Breaks For Additive Season and Trend (BFAST)

Theoretical background and results

Dmitry Serykh
Department of Computer Science

Introduction

- 7.5 ECTS project from Block 5
- This presentation:
  - Introduces Breaks For Additive Season and Trend (BFAST) sturcutral change detection algorithm (not to be confused with BFAST Monitor)
  - Includes examples of BFAST application to multiple datasets
  - Describes the steps of the BFAST algorithm
  - Elaborates on most important steps of the algorithm and the underlying theory

BFAST

- In the paper from 2010, Verbesselt et. al outline a generic change detection approach that combines iterative decomposition into trend, seasonal and remainder components and detection and characterizing of breakpoints within the trend and seasonal components.
- For BFAST, we consider a following data model:

$$Y_t = T_t + S_t + R_t, \quad t = 1, ..., n$$

  where:
  - Y_t is the observation at time t
  - T_t is the trend component at time t
  - S_t is the seasonal component at time t
  - R_t is the remainder component at time t
  - n is the number of observations in the time series

BFAST Example - `harvest`

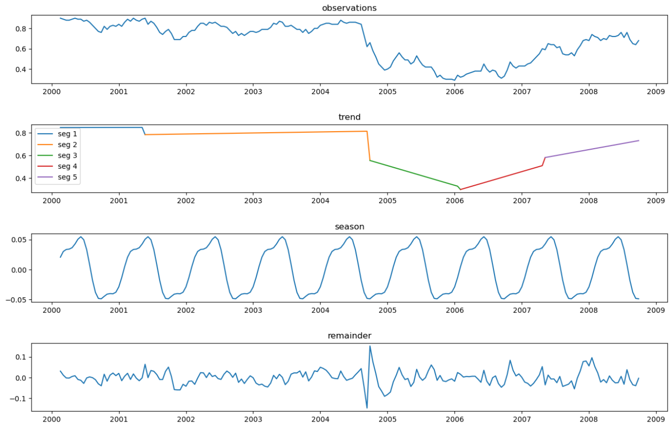


Figure: Normalized Difference Vegetation Index (NDVI) time series for a pine plantation, with 23 observation per year. NDVI is an estimate of the density of green on an area of land. There are 4 breakpoints in the trend component. Harmonic seasonal model

BFAST Example - `simts`

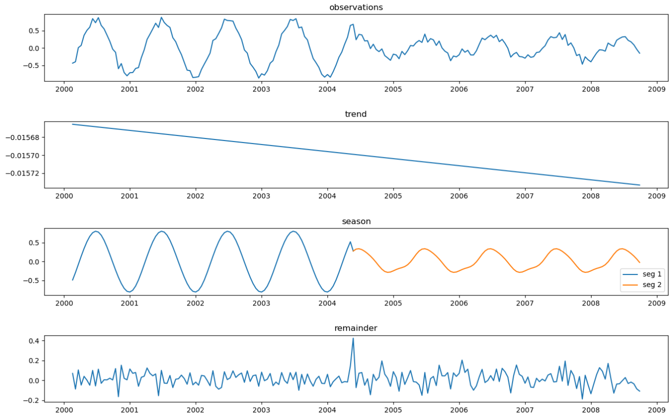


Figure: Simulated seasonal 16-day NDVI time series. The seasonal component has a single breakpoint. Harmonic seasonal model

BFAST Example - `nile`

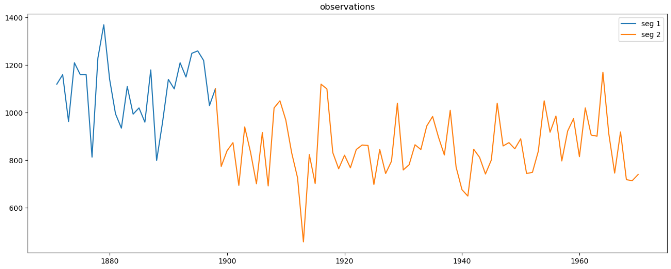


Figure: Measurements of the annual flow of the river Nile with apparent breakpoint near 1898 when the dam was built. There is no seasonal component, since there is one observation each year

