A Systematic Evaluation of Change Point Detection on Social Media Data

Fangming(Will) Liao

Abstract. Detecting changes in social media significantly help us track the sub-events through a major event. Currently, there are various change point detection (CPD) statistical models, and we focus some of them in our study. Experiments were carried out on both simulated data and Tweeter data from 2016 UEFA European Championship. Specific performances of algorithms under different experimental conditions are analyzed.

1. Introduction

Nowadays, there are dramatically increasing information and data flowing over the Internet, especially some small text documents such as micro-blogs and Tweeter. Change point problems are about to detect the turning points in the information extracted from these documents as immediate and accurate as possible, and these detections can be referred to unforeseen events and help us with identifying them. Currently there are various change point detection (CPD) methods that were published with source code in multiple programming languages, each with different properties and performances under diverse circumstances. In our study, we focus on Bayesian Change Point Analysis (bcp), Non-Parametric Multiple Change-Point Analysis (ecp), BreakoutDetection (breakout), Bayesian Online Changepoint Detection (oCPD) and Bayesian Online Changepoint Detection plus trend (otCPD). The change point analysis was performed and compared among different algorithms using both simulated data and real data problem. As the experiment conditions are oriented-controlled, specific performance under certain conditions of algorithms is expected.

2. Change Point Detection

2.1. bcp

The Bayesian Analysis for change point problems was based on product partition models, which assumes the probability of any partition is proportional to a product of prior cohesions, one for each block in the partition, and that given the blocks the parameters in different blocks have independent prior distributions. [1] Before 2014, the method was designed to perform Bayesian change point analysis on univariate time series. The

R package "bcp" publishes in July 2015 allows estimation of change point models with multivariate responses and returns the posterior probability of a change point occurring at each time index in the series. Essentially, the package was only designed to detect changes in the mean of independent Gaussian observations, which has following assumptions:
- There is an underlying sequence of parameters partitioned into contiguous blocks of equal parameter values. [1]
- Observations are independent in different blocks given the sequence of parameters. [1]
- The normal error assumption could be replaced by any other parametric assumption and similar analyses carried through. [1]
- In the multivariate case, a common change point structure, means are constant within each block of each sequence, but may differ across sequences within a given block. [2]
- Observations are independent, identically distributed normal, with constant means within blocks and constant variance throughout each sequence. [2]
- In linear regression analysis, the observations (x,y), where x may be multivariate, are partitioned into blocks, and that linear models are appropriate within each block. [2]

2.2. ecp

Different from bcp, the R package of ecp does not assume the observations to be normally distributed, and it solves many of the limitation of the current available change point packages. The advantage of ecp is that it's able to perform multiple change point analysis for both univariate and multivariate time series, making as few assumptions as possible. The only assumptions placed on distributions are that the absolute αth moment exists, for some $\alpha \in (0, 2]$, and that observations are independent over time. [3]

2.3. breakout

The method breakout was a novel statistical technique to automatically detect breakouts in cloud data, employing Energy Statistics to detect both breakouts in both application as well as system metrics. [4]
Since the underlying algorithm is referred to as E-Divisive with Medians(EDM), which is non-parametric, it has less limitations than bcp, and might work better in some practical cases. The R package is available at https://github.com/twitter/BreakoutDetection. Nevertheless, this algorithm only works on univariate data.

2.4. Online CPD

Online CPD (oCPD), namely Bayesian Online changepoint detection, is method that examines the case where the model parameters before and after the change point are independent and an online algorithm was derived for exact inference of the most recent changepoint. [5] The assumption that a sequence of observations $x_1, x_2, ..., x_T$ may be

divided into non-overlapping product partition was made and furthermore, the data with in each partition ρ are i.i.d. from some probability distribution $P(x_t|\eta_\rho)$ are assumed as well, where the parameter $\eta_\rho, \rho = 1, 2, ...$ are taken to be i.i.d.. [5]
Comparing to bcp and ecp, one of oCPD's limitations is that it cannot adjust detections according to known number of changepoint references on offline algorithm. To improve this, we enable oCPD to filter N most likely detections, given the number of reference N, by selecting top N detections with highest posterior probabilities.
Bases on oCPD, a method that takes consider of linear trend was developed, which is called Online CPD + trend. This algorithm can be more useful in many practical cases since commonly there is a trend in practice. The detection selection by number of reference was also added to this method.

