Check for updates

Identifying Shifts in Collective Attention to Topics on Social Media

Yuzi He, Ashwin Rao, Keith Burghardt(✉), and Kristina Lerman

USC Information Sciences Institute, Marina del Rey, CA 90292, USA
keithab@isi.edu

Abstract. The complex, ever-shifting landscape of social media can obscure important changes in conversations involving smaller groups. Discovering these subtle shifts in attention to topics can be challenging for algorithms attuned to global topic popularity. We present a novel unsupervised method to identify shifts in high-dimensional textual data. By utilizing a random selection of date-time instances as inflection points in discourse, the method automatically labels the data as before or after a change point and trains a classifier to predict these labels. Next, it fits a mathematical model of classification accuracy to all trial change points to infer the true change points, as well as the fraction of data affected (a proxy for detection confidence). Finally, it splits the data at the detected change and repeats recursively until a stopping criterion is reached. The method beats state-of-the-art change detection algorithms in accuracy, and often has lower time and space complexity. The method identifies meaningful changes in real-world settings, including Twitter conversations about the Covid-19 pandemic and stories posted on Reddit. The method opens new avenues for data-driven discovery due to its flexibility, accuracy and robustness in identifying changes in high dimensional data.

Keywords: Change point detection · Confusion · Reddit · Twitter · Attention shift detection

1 Introduction

The ever-growing volume of information in online social media creates a competition among content producers for the limited attention of content consumers [11]. When coupled with the news cycle of media reporting, the attention competition creates a highly dynamic information environment, where some topics are widely discussed and frequently re-shared, while others languish in obscurity. Changes in these patterns suggest important events, which have been detected with generative models, such as dynamic topic models [6]. These approaches, however, focus on the most popular topics and may miss the subtler changes and shifts in attention that occur in the heterogeneous and dynamic information ecosystem of social media. These shifting patterns of attention hold a clue to what disparate

R. Thomson et al. (Eds.): SBP-BRiMS 2021, LNCS 12720, pp. 224–234, 2021.
https://doi.org/10.1007/978-3-030-80387-2_22

communities find important and can help better explain the complex dynamics of the information ecosystem.

We propose a change detection method called *Meta Change Detection* (MtChD), to detect change points, such as shifts of attention to topics in online conversations. This method is in contrast to previous papers on emerging topics (c.f., [2]) by finding subtle changes in the words and sentences discussed. While a number of change detection methods have been developed (c.f., [21]), they do not work well with high-dimensional data, such as text, due to large number of fitting parameters or high memory usage, and few provide confidence about the quality of segmentation of data before and after the change. Another challenge is that social media data is both massive and sparse: many people participate, but most contribute relatively little text. This sparsity introduces noise into change point estimation because changes could be due to shifts in active users rather than topics, and we know rather little about each person individually.

To address these challenges, MtChD uses "confusion" [22] and a novel model to estimate both when the change occurs and the fraction of data affected by change for any number of changes in the dataset. Confusion attempts to *confuse* a model by labeling the same state (before or after the change) with respective labels, even when these labels may be wrong [22]. Significant changes in the accuracy of predicting these labels indicate a potential change. In contrast to previous work, however, we detect changes as differences in accuracy compared to a null model (predicting the majority class). The novel mathematical model for accuracy also estimates the amount of data changed. This acts as a confidence proxy, something not often used in previous methods: we are more confident in a change point if a large fraction of subsequent data changes. Also, in contrast to Nieuwenburg et al. [22], data is then split at this change point and the method is repeated recursively until an arbitrary stopping point, allowing us to detect any number of change points, as well as the degree of confidence we have in each change.

We validate MtChD by applying it to high-dimensional data sets, both synthetic and real-world, to demonstrate that it can detect changes with higher accuracy than [22] and other state-of-the-art baselines, even on sparse and noisy data with missing values. We apply the proposed method to large public datasets of tweets about the Covid-19 pandemic [7] and scary stories posted in the nosleep subreddit (www.reddit.com/r/nosleep). We show that MtChD accurately infers meaningful shifts within Twitter conversations and Reddit stories.

The rest of the paper is organized as follows. We first review related work, and then present details of the confusion-based change detection method including a mathematical model that quantifies these changes. Finally, we present results for changes in synthetic and real datasets and discuss implications.

