# Forecasting at Scale

Christian Barz

2020-07-23

1 Prophet's goals and alternative

2 Prophet model

3 Stan model and fitting

4 Future Talk's

5 Further reading

# Prophet's goals and alternative

# Prophet library

The Prophet library[1] is a model and a framework, that targets

- non-experts with background business knowledge

[1] Taylor, Letham, Forecasting at scale, 2017, The American Statistician

# Prophet library

The Prophet library[1] is a model and a framework, that targets

- non-experts with background business knowledge
- less knowledge about time series is required

[1] Taylor, Letham, Forecasting at scale, 2017, The American Statistician

# Prophet library

The Prophet library[1] is a model and a framework, that targets

- non-experts with background business knowledge
- less knowledge about time series is required
- easy to configure (analyst in the loop)

[1] Taylor, Letham, Forecasting at scale, 2017, The American Statistician

# Prophet library

The Prophet library[1] is a model and a framework, that targets

- non-experts with background business knowledge
- less knowledge about time series is required
- easy to configure (analyst in the loop)
- easily interpretable parameters

[1] Taylor, Letham, Forecasting at scale, 2017, The American Statistician

# Prophet library

The Prophet library[1] is a model and a framework, that targets

- non-experts with background business knowledge
- less knowledge about time series is required
- easy to configure (analyst in the loop)
- easily interpretable parameters
- flexible for a wide range of business problems

[1] Taylor, Letham, Forecasting at scale, 2017, The American Statistician

# Alternative

The `forecast` R-package contains some alternative for automated times series models.

- `auto.arima`[2], fits mutliple ARIMA models and take the best fit

[2] Hyndman, Khandakar et al. 2007, Automatic time series for forecasting: the forecast package for R

# Alternative

The `forecast` R-package contains some alternative for automated times series models.

- `auto.arima`[2], fits mutliple ARIMA models and take the best fit
- `ets`[3], fits mutliple exponential smoothing models and take the best fit

[2] Hyndman, Khandakar et al. 2007, Automatic time series for forecasting: the forecast package for R

[3] Hyndman, Koehler., Snyder & Grose, 2002, 'A state space framework for automatic forecasting using exponential smoothing methods'

# Alternative

The `forecast` R-package contains some alternative for automated times series models.

- `auto.arima`[2], fits mutliple ARIMA models and take the best fit
- `ets`[3], fits mutliple exponential smoothing models and take the best fit
- `snaive`[4], random walk model with seasonality

[2] Hyndman, Khandakar et al. 2007, Automatic time series for forecasting: the forecast package for R

[3] Hyndman, Koehler., Snyder & Grose, 2002, 'A state space framework for automatic forecasting using exponential smoothing methods'

[4] De Livera, Hyndman. & Snyder, 2011, 'A state space framework for automatic forecasting using exponential smoothing methods'

# Prophet model

## Generalized additive models

Prophet uses a generalized additive model (GAM)[5]

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $y(t)$ target

[5] Hastie & Tibshirani, 1987, 'Generalized additive models: some applications'

# Generalized additive models

Prophet uses a generalized additive model (GAM)[5]

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $y(t)$ target
- $g(t)$ trend function

[5] Hastie & Tibshirani, 1987, 'Generalized additive models: some applications'

# Generalized additive models

Prophet uses a generalized additive model (GAM)[5]

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $y(t)$ target
- $g(t)$ trend function
- $s(t)$ seasonal model

[5] Hastie & Tibshirani, 1987, 'Generalized additive models: some applications'

# Generalized additive models

Prophet uses a generalized additive model (GAM)[5]

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $y(t)$ target
- $g(t)$ trend function
- $s(t)$ seasonal model
- $h(t)$ represents holiday effects, i.e. irregular schedules over one or more days

[5] Hastie & Tibshirani, 1987, 'Generalized additive models: some applications'

# Generalized additive models

Prophet uses a generalized additive model (GAM)[5]

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

- $y(t)$ target
- $g(t)$ trend function
- $s(t)$ seasonal model
- $h(t)$ represents holiday effects, i.e. irregular schedules over one or more days
- $\epsilon_t$ error term (not accomodated by the model)

[5] Hastie & Tibshirani, 1987, 'Generalized additive models: some applications'