3. Experiments

3.1. Datasets

3.1.1. Simulations First and foremost, we simulated several datasets, five of which are representative for typical circumstances. And they are:
1. Simple case: where there is only one sequence of data, divided into 4 periods of Gaussian distributions with different constant means and variances.
2. Multivariate case1: multivariate with change in covariance, where there are 3 sequences of data with 3 periods, and they are all normal distribution with same mean and variance. But there is a high correlation between each two of the sequence in the second period, while no correlation in the first and third.
3. Multivariate case2: multivariate with change in variance, where there are 3 sequences of data with 3 periods, and they are all normal distributed with same mean but a higher variance on the second period, and no correlation.
4. Linear Breaks: where there are 3 discrete normally distributed linear sequences of 3 different slopes and intercept with the same variance.
5. Linear Trend: where there is a continuous curve of 4 different normally distributed linear sequence with different mean and variance.

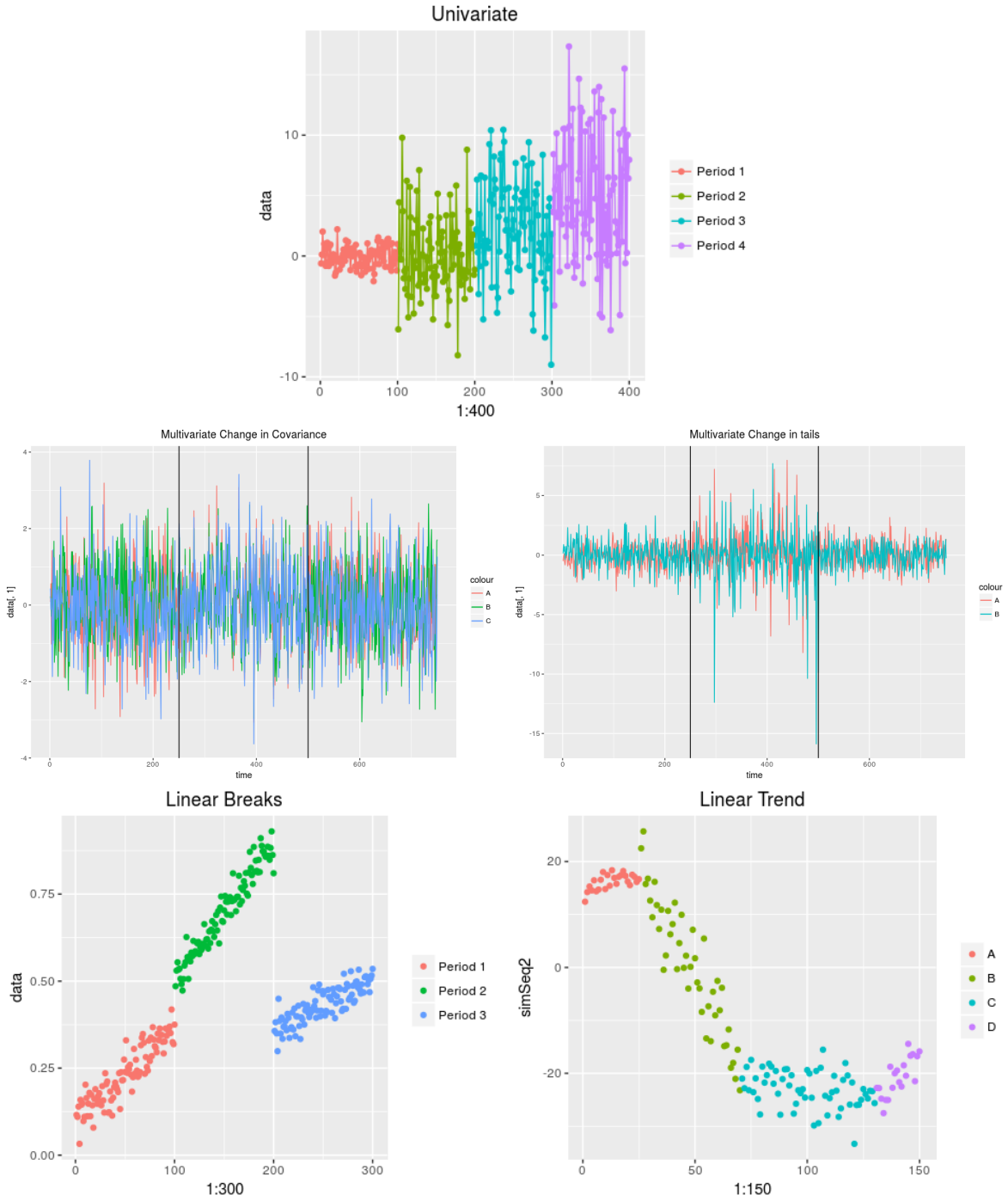