2 Related Works

Change detection, especially when the number of change points is unknown, usually involves three steps: choosing a cost function on how homogenous data is between change points, determining the change search method, and constraining the number of change points [21]. There is a wide variety of costs functions, from parametric (e.g., maximum likelihood of parametric distributions) to non-parametric (e.g., non-parametric maximum likelihood or kernel-based methods). Parametric methods started with CUMSUM [16], which is used to detect changes in the mean of data with a normal distribution, and have been advanced on with methods such as generalized likelihood ratio (GLR) test [20] and extensions of this technique (c.f., [4,21]). The GLR test seeks to reject a null assumption which states that observations before and after a proposed change point t_0 follow the same distribution, therefore when the null assumption is rejected, a change probably occurs. Parametric cost functions, however, often suffer from the curse of dimensionality and require data to follow a particular distribution, which does not necessarily hold in general. While semi-parametric [5] and non-parametric methods, such as non-parametric maximum likelihood estimation, have been developed, they can be computationally expensive on large or high-dimensional data [21]. Kernel methods have instead been introduced to describe distributions that are difficult to parameterize [3]. Finally, there are some methods that do not easily fit into these categories, such as hidden Markov models (HMM) [18] and Bayesian change detection methods [1,23].

A method that is similar to our change detection method is by Nieuwenburg et al. [22], in which a *confusion scheme* is used to distinguish phases in a physical system. More specifically, the classifier identifies the critical point $c \in (a, b)$ by testing candidate critical points $c' \in C'$. When c' is between (a, c), the classifier correctly labels data between (a, c') and but incorrectly labels data between (c', c) with a different label. We take advantage of this unexpected similarity between changes and phases to create MtChD, which also uses confusion, but is more robust to change points occurring near the edges of data streams. Furthermore, Nieuwenburg et al., [22] do not attempt to discover multiple phase changes (i.e., change points), while the present method can. Another method similar to ours is called Unsupervised Change Analysis [10] which also creates change labels. The paper, however, focuses on explaining changes and not on finding changes.

We improve on these previous methods in several critical ways. First, we offer an agnostic framework to detect change points through the accuracy variation of an arbitrary classifier. This can significantly reduce the time and space complexity of the method compared to using cost functions like kernels or search methods like dynamic programming [21], and the method is flexible to the type of data that it analyzes. By fitting classifier accuracy to a model, we also avoid significant costs of analyzing high-dimensional data. Due to memory complexity, MtChD can easily handle data with several thousand dimensions, while GLR or Kernel based methods can struggle with a few dozen. Moreover, the model can estimate how much of the data was modified after a change, which acts as a change confidence proxy. Finally, our method can outperform baseline methods with near-linear time complexity and near-constant memory complexity, which is a significant improvement over the competing methods.

3 Methods and Materials

Problem Statement. Assume we have data of the form $(X_i, t_i), i = 1, \ldots, n$, where X is an arbitrarily high dimensional vector and t is an external control parameter such as time. We refer to t as the *indicator* and look for a change point in indicator t. Assume there is a change at t_0 such that some data before the change and some after the change have different distributions. In many datasets, however, only a fraction of data, $0 \leq \alpha \leq 1$, may show observable changes. *Our goal is to infer the change point,* t_0*, and the fraction of data that undergoes a change,* α*, given the observations* (X_i, t_i)*.* For clarity, just as t_0 varies with each change point, so does α.

Step 1: Confusion-Based Training. Similar to [22], we assume a trial change point $t = t_a$ and label the observed data before t_a as belonging to class $\tilde{y}_i = 0$ (no change), and the data after t_a as class $\tilde{y}_i = 1$ (change). We then train a classifier to predict the labels $\tilde{y}_i$ from the features X_i, of an arbitrary number of dimensions. We plot the accuracy of the classifier as a function of trial change point t_a. In case a true change point exists in the observed range of t, the accuracy vs. t_a curve will significantly increase over the baseline accuracy, which is the majority class ratio of labels $\tilde{y}_i$. The shape of the curve will be affected both by the actual change point, t_0, and the fraction of data points affected by change, α. The classifier one can use could be anything – we use random forest and multi-layer perceptrons as examples in this paper. For each candidate change point, t_a, classifiers are trained on a random subsample of 50% of data, validated on 30%, and tested on 20%. The test set is used to judge the accuracy of the learned models for each t_a. This step is known as confusion-based training.