# The trend model

Trend can be modeled by

- saturated growth model

# The trend model

Trend can be modeled by

- saturated growth model
- piecewise linear model

# Nonlinear Saturating growth (basic)

A basic logistic growth model is given by

$$g(t) = \frac{C}{1 + exp(-k(t - m))}$$

where

- $C$ is the carrying capacity (upper bound)

# Nonlinear Saturating growth (basic)

A basic logistic growth model is given by

$$g(t) = \frac{C}{1 + exp(-k(t - m))}$$

where

- $C$ is the carrying capacity (upper bound)
- $k$ is the growth rate

# Nonlinear Saturating growth (basic)

A basic logistic growth model is given by

$$g(t) = \frac{C}{1 + exp(-k(t-m))}$$

where

- $C$ is the carrying capacity (upper bound)
- $k$ is the growth rate
- $m$ is an offset parameter

# Nonlinear Saturating growth (time dependent parameters)

- $C$ (carrying capacity) and $k$ (growth rate) are usually not constant

# Nonlinear Saturating growth (time dependent parameters)

- $C$ (carrying capacity) and $k$ (growth rate) are usually not constant
- incoporate trend changes by explicite defined change points

# Nonlinear Saturating growth (time dependent parameters)

- $C$ (carrying capacity) and $k$ (growth rate) are usually not constant
- incoporate trend changes by explicite defined change points
- change point can be set by analyst (e.g. product launches) or automatically selected

# Nonlinear Saturating growth (time dependent carrying capacity)

TBA - this is not explained in the non reviewed paper

# Nonlinear Saturating growth (non constant growth rate - idea)

let $s_1, \ldots, s_n$ be change points (time stamps).

define $\delta \in \mathbb{R}^n$ (vector of growth rate adjustments)

define growth rate at time $t$ by

$$k(t) := k + \sum_{j: s_j < t} \delta_j$$

where $k$ is the base growth rate

# Nonlinear Saturating growth (non constant growth rate - mathematically correct)

define $a(t) \in \{0, 1\}^n$ by

$$a_j(t) := \begin{cases} 1, & \text{for } t \leq s_j \\ 0, & otherwise \end{cases}$$

then rate at time $t$ is given by

$$k(t) := k + a(t)^T \delta$$

# Nonlinear Saturating growth (non constant growth rate - mathematically correct 2)

The offset parameter $m$ is than adjusted according to the change points by:

$$\gamma_i := \left(s_j - m - \sum_{l<j} \gamma_j\right)\left(1 - \frac{k + \sum_{l<j} \delta_l}{k + \sum_{l \leq j} \delta_l}\right)$$

# Nonlinear Saturating growth (non constant growth rate - mathematically correct 3)

Putting all together we recieve :

$$
\begin{aligned}
g(t) &= \frac{C(t)}{1 + exp(-k(t)(t - m(t))} \\
&= \frac{C(t)}{1 + \exp(-(k + a(t)^T\delta)(t - (m + a(t)^T\gamma)))}
\end{aligned}
$$

# Linear trend with Changepoints

using the change points from above a piecewise linear model is given by

$$\begin{aligned} g(t) &= k(t) \cdot t + m(t) \\ &:= (k + a(t)^T \delta) * t + (m + a(t)^T \gamma) \end{aligned}$$

# To do

- We have to fit the unknown parameters $k$, $m$, and $\delta$

where

# To do

- We have to fit the unknown parameters $k$, $m$, and $\delta$
- recall Bayes theorem

$$P(\delta|Data) = \frac{P(Data|\delta) \cdot P(\delta)}{P(Data)}$$

where

# To do

- We have to fit the unknown parameters $k$, $m$, and $\delta$
- recall Bayes theorem

$$P(\delta|Data) = \frac{P(Data|\delta) \cdot P(\delta)}{P(Data)}$$

where

- $P(\delta|Data)$ is the conditional probability to observe $\delta$ given $Data$

# To do

- We have to fit the unknown parameters $k$, $m$, and $\delta$
- recall Bayes theorem

$$P(\delta|Data) = \frac{P(Data|\delta) \cdot P(\delta)}{P(Data)}$$

where

- $P(\delta|Data)$ is the conditional probability to observe $\delta$ given $Data$
- $P(Data|\delta)$ is the likelihood, which can be interpreted as the probability to observe $Data$ given $\delta$

# Automatic changepoint selection

Change points are automatically detected by putting a sparse prior on $\delta$, i.e.

