An assessment of practitioners approaches to forecasting in the presence of changepoints.

Chapman, Jamie-Leigh and Killick, Rebecca
Department of Mathematics & Statistics, Lancaster University

Abstract

A common challenge in time series is to forecast data which suffers from structural breaks or changepoints which complicate modeling. If we naively forecast using one model for the whole data, the model will be incorrect and thus our forecast error will be large. There are two common practices to account for these changepoints when the goal is forecasting: 1) Pre-process the data to identify the changepoints, incorporating them as dummy variables in modeling the whole data; 2) Include the changepoint estimation into the model and forecast using the model fit to the last segment. This article examines these two practices, using the computationally exact PELT algorithm for changepoint detection, comparing and contrasting them in the context of an important Software Engineering application.

1 Introduction

Structural breaks and changepoints occur in time series data arising from a variety of fields including; medicine[1], environment[2,3], psychology[4] and finance[5]. A key goal in many applications is to understand the dynamics of a time series to produce accurate forecasts into the future. If there are changepoints within a time series, and these changepoints are not accounted for, then the estimated dynamics are distorted. This paper considers the different methods for accounting for changepoints when forecasting time series and compares and contrasts them.

Forecasting in the presence of changepoints is considered by[6–8] and in particular,[9] discuss and quantify the costs associated with ignoring changepoints when forecasting in macroeconomic and financial settings. In contrast we consider *how*

the changepoints are taken into account and present the findings for two common approaches.

Let us denote our time series data as $\{y_i\}_{i=1,\ldots,n}$, for which we make no assumptions regarding the data generating process. Suppose we wish to forecast future observations $\{y_i\}_{i=n+1,\ldots,n+h}$ for some horizon h. Then, it is common to first select a class of time series models, seemingly appropriate for the data, from which forecasts will be generated. In what follows, the class of time series models we use for forecasting are seasonal Autoregressive Moving Average (ARMA) models of known frequency f. We denote this model as $\mathrm{ARMA}(p,q) \times (P,Q)_f$ and write:

$$y_t = \mu + \frac{\theta(B)\Theta(B^f)}{\phi(B)\Phi(B^f)}\epsilon_t, \tag{1}$$

where $\phi(B)$ is the autoregressive operator and $\theta(B)$ is the moving average operator, each represented as a polynomial in the backwards shift operator given as

$$\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p, \tag{2}$$
$$\theta(B) = 1 + \theta_1 B + \ldots + \theta_q B^q. \tag{3}$$

Similarly, $\Theta(B^f)$ and $\Phi(B^f)$ are the seasonal moving average and autoregressive operators, respectively, given by

$$\Phi(B) = 1 - \Phi_1 B^f - \ldots - \Phi_p B^{fP}, \tag{4}$$
$$\Theta(B) = 1 + \Theta_1 B^f + \ldots + \Theta_q B^{fQ}. \tag{5}$$

The noise process ϵ_t in equation (1) is i.i.d. Gaussian with mean zero and variance σ^2. Here we are using a seasonal ARMA model rather than seasonal differencing to allow seasonal lags in both the AR and MA components. One could use seasonal differencing but estimating the order of the differencing becomes challenging as the presence of changepoints makes any test invalid.

Once the class of time series models has been chosen, the order and parameters are often estimated using all of the historical data. However, if the data are subject to changes then using all of this data may not be appropriate.

If the data are subject to changepoints, then[6] propose to only use post-break data to estimate the time series model used for forecasting. They estimate the location of the break to be the most recent changepoint which is obtained using a reversed CUSUM procedure[10]. In further work,[11] propose that if the goal is to minimize the mean squared forecast error, then some pre-break data may be useful for model fitting. This so called "trade off window" approach of[11], using both

pre- and post-break data, is motivated by the trade off between bias and forecast error variance. Provided that the changepoint is not too large (in magnitude), by introducing more observations, they are reducing variance at the cost of possible bias which may overall result in improved forecasts.

In this article we consider two approaches that are used in practice to forecast in the presence of changepoints. The first approach *pre-estimates* the changepoints and, acknowledging that there may be more complex dynamics, uses the changepoint locations as dummy variables in the model in equation (1), denoted the *pre-estimation* approach. The second approach creates a *changepoint model* where each segment is assumed to follow model (1) and uses only the model fit from the final segment to produce forecasts, denoted the *modeling* approach.

In each of the approaches, in order to detect changepoints, we use a *penalized cost function* approach which solves the constrained minimization problem exactly. In such a setting, given a sequence of observations $\{y_i\}_{i=1,\ldots,n}$, the aim is to find the number of changes, m, and the associated changepoints, $\{\tau_j\}_{j=1,\ldots,m}$, which minimize:

$$\sum_{j=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{j-1}+1):\tau_j})\right] + \beta m. \tag{6}$$

The m changepoints cause the data to be split into $m+1$ independent segments such that segment j contains the observations $y_{(\tau_{j-1}+1):\tau_j}$. We necessarily set $\tau_0 = 1$ and $\tau_{m+1} = n$. In practice we impose a minimum segment length, g, such that $\tau_{j+1} - \tau_j \geq g \geq 2$. Naturally the length of the segment constrains the maximum order of the model from (1). The first term in equation (6) is a cost function for the segment $y_{(\tau_{j-1}+1):\tau_j}$. The second term in equation (6) is a penalty which guards against over fitting. Different methods can be adopted in order to minimize equation (6). Here, we use the Pruned Exact Linear Time (PELT) algorithm[12], to minimize equation (6) as it solves the constrained optimization problem exactly using a computationally efficient strategy.

The structure of this article is as follows. In Sections 2 and 3, we describe the *pre-estimation* and *modeling* approaches for forecasting in the presence of changepoints. In Section 4 we compare each of these methods to their non-changepoint counterparts. Finally, in Section 5, we test our methods on two time series from a Software Engineering problem.