BFAST Example - `ndvi`

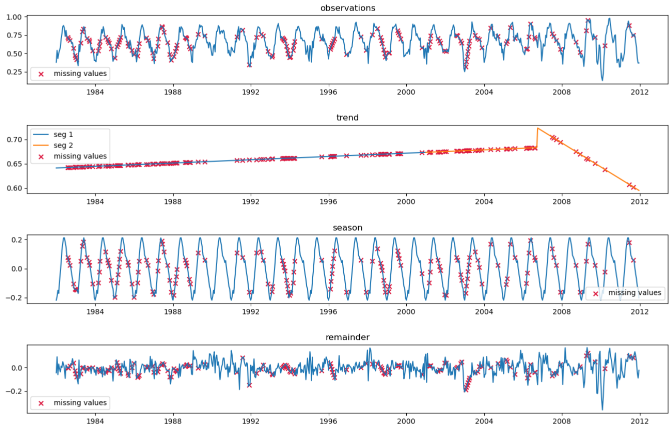


Figure: A random NDVI time series with missing values. Frequency is set to 24. There is a single breakpoint in the trend component. “dummy” seasonal model.

Trend Component

- We assume that T_t is piecewise linear and has breakpoints $t_1^*, \ldots, t_m^*$, where m is the number of breakpoints in the trend component and set $t_0^* = 0$. Then, the trend component can be described as

$$T_t = \alpha_j + \beta_j t \quad \text{for} \quad t_{j-1}^* < t \leq t_j^*$$

  where:
  - j is the number of the next breakpoint, i.e. $j = 1, ..., m$.
  - α_j and β_j are the corresponding linear coefficients
- One of the steps of the BFAST algorithm is to determine whether breakpoints are occurring within the trend component. The test is based on Ordinary Least Squares (OLS) regression, where we assume that there are no breakpoints in T and apply the OLS to the following model:

$$\mathrm{T} = \mathrm{X}_T \beta$$

$$\mathrm{T} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{bmatrix} \quad \mathrm{X}_T = \begin{bmatrix} 1 & T_1 \\ 1 & T_2 \\ \vdots & \vdots \\ 1 & T_n \end{bmatrix} \quad \beta = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix}$$

Trend Component - Continued

- Another step of BFAST algorithm is to estimate the number and position of breakpoints in the trend component. The resulting estimated $\hat{m}$ breakpoints $\hat{t}_1^*, \ldots, \hat{t}_{\hat{m}}^*$ can then be used to reconstruct the estimate of the trend component $\hat{T}$:

$$\hat{T} = \bar{X}_T \bar{\beta}$$

$$\bar{X}_T = \begin{bmatrix} 1 & T_{\hat{t}_0^*} & 0 & 0 & \ldots & 0 \\ 1 & T_{\hat{t}_0^*+1} & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & T_{\hat{t}_1^*} & \ldots & 0 \\ 0 & 0 & 1 & T_{\hat{t}_1^*+1} & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 & T_n \end{bmatrix} \quad \bar{\beta} = \begin{bmatrix} \alpha_1 \\ \beta_1 \\ \vdots \\ \alpha_{\hat{m}} \\ \beta_{\hat{m}} \end{bmatrix}$$

where $\bar{X}_T$ is a $(n \times 2\hat{m})$ partition matrix

Seasonal Component

- There are two seasonal models that were introduced by Verbesselt et al.: Harmonic and “dummy”. This presentation only covers the former.
- The breakpoints in the seasonal component can occur at different times than the breaks in the trend component. Let $t_1^{\#}, \dots, t_p^{\#}$ be the breakpoints in the seasonal component, where p is the number of breakpoints and $t_0^{\#} = 0$.
- For $t_{j-1}^{\#} < t \leq t_j^{\#}$, the seasonal term can be expressed as:

$$S_t = \sum_{k=1}^{K} \left[\gamma_{j,k} \sin\left(\frac{2\pi kt}{s}\right) + \theta_{j,k} \cos\left(\frac{2\pi kt}{s}\right) \right]$$

  where

  - K is the harmonic term, i.e. the number of pairs of harmonic terms: ($K = 3$ is used in BFAST)
  - $\gamma_{j,k}$ and $\theta_{j,k}$ are the seasonal coefficients
  - t is the observation time.
  - s is the period of seasonality (e.g. number of observations per year)

