



★ 4.18

Оценка

51.65

Рейтинг

Цифра

Разрабатываем и внедряем промышленные ИТ-решения

Подписан



piero

22 июл 2022 в 14:01



Еще один подход к аугментации табличных данных

🕒 12 мин 👁 3.5K

Блог компании Цифра, Алгоритмы*, Машинное обучение*, Искусственный интеллект

Известно множество практик борьбы с выбросами во входных данных – применяются как статистические методы, так и *ML/DL*-подходы. Каждый специалист по анализу данных использует их в процессе предобработки данных. Однако часто при этом отсеиваются исключительно ценные замеры, которые на самом деле выбросами не являются, а представляют собой редко встречающиеся состояния измеряемого объекта или процесса.

"А стоит ли возиться с этими "редкими" состояниями?", – спросит нас любознательный читатель. "Да", – уверенно ответим мы и далее приведем один простой и показательный пример, а сразу за ним расскажем, как извлечь пользу из этих редких состояний. Конкретнее – мы предложим подход для генерации дополнительных измерений в задачах восстановления регрессии временного ряда.

Под "мы" здесь и далее имеются в виду я – Семён Косяченко – и мои коллеги Александр Подвойский и Александр Калиниченко. Вместе мы разрабатываем решения на основе *ML* для нефтегазовой отрасли в компании "Цифровая индустриальная платформа" (совместное предприятие ГК "Цифра" и "Газпром нефть").

Простой, но крайне показательный пример

Допустим, у вас есть масляный трансформатор, температура масла которого измеряется раз в час. Пусть этот трансформатор почти всегда находится в работоспособном состоянии и имеет температуру масла 20-100°C. Но есть один замер, когда температура масла была более 200°C, при этом трансформатор неработоспособен (еще бы, он горит!!!).

Что сделает специалист по анализу данных, в предобработке? Правильно, первым делом он



+3



29



3

трансформатора. Точнее сказать, сам замер не так уж и важен – нас не слишком волнует состояние трансформатора во время пожара. Его поздно анализировать – его тушить пора. Но, кажется, есть возможность интерполировать пространство <температура – состояние трансформатора> и понять, что повышение температуры ведет к его неработоспособности.

Как же это сделать, когда есть единственный замер, при котором трансформатор горит? "Очень просто", – ответим мы (на самом деле нет). Предположим, что температура и работоспособность трансформатора связаны, например, линейно. Тогда, имея нормальное и "горящее" состояние трансформатора, можно дополнить историю измерений некоторыми промежуточными состояниями – сгенерировать эти состояния. Это называется аугментацией данных, увеличением обучающей выборки через добавление в нее дополнительных данных – модификаций измерений из имеющейся выборки.

Аугментацию широко применяют в предпроцессинге для повышения качества обучаемых моделей в таких областях, как классификация изображений и биологических сигналов, распознавание речи и прочее. В этой статье мы будем говорить о временных рядах, представленных в табличном виде. Задача восстановления регрессии временного ряда – одна из наиболее часто встречающихся на практике, поэтому мы постарались сделать работоспособный, как нам кажется, модуль, которым можно пользоваться. Скачать его можно [здесь](#).

Для аугментации данных используют различные библиотеки, например, `tabgan`, которые реализуют генерацию (аугментацию) как табличных, так и других видов данных. Процесс аугментации табличных данных упрощенно можно описать так: пусть есть таблица T которая состоит из столбцов – признаков и строк – измерений. Признаки этой таблицы имеют неизвестное совместное распределение P . Каждая строка независимо выбирается из P . Задача состоит в обучении генеративной модели. Эта модель будет иметь способность генерировать синтетические данные T_{sync} с распределением, подобным P . Обучение модели на T_{sync} может во многих случаях добиться лучшего качества и робастности модели по сравнению с моделью, построенной на исходной выборке T .

Наш подход к аугментации

Пусть задача определяется пакетом временных рядов как описательными признаками системы, связанными с целевым вектором y . Целевой вектор представляет собой временной ряд той же частоты. Этот пакет временных рядов и целевой вектор будут подвержены аугментации.

