Дата последней сборки документа:
18 января 2024 г. в 05:04

Заметки по прогнозированию временных рядов

Содержание

1. Вводные замечания

Возможно самой простой формой прогнозирования является *скользящее среднее*. Часто скользящее среднее используется в качестве *метода сглаживания*, чтобы найти более гладкую линию для данных с большим количеством вариаций [2]. Каждую точку данных можно описать средним значением n окружающих точек данных, где n – обозначает размер окна. При размере окна 10 мы опишем точку данных так, чтобы ее значения представляло собой среднее значение 5 значений, предшествующих точке, и 5 значений, следующих после точки. При прогнозировании будущие значения рассчитываются как среднее n предыдущих значений, поэтому при размере окна 10 речь будет идти о среднем значении 10 предыдущих значений.

Суть сглаживания скользящим средним заключается в том, что вам нужен большой размер окна, чтобы сгладить шум и уловить фактический тренд, но при большом размере окна ваши прогнозы будут значительно отставать от тренда, поскольку вы будете использовать все более ранние наблюдения для вычисления среднего.

Идея экспоненциального сглаживания заключается в том, что мы применяем *экспоненциально уменьшающиеся веса* к усредняемым по времени значениям, придавая недавним значениям больший вес, а большее ранним значениям – меньший.

Хольт усовершенствовал технику экспоненциального сглаживания, чтобы она позволяла учитывать тренд, и назвал ее *двойным экспоненциальным сглаживанием* (double exponential smoothing). А в сотрудничестве с Винтерсом Хольт добавил поддержку сезонности в 1960 году, и техника получала название *тройного экспоненциального сглаживания* (экспоненциальное сглаживание Хольта-Винтерса).

Недостатком этих методов прогнозирования является то, что *они могут медленно приспосабливаться к новым тенденциям*, и поэтому прогнозируемые значения отстают от реальности – *они плохо работают, когда требуется более длительные горизонты прогнозирования*, и существует мнрожество гиперпараметров для настройки, что может быть трудным и очень времязатратным процессом [2, стр. 14].

ARIMA и модель Бокса-Дженкинса часто используется как вазимозаменяемые термины, хотя технически модель Бокса-Дженкинса относится к методу оптимизации параметров для ARIMA-модели.

ARIMA – это аббривиатура трех понятий: *автогрессия*, *интегрированное* и *скользящее среднее*. *Авторегрессия* означает, что модель использует зависимость между точкой данных и некоторым количеством запаздывающих точек данных (лагов). То есть модель делает прогноы на основе предыдущих значений. Это похоже на предсказание того, что завтра будет тепло, потому что до сих пор было тепло всю неделю.

Интегрированное означает, что вместо применения любой исходной точки данных используется разница между этой точкой данных и некоторой предыдущей точкой данных. По сути, это означает, что мы преобразуем ряд значений в ряд изменений значений. Интуитивно это предпалагает, что завтра будет более или менее такая же температура, как сегодня, потому что температура всю неделю сильно не менялась.

Проблема с ARIMA-моделями заключается в том, что они не поддерживают сезонность или данные с повторяющимися циклами, такими как повышение температуры днем и падение ночью или повышение летом и падение зимой. SARIMA (Seasonal ARIMA) бала разработана для преодоления данного недостатка. Подобно нотации ARIMA, нотация для модели SARIMA представляет собой $SARIMA(p,d,q)(P,D,Q)m$, где P – порядок сезонной авторегрессии, D – порядок сезонного дифференциирования, Q – порядок сезонного скользящего среднего, а m – периодичность (количество периодов в полном сезонном цикле).

VARIMA для случаев с несколькими временными рядами в качестве векторов. FARIMA – дробная ARIMA и ARFIMA – дробная проинтегрированная ARIMA. Последние две включают дробную степень дифференциирования, обеспечивающую длительную память в том смысле, что наблюдения, удаленные друг от друга с точки зрения времени, могут иметь несущественные зависимости.

SARIMAX – сезонная ARIMA, где X означает экзогенные или дополнительные переменные, добавленные в модель, например добавление прогнгоза осадков в модель прогнозирования температуры.

ARIMA обычно показывает очень хорошие результаты, но недостатками являются сложность подбора параметров и необходимость тщательного разведочного анализа. Настройка и оптимизация моделей ARIMA часто требуют значительных вычислительных ресурсов, а успешные результаты могут зависеть от навыков и опыта прогнозиста.

Когда дисперсия данных не является постоянной во времени, ARIMA-модели сталкиваются с проблемами [2, стр. 16]. В экономике и финансах непостоянство дисперсии может быть обычным явлением. Для решения этой проблемы были разработаны модели *авторегрессии условной гетероскедостичности* (Autoregressive Conditional Heteroscedasticity – ARCH). Гетероскедостичность – это способ сказать, что дисперсия или разброс данных не являются постоянными повсюду, а противоположенным термином является гомоскедостичность.

Тим Боллерслев и Стивен Тейлор в 1986 году дополнили модель ARCH компонентой *скользящего среднего*, предложив модель Generalized ARCH, или GARCH. Когда колебания случайны, может быть полезна GARCH.

Обе модели ARCH и GARCH не могут обрабатывать ни тренд, ни сезонность, хотя на практике часто для работы с *сезонными колебаниями* и *трендом* временного ряда применяется ARIMA-модель, а затем для моделирования *ожидаемой дисперсии* используется ARCH-модель.

Градиентный бустинг сейчас стал популярен для прогнозирования временных рядов. Следует помнить, что деревья не умеют экстраполировать! Если временной ряд содержит тренд, можно попробовать детрендинг. Мы удаляем из ряда тренд, предварительно спрогнозированный ли-

нейной моделью. На полученных остатках обучаем градиентный бустинг и получаем прогнозы, к ним добавляем тренд. Преимуществом градиентного бустинга является способность фиксировать нелинейные зависимости и взаимодействия высокого порядка.

2. Математический аппарат Facebook Prophet

2.1. Библиотека Facebook Prophet

Используется модель разложимых временных рядов (Harvey & Peters, 1990) с тремя основными компонентами:

- тренд,
- сезонность,
- праздники

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

где $g(t)$ – функция тренда, моделирующая непериодические изменения значений временного ряда, $s(t)$ представляет собой периодические изменения (например, еженедельную и ежегодную сезонность), а $h(t)$ – представляет собой эффекты праздников, которые возникают в течение одного или нескольких дней. Член ошибки ε_t представляет собой любые случайные возмущения, которые не учитываются моделью.

Эта спецификация аналогична *обобщенной аддитивной модели* (GAM) (Hastie& Tibshirani, 1987), классу регрессионных моделей с потенциально нелинейными сглаживаниями, применяемыми к регрессорам.

Выбор в пользу GAM имеет то преимущество, что она легко декомпозируется и допускает включение новых компонент по мере необходимости, например при выявлении нового источника сезонности. GAM также обучается очень быстро, либо с помощью бэкфиттинга, либо с помощью L-BFGS (Byrd et al., 1995) (мы предпочитаем последнее), чтобы пользователь мог интерактивно изменять параметры модели.

По сути, мы формулируем проблему прогнозирования как задачу подгонки кривой, которая внутренне отличается от моделей временных рядов, поскольку те явно учитывают структуру временной зависимости в данных. Хотя мы отказываемся от некоторых важных преимуществ использования генеративной модели типа ARIMA, выбор в пользу *GAM* обеспечивает ряд практических преимуществ [2, стр. 24]:

- гибкость, заключающуюся в том, что мы можем *легко учесть сезонность с несколькими периодами* и позволить аналитикам выдвинуть разные предположения о трендах,
- в отличие от моделей ARIMA, *измерения не должны находиться на одинаковом расстоянии друг от друга* и нам не нужно интерполировать пропущенные значения, возникшие, например, по причине удаления выборосов,
- обучение выполняется очень быстро, что позволяет аналитику интерактивно исследовать большое количество спецификаций модели,
- прогнозная модель имеет легко интерпретируемые параметры, которые аналитик может изменить согласно выдвинутым предположениям относительно прогнозов.

2.1.1. Модель тренда

В Facebook реализованы две модели тренда:

- кусочно-логистическая модель роста с насыщением,
- кусочно-линейную модель.

Нелинейные рост с насыщением Для прогнозирования роста основной компонентой процесса генерации данных является модель, предсказывающая, как выросла численность население и как она будет расти дальше. Моделирование роста в Facebook обычно похоже на рост населения в естественных экосистемах, где наблюдается нелинейный рост, который насыщается при предельной пропускной способности. Предельной пропускной способностью для количества пользователей Facebook в определенном регионе может быть количество людей, имеющих доступ к сети Интернет. Такой рост обычно моделируется с помощью логистической модели роста, которая имеет вид

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

где C – верхний порог (пропускная способность), k – скорость роста, m – параметр смещения, позволяющий сдвигать функцию вдоль оси времени.

Предельная пропускная способность непостоянна, поскольку количество людей в мире, которые имеют доступ к Интернету, увеличивается. Таким образом, мы заменяем фиксированную пропускную способность C на изменяющуюся во времени пропускную способность $C(t)$. Во-вторых, скорость роста непостоянна.