2 Pre-Estimation Approach

Time series are often prone to changes in mean. However, if these changes are not modeled, then the autocorrelation across time may be estimated incorrectly, potentially indicating a long memory model when inappropriate[13]. The seasonal ARMA model in equation (1), relies on capturing the autocorrelation of the time series appropriately. Often changepoint detection is part of the pre-processing of data prior to further analysis. In this vein the locations of the changepoints themselves are not of particular interest and, just as with detecting outliers, changepoint estimation is conducted to "clean" the data. Thus, the estimated changepoints are entered as dummy regressors in the seasonal ARMA model (1).

2.1 The Model

Suppose we wish to forecast time series data $\{y_i\}_{i=1,\dots,n}$ using the seasonal ARMA model in equation (1). Prior to estimating this model, we first detect any changes in mean. To do this, we take a penalized likelihood approach to changepoint detection (6). In this setting, we replace the cost function $\mathcal{C}(\cdot)$, in equation (6), with twice the negative log-likelihood for a Gaussian distribution with common variance and segment specific mean. We estimate the global variance by the median of the variances of a moving window of size 30. Using the median decreases the effect of the increased variances in windows containing the mean changes. We use a window of size 30 as this strikes a balance between wanting a large window size to avoid too much variance in the estimation and wanting a small window size to avoid the inclusion of changepoints within too many windows. We should be clear that we do not assume this is an appropriate model for the data, but is a "broad brush" to identify large changes in mean and is the approach that practitioners often take in practice.

The second component of equation (6) is the penalty used to prevent overfitting to the mean of the data. We are assuming that $\{y_i\}_{i=1,\dots,n}$ are independent Gaussian observations. However, in a time series setting our data will contain autocorrelation. Despite this,[14] demonstrate that minimizing equation (6) is still effective at locating changes in mean but we need to inflate our penalty to avoid overfitting.

Having minimized equation (6) using a Gaussian cost function and an inflated penalty, the result is m' changepoint locations, $\{\tau_j\}_{j=1,\dots,m'}$, estimating changes in the mean level of the time series. Using these m' changepoint locations, we can

produce $(m'+1)$ segment indicators:

$$v_t^j = \begin{cases} 1 & \text{if } \tau_{j-1} < t \leq \tau_j, \\ 0 & \text{otherwise.} \end{cases} \quad \text{for } j = 2, \ldots, m'+1. \tag{7}$$

Our data can now be modeled using the following linear relationship:

$$y_t = \sum_{j=1}^{m'+1} \mu_j v_t^j + r_t = \sum_{j=1}^{m'+1} \mu_j v_t^j + \frac{\theta(B)\Theta(B^f)}{\phi(B)\Phi(B^f)}\epsilon_t, \tag{8}$$

where $v = (v_1, \ldots, v_{m'+1})$ are the segment indicators and the remaining parameters are as in equation (1). See[15] for consideration of this model.

In equation (8), $\{\mu_j\}_{j=1,\ldots,m'+1}$ are the size of the mean change to be estimated. In order to produce forecasts, we estimate the model (8) using all of the historical data. When the model is estimated, it is preferable to minimize the sum of squared values ϵ_t, and not the r_t, as this takes into account the estimation of the mean for each segment. Alternatively, it is also common practice to first estimate the changepoint model and then estimate the seasonal ARMA model on the residuals. This is a similar approach, however the estimates of the seasonal ARMA structure are potentially biased by mis-estimation of the overall mean. Thus, we do not consider this approach further. In either case, the estimated model can then be used to produce forecasts assuming no changes occur in the forecast period.

In Section 4 we see how the *pre-estimation* approach behaves in a simulation study. In the following section, we describe an alternative approach, incorporating the changepoint estimation into the seasonal ARMA model.

3 Modeling Approach

In the approach described in Section 2, the seasonal ARMA model (1) is estimated (via maximum likelihood estimates) using all of the historical data $\{y_i\}_{i=1,\ldots,n}$ and is assumed not to vary. In practice, the parameters of this model may change over time. In such a case, we may not want to use all of the historical data to estimate the model. Here, we outline an alternative approach, which instead estimates the changepoints and the varying time series structure together.

3.1 The Model

Suppose again that we wish to forecast time series data $\{y_i\}_{i=1,\ldots,n}$ using the seasonal ARMA model in equation (1). However, we do not necessarily want to

use all of the historical data for forecasting as we believe that the model structure varies. Hence, in order to determine which observations should be used to estimate the model in equation (1), we propose to use a cost function, $\mathcal{C}(\cdot)$, in equation (6), which is based upon the log-likelihood of a seasonal ARMA model. That is, we use the cost function $\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) = -2\ell(y_{1:n};\theta)$, where $\ell(\cdot)$ is the log-likelihood of a seasonal Autoregressive Moving Average (ARMA) model.

It should be noted that in the above framework, we are allowing both the order of the ARMA model to change, and the associated coefficients. In addition to this, we are also allowing for a change in mean level to occur by the inclusion of μ_i in equation (1).

Once we have minimized equation (6), we forecast from the model in the last segment, using the data $\{y_i\}_{i=\tau_m,\ldots,n}$, where τ_m is the final changepoint location detected.

There are several practical considerations when applying this approach which we discuss in the following section.

3.2 Discussion

In the previous section we outlined an approach which only uses the most recent segment of the data for model estimation. Consequently, once a changepoint has occurred, we are deeming pre-change data uninformative. It is therefore important to carefully consider the choice of minimum segment length, g, in equation (6).