Seasonal Component - Continued

- Similarly to the Trend component, we apply the OLS to the model:

$$S = X_S \omega$$

$$X_S = \begin{bmatrix} \sin {}^{2\pi}/_{f} & \cos {}^{2\pi}/_{f} & \sin {}^{4\pi}/_{f} & \cos {}^{4\pi}/_{f} & \sin {}^{6\pi}/_{f} & \cos {}^{6\pi}/_{f} \\ \sin {}^{4\pi}/_{f} & \cos {}^{4\pi}/_{f} & \sin {}^{8\pi}/_{f} & \cos {}^{8\pi}/_{f} & \sin {}^{12\pi}/_{f} & \cos {}^{12\pi}/_{f} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sin {}^{2\pi n}/_{f} & \cos {}^{2\pi n}/_{f} & \sin {}^{4\pi n}/_{f} & \cos {}^{4\pi n}/_{f} & \sin {}^{6\pi n}/_{f} & \cos {}^{6\pi n}/_{f} \end{bmatrix}$$

$$\omega = (\gamma_1, \theta_1, \gamma_2, \theta_2, \gamma_3, \theta_3)^\top$$

- At a later step of the algorithm, we reconstruct the estimate of the seasonal component from the $\hat{p}$ estimated breakpoints $t_1^\#, \ldots, t_{\hat{p}}^\#$

$$\hat{S} = \bar{X}_S \bar{\omega}$$

where $\bar{X}_S$ is the $(n \times 6\hat{p})$ partition matrix that is built in the same way as $\bar{X}_T$ and $\bar{\omega}$ is the vector of harmonic parameters of length $6\hat{p}$.

BFAST Algorithm Steps

- **Estimate S_t using STL decomposition, resulting in $\hat{S}_t$**
- Iterate, until the number and position of the breakpoints do not change during the iteration or the maximum allowed number of iterations is reached:
  1. **Calculate the deasonalized time series:** $V_t = Y_t - \hat{S}_t$
  2. **Apply the OLS-MOSUM test** to $x = \mathrm{X}_T$ and $y = V_t$ If the returned p-value is lower than the significance level α, **estimate the number and position of the trend components $\hat{t}_1^*, \dots, \hat{t}_{\hat{m}}^*$** using the breakpoint estimation algorithm by Bai and Perron.
  3. **Compute the trend coefficients** α_j and β_j for $j = 1, \dots, m$ as vector $\bar{\beta}$ using OLS regression. Then build a partition matrix $\bar{\mathrm{X}}_T$. Set the trend estimate $\hat{T}_t = \bar{\mathrm{X}}_T \bar{\beta}$
  4. **Calculate the detrended time series**: $W_t = Y_t - \hat{T}_t$
  5. **Apply the OLS-MOSUM test** to $x = \mathrm{X}_S$ and $y = W_t$. If the test show that the breakpoints are present in the seasonal data, **estimate the number and position of the breakpoints in the seasonal component** $t_1^{\#}, \dots, t_{\hat{p}}^{\#}$ using the breakpoint estimation algorithm
  6. **Compute the coefficients for the seasonal component and reconstruct** $\hat{S}_t = \bar{\mathrm{X}}_S \bar{\omega}$
- **Return** $\hat{T}_t$, $\hat{S}_t$, $\hat{R}_t = Y_t - \hat{T}_t - \hat{S}_t$, $(\hat{t}_1^*, \dots, \hat{t}_{\hat{m}}^*)$, $(\hat{t}_1^{\#}, \dots, \hat{t}_{\hat{p}}^{\#})$

Ordinary Least Squares (OLS) Regression - Quick Recap

- A crucial component of BFAST
- Linear Model

$$\mathrm{Y} = \mathrm{X}\beta + \varepsilon$$

- Wish to find a solution to a quadratic minimization problem

$$\hat{\beta} = \mathsf{argmin}_{\beta} \| Y - \mathrm{X}\beta \|^2$$

- $\hat{\beta}$ is the OLS estimator for β and can be found using the explicit formula:

$$\hat{\beta} = \left(\mathrm{X}^\top \mathrm{X}\right)^{-1} \mathrm{X}^\top \mathrm{Y}$$

