Системы искусственного интеллекта

«02.03.03 - Математическое обеспечение и администрирование информационных систем направленность разработка и администрирование информационных систем»

http//vikchas.ru

https://www.famous-scientists.ru/3653/

Тема 2.Введение в искусственный интеллект и машинное обучение

Лекция 7 «Внедрения искусственного интеллекта в бизнес»

Часовских Виктор Петрович
д-р техн. наук, профессор кафедры ШИиКМ

ФГБОУ ВО «Уральский государственный экономический университет»

Екатеринбург 2022

Преимущества линейных моделей

Линейные модели легко интерпретируемы: человеку легко понять, почему для объекта выполнено именно такое предсказание.

Линейные модели, как правило, решают задачу с приемлемым уровнем качества, однако уступают более мощным алгоритмам, ансамблям решающих деревьев и нейронным сетям.

С другой стороны, качество линейных моделей можно значительно повысить, придумав новые признаки, вычисляемые на основе исходных признаков (например, добавив квадраты признаков), — при этом свойство интерпретируемости сохраняется. Благодаря своей интерпретируемости линейные модели очень популярны в бизнес-задачах.

Линейные модели подразумевают, что предсказание вычисляется как сумма значений признаков объекта, умноженных на веса.

Веса настраиваются по данным в процессе обучения и определяют важность признаков.

Популярность линейных моделей обосновывается тем, что их предсказания легко интерпретировать, однако линейные модели не являются рекордсменами по точности предсказаний. Инструменты для решения задачи регрессии

Алгоритм — это просто последовательность действий, и чтобы использовать их для обучения по конкретному набору данных, алгоритмы нужно запрограммировать.

Разумеется, программировать алгоритм каждый раз заново не нужно, а нужно воспользоваться существующими инструментами, в которых методы запрограммированы качественно и эффективно.

Для задач классификации и регрессии чаще всего используют язык программирования **Python** и его библиотеку **Sklearn**, в ней реализовано большинство упомянутых выше алгоритмов. Для построения линейных моделей также часто используют библиотеку **Vowpal Wabbit**.

Sklearn - это библиотека машинного обучения для языка программирования Python, которая предоставляет множество возможностей, таких как многоступенчатый анализ, регрессия и алгоритмы кластеризации.

К возможностям **Vowpal Wabbit**, важным для машинного обучения, относятся непрерывное обучение (обучение в подключенном режиме), сокращение размерностей и интерактивное обучение.

Vowpal Wabbit также позволяет решать задачи, в которых данные модели не могут разместиться в памяти ЭВМ.

Основными пользователями Vowpal Wabbit являются специалисты по обработке и анализу данных, которые ранее использовали платформу для задач машинного обучения, таких как классификация, регрессия, моделирование разделов или факторизация матриц.

Переобучение

Обучающие данных используются для создания алгоритма предсказания.

По результатам обучения мы ожидаем, что алгоритм предсказания делает более-менее точные предсказания для объектов из обучающих данных — иначе зачем мы вообще обучали этот алгоритм. Однако может случиться, что алгоритм делает хорошие предсказания только для обучающих данных. Иными словами, алгоритм запомнил, зазубрил классы/числа для обучающих объектов, но не нашел никаких зависимостей между признаками и целевой переменной (классом/числом). Такой алгоритм будет плохо работать на шаге внедрения и называется переобученным.

Чтобы контролировать возникновение переобучения, на практике всегда выделяют два набора данных:

обучающие и тестовые.

Первый набор используется для обучения алгоритма, а второй — для контроля ошибки обученного алгоритма на новых данных, тех, которые не входили в обучение.

Что нужно предсказать в задаче регрессии?

Признаки по весам

Класс по признакам

Число по признакам

Веса по классам

В чем состоит обучение линейной модели?

Найти среднее целевой переменной

Найти подходящие веса

Найти порог измерений

Найти произведение признаков на веса

О метриках

Мы познакомились с задачами классификации и регрессии - двумя самыми распространенными задачами машинного обучения.

Для создания алгоритма регрессии или классификации обычно применяют линейные модели, ансамбли решающих деревьев или нейронные сети.

