant news pieces for the specific stock are selected according to the concurrency of the news pieces and the detected change points. LDA training is implemented to determine the similarity between relevant news and unseen news by their topic representations.

- **Statistical Conclusion Validity:**

  Statistical conclusion validity refers to the degree to which conclusions about the relationship among variables are correct. To measure the performance of change point detection, appropriate statistical tests are used to test the equality of mean or variance for two samples, one with data points before a change point and the other with data points after the same change point. The underlying assumptions of a two-sample t-test and f-test, such as a normality assumption, are met before applying the tests. It is important for a statistical test that the measuring process is reliable. In this case, the data points are historical stock data selected from reliable sources and the samples are homogeneous, so the measuring process should be reliable.

Chapter 5

Conclusion

Prediction of stock switching points is an important asset for stock market traders. Traders need to make trading decisions with high stakes and high benefits. In this research, a new approach is suggested to solving the problem of predicting stock switching points. The proposed technique first detects stock change points from historical stock data by using an OBCD technique. Second, the technique determines what news is relevant to past detected stock switching points and then trains an LDA model using these news pieces to determine what underlying topics are associated with earlier detected stock switching points. Using these topics helps to determine how much news pieces that discuss the same topics can be used as a predictor for future stock change points. The whole experiment is implemented for three different stocks (AAPL, MSFT and FB) and include five years of data. The well-structured Dow Jones data set is used as the source of financial news.

The results show that OBCD technique is able to detect the change points as verified by the p-values of a t-test and an f-test. For each detected change point, at least one of the generative parameter, mean or variance, has changed significantly. The visualized stock price signals and their corresponding run lengths show how the method is able to detect change points precisely. It is also clear from the results that the performance of the OBCD is better compared to the average performance of random guessing for all reported metrics, for all stocks studied and for all different window sizes.

The results also show that combining OBCD, LDA model and text similarity measures can help to best capture stock price switching points. The combination has the advantage of detecting stock switching points from historical stock data and training the text mining technique to use only relevant news. Relevant news consists of news published close to the time of switching points. The hidden structure inferred from a set of relevant news items can be used to predict future switching points. This will result in reducing the storage cost of news for text mining platforms which aim to predict stock market movements.

Even though the performance is not same for all the investigated stocks, all of them increase the desired window sizes, leading to better results for all reported metrics. It is clear that when the system lets the method predict change points in a wider range, the performance becomes better. On the other hand, different metrics might be of different degrees of importance for different traders and stakeholders.

These metrics are related to each other and improving one metric might worsen the others. Parameters of this technique can be modified according to the importance of any of these metrics.

Contributions: The proposed method has the advantage of using only historical stock data and news related to historical changes to predict future stock switching points. Previous proposed methods have mainly focused on only one part of these whole process: change point detection or change point prediction. Combining a change point detection technique and text mining approaches for the purpose of stock switching points prediction is a novel technique explored in this research.

Future works: Future researchers can implement this method for different companies using a larger datasets. Furthermore, the LDA model may be applied for other settings and contexts. The model roposed in this study can be used as the baseline model for comparing the results of the extended LDA algorithms, or for comparing other types of topic modeling techniques, such as matrix factorization or its extended version, tensor factorization.

Appendix 1

Expand contraction [80]

```
CONTRACTION_MAP = {
    "ain't": "is not",
    "aren't": "are not",
    "can't": "cannot",
    "can't've": "cannot have",
    "'cause": "because",
    "could've": "could have",
    "couldn't": "could not",
    "couldn't've": "could not have",
    "didn't": "did not",
    "doesn't": "does not",
    "don't": "do not",
    "hadn't": "had not",
    "hadn't've": "had not have",
    "hasn't": "has not",
    "haven't": "have not",
    "he'd": "he would",
    "he'd've": "he would have",
    "he'll": "he will",
    "he'll've": "he he will have",
    "he's": "he is",
    "how'd": "how did",
    "how'd'y": "how do you",
    "how'll": "how will",
    "how's": "how is",
    "I'd": "I would",
    "I'd've": "I would have",
```