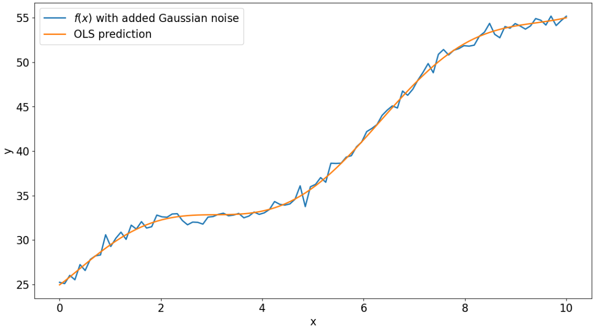
- A numerically stable solution can be obtained using QR-decomposition or Moore-Penrose pseudoinverse of X (an expansive topic in itself).

OLS-Regression for Non-linear Functions

- Linear regression can be used to estimate linear parameters for non-linear functions.
- E.g. $f(x) = 25 + 2x^{1.2} + 3\sin x$, then:

$$\mathrm{X} = \begin{bmatrix} 1 & x_1^{1.2} & \sin x_1 \\ 1 & x_2^{1.2} & \sin x_2 \\ \vdots & \vdots & \vdots \\ 1 & x_n^{1.2} & \sin x_n \end{bmatrix} \quad \beta = \begin{bmatrix} 25 \\ 2 \\ 3 \end{bmatrix}$$

OLS Example


$f(x)$ with added Gaussian noise
OLS prediction
y
x
55
50
45
40
35
30
25
0
2
4
6
8
10

Seasonal and Trend decomposition using LOESS (STL) - Intro

- STL, as first described by Cleveland et al. is a decomposition of a time series (Y_v) into a trend (T_v), seasonal (S_v) and remainder (R_v) s.t:

$$Y_v = T_v + S_v + R_v \text{ for } v \in 1 \ldots N$$

- STL is used to find the initial estimate of the seasonal component ($\hat{S}$) in BFAST

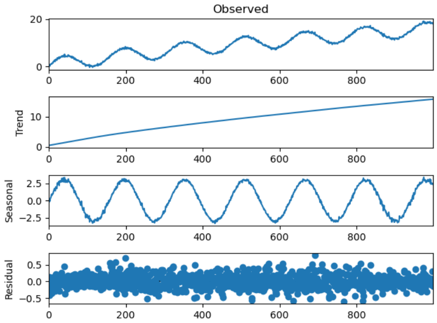


Figure: $f(x) = x^{0.75} + 2\sin(x)$ with added noise $\sim \mathcal{N}(0, 0.25)$

Locally Estimated Scatterplot Smoothing (LOESS)

- For all x in the time series fit a curve $\hat{g}(x)$ by giving the other points x_i a weight v_i.
- Select the value of the smoothing factor $q \in \mathbb{Z}^+$ and let $\lambda_q(x)$ be the distance from x to q'th closest x_i. For $q > n$:

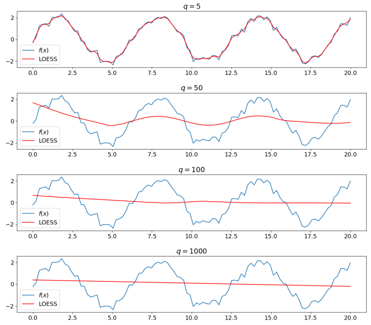
$$\lambda_q(x) = \frac{q \cdot \lambda_n(x)}{n}$$

- We calculate the weights using the tricube weight function:

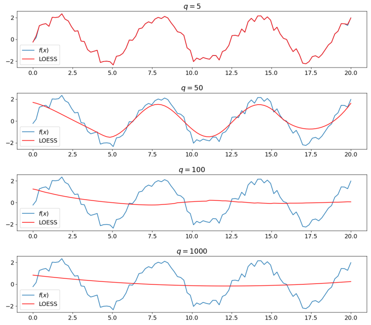
$$v_i = \left(1 - \left(\frac{|x_i - x|}{\lambda_q(x)}\right)^3\right)^3$$

  for $|x_i - x| \geq \lambda_q(x)$, set $v_i = 0$
- Use local weighted least squares regression with weights v_i to calculate $\hat{g}(x)$.