Figure 1: Simulated data: Univariate with four parts (top); multivariate 1; multivariate 2; linear breaks and linear trends.

3.1.2. European Championship For the practical data analysis, we collected Tweeter data of 18 of the 2016 UEFA European Championship from Tweeter API, and each tweet was annotated with sentiment and emotion probability by NRC.

The attribute annotated to the original data used sentiment from emotions for each Tweet, which was a probability distribution of three emotion levels: positive, neutral

and negative. Sentiment was introduced as an attributes that can give us more information than number of tweets only, so that we could more accurately detect the change points in some cases. The original data contains both English and non-English Tweets, in our study only English Tweets were used for both count and sentiment computation.

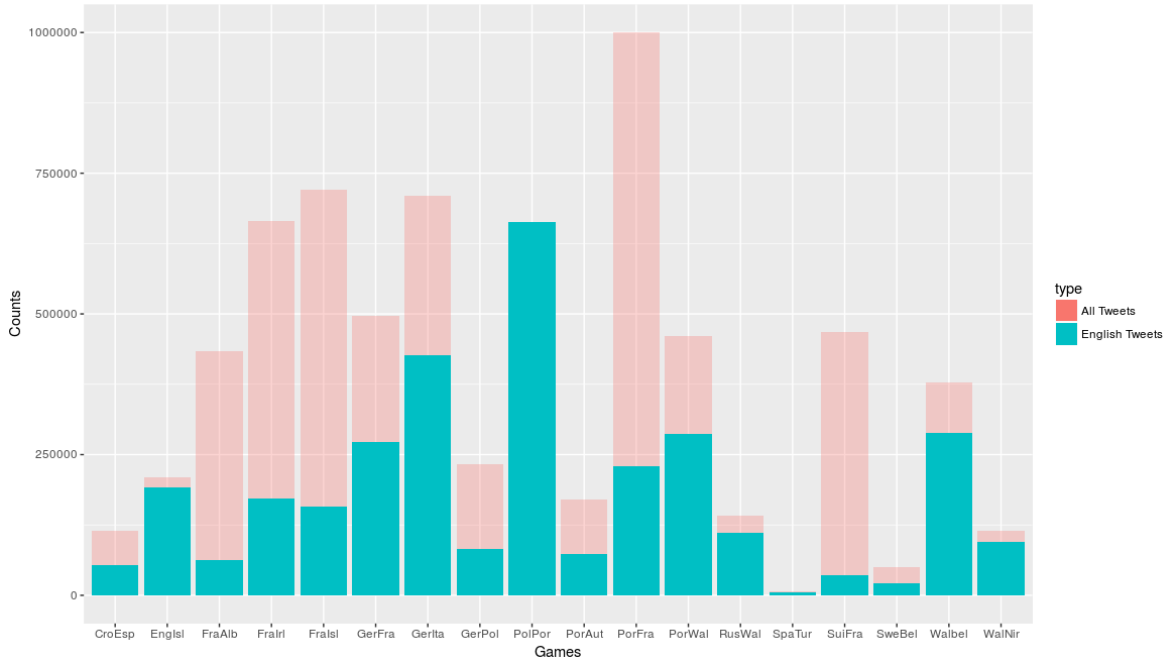

Figure 2: EuroGames data collection: basic statistics.

All tweets were divided into buckets according to different time interval, counts and sentiments scores were computed as sum and average respectively within each time interval of 15 sec, 30 sec and 60 sec.