"I'll": "I will",
"I'll've": "I will have",
"I'm": "I am",
"I've": "I have",
"i'd": "i would",
"i'd've": "i would have",
"i'll": "i will",
"i'll've": "i will have",
"i'm": "i am",
"i've": "i have",
"isn't": "is not",
"it'd": "it would",
"it'd've": "it would have",
"it'll": "it will",
"it'll've": "it will have",
"it's": "it is",
"let's": "let us",
"ma'am": "madam",
"mayn't": "may not",
"might've": "might have",
"mightn't": "might not",
"mightn't've": "might not have",
"must've": "must have",
"mustn't": "must not",
"mustn't've": "must not have",
"needn't": "need not",
"needn't've": "need not have",
"o'clock": "of the clock",
"oughtn't": "ought not",
"oughtn't've": "ought not have",
"shan't": "shall not",
"sha'n't": "shall not",
"shan't've": "shall not have",
"she'd": "she would",
"she'd've": "she would have",
"she'll": "she will",
"she'll've": "she will have",
"she's": "she is",
"should've": "should have",
"shouldn't": "should not",

"shouldn't've": "should not have",
"so've": "so have",
"so's": "so as",
"that'd": "that would",
"that'd've": "that would have",
"that's": "that is",
"there'd": "there would",
"there'd've": "there would have",
"there's": "there is",
"they'd": "they would",
"they'd've": "they would have",
"they'll": "they will",
"they'll've": "they will have",
"they're": "they are",
"they've": "they have",
"to've": "to have",
"wasn't": "was not",
"we'd": "we would",
"we'd've": "we would have",
"we'll": "we will",
"we'll've": "we will have",
"we're": "we are",
"we've": "we have",
"weren't": "were not",
"what'll": "what will",
"what'll've": "what will have",
"what're": "what are",
"what's": "what is",
"what've": "what have",
"when's": "when is",
"when've": "when have",
"where'd": "where did",
"where's": "where is",
"where've": "where have",
"who'll": "who will",
"who'll've": "who will have",
"who's": "who is",
"who've": "who have",
"why's": "why is",
"why've": "why have",

"will've": "will have",
"won't": "will not",
"won't've": "will not have",
"would've": "would have",
"wouldn't": "would not",
"wouldn't've": "would not have",
"y'all": "you all",
"y'all'd": "you all would",
"y'all'd've": "you all would have",
"y'all're": "you all are",
"y'all've": "you all have",
"you'd": "you would",
"you'd've": "you would have",
"you'll": "you will",
"you'll've": "you will have",
"you're": "you are",
"you've": "you have"

Appendix 2

OBCD for Normal distribution

1- Normal distribution with Known mean and Unknown variance:

```
def inference_Knownmean_Unknownvariance(x, hazard_func, mu0, alpha0, beta0):
    # MATRIX THAT HOLD THE PROBABILITY OF CURRENT RUN_LENGTH
     R = np.zeros([(len(x)+1), len(x)])
     # INITIALIZE
    R[0, 0] = 1.0
    mu0 = np.array([mu0])
    alpha0 = np.array([alpha0])
    beta0 = np.array([beta0])
    # TRACK THE CURRENT SET OF PARAMETERS.
    muT = mu0
    alphaT = alpha0
    betaT = beta0
    # KEEP TRACK OF THE MAX
    maxes = np.zeros([(len(x))])
    # LOOP OVER THE DATA
    for t in range(len(x)-1):
        # PREDICTIVE PROBABILITIES
        predprobs = studentpdf(x[t], muT, betaT / alphaT, 2 * alphaT)
        haz = hazard_func(np.arange(t + 1))
        # GROWTH PROBABILITY
        R[1:t+2,t+1] = R[0:t+1,t] * predprobs * (1 - haz)
        # change point PROBABILITIES
        R[0, t+1] = (R[0:t+1, t] * predprobs * haz).sum()
        R[:, t+1] = R[:, t+1] / (R[:, t+1].sum())
        # UPDATES
        muT0 = np.concatenate([mu0,muT])
```

```
        alphaT0 = np.concatenate([alpha0, alphaT + 0.5])
        betaT0 = np.concatenate([beta0, betaT + ((x[t] - muT) ** 2) / 2])
        muT = muT0
        alphaT = alphaT0
        betaT = betaT0
        maxes[t] = R[:, t].argmax()
    return R, maxes
```

