Введение в машинное обучение

Линейная регрессия

Задача

Восстановить (сложную) зависимость по конечному числу примеров

Пример задачи

Предсказать цену на квартиру по ее характеристикам

- площадь
- расположение (район, расстояние до метро)
- ремонт, наличие мебели
- и т.д.

Обозначения

- x - объект, *sample* - для чего хотим делать предсказания (**квартира**)
- $\mathbb{X}$ - пространство всех возможных объектов (**все квартиры на свете**)
- y - ответ, целевая переменная, *target* - что предсказываем (**цена**)
- $\mathbb{Y}$ - пространство ответов - все возможные значения ответа ($\in \mathbb{R}, >= 0$)

- $X = \{(x_1, y_1), \ldots, (x_\ell, y_\ell)\}$ - обучающая выборка: **все пары (квартира - ее цена)**
- ℓ - размер обучающей выборки

Представление объектов

- Как сделать объекты машиночитаемыми?
- Представить в виде набора признаков - числовых характеристик.
- Каждый объект - вектор чисел $x = (x^1, \ldots, x^d)$
- d - количество признаков
- Обучающая выборка - матрица размера ℓ x d
- Каждая строка - объект, каждый столбец - признак.

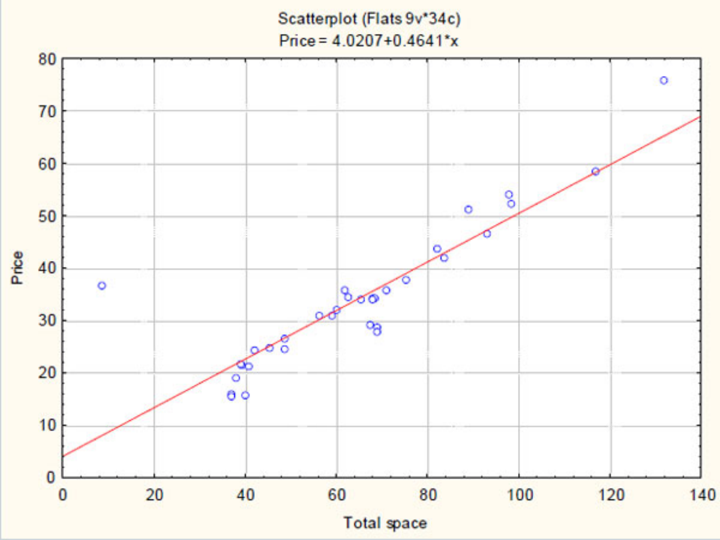
Линейная регрессия

- Функция, которая отображает $\mathbb{X}$ в $\mathbb{Y}$
- Хотим засунуть в нее вектор признаков объекта x и получить на выходе соотвествующий ему ответ y
- Сводится к суммированию значений признаков x с некоторыми коэффициентами (весами) w + свободный коэффициент w_0:

$$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j. \tag{1}$$

Парная ререссия

Частный случай линейной регрессии с одним признаком:



Линейная регрессия

Площадь	Расстояние до метро	Цена квартиры
100	200	10600
35	500	3500
50	100	5800

$$Price = 100S - 2Distance + 1000 \tag{2}$$

Обучение

- Обучиться = подобрать значения весов так, чтобы модель давала ответы как можно ближе к реальности.
- ***Минимизация эмпирического риска*** - добиться минимально возможной ошибки на обучающей выборке.
- Как измерить ошибку?

Функция потерь

- обозначается Q
- Чаще всего используется ***Mean Squared Error***:

$$MSE(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2 \,. \tag{3}$$

- Среднее арифметическое квадрата разницы предсказанного ответа и реального.

Функция потерь

- Получаем задачу оптимизации (считаем, что среди признаков есть константный, и поэтому свободный коэффициент не нужен):

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\left(\langle w, x_i\rangle - y_i\right)^2 \to \min_w \tag{4}$$

Методы оптимизации

- Метод Наименьших Квадратов (применим для линейной регрессии и квадратичной функции потерь)
- Градиентный спуск (общий подход)

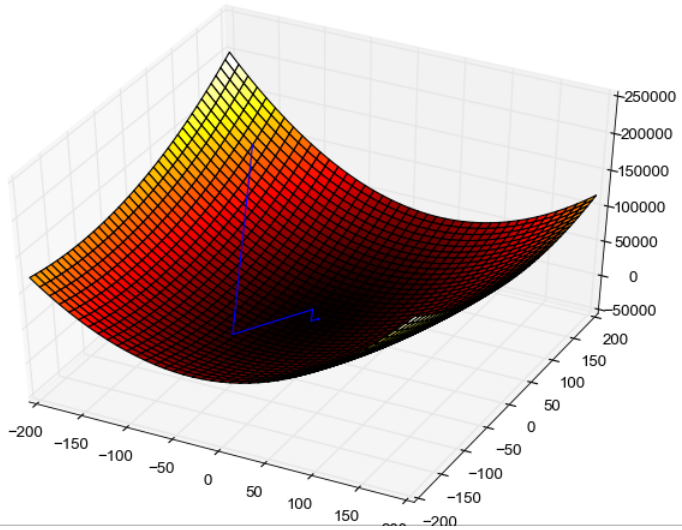
Градиент

- Градиентом функции называется вектор ее частных производных:

$$\nabla f(x_1, \dots, x_d) = \left(\frac{\partial f}{\partial x_j} \right)_{j=1}^{d} \tag{5}$$

- Частная производная - производная многомерной функции по одной из переменных.
- **Градиент является направлением наискорейшего роста функции, а антиградиент (т.е. $-\nabla f$) — направлением наискорейшего убывания.**
- Причем норма (длина) градиента равна величине скорости возрастания/убывания.

Градиент



250000
200000
150000
100000
50000
0
−50000
−200
−150
−100
−50
0
50
100
150
200
150
100
50
0
−50
−100
−150
−200

Градиентный спуск

- $W^{(0)} \leftarrow N(0,1)$ инициализировать начальное приближение для вектора весов случайными небольшими значениями.
- for i in range(max_iter):
  - $\nabla_w Q = [\frac{\partial Q}{\partial w_0}, \frac{\partial Q}{\partial w_1}, ..., \frac{\partial Q}{\partial w_d}]$ - посчитать градиент в точке $W = W^{(t)}$
  - $W^{(t+1)} = W^{(t)} - \alpha * \nabla_w Q$ - обновить вектор весов, сделав шаг длины α в сторону антиградиента
  - Если $||\nabla_w Q||_2 < \epsilon$ - если норма градиента стала меньше некоторой заданной малой величины (значние функции потерь больше (почти) не уменьшается), то мы считаем, достигли минимума и прерываем цикл
- α, ϵ, max_iter - гиперпарметры, значение которых задается заранее