Когда алгоритм обучен, еще один важный шаг — понять, насколько хорошо или плохо он работает, иными словами, оценить качество выполняемых алгоритмом предсказаний.

Рассмотрим измерение качества в регрессии и классификации.

Стоит отметить важность правильного выбора метрики качества.

При обучении алгоритма специалист по машинному обучению будет ориентироваться на метрику, и если метрика выбрана неправильно, например, не отражает бизнес-требований, то и итоговый алгоритм может оказаться бесполезным.

Различают онлайн- и офлайн-метрики.

Онлайн-метрики измеряют реальный эффект от внедрения решения в продукт, например увеличение дохода или уровня удовлетворенности клиентов.

Конкретный выбор онлайн-метрик зависит от области: в маркетинге часто используется **Click-Though Rate (CTR),** например - требуется спрогнозировать конверсию рекламных объявлений, т.е. вычислить CTR-рейтинг (click through rate) или показатель кликабельности.

Эта важная метрика эффективности **интернет-маркетинга** определяет отношение числа кликов на рекламное объявление к числу показов и измеряется в процентах.

Если реклама была показана 10 раз и на нее кликнули 2 раза, то CTR равен 20 %, что считается весьма чрезвычайно высоким значением.

В онлайн-сервисах используется среднее время просмотра сайта пользователем.

В бизнесе применяется LTV (Lifetime Value) — это совокупная прибыль компании, получаемая от одного клиента за все время сотрудничества с ним. Увеличивается при уменьшении уровня оттока клиентов (Churn Rate). Каждая компания стремится увеличить LTV, удерживая клиента с помощью различных мер повышения лояльности (скидки, акции, подарки и пр.), т.к. привлечение нового пользователя обходится в 8-10 раз дороже.

Современные маркетинговые инструменты (CRM-системы, рекламные платформы) вместе с технологиями анализа данных на базе Big Data и Machine Learning помогают бизнесу увеличить LTV.

Онлайн-оценивание качества также часто подразумевает проведение A/B-теста, т.е. внедрение модели для части пользователей и сравнение показателей

Очевидные недостатки онлайн-метрик — сложность выполнения такой оценки и необходимость внедрения: если окажется, что алгоритм работает плохо, часть клиентов может пострадать от взаимодействия с ним.

Во время построения алгоритма используют **офлайн-метрики**: они не измеряют финансовые показатели, а измеряют лишь **точность предсказаний**, зато вычисляются быстро и не требуют внедрения.

Рассмотрим офлайн-метрики.

Задачи регрессии. Напомним, что в этой задаче необходимо научиться предсказывать числовые значения, например стоимость дома, возраст клиента или длину беспроцентного периода по кредитной карте.

Пример. Будем рассматривать предсказание спроса на товар: нужно предсказать, какое количество товара потребуется в точке продаж в течение следующей недели.

При оценке качества работают с таблицей, состоящей из двух столбцов: правильные значения и предсказанные.

Как было определено ранее, оценивать качество алгоритма стоит по объектам, которые алгоритм не видел во время обучения, то есть по тестовым данным.

Соответственно, в таблицу записываются предсказания для тестовых объектов и те значения, которые соответствуют этим объектам в данных.

Для простоты рассмотрим четыре тестовых объекта (четыре строки таблицы), на практике тестовые выборки состоят из тысяч или даже миллионов объектов

Номер товара	Значение из выборки (сколько единиц товара в реальности потребовалось, тысяч единиц)	Предсказанное значение (тысяч единиц)
1	20	18
2	15	19
3	14	12
4	15	25

Визуально можно проанализировать качество предсказаний: для первого и третьего товара выполнено довольно точное предсказание (разница в две тысячи единиц), для второго товара ошибка уже больше (четыре тысячи единиц), а для четвертого — совсем большая (10 тысяч единиц).

Однако в практических сценариях с тысячами выполнить такой визуальный анализ не получится - нужен агрегированный показатель качества.

Как считаются метрики регрессии

MAE (средняя абсолютная ошибка) — необходимо посчитать модуль разницы между прогнозом и реальным значением для всех объектов, а затем поделить разницу на число объектов.