Мы включаем изменения тренда в модель роста, явно определяя точки изменения (change points), в которых скорость роста может измениться.

Тогда кусочно-логистическая модель роста принимает вид [2, стр. 26]

$$g(t) = \frac{C(t)}{1 + \exp\left(-(k + a(t)^T)\delta)(t - (m + a(t)^T)\gamma\right)}$$

По сути $C(t)$ – это функция верхней границы тренда.

Линейный тренд с точками изменения При прогнозировании задач, в которых нет роста с насыщением, кусочно-постоянная скорость роста дает экономную и часто полезную модель. В таком случае модель тренда выглядит следующим образом

$$g(t) = (i + a(t)^T\delta)t + (m + a(t)^T\gamma),$$

где k – скорость роста, δ содержит корректировки скорости, m – параметр смещения, а γ_i устанавливается равной $-s_j\delta_j$, чтобы функция была непрерывной.

Автоматический отбор можно выполнить вполне естественным путем, выбрав априорное распределение Лапласа для δ. Что немаловажно, применение распределения Лапласа для корректировок δ не влияет на первоначальную скорость роста τ, поэтому когда τ стремится к 0, обучение сводится к стандартному (а не кусочному) логистическому или линейному росту.

Когда происходит экстраполяция модели за пределы исторических данных для получения прогноза, тренд получает постоянную скорость. Каждая из этих точек имеет изменение скорости $\delta_j \approx Lapalce(0, \tau)$. Параметр τ – коэффициент масштаба для автоматического выбора точек

смены тренда. Мы симулируем значения будущих изменений скорости, которые подражают прошлым значениям путем замены τ на дисперсию, оцениваемую по имеющимся данным.

Предположение, что тренд продолжит меняться с той же частотой и скоростью изменений, что и в исторических данных, является довольно сильным, поэтому мы не ожидаем высокой точности от доваерительных интервалов. Однако они являются полезным показателем уровня неопределенности и в особенности *показателем переобучения*.

Временные ряды в бизнес-задачах часто имеют *многопериодную сезонность* как результат человеческого поведения, которые они отражают. Например, 5-дневная рабочая неделя может оказывать влияние на временной ряд, повторяющееся каждую неделю, в то время как графики отпусков и школьных каникул могут вызывать эффекты, повторяющиеся каждый год.

Мы предложили использовать ряды Фурье, чтобы получить гибкую модель периодических изменений (Harvey & Shephard, 1993). Пусть P – постоянное значение периода для рассматриваемого временного ряда (например, $P = 365.2$ для годовых данных или $P = 7$ для недельных данных). Мы можем аппроксимировать произвольные сезонные эффекты с помощью стандартного ряда Фурье

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

Подгонка сезонности требует оценки $2N$ параметров $[a_1, b_1, \ldots, a_N, b_N]^T$. Это делается путем построения матрицы векторов сезонности для каждого значения t в наших исторических и прогнозных данных, например ниже приведен пример для годовой сезонности и $N = 10$

$$X(t) = \left[\cos \left(\frac{2\pi \cdot 1 \cdot t}{365.25} \right), \ldots, \sin \left(\frac{2\pi \cdot 10 \cdot t}{365.25} \right) \right]$$

Тогда сезонная компонента будет иметь вид

$$s(t) = X(t)\beta.$$

В нашей генеративной модели мы берем $\beta \sim Normal(0, \sigma^2)$, чтобы сгладить сезонность. Для годовой и недельной сезонности были найдены значения $N = 10$ и $N = 3$ соответственно, которые работают достаточно хорошо для большинства задач. Выбори этих параметров можно автоматизировать с помощью критерия отбора модели типа AIC [2, стр. 28].

Для оценивания параметров обучаемой модели используются принципы байесовской статистики: либо поиск апостериорного максимума (MAP) с помощью L-BFGS, либо полный байесовский вывод.

Выбрать наблюдения из данного диапазона

```
df.set_index("date").between_time("8:00", "18:00")  # работает только временными индексами
# или так
df[(df["date"].dt.hour >= 8) & (df["date"].dt.hour < 18)].set_index("date")
```

Список литературы

1. *Маккинли У.* Python и анализ данных, 2015. – 482 с.

2. *Груздев А.* Прогнозирование временных рядов с помощью Facebook, Prophet, ETNA, sktime и LinkedIn Greykite: Строим, настраиваем, улучшаем модели прогнозирования временных рядов с помощью специальных библиотек. – М.: ДМК Пресс, 2023. – 780 с.