KARPOV.COURSES >>>
КОНСПЕКТ

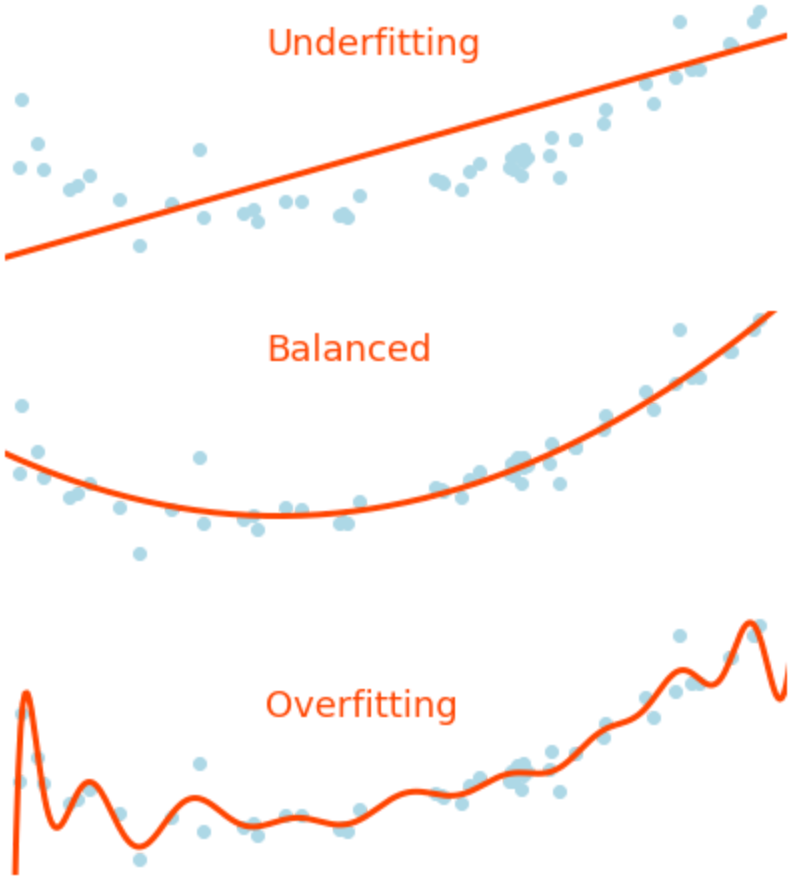
Конспект > 6 урок > Мультиколлинеарность, регуляризация и масштабирование признаков

>Оглавление

>Проблема переобучения в МО

Основная проблема полиномиальной модели - это переобучение.

Переобучение(пере- в значении «слишком», англ.overfitting) — явление, когда построенная модель хорошо объясняет примеры из обучающей выборки, но относительно плохо работает на примерах, не участвовавших в обучении (на примерах из тестовой выборки).



Оценить степень переобучения можно посмотрев на коэффициенты β (coef) Характерной чертой переобученной модели являются веса, значительно большие, чем ожидалось в действительности (трехзначные и более числа).

Можно представить модель, которая на основе двух признаков - площади и расположения рассчитывает стоимость квартиры:

цена = 3 000 000 * м^2 - 5 000 000 *расстояние до метро$\left(\text{м}^2\right)$

Согласно этой модели при уменьшении расстояния до метро на 1 метр, цена возрастет на пять миллионов, что кажется не совсем адекватным.

Отсюда возникает вопрос: можем ли мы контролируя размер весов модели препятствовать ее переобучению? Ответ: да, можем!

Давайте будем штрафовать модель за каждую единицу в норме весов.

