Дополнительная практическая работа: «Множественная линейная регрессия»

Оглавление

Цель работы

Получить практические навыки использования линейной регрессии.

Задачи работы

1. Научиться аналитически решать задачу линейной регрессии.
2. Научиться решать задачу линейной регрессии с помощью библиотеки sklearn.

Перечень обеспечивающих средств

1. ПК.
2. Учебно-методическая литература.
3. Задания для самостоятельного выполнения.

Общие теоретические сведения

Линейная регрессия

Данные: пары значений $(x_i,\ y_i)$, где $i = 1, \dots, N$.

x называется предиктором или регрессором,

y называется зависимой переменной.

Задача: Найти такие значения *a* и *b*, чтобы функция $f(x) = ax + b$ как можно точнее аппроксимировала *y*, т.е. чтобы $f(x_i) \approx y_i$ для всех *i*.

Метрика производительности – среднеквадратичная ошибка:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$

где y_i - значение из данных, $f(x_i)$- результат работы модели.

Градиентный спуск

1. Случайным образом выбираем точку (a_0, b_0).
2. Вычисляем значения частных производных ошибки.
3. Изменяем координаты так, чтобы двигаться в сторону уменьшения производной:

   $a_i = a_{i-1} - \alpha \frac{\partial \mathrm{MSE}}{\partial a}(a_{i-1}),\ b_i = b_{i-1} - \beta \frac{\partial \mathrm{MSE}}{\partial b}(b_{i-1})$.
4. Если $\mathrm{MSE}(a_i, b_i) - \mathrm{MSE}(a_{i-1}, b_{i-1})$ достаточно мало, то завершаем. Иначе – возвращаемся к шагу 2.

Допущения линейной регрессии

Остатки: величины $y_i - f(x_i)$.

Допущения линейной регрессии

- Между *x* и *y* есть линейная зависимость.
- Остатки распределены нормальным образом.
- Среднее значение остатков равно нулю.
- Дисперсия остатков постоянна.

Ограничения линейной регрессии

- Низкая точность при аппроксимации нелинейных функций.
- Нельзя использовать для вычислений вне известного интервала.
- Считаем, что предикторы не содержат ошибок измерений.
- Нет ограничений области значений.

Регуляризация

Если данных мало, а модель сложная, то высока вероятность переобучения.

Регуляризация – добавление дополнительных слагаемых к метрике производительности для того, чтобы штрафовать модель за излишне сложные решения и, таким образом, препятствовать переобучению.

Смещение увеличивается, разброс уменьшается.

L1-регуляризация (lasso)

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda_1 \sum_i |a_i|$$

$$\sum_i -y_i \log f(z_i) - (1 - y_i)\log(1 - f(z_i)) + \lambda_1 \sum_i |a_i|$$

L1-регуляризация обнуляет параметры a_i, которые вносят в основном шум.

L2-регуляризация (ridge)

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda_2 \sum_i a_i^2$$

$$\sum_i -y_i \log f(z_i) - (1 - y_i)\log(1 - f(z_i)) + \lambda_2 \sum_i a_i^2$$

L2-регуляризация не даёт значениям параметров a_i бесконтрольно увеличиваться.

Elastic Net регуляризация

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 + \lambda_1 \sum_i |a_i| + \lambda_2 \sum_i a_i^2$$

$$\sum_i -y_i \log f(z_i) - (1 - y_i)\log(1 - f(z_i)) + \lambda_1 \sum_i |a_i| + \lambda_2 \sum_i a_i^2$$

Множественная линейная регрессия

Данные: $(x_{1i}, x_{2i}, \dots, x_{Ki}, y_i)$, где $i = 1, \dots, N$.

Задача: Найти такие значения a_k, где *k* = 1, …, *K*, чтобы функция

$$f(x_1, x_2, \dots, x_K) = a_1 x_1 + a_2 x_2 + \cdots + a_K x_K + b$$

как можно точнее аппроксимировала *y*,

т.е. чтобы $f(x_1, x_2, \dots, x_K) \approx y_i$ для всех *i*.

$b = a_0 x_0$, где $x_0 = 1$.

$$f(x_1, x_2, \dots, x_K) = a_0 x_0 + a_1 x_1 + a_2 x_2 + \cdots + a_K x_K = \sum_{k=0}^{K} a_k x_k$$

Задание

Пояснение

Для сохранения результатов данной работы вам понадобится один файл. Назовите его «*Фамилия* – дополнительное задание.ipynb».

Часть 1

- Обновите свой репозиторий, созданный в практической работе №1, из оригинального репозитория: https://github.com/mosalov/Notebook_For_AI_Main.

Часть 2

- Откройте свой репозиторий в Binder (https://mybinder.org/).
- Откройте файл «task5.ipynb».
- Изучите, при необходимости – выполните повторно, приведённый в файле код.
- С помощью библиотек sklearn по аналогии с имеющимся кодом решите задачу линейной регрессии, а также примените L1, L2 и ElasticNet регуляризации для случая множественной регрессии: зависимая переменная – Weight, регрессоры: Length1, Length2, Length3, Height, Width.
- Сохраните код в ipynb-файле. Необходимые пояснения опишите в своём docx/doc-файле.

Замечания

1. X_train будет двумерным массивом – это нормально.
2. Сразу нормируйте значения регрессоров и зависимой переменной.
3. При нормировке вы можете передавать в MinMaxScaler сразу весь необходимый массив.
4. Т.к. x_train – двумерный массив, к нему не нужно применять reshape. К одномерному y_train – нужно.
5. Модели линейной регрессии будут возвращать массив в качестве значения coef_.
6. Вы не сможете без дополнительных ухищрений нарисовать графики, поэтому можете этого не делать.

Требования к отчету

Загрузите свой файл в репозиторий, созданный в практическом задании №1 по пути: «Notebook_For_AI_Main/2020 Осенний семестр/Дополнительное практическое задание/» и сделайте пул-реквест.

Литература

- https://habr.com/ru/post/514818/
- https://habr.com/ru/post/474602/
- http://statistica.ru/theory/osnovy-lineynoy-regressii/
- http://statistica.ru/theory/logisticheskaya-regressiya/
- https://habr.com/ru/post/485872/
- https://habr.com/ru/company/ods/blog/323890/#metod-maksimalnogo-pravdopodobiya
- https://dyakonov.org/2018/03/12/%D0%BB%D0%BE%D0%B3%D0%B8%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F-%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B8/