

Capturing different daily profiles #434

New issue

Open

gs9824 opened this issue 16 days ago · 8 comments

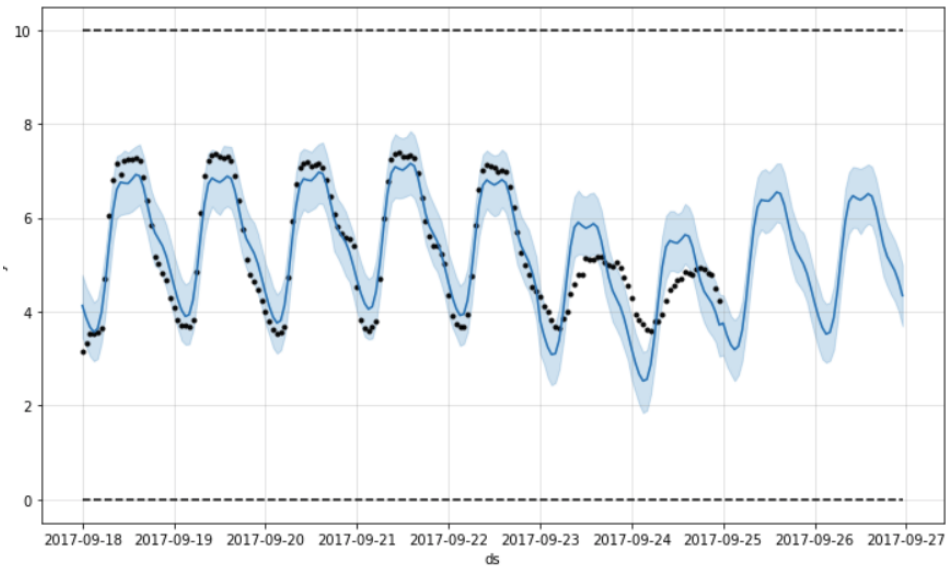


gs9824 commented 16 days ago

I'm using prophet on the following dataset

[ts_1_week.txt](#)

I'm trying to fit the model as good as possible. This works reasonably, but weekends seem to have a different profile - which the model does not seem to capture (take up is later on weekend days)



I tried to add a specific regressor for weekenddays, but this does not seem to make a lot of difference.

```
m_1_week = Prophet(growth='logistic', weekly_seasonality=True)
m_1_week.add_regressor('weekend')
ts_1_week['floor']=0
ts_1_week['cap']=10
m_1_week.fit(ts_1_week);

future = m_1_week.make_future_dataframe(periods=48, freq = "h", include_history = True)
future['floor']=0
future['cap']=10
future['weekend'] = future['ds'].apply(weekend)

forecast = m_1_week.predict(future)
```

There seems to be only 1 daily profile, while it would make sense to have different ones for different day types

Assignees

No one assigned

Labels

enhancement

Projects

None yet

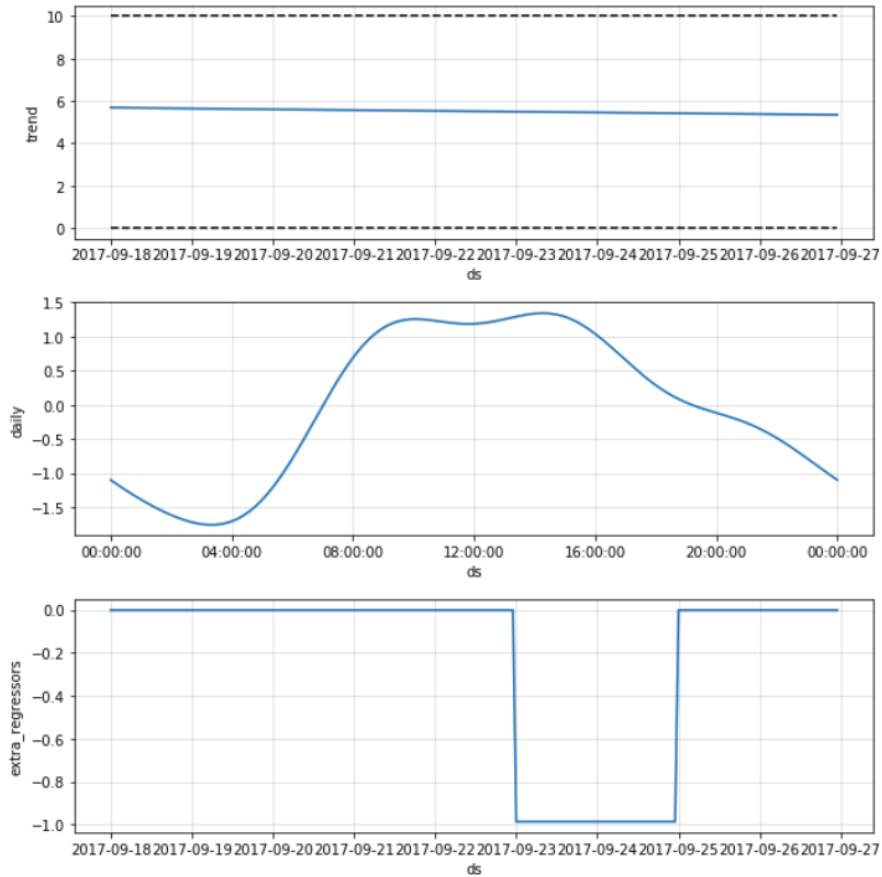
Milestone

No milestone

Notifications

3 participants





Is there a way to model the differences between weekdays & weekends in a better way ?