Для этого добавим еще одну составляющую в уже известный нам функционал качества:

$Q(a(x,\beta),X) + \lambda \cdot R(\beta)$ —> min_β

, где λ - некая константа, а R - функция регуляризации для β, например это может быть сумма квадратов β_1 и β_2

Вспомним как мы минимизировали функционал качества для нашей переобученной модели по предсказанию цены квартиры:

$Q^*_{train} = Q_{train}(\beta_1^; \beta_2^) = Q_{train}(3,000,000; -5,000,000) = X$

Теперь воспользуемся новой формулой минимизации и сравним результат:

$Q_{regul} = Q_{train}(\beta_1; \beta_2) + \lambda \cdot R(\beta) = Q_{train}(\beta_1; \beta_2) + 10 \cdot (\beta_1^2 + \beta_2^2)$

$(\beta_1; \beta_2) = (20,000; -10,000)$

Как видим, теперь значение коэффициентов стало на порядок ниже

При этом качество новой модели на тренировочной выборке будет давать худшие показатели, поэтому стоит быть к этому готовыми.

Цель подобрать такое значение \lambda, при котором модель покажет оптимальные значения на трейне и тесте.

Статья о переобучении

>Регуляризация

Теперь посмотрим как можно настраивать регуляризатор (R(\beta))

Выделяют два типа:

- **L-2(Ridge)** - когда мы в качестве регуляризатора берем сумму квадратов всех параметров модели

$R(\beta) = \sum(\beta_i^2) = \beta_1^2 + \beta_2^2 ... \beta_n^2$

- **L-1(Lasso)** - когда мы в качестве регуляризатора берем сумму модулей всех параметров модели

$$R(\beta) = \sum(|\beta_i|) = |\beta_1| + |\beta_2| + ... + |\beta_n|$$

Давайте посмотрим на роль регуляризатора в градиентном спуске.

Появление дополнительного слагаемого$\lambda \cdot \Delta R(\beta)$ в функции для градиента приведет к его условному “растягиванию”, а значит величина шага будет также увеличиваться

$$\Delta Q_{L_1} = \Delta Q + \lambda \cdot (sgn(\beta_1)sgn(\beta_2))...sgn(\beta_n)$$
$$\Delta Q_{L_2} = \Delta Q + \lambda \cdot (2\beta_1 2\beta_2 ... 2\beta_n)$$

Статья о регуляризации и ее перевод

>Масштабирование

В предыдущей части могло сложиться впечатление, что трехзначные и более веса у модели - однознячный признако переобучения, от которого нужно избавляться. Однако это не всегда так. Вернемся к примеру, где мы предсказываем цену квартиры, выраженную в рублях(шести-семизначное число) на основе метража(двузначное число) и расстояния до метро(сотни метров):

цена =$\beta_1 \cdot \text{м}^2 - \beta_2 \cdot$ расстояние до метро

Естественно ,чтобы решить подобное уравнение коэффициент β_1 должен быть порядка ста тысяч, а β_2 - порядка тысячи.

Бывает и такая ситуация, что только один из коэффициентов оказывается неадекватно большим и тогда при добавлении регуляризатора пострадают остальные коэффициенты, так как они также будут штрафоваться.

Избежать такой ситуации может помочь приведение признаков к примерно одному порядку. То есть если один признак находится в пределах десятка метров, а второй сотен метров, то лучше выразить последний через дециметры.

Но как приводить признаки к единому порядку, не погружаясь для этого данные и не прикидывая значения вручную?

Для этого в машинном обучении существуют специальные методы.

StandardScaler

$d_j = \frac{d_j - \mu}{\sigma}$

где, d_j - некий признак объекта, μ - среднее по выборке:

$\mu = \frac{1}{n} \sum d_j$

,а σ - среднеквадратичное отклонение от ‘этого среднего:

$\sigma^2 = \frac{1}{n} \sum (d_j - \mu)^2$

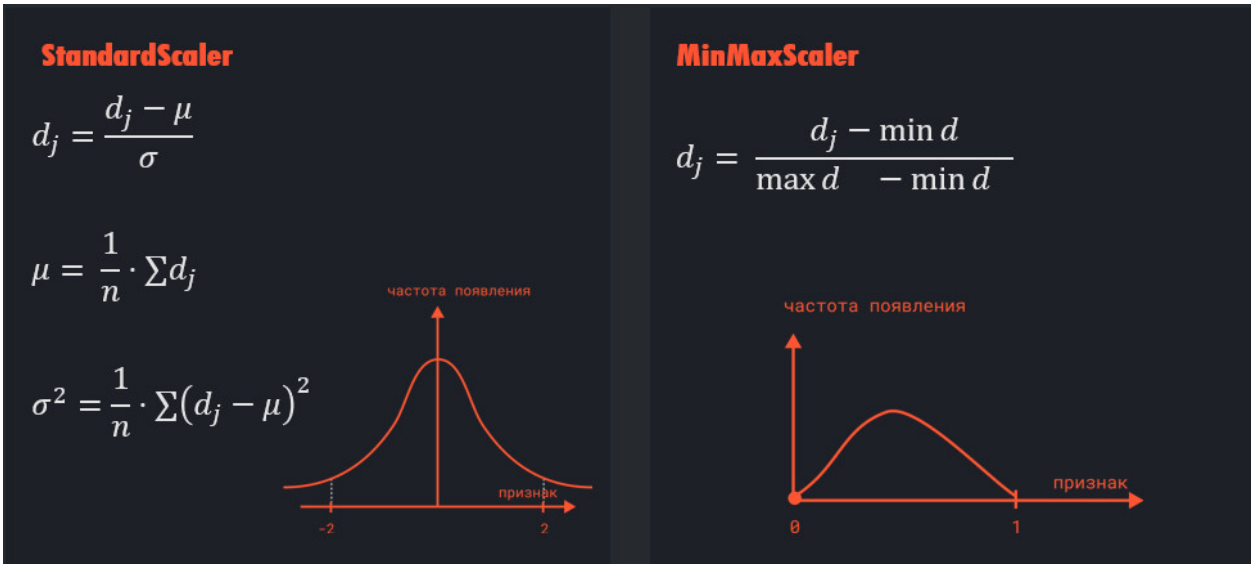
Таким образом мы получим стандартное нормальное распределение значений признаков в интервале $(-2, 2)$

MinMaxScaler

суть преобразования сводится к тому, что из каждого значения вычитается минимальное делится на разницу максимального и минимального

$d_j = \frac{d_j - dmin}{d_{max} - d_{min}}$

В итоге также получаем распределение признаков, приближенное к равномерному, лежащее в интервале $[0, 1]$.

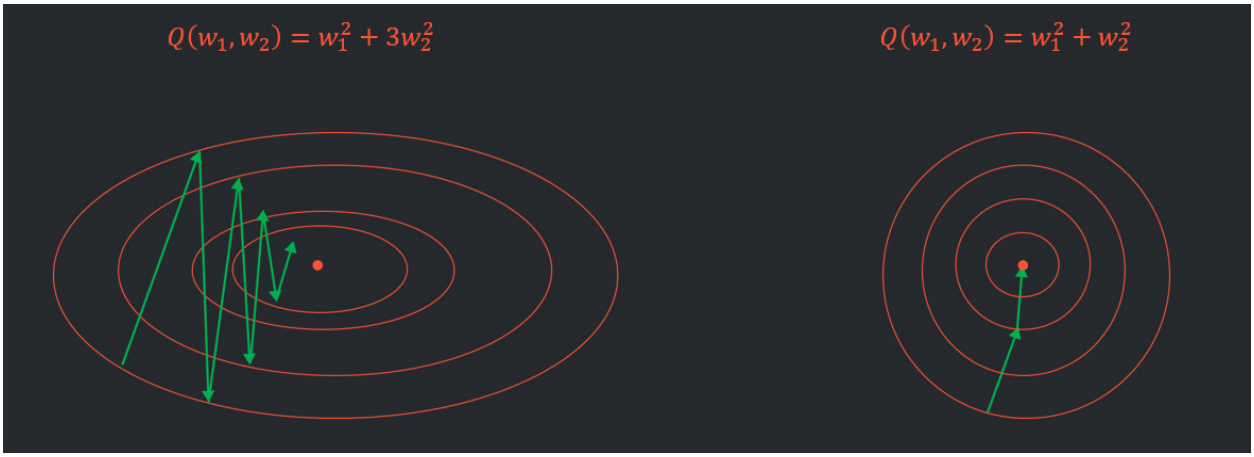


Подробнее о методах масштабирования

Масштабирование коэффициентов и градиентный спуск

А что дает масштабирование применительно к другим алгоритмам машинного обучения? Если мы возьмем модель градиентного спуска и промасштабируем признаки, то получим сокращение времени сходимости градиента.