Допустим, что наш пакет временных рядов предварительно очищен от выбросов, а все измерения, выглядящие как выброс, имеют физический смысл. Тогда наша задача – выявить **статистически значимые изолированные наблюдения**, которые можно ошибочно принять за выбросы, и обогатить окрестности этих изолированных наблюдений синтетическими данными. Синтетические данные здесь строятся в предположении линейной связи между ординатами признаков в оговоренной окрестности.

Таким образом, наш подход к аугментации можно описать так:

1. Проверяем целевой временной ряд на наличие статистически значимых отклонений с помощью модифицированной Z-оценки. Если таких отклонений нет, то алгоритм аугментации заканчивает работу. Если есть, то для всех найденных измерений с отклонениями выполняем пункт 2 и 3.
2. Определяем окрестность области линейной интерполяции данных. Правая и левая границы окрестности – ближайшие соседние измерения.
3. Производим аугментацию данных в определенной на предыдущем шаге области методом линейной интерполяции. В результате мы получим расширенную выборку, составленную на основе имеющихся измерений. Наглядно показано на Рисунке 1.

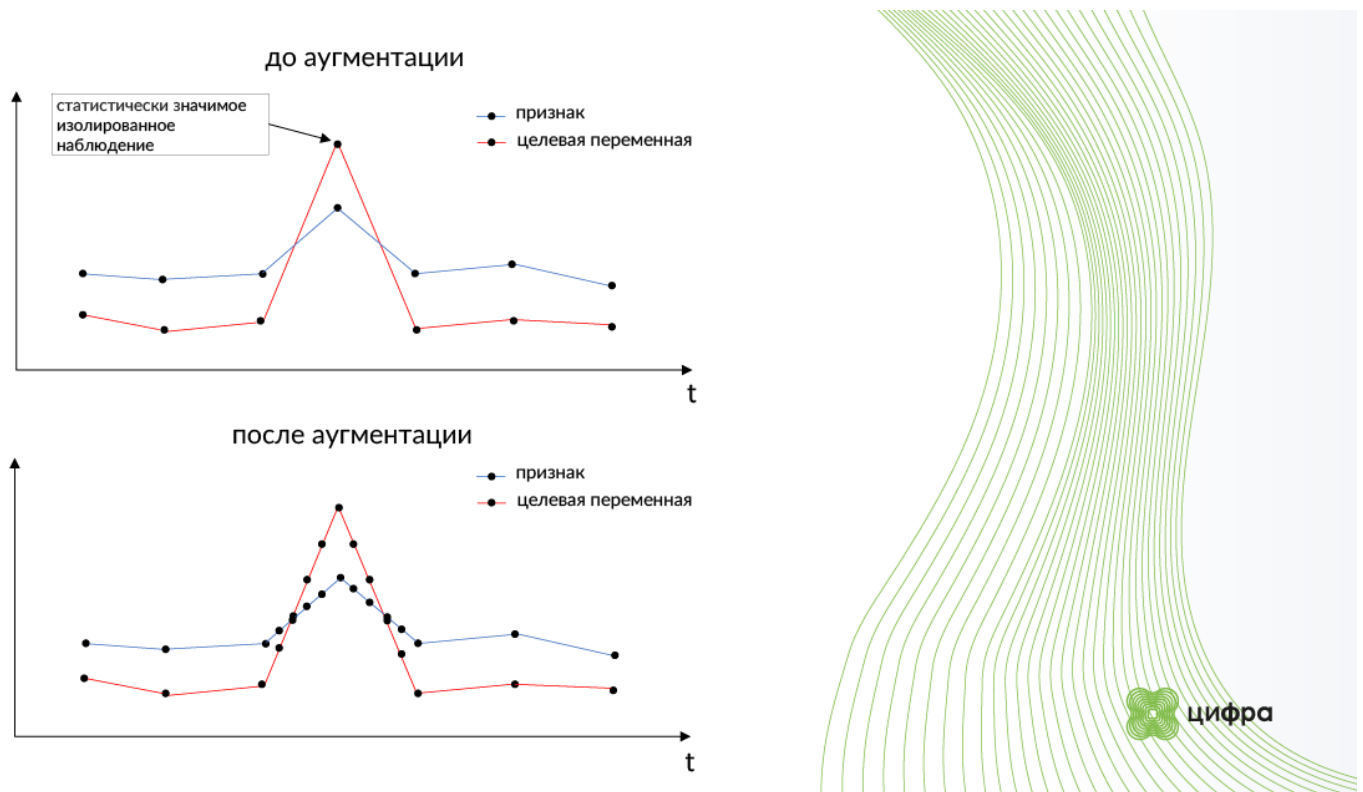


Рисунок 1. Пример исходных данных (сверху) и аугментированных данных (снизу). По оси x – время, по оси y – значение фичи и целевой переменной.