gs9824 changed the title from **Capturing different daily differences** to **Capturing different daily profiles** 16 days ago



vhpietil commented 15 days ago

Do you have more data to share? I have an approach which I would like to try. But when you have only one week of data, the model easily overfits.



gs9824 commented 15 days ago • edited ▼

Thanks for your response and willingness to look into this.

Yes, the original series is 4 weeks. The original data values are in the yy column, the y column is the log of the yy values -- as that seemed to give better results.

Here it is :

[ts_4_weeks.txt](#)



vhpietil commented 15 days ago

You are right that there is only one daily seasonality that is applied for all dates. This makes sense with temperature and other things that are not interested in weekends.

When you used specific regressor for the weekend, you put constant 1 for every hour of the weekend. It just scales whole weekend up or down but cannot change the shape of the seasonality on weekends.

You can make an external regressor by taking some number of sine and cosine functions (fourier series) in the same way prophet estimates the seasonalities. When you add this external regressor only on saturday and sunday, prophet can estimate how the "normal" daily seasonality need to be changed in order to get right seasonality for the weekends.

Unfortunately I don't have time to polish my approach more, but here is the code and the results. I code in R.

```
library(prophet)
library(readr)
library(dplyr)

# Read the data
df <- read_csv("ts_4_weeks.txt")

# Fourier series, time_diff and set_date are functions
# from prophet-package
# These are used to make extra regressors
fourier_series <- function(dates, period, series.order) {
  t <- time_diff(dates, set_date('1970-01-01 00:00:00'))
  features <- matrix(0, length(t), 2 * series.order)
  for (i in 1:series.order) {
    x <- as.numeric(2 * i * pi * t / period)
    features[, i * 2 - 1] <- sin(x)
    features[, i * 2] <- cos(x)
  }
  return(features)
}

time_diff <- function(ds1, ds2, units = "days") {
  return(as.numeric(difftime(ds1, ds2, units = units)))
}

set_date <- function(ds = NULL, tz = "GMT") {
  if (length(ds) == 0) {
    return(NULL)
  }

  if (is.factor(ds)) {
    ds <- as.character(ds)
  }

  if (min(nchar(ds)) < 12) {
    ds <- as.POSIXct(ds, format = "%Y-%m-%d", tz = tz)
  } else {
    ds <- as.POSIXct(ds, format = "%Y-%m-%d %H:%M:%S", tz = tz)
  }
  attr(ds, "tzone") <- tz
  return(ds)
}

# Create extra regressors.
# These are basically sin and cos signals with varying periods
# Dates are joined to the regressors and then I filter only saturday and sunday
# Names of the columns are V1...V(2*series.order)
series.order <- 3
regressors <- fourier_series(df$ds, 1, series.order) %>%
  as_data_frame %>%
  cbind(ds = df$ds) %>%
  filter((lubridate::wday(df$ds) %in% c(1,7)))

# Initialize prophet
m <- prophet(weekly.seasonality = TRUE)

# Add regressors to the model one by one
for (i in 1:(length(regressors)-1)) {
  m <- add_regressor(m, paste0("V", i))
}

# Join regressors to the data, replace NAs with zeros and fit
m <- fit.prophet(m, df = df %>% left_join(regressors) %>% replace(is.na(.), 0))

# Create future dataframe and extra regressors for those datetimes
future = make_future_dataframe(m, 24*7, freq = 3600)

series.order <- 3
future_regressors <- fourier_series(future$ds, 1, series.order) %>%
  as_data_frame %>%
```