2- Normal distribution with Unknown mean and Unknown variance:

```
def inference_Unknownmean_Unknownvariance(x, hazard_func, mu0, kappa0, alpha0, beta0):
    # MATRIX THAT HOLD THE PROBABILITY OF CURRENT RUN_LENGTH
     R = np.zeros([(len(x)+1), len(x)])
     # INITIALIZE
    R[0, 0] = 1.0
    mu0 = np.array([mu0])
    alpha0 = np.array([alpha0])
    beta0 = np.array([beta0])
    beta0 = np.array([kappa0])
    # TRACK THE CURRENT SET OF PARAMETERS.
    muT = mu0
    kappaT = kappa0
    alphaT = alpha0
    betaT = beta0
    # KEEP TRACK OF THE MAX
    maxes = np.zeros([(len(x))])
    # LOOP OVER THE DATA
    for t in range(len(x)-1):
        # PREDICTIVE PROBABILITIES
        predprobs = studentpdf(x[t], muT, betaT * (kappaT + 1) / (alphaT * kappaT), 2 * alphaT)
        haz = hazard_func(np.arange(t + 1))
        # GROWTH PROBABILITY
        R[1:t+2,t+1] = R[0:t+1,t] * predprobs * (1 - haz)
        # change point PROBABILITIES
        R[0, t+1] = (R[0:t+1, t] * predprobs * haz).sum()
        R[:, t+1] = R[:, t+1] / (R[:, t+1].sum())
        # UPDATES
        muT0 = np.concatenate([mu0,muT])
        alphaT0 = np.concatenate([alpha0, alphaT + 0.5])
        betaT0 = np.concatenate([beta0, betaT + ((x[t] - muT) ** 2) / 2])
        kappaT0 = np.concatenate([kappa0, kappaT + 1])
```

```
        muT = muT0
        alphaT = alphaT0
        kappaT = kappaT0
        betaT = betaT0
        maxes[t] = R[:, t].argmax()
    return R, maxes
```

3- Normal distribution with Unknown mean and Known variance:

```
def inference_Unknownmean_Knownvariance(x, hazard_func, mu0, sigma20, SIG2):
    # MATRIX THAT HOLD THE PROBABILITY OF CURRENT RUN_LENGTH
     R = np.zeros([(len(x)+1), len(x)])
     # INITIALIZE
    R[0, 0] = 1.0
    SIG2 = np.array([SIG2])
    mu0 = np.array([mu0])
    sigma20 = np.array([sigma20])
    # TRACK THE CURRENT SET OF PARAMETERS
    muT = mu0
    sigma2T = sigma20
    # KEEP TRACK OF THE MAX
    maxes = np.zeros([(len(x))])
    # LOOP OVER THE DATA
    for t in range(len(x)-1):
        # PREDICTIVE PROBABILITIES
        predprobs = normaldis(x[t], muT, sigma2T + SIG2)
        haz = hazard_func(np.arange(t + 1))
        # GROWTH PROBABILITY
        R[1:t+2,t+1] = R[0:t+1,t] * predprobs * (1 - haz)
        # change point PROBABILITIES
        R[0, t+1] = (R[0:t+1, t] * predprobs * haz).sum()
        R[:, t+1] = R[:, t+1] / (R[:, t+1].sum())
        # UPDATES
        sigma2T0 = np.concatenate([sigma20, (1 / ((1 / SIG2) + (1 / sigma2T)))])
        muT0 = np.concatenate([ mu0, ((SIG2 * muT + sigma2T * x[t]) / (sigma2T+SIG2))])
        muT = muT0
        sigma2T = sigma2T0
        maxes[t] = R[:, t].argmax()
    return R, maxes
```

Appendix 3

Performance metrics

1- False positive rate:

```
def performancebyfprate(predictdata,actualdata,h):
    TP = 0
    f = False
    for i in predictdata :
        for j in actualdata:
            if abs((i-j).days) <= h :
                f=True
         if f==True:
            TP += 1
        f = False
    return (1-(TP/len(predictdata)))
```

2- Mean delay for detection:

```
def performancebymean_delay_detection(predictdata,actualdata):
    delay=0
    for j in actualdata:
        x = float("inf")
        for i in predictdata:
            if abs((i-j).days) <= x :
                x = abs((i-j).days)
        delay = delay+x
    meandelay = delay/len(actualdata)
    return (meandelay)
```