Accuracy as a function of t_a varies significantly. Near the beginning and end of the dataset, accuracy is nearly 1 because we can almost certainly say that data is before (if $t_a \ll t_0$) or after (if $t_a \gg t_0$) a change point. More specifically, if we have a candidate critical point near the beginning or end of the data, almost all of the data would be after or before the critical point respectively. Accuracy predictably decreases away from the extremes, but, peaks around the true change point, thus forming a W shape [22].

Step 2: Modeling Acc. vs. t_a Curve. The novelty of our work is to model this accuracy curve in order to infer t_0 and α. This natural extension of the previous work provides substantial improvements in change point estimation. We first define the cumulative distribution of t, $F(t) = 1/T \sum_i t_i < t$. Data can fall into three categories (or three distinguishable distributions), a distribution that does not change, S_u, which comprises $1 - \alpha$ of all data, a distribution before a change ($t \leq t_0$), S_0, and a distribution after this change ($t > t_0$), S_1. We do not know these distributions *a priori* but we assume the trained classifier will be able to distinguish these distributions using data X.

Assume that the distribution of t is independent of the discrete distribution that a data point belongs to a distribution that does not change after the change point, S_u, or changes from a distribution S_0 to a distribution S_1.

For clarity, we never have to know what distribution each data point lies in, nor even what the distribution looks like. For confusion, recall we label data as 0 if $t' \leq t$ and 1 otherwise. Given candidate change point t_a, $P_{S_{u,0}} = (1-\alpha)F(t_a)$ of data in S_u is labeled 0 and $P_{S_{u,1}} = (1-\alpha) - P_{S_{u,0}}$ is labeled 1. On top of this, for a data point in S_u, the expected predicting accuracy should be $\frac{1}{1-\alpha}\max(P_{S_{u,0}}, P_{S_{u,1}})$. Similarly, we can calculate the ratio of data labeled as 0 or 1 in S_0 and S_1. With real change point locate at t_0, given any t, we assume that among α fraction of data affected by change, $\theta(t-t_0)$ fraction of data belongs to S_1 and $1-\theta(t-t_0)$ fraction of data belongs to S_0. Here $\theta(\cdot)$ is the Heaviside step function. We can calculate for S_1, which has fraction $\alpha(1-F(t_0))$, $P_{S_{1,1}} = \max(\alpha(F(t_a)-F(t_0)), 0)$ and $P_{S_{1,0}} = \alpha(1-F(t_0)) - P_{S_{0,1}}$. The expected predicting accuracy for S_1 is thus $\frac{1}{\alpha(1-F(t_0))}\max(P_{S_{1,0}}, P_{S_{1,1}})$. Finally, S_0 has a fraction of $\alpha F(t_0)$. The total fraction of data labeled "0" in S_0 and S_1 is $\alpha F(t_a)$, $P_{S_{0,0}} = \alpha F(t_a) - P_{S_{1,0}}$, therefore the proportion of S_0 incorrectly labeled "1" is $P_{S_{0,1}} = \alpha F(t_0) - P_{S_{0,0}}$. The expected predicting accuracy for data point in S_0 is then $\frac{1}{\alpha F(t_0)}\max(P_{S_{0,0}}, P_{S_{0,1}})$.

We then leverage the results above to estimate the accuracy curve as $\tilde{A}cc(t_a) = \max(P_{S_{u,0}}, P_{S_{u,1}}) + \max(P_{S_{0,0}}, P_{S_{0,1}}) + \max(P_{S_{1,0}}, P_{S_{1,1}})$. These variables only depend on $F(t)$, which can be directly estimated from data, and the free parameters t_0 and α. We therefore do not need explicit knowledge of distributions S_0, S_1 and S_u. To estimate t_0 and α, we can do a fast grid search such that the squared difference between the estimated and actual accuracy is minimized, and find data closely aligns with this model.

Multiple Changes. To identify multiple changes, we use recursive binary splitting. We first use the change detection method to find a change point, and split at this point. This, in turn, creates two subsets of data from which we can find additional changes, and split this data, in recursion. We stop splitting a node when we hit the minimum length of range t_c or maximum depth of the binary tree D. The time complexity is $O(TD)$ for binary segmentation depth, D, and number of data points, T. Because D is fixed to a small value, such as 3, the splitting process is almost linear in time. The space complexity only depends on the classifier used, so it can be efficient even in high-dimensional datasets. Relevant code pertaining to our analysis has been made publicly available through GitHub[1].