In addition, streaming data was employed to simulate real large on-line data streams. On Euro Games, stepwise experiments were carried out with 30 seconds time interval, on two experimental conditions, one with CPD in a 50 minutes time interval and 25 minutes for each step, with a 25 minutes overlapping interval while making each step; and another with 30 minutes per interval and 5 minutes per step.

Throughout all 18 games, we compared the performance between each two attributes and intended to see how many time each method won.

The original annotated data was in JSON file, loaded and processed using R programming language.

3.2. Evaluation

The performances of change point detection methods were measured by comparing true detections with known event references, using the precision, recall and F-scores [6], which are computed by following:

$$Precision = \frac{TrueDetections}{AllDetections}$$

$$Recall = \frac{TrueDetections}{References}$$

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

To determine the number of true detections, the algorithm used to find the minimum distances between detections and reference, and a detection would be considered as a true detection if that distance is smaller than the tolerance time window; however, in some cases this method can miscalculate the number of true detections. For example, if we have two references at 50 and 60, and two detections at 57 and 65, given a time window of 8, by previous method 57 would be counted as a true detection as the distance between 57 and 60 is 3, but 65 wouldn't be counted. In this case, 57 apparently should be considered as a true detection to 50 and 65 is to 60. Therefore, it is more reasonable to list all possible true detections within the tolerance time window, then we find the earliest detection that can be considered to the first reference, so on so forth, which make sure we don't miss any reference that was detected.

For of all evaluations, bcp was run on each data for 50 times and the average precision, recall, F-score were recorded.

All experiments were carried out the twice with given the algorithms the number of references and not.

On the simulated data, a symmetric tolerance time window of 5 units is applied, i.e. a detection would be consider as a true-positive if $|\Delta t| \leq 5$, where Δt is the distance

between the detection and reference. Meanwhile, a symmetric tolerance time window of 3 minutes is applied on EuroGames.

As for the stepwise streaming data, detections in separate batches are cumulated together, then overall performances were computed after removing duplicated detections. To be consistent with practical situations, no number of references is given to the algorithm.

4. Results

From the simulated data, bcp performs up to its purpose and assumptions. As bcp is designed to detect the change in the means, it fails to detect the first change point in the simple case where the distribution of data changes from $N(0,1)$ to $N(0,3)$, it also mistakenly detects both of the change points in the multivariate case2 where only the variance of data changes.

Besides, bcp has a low tolerance of variance. When the number of references is not given, it missed the third change point which is in between of $N(2,4)$ and $N(4,5)$, where the change in mean is even less than variance on both sides, it also made many false detections on the second multivariate case where there is a large variance on the second period.

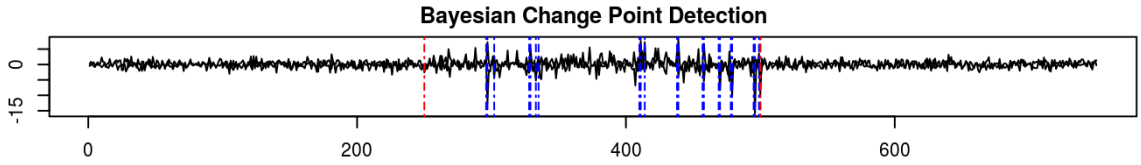

Figure 3: bcp on Multivariate case2: bcp made many false detections on the second period where the variance of each sequence is large.

Moreover, as bcp assumes the observations to be independent Gaussian within blocks throughout each sequence, the detections bcp made on the first multivariate case are all on the second period where there is large covariance between each sequence but no change in mean and variance among blocks.

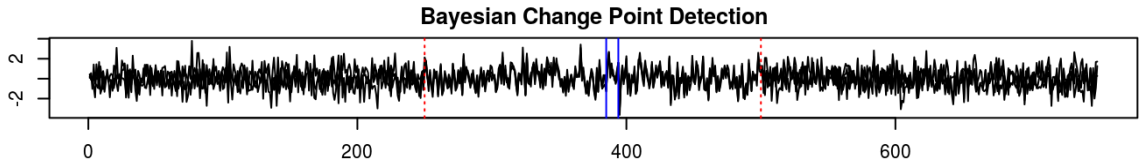

Figure 4: bcp on Multivariate case1: bcp made detections on the period where there is a high covariance among sequences.