LOESS Examples



(a) $d = 1$



(b) $d = 2$

Figure: LOESS applied to $f(x) = 2\sin(x)$ with added Gaussian noise and $n = 100$. d is the degree of the fitted polynomial, q is the smoothing factor, when $q \to \infty$, LOESS would be equivalent to an ordinary OLS fit of degree d

STL - Algorithm Steps

- We begin by setting:

$$T_v^0 = 0, \quad R_v^0 = 0, \quad \rho_v = 1$$

- Then for k in $(0, \ldots, n_{\text{iter}} - 1)$:
  1. **Detrending**: $V_v = Y_v - T_v^k$, where k is the iteration number of the inner loop
  2. **Cycle-Subseries Smoothing**: V_v is split into cycle-subseries, calculate the mean average subseries resulting in C^{k+1}.
  3. **Low-pass Filter of Smoothed Cycle-Subseries**: Apply the low-pass filter to C_{k+1}. This is accomplished by application of two moving averages (convolutions) of length n_p, followed by another moving average of length 3, followed by LOESS smoothing with $q = n_l$ and $d = 1$. The result is saved as L^{k+1}
  4. **Detrending of the Smoothed Cycle-Subseries**: $S^{k+1} = C^{k+1} - L^{k+1}$
  5. **Deseasoning**: $W_v = Y_v - S_v^{k+1}$.
  6. **Trend Smoothing**: Apply LOESS to W_v with $q = n_t$, resulting in T^{k+1}
- Return $T_v^{n_{\text{iter}}}$, $S_v^{n_{\text{iter}}}$ and $R_v^{n_{\text{iter}}} = Y_v - T_v^{n_{\text{iter}}} - S_v^{n_{\text{iter}}}$

OLS-MOSUM Test - The Model

- One step of the BFAST algorithm is to detect structural change in the trend and seasonal components before we commit to the resource-demanding estimation of the number and location of the breakpoints.
- For each observation $i \in (1, \ldots, n)$ we consider a following linear model:

$$y_i = x_i^\top \beta_i + u_i$$

  where:
  - $x_i = (1, x_{i2}, x_{i3}, ..., x_{ik})^T \in \mathbb{R}^k$
  - $u_i \in \mathbb{R}$ is the residual term that is independently and identically distributed with mean $\mu = 0$ and variance σ^2.
- We can then test for structural change by testing the null hypothesis:

$$H_0 : \quad \beta_i = \beta_0 \quad (i = 1, \ldots, n)$$

OLS-MOSUM Test - Continued

- $\hat{\beta}^{(n)}$ is the ordinary least squares (OLS) estimate of the regression coefficients based on all the observations up to n,
- Let the OLS residuals (estimates of u_i) be defined as:

$$\hat{u}_i = y_i - x_i^T \hat{\beta}^{(n)} \tag{1}$$

the variance estimate would then be:

$$\hat{\sigma}^2 = \frac{1}{n-k} \sum_{i=1}^{n} \hat{u}_i^2 \tag{2}$$

Empirical Fluctuation Process (OLS-MOSUM)

- It is possible to detect structural change by analyzing moving sum of residuals ($\hat{u}$)
- The resulting empirical fluctuation process consists of a sum of a fixed number of residuals in a data interval, which size is determined by the value of the parameter $h \in (0,1)$ (bandwidth).
- The OLS-based MOSUM process at time t is given by:

$$M(t) = \frac{1}{\hat{\sigma}\sqrt{n}} \left(\sum_{i=\lfloor N_n t \rfloor + 1}^{\lfloor N_n t \rfloor + \lfloor nh \rfloor} \hat{u}_i \right) \quad (0 \leq t \leq 1 - h) \tag{3}$$

where $N_n = (n - \lfloor nh \rfloor)/(1 - h)$ and $\lfloor nh \rfloor$ is the size of the window.

Significance Testing

- A key observation is that if a structural change takes place at t_0, the OLS-MOSUM path would also have a strong shift at t_0. We reject the null hypothesis of there being no structural change, when the fluctuation of the OLS-MOSUM process becomes too large.
- In practice, we determine whether the null hypothesis can be rejected using a significance test (also called statistical hypothesis testing).
- First, we calculate the test statistic. For the residual-based OLS-MOSUM process, it is defined as:

$$S_{\text{MOSUM}} = \max(|M(t)|) \quad \text{for } 0 \leq t \leq (1-h) \tag{4}$$

Significance Testing - Continued

- This formulation is not usable in a context of an implementation, since we are working with infinite set of real numbers from 0 to $1-h$.
- Another key observation is that $M(t)$ returns $n-\lfloor nh \rfloor+1$ unique values for $0 \leq t \leq (1-h)$:
- Let:

$$\bar{M}(t') = \frac{1}{\hat{\sigma}\sqrt{n}} \left(\sum_{i=t}^{t'+\lfloor nh \rfloor} \hat{u}_i \right) \quad (t' = 1, 2, \ldots, n-\lfloor nh \rfloor+1)) \tag{5}$$

- Then: $\max(|M(t)|) = \max(|\bar{M}(t')|)$ and we have:

$$S_{\text{MOSUM}} = \max(|\bar{M}(t')|) \quad \text{for } t' \text{ in } 1, 2, .., (n-\lfloor nh \rfloor+1) \tag{6}$$

Significance Testing - Continued 2

- We use the value of S_{MOSUM} to calculate the probability of getting such sample, given that the null hypothesis holds, using the critical values approach. The p-value is calculated from the table of simulated asymptotic critical values of the Moving Estimate (ME) tests with the maximum norm, given by Chu et al. (1995).
- We then compare the resulting probability with a chosen confidence interval, which is a value $0 < \alpha < 1$ (typically the value of α is around 0.05)
- If the resulting probability is below the value of α, we reject the null hypothesis, hence a structural change is detected in the time series.

OLS-MOSUM Test - Steps of the Algorithm

1. **Calculate $\hat{\beta}^{(n)}$ using OLS-based linear regression from matrix X and vector y**
2. **Calculate the vector of OLS residuals $\hat{\mu}$**
3. **Calculate the standard deviation $\hat{\sigma}$**
4. **Calculate the residual-based OLS-MOSUM process as a vector of size $n - \lfloor nh \rfloor + 1$**

```
nh = floor(n * h)
e = concat([0], residuals)
process = cumsum(e)
process = process[nh:n] - process[0:(n - nh + 1)]
process = process / (sigma * sqrt(n))
```

5. **Calculate the test statistic $S_{\textbf{test}}$** = `max(abs(process))`
6. **Calculate the p-value using the table of critical values and linear interpolation**
7. **Return the result**:
   If $p \leq \alpha$, we reject the null-hypothesis. Structural change is detected

Breakpoint Estimation - Intro

- Described in the paper by Bai and Perron from 2003
- Estimate the number and position of breakpoints in a time series using a dynamic programming algorithm and Bayesian Information Criterion (BIC)

Breakpoint Estimation - The Model

- We assume a pure structural change model for m breaks ($m+1$ segments):
$$y_t = x_t^\top \beta_j + u_t \quad t = T_{j-1}+1, \ldots, T_j$$
for $j = 1, \ldots, m+1$, and where:
  - $x_t \in \mathbb{R}^q$ is the value of the independent variable at time $t = 1, \ldots, T$
  - $y_t \in \mathbb{R}$ is the observation at time t
  - $\beta_j : \ (j = 1, \ldots, m+1)$ is the vector of coefficients for the segment j
  - $u_t \in \mathbb{R}$ is the disturbance (error) at time t
  - $(T_1, \ldots, T_m)$ are the unknown indices of the m breakpoints. We additionally set $T_0 = 0$ and $T_{m+1} = T$
- In other words, we split the time series into $m+1$ segments (of potentially different sizes) and perform linear regression independently for each segment, where for segment i, we estimate a coefficient vector β_i. Hence, we find the estimate of the coefficient vector $(\hat{\beta})$ for each m-partition $(T_1, ..., T_m)$ by minimizing the sum of squared residuals:

$$\sum_{i=1}^{m+1} \sum_{i=T_{i-1}+1}^{T_i} \left[y_t - x_t^\top \beta_i \right]^2$$

- Let $\{T_j\}$ denote an m-partition $(T_1, ..., T_m)$ and $S_T(T_1, ..., T_m)$ denote the resulting sum of squared residuals. Since we can estimate the coefficient vector for each partition, we can minimize the sum of squared residuals by finding the optimal position for the breakpoints

$$(\hat{T}_1, ..., \hat{T}_m) = \operatorname{argmin}_{T_1,...,T_m} S_T(T_1, \ldots, T_m)$$