Описанный нами подход имеет два основных гиперпараметра. Первый – частота или интенсивность аугментации (т. е. количество новых аугментированных измерений) – определяется гиперпараметром N . Второй – гиперпараметр K – определяет порог для сравнения модифицированной Z-оценкой. Чем выше этот порог, тем меньше окрестностей статистически значимых состояний будут подвергнуты аугментации.

Как мы оценивали наш подход

Точность обученной модели мы проверяли с помощью *SMAPE* (симметричный средний процент отклонения). Так как наши данные представляют собой временные ряды, выбор именно этой метрики кажется наиболее логичным. Вычисляется *SMAPE* по формуле:

$$SMAPE = \frac{\sum_{n=0}^{M-1} |y_n - y_n^b|}{\sum_{n=0}^{M-1} |y_n + y_n^b|}$$

где, y_n – истинные значения целевой переменной, y_n^b – предсказание модели, n – индекс измерения, M – количество измерений в выборке.

Здесь следует сделать несколько ремарок:

1. $SMAPE$ плохо работает в окрестности нуля. Поэтому, для простоты всех последующих рассуждений, скажем, что в наших исследованиях целевая переменная y всегда была больше 1.
2. Предлагаемый нами прием аугментации предполагает, что временной ряд обладает признаком *стационарности* – статистические характеристики ряда (выборочное среднее, дисперсия) не изменяются с течением времени. Проверить гипотезу о стационарности можно с помощью одного из тестов на единичные корни (Unit root test), например, с помощью расширенного теста Дики-Фуллера или с помощью теста КПСС. Если же у вас есть основания считать, что временной ряд адекватно описывается аддитивной моделью (модель представляет собой сумму несущего тренда и случайного сигнала), скажем, с гармоническим трендом, то можно выявить этот тренд простым экспоненциально-взвешенным средним, а затем удалить из исходного временного ряда ординаты тренда и привести его таким образом к стационарному состоянию, а далее применять предлагаемую нами технику.

Мы сделали генератор временных рядов и провели качественный и количественный анализ работы нашего алгоритма аугментации на этих искусственно сгенерированных примерах. Все примеры есть в [репозитории](#). Здесь же разберем два из них (внимание: нумерация примеров дана так же, как они представлены в репозитории, – так будет легче ориентироваться).

Сгенерированный пример 3

Мы взяли один стационарный и равномерно зашумленный параметр x при этом целевая переменная y сильно коррелировала с x . Во временной ряд мы добавили резкие всплески, которые обычно плохо восстанавливаются стандартными ML -моделями, провели обучение на истории до красной вертикальной линии (см. Рис. 2) и восстановили тестовую часть – после красной линии. Далее мы оценили результат по $SMAPE$ и... получили отличные результаты: качество модели на аугментированных данных лучше, чем качество модели, построенной на исходных данных (см. Таблицу 1). В этом эксперименте и далее в качестве модели мы использовали Random forest с 200 деревьями – нам нужен был быстрый нелинейный регрессор.

