# Prophet

Версия 2

опубликована в 2017

Taylor, Letham
Forecasting at scale
2017

$$Y_t = T_t + S_t + X_t + \epsilon_t$$

аддитивная регрессионная модель с четырьмя основными компонентами:

**тренд** == кусочно-линейной регрессии или
кусочной логистической кривой;

**годовая сезонность** == ряд Фурье;

**недельная сезонность** == индикаторные переменные дней недели

**интервенции** официальные праздничные и выходные дни - Новый год, Рождество и т.п., а также другие дни, во время которых свойства временного ряда могут существенно измениться - спортивные или культурные события, природные явления и т.п.; как и в случае с днями недели, такие дни представлены в модели в виде индикаторных переменных.

О рядах Фурье с картинками
https://www.jezzamon.com/fourier/

Оценивание параметров подгоняемой модели выполняется с использованием принципов байесовской статистики (либо методом нахождения апостериорного максимума (MAP), либо путем полного байесовского вывода). Для этого применяется платформа вероятностного программирования Stan.

Prophet представляет собой удобный интерфейс для работы с этой платформой
- из среды R
- посредством библиотеки **fbprophet** для Python -).

В таблицах всегда
столбцу с датами присвоено имя ds,
столбцу со значениями временного ряда - имя y.

Этого требует Prophet.
Использование каких-либо других имен приведет к ошибке при вызове соответствующих функций.

Внешние регрессоры объединяются в один.
Невозможно оценить влияние индивидуального регрессора на поведение временного ряда.

## Аргументы функции prophet()

growth
- Тип тренда. Принимает два возможных значения - "linear" ("линейный" - принято по умолчанию) и "logistic" ("логистический").

changepoints
- текстовый вектор с датами (в формате "YYYY-MM-DD"), соответствующими "переломным моментам", или "точкам излома" в y (т.е., датам, когда, как предполагается, произошли существенные изменения в тренде временного ряда). Если этот вектор не указан, то такие переломные моменты будут оценены автоматически.

n.changepoints
- предполагаемое количество "переломных моментов" (25 по умолчанию). Если аргумент changepoints задан, то аргумент n.changepoints будет проигнорирован. Если же changepoints не задан, то n.changepoints потенциальных точек излома будут распределены равномерно в пределах исторического отрезка, задаваемого аргументом changepoint.range.

changepoint.range
- доля исторических данных (начиная с самого первого наблюдения), в пределах которых будут оценены точки излома. По умолчанию составляет 0.8 (т.е. 80% наблюдений).

yearly.seasonality
- Параметр настройки годовой сезонности (т.е. закономерных колебаний в пределах года). Принимает следующие возможные значения: "auto" (автоматический режим, принят по умолчанию), TRUE, FALSE или количество членов ряда Фурье, с помощью которого аппроксимируется компонент годовой сезонности.

weekly.seasonality
- Параметр настройки недельной сезонности (т.е. закономерных колебаний в пределах недели). Возможные значения те же, что и у yearly.seasonality.

daily.seasonality
- Параметр настройки дневной сезонности (т.е. закономерных колебаний в пределах дня). Возможные значения те же, что и у yearly.seasonality.

holidays

- Таблица, содержащая два обязательных столбца: holiday (текстовая переменная - названия "праздников" и других важных событий, потенциально влияющих на свойства временного ряда) и ds (даты). По желанию в такую таблицу можно добавить еще два столбца - lower_window и upper_window
задают отрезок времени вокруг соответствующего события. Так, например, при "lower_window = -2" в модель будут добавлены 2 дня, предшествующие соответствующему событию. Также по желанию можно добавить столбец prior_scale - априорное значение стандартного отклонения (нормального) распределения, с помощью которого моделируется эффект того или иного события.

seasonality.mode
- Режим моделирования сезонных компонентов. Принимает два возможных значения: "additive" (аддитивный, принят по умолчанию) и "multiplicative" (мультипликативный).

seasonality.prior.scale
- Параметр, задающий "силу" сезонных компонентов модели (10 по умолчанию). Более высокие значения приведут к более "гибкой" модели, а низкие - к модели со слабее выраженными сезонными эффектами. Этот параметр можно задать отдельно для каждого типа сезонности с помощью функции add_seasonality().

holidays.prior.scale
- Параметр, задающий выраженность эффектов "праздников" и других важных событий (10 по умолчанию). Если таблица, подаваемая на аргумент holidays, имеет столбец prior_scale (см. выше), то аргумент holidays.prior.scale будет проигнорирован.

changepoint.prior.scale
- Параметр, задающий "гибкость" автоматического механизма обнаружения "переломных моментов" в y (0.05 по умолчанию). Более высокие значение позволят иметь больше таких точек излома.

mcmc.samples
- Целое число (0 по умолчанию). Если >0, то параметры модели будут оценены путем полного байесовского анализа с использованием mcmc.samples итераций алгоритма MCMC.

interval.width
- Число, определяющее ширину доверительного интервала для предсказанных моделью значений (0.8 по умолчанию, что соответствует 80%-ному интервалу). При "mcmc.samples = 0" этот интервал будет оценен с использованием MAP-метода и только на основе неопределенности в отношении тренда в y. Если же "mcmc.samples > 0", то доверительные интервалы будут оцениваться с учетом неопределенности в отношении оценок всех параметров модели (включая сезонные компоненты).

uncertainty.samples
- Количество итераций для оценивания доверительных интервалов (1000 по умолчанию).

fit
- Логическое значение (TRUE по умолчанию). При "fit = FALSE" произойдет только инициализация модельного объекта, но не подгонка самой модели.

... - дополнительные параметры, которые передаются на функцию fit.prophet().

## Точки излома тренда

можно задать точки излома самостоятельно (с помощью аргумента changepoints),

можно доверить их определение пакету prophet.

Автоматический режим

Я бы использовал критерий Чоу, у них иначе...

сначала
25 потенциальных точек излома будут равномерно распределены в пределах интервала, который охватывает первые 80% наблюдений из обучающей выборки.

Эти 25 точек - лишь предполагаемые места существенных изменений в тренде: в большинстве случаев на практике тренд временного ряда не изменяется так часто. Поэтому в ходе подгонки модели срабатывает механизм регуляризации (подобный L1-регуляризации), в результате чего выбирается минимально необходимое количество точек излома. Изобразить эти автоматически обнаруженные точки излома можно с помощью функции add_changepoints_to_plot().

changepoint.prior.scale - этот параметр управляет гладкостью тренда. Увеличение этого параметра (по умолчанию он равен 0.05), тем больше точек излома останется в подогнанной модели.

?Prophet

```
Init signature:
Prophet(
    growth='linear',
    changepoints=None,
    n_changepoints=25,
```

```
    changepoint_range=0.8,
    yearly_seasonality='auto',
    weekly_seasonality='auto',
    daily_seasonality='auto',
    holidays=None,
    seasonality_mode='additive',
    seasonality_prior_scale=10.0,
    holidays_prior_scale=10.0,
    changepoint_prior_scale=0.05,
    mcmc_samples=0,
    interval_width=0.8,
    uncertainty_samples=1000,
    stan_backend=None,
)
Docstring:
Prophet forecaster.

Parameters
----------
growth: String 'linear' or 'logistic' to specify a linear or
     logistic trend.
changepoints: List of dates at which to include potential
     changepoints. If not specified, potential changepoints are
     selected automatically.
n_changepoints: Number of potential changepoints to include.
     Not used if input `changepoints` is supplied.
     If `changepoints` is not supplied, then n_changepoints
     potential changepoints are selected uniformly from the first
     `changepoint_range` proportion of the history.
changepoint_range: Proportion of history in which trend
     changepoints will be estimated.
     Defaults to 0.8 for the first 80%.
     Not used if `changepoints` is specified.
yearly_seasonality: Fit yearly seasonality.
    Can be 'auto', True, False, or a number of Fourier terms to
     generate.
weekly_seasonality: Fit weekly seasonality.
    Can be 'auto', True, False, or a number of Fourier terms to
     generate.
daily_seasonality: Fit daily seasonality.
    Can be 'auto', True, False, or a number of Fourier terms to
     generate.
```

```
holidays: pd.DataFrame with columns holiday (string) and
     ds (date type) and optionally columns lower_window and
     upper_window which specify a range of days around the date
     to be included as holidays. lower_window=-2 will include
     2 days prior to the date as holidays.
     Also optionally can have a column prior_scale specifying the
prior scale for that holiday.
seasonality_mode: 'additive' (default) or 'multiplicative'.
seasonality_prior_scale: Parameter modulating the strength of the
    seasonality model. Larger values allow the model to fit
     larger seasonal fluctuations, smaller values dampen the
     seasonality. Can be specified for individual seasonalities
     using add_seasonality.
holidays_prior_scale: Parameter modulating the strength of the
     holiday components model, unless overridden in the holidays
     input.
changepoint_prior_scale: Parameter modulating the flexibility of
     the automatic changepoint selection. Large values will allow
     many changepoints, small values will allow few changepoints.
mcmc_samples: Integer, if greater than 0, will do full Bayesian
     inference with the specified number of MCMC samples. If 0,
     will do MAP estimation.
interval_width: Float, width of the uncertainty intervals provided
    for the forecast. If mcmc_samples=0, this will be only the
     uncertainty in the trend using the MAP estimate of the
     extrapolated generative model. If mcmc.samples>0, this will be
     integrated over all model parameters, which will include
     uncertainty in seasonality.
uncertainty_samples: Number of simulated draws used to estimate
    uncertainty intervals. Settings this value to 0 or False will
     disable uncertainty estimation and speed up the calculation.
    uncertainty intervals.
stan_backend: str as defined in StanBackendEnum default: None -
will try to iterate over all available backends and find the
working one


File:           c:\users\user\anaconda3\lib\site-packages\
fbprophet\forecaster.py
```

## Гладкость сезонных компонент

Сезонные компоненты
с помощью частичных сумм ряда Фурье.
Число членов ряда == "порядок ряда"
по умолчанию для (внутри-)годовой сезонности порядок ряда составляет 10.

В случае с моделью M4 это приводит к следующему компоненту годовой сезонности (обратите внимание на использование скрытой функции plot_yearly(), которую можно вызвать только обычным в таких случаях образом, т.е. указав имя пакета в сочетании с тройным двоеточем перед именем скрытой функции):

```
prophet:::plot_yearly(M4)
```

## эффекты "праздников"

термин "праздник" есть результат прямого перевода термина "holiday", принятого в Prophet.

"праздники" это
-"настоящие" официальные праздничные и выходные дни (например, Новый год, Рождество и т.п.),
-интервенции, то есть события, во время которых свойства моделируемой зависимой переменной существенно изменяются (спортивные или культурные мероприятия, природные явления и т.п.).

Поэтому термины "праздник", интервенция" и "событие" будут использованы как синонимы.

отдельная таблица, содержащая два обязательных столбца:
holiday (названия "праздников" и других важных событий) и
ds (даты в стандартном для R формате YYYY-MM-DD)