It important that the minimum segment length is not set so small such we are producing out-of-sample forecasts based only on a small amount of data. In particular, if the data has seasonality, then we must allow enough observations in a segment to estimate this seasonality. Although, the longer the minimum segment length, the more time we have to wait to detect a change. Consequently, we could be fitting an incorrect model to the last segment of the data therefore introducing bias into our model.

The combination of penalty and minimum segment length can have a large influence on the detected changepoint locations and hence the window we are using to estimate our forecasting model. The combination of these two allows us to control the trade off between the bias and variance of our forecasts. As such, in practice, one could compare, or combine, multiple forecasting models based upon the different segmentations obtained when changing the combination of minimum segment length and penalty. However, this is beyond the scope of this paper.

A further consideration is the number of parameters we are fitting. This is unknown *a priori* as both approaches use model selection to identify the order of

the model. However, if there is a changepoint present then the modeling approach is likely to estimate more parameters than the pre-estimation approach. This is because the modelling approach fits a full ARMA model either side of the change, whereas the pre-estimation only adds a single extra parameter for the post-change mean.

The following section considers the performance of the *pre-estimation* and *modeling* approaches in a simulation study.

4 Simulation Study

In this simulation study we test the performance of the *pre-estimation* and *modeling* approaches in forecasting. Specifically, we compare the following approaches:

- **M0 (*naive*):** A (seasonal) ARMA model estimated using data points $[1, n]$;
- **M1 (*pre-estimation*):** A (seasonal) ARMA model with regressors, each of which represent a mean level, which is estimated using data points $[1, n]$, as described in Section 2;
- **M2 (*modeling*):** A (seasonal) ARMA model which is estimated using data points $[\hat{\tau}_m, n]$, as described in Section 3.

In each approach, we estimate the seasonal ARMA model (1) using the `auto.arima` function from the `forecast` package[16] available for `R`[17]. In applying the `auto.arima` function we use the default settings except that we do not allow any of the parameters p, q, P or Q in equation (1) to exceed three.

To detect changes in mean for approach M1 we use the `cpt.mean` function from the `changepoint` package[18] in `R`. This function implements the PELT algorithm[12] for a change in mean under the assumption of Gaussian data. We set a minimum segment length of $g = 2$. We use a scaled BIC penalty[19] ($6 \log n$) to account for potential autocorrelation. Note that the `cpt.mean` function is written in such a way that it assumes that the constant variance across the data is equal to one. As such, we pre-scale the data to have variance one prior to detecting changes in mean.

For approach M2, the piecewise seasonal ARMA model is again fit using the `auto.arima` function. The penalty we use in equation (6) is the Modified Bayes Information Criteria (MBIC)[20]. The MBIC penalty accounts for the lengths of the segments and encourages changes to be distributed evenly across the dataset. This

is useful for forecasting as we want to discourage small segment lengths. We set a minimum segment length of $g = 8$, to ensure enough observations to fit an ARMA process.

We fit the models M0, M1 and M2 to a range of generative models detailed below. In each instance we simulate 500 replications, in which the error process is given by $\epsilon_t \sim \mathcal{N}(0, 1)$, and report a selection of commonly used in-sample and out-of-sample performance metrics:

- Mean Error (ME);
- Root Mean Squared Error (RMSE);
- Mean Absolute Error (MAE);
- Mean Percentage Error (MPE);
- Mean Absolute Scaled Error (MASE)[1];

All results are reported in Table 1 for the training (in-sample) set and in Table 2 for the test (out-of-sample) set. The test set is four observations and uses a rolling one-step ahead forecast.

The following generated models are used.

(a) An AR(2) model with no seasonal components. This scenario is designed to asses the method when there are no changepoints and hence the most appropriate model is M0. Specifically, for this model, we simulate from

$$Y_t = 0.8Y_{t-1} - 0.2Y_{t-2} + \epsilon_t, \qquad 1 \leq t \leq 512. \tag{9}$$

For scenario (a), M1 produces an overall better in-sample fit (Table 1) than the other two models. This over-fitting of the data is due to the presence of autocorrelation which can induce features that resemble changes in mean when independence is assumed[3]. Figure 1 shows a single realization from scenario (a) along with incorrectly detected changes in mean. Despite inflating the penalty to account for some autocorrelation, changes in mean are still detected. Consequently, M1 over-fits to the level of the time series, and as a result, will miss-specify the autoregressive parameters of the model.

Tables 1 and 2 for M0 and M2 are the same as no changes are detected once the autocorrelation is modeled. This suggests a low false positive rate for detecting

[1]MASE calculation is scaled using MAE of training set naive forecasts for non-seasonal time series, training set seasonal naive forecasts for seasonal time series and training set mean forecasts for non-time series data[16].

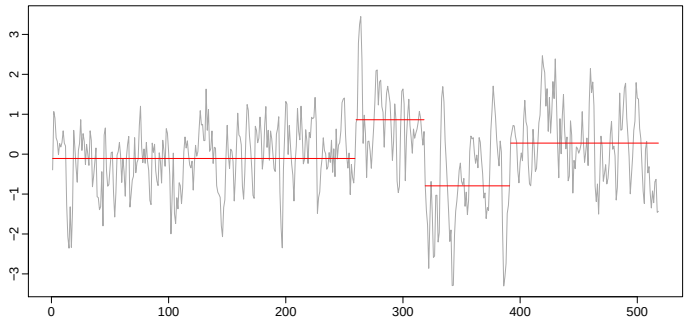


Figure 1: A realization $\{Y_t\}$ from scenario (a) with detected changes in mean. We can see that although there are no 'true' changes in mean, the autocorrelation causes periods of lower and higher mean values.

changes in the ARMA model and implies that very few changes are detected. For the test set, M0 and M2 produce better out-of-sample forecasts, which confirms the over fitting of M1.