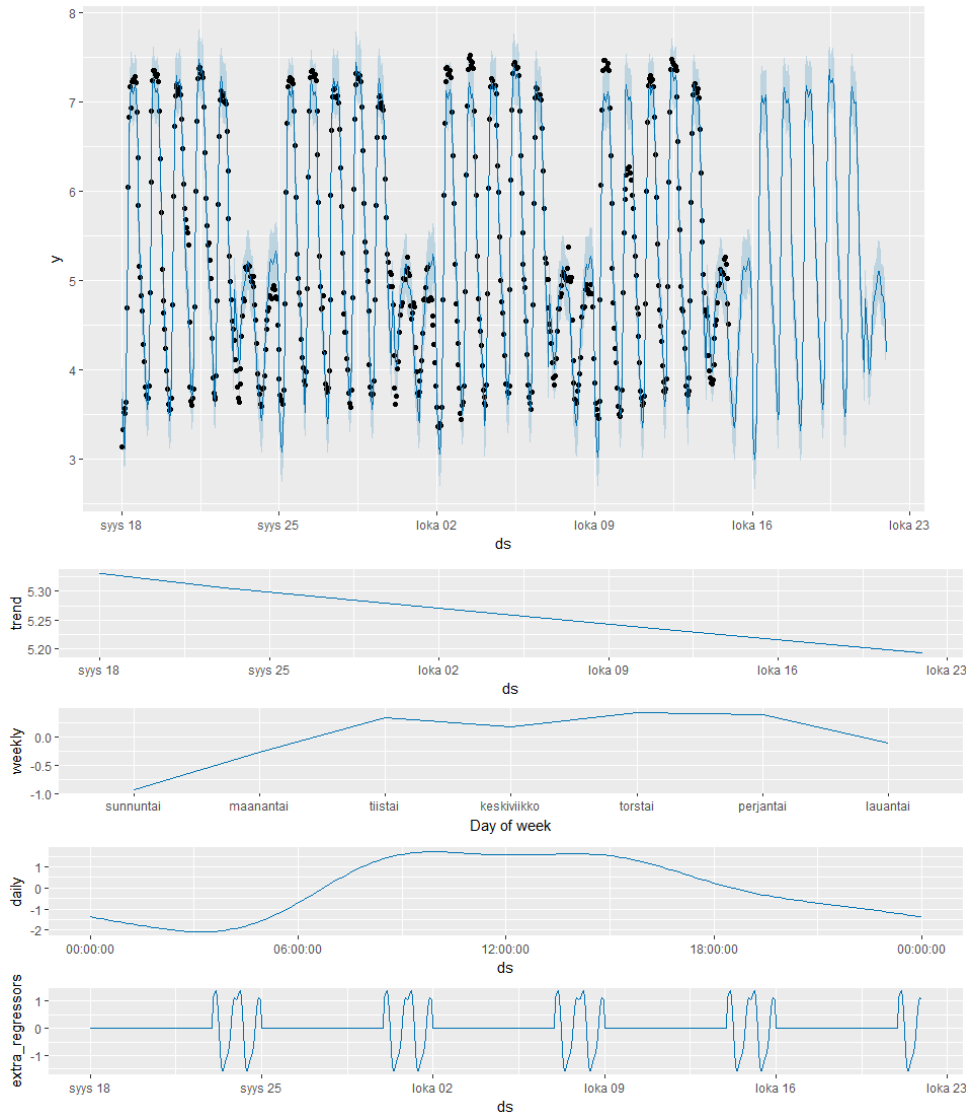
```

cbind(ds = future$ds) %>%
  filter((lubridate::wday(future$ds) %in% c(1,7)))

# Profit
forecast <- predict(m, future %>% left_join(future_regressors) %>% replace(is.na(.), 0))

plot(m, forecast)

```



You can see that the extra regressor modifies daily seasonality on the weekends and the overall fit is pretty good. There is a weird artefact/spike in the first hours of the saturday. It probably can be explained some how and dealt with.

@bletham What do you think about this approach. I have noticed that the weekly seasonality can be pretty different during winter and summer and this approach has given me quite promising results. Do you see some problem with this approach? Should I do this in some different way?



1

vhpietil referenced this issue 15 days ago

Large variations in weekly seasonality throughout the year #425

Open



bletham commented 15 days ago

Contributor

It looks like the weekly seasonality is a multiplicative factor on top of the daily seasonality.

Currently, seasonality and extra regressors are always additive, which means that $y(t) = \text{trend}(t) + \text{daily_effect}(t) + \text{extra_regressor}(t)$, so `extra_regressor` has a constant effect throughout the day. You can see in the plot that the model estimate dips down a bit on those days, but since it isn't able to scale the daily seasonality (multiplicative factor) it isn't able to fit well.

Multiplicative seasonality is a current priority (#254). In the meantime, what @vhpietil proposed would be the right thing to do: Manually create a separate daily seasonality for the weekend days. It's a bit of a pain as you can see but I would expect it to work well here (and certainly seems to).

This is related to #425 which got some benefit from splitting seasonalities out by time. We should probably think of a cleaner way to do this sort of thing.



bletham commented 15 days ago

Contributor

Just as an FYI, in R you can call the not-exported functions from prophet using `:::`, like `prophet:::fourier_series` so you don't have to redefine them externally.



gs9824 commented 14 days ago

Many thanks for your responses & suggestions. I will try these out & report back



gs9824 commented 14 days ago

I can confirm the results of vhpietil. Maybe there is indeed a need for a feature to do this more or less automatically (add a regressor for specific days, specific weeks, ...).


Many thanks for helping out in this :-)



bletham commented 10 days ago

Contributor

Yeah, let's leave this open as an enhancement to have an easier way to add these types of specific seasonal regressors.

 bletham added the **enhancement** label 10 days ago