3- Mean time bet en false alarm:

```
def performancebymean_time_false_alarm(predictdata,actualdata,h):
    false_alarms = []
    f=False
    for i in predictdata :
        for j in actualdata:
            if abs((i-j).days) <= h :
                f=True
        if f==False:
            false_alarms = false_alarms + [i]
        f = False
    false_alarms.sort()
    diff = 0
    for i in range(len(false_alarms) - 1):
        diff += (false_alarms[i + 1] - false_alarms[i]).days
    if len(false_alarms)==1:
        mean = 0
    else:
        mean = diff / (len(false_alarms) - 1)
    return (mean)
```

4- Probability on non-detection:

```
def performanceby_probability_of_nondetection(predictdata,actualdata,h):
    TP = 0
    f = False
    for i in actualdata :
        for j in predictdata:
            if abs((i-j).days) <= h :
                f=True
        if f==True:
            TP += 1
        f = False
    return (1-(TP/len(actualdata)))
```

Bibliography

[1] Sudeshna Adak. Time-dependent spectral analysis of nonstationary time series. *Journal of the American Statistical Association*, 93(444):1488–1501, 1998.

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

[3] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. Deep learning for stock prediction using numerical and textual information. In *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*, pages 1–6. IEEE, 2016.

[4] Merve Alanyali, Helen Susannah Moat, and Tobias Preis. Quantifying the relationship between financial news and the stock market. *Scientific reports*, 3:3578, 2013.

[5] Sylvain Arlot, Alain Celisse, and Zaıd Harchaoui. Kernel change-point detection. *arXiv preprint arXiv:1202.3878*, 6, 2012.

[6] Alexander Aue and Lajos Horváth. Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16, 2013.

[7] Marco Avellaneda and Sasha Stoikov. High-frequency trading in a limit order book. *Quantitative Finance*, 8(3):217–224, 2008.

[8] Jushan Bai. Least squares estimation of a shift in linear processes. *Journal of Time Series Analysis*, 15(5):453–472, 1994.

[9] Jushan Bai. Least absolute deviation estimation of a shift. *Econometric Theory*, 11(3):403–436, 1995.

[10] Jushan Bai. Testing for parameter constancy in linear regressions: an empirical distribution function approach. *Econometrica: Journal of the Econometric Society*, pages 597–622, 1996.

[11] Jushan Bai. Estimation of a change point in multiple regression models. *Review of Economics and Statistics*, 79(4):551–563, 1997.

[12] Jushan Bai. Common breaks in means and variances for panel data. *Journal of Econometrics*, 157(1):78–92, 2010.

[13] Jushan Bai and Pierre Perron. Computation and analysis of multiple structural change models. *Journal of applied econometrics*, 18(1):1–22, 2003.

[14] Lawrence Bardwell, Paul Fearnhead, et al. Bayesian detection of abnormal segments in multiple time series. *Bayesian Analysis*, 12(1):193–218, 2017.

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

[16] Michèle Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993.

[17] Lucien Birgé and Pascal Massart. Gaussian model selection. *Journal of the European Mathematical Society*, 3(3):203–268, 2001.

[18] David M Blei and Michael I Jordan. Variational methods for the dirichlet process. In *Proceedings of the twenty-first international conference on Machine learning*, page 12. ACM, 2004.

[19] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[20] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8, 2011.

[21] Jacob Boudoukh, Ronen Feldman, Shimon Kogan, and Matthew Richardson. Which news moves stock prices? a textual analysis. Technical report, National Bureau of Economic Research, 2013.

[22] Leif Boysen, Angela Kempe, Volkmar Liebscher, Axel Munk, Olaf Wittich, et al. Consistencies and rates of convergence of jump-penalized least squares estimators. *The Annals of Statistics*, 37(1):157–183, 2009.

[23] Michael Braun and Jon McAuliffe. Variational inference for large-scale models of discrete choice. *Journal of the American Statistical Association*, 105(489):324–335, 2010.

[24] Boris E Brodsky, Boris S Darkhovsky, Alexander Ya Kaplan, and Sergei L Shishkin. A nonparametric method for the segmentation of the eeg. *Computer methods and programs in biomedicine*, 60(2):93–106, 1999.

[25] Kevin Canini, Lei Shi, and Thomas Griffiths. Online inference of topics with latent dirichlet allocation. In *Artificial Intelligence and Statistics*, pages 65–72, 2009.