(b) An AR(2) model with no seasonal components and a change in mean level. This scenario is designed to asses the approaches when there are no changes in AR structure but there is a change in mean. This scenario should favour M1, if a single mean is estimated. However M2 could also perform well despite the AR structure being estimated differently either side of the mean change. Specifically, for this model, we simulate from

$$Y_t = \begin{cases} 0.8Y_{t-1} - 0.2Y_{t-2} + \epsilon_t & 1 \leq t \leq 256 \\ 2 + 0.8Y_{t-1} - 0.2Y_{t-2} + \epsilon_t & 256 \leq t \leq 512 \end{cases} \tag{10}$$

For scenario (b), we can see from the results in Table 1 that approach M1 produces a better fit to the training set overall, this is expected as it is the most appropriate method to use for the scenario. Out-of-sample however, the results in Table 2 show that M0 produces the best forecasts with M2 producing similar values.

When we compare M1 to M2, overall M2 produces better forecasts. This suggests that M2 is detecting the change in mean more effectively. If it were not, we would expect M1 to outperform M2 because M1 would be estimating the ARMA model using all of data. M2 is more capable of detecting the true location of the change in mean because the cost function used in equation (6) is true to the data generating process.

Overall, the results in Tables 1 and 2 for this scenario suggest that forecasts with the correct model are not always more accurate than forecasts from an incorrect model. This confirms previous similar findings in[21].

(c) A piecewise AR(2) model with changing coefficients. The scenario should favour the approach in M2. We simulate from

$$Y_t = \begin{cases} 0.8Y_{t-1} - 0.2Y_{t-2} + \epsilon_t & 1 \leq t \leq 256 \\ 0.5Y_{t-1} - 0.1Y_{t-2} + \epsilon_t & 256 \leq t \leq 512 \end{cases} \tag{11}$$

As expected M2 produces a better in-sample fit to the data (Table 1). The in-sample results for M0 and M1 differ, suggesting that M1 is detecting changes in mean as a consequence of the autocovariance. The almost zero mean error for model M2 suggests that the changepoints are being detected with very high accuracy. Overall, the results in Tables 1 and 2, for the training and test set respectively, support the use of model M2.

(d) A piecewise AR model with changing AR order and a short segment at the beginning of the time series. Again model M2 should perform the best for this scenario. We simulate from

$$Y_t = \begin{cases} 0.1Y_{t-1} - 0.6Y_{t-2} - 0.3Y_{t-3} + \epsilon_t & 1 \leq t \leq 50 \\ 0.3Y_{t-1} + \epsilon_t & 51 \leq t \leq 512 \end{cases} \tag{12}$$

In this scenario both the order and the coefficients of the AR model change and thus M2 can capture this. We can see from the results in Tables 1 and 2, that as expected, M2 produces the best in-sample results, and it also achieves the best out-of-sample forecasts. Once again, the very small mean error suggests that the changepoints are being detected with high accuracy. Model M1 produces the worst out-of-sample forecasts, likely due to the over fitting of the changepoint process.

(e) A piecewise AR model with changing AR order and a short segment at the end of the time series. This is subtly different from scenario (d) as the changepoint is towards the end of the series thus may affect the forecasts more substantially. Whilst we expect M2 to be the best model again, it will be interesting to see how the other models behave. For this model, we simulate from

$$Y_t = \begin{cases} 0.1Y_{t-1} - 0.6Y_{t-2} - 0.3Y_{t-3} + \epsilon_t & 1 \leq t \leq 462 \\ 0.3Y_{t-1} + \epsilon_t & 462 \leq t \leq 512 \end{cases} \tag{13}$$

Table 2 shows that M2 produces better out-of-sample forecasts. However, M1 produces the best in-sample forecasts due to over-fitting. Table 2 shows that the

results are similar to scenario (d) except for MAPE which is considerably higher for model (e) than in model (d). In addition, the MPE is very poor for M2. This is expected because scenario (d) has a longer segment at the end of the data which will produce a better model fit with less variability and thus improved forecasts. This suggests that M2 is successfully detecting the changepoint towards the end of the data. We would expect, that as the length of the final segment becomes smaller, forecasts will become less accurate for approach M2 as the time series model is being estimated with increasingly less data. However, a point will be reached such that the final change is too close to the end of the data to detect, at which point M2 will perform similarly to M0.

(f) A piecewise seasonal ARMA(2,0)(1,0) model, frequency 4, whose seasonal component has a change in coefficients. As the seasonal component is changing the most appropriate model is M2. Specifically, for this model, we simulate from

$$Y_t = \begin{cases} 0.9Y_{t-1} - 0.2Y_{t-2} - 0.9Y_{t-4} + \epsilon_t & 1 \leq t \leq 256 \\ 0.9Y_{t-1} - 0.2Y_{t-2} - 0.2Y_{t-4} + \epsilon_t & 256 \leq t \leq 512 \end{cases} \tag{14}$$

For this scenario, we can see again from Tables 1 and 2 that approach M2 produces the best results for both in-sample and out-of-sample forecasts. Both M0 and M1 produce very poor results in comparison to M2. This suggests that M2 is accurately detecting the changes in seasonal coefficient.

(g) A piecewise seasonal AR model whose seasonal component has a change in order, i.e. an ARMA(2,0)(2,0) changes to an ARMA(2,0)(1,0). Specifically, for this model, we simulate from

$$Y_t = \begin{cases} 0.9Y_{t-1} - 0.2Y_{t-2} - 0.9Y_{t-4} - 0.8Y_{t-8} + \epsilon_t & 1 \leq t \leq 256 \\ 0.9Y_{t-1} - 0.2Y_{t-2} - 0.9Y_{t-4} + \epsilon_t & 256 \leq t \leq 512 \end{cases} \tag{15}$$

In scenario (g) the seasonality component of the model exhibits a change in order. Approach M2 captures this the best in-sample and out-of-sample, with M0 producing the poorest in-sample results. This demonstrates, that as the nature of the changes become more complex, approach M2 is best at capturing them. This overall results in improved forecasts.