4 Data

Online Discussions About Covid-19. We apply our method to a large dataset of Covid-19 tweets [7]. This dataset consists of 115M tweets from users across the globe, collected since January 21, 2020. These tweets contain at least one of a predetermined set of Covid-19-related keywords (e.g., coronavirus, pandemic, Wuhan, etc.). Since this dataset provides geolocation data for only 1% of the users, we leverage a fuzzy matching approach [12] to geolocate users within

[1] https://github.com/yuziheusc/confusion_multi_change.

the US. We want to understand the significant shifts in attention during the earliest era of Covid-19, from January 21 until March 31, 2020, of which 7.6 million tweets are geo-located to within the US using methods by Chen et al., [7]. We then subsample 200K tweets at random each month, for a total of 600K tweets, to simplify our analysis. Text is pre-processed through removal of stopwords, links, account names, and special characters (e.g., !?%#). Only English language tweets are considered. We then use the tf-idf vectorizer (with 2.2K terms) from Python's *scikit-learn* library [17] in order to generate the tf-idf vectors.

Reddit Stories. We extract Reddit posts from a popular horror story writing subreddit called *nosleep* using the Python Reddit API Wrapper (PRAW). We focus on posts created between January 1, 2019 and June, 2020 to understand both seasonal changes in stories (e.g., Halloween and Christmas), as well as changes in stories since the Covid-19 pandemic, creating 35.4K stories. Data pre-processing includes removing posts labeled "[removed]" and "[deleted]". Text cleaning and tf-idf vectorization (with 25K terms) follow the same methodology as in the Twitter dataset.

5 Experiments and Results

Synthetic Data. To test the change detection method, we generate two-dimensional data located in a unit box in a chessboard pattern with $n_c \times n_c$ squares, where n_c is a tunable parameter, as shown in the left panel of Fig. 1. These data are uniformly distributed in red squares when $t \leq t_0$, and green squares for $t > t_0$, with $0 \leq t \leq 1$. Our motivation for this synthetic example is that as n_c increases, it becomes harder to distinguish the change in data. Therefore, the data is not meant to be realistic but simple to construct with a change detection task difficulty controlled by n_c. For first part of this experiment, we set $t_0 = 0.5$, the size of the data $N = 8K$, and n_c equal to either 2, 6, or 10. For second part of this experiment, we fix $n_c = 6$ and we vary t_0 between 0.2 and 0.8. If t_0 differs from 0.5, the population before and after the change will be unbalanced, which makes the task of inferring t_0 more challenging [14].

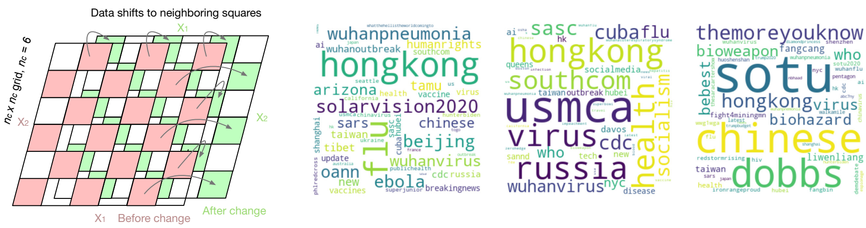

Fig. 1. The figure on the left illustrations of synthetic data, where observations have two features x_1 and x_2. Data lies in red squares before changes at t_0 and moves to green squares after t_0. In the illustration the data is an $n_c \times n_c$ grid, with $n_c = 6$; The 3 figures on the right shows word cloud for Covid-19 tweets in period 01/21–01/30, 01/30–02/04 and 02/04–02/11. (Color figure online)

We ran six trials for our method and competing algorithms. For MtChD and vanilla confusion [22], 50% of the data is used as training, 30% for validation, and 20% for testing, where training and validation data is randomly sampled during each trial. Validation data is used to tune hyperparameters of the classifiers used. In competing methods, we randomly sample 70% of data in each trial, except for Bayesian change detection, where 18.8% of data (around 1.5K) is sampled due to computational limitations (this method takes much longer than other competing methods).