[26] Jie Chen and Arjun K Gupta. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical association*, 92(438):739–747, 1997.

[27] Jie Chen and Arjun K Gupta. *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*. Springer Science & Business Media, 2011.

[28] Scott Chen, Ponani Gopalakrishnan, et al. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *Proc. darpa broadcast news transcription and understanding workshop*, volume 8, pages 127–132. Virginia, USA, 1998.

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

[30] David Roxbee Cox. *Analysis of survival data*. Routledge, 2018.

[31] Frédéric Desobry, Manuel Davy, and Christian Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.

[32] Idris A Eckley, Paul Fearnhead, and Rebecca Killick. Analysis of changepoint models. *Bayesian Time Series Models*, pages 205–224, 2011.

[33] Paul Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and computing*, 16(2):203–213, 2006.

[34] William Feller. *An introduction to probability theory and its applications*, volume 1. Wiley, New York, 1968.

[35] Klaus Frick, Axel Munk, and Hannes Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580, 2014.

[36] Piotr Fryzlewicz. Unbalanced haar technique for nonparametric function estimation. *Journal of the American Statistical Association*, 102(480):1318–1327, 2007.

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

[38] Roman Garnett, Michael A Osborne, Steven Reece, Alex Rogers, and Stephen J Roberts. Sequential bayesian prediction in the presence of changepoints and faults. *The Computer Journal*, 53(9):1430–1446, 2010.

[39] Roman Garnett, Michael A Osborne, and Stephen J Roberts. Sequential bayesian prediction in the presence of changepoints. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 345–352. ACM, 2009.

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

[41] Wael H Gomaa and Aly A Fahmy. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18, 2013.

[42] Yann Guédon. Exploring the latent segmentation space for the assessment of multiple change-point models. *Computational Statistics*, 28(6):2641–2678, 2013.

[43] Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42. ACM, 1999.

[44] Fredrik Gustafsson and Fredrik Gustafsson. *Adaptive filtering and change detection*, volume 1. Citeseer, 2000.

[45] Ning Hao, Yue Selena Niu, and Heping Zhang. Multiple change-point detection via a screening and ranking algorithm. *Statistica Sinica*, 23(4):1553, 2013.

[46] Zaid Harchaoui and Olivier Cappé. Retrospective mutiple change-point estimation with kernels. In *Statistical Signal Processing, 2007. SSP'07. IEEE/SP 14th Workshop on*, pages 768–772. IEEE, 2007.

[47] Zaïd Harchaoui, Eric Moulines, and Francis R Bach. Kernel change-point analysis. In *Advances in neural information processing systems*, pages 609–616, 2009.

[48] Matthew Hoffman, Francis R Bach, and David M Blei. Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[49] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[50] Thomas Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR Forum*, volume 51, pages 211–218. ACM, 2017.

[51] Venkata Jandhyala, Stergios Fotopoulos, Ian MacNeill, and Pengyu Liu. Inference for single and multiple change-points in time series. *Journal of Time Series Analysis*, 34(4):423–446, 2013.

[52] Armand Joulin, Augustin Lefevre, Daniel Grunberg, and Jean-Philippe Bouchaud. Stock price jumps: news and volume play a minor role. *arXiv preprint arXiv:0803.1769*, 2008.

[53] Steven M Kay. Fundamentals of statistical signal processing, vol. ii: Detection theory. *Signal Processing. Upper Saddle River, NJ: Prentice Hall*, 1998.

[54] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 289–296. IEEE, 2001.

[55] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. In *Data mining in time series databases*, pages 1–21. World Scientific, 2004.

[56] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[57] Stanley IM Ko, Terence TL Chong, Pulak Ghosh, et al. Dirichlet process hidden markov multiple change-point model. *Bayesian Analysis*, 10(2):275–296, 2015.

[58] Rémi Lajugie, Francis Bach, and Sylvain Arlot. Large-margin metric learning for constrained partitioning problems. In *International Conference on Machine Learning*, pages 297–305, 2014.

[59] Hon Fai Lau and Shigeru Yamamoto. Bayesian online changepoint detection to improve transparency in human-machine interaction systems. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 3572–3577. IEEE, 2010.

