made by abdelmomen ben nasr

In this Presentation

The Prophet Forecasting Model

Facebook Prophet is a forecasting tool developed by Facebook's Core Data Science team to produce reliable and high-quality forecasts at scale. It was introduced to address the challenges associated with forecasting, challenges can be associated with producing reliable and high-quality forecasts include identifying trends and seasonality patterns, detecting and removing outliers, and dealing with missing data. Additionally, analysts may need to adjust the model parameters to account for changes in the underlying data generating process, or to incorporate domain knowledge into the forecasting procedure.

The Idea behind The Prophet Model

The idea behind Facebook Prophet is to provide a practical approach to producing reliable and high-quality forecasts, even when dealing with a variety of time series and limited expertise in time series modeling. This is achieved through a decomposable time series model that includes trend, seasonality, and holiday components, as well as a modular regression model that allows analysts to select the components that are relevant to their forecasting problem and easily make adjustments as needed. The system also includes a performance monitoring component that allows analysts to measure and track forecast accuracy, and flag forecasts that should be checked manually to help make incremental improvements.

GAM (Generalised Additive Model)

A GAM is a linear model with a key difference when compared to Generalised Linear Models such as Linear Regression. A GAM is allowed to learn non-linear features.

In Linear regression, the equation is defined by the sum of a **linear combination of variables**. Each variable is given a weight, β and added together.

$$Z = \beta_0 x_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

In GAMs, the assumption that the target can be calculated using a linear combination of variables is replaced by simply using a non-linear combination of variables, denoted by *s*, for '**smooth function**'.

$$Z = s_0 x_0 + s_1 x_1 + \ldots + s_n x_n$$

The formula of s is given by :

$$s(x) = \sum_{k=1}^{k} \beta_k b_k(x)$$

where

- β is a weight, the relative importance assigned to each observation in the dataset during the model fitting process.
- b is a basis expansion, they represent a predictor variable using a set of special functions instead of using the original values directly. These special functions are chosen to capture the patterns and shape in the data thus rendering the model more non-linear friendly

Exponential smoothing

Exponential smoothing is a time series method for forecasting univariate time series data. Time series methods work on the principle that a prediction is a weighted linear sum of past observations or lags. The Exponential Smoothing time series method works by assigning exponentially decreasing weights for past observations. It is called so because the weight assigned to each demand observation is exponentially decreased.

The model assumes that the future will be somewhat the same as the recent past. The only pattern that Exponential Smoothing learns from demand history is its level - the average value around which the demand varies over time.

Exponential smoothing is generally used to make forecasts of time-series data based on prior assumptions by the user, such as seasonality or systematic trends.

Exponential smoothing is a broadly accurate forecasting method for short-term forecasts. The technique assigns larger weights to more recent observations while assigning exponentially decreasing weights as the observations get increasingly distant. This method produces slightly unreliable long-term forecasts.

Exponential smoothing can be most effective when the time series parameters vary slowly over time

The main types of Exponential Smoothing forecasting are

- Simple or Single Exponential Smoothing
- Double Exponential Smoothing
- Triple Exponential Smoothing

Simple or Single Exponential Smoothing

Simple or single exponential smoothing (SES) is the method of time series forecasting used with univariate data with no trend and no seasonal pattern. It needs a single parameter called alpha (a), also known as the smoothing factor. Alpha controls the rate at which the influence of past observations decreases exponentially. The parameter is often set to a value between 0 and 1.

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1} = s_{t-1} + \alpha (x_t - s_{t-1})$$

where

st = smoothed statistic (simple weighted average of current observation xt)

st-1 = previous smoothed statistic

α = smoothing factor of data; 0 < α < 1

t = time period

Double Exponential Smoothing

This method is known as Holt's trend model or second-order exponential smoothing. Double exponential smoothing is used in time-series forecasting when the data has a linear trend but no seasonal pattern. The basic idea here is to introduce a term that can consider the possibility of the series exhibiting some trend.
In addition to the alpha parameter, Double exponential smoothing needs another smoothing factor called beta (b), which controls the decay of the influence of change in trend. The method supports trends that change in additive ways (smoothing with linear trend) and trends that change in multiplicative ways (smoothing with exponential trend).

Double exponential smoothing formulas are :

$$
\begin{aligned}
&\mathrm{S}_1 = x_1\\
&B_1 = x_1 - x_0\\
&\text{For } t > 1,\ \ \mathrm{S}_t = \alpha x_t + (1-\alpha)(S_{t-1} + B_{t-1})\\
&\beta_t = \beta(S_t - S_{t-1}) + (1-\beta)B_{t-1}
\end{aligned}
$$

where,

bt = best estimate of the trend at time t

β = trend smoothing factor; 0 < β <1

Triple Exponential Smoothing

This method is the variation of exponential smoothing that's most advanced and is used for time series forecasting when the data has linear trends and seasonal patterns. The technique applies exponential smoothing three times – level smoothing, trend smoothing, and seasonal smoothing. A new smoothing parameter called gamma (g) is added to control the influence of the seasonal component.
The triple exponential smoothing method is called Holt-Winters Exponential Smoothing, named after its contributors, Charles Holt and Peter Winters.

Holt-Winters Exponential Smoothing has two categories depending on the nature of the seasonal component:

- Holt-Winter's Additive Method - for seasonality that is addictive.
- Holt-Winter's Multiplicative Method – for seasonality that is multiplicative.

The Math behind The Prophet Model

The Facebook Prophet model uses a decomposable time series model with three main components: trend, seasonality, and holidays and they are combined into this equation:

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

where g(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes (e.g., weekly and yearly seasonality), and h(t) represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term et represents any idiosyncratic changes which are not accommodated by the model, e(t) is usually assumed to be normally distributed.

This specification is similar to a generalized additive model (GAM) but Phrophet on the other hand uses only time as a regressor but possibly several linear and non-linear functions of time as components. Modeling seasonality as an additive component is the same approach taken by exponential smoothing (Gardner 1985). Multiplicative seasonality, where the seasonal effect is a factor that multiplies g(t), can be accomplished through a log transform.

The Trend Model :Nonlinear, Saturating Growth

For growth forecasting, the core component of the data generating process is a model for how the population has grown and how it is expected to continue growing. Modeling growth at Facebook is often similar to population growth in natural ecosystems (e.g., Hutchinson 1978), where there is nonlinear growth that saturates at a carrying capacity. For example, the carrying capacity for the number of Facebook users in a particular area might be the number of people that have access to the Internet. This sort of growth is typically modeled using the logistic growth model, which in its most basic form is :

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

with C the carrying capacity, k the growth rate, and m an offset parameter.

There are two important aspects of growth at Facebook that are not captured in the logistic growth model, First, the carrying capacity is not constant, caused by the increasing number of people having access to the internet, thus the fixed capacity C is replaced with a time-varying capacity C(t)
Secondly the growth rate is not constant. New products can profoundly alter the rate of growth in a region, so the model must be able to incorporate a varying rate in order to fit historical data.

The Trend Model :Nonlinear, Saturating Growth

To incorporate trend changes in the growth model by explicitly defining changepoints where the growth rate is allowed to change. Suppose there are S changepoints at times sj , j = 1, . . . , S.

We define a vector of rate adjustments δ ∈ R ^(S) , where δj is the change in rate that occurs at time sj . The rate at any time t is then the base rate k, plus all of the adjustments up to that point: $k + \sum_{j:t>s_j} \delta_j$. This is represented more cleanly by defining a vector a(t) ∈ {0, 1}^(S) such that

$$a_j(t) = \begin{cases} 1, & \text{if } t \geq s_j, \\ 0, & \text{otherwise.} \end{cases}$$

The rate at time t is then $k + \mathbf{a}(t)^\intercal \boldsymbol{\delta}$. . When the rate k is adjusted, the offset parameter m must also be adjusted to connect the endpoints of the segments. The correct adjustment at changepoint j is easily computed as

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

The piecewise logistic growth model is then :

$$g(t) = \frac{C(t)}{1 + \exp(-(k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})(t - (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma})))}$$

The Trend Model :Linear Trend with Changepoints

For forecasting problems that do not exhibit saturating growth, a piece-wise constant rate of growth provides a parsimonious and often useful model. Here the trend model is

$$g(t) = (k + \mathbf{a}(t)^\intercal \boldsymbol{\delta})t + (m + \mathbf{a}(t)^\intercal \boldsymbol{\gamma})$$

where as before k is the growth rate, δ has the rate adjustments, m is the offset parameter, and γj is set to -sjδj to make the function continuous.

The Trend Model :Automatic Changepoint Selection

The changepoints s_j could be specified by the analyst using known dates of product launches and other growth-altering events, or may be automatically selected given a set of candidates. Automatic selection can be done quite naturally with the formulation in Nonlinear, Saturating Growth and in Linear Trend with Changepoints by putting a sparse prior on δ.
We often specify a large number of changepoints (e.g., one per month for a several year history) and use the prior
δ_j Laplace(0, τ). The parameter τ directly controls the flexibility of the model in altering its rate. Importantly, a sparse prior on the adjustments δ has no impact on the primary growth rate k, so as τ goes to 0 the fit reduces to standard (not-piecewise) logistic or linear growth.

Seasonality

Business time series often have multi-period seasonality as a result of the human behaviors they represent. For instance, a 5-day work week can produce effects on a time series that repeat each week, while vacation schedules and school breaks can produce effects that repeat each year. To fit and forecast these effects we must specify seasonality models that are periodic functions of t.

Thus using Fourrier series to provide flexibility and having an approximate arbitrary smooth seasonal effects with a standard Fourier series :

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right)$$

where s(t) is the seasonality component at time t, a(n) and b(n) are the Fourier coefficients for the nth harmonic, N is the number of harmonics used to model the seasonality, and P is the period of the seasonality (e.g., 7 for weekly data, 365.25 for yearly data).

Holidays and Events

The Facebook Prophet model uses the holiday component to account for holidays that occur on irregular schedules. This involves treating each day around a holiday as a separate holiday, allowing the model to capture the varying effects of holidays on different days or times of the year. Analysts provide a custom list of past and future events, including the event or holiday's name and country, to incorporate holidays into the model. The model combines global and country-specific holidays for accurate forecasting.

Incorporating the list of holidays into the model is made straightforward by assuming that the effects of holidays are independent. For each holiday i, let D(i) be the set of past and future dates for that holiday.

The model adds an indicator function representing whether time t is in D(i), which is multiplied by a parameter w(i) representing the effect of the holiday. The holiday component is then defined as the sum of the effects of all holidays in the list:

$$h(t) = \sum_{i=1}^{M} [w(i) \cdot z(t, D(i))]$$

where h(t) is the holiday component at time t, w(i) is the effect of the ith holiday, z(t, D(i)) is an indicator variable that takes the value 1 if time t is in the set of dates D(i) for the ith holiday, and 0 otherwise, and M is the total number of holidays in the list.

FB PROPHET VS ARIMA

We are, in effect, framing the forecasting problem as a curve-fitting exercise, which is inherently different from time series models that explicitly account for the temporal dependence structure in the data. While we give up some important inferential advantages of using a generative model such as an ARIMA, this formulation provides a number of practical advantages:

– Flexibility: We can easily accommodate seasonality with multiple periods and let the analyst make different assumptions about trends.

– Unlike with ARIMA models, the measurements do not need to be regularly spaced, and we do not need to interpolate missing values e.g. from removing outliers. – Fitting is very fast, allowing the analyst to interactively explore many model specifications, for example in a Shiny application (Chang et al. 2015).

– The forecasting model has easily interpretable parameters that can be changed by the analyst to impose assumptions on the forecast. Moreover, analysts typically do have experience with regression and are easily able to extend the model to include new components.

Thanks for your attention !