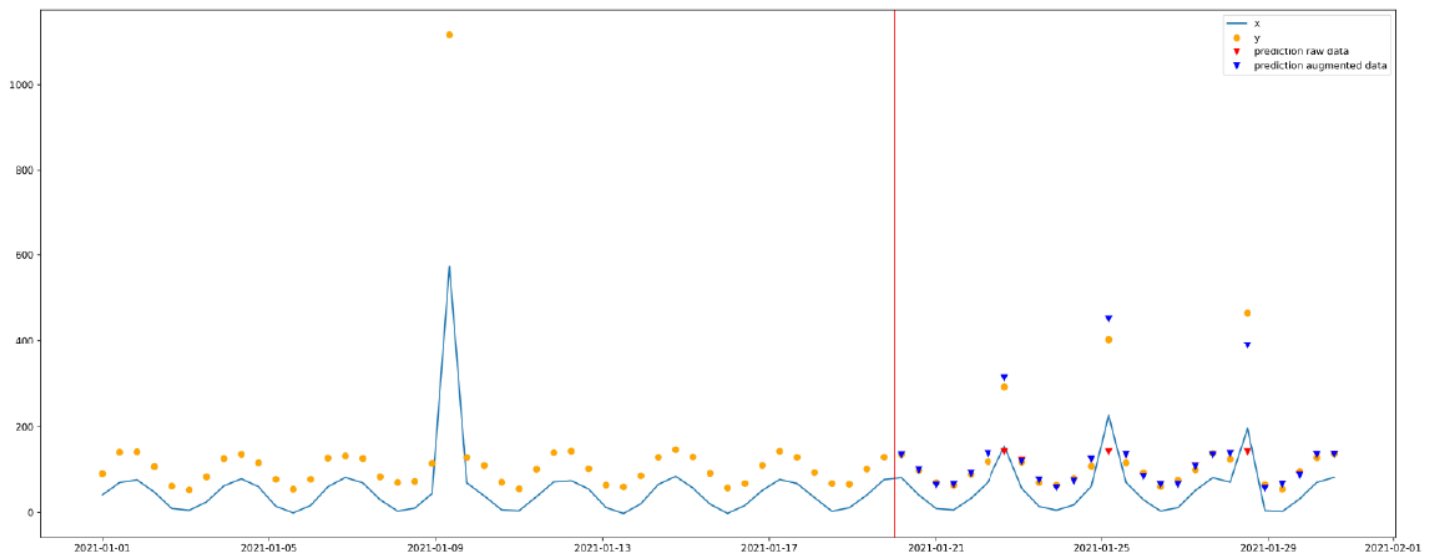


Рисунок 2. Результат применения модели на сырых и аугментированных данных. Красная вертикальная линия разделяет тренировочную и тестовую выборку.

Сгенерированный пример 5

В этом примере мы взяли аддитивно-мультипликативную модель процесса с гармоническим трендом, случайным несущим процессом и автокорреляционной функцией заданного типа. В гармонической составляющей можно управлять смещением фазы, амплитудой, частотой, а в случайной – стандартным отклонением.

Заданный стационарный нормальный непрерывный случайный процесс строится в виде дискретной последовательности в соответствии с этой формулой:

$$\xi_n = a_0 q_n + b_1 \xi(n-1) \quad (2)$$

где, q_n – последовательность независимых нормальных случайных величин ($N(0, 1)$), n - номер замера временного ряда, $n \in [0, L-1]$, L – количество замеров временного ряда.

$$a_0 = \sigma \sqrt{1 - \rho^2}, \quad b_1 = \rho, \quad \rho = e^{-\omega \Delta t}$$

где Δt - шаг дискретизации, σ, ω – некоторые параметры временного ряда.

Мы сгенерировали четыре независимых случайных процесса (x_{in} с разными σ, ω) по вышеприведенной формуле. Кроме этого, во втором из четырех процессов мы добавили "выбросы" в тренировочной и тестовой выборке и для более наглядной визуализации сдвинули и смасштабировали сгенерированные ряды так, чтобы они не "налезали" друг на друга на графике. Таким образом:

$$x_{in} = \xi(\sigma_i, \omega_i) \cdot multi_i + adder_i + v_{in}, (3)$$

где x_{in} – сгенерированный ряд с параметрами σ_i, ω_i , i – индекс номера временного ряда, $i \in [0, 3]$. $multi_i, adder_i$ – параметры масштабирования и сдвига соответственно, v_{in} – компонента, отвечающая за выброс.

v_{in} почти всегда равно нулю, кроме нескольких специальных точек, которые в наших экспериментах задавались вручную. На Рис.3 ненулевые v_{in} можно видеть в моменты времени, когда y существенно выше среднего значения.