Comparing to bcp, ecp appears to have a higher tolerance of variance, and it can detect the change of all mean, variance and covariance. Therefore, ecp has a pretty good performance on most of the simulations.

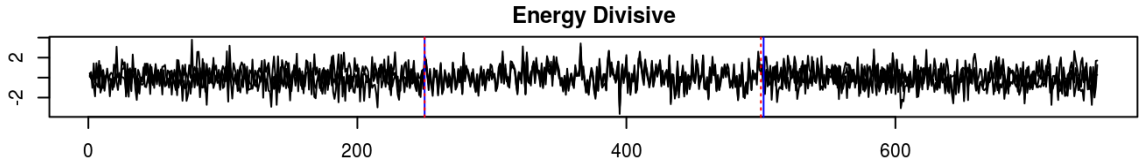

Figure 5: ecp is the only one that detected the change in covariance

Except for correctly detecting both change point in the linear breaks case, which is the same as most other algorithms, breakout did pretty poor on other two univariate sequences we simulated.

Although oCPD can also be impacted by large variance, it is more compatible with detecting changes in covariance than bcp. In the simple case, it didn't detect the third change point as bcp, which might be caused by the same reason of large variance, however, it was not affected by the large covariance in the multivariate case with change in covariance. otCPD has a very similar performance to oCPD, but it is obviously improved. In the linear break case, otCPD perfectly detected both of the change points with a proper merge size, while oCPD detected more on the linear trend, and otCPD also successfully avoid the two detections oCPD made on the second period where the trend is steep.

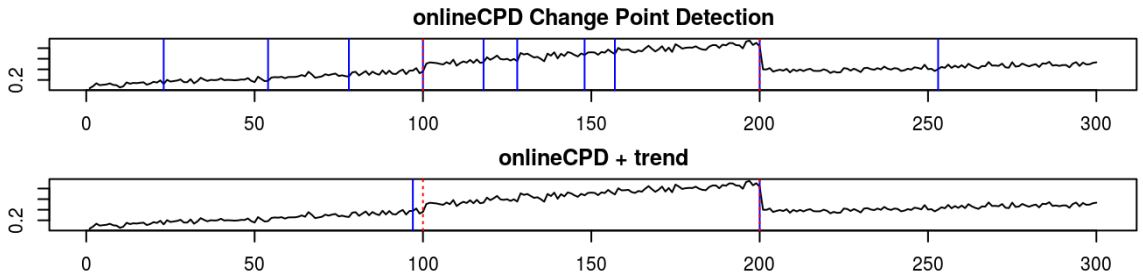

Figure 6: oCPD vs. otCPD on linear breaks

In addition, the filter improvement made to oCPD and otCPD enables them to perform much better.

For the EuroGames data we analyze the result in three parts:

1. Compare the performance of different methods, when given the number of references, using different time interval.

2. Compare the performance of different methods, when the number of references is not given, using different time interval.

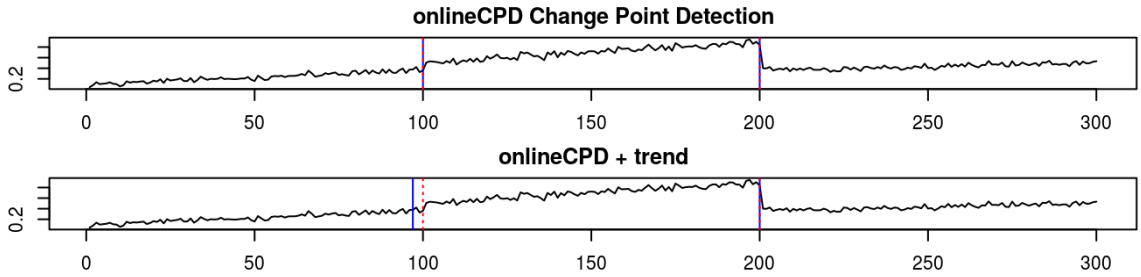

Figure 7: oCPD vs. otCPD on linear breaks after giving the number of references

3. Compare the average performance of ecp and otCPD on count and sentiment, as they are the best methods.

In the following two tables we showed the result after comparing the number of victories through all 18 games between each two methods.