$$\delta_j \sim \mathit{Laplace}(0, \tau)$$

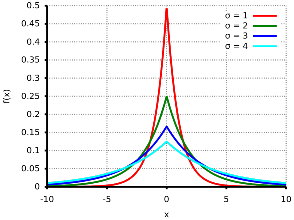


Figure 1: LaPlace density function from wikipedia, here $\sigma = \tau$

# The seasonal model

The seasonality is approximated by a (truncated) Fourier serie:

$$s(t) = \sum_{i=1}^{N} (a_n \cos(\frac{2\pi it}{P}) + b_n \frac{2\pi it}{P}))$$

where

- P is the period in days, for example P =7 for weekly seasonality

# The seasonal model

The seasonality is approximated by a (truncated) Fourier serie:

$$s(t) = \sum_{i=1}^{N} (a_n \cos(\frac{2\pi it}{P}) + b_n \frac{2\pi it}{P}))$$

where

- P is the period in days, for example P =7 for weekly seasonality
- N truncates the series and is a model parameter for the user to adjust the fitting

## The seasonal model (fitting)

Fitting requires to estimate $2N$ parameters $\beta = (a_1, b_1, \dots, a_N, b_N)$.

This is done by constructing a matrix of seasonality vectors for each $t$ in our historical and future data, e.g. for weekly seasonality and $N = 3$

$$X(t) := (\cos(\frac{2\pi(1)t}{7}), \dots, \sin(\frac{2\pi(3)t}{7}))$$

Hence the seasonal component is

$$s(t)X(t) \cdot \beta$$

where the prior on $\beta$ is normaly distributed: $\beta \sim \text{Normal}(0, \sigma^2)$.

# Fourier series

TBF

- recall how is an element in an vector space is represented in a basis

# Fourier series

TBF

- recall how is an element in an vector space is represented in a basis
- the functions $e_k(x) := \exp(\frac{2\pi ikx}{P})$ is a basis of a (dense) subspace in the vector space of $P$ periodic functions

# Fourier series

TBF

- recall how is an element in an vector space is represented in a basis
- the functions $e_k(x) := \exp(\frac{2\pi ikx}{P})$ is a basis of a (dense) subspace in the vector space of $P$ periodic functions
- the basis functions $e_k$ can be expressed as a function in sin and cos

# Fourier series

TBF

- recall how is an element in an vector space is represented in a basis
- the functions $e_k(x) := \exp(\frac{2\pi ikx}{P})$ is a basis of a (dense) subspace in the vector space of $P$ periodic functions
- the basis functions $e_k$ can be expressed as a function in sin and cos
- convergence of Fourier series

# "Holidays and Events"

TBF

# Stan model and fitting

# Stan model and fitting

Stan is a platform for Bayesian inference using MCMC sampling and more.

# Stan model with logistic growth trend model

```
model {
// Priors
k ~ normal(0, 5);
m ~ normal(0, 5);
epsilon ~ normal(0, 0.5);
delta ~ double_exponential(0, tau);
beta ~ normal(0, sigma);
// Logistic likelihood
y ~ normal(C ./ (1 + exp(-(k + A * delta)
                        .* (t - (m + A * gamma))) +
             X * beta, epsilon);
}
```

# Stan fitting for prophet

- Prophet uses the L-BFGS algorithm from Stan to fit the GAM

# Stan fitting for prophet

- Prophet uses the L-BFGS algorithm from Stan to fit the GAM
- BFGS = Broyden–Fletcher–Goldfarb–Shanno algorithm, quasi Newton method

# Future Talk's

# regarding Prophet

- how the time dependet carrying capacity is determined ?

## regarding Prophet

- how the time dependet carrying capacity is determined ?
- how additional regressors are included ?

# regarding Prophet

- how the time dependet carrying capacity is determined ?
- how additional regressors are included ?
- how outliers are handeled ?

# theoretical background

- Stan models

# theoretical background

- Stan models
- GAMS

# theoretical background

- Stan models
- GAMS
- Baysian models

# theoretical background

- Stan models
- GAMS
- Baysian models
- L-BFGS

# theoretical background

- Stan models
- GAMS
- Baysian models
- L-BFGS
- Marcov chain Monte Carlo

# Further reading

# Further reading

- The textbook for forecast R-package, but with a lot of theory and practice