(h) A piecewise ARMA model which changes from an ARMA(1,0) to an ARMA(1,1). For this scenario, a moving average term is introduced in the second segment of the time series. We specifically simulate from

$$Y_t = \begin{cases} 0.3Y_{t-1} + \epsilon_t & 1 \leq t \leq 300 \\ 0.3Y_{t-1} + \epsilon_t + 0.7\epsilon_{t-1} & 300 \leq t \leq 512 \end{cases} \tag{16}$$

Approach M2 best captures the introduction of a moving average term in the second segment of time series realised from scenario (h). This is most clear in-sample (Table 1). This again illustrates that as the nature of the change becomes more complex, approach M2 performs best.

Overall we can conclude that the inclusion of changepoints in the modeling stages of forecasting, i.e. approach M2, produces better results. In particular, when the time series exhibits changes in its seasonal structure, or changes in AR order, then estimating the model using the final segment of the data can outperform estimation based upon the entire data set.

At times, estimating the model using all the data, whilst including regressors for changes in the mean level, can over fit the data. However, as these changes begin to occur in higher order structures of the time series, for example in scenario (g), the inclusion of these regressors produces better out-of-sample forecasts.

In the following, we consider forecasting for a software engineering problem using each of the approaches.

5 Application to Software Run-Time Prediction

A recent study found that website sales fall by roughly 7% for each extra 0.1s a page takes to display[22]. Thus accurately measuring and predicting software performance is a vital task for software engineers. However, the ever-increasing levels of non-determinism in modern hardware and software mean that many programs have unpredictable and surprising performance patterns that undermine current benchmarking performance methodologies.

One surprising aspect of software run-times is that they are subject to changepoints. We consider here two examples from[23] which are depicted in Figure 2. This experiment controlled the environment for the benchmark to ensure exact replication across tests. The data recorded is the time taken to run the benchmark in seconds across 2000 replications. It is clear from Figure 2 that whilst very different in manifestation, both benchmark run times contain changepoints which will affect a naive forecast. Typically, hundreds of models and forecasts would be produced, and it would be impractical to inspect each one. Thus we have intentionally picked two benchmarks with different dynamics to assess.

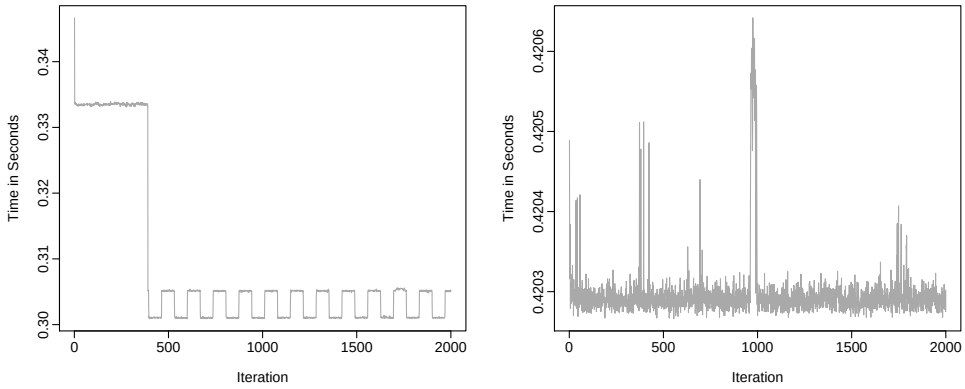
We will compare the performance of approaches M0, M1 and M2 on these two runtime processes whose dynamics are very different. To do this, we perform an extending window estimation. To begin, we fix an initial estimation period from the iteration 1 until iteration 1950. Then we forecast 1 step ahead and calculate

	ME	RMSE	MAE	MPE	MAPE	MASE
Scenario (a)						
M0	0.0016	1.0060	0.8037	**47.0690**	346.8433	0.9177
M1	**-0.0000**	**0.9914**	**0.7916**	49.0590	**336.7375**	**0.9040**
M2	0.0016	1.0060	0.8037	**47.0690**	346.8433	0.9177
Scenario (b)						
M0	0.0333	1.0255	0.8175	**-1.3018**	264.2667	0.9309
M1	-0.0029	**1.0002**	**0.7984**	-13.5072	250.8929	**0.9089**
M2	**-0.0008**	1.0017	0.7985	-12.6762	**203.3246**	0.9101
Scenario (c)						
M0	0.0003	1.0106	0.8055	52.8808	296.6232	0.8728
M1	0.0017	0.9979	0.7951	**39.7553**	301.7013	0.8617
M2	**-0.0000**	**0.9966**	**0.7938**	50.4126	**263.5910**	**0.8392**
Scenario (d)						
M0	0.0013	1.0400	0.8285	89.0696	**293.5366**	0.7906
M1	-0.0008	1.0274	0.8186	**86.7520**	299.5224	**0.7812**
M2	**0.0001**	**1.0046**	**0.8034**	70.5495	305.7549	0.8053
Scenario (e)						
M0	**-0.0001**	1.0206	0.8124	**-10.0230**	665.4235	0.5624
M1	-0.0007	**1.0113**	**0.8056**	-31.3873	679.9682	**0.5569**
M2	0.0037	1.0374	0.8261	-191.9323	**604.2020**	0.5932
Scenario (f)						
M0	-0.0019	1.7558	1.3751	6461.9811	7115.0305	0.5908
M1	**-0.0011**	1.7444	1.3670	6464.0948	7118.3103	**0.5855**
M2	0.0167	**1.2740**	**1.0198**	**41.7718**	**239.5152**	0.8775
Scenario (g)						
M0	0.0030	2.3865	1.8896	34.7642	265.1804	0.5545
M1	**-0.0003**	2.2755	1.7986	**33.3716**	255.8116	0.5286
M2	-0.0087	**1.8575**	**1.4690**	34.4501	**191.2260**	**0.4271**
Scenario (h)						
M0	-0.0015	1.0671	0.8522	692.5047	958.4427	0.8537
M1	**0.0003**	1.0481	0.8368	625.4978	922.4785	0.8384
M2	0.0020	**1.0011**	**0.8001**	**86.2657**	**318.5515**	**0.7989**