Когда признаки сильно различаются, линии уровня имеют форму, приближенную к эллипсу. При масштабировании картина уже будет скорее напоминать окружности, а перпендикуляры касательных к окружности всегда будет направлены в центр, то есть в искомую точку минимума.



Еще один приятный бонус масштабирования - когда признаки однородны, мы можем сразу оценить важность того или иного признака по величине его коэффициента.
Чем больше коэффициент - тем важнее влияние конкретного признака на таргет.

Общий алгоритм регуляризации:

1. Масштабирование признаков(приведение к одному порядку)
2. Регуляризация модели(Ridge/Lasso)
3. Выбор подходящего параметра λ
4. Сравнение коэффициентов между собой и выводы на основе этого

>Ликбез №1: Условный экстремум

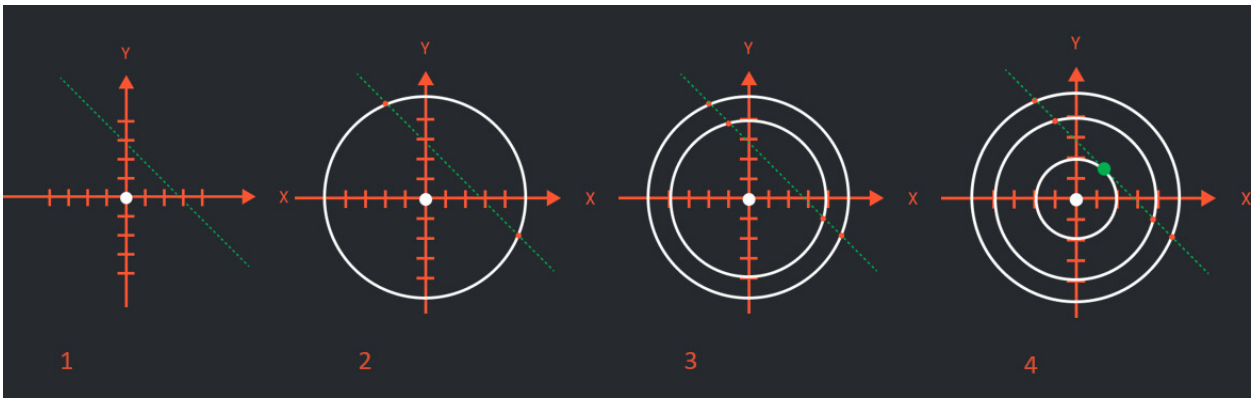
Предположим у нас есть функция, которую хотим минимизировать:

$z(x, y) = x^2 + y^2$—> min

Добавим некоторое ограничение, например

$$x + y - 2 \cdot \sqrt{2} = 0$$

Зафиксируем некоторое случайное значение функции, допустим $C = 25$ и посмотрим на линию уровня(рис.1)



Как можно видеть на графике, линия уровня пересекается с линией ограничения в двух точках(рис.2)

Будем уменьшать C до тех пор, пока не получим единственную точку касания линии уровня и линии ограничения $C = 4$ - это и будет искомая точка условного экстремума(рис.4)