•

MSE (среднеквадратическая ошибка) — необходимо посчитать разницу между прогнозом и реальным значением для каждого объекта, а затем возвести каждую в квадрат, сложить результаты и разделить на число объектов.

•

RMSE (корень из среднеквадратической ошибки) — необходимо посчитать разницу между прогнозом и реальным значением для каждого объекта, возвести каждую в квадрат, сложить результаты, поделить на число объектов, а затем взять корень из получившегося среднего значения.

•

MAPE (средняя процентная ошибка) — необходимо посчитать разницу между прогнозом и реальным значением, а затем поделить ее на реальное значение, получив среднее в виде %.

Чтобы получить агрегированный показатель качества, логично усреднить ошибки по объектам, то есть сложить все ошибки (отклонения предсказания от правильного ответа) и разделить на количество объектов: в нашем примере получится

$$\frac{2+4+2+10}{4} = 4{,}5$$

тысячи единиц. Иными словами, алгоритм в среднем ошибается на 4,5 единицы: на каких-то товарах больше, на каких-то — меньше.

Такая метрика называется средняя абсолютная ошибка (mean absolute error, MAE), а формулой ее можно записать так:

$$\mathrm{MAE}\left(Y^{true}, Y^{pred}\right) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^{true} - y_i^{pred}\right|$$

Где $\boldsymbol{Y^{true}}$ – правильные ответы для выборки,

$\boldsymbol{Y^{pred}}$ – предсказание для той же выборке

$\boldsymbol{y_i^{true}}$ - правильный ответ для i-го объекта

$\boldsymbol{y_i^{pred}}$ - предсказание для i-го объекта

всего объектов в выборке **n**

Часто используют похожую метрику, называемую средней квадратичной ошибкой (mean squared error, MSE): при ее вычислении усредняют квадраты ошибок:

$$\mathbf{MSE}\left(Y^{true}, Y^{pred}\right)=\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{true}-y_i^{pred}\right)^2$$

В нашем примере получится MSE = $\frac{4+16+4+100}{4}$ = 31

Но мы получили порядок не тысяч единиц товара, а «квадратов тысяч». Чтобы вернуться к исходным единицам, нужно взять квадратный корень — получится root mean square error. RMSE. В нашем примере RMSE = $\sqrt{\text{MSE}}$ = $\sqrt{31}$ = 5,56 единиц.

По сравнению с RMSE, метрика MAE более интуитивна, т.к. усредняются сами отклонения, но RMSE удобнее использовать при обучении алгоритмов. Хотя для MAE обучение тоже успешно выполняется.

Еще одна особенность метрики MAE — она более устойчива к выбросам, чем RMSE. Это означает, что если для одного объекта ошибка очень большая (объект-выброс), а для остальных объектов — маленькая, то значение MAE «подскочит» от этого одного объекта меньше, чем RMSE, т.к. в RMSE ошибки возводятся в квадрат.

В нашем примере объектом-выбросом является четвертый товар, и значение MAE на одну тысячу единиц меньше, чем значение RMSE.

Если с точки зрения бизнес-задачи важно, чтобы ошибки на всех объектах были примерно одинаковы (выбросы нежелательны), стоит использовать RMSE, если же наоборот, допустимо пару раз ошибиться, но достичь меньших ошибок на большинстве объектов. — стоит использовать MAE.

Иногда ошибка в меньшую или в большую сторону имеет разное влияние на бизнес - процесс. Эту особенность тоже можно учесть в метрике.

Например, если мы предскажем на одну тысячу единиц товара меньше, чем реально потребуется, то потеряем прибыль: некоторым клиентам не достанется товара.

А если мы предскажем на одну тысячу единиц больше товара, чем реально потребуется, то появятся дополнительные издержки на хранение товара.

Предположим, что товар занимает мало места и расходы на хранение невелики, тогда лучше ошибиться в большую сторону, чем в меньшую. В этом случае отрицательную и положительную разницу доумножают на разные коэффициенты, например 0,5 и 1,5.