Table 1: F-score order when the number of references is not given

	Counts	Sentiment
15 Sec	ecp >otCPD >oCPD >breakpout >bcp	otCPD >ecp >oCPD >bcp
30 Sec	otCPD >ecp = breakout >oCPD >bcp	ecp >otCPD >oCPD >bcp
60 Sec	breakout >otCPD >ecp >bcp >oCPD	ecp >otCPD >bcp >oCPD

Table 2: F-score order when the number of references is given

	Counts	Sentiment
15 Sec	otCPD >bcp = ecp = oCPD >breakout	ecp >otCPD >bcp >oCPD
30 Sec	otCPD >ecp >breakout >bcp >oCPD	ecp >>bcp >otCPD >oCPD
60 Sec	ecp >breakout >= bcp >>otCPD >oCPD	ecp >>bcp >>otCPD >>oCPD

From table one the following patterns were observed:

(i) The rank of breakout and bcp increases as the time interval increase, which might be due to their sensitivity of variance.

(ii) The rank of oCPD and ecp decreases as the time interval increases, as they might be better at dealing with small interval of data with more noise, however, ecp is commonly the best method on sentiment.

(iii) Generally, otCPD performs better than ecp on count and ecp performs better than otCPD on sentiment,

(iv) As the time interval increases, bcp seems to have a better and better performance than oCPD, which might also be due to bcp's sensitivity to variance.

To visually demonstrate the increase in noise level as time interval increases, here are two selected case:

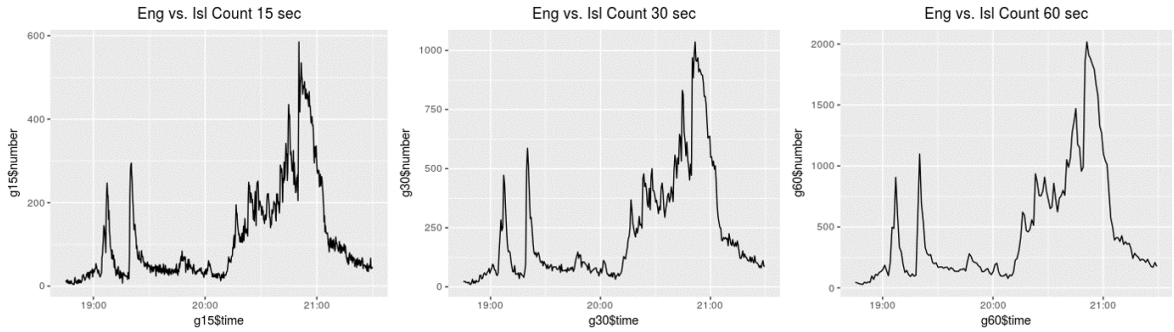

Figure 8: EuroGame: Eng vs. Isl on count

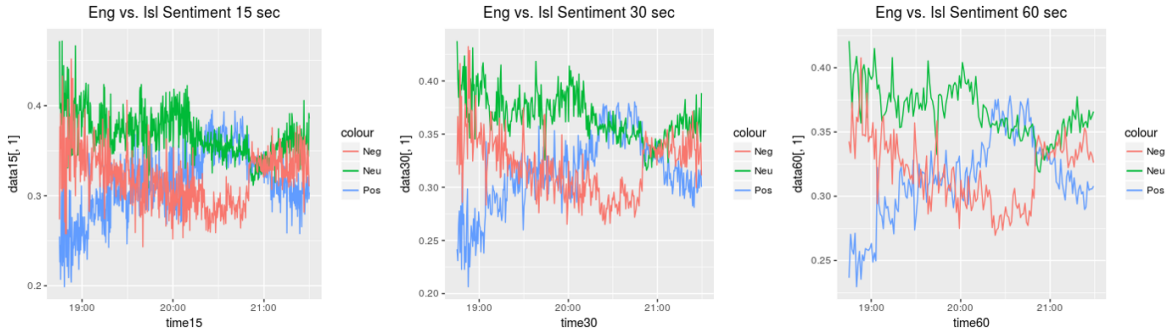

Figure 9: EuroGame: Eng vs. Isl on sentiment

On the other hand, if we give the algorithms the number of references, ecp and bcp are largely improved, while oCPD and otCPD's performances almost stay the same, and some new patterns are observed:

(i) ecp performs much better than all other algorithms on sentiment.

