R Fit a Time Series with Polynomial and Analytical Expressions for Coefficients

Fan Wang

2021-08-11

Contents

1 Polynomial Time Series

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

1.1 Analytical Solution for Time-Series Polynomial Coefficients

This file is developed in support of Bhalotra, Fernandez, and Wang (2021), where we use polynomials to model share parameter changes over time in the context of a nested-CES problem.

1.1.1 Mth Order Polynomial and its Derivative

We have a polynomial of Mth order:

$$y\left(t\right) = a_0 + a_1 \cdot t + a_2 \cdot t^2 + ... + a_M \cdot t^M$$

Taking derivative of of y with respect to t, we have:

$$\begin{aligned}
\frac{d^{M-0}y}{dt^{M-0}} &= a_{M-0} \cdot \frac{(M-0)!}{0!} \cdot t^{-0+0} \\
\frac{d^{M-1}y}{dt^{M-1}} &= a_{M-0} \cdot \frac{(M-0)!}{1!} \cdot t^{-0+1} + a_{M-1} \cdot \frac{(M-1)!}{0!} \cdot t^{-1+1} \\
\frac{d^{M-2}y}{dt^{M-2}} &= a_{M-0} \cdot \frac{(M-0)!}{2!} \cdot t^{-0+2} + a_{M-1} \cdot \frac{(M-1)!}{1!} \cdot t^{-1+2} + a_{M-2} \cdot \frac{(M-2)!}{0!} \cdot t^{-2+2} \\
\frac{d^{M-3}y}{dt^{M-3}} &= a_{M-0} \cdot \frac{(M-0)!}{3!} \cdot t^{-0+3} + a_{M-1} \cdot \frac{(M-1)!}{2!} \cdot t^{-1+3} + a_{M-2} \cdot \frac{(M-2)!}{1!} \cdot t^{-2+3} + a_{M-3} \cdot \frac{(M-3)!}{0!} \cdot t^{-3+3}
\end{aligned}$$

Given the structure above, we have the following formulation for polynomial derivative, where we have a polynomial of M^{th} order and we are interested in the N^{th} derivative:

$$\frac{d^N y(M)}{dt^N} = \sum_{i=0}^{M-N} \left(a_{M-i} \cdot \frac{(M-i)!}{(M-N-i)!} \cdot t^{M-N-i} \right)$$

Note that the M^{th} derivative of a M^{th} order polynomial is equal to $(a_M \cdot M!)$

1.1.2 Repeatedly Taking Differences of Differences

Suppose the data generating process is a M^{th} order polynomial, then differencing of time-series observables can be used to identify polynomial coefficients.

In the simplest case of a 1st order polynomial with $Y_t = a_0 + a_1 \cdot t$, given 2 periods of data, $t = 0, t = 1$, we have $a_1 = Y_2 - Y_1$, and $a_0 = Y_0$. The rate of change in Y over t captures coefficient a_1.

For a second order polynomial, while the first derivative is time-varying, the second derivative, acceleration, is invariant over t. Similarly, for any M^{th} order polynomial, the M^{th} derivative is time-invariant.

Because of this time-invariance of the M^{th} derivative, the differencing idea can be used to identify a_M, the coefficient for the highest polynomial term for the M^{th} order polynomial.

Now we difference observable y_t overtime. We difference the differences, difference the differences of the differences, and difference the differences of the differences of the differences, etc. It turns out that the difference is a summation over all observations $\{y_t\}_{t=1}^{T}$, with the number of times each y_t term appearing following Pascal's Triangle.

Specifically:

- 1st difference is $y_1 - y_0$
- 2nd difference is $(y_2 - y_1) - (y_1 - y_0) = y_2 - 2y_1 + y_0$
- 3rd difference is $((y_3 - y_2) - (y_2 - y_1)) - ((y_2 - y_1) - (y_1 - y_0)) = y_3 - 3y_2 + 3y_1 - y_0$
- 4th difference is $(((y_4 - y_3) - (y_3 - y_2)) - ((y_3 - y_2) - (y_2 - y_1))) - (((y_3 - y_2) - (y_2 - y_1)) - ((y_2 - y_1) - (y_1 - y_0))) = y_4 - 4y_3 + 6y_2 - 4y_1 + y_0$
- 5th difference is $((((y_5 - y_4) - (y_4 - y_3)) - ((y_4 - y_3) - (y_3 - y_2))) - (((y_4 - y_3) - (y_3 - y_2)) - ((y_3 - y_2) - (y_2 - y_1)))) - ((((y_4 - y_3) - (y_3 - y_2)) - ((y_3 - y_2) - (y_2 - y_1))) - (((y_3 - y_2) - (y_2 - y_1)) - ((y_2 - y_1) - (y_1 - y_0)))) = y_5 - 5y_4 + 10y_3 - 10y_2 + 5y_1 - y_0$

Note that the pattern has alternating signs, and the coefficients are binomial. We have the following formula:

$$\Delta^M = \sum_{i=0}^{M} \left((-1)^i \cdot \frac{M!}{(M-i)!i!} \right) \cdot y_{(M-i)}$$

When there are T periods of data, and we are interested in the $T - 1$ difference, the differencing formula is unique. However, for less than $T - 1$ difference, we can use alternative consecutive data segments. Specifically, given T periods of data from $t = 1$ to $t = T$, we have the notation Δ_τ^M where τ is the starting time. We have, for $M \leq T - 1$:

$$\Delta_\tau^M \left(\{y_t\}_{t=1}^{T} \right) = \sum_{i=0}^{M} \left((-1)^i \cdot \frac{M!}{(M-i)!i!} \right) \cdot y_{(\tau + (M-i))} \text{ for } 1 \leq \tau \leq T - M$$

1.1.3 Solutions for Polynomial Coefficients with Differences of Differences

Intuitively, for a M^{th} order polynomial, the coefficient on the highest polynomial term is a function of the $(M-1)^{th}$ difference. Coefficients of lower polynomial terms, $m < M$, are function of the $(m-1)^{th}$ difference along with higher order polynomial coefficients already computed.

Formally, we have, for a M^{th} order polynomial, a vector of $\{a_m\}_{m=0}^{M}$ $M + 1$ polynomial coefficients. For formula for the coefficient for the largest polynomial is:

$$a_M = \sum_{i=0}^{M} \left((-1)^i \left((M-i)!i! \right)^{-1} \right) y_{(M-i)}$$

Given this, we have also, given T periods of data from $t=1$ to $t=T$:

$$a_{M-1} = \sum_{i=0}^{M-1} \left((-1)^i \left((M-1-i)!i! \right)^{-1} \right) \cdot \left(y_{(\tau+(M-1-i))} - a_M \cdot t^M \right) \text{ for } 1 \leq \tau \leq T-M-1$$

Using one formula, given a_{m+1}, we have:

$$a_m = \sum_{i=0}^{m} \left((-1)^i \left((m-i)!i! \right)^{-1} \right) \cdot \left(y_{(\tau+(m-i))} - \sum_{j=0}^{M-m-1} a_{M-j} \cdot t^{M-j} \right) \text{ for } 1 \leq \tau \leq T-m$$

1.1.4 Identifying Polynomial Coefficients with Differences for Third Order Polynomial

To illustrate, we test the formulas with a 3rd order polynomial, and derive some 3rd-order specific formulas.

For data from a 3rd order polynomial data generating process, we can use the 3rd difference to identify the coefficient in front of x^3. With this, we can iteratively to lower polynomials and identify all relevant coefficients.

Specifically, using equations from the two sections above, we have:

$$\begin{aligned} \frac{d^3 y(3)}{dt^3} &= y_3 - 3y_2 + 3y_1 - y_0 \\ \frac{d^3 y(3)}{dt^3} &= 3! \cdot a_3 \end{aligned}$$

Combining the two equations we have, that a_3 is the 3rd difference divided by 6:

$$\begin{aligned} 3! \cdot a_3 &= y_3 - 3y_2 + 3y_1 - y_0 \\ a_3 &= \frac{y_3 - 3y_2 + 3y_1 - y_0}{3 \cdot 2} \end{aligned}$$

For the linear and cubic terms, we have:

$$\begin{aligned} \frac{d^2 y(M=3)}{dt^2} &= 3 \cdot a_2 + 6 \cdot a_3 \cdot t \\ \frac{d^1 y(M=3)}{dt^1} &= a_1 + 2 \cdot a_2 + 3 \cdot a_3 \cdot t^2 \end{aligned}$$

Note that the 3rd derivative of a 3rd order polynomial is a constant, but the 2nd derivative of a 3rd order polynomial is not. This means that to use the second difference to identify a_2 parameter, we first have to difference out from y_t the impact of the 3rd polynomial term, which we can because we know a_3 now.

Differencing out the 3rd term, we have now the 2nd derivative of a 2nd order polynomial:

$$\frac{d^2 \left(y(M=3) - a_3 \cdot t^3 \right)}{dt^2} = \frac{d^2 \left(\widehat{y}(M=2) \right)}{dt^2} \ ,$$

where $\widehat{y}\left(M,2\right) = y\left(M\right) - a_3 \cdot t^3$.

So this means we have:

$$
\begin{aligned}
\frac{d^2\widehat{y}(3,2)}{dt^2} &= (1\cdot y_2 - 2\cdot y_1 + 1\cdot y_0) - a_3\cdot\left(1\cdot 2^3 - 2\cdot 1^3 + 1\cdot 0^3\right)\\
&= (1\cdot y_2 - 2\cdot y_1 + 1\cdot y_0) - a_3\cdot\left(2^3 - 2\right)\\
\frac{d^2\widehat{y}(3,2)}{dt^2} &= 2!\cdot a_2
\end{aligned}
$$

Given the value for a_3, we have:

$$
\begin{aligned}
2!\cdot a_2 &= (1\cdot y_2 - 2\cdot y_1 + 1\cdot y_0) - a_3\cdot 6\\
2!\cdot a_2 &= (1\cdot y_2 - 2\cdot y_1 + 1\cdot y_0) - \frac{y_3 - 3y_2 + 3y_1 - y_0}{3\cdot 2}\cdot\left(2^3 - 2\right)\\
2!\cdot a_2 &= (y_2 - 2y_1 + y_0) - (y_3 - 3y_2 + 3y_1 - y_0)\\
2!\cdot a_2 &= -y_3 + 4y_2 - 5y_1 + 2y_0\\
a_2 &= \frac{-y_3 + 4y_2 - 5y_1 + 2y_0}{2}
\end{aligned}
$$

Following the same strategy, we can also find a_1. Let $\widehat{y}\,(M,1) = y\,(M) - a_3\cdot t^3 - a_2\cdot t^2$

$$
\begin{aligned}
\frac{d^2\widehat{y}(3,1)}{dt^2} &= (y_1 - y_0) - a_3\cdot\left(1^3 - 0^3\right) - a_2\cdot\left(1^2 - 0^2\right)\\
&= (y_1 - y_0) - a_3 - a_2\\
\frac{d^2\widehat{y}(3,1)}{dt^2} &= 1!\cdot a_1
\end{aligned}
$$

Hence:

$$
\begin{aligned}
a_1 &= (y_1 - y_0) - a_3\cdot\left(1^3 - 0^3\right) - a_2\cdot\left(1^2 - 0^2\right)\\
&= (y_1 - y_0) - a_3 - a_2\\
&= (y_1 - y_0) - \frac{y_3 - 3y_2 + 3y_1 - y_0}{3\cdot 2} - \frac{-y_3 + 4y_2 - 5y_1 + 2y_0}{2}\\
&= \frac{6y_1 - 6y_0}{6} - \frac{y_3 - 3y_2 + 3y_1 - y_0}{6} - \frac{-3y_3 + 12y_2 - 15y_1 + 6y_0}{6}\\
&= \frac{6y_1 - 6y_0 - y_3 + 3y_2 - 3y_1 + y_0 + 3y_3 - 12y_2 + 15y_1 - 6y_0}{6}\\
&= \frac{2y_3 - 9y_2 + 18y_1 - 11y_0}{6}
\end{aligned}
$$

Finally, we know that $a_0 = y_0$. We have now analytical expressions for each of the 4 polynomial coefficients for a 3rd order polynomial. Given data from the data-generating process, these would back out the underlying parameters of the data generating process using data from four periods at $t = 0, 1, 2, 3$.

1.1.5 Third Order Polynomial Simulation and Solving for Parameters

Now we generated a time-series of values and solve back for the underlying polynomial coefficients.

```r
# polynomial coefficients
set.seed(123)
ar_coef_poly <- rnorm(4)
# time right hand side matrix
ar_t <- 0:3
ar_power <- 0:3
mt_t_data <- do.call(rbind, lapply(ar_power, function(power) {
```

```r
  ar_t^power
}))
# Final matrix, each row is an observation, or time.
mt_t_data <- t(mt_t_data)
# General model prediction
ar_y <- mt_t_data %*% matrix(ar_coef_poly, ncol = 1, nrow = 4)
# Prediction and Input time matrix
mt_all_data <- cbind(ar_y, mt_t_data)
st_cap <- paste0(
  "C1=Y, each row is time, t=0, incremental by 1, ",
  "each column a polynomial term from 0th to higher."
)
kable(mt_all_data, caption = st_cap) %>% kable_styling_fc()
```

C1=Y, each row is time, t=0, incremental by 1, each column a polynomial term from 0th to higher.

-0.5604756	1	0	0	0
0.8385636	1	1	1	1
5.7780698	1	2	4	8
14.6810933	1	3	9	27

Backing out coefficients using the formulas for 3rd order polynomial derived above, we have:

```r
# The constant term
alpha_0 <- ar_y[1]
# The cubic term
alpha_3 <- as.numeric((t(ar_y) %*% c(-1, +3, -3, +1))/(3*2))
# The quadratic term, difference cubic out, alpha_2_1t3 = alpha_2_2t4
ar_y_hat <- ar_y - alpha_3*ar_t^3
alpha_2_1t3 <- as.numeric((t(ar_y_hat[1:3]) %*% c(1, -2, +1))/(2))
alpha_2_2t4 <- as.numeric((t(ar_y_hat[2:4]) %*% c(1, -2, +1))/(2))
alpha_2 <- alpha_2_1t3
# The linear term, difference cubic out and quadratic
ar_y_hat <- ar_y - alpha_3*ar_t^3 - alpha_2*ar_t^2
alpha_1_1t2 <- as.numeric((t(ar_y_hat[1:2]) %*% c(-1, +1))/(1))
alpha_1_2t3 <- as.numeric((t(ar_y_hat[2:3]) %*% c(-1, +1))/(1))
alpha_1_3t4 <- as.numeric((t(ar_y_hat[3:4]) %*% c(-1, +1))/(1))
alpha_1 <- alpha_1_1t2
# Collect results
ar_names <- c("Constant", "Linear", "Quadratic", "Cubic")
ar_alpha_solved <- c(alpha_0, alpha_1, alpha_2, alpha_3)
mt_alpha <- cbind(ar_names, ar_alpha_solved, ar_coef_poly)
# Display
ar_st_varnames <- c('Coefficient Counter', 'Polynomial Terms', 'Solved Coefficient Given Y', 'Actual DG
tb_alpha <- as_tibble(mt_alpha) %>%
  rowid_to_column(var = "polynomial_term_coef") %>%
  rename_all(~c(ar_st_varnames))
# Display
st_cap = paste0('Solving for polynomial coefficients.')
kable(tb_alpha, caption = st_cap) %>% kable_styling_fc()
```

Note that given that the data is exact output from DGP, and we have the same number of data and parameters, parameters are exactly identified. However, this is only really true for the a_3 parameter, which requires all four periods of data. - For a_2, it is over-identified, we can arrived at it, given a_3, either with difference of

Solving for polynomial coefficients.

Coefficient Counter	Polynomial Terms	Solved Coefficient Given Y	Actual DGP Coefficient
1	Constant	-0.560475646552213	-0.560475646552213
2	Linear	-0.230177489483281	-0.23017748948328
3	Quadratic	1.55870831414913	1.55870831414912
4	Cubic	0.0705083914245757	0.070508391424576

differences using data from $t = 1, 2, 3$ or using data from $t = 2, 3, 4$. - For a_1, it is also over-identified, given a_3 and a_2. The difference of $t = 1, 2$, $t = 2, 3$ or $t = 3, 4$ can all identify a_1.

Note also that the solution above can be found by running a linear regression as well. The point of doing the exercise here and showing analytically how layers of differences of differences identify each polynomial coefficient is to show what in the underlying variation of the data is identifying each of the polynomial term.

In effect, all identification is based on the fact that the M^{th} order polynomial's M^{th} derivative is a constant, it is invariant over t. This is the core assumption, or restriction of the otherwise highly flexible polynomial functional form. With this core invariance at the max degree derivative condition, all other parameters are obtained through the simple act of differencing.