The methods tested are as follows: the vanilla confusion method of Nieuwenburg et al. [22], GLR, dynamic programming segmentation (DP) with different loss kernels [19], and Bayesian detection with different priors and likelihood functions [1]. While alternative segmentation methods can be used besides DP, such as binary segmentation [8], bottom up methods [13] and window based methods [21], DP is found to outperform these alternatives (not shown). For Bayesian change point detection, a conditional prior on change point and a likelihood function needs to be defined. We used uniform and geometric distributions as priors, and applied the Gaussian, individual feature model [23], and full covariance model [23] as likelihood functions.

The results are shown in Table 1. We see that for $n_c = 2$, optimal segmentation methods (*DP+RBF*) perform as well as ours. Vanilla confusion performs well when change point is in the center of data ($t_0 = 0.5$). Otherwise our method outperforms competing methods, especially when the change point is different from 0.5. Of the two classifiers used by MtChD, random forest performs best.

Online Discussions About Covid-19. Now that we show our method works well for single-change detection, we move on to more complicated multi-change detection in empirical data, with results shown in Table 3. We start by identifying shifts in tweets about Covid-19 (embedded into tf-idf vectors), where the word cloud of hashtags for the first three periods are shown in the right panels of Fig. 1. We ran binary segmentation using MtChD with a Random Forest classifier, maximum segmentation depth of three and minimum time length between changes set to four days. The dates of the change points identified by the method are listed in Table 3 . The time intervals between changes match with the period of a typical news cycle [15], which is between 5 to 9 days. Results are robust in the way that when the minimum length is increased to of 5 days, a subset of changes (01/30, 02/11, 02/16, 02/21 and 02/28) are found. Next, we analyze the discovered change points and interpret the findings by highlighting topics that shift the collective attention. To validate the results, we compare the change points found with the news events, as shown in Table 2.

Table 1. Comparison of performance of MtChD and competing methods. The entries show the mean change point $\mu(t_0)$ in comparison to the actual change point t_0. *(RF)* and *(MLP)* correspond to using a random forest and multilayer perceptron classifiers for confusion. Row 3 and row 4 show performance of naive confusion and GLR test. Row 5 to row 7 shows performance of DP, a non-approximate segmentation method, while other segmentation methods that perform worse are not shown (see main text). The cost functions used are *RBF* (RBF kernel), *L1* (L_1 loss function), and *L2* (L_2 loss function). The last four rows are for Bayesian change detection with a *uniform* prior or *Geo* (geometric) prior. *Gassusian* stands for Gaussian likelihood function, *IFM* is the individual feature model and *FullCov* is the full covariance model. Bold values indicate change points that are closest to the correct value.

		2×2 $t_0 = 0.5$	10×10 $t_0 = 0.5$	6×6 $t_0 = 0.2$	6×6 $t_0 = 0.8$
MtChD (RF)	$\mu(t_0)$ $\mu(\alpha)$	$\mathbf{0.500 \pm 0.003}$ $\mathbf{0.949 \pm 0.008}$	$\mathbf{0.496 \pm 0.005}$ $\mathbf{0.66 \pm 0.02}$	$\mathbf{0.195 \pm 0.005}$ $\mathbf{0.65 \pm 0.03}$	$\mathbf{0.802 \pm 0.002}$ $\mathbf{0.66 \pm 0.03}$
MtChD (MLP)	$\mu(t_0)$ $\mu(\alpha)$	$\mathbf{0.503 \pm 0.003}$ $\mathbf{0.96 \pm 0.01}$	0.58 ± 0.06 0.009 ± 0.008	0.56 ± 0.05 0.005 ± 0.004	0.5 ± 0.1 0.02 ± 0.02
Confusion (RF)	$\mu(t_0)$	$\mathbf{0.497 \pm 0.002}$	$\mathbf{0.4973 \pm 1E{-}4}$	0.23 ± 0.04	0.54 ± 0.09
GLR	$\mu(t_0)$	$\mathbf{0.5003 \pm 4E{-}4}$	0.6 ± 0.3	0.24 ± 0.04	0.81 ± 0.03
DP+RBF	$\mu(t_0)$	$\mathbf{0.5002 \pm 4E{-}4}$	0.3 ± 0.2	0.4 ± 0.3	0.84 ± 0.07
DP+L2	$\mu(t_0)$	0.95 ± 0.01	0.5 ± 0.4	0.4 ± 0.3	0.3 ± 0.4
DP+L1	$\mu(t_0)$	0.957 ± 0.007	0.4 ± 0.3	0.6 ± 0.4	0.2 ± 0.3
Uniform+Gaussian	$\mu(t_0)$	0.5 ± 0.2	0.5 ± 0.2	0.6 ± 0.3	0.5 ± 0.3
Uniform+IFM	$\mu(t_0)$	0.997 ± 0.003	0.998 ± 0.003	0.999 ± 0.002	0.999 ± 0.001
Uniform+FullCov	$\mu(t_0)$	$\mathbf{0.4985 \pm 2E{-}4}$	$0.9989 \pm 9E{-}4$	0.99 ± 0.01	0.997 ± 0.004
Geo+Gaussian	$\mu(t_0)$	0.028 ± 0.004	0.028 ± 0.004	0.033 ± 0.006	0.025 ± 0.004