Таким образом, безусловный экстремум для функции

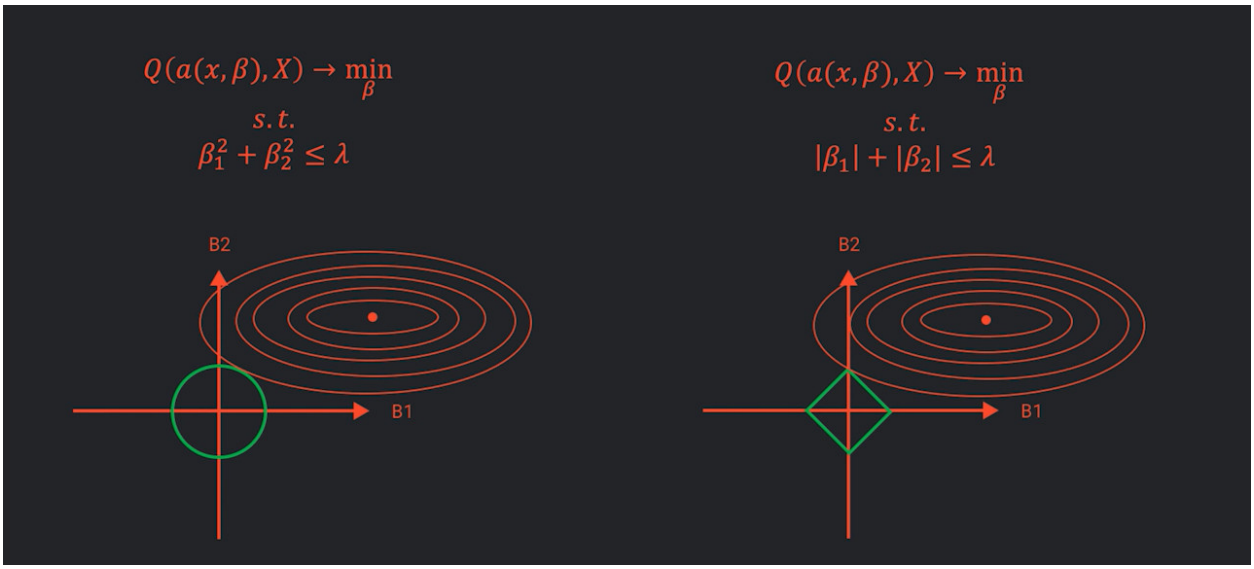
$z(0; 0) = 0$

$z(\sqrt{2}; \sqrt{2}) = 4$

Теперь применим это к задаче регуляризации модели машинного обучения. Как вы уже могли догадаться, ее решение будет сводиться к нахождению условного экстремума :

$Q(a(x,\beta),X) —> min_\beta$

$R(\beta) \leq \lambda$



Условный экстремум для функций регуляризации Ridge (слева) и Lasso (справа)

>Мультиколлинеарность

Еще одна проблема, с которой можно столкнуться при построении линейных моделей.

Мультиколлинеарность - наличие линейной завивимости между признаками, т.е. когда можно выразить один признак через другой. Это приводит к целому комплексу проблем:

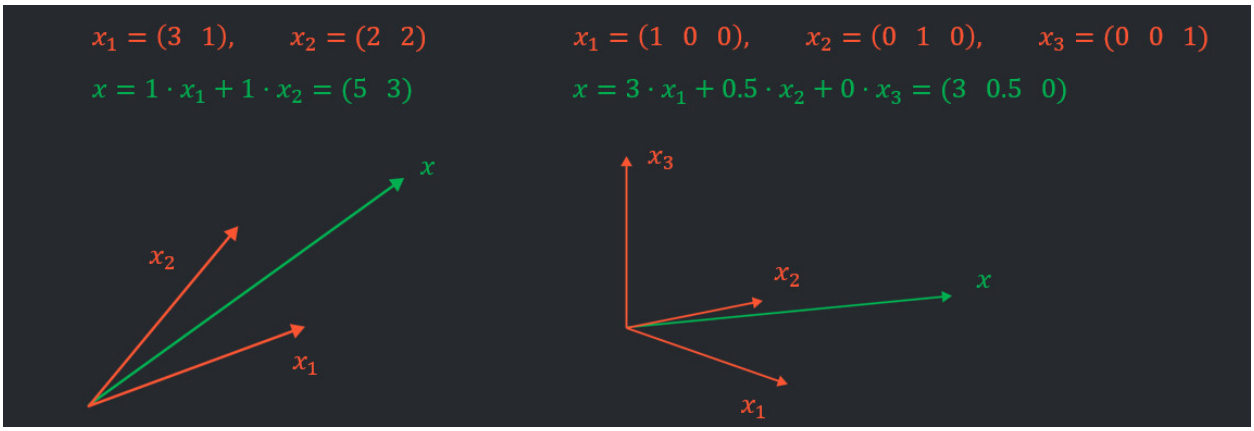
- Неустойчивости модели и очень часто к переобучению модели.
- В случает OLS регрессии - отсутствие одного минимума.

- