Вопросы и ответы

Много сезонных составляющих

Много сезонных составляющих: план

- Наложение нескольких частот.
- Краткое напоминание STL.
- MSTL = STL много раз.

Картинка

Дневные данные, много частот

Что делать со сложной сезонностью?

- Использовать подходящую модель:
  ARIMA + предикторы Фурье, PROPHET, TBATS, …
- Разложить ряд на много составляющих:

$$y_t = trend_t + seas_t^{(1)} + seas_t^{(2)} + remainder_t$$

Вспоминаем STL

На входе:

Ряд y_t.

- n_p — периодичность сезонности, например, $n_p = 12$.
- n_l — сила сглаживания низкочастотного фильтра.
- n_s — сила сглаживания сезонных подрядов.
- n_t — сила сглаживания при выделении тренда.

На выходе:

Разложение $y_t = trend_t + seas_t + remainder_t$.

Применим STL последовательно!

1. Первичное выделение сезонных компонент.
2. Корректировка сезонных компонент.
3. Добываем тренд и остаток.

MSTL = STL много раз!

Шаг 1. Первичное выделение сезонных компонент.

1. Запустим STL для выделения сезонности высокой частоты.
   Запомним выделенную компоненту $seas_t^{(1)}$ и удалим её из ряда, $y_t^{(-1)} = y_t - seas_t^{(1)}$.
2. Запустим STL для выделения сезонности средней частоты.
   Запомним выделенную компоненту $seas_t^{(2)}$ и удалим её из ряда, $y_t^{(-1,2)} = y_t^{(-1)} - seas_t^{(2)}$.
3. …

Уточняем сезонные компоненты

Шаг 2. Корректировка сезонных компонент.

1. Временно возвращаем в полностью очищенный ряд найденную сезонность высокой частоты.
   Запускаем STL и получаем уточнённую компоненту $seas_t^{(1)}$, удаляем её из ряда и получаем уточнённый очищенный ряд.
2. Временно возвращаем в полностью очищенный ряд найденную сезонность средней частоты.
   Запускаем STL и получаем уточнённую компоненту $seas_t^{(2)}$, удаляем её из ряда и получаем уточнённый очищенный ряд.
3. …

Завершаем алгоритм

Шаг 3. Добываем тренд и остаток.

Тренд и остаток берем из самого последнего STL разложения, уточнявшего сезонные компоненты.

Много сезонных составляющих: итоги

- MSTL — быстрый и устойчивый алгоритм разложения ряда.
- Теоретически MSTL может работать с пропусками.
- Есть другие алгоритмы: ARIMA + предикторы Фурье, TBATS, PROPHET, …

Данные прерывающиеся нулями

Данные прерывающиеся нулями: план

- Нули в данных.
- Алгоритм Кростона.

Откуда нули в данных?

Счётные данные с небольшим ожиданием:

- Ежедневное количество пожаров в небольшое городе.
- Еженедельное количество завершенных писателем романов.
- …

Как моделировать?

- Специальные модели для счётных данных.
  Используют распределение Пуассона, отрицательное биномиальное, …
- Простой алгоритм Кростона.
  Подходит для несезонных данных, основан на экспоненциальном сглаживании.

Напоминание про ETS(ANN)

Уравнения модели:

$$\begin{cases} y_t = \ell_{t-1} + u_t \\ \ell_t = \ell_{t-1} + \alpha u_t \end{cases}$$

$$\ell_t = \alpha y_t + (1 - \alpha)\ell_{t-1}$$

Прогноз на 1 шаг вперёд:

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t.$$

Алгоритм Кростона

Шаг 1. Разобьём исходный ряд (y_t)

$$3, 0, 2, 0, 0, 4, 0, 0, 0, 3, 0, 1, \ldots$$

на ряд положительных значений (q_t)

$$3, 2, 4, 3, 1, \ldots$$

и длины нулевых промежутков (a_t):

$$1, 2, 3, 1, \ldots$$

Шаг 2. Применим простое экспоненциальное сглаживание.

$$\begin{cases} \hat{q}_{t+1} = \alpha_q q_t + (1 - \alpha_q)\hat{q}_t \\ \hat{a}_{t+1} = \alpha_a a_t + (1 - \alpha_a)\hat{a}_t \end{cases}$$

Параметры: α_a, α_q, $\hat{a}_0$, $\hat{q}_0$.

Прогнозирование

Из алгоритма Кростона можно извлечь:

- $\hat{q}_{T+1}$ — прогноз следующего ненулевого числа.
- $\hat{a}_{T+1}$ — прогноз длины нулевого промежутка.
- $\hat{y}_{T+1} = \hat{q}_{T+1}/\hat{a}_{T+1}$ — прогноз для исходного ряда.

Сравнение прогнозов

Используйте MASE:

$$MASE = \frac{|q_{T+1}| + |q_{T+2}| + \ldots + |q_{T+H}|}{H},$$

где

- e_t — ошибка прогноза;
- $q_t = \frac{e_t}{MAE^{naive}}$ — ошибка прогноза, отмасштабированная на среднюю абсолютную ошибку наивного прогноза.

Данные прерывающиеся нулями: итоги

- Как правило, много нулей в счётных данных.
- Алгоритм Кростона подойдёт для несезонных данных.
- Алгоритм Кростона нестатический: нет прогнозных интервалов.

Сравнение двух прогнозов

Сравнение прогнозов: план

- Тест Диболда-Мариано.
- Предпосылки теста.
- Реализация теста.

Тест Диболда-Мариано

- Предназначен для сравнения двух прогнозов.
- Сравнивает прогнозы на заданный горизонт прогнозирования h.
- Не является оптимальным для сравнения моделей.
- Не подходит для попарного сравнения множества прогнозов.

Предпосылки DM-теста

Рассмотрим разницу потерь двух прогнозов:

$$d_t = e_{A,t}^2 - e_{B,t}^2, \quad e_{\text{Model},t} = \hat{y}_{\text{Model},t} - y_t;$$

Разница d_t предполагается стационарной:

$$\mathbb{E}(d_t) = \mu_d,$$

$$\operatorname{Cov}(d_t, d_{t-k}) = \gamma_k,$$

в частности,

$$\operatorname{Var}(d_t) = \gamma_0.$$

Способ тестирования

При верной $H_0 : \mu_d = 0$:

$$DM = \frac{\bar{d}}{se(\bar{d})} \to \mathcal{N}(0; 1),$$

где $se^2(\bar{d})$ — состоятельная оценка для $\mathrm{Var}(\bar{d})$.

На практике оценивают регрессию на константу

$$\hat{d}_t = \hat{\beta}_1,$$

получают $\hat{\beta}_1 = \bar{d}$ и используют готовые робастные стандартные ошибки,

$$DM = \frac{\hat{\beta}_1}{se_{HAC}(\hat{\beta}_1)}.$$

Как устроена робастная оценка?

Сравниваем прогнозы по P точкам,

$$\operatorname{Var}(\bar{d}) = \frac{(\operatorname{Var}(d_1) + \operatorname{Var}(d_2) + \ldots + 2\operatorname{Cov}(d_1, d_2) + \ldots)}{P^2}$$

Из стационарности d_t:

$$\operatorname{Var}(\bar{d}) = \frac{P\gamma_0 + 2(P-1)\gamma_1 + 2(P-2)\gamma_2 + \ldots}{P^2}$$

Наивная оценка:

$$\widehat{\operatorname{Var}}(\bar{d}) = \frac{P\hat{\gamma}_0 + 2(P-1)\hat{\gamma}_1 + 2(P-2)\hat{\gamma}_2 + \ldots}{P^2}$$

Почему сравнение прогнозов?

Нюанс: сравнение прогнозов и сравнение моделей — разные задачи.

Модель может сильно выигрывать по простоте и немного проигрывать по прогнозам.

На малой выборке потеря информации о качестве прогнозов на обучающей выборке существенна.

На практике часто говорят «моделей».

Сравнение двух прогнозов: итоги

- Тест Диболда-Мариано подходит для сравнения двух прогнозов.
- Сравнение прогнозов и сравнение моделей — немного разные задачи.

Сравнение множества прогнозов

Сравнение множества прогнозов: план

- RC-теста Уайта.
- Стационарный бутстрэп.
- Уточнения SPA-теста.

RC-тест Уайта и SPA-тест Хансена

RC = Reality Check, проверка реальностью;

SPA = Superior Predictive Ability, превосходящее качество прогнозов.

- Два похожих теста, предназначенных для сравнения множества прогнозов с эталонным.
- Сравнивают прогнозы на заданный горизонт прогнозирования h.
- Не являются оптимальным для сравнения моделей.
- SPA-тест — более робастная вариация RC-теста.
- Предполагают оптимальные параметры всех алгоритмов.

Обозначения

- $e_{jt} = \hat{y}_{jt} - y_t$ — ошибки прогноза алгоритма j;
- Превосходство алгоритмов над эталонным в момент t:

$$d_t = \begin{pmatrix} e^2_{\text{bench},t} - e^2_{At} \\ e^2_{\text{bench},t} - e^2_{Bt} \\ e^2_{\text{bench},t} - e^2_{Ct} \\ \ldots \end{pmatrix}$$

- $\bar{d} = \sum d_t / P$ — среднее превосходство алгоритмов, P — число наблюдений, по которым идёт сравнение.

Гипотезы RC-теста:

H_0: $\mathbb{E}(d_t) = 0.$

H_a: $\max(\mathbb{E}(d_t)) > 0.$

Реализация RC-теста

1. Находим значение наилучшего среднего превосходства $RC = \max(\bar{d})$.
2. Генерируем бутстрэп-копию траектории d_t:
$$d_1, d_2, \ldots, d_P \quad \to \quad d_1^*, d_2^*, \ldots, d_P^*.$$
3. По бустрэп-копии находим для каждого алгоритма j
$$\Delta_j = \bar{d}_j^* - \bar{d}_j.$$
4. Находим $RC^* = \max(\Delta)$.
5. Повторяем (2-4) 10000 раз, получаем RC_1^*, ..., RC_{10000}^*.
6. Считаем P-значение как долю RC^*, оказавшихся больше фактического RC.

Бутстрэп-подделка

Генерируем бутстрэп-копию траектории d_t:

$$d_1, d_2, \ldots, d_P \quad \rightarrow \quad d_1^*, d_2^*, \ldots, d_P^*.$$

- Бутстрэп-подделка имеет длину исходного ряда.
- Состоит из случайных, возможно накладывающихся, фрагментов ряда.
- Длина каждого фрагмента имеет геометрическое распределение.

Уточнения SPA-теста

Стьюдентизация (нормировка) среднего превосходства каждого алгоритма.

$$\bar{d} = \begin{pmatrix} \bar{d}_A \\ \bar{d}_B \\ \bar{d}_C \\ \dots \end{pmatrix} \rightarrow d_{stud} = \begin{pmatrix} \bar{d}_A / se(\bar{d}_A) \\ \bar{d}_B / se(\bar{d}_B) \\ \bar{d}_C / se(\bar{d}_C) \\ \dots \end{pmatrix}$$

$$SPA = \max(d_{stud})$$

Уточнения SPA-теста

Предварительное деление алгоритмов на «хорошие» и «плохие».

По бустрэп-копии находим для каждого алгоритма j

$$\Delta_j = \begin{cases} \bar{d}_j^* / se(\bar{d}_j) \text{ для плохого алгоритма } j \\ (\bar{d}_j^* - \bar{d}_j) / se(\bar{d}_j) \text{ для хорошего алгоритма } j. \end{cases}$$

Сравнение множества прогнозов: итоги

- SPA-тест и RC-тест подходят для сравнения множества прогнозов.
- SPA-тест Хансена имеет большую мощность.
- Иногда названия SPA и RC путают.
- SPA-тест используют, например, для сравнения торговых стратегий.
- Сравнение прогнозов и сравнение моделей — разные задачи.