В нашем примере коэффициент 1,5 будет применен к товарам 1 и 3 (18<20 12<14 к остальным двум товарам будет применен коэффициент 0,5), а значение метрики будет равно

$$\frac{1,5*2+0,5*4+1,5*2+0,5*10}{4} = 3,25$$

Из-за высокого коэффициента алгоритму будет выгоднее уменьшить ошибки для объектов 1 и 3, чем 2 и 4.
Эту метрику называют **квантильной ошибкой** (**Quantile loss**).

Интерпретация метрик

Само по себе значение метрик MSE или MAE можно сравнивать со средним значением целевой переменной: например, нам нужно предсказывать десятки, при этом допустимы ошибки порядка единиц. Если хочется получать значения ошибки в процентах («алгоритм в среднем ошибается на столько-то процентов»), можно

использовать метрики с нормировками.

К примеру, метрика **MAPE** (mean average percentage error) усредняет значения ошибок, деленных на значение целевой переменной

$$\text{MAPE}\left(Y^{true}, Y^{pred}\right) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^{true} - y_i^{pred}}{y_i^{true}}\right|$$

В нашем примере получится

$$\frac{\frac{2}{20}+\frac{4}{15}+\frac{2}{14}+\frac{10}{15}}{4} \sim 29\%$$

Иными словами, алгоритм в среднем ошибается на 29 процентов.

Важно понимать, что идеальных алгоритмов, как и нулевых значений метрик ошибок, в машинном обучении не бывает: такова суть этой области, что она помогает выполнять приблизительные предсказания.

Величину допустимой ошибки определяет заказчик (в терминах выбранной метрики). Например, заказчик может сказать, что допустимы значения MAPE не более 20% на тестовой выборке. Если после обучения алгоритма значение MAPE получилось больше, можно попробовать настроить алгоритм лучше или собрать больше данных.

Оценка ошибки **сверху**

Какие значения метрики являются допустимыми, зависит от бизнес-задачи.

Например - продать квартиру дороже или дешевле на 100 тысяч рублей может быть допустимо и многократно встречалось в практике фирмы, а на 300 тысяч рублей — уже нет.

Помимо этого, в качестве верхней оценки на допустимую ошибку можно использовать ошибку некоторого простого базового решения (**бейзлайна** - простая модель или эвристика, используемые как ориентир для оценки качества работы модели. Базовый уровень помогает разработчикам моделей определить минимальную ожидаемую производительность по конкретной задаче). Самый простой бейзлайн это константный алгоритм, он всегда предсказывает одно и то же число для всех объектов. Если вдруг случилось так. что у разработанного алгоритма ошибка больше, чем у константного бейзлайна. значит, в процессе обучения алгоритма была допущена оплошность.

Также стоит отметить, что никаких ограничений на выбор метрики, кроме бизнес - требований, нет: можно использовать популярные MAE или MAPE, можно комбинировать метрики (например, вводить коэффициенты для квадратичной ошибки, а не абсолютной, как в примере выше), а можно придумать свою метрику.

Если в задаче величина ошибки не важна, а важно только то. меньше ли она заданного порога, то можно ввести метрику «процент случаев, когда ошибка меньше порога». В нашей задаче с предсказанием спроса такая метрика подойдет, если товар хранится коробками {даже если осталась лишь одна единица, она занимает целую коробку), при этом мы можем хранить не больше одной коробки на один вид товара.

В нашем примере, если в коробку влезает три тысячи единиц товара, значение указанной придуманной метрики будет равно $\frac{2}{4} = 50\%$: в половине случаев ошибка меньше трех тысяч, в половине - больше.

Статьи для понимания основных понятий технологии блокчейн и СУБД ADABAS на сайте https://www.famous-scientists.ru/3653/

В ссылке *Публикации в изданиях Российской Академии Естествознания*: 37 / перечень публикаций

1. BLOCKCHAIN – ОСНОВНЫЕ ПОНЯТИЯ И РОЛЬ В ЦИФРОВОЙ ЭКОНОМИКЕ
2. Преимущества СУБД ADABAS в статье "*Обоснование выбора среды для проектирования и реализации системы оценки углерод депонирующей способности лесов России*"