[60] Marc Lavielle and Eric Moulines. Least-squares estimation of an unknown number of shifts in a time series. *Journal of time series analysis*, 21(1):33–59, 2000.

[61] Sangno Lee, Jaeki Song, and Yongjin Kim. An empirical comparison of four text mining methods. *Journal of Computer Information Systems*, 51(1):1–10, 2010.

[62] Xiaodong Li, Haoran Xie, Li Chen, Jianping Wang, and Xiaotie Deng. News impact on stock price return via sentiment analysis. *Knowledge-Based Systems*, 69:14–23, 2014.

[63] Henry H Liu. *Software performance and scalability: a quantitative approach*, volume 7. John Wiley & Sons, 2011.

[64] Anuj Mahajan, Lipika Dey, and Sk Mirajul Haque. Mining financial news for major events and their impacts on the market. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 1, pages 423–426. IEEE, 2008.

[65] Robert Maidstone, Paul Fearnhead, and Adam Letchford. Efficient analysis of complex changepoint models. 2012.

[66] Robert Maidstone, Toby Hocking, Guillem Rigaill, and Paul Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and computing*, 27(2):519–533, 2017.

[67] Burton G Malkiel. Efficient market hypothesis. In *Finance*, pages 127–134. Springer, 1989.

[68] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[69] M-A Mittermayer. Forecasting intraday stock price trends with text mining techniques. In *system sciences, 2004. proceedings of the 37th annual hawaii international conference on*, pages 10–pp. IEEE, 2004.

[70] Kevin P Murphy. Conjugate bayesian analysis of the gaussian distribution. *def*, $1(2\sigma 2)$:16, 2007.

[71] Thien Hai Nguyen and Kiyoaki Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1354–1364, 2015.

[72] Frank Nielsen and Vincent Garcia. Statistical exponential families: A digest with flash cards. *arXiv preprint arXiv:0911.4863*, 2009.

[73] Chong Oh and Olivia Sheng. Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement. In *Icis*, pages 1–19. Citeseer, 2011.

[74] Laurent Oudre, Alexandre Lung-Yut-Fong, and Pascal Bianchi. Segmentation of accelerometer signals recorded during continuous treadmill walking. In *Signal Processing Conference, 2011 19th European*, pages 1564–1568. IEEE, 2011.

[75] Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998.

[76] Vladimir Pavlovic, James M Rehg, and John MacCormick. Learning switching linear models of human motion. In *Advances in neural information processing systems*, pages 981–987, 2001.

[77] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577. ACM, 2008.

[78] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Elsevier, 1990.

[79] Stephen Roberts, Michael Osborne, Mark Ebden, Steven Reece, Neale Gibson, and Suzanne Aigrain. Gaussian processes for time-series modelling. *Phil. Trans. R. Soc. A*, 371(1984):20110550, 2013.

[80] D. Sarkar. *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from your Data*. Apress, 2016.

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

[82] Young-Woo Seo, Joseph Giampapa, and Katia Sycara. Text classification for intelligent portfolio management. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 2002.

[83] Padhraic Smyth, Max Welling, and Arthur U Asuncion. Asynchronous distributed learning of topic models. In *Advances in Neural Information Processing Systems*, pages 81–88, 2009.

[84] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[85] Bien Aik Tan, Peter Gerstoft, Caglar Yardim, and William S Hodgkiss. Change-point detection for recursive bayesian geoacoustic inversions. *The Journal of the Acoustical Society of America*, 137(4):1962–1970, 2015.

[86] James D Thomas and Katia Sycara. Gp and the predictive power of internet message traffic. In *Genetic Algorithms and Genetic Programming in Computational Finance*, pages 81–102. Springer, 2002.

[87] Charles Truong, Laurent Oudre, and Nicolas Vayatis. A review of change point detection methods. *arXiv preprint arXiv:1801.00718*, 2018.

[88] Clara Vega. Stock price reaction to public and private information. *Journal of Financial Economics*, 82(1):103–133, 2006.

[89] Boyi Xie, Rebecca J Passonneau, Leon Wu, and Germán G Creamer. Semantic frames to predict stock price movement. 2013.

[90] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528, 2003.

[91] Yi-Ching Yao et al. Estimating the number of change-points via schwarz'criterion. *Statistics & Probability Letters*, 6(3):181–189, 1988.