ФиПЛ-2022

Линейная регрессия

Метрики. Функционал ошибки. Градиентный спуск

Линейная регрессия: веса

- Цель: подобрать такие коэффициенты уравнения прямой, чтобы по нашим признакам можно было угадать примерный ответ (целевую переменную):

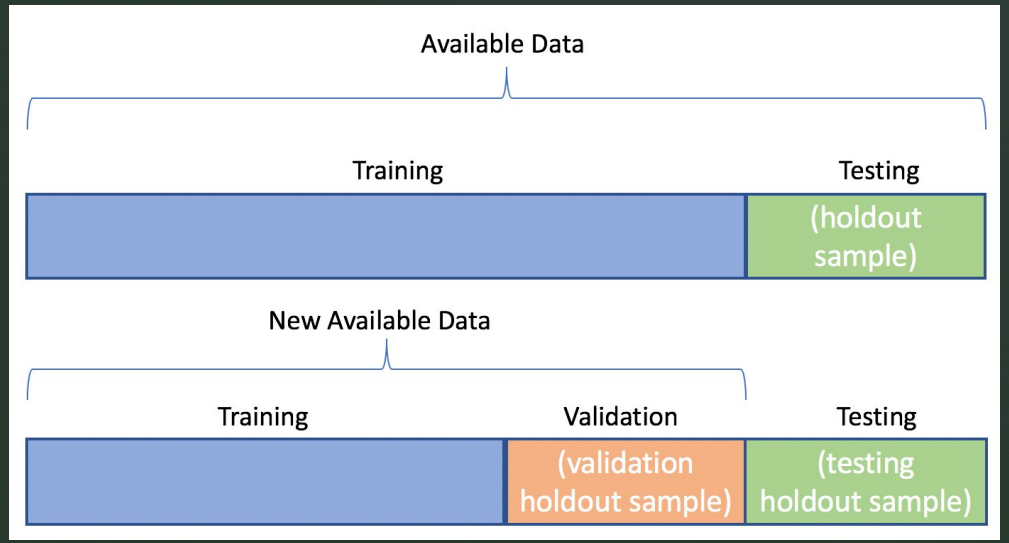
$$w_1 x_1 + w_2 x_2 + \ ... + w_0 = y$$

- $x_1, x_2, ...$ - это наши признаки (площадь квартиры, время до метро...)
- y – это целевая переменная (цена квартиры)
- $w_1, w_2, ...$ - это веса, или коэффициенты
- w_0 - это свободный коэффициент (шум)

Как оценить качество обученного алгоритма?

- Надо отложить кусочек всех данных, чтобы алгоритм их не видел: а потом на них можно тестировать
- По какой формуле оценивать качество? Например, среднеквадратичная ошибка:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

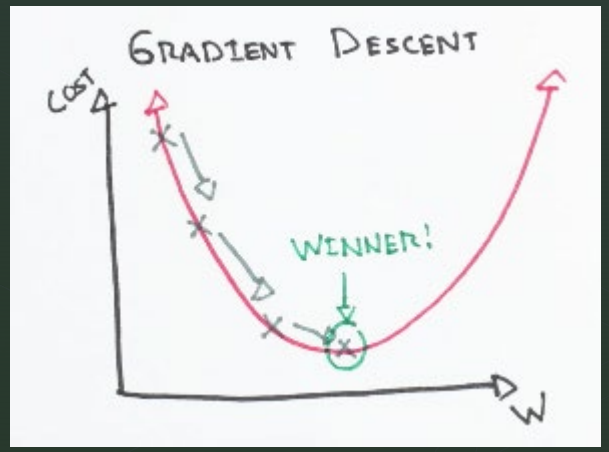


Как подобрать веса?

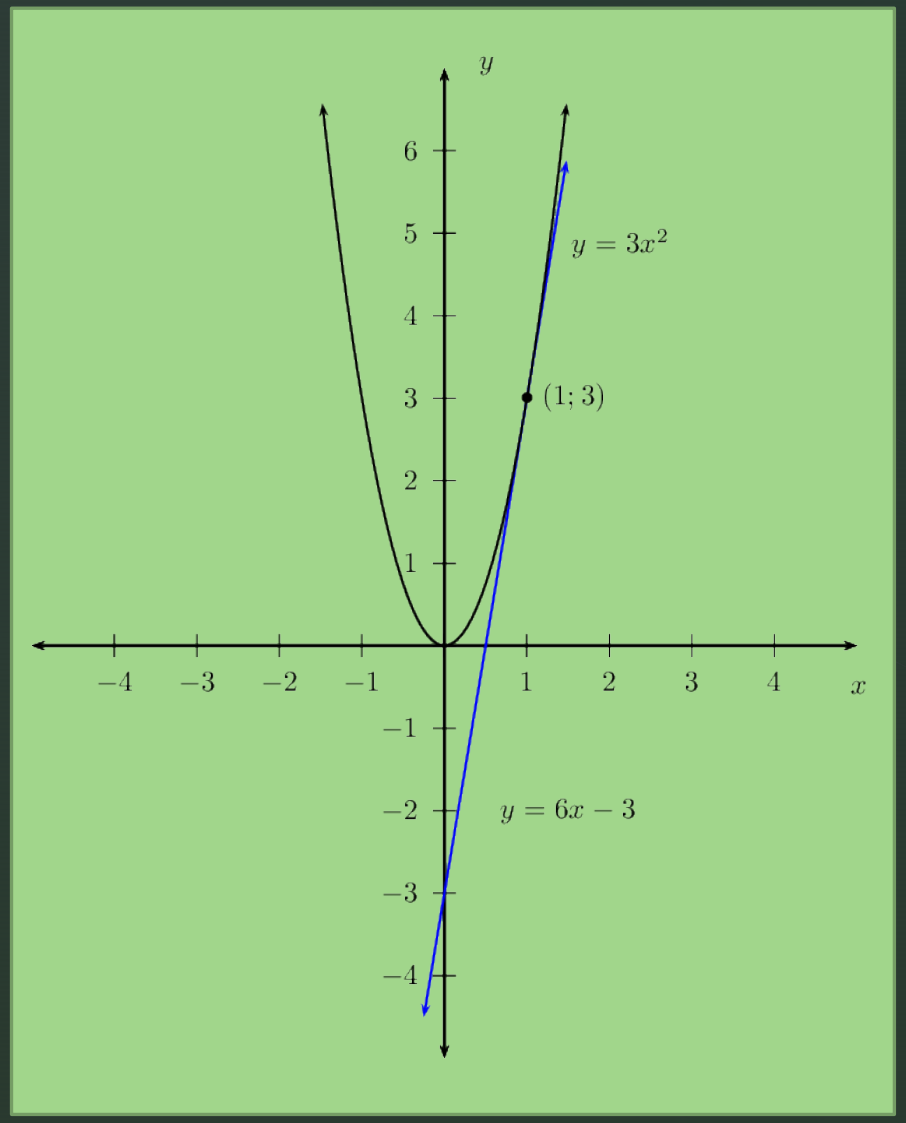
- Инициализируем их случайным образом
- Считаем, насколько ошибся наш алгоритм с такими весами на объектах из обучающей выборки
- Корректируем веса...
- НО КАК ИХ КОРРЕКТИРОВАТЬ?

Великий, ужасный – градиентный спуск

- Основная идея: нам нужно найти минимум функции ошибки ($\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y_i})^2$), то есть, такие коэффициенты, при которых ошибка самая маленькая
- Есть такой специальный математический алгоритм, который задействует производные...
- Градиент – касательная к нашей функции, которая смотрит в ту сторону, куда функция растет



Градиентный спуск



- Градиент – это на самом деле (частная) производная
- Magick: если будем вычислять значение градиента и вычитать его из наших весов, то веса будут становиться лучше!
- Слишком много про это лучше не думать, а то недолго и моск сломать...

Виды градиентного спуска

- Обычный градиентный спуск: считаем градиенты для каждого веса на всех примерах и потом вычисляем усредненные градиенты, которые вычитаем из их весов

  $w_{i+1} = w_i - \eta \nabla f(w_i)$ (η – learning rate, скорость спуска)

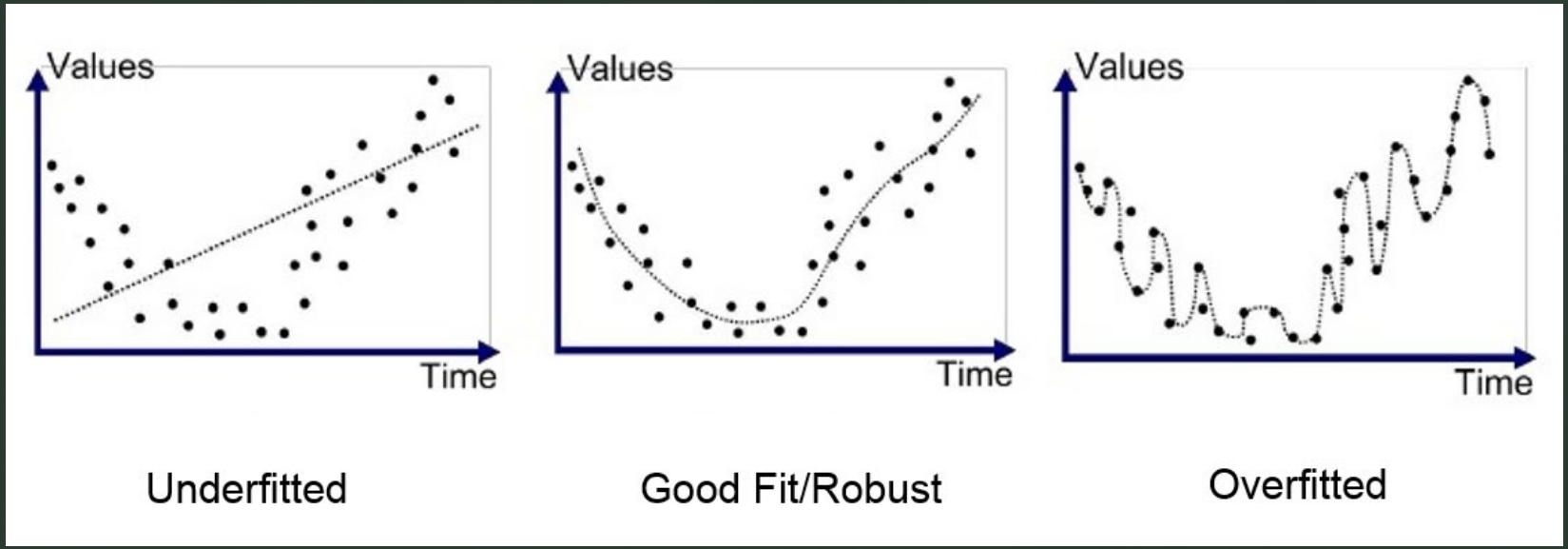
- Но это же для каждого шага обучения вычислять градиенты для всех весов и для всех примеров в выборке! Затратно
- А что, если считать не для всех примеров сразу?..

Виды градиентного спуска

- Стохастический (SGD)
- Mini-batch (MBGD)
- Nesterov Momentum
- Adam
- Adadelta
- Adagrad
- Тысячи их!

Переобучение и недообучение

- Переобучение: алгоритм тупо заучил существующие в тренировочной выборке объекты (слишком долго гоняли)
- Недообучение: недогоняли



Категориальные признаки и OHE

- Хотим угадывать цену квартиры по району, в котором она находится: Мытищи – фу, Красная Пресня – найс
- Но они не числовые?.. Просто заменить их на 1-2-3 нечестно
- Ответ: использовать One Hot Encoding (OHE)

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Наконец – практическая часть!

устанавливаем библиотеку scikit-learn:

pip install scikit-learn