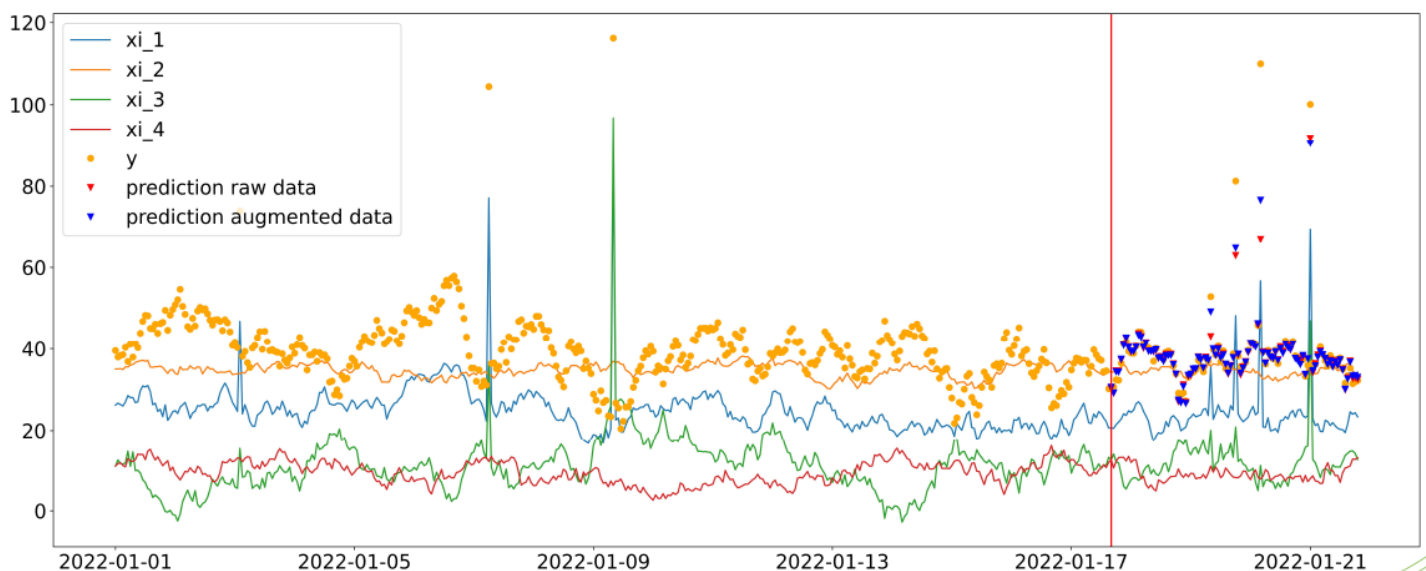
Целевой ряд y был сгенерирован как взвешенная сумма x_{in}

$$y_n = \sum_{i=0}^3 w_i \cdot x_{in}, \quad (4)$$

где w_i – весовой коэффициент i -ого ряда.

В итоге мы получили пакет временных рядов длиной в 500 измерений. 400 первых измерений мы использовали для обучения модели, 100 последних – для тестирования.

Далее, мы провели аугментацию тренировочной выборки предлагаемым методом и сравнили результаты моделей, обученных на аугментированных и исходных выборках. Результаты можно видеть на Рисунке 3. Отметим, что модель, построенная на аугментированных данных, лучше обрабатывает "выбросы". Количественные оценки моделей приведены в **Таблице 1**.



Для чего нам потребовалась аугментация?

Как вы, наверное, уже догадались, мы занимались аугментацией данных не просто так – у нас была реальная производственная задача: нужно было восстановить истинный дебит нефти на нефтяном месторождении.

Дебит – это объем жидкости (или газа), добываемой из скважины за единицу времени (сутки, час и др.). Может характеризовать добычу нефти, газа, газоконденсата, воды.

Представьте себе месторождение: сотни скважин качают нефть из-под земли на поверхность, и затем она направляется в общую трубу – коллектор. На поверхности устья скважины группируются в так называемые кусты. Каждый куст – это примерно 4-12 скважин. При этом расходомер, который измеряет дебит, установлен не на каждой скважине, как можно было бы предположить, а на каждом кусте. Принцип работы простой: расходомер измеряет дебит с одной скважины в течение нескольких часов и затем переключается на другую скважину, измеряет ее дебит в течение нескольких часов и далее переключается на третью. Таким образом дебит каждой скважины замеряется только 1-4 часа в день, а все остальное время этот показатель нам неизвестен.