Table 1: Mean Error, Root Mean Square Error, Mean Absolute Error, Mean Percentage Error, and Mean Absolute Square Error, to four decimal places, for the **in-sample** forecasts for 500 realizations of scenarios (a)-(h) using approaches M0, M1 and M2.

	ME	RMSE	MAE	MPE	MAPE	MASE
Scenario (a)						
M0	**-0.1563**	**1.1671**	**1.0282**	**113.3594**	**169.0906**	**1.1740**
M1	-0.2580	1.3524	1.1853	130.0618	184.9175	1.3521
M2	**-0.1563**	**1.1671**	**1.0282**	**113.3594**	**169.0906**	**1.1740**
Scenario (b)						
M0	-0.1769	**1.1521**	**1.0119**	**-17.0005**	221.3590	**1.1517**
M1	-0.2434	1.3917	1.2360	-202.3013	456.4171	1.4062
M2	**-0.1411**	1.1583	1.0183	-17.8884	**206.2740**	1.1602
Scenario (c)						
M0	0.1661	1.0751	0.9265	68.4680	188.4346	1.0055
M1	**0.1211**	1.1043	0.9442	110.8610	196.6434	1.0249
M2	0.1711	**1.0643**	**0.9143**	**68.0920**	**156.3316**	**0.9695**
Scenario (d)						
M0	**0.0313**	0.9381	0.7881	102.6425	160.8289	**0.7535**
M1	0.0325	0.9767	0.8260	129.5255	207.3461	0.7888
M2	0.0354	**0.9100**	**0.7644**	**76.8555**	**133.8241**	0.7679
Scenario (e)						
M0	**-0.0711**	1.1381	0.9713	90.1890	216.4954	**0.6714**
M1	-0.1162	1.1723	1.0029	100.3486	244.7225	0.6941
M2	-0.0752	**1.1093**	**0.9436**	**86.3775**	**201.4708**	0.6774
Scenario (f)						
M0	-0.0669	1.8831	1.6579	117.3606	278.6023	**0.7133**
M1	-0.1936	2.1786	1.9157	178.7814	263.3264	0.8164
M2	**-0.0197**	**1.7353**	**1.5182**	**90.9365**	**194.1363**	1.3165
Scenario (g)						
M0	-0.1240	2.7002	2.3515	81.9197	**142.3822**	0.6852
M1	**0.0237**	6.0104	5.3974	**-54.1455**	460.6030	1.5817
M2	-0.2269	**2.1403**	**1.8835**	61.0397	185.1860	**0.5466**
Scenario (h)						
M0	**-0.0771**	1.2732	1.1167	103.9533	**112.0250**	1.1220
M1	-0.1899	1.3714	1.2174	**79.4708**	145.6388	1.2230
M2	-0.0983	**1.2607**	**1.1050**	109.0164	127.0335	**1.1112**

Table 2: Mean Error, Root Mean Square Error, Mean Absolute Error, Mean Percentage Error and Mean Absolute Square Error, to four decimal places, for the **out-of-sample** forecasts for 500 realizations of scenarios (a)-(h) using approaches M0, M1 and M2.



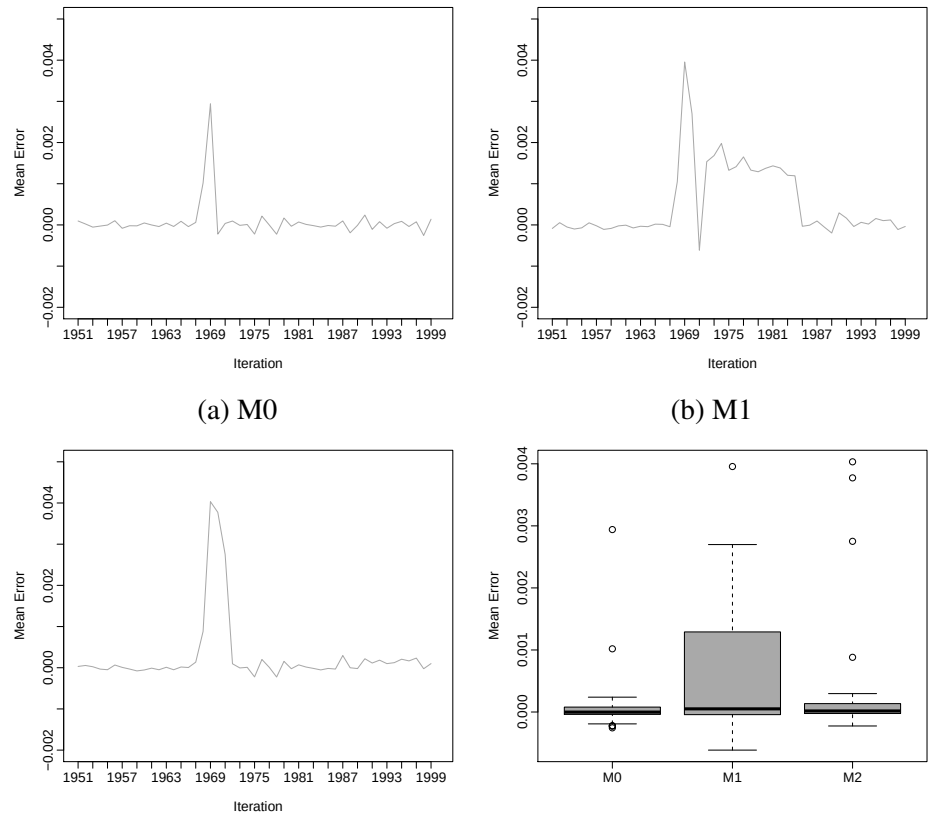
(a) Fannkuch Redux, Hotspot, Linux_{4790} (b) Spectral Norm, LuaJIT, Linux_{4790}