Reddit Stories. We also applied our method to horror stories posted on reddit.com, the subreddit *r/nosleep*, with stories embedded using tf-idf. We find variations in the topics of stories, such as Jul 17 to September 25, 2019 ("camping" and "summer") appear, reflecting recreation activities in the US. The next change on September 25 to November 17 ("halloween") signals the topic of Halloween and November 17 to January 2nd ("santa" and "christmas"), corresponds to the holidays. Potentially inspired by Covid-19 restrictions, there were stories about "quarantine" from March 29 to May 4, 2020. Finally, quarantining became old news again, and discussions shifted in the final months until June 2020 back to stories on "rules". As a baseline, we used GLR and DP with an RBF kernel. Due to the limitations of memory, we first perform truncated SVD [9] to transform the tf-idf vector into a 64-dimensional vector. Then we down-sampled to 8K observations from the full dataset since dynamic programming runs in $O(T^2)$. We find that not only is our method able to process the full dataset, it can find more physically meaningful change points.

Table 2. Change points automatically identified in Covid-19 tweets and important events occurring on those dates.

Change point		*Events*
Date	α	
01–30	0.355	First confirmed case of person-to-person transmission of the "Wuhan Virus" in the US
02–04	0.341	Diamond Princess cruise ship quarantined. Ten people on cruise ship near Tokyo have virus
02–11	0.327	WHO announced official name for "COVID-19"
02–16	0.243	More than 300 passengers from the Diamond Princess are traveling in the US chartered planes
02–21	0.441	1^{st} Covid-19 death in Italy (02–22)
02–28	0.366	First Covid-19 death in US (02–29)
03–04	0.447	California declares state of emergency. South Korea confirms 3 new deaths and 438 additional cases of novel coronavirus
03–09	0.269	Italy lockdown; Grand Princess cruise ship docks in Oakland
03–15	0.303	First lockdown orders in parts of California; national emergency declared (3/13)
03–24	0.146	US sees deadliest day with 160 deaths

Table 3. Comparison with baseline change detection. (Left) Tweets and (right) r/nosleep.

Covid-19 tweets			Reddit stories		
Our result	GLR	DP+RBF	Our result	GLR	DP+RBF
01-30-20	02-07-20	01-27-20	03-10-19	03-26-19	04-10-19
02-04-20	02-08-20	01-28-20	06-05-19	06-03-19	04-12-19
02-11-20	02-08-20	01-31-20	07-17-19	08-11-19	11-06-19
02-16-20	02-08-20	02-13-20	09-25-19	11-05-19	01-13-20
02-21-20	02-09-20	02-15-20	11-17-19	12-20-19	01-29-20
02-28-20	02-09-20	02-26-20	01-02-20	01-30-20	02-19-20
03-04-20	02-17-20	02-29-20	02-21-20	03-02-20	03-10-20
03-09-20	02-17-20	03-02-20	03-29-20	04-03-20	03-31-20
03-15-20	02-17-20	03-07-20	05-24-20	04-09-20	04-07-20
03-24-20	02-27-20	03-13-20			

6 Conclusions

In this paper, we aim to identify and understand the shifts of conversation on social media. In contrast to emergent topic detection, which detects new topics of interest, our method identifies when the distribution of features in high-dimensional streams of text changes. We create a method to robustly detect multiple changes within these conversations which appear to represent intuitive and realistic changes in conversations. Moreover, quantitative and qualitative comparisons to baseline methods show improved detection of changes. Our method has a unique feature – it allows us to quantify the fraction (parameter α) of data which shows observable changes. This parameter can be interpret as the significance of a certain change.