(ii) bcp becomes better as the time interval increases, and it beats both of oCPD and otCPD except it is only better than oCPD on 15 sec.

As we were intended to see if sentiment is a better measure than count, we choose the algorithms with best performances: ecp and otCPD, and see how they perform on count and sentiment. Still, we use all 18 games and see on how many the performance on sentiment is better than count. We had the following result:

The result in the table indicates that ecp has a overall better performance on sentiment if knowing the number of references, otherwise a similar performance on both cases is expected. On the other hand, otCPD has a similar performance on count and

Table 3: ecp

	15 sec	30 sec	60 sec
Count vs. Sentiment	Win / Tie / Lose	Win / Tie / Lose	Win / Tie / Lose
unknown references	10 / 0 / 8	7 / 0 / 11	9 / 0 / 9
known references	5 / 1 / 12	5 / 2 / 11	3 / 4 / 11

Table 4: otCPD

	15 sec	30 sec	60 sec
Count vs. Sentiment	Win / Tie / Lose	Win / Tie / Lose	Win / Tie / Lose
unknown references	9 / 0 / 9	13 / 1 / 4	16 / 0 / 2
known references	8 / 0 /10	13 / 1 / 4	16 / 0 / 2

sentiment with time interval 15 second, but it performs better and better on count as the time interval increases.

The first case of stepwise experiment has the same result as the offline mode with 30 seconds time interval, but when we change the batch size and time gap between each data-in to smaller units, bcp becomes better and bcp, oCPD, ecp have a similar performance.

Table 5: Stepwise Stream Data

	Performance
25 mins step, 50 mins batch size	otCPD >ecp >oCPD >bcp
5 mins step, 30 mins batch size	otCPD >= bcp = oCPD = ecp

5. Discussion

Although oCPD and otCPD are able to filter detections when the detections are more than references by selecting detections of the number of references highest post-probabilities, detections are usually less than references, which usually makes them at a disadvantage if all algorithms are given references comparing to bcp and ecp.

Besides, bcp and breakout seem to have a low tolerance of noises, while oCPD and otCPD perform better when more details are included.

6. Conclusion

The experiments were performed on both simulated and real data, and the results from Euro Games verified our observation on simulations.

Overall, otCPD is indeed improved as it is always better than oCPD, and it has a pretty good performance over others on univariate case when containing more information(smaller time interval) no matter the number of references is given or not, otherwise ecp seems to be a more powerful method among all.

Besides, if we don't know the number of references, otCPD would be our first choice on univariate cases, and breakout can also be considered if the curve is smooth as the algorithm is fast, and otCPD could be a second choice after ecp on multivariate cases. Moreover, if we know the number of references bcp can be a good choice after ecp.

Lastly, the stepwise streaming data verified the excellent performance of otCPD on univariate time series when it comes to more practical situation.

References

[1] Daniel Barry; J. A. Hartigan. "A Bayesian Analysis for Change Point Problems". In: *Journal of the American Statistical Association* 88.421 (1993), pp. 309–319.

[2] Chandra Erdman and John W. Emerson. "A Fast Bayesian Change Point Analysis for the Segmentation of Microarray Data". In: *Bioinformatics* 24.19 (2008), pp. 2143–2148. URL: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/btn404.

[3] Nicholas A. James; David S. Matteson. "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data". In: *Journal of Statistical Software* 62.7 (2014).

[4] N. A. James, A. Kejariwal, and D. S. Matteson. "Leveraging Cloud Data to Mitigate User Experience from "Breaking Bad"". In: *ArXiv e-prints* (Nov. 2014). arXiv: 1411.7955 [stat.ME].

[5] R. Prescott Adams and D. J. C. MacKay. "Bayesian Online Changepoint Detection". In: *ArXiv e-prints* (Oct. 2007). arXiv: 0710.3742 [stat.ML].

[6] Cyril Goutte and Eric Gaussier. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation". In: *Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings*. Ed. by David E. Losada and Juan M. Fernández-Luna. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 345–359. ISBN: 978-3-540-31865-1. DOI: 10.1007/978-3-540-31865-1_25. URL: https://doi.org/10.1007/978-3-540-31865-1_25.