Figure 2: Runtime in seconds for the specified; benchmark, virtual machine, machine, across 2000 iterations.

the mean error of the forecast. Having done this, we extend the estimation period by one time step and again forecast one step ahead and calculate the mean error. We iterate this procedure up until iteration 2000 to produce an expanding window forecast for runtimes over 50 windows.

Figure 3 shows the results for the expanding window forecasts for the Fannkuch Redux, Hotspot, Linux_{4790} benchmark. This data set looks as though it may have a constant second order structure and simply be subject to changes in mean behaviour, thus we would expect M1 to be preferred. Each of the models have a similar average mean error for the forecasts. However we can clearly see that model M1 is not performing as well as M0 or M2 as it has higher variability. Note that within the window period there is a changepoint around 1970. Model M1 correctly finds this large change, the larger error instead comes from the inability to accurately estimate the post-change mean. It takes around 12 observations before the post-change mean is consistently estimated - recall that M1 does not take the autocorrelation into account when estimating the mean. In contrast model M0 adapts to the changepoint quickly and M2 is restricted by the minimum segment length as expected.

For the second example, Spectral Norm, LuaJIT, Linux_{4790}, there appear to be fewer changes in mean, a potential changing second order structure, and a longer segment at the end with no changes during the forecast window. Thus we may expect M2 to be preferred. Figure 4 shows the results for the expanding window forecasts. Models M0 has the smallest mean error but the largest variance. The



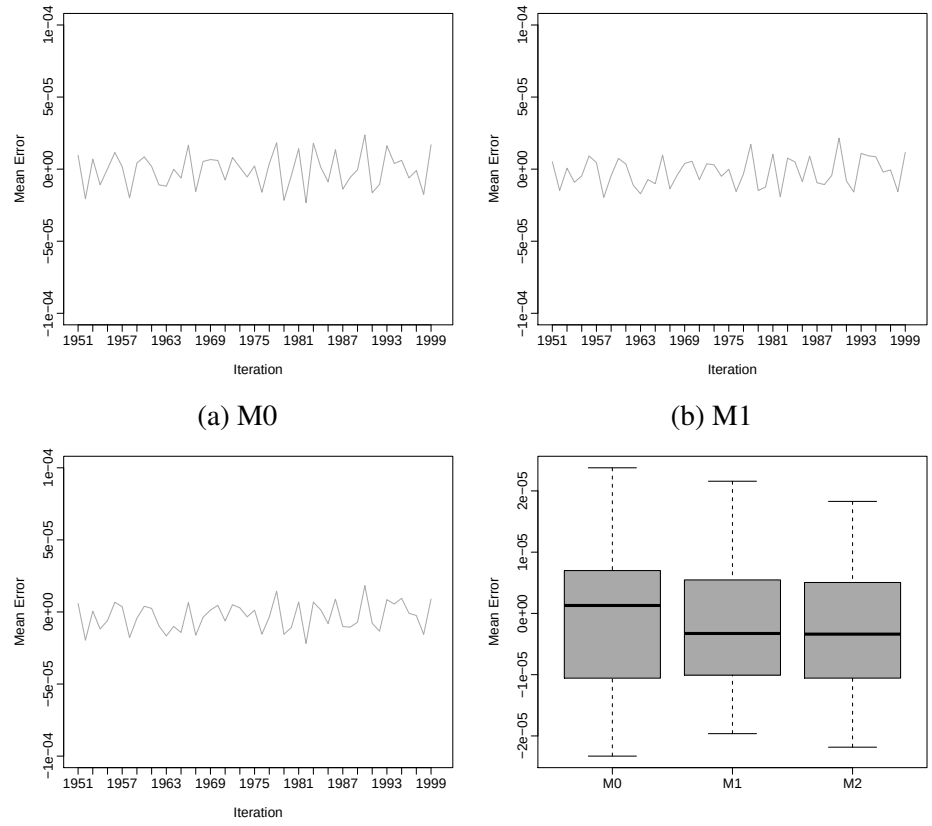
(a) M0

(b) M1

(c) M2

(d) Box Plot Comparison of the ME

Figure 3: **Fannkuch Redux, Hotspot, Linux**$_{4790}$: The Mean Error for a one step ahead forecast with model estimation period starting at iteration 1950 and ending as indicated by the x-axis of the plots. Figures (a) - (c) show the expanding window Mean Errors of the forecast for models M0, M1 and M2 respectively, and figure (d) compares the Mean Errors for each of the models.



(a) M0 (b) M1

(c) M2 (d) Box Plot Comparison of the ME

Figure 4: **Spectral Norm, LuaJIT, Linux**$_{4790}$: The Mean Error for a one step ahead forecast with model estimation period starting at iteration 1950 and ending as indicated by the x-axis of the plots. Figures (a) - (c) show the expanding window Mean Errors of the forecast for models M0, M1 and M2 respectively, and figure (d) compares the Mean Errors for each of the models.

mean error does not follow a Normal distribution as it exhibits strong left-skew. In contrast whilst model M1 and M2 have larger negative error, they do appear to be more symmetric with M2 slightly more symmetric than M1, indicating a better model fit.