- There is a finite number of possible breakpoints, and only a subset of possible partitions is feasible. There are $T(T+1)/2$ (sum of integers from 1 to T) possible segments that can be chosen. This can be demonstrated by building a matrix of possible segments, with starting date on the y-axis and terminal date on the y-axis. A length of a segment is positive, hence we can eliminate one half of the potential segments. There are further reductions that can be made that reduce the number of feasible segments to $T(T+1)/2 - \big(T(h-1) - mh(h-1) - (h-1)^2 - h(h-1)/2\big)$, where h is the minimal segment length.

Upper-diagonal Matrix of Sums of Squared Residuals

stop

start	1	2	3	4	5	6	7	8	9
1	nf_1	nf_1	nf_1	f	f	f	nf_2	nf_2	nf_2
2		nf_1	nf_1	nf_1	nf_3	nf_3	nf_2	nf_2	nf_2
3			nf_1	nf_1	nf_1	nf_3	nf_2	nf_2	nf_2
4				nf_1	nf_1	nf_1	f	f	f
5					nf_1	nf_1	nf_1	f	f
6						nf_1	nf_1	nf_1	f
7							nf_1	nf_1	nf_1
8								nf_1	nf_1
9									nf_1

Table: An example of an upper-triangular matrix of sums of squared residuals for $T = 9$, $h = 3$, $m = 2$. We must compute sum of squared residuals for all feasible segments (f).

Recursive Residuals

- In order to calculate the sum of squared residuals for each feasible segment in the upper-triangular matrix, me must use the recursive residuals of a linear regression model. Let us assume a simple regression model with no breakpoints

$$y_i = x_i^\top \beta + u_i \quad i = (1, \dots, n)$$

$$Y = \mathrm{X}\beta + U$$

where Y, U are $n \times 1$ vectors, X is a $(n \times k)$ matrix and β is $k \times 1$ vector.

- Let $\hat{\beta}^{(i)}$ be the OLS-estimate of β based on all observations up to i, and $X^{(i)}$ be the subset of the X with rows from k to i. Then the recursive residuals can be calculated:

$$v(i) = \frac{y_i - x_i^\top \hat{\beta}^{(i-1)}}{\sqrt{1 + x_i^\top \left(X^{(i-1)\top} X^{(i-1)}\right)^{-1} x_i}} \quad (i = k, \dots, n)$$

Calculating the SSR Table

Since we are considering a pure structural change model, there are no constraints between the segments and we can therefore apply OLS for each segment independently.
Let SSR(i,j) be the sum of squared residuals, acquired from application of OLS to segment that starts at i and ends at j. It can be calculated using a recursive procedure:

$$\text{SSR}(i,j) = \text{SSR}(i,j-1) + v(i)^2$$

We can calculate all row in the table in parallel, and each row can be calculated in a following manner:

```
betas = map (\i -> OLS(X[k:i], y[k:i])) [k:n]
SSR_ROW(i) = scan (+) 0 (map2 (\j b -> v(j, b) ** 2) [k:i] betas)
```

Dynamic Programming Algorithm

- After the table of squared residuals is constructed, we can apply the dynamic programming algorithm in order to find the optimal partition that solves the following recursive problem:

$$\mathrm{SSR}\left(\{T_{m,T}\}\right) = \min_{mh \le j \le T-h} \left[\mathrm{SSR}\left(\{T_{m-1,j}\}\right) + \mathrm{SSR}(j+1, T)\right]$$

  where m is the number of breakpoints, T is the number of observations, h is the minimal segment length (distance between two breakpoints) and $\mathrm{SSR}\left(\{T_{r,n}\}\right)$ be the sum of squared residuals associated with the optimal partition that contains r breaks, using the first n observations.
- We can unroll the recursive calls in order to gain more insight into the workings of this method and get a following expression:

$$\begin{gathered} SSR\left(\{T_{m,T}\}\right) = \\ \min_{mh \le j_1 \le T-h}\left[SSR\left(j_1+1, T\right) + \right. \\ \min_{(m-1)h \le j_2 \le j_1-h}\left[SSR\left(j_2+1, j_1\right) + \right. \\ \vdots \\ \min_{h \le j_m \le j_{m-1}-h}\left[\mathrm{SSR}\left(1, j_m\right) + \mathrm{SSR}\left(j_m+1, j_{m-1}\right)\right] \ldots]]] \end{gathered}$$

