К лекции 5. Модель Prophet

Д. В. Чупраков

13 февраля 2024 г.

1 Модель prophet

В 2017 г. специалисты компании Facebook объявили о разработанном ими новом пакете для прогнозирования временных рядов — prophet («пророк»), который позволяет создавать прогнозные модели в (полу–)автоматическом режиме.

Подробное описание реализованной в prophet методологии можно найти в статье Taylor and Letham (2017)[1].

В основе этой методологии лежит процедура подгонки аддитивных регрессионных моделей (Generalized Additive Models, GAM) следующего вида:

$$y(t) = g(t) + s(t) + h(t) + x(t) + \varepsilon_t,$$

где $g(t)$ — функция, аппроксимирующая тренд ряда, $s(t)$ — функции, описывающая сезонные колебания (например, годовые, недельные и т.п.) соответственно, $h(t)$ — функция, отражающая эффекты праздников и других изолированных, но оказывающих существенное влияние событий, $x(t)$ — функция, отражающая влияние внешних регрессоров, ε_t — нормально распределенные случайные возмущения.

1.1 Модель тренда

Тренд в модели Prophet описывается кусочно-логистической функцией

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

где $C(t)$ — верхний порог (емкость источника) в момент времени t, k — скорость роста, m — параметр смещения, если временной ряд демонстрирует кумулятивный роста подобный модели роста населения и кусочно-линейной функцией:

$$g(t) = kt + m,$$

если кумулятивного роста не наблюдаетмся.

Изменения вида тренда происходят в моменты времени $s_1, s_2, \ldots s_S$.

[1] Taylor, S. J., Letham, B. (2017). Forecasting at Scale. The American Statistician, 72(1), 37–45. doi:10.1080/00031305.2017.1380080 URL: http://lethalletham.com/ForecastingAtScale.pdf

Введем вектор корректировок скорости $\delta \in \mathbb{R}^S$, где δ_j — изменение скорости роста временного ряда, наблюдаемое в момент времени s_j. Тогда скорость изменения тренда в каждый момент времени t получается из базовой скорости k прибавлением всех корректировое до этого момента:

$$v_t = k + \sum_{j:t>s_j} \delta_j = k + a(t)^T \delta$$

где вектор $a(t) \in \{0,1\}^S$ задан своими компонентами по следующей формуле:

$$\vec{a}_j(t) = \begin{cases} 1, & t \geqslant s_j \\ 0, & t < s_j \end{cases}$$

Аналогично смещение в точке изменения s_j легко вычисляется по формуле

$$\gamma_j = \Big(s_j - m - \sum_{l<j} \gamma_l\Big) \frac{\delta_j}{k + \sum\limits_{l \leqslant j} \delta_l}$$

Итак, функция тренда на каждом промежутке имеет один из видов:

$$g(t) = \frac{C(t)}{1 + \exp\Big(-\big(k + \vec{a}(t)^T\vec{\delta}\,\big)\big(t - (m + \vec{a}(t)^T\vec{\gamma}\,)\Big)}$$
$$g(t) = \big(k + \vec{a}(t)^T\vec{\delta}\big) \cdot t + \big(m + \vec{a}(t)T\vec{\gamma}\big),$$

где вектор $\vec{\gamma}$ определен покомпонентно равенствами $\gamma_j = -s_j\delta_j$ чтобы обеспечить непрерывность функции тренда.

Точки изменения s_j могут быть явно заданы исследователем, на основе знания событий, влияющих на динамику исследуемого процесса или могут быть автоматически выбраны из набора кандидатов.

При автоматическом выборе точек изменения тренда оценивается вектор $\vec{\delta}$ в предположении, что его компоненты распределены по закону Лапласа: $\delta_j \sim \text{Laplace}(0, \tau)$. Параметр τ управляет гибкостью модели при изменении ее скорости. если τ стремится к 0, то тренд вырождается в не кусочную логистическую или линейную функцию.

Будущие изменения скорости, которые имитируют изменения в прошлом, определяя параметр τ методами байесовской статистики: методом нахождения апостериорного максимума (MAP) и путем полного байесовского вывода.

Точки изменения тренда выбираются случайным образом так, чтобы средняя частота точек изменений совпадала с таковой в истории. То есть $\delta_j \sim \text{Laplace}(0, \tau)$ с верятностью $\frac{S}{T}$ и $\delta_j = 0$ с вероятностью $1 - \frac{S}{T}$.

1.2 Сезонность

Временные ряды часто имеют многопериодную сезонность. Например, рабочий график обеспечивает недельную сезонность, а климатические факторы — годовую.

Чтобы обеспечить учет сезонностей с разными периодами Функция сезонности представляется в виде отрезка ряда Фурье:

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

где P — регулярный период, который, как мы ожидаем, будет иметь временной ряд (например, $P = 365.25$ для годовых данных или $P = 7$ для недельных данных).

Таким образом функция сезонной компоненты может быть представлена в векторном виде $s(t) = \vec{X}(t) \cdot \vec{\beta}^T$ где

$$\vec{X}(t) = \left(\cos \frac{2\pi \cdot 1 \cdot t}{P}, \sin \frac{2\pi \cdot 1 \cdot t}{P}, \ldots, \cos \frac{2\pi \cdot N \cdot t}{P}, \sin \frac{2\pi \cdot N \cdot t}{P} \right)$$
$$\vec{\beta} = (a_1, b_1, \ldots a_N, b_N)$$

Иными словами, вычисление функции сезонности требует оценки $2N$ параметров? что делается путем построения матрицы векторов сезонности для каждого значения t в исторических данных.

Элементы вектора $\vec{\beta}$ берутся из нормального распределения $N(0, \sigma^2)$.

Усечение ряда до N компонент реализует фильтр высоких частот поэтому числа частот ведет к повышенному риску переобучения, но позволяет моделировать быстро меняющиеся процессы. Эмпирически для годовой и недельной сезонности подходят значения $N = 10$ и $N = 3$ соответственно. Выбор этих параметров может быть автоматизирован с помощью информационных критериев качества модели, например AIC.

1.3 Праздники

Праздники и события вызывают значительные, в некоторой степени предсказуемые изменения во многих временных рядах бизнеса и часто не следуют периодической схеме (праздники по лунному календарю, фестивали и т. д., презентации новых товаров и т. д.), поэтому их последствия плохо моделируются тригонометрическими функциями. В то же время влияние конкретного праздника на временные ряды часто бывает одинаковым из года в год, поэтому важно учитывать это в прогнозе.

Будем считать, что все праздники независмсимы. Поэтому, для каждого праздника i пусть D_i — набор прошлых и будущих дат для этого праздника. Рассмотрим характеристическую функцию $\mathbf{1}(t \in D)$ для каждого праздника D сопоставляющую моменту времени наличие праздника. присваиваем каждому празднику параметр $\vec{\kappa}_i$,характеризующий изменение в прогнозе. Это делается аналогично сезонности путем генерации матрицы регрессоров

$$Z(t) = (\mathbf{1}(t \in D_1), \ldots, \mathbf{1}(t \in D_L))$$

и найдем $h(t) = Z(t) \cdot \vec{\kappa}^T$ Как и в случае с сезонностью, используются нормально распределенные значения κ с нулевым математическим ожиданием.

2 Интерфейс Prophet

Установка: `python -m pip install prophet` или `conda install -c conda-forge prop`
Пакет до версии 1.0 носил название `fbprophet` поэтому в сети много информации именно под такое написание пакета. Ставить же надо пакет prophet.

Структура входного файла: Формат обучающей выборки имеет два обязательных поля `xs` — временная метка и `y` - значение уровня. Остальные поля являются полями внешних признаков. Также возможно указатние емкости источника — столбец `cap`.

Структура файла праздников:

Структура конструктора класса Prophet:
Модель Prophet подключается стандартно: `from prophet import Prophet` Аргументы конструктора следующие:

- `growth` — тип тренда: `linear`, `logistic` или `flat`
- `yearly_seasonality`, `weekly_seasonality`, `daily_seasonality` — виды сезонностей: принимает логическое значение, число компонент в ряде Фурье или `'auto'`.
- `holidays` — таблица праздников.
- `seasonality_mode` и `holidays_mode` тип сезонности и праздников: `'additive'`, `'multiplicative'`.
- `seasonality_prior_scale`, `holidays_prior_scale` — определяют силу соответствующей компоненты Большие значения приводят к большему влиянию соответствующей компоненты
- `changepoints` — список точек изменения тренда
- `n_changepoints` — число точек изменения тренда
- `changepoint_range`: Доля исторических данных, в которой будут искаться точки изменения тренда - число от 0 до 1по умолчанию — 0.8.
- `changepoint_prior_scale` — Параметр, регулирующий гибкость автоматического выбора точки изменения. С увеличением значения увеличивается число точек изменения тренда.
- `mcmc_samples` — число итераций байесовского вывода. если 0, то используется алгоритм MAP оценки.

Полезное:

```
from prophet.diagnostics import cross_validation, performance_metrics
from prophet.plot import plot_cross_validation_metric
plot_cross_validation_metric(df_cv, metric = 'rmse');
df_cv = cross_validation(m,
horizon = '31 days', period = '16 days', initial = '365 days', parallel = 'processes')
m.add_regressor('regressor_1')
model.add_seasonality(name='weekly_on_winter',
period=7, fourier_order=3, condition_name='зима')
```