6 Discussion and Conclusion

In this article we have assessed two commonly used approaches to forecasting which incorporate changepoints. We have shown that these two approaches have different strengths depending on the dynamics of the data. In addition to this, we have shown that forecasts can be based on less historical data, whilst still producing reasonable forecasts. As data is becomingly large scale, the need for reducing the amount of data used to fit models is becoming increasingly important, and questions such as "how much of my data is relevant for forecasting" can be potentially answered using changepoint methodology.

The two modeling frameworks presented here are very flexible. We can produce variants on our models by altering the minimum segment length and penalty choice. The choice of minimum segment length and penalty together, give us control over the trade off between bias and forecast error variance, especially in approach M1. This methodology is not restricted to the models considered here and can be adapted to other time series models provided we can define the cost function for a segment. For example, the seasonal ARMA model could be replaced with an exponential smoothing or GARCH model.

It may be the case, that in practice, the cost function for a segment is harder to define. In such a case, the M1 approach could instead be used in a post-processing step. To do this, the methodology can be applied to residual errors of the time series model from M0, such an approach is compared from a model fit perspective in[3].

Finally, we applied our methodology to forecasting software runtimes and saw that different benchmarks required different approaches. It would be interesting to investigate this further to see if using the benchmark attributes, as identified in[23], as a covariate indicates which approach performs best.

7 Acknowledgements

The authors are grateful to Idris A. Eckley for comments on an early draft of this work. Jamie-Leigh Chapman was supported by the STOR-i Doctoral Training Centre under grant EP/H023151/1.

References

[1] Rodgers J. Gracey M. Martial F.P. Wynne J. Ryan S. Twining C.G. Cootes T.F. Killick R. Lucas R.J. Storchi, R. Measuring vision using innate behaviours in mice with intact and impaired retina function. *Scientific Reports*, 9(10396), 2019. doi: 10.1038/s41598-019-46836-y. URL https://doi.org/10.1038/s41598-019-46836-y.

[2] J. Rachel Carr, Heather Bell, Rebecca Killick, and Tom Holt. Exceptional retreat of novaya zemlya's marine-terminating outlet glaciers between 2000 and 2013. *The cryosphere.*, 11(5):2149–2174, September 2017. URL http://dro.dur.ac.uk/23156/.

[3] Claudie Beaulieu and Rebecca Killick. Distinguishing trends and shifts from memory in climate data. *Journal of Climate*, 31 (23):9519–9543, 2018. doi: 10.1175/JCLI-D-17-0863.1. URL https://doi.org/10.1175/JCLI-D-17-0863.1.

[4] A. Nazareth, R. Killick, A.S. Dick, and S.M. Pruden. Strategy selection versus flexibility: Using eye-trackers to investigate strategy use during mental rotation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(2):232–245, 2019.

[5] Catalin Starica and Clive Granger. Nonstationarities in stock returns. *The Review of Economics and Statistics*, 87(3):503–522, August 2005. URL https://ideas.repec.org/a/tpr/restat/v87y2005i3p503-522.html.

[6] M Hashem Pesaran and Allan Timmermann. Market timing and return prediction under model instability. *J. Empirical Finance*, 9(5):495–510, 2002.

[7] Todd E Clark and Michael W McCracken. The power of tests of predictive ability in the presence of structural breaks. *J. Econometrics*, 124(1):1–31, 2005.

[8] Graham Elliott. Forecasting when there is a single break. *Manuscript, University of California at San Diego*, 2005.

[9] M Hashem Pesaran and Allan Timmermann. How costly is it to ignore breaks when forecasting the direction of a time series? *Int. J. Forecasting*, 20(3):411–425, 2004.

[10] Robert L Brown, James Durbin, and James M Evans. Techniques for testing the constancy of regression relationships over time. *J R Stat Soc Series B Stat Methodol*, pages 149–192, 1975.

[11] M Hashem Pesaran and Allan Timmermann. Selection of estimation window in the presence of breaks. *J. Econometrics*, 137(1):134–161, 2007.

[12] Rebecca Killick, Paul Fearnhead, and Idris Eckley. Optimal detection of changepoints with a linear computational cost. *J. Am. Stat. Assoc.*, 107(500): 1590–1598, 2012.

[13] Ben Norwood and Rebecca Killick. Long memory and changepoint models: a spectral classification procedure. *Stat. Comput.*, 28(2):291–302, 2018.

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

[15] Jushan Bai and Pierre Perron. Estimating and testing linear models with multiple structural changes. *Econometrica*, pages 47–78, 1998.

[16] Rob J Hyndman, Yeasmin Khandakar, et al. *Automatic time series for forecasting: the forecast package for R*. Number 6/07. Monash University, Department of Econometrics and Business Statistics, 2007.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2018. URL `https://www.R-project.org/`.

[18] Rebecca Killick and Idris Eckley. changepoint: An r package for changepoint analysis. *J. Stat. Softw*, 58(3):1–19, 2014.

[19] Gideon Schwarz et al. Estimating the dimension of a model. *Ann. Statist.*, 6 (2):461–464, 1978.

[20] Nancy R Zhang and David O Siegmund. A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):22–32, 2007.

[21] T. Kley, P. Preuss, and P. Fryzlewicz. Predictive, finite-sample model choice for time series under stationarity and non-stationarity. stats.lse.ac.uk/fryzlewicz/articles.html, 2017.

[22] Akamai. Akamai online retail performance report: Milliseconds are critical, 2017. URL `https://www.akamai.com/uk/en/about/news/press/2017-press/akamai-re` Accessed 28/11/2019.

[23] E. Barrett, C.F. Bolz, R. Killick, S. Mount, and L. Tratt. Virtual machine warmup blows hot and cold. In *ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages and Applications*, 2017.