Dynamic Programming Algorithm - Continued

- We start the procedure by solving the minimization problem from the last line of the expression. We evaluate the optimal one-break partition for all sub-samples that allow a possible break ranging from observations h to $T-mh$ (since we still have to fit m segments of minimal size h after the first breakpoint). We then store $T-(m+1)h+1$ optimal breakpoints along with the corresponding sums of squared residuals. These one-break partitions would have an endpoint between $2h$ and $T-(m-1)h$.
- Then, we continue to 2-partitions that can have and endpoint between $3h$ and $T-(m-2)h$. For each of $T-(m+1)h+1$ endpoints, we select an 1-partition that we have calculated earlier that would minimize the sum of squared residuals. We then store the 2-partitions and corresponding sums of squared residuals.
- We iterate until we have computed the $T-(m+1)h+1$ $(m-1)$-partitions with endpoints between $(m-1)h$ and $T-2h$. We finish by finding the optimal $(m-1)$ partition that gives the overall minimal sum of squared residuals, when the last segment is appended to it.

Example

Consider this example. Let $T = 20$, $h = 4$ and $m = 3$.

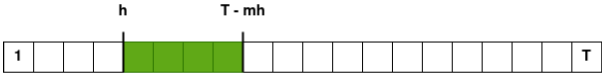


Figure: Allowed placements of j_m

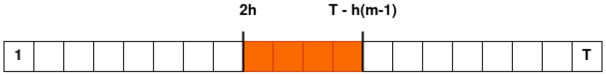


Figure: Possible endpoints of 1-partion

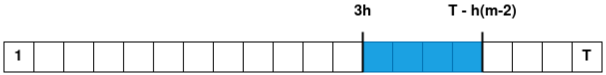


Figure: Possible start points of the last segment

BIC

- For each possible number of breakpoints, up to some limit $m_{\max}$, we find the sum of squared residuals for all breakpoints and then calculate the Bayesian Information Criterion:

  $$\text{BIC} = n_{\text{df}} \ln n - 2\hat{L}$$

  where:
  - n_{df}: the number of degrees of freedom for our model, in our case $n_{\text{df}} = (k+1)(m+1)$, where k is the dimensionality of the vector x_i and m is the number of breakpoints.
  - n is the number of observations
  - $\hat{L}$ is the maximized value of the log-likelihood of the model:

    $$\hat{L} = -\frac{n}{2}\left(\ln\left(S_T(\hat{T}_1, .., \hat{T}_m)\right) + \ln\frac{2\pi}{n} + 1\right)$$

    where $S_T(\hat{T}_1, .., \hat{T}_m)$ is the sum of squared residuals for the optimal m-partition of the time series (time series with m breakpoints).
- The optimal number of breakpoints would be the number of breakpoints for which the value of BIC is lowest.

Algorithm Steps

1. **Calculate the upper-triangular matrix of sums of squared residuals**
2. **Find the largest number of allowed breakpoints**
   Given a value of $0 \leq h' \leq 1$, which signifies the smallest allowed size of a segment, and m_{user} which is the maximum number of breakpoints, determined by the user, we can then find the largest number of allowed breakpoints $m_{\text{max}} = \min(\lfloor \frac{1}{h'} \rfloor + 1, m_{\text{user}})$.
3. **Create a table `BIC` of size $m_{\textbf{max}} + 1$**
4. **For $m \in (m_{\textbf{max}}, ...0)$, do following:**
   1. Find the optimal position of the m breakpoints, using the dynamic programming algorithm
   2. Calculate the sum of squared residuals, and then use it to calculate the Bayesian Information Criterion BIC_m of this partition
   3. Add BIC_m to the table
5. **Find the value of m_{best} that corresponds to the minimal value in the `BIC`**
6. **Use the dynamic programming algorithm to find the m_{best} breakpoints**

Links

- Full project report can be found here: `https://raw.githubusercontent.com/mortvest/bfast-py/master/report/main.pdf`
- The source code can be found in the GitHub repository: `http://github.com/mortvest/bfast-py`

Questions

?