There are, however, important limitations of our approach. First, multiple changes are found with a simple binary segmentation, which is only meant to find approximate change points [21]. While this allows us to dramatically speed up computation, it may compromise on accuracy. Next, the social media data we explore has no ground truth about changes except for daily news. So we cannot assess whether our method, or competing methods, correctly found all change points. This may affect conclusions about what are the most important changes within social media early in the Covid-19 pandemic. A more detailed analysis involving tweets from different languages and from across the globe would be a promising candidate for future research.

These limitations, however, point to promising future work. For example, it will be important to explore advancing on the binary segmentation approach in order to sacrifice some potential speed for greater accuracy or precision. Next, we should compare against realistic data with a fixed number of known change points to determine the overall accuracy of this method. Finally, these results should be extended to other high-dimensional datasets, including video.

Acknowledgments. This work was funded in part by DARPA (W911NF-17-C-0094 and HR00111990114) and AFOSR (FA9550-20-1-0224).

References

1. Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection. arXiv preprint arXiv:0710.3742 (2007)
2. Alkhodair, S.A., Ding, S.H., Fung, B.C., Liu, J.: Detecting breaking news rumors of emerging topics in social media. Inf. Process. Manag. **57**(2), 102018 (2020)
3. Arlot, S., Celisse, A., Harchaoui, Z.: A kernel multiple change-point algorithm via model selection. JMRL **20**(162), 1–56 (2019)
4. Barber, J.: A generalized likelihood ratio test for coherent change detection in polarimetric SAR. IEEE GRSL **12**(9), 1873–1877 (2015)
5. Bardet, J.M., Kengne, W.C., Wintenberger, O.: Detecting multiple change-points in general causal time series using penalized quasi-likelihood. arXiv preprint: arXiv:1008.0054 (2010)
6. Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: ICML, pp. 113–120 (2006)
7. Chen, E., Lerman, K., Ferrara, E.: Tracking social media discourse about the COVID-19 pandemic: development of a public coronavirus Twitter data set. JPHS **6**(2), e19273 (2020)
8. Fryzlewicz, P., et al.: Wild binary segmentation for multiple change-point detection. Ann. Stat. **42**(6), 2243–2281 (2014)

9. Halko, N.: Finding structure with randomness: stochastic algorithms for constructing approximate matrix decompositions. arXiv:0909.4061 (2009)
10. Hido, S., Idé, T., Kashima, H., Kubo, H., Matsuzawa, H.: Unsupervised change analysis using supervised learning. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 148–159. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68125-0_15
11. Hodas, N.O., Lerman, K.: How limited visibility and divided attention constrain social contagion. In: SocialCom 2012 (2012)
12. Jiang, J., Chen, E., Yan, S., Lerman, K., Ferrara, E.: Political polarization drives online conversations about COVID-19 in the United States. Hum. Behav. Emerg. Technol. **2**, 200–211 (2020)
13. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. In: ICDM, pp. 289–296. IEEE (2001)
14. Leichtle, T., Geith, C., Lakes, T., Taubenböck, H.: Class imbalance in unsupervised change detection: a diagnostic analysis from urban remote sensing. Int. J. Appl. Earth Obs. Geoinf. **60**, 83–98 (2017)
15. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: KDD, pp. 497–506 (2009)
16. Page, E.S.: Continuous inspection schemes. Biometrika **41**(1–2), 100–115 (1954)
17. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
18. Raghavan, V., Galstyan, A., Tartakovsky, A.G.: Hidden Markov models for the activity profile of terrorist groups. Ann. Appl. Stat. **7**(4), 2402–2430 (2013)
19. Rigaill, G.: A pruned dynamic programming algorithm to recover the best segmentations with 1 to K_max change-points. J. de la Société Française de Stat. **156**(4), 180–205 (2015)
20. Siegmund, D., Venkatraman, E.: Using the generalized likelihood ratio statistic for sequential detection of a change-point. Ann. Stat. **23**(1), 255–271 (1995)
21. Truong, C., Oudre, L., Vayatis, N.: Selective review of offline change point detection methods. Sig. Process. **167**, 107299 (2020)
22. Van Nieuwenburg, E.P., Liu, Y.H., Huber, S.D.: Learning phase transitions by confusion. Nat. Phys. **13**(5), 435–439 (2017)
23. Xuan, X., Murphy, K.: Modeling changing dependency structure in multivariate time series. In: IMLS, pp. 1055–1062 (2007)