Задача состоит в том, чтобы восстановить почасовой дебит каждой скважины в периоды времени, когда измеряются другие скважины. При этом постоянно измеряются следующие параметры: сила тока, напряжение, температура смеси, температура двигателя, нагрузка двигателя, частота тока, количество качаний (если насос штанговый), давление в линии, на буфере и на входе в насос.

Продуктовое решение данной проблемы предполагает глубокий подход к построению дополнительных параметров для модели (фича-инжиниринг), но в приведенных экспериментах для простоты эти шаги предобработки данных опущены. Также мы не останавливались на других обязательных шагах предобработки данных – фильтрации и др.

Чтобы проверить наш подход на реальных производственных данных, мы взяли настоящие данные 12 скважин с двух месторождений (9 с первого и 3 со второго) и провели с ними ровно те же операции по аугментации, что и с искусственно сгенерированными выборками. Все данные были анонимизированы и зашумлены. Каждый из имеющихся у нас временных пакетов мы разбили на тренировочную и тестовую выборку в соотношении примерно 2 к 1, как и с искусственно сгенерированными выборками. Модели обучались отдельно на каждом пакете данных. Далее мы проверили качество обученных моделей по *SMAPE*, результаты смотрите в **Таблице 1** по реальным примерам (номера примеров так же, как и в случае с искусственно сгенерированными, даны так, как они представлены в репозитории).

Как вы видите в **Таблице 1**, мы не ограничились сравнением моделей, обученных на исходных и аугментированных по нашей методологии данных, а кроме этого проверили, как наш подход к

аугментации показал себя в сравнении с tabgan. Подробнее про tabgan можно прочитать [здесь](#). Результаты получились разгромные для tabgan. Очевидно, основная идея tabgan – лучше описать функцию совместного распределения параметров из пакета временного ряда – провалилась, тогда как предлагаемый нами подход аугментации, который сфокусирован на восстановлении зависимости в плохо описанной области, в среднем, работал удачнее.

Имя сэмпла данных	<i>SMAPE</i> исходных данных	<i>SMAPE</i> аугментированных данных	<i>SMAPE</i> tabgan
сгенерированный пример 1	0,12	0,02	0,28
сгенерированный пример 2	0,07	0,16	0,27
сгенерированный пример 3	0,15	0,05	0,32
сгенерированный пример 4	0,06	0,06	0,04
реальный пример 1	0,03	0,03	0,03
реальный пример 2	0,01	0,02	0,01
реальный пример 3	0,05	0,07	0,06
реальный пример 4	0,15	0,13	0,13
реальный пример 5	0,06	0,05	0,07
реальный пример 6	0,05	0,05	0,03
реальный пример 7	0,08	0,07	0,09
реальный пример 8	0,03	0,03	0,03
реальный пример 9	0,06	0,04	0,21
реальный пример 10	0,05	0,07	0,05
реальный пример 11	0,05	0,03	0,05
реальный пример 12	0,06	0,06	1,02
среднее качество по <i>SMAPE</i>	0,068	0,060	0,24

Таблица 1. Результаты проведенных экспериментов по SMAPE-метрике.

**Все примеры, кроме сгенерированных, могут быть найдены в папке examples->data (см. репозиторий). Сгенерированные данные были получены из генераторов данных (data_generators). Все random_seed зафиксированы для воспроизводимости экспериментов.*

Заметим, что все описанные выше эксперименты проводились с лучшими найденными и зафиксированными гиперпараметрами N и K для предлагаемого подхода аугментации и лучшими найденными гиперпараметрами для tabgan. Более того, мы не поленились и провели дополнительные эксперименты для проверки робастности результата в рамках однородных данных. А именно, мы пробовали найти оптимальные значения гиперпараметров для первых шести скважин первого месторождения и применить их на трех оставшихся скважинах того же месторождения. Результаты тоже положительные: подобранные оптимальные значения гиперпараметров можно переносить между однородными временными пакетами. Эти эксперименты найдете в нашем репозитории.

Еще в ходе экспериментов мы поняли, что tabgan надо "уметь готовить". С прискорбием должны сообщить, что мы этого делать не умеем. У нас так и не получилось подобрать гиперпараметры таким образом, чтобы получить стабильное "неухудшение" качества модели. Наша попытка найти удачное применение tabgan к задаче аугментации временного ряда не увенчалась успехом, хотя обратные примеры встречаются – они описаны здесь, например.

Кстати, мы выявили существенную разницу во времени работы tabgan и предлагаемого нами подхода аугментации. В нашем подходе время на построение аугментированной выборки зависит от данных и растет линейно в зависимости от количества измерений в исходной выборке. Если не найдены "выбросы" в данных, то алгоритм заканчивает работу без результата практически мгновенно. Во всех наших примерах аугментация занимала доли секунды. Tabgan работает долго и упорно (и часто с отрицательным результатом), время генерации данных при этом, кажется, нелинейно зависит от длины входной выборки, на больших датасетах генерация может занимать минуты и даже часы. Мы попробовали запустить tabgan с настройками по умолчанию на датафрейме из 65 тысяч строк и сильно расстроились. Ниже представлена **Таблица 2**, в которой мы показали, как много времени тратится на аугментацию нашим методом и tabgan в зависимости от количества строк в датафрейме. Еще tabgan потребляет гораздо больше памяти (хотя количественных замеров мы не проводили). Опять же заметим, что в совсем коротких датасетах tabgan вообще не смог произвести аугментацию.

имя сэмпла данных	augmented data (сек.)	tabgan (сек.)	количество строк в датафрейме
сгенерированный пример 4	0.01	ошибка обработки	660
реальный пример 3	0.04	13.3	3650

реальный пример 13	0.15	22 минуты	65 000
реальный пример 6	0.06	6.6 секунд	6500

Подведем итог

По сравнению с tabgan наш метод аугментации работает очень быстро, он легкий, элементарно интерпретируется и в своей нише применения позволяет добиться лучшего качества применяемой модели, чем tabgan. При правильно подобранных гиперпараметрах он стабильно не ухудшает качество модели.

В вышеприведенном абзаце стоит подчеркнуть **"в своей нише"**. Что же это за ниша? Во-первых, скорее всего, это временные ряды, по крайней мере, все свои эксперименты мы проводили над временными рядами и бессовестно эксплуатировали свойство соседства измерений. Во-вторых, если внимательно посмотреть на графики примеров, а еще лучше – поковыряться в нашем репозитории, то станет понятно, что мы реализуем, по большому счету, один паттерн в данных. Этот паттерн можно описать как "наличие статистически-значимого отдельно стоящего измерения, которое коррелирует с изменением хотя бы одной фичи временного ряда". В рассмотренных нами примерах как раз содержится такой паттерн. Если мы интерполируем пространство между этим "выбросом" и основным облаком данных, то состояния из тестовой выборки, попавшие в эту (интерполируемую) окрестность, скорее всего будут предсказываться лучше, чем без таковой интерполяции.

Широка ли эта ниша применения? Совершенно точно нет. По сути, мы впервые столкнулись с таким паттерном за долгую практику и, тем не менее, если в ваших данных встречаются подобные "выбросы", то не торопитесь их выбрасывать. Вполне возможно, что вам удастся с ними подружиться, как удалось нам.

Полезные ссылки:

1. [Репозиторий Tabgan](#).
2. [Использование tabgan для аугментации табличных данных 1](#)
3. [Использование tabgan для аугментации табличных данных 2](#)
4. [Наш репозиторий](#).
5. Быков В. В. Цифровое моделирование в статистической радиотехнике. Изд. "Советское радио", 1971. Стр. 99
6. [Тест КПСС](#).

Теги: data science, data mining, data analysis, outlier detection, augmentation, statistics, mach, tabgan, numerical methods, simulation

Хабы: Блог компании Цифра, Алгоритмы, Машинное обучение, Искусственный интеллект



Цифра

Разрабатываем и внедряем промышленные ИТ-решения

Подписан



[Сайт](#) [Telegram](#) [ВКонтакте](#)



8

Карма

0

Рейтинг

Семен @piero

анализ данных

Комментарии 3



dendron

22 июл 2022 в 23:11



А можно для абсолютно несведущих в нейросетях (а так же в "робасности", гипер-параметрах и SMAPE) пояснить в чём отличие изложенного в статье от, собственно, линейной интерполяции?



