

sarimaTD_example

Graham Casey Gibson

3/13/2020

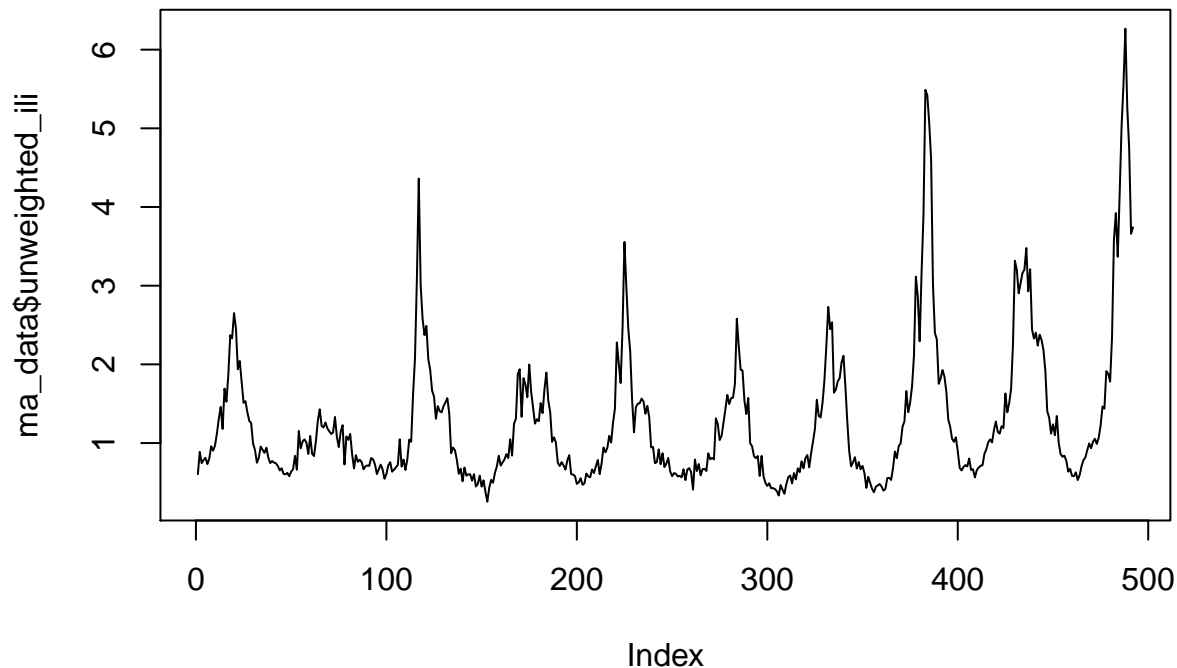
An example of cdcForecastUtils for the sarimaTD model

First we load up sarimaTD and grab the cdc ILI data for Massachusetts.

```
library(sarimaTD)
library(cdcForecastUtils)

flu_data <- download_and_preprocess_state_flu_data()

## Loading required package: cdcfluview
## Loading required package: MMWRweek
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
## Loading required package: lubridate
##
## Attaching package: 'lubridate'
## The following object is masked from 'package:base':
##
##   date
ma_data <- flu_data[flu_data$region == "Massachusetts",]
plot(ma_data$unweighted_ili,type='l')
```



We then fit sarimaTD up to the current date

```
sarimaFit <- sarimaTD::fit_sarima(tail(ma_data$unweighted_ili,200),
                                ts_frequency = 52)
```

Next, we generate a matrix of nsim till end of season. In order to do that we first get the current date and figure out how many steps there are left in the current season. We know that the season ends on YYYY-EW20.

```
current_date_in_cdc_format <- paste0(tail(ma_data$year,1), "-EW", ifelse(nchar(tail(ma_data$week,1)) == 2, t, 1))
season_end <- "2020-EW20"
time_left_in_season <- get_time_left_in_season(current_date_in_cdc_format, season_end)

preds <- simulate(
  object = sarimaFit,
  nsim = 1000,
  seed = 1,
  newdata = ma_data$unweighted_ili,
  h = time_left_in_season
)
```

We next append the observed data from the current season which we know starts on YYYY-EW40 to the predictions to create a trajectory matrix.

```
season_start <- "2019-EW40"
time_from_start_of_season <- get_time_from_start_of_season(season_start, current_date_in_cdc_format)

trajectory_matrix <- cbind(matrix(rep(tail(ma_data$unweighted_ili, time_from_start_of_season), 1000), nrow = 1000, ncol = time_from_start_of_season),
                             matrix(rep(NA, 1000 * (time_left_in_season - time_from_start_of_season)), nrow = 1000, ncol = time_left_in_season - time_from_start_of_season))
```

Finally, we convert the predicted trajectory matrix to a submission data frame.

```
submission_df <- trajectories_to_short_term_and_seasonal_binned_distributions(trajectories = trajectory_matrix,
                                                                              bins = c(seq(0, 13, by = .1), 100),
                                                                              season_start = season_start,
```

```

season_end = season_end,
current_time=current_date_in_cdc_format,
nsim=1000,
h_max=6 )

```

```

head(submission_df)

```

```

##   bin_start_incl bin_end_notincl value      target type forecast_week
## 1             0         0.1      0 1 wk ahead Bin    2020-EW10
## 2            0.1         0.2      0 1 wk ahead Bin    2020-EW10
## 3            0.2         0.3      0 1 wk ahead Bin    2020-EW10
## 4            0.3         0.4      0 1 wk ahead Bin    2020-EW10
## 5            0.4         0.5      0 1 wk ahead Bin    2020-EW10
## 6            0.5         0.6      0 1 wk ahead Bin    2020-EW10

```