BSTS Part 3 Script

Slide 1

Congratulations! You've made it to the moment that you were surely waiting for. Now that you understand the mysteries behind Bayesian structural time series, you will begin to master the practice of Bayseian structural time series.

Here is what you have to look forward to in this part of your adventure (-> Slide 3)

Slide 3

First, we will talk about the SST data that you have seen in previous videos.

(->) Then we will see how to use the R package *bsts*.

(->) We will see how to fit the structural time series models we saw in previous videos. This will be demonstrated for

(->) the local level model.

(->) the local linear trend model.

(->) and the local linear trend with seasonality model.

(->) We will also see how to obtain and plot the posterior distribution,

(->) forecast,

(->) and compare the three models.

We encourage you follow along! To do so, see the links in the description below. (-> Slide 4)

Slide 4

The data we will be using in this video is the sea surface temperature of the first 30 meters around Gibraltar.

(->) The data was collected every 12 days by instruments floating in the water.

(->) Here, we consider the sea surface temperature from January 2004 to November 2017.

(->) The data was obtained from the website Argovis. At Argovis.com, you can access data taken from instruments floating in the ocean all over the world. Tyler has even written a Python API so you can obtain data from your favorite ocean region!

(-> Slide 5)

Slide 5

This is what the SST data we will be working with looks like. Most of the temperatures lie between about 17 and 23 degress celcius and there is obviously some seasonality here. Let's see how we can fit some Bayesian structural time series to this data. (-> Slide 6)

Slide 6

Before we start fitting models, we need to do a little setting up. First, we load *readr* to read in the data, and of course the *bsts* package. Note that *bsts* loads several other packages as well.

We read in the data, then convert it to a time series object. We set the frequency to 30 because there are approximately 30 twelve day periods in a year. Then we produce the plot shown on the previous slide. (-> Slide 7)

Slide 7

Remember that the local level model looks like this. y_t is our data, and μ_t is the state component.

To fit models using *bsts*, we first need to construct the state. We start by creating an empty list to hold our state specification. Then we use *AddLocalLevel* to say that we want to add a level component to our state.

To fit the model, we use the function *bsts.* Along with our time series, we pass our state specification, which we just created, and the number of MCMC iterations we want from the posterior distribution. Here we only use 1,000 iterations. But this is just because we want to save time. In practice you would want a much larger posterior sample. (-> Slide 8)

Slide 8

Once we fit our model and simulate draws from the posterior, we might be interested in plotting the results. *bsts* provides several different plotting methods.

If we use *plot* with a bsts object, what we get is a plot of the posterior mean conditional on the data. The points on the plot are our original data. (-> Slide 9)

Slide 9

If we specify *components* in the plot function, we get a plot of the posterior of the state components. In this case, the local level. Based on this plot, it seems like the local level is doing a pretty good job of modeling the SST data. (-> Slide 10)

Slide 10

Another option we can give to the *plot* function is *residuals.* This of course plots the residuals. The shading here is the distribution of the residuals at each time point. Darker regions have higher density.