+2

Ответить



PyLeor

25 июл 2022 в 14:02



В статье используется прием локальной интерполяции для окрестности изолированного физически значимого измерения. То есть предполагается, что рассматриваемый набор данных позволяет без значимых искажений считать связи в пакете временных рядов в локальной окрестности изолированного точки линейными. Что-то близкое по концепции используется в нелинейной модели снижения размерности LLE <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.LocallyLinearEmbedding.html>



+2

Ответить



4knowledge

6 янв в 15:40

Спасибо, что обратили внимание на библиотеку tabgan. Для генерации пробовали использовать режим с временными рядами с TimeGAN?

Второй момент, тут звучит больше как задача ловля аномалий во временных рядах
<https://medium.com/towards-data-science/anomaly-detection-time-series-4c661f6f165f>

 0


Ответить

Ваш комментарий


Нажмите "/" для вызова меню

Отправить



Публикации


ЛУЧШИЕ ЗА СУТКИ ПОХОЖИЕ





bodyawm

5 часов назад


Эти кристаллы доживают последние деньки — почему мощные процессоры и видеокарты середины нулевых умирают?


 Простой


 9 мин


 13K

Ретроспектива

 +47

 31


 25





ru_vds

вчера в 16:00

Как зловердные приложения могут скрывать работу с буфером обмена в Android 14


 Средний


 8 мин


 2.2K


Кейс

Перевод

 +34

 39

 2



anton-artemenko

6 часов назад

Как работает врачебная тайна

🕒 11 мин 👁 2.3K

💎 +29

🔖 12

💬 17



CyberexTech

23 часа назад

«Глупый» электросчетчик VS умный дом или как интегрировать электросчетчик без смарт функций в Home Assistant

🟢 Простой

🕒 6 мин

👁 8.5K

Кейс

💎 +26

🔖 78

💬 44



iliazeus

4 часа назад

Как и почему я писал для Флиппера на Си-с-классами

💧 Средний

🕒 8 мин

👁 1.5K

Кейс

💎 +22

🔖 19

💬 4



Revertis

23 часа назад

Предстоящий выпуск Yggdrasil v0.5

💧 Средний

🕒 9 мин

👁 2.6K

Перевод

💎 +19

🔖 14

💬 2



AlexAgree

5 часов назад

Анализ 10 000 вопросов с технических интервью: частотность и вероятность встречи

🟢 Простой

🕒 3 мин

👁 3.4K

Из песочницы

💎 +18

🔖 44

💬 7



OlgaRode

3 часа назад

Киберпреступник глазами российских психологов: черты, мотивы, ценности, отклонения

Простой

14 мин

1.2K

+16

18

3



SLY_G

19 часов назад

Дайджест научпоп-новостей за неделю, о которых мы ничего не писали

7 мин

2.8K

Дайджест

+16

9

6



MaxRokatansky

22 часа назад

Малоизвестные библиотеки Python для анализа данных, которые сделают вашу жизнь проще

Средний

14 мин

8.4K

Обзор

+14

140

3

Показать еще

ИНФОРМАЦИЯ

Сайт	www.zyfra.com
Дата регистрации	7 июля 2021
Дата основания	5 сентября 2017
Численность	1 001–5 000 человек
Местоположение	Россия

4 часа назад

Как концепция EOM и графический конфигуратор моделей в ZIoT облегчают жизнь промышленным разработчикам

 152  0

16 окт в 10:33

Wish You Were Here или Как найти коллегу, который захочет остаться

 3K  10

30 авг в 14:06

Модульное тестирование производительности в Kubernetes или Как мы выиграли время на чай и здоровый сон для НТ-инженера

 1.9K  1

7 авг в 13:23

Что такое «хорошо» и что такое «плохо» в NiFi. Часть 3

 1.8K  0

5 июл в 17:17

Потому что через бизнес-процессы лучше доходит

 3.3K  6

Ваш аккаунт	Разделы	Информация	Услуги
Профиль	Статьи	Устройство сайта	Корпоративный блог
Трекер	Новости	Для авторов	Медийная реклама
Диалоги	Хабы	Для компаний	Нативные проекты
Настройки	Компании	Документы	Образовательные
ППА	Авторы	Соглашение	программы
	Песочница	Конфиденциальность	Стартапам
			Спецпроекты



[Настройка языка](#)

[Техническая поддержка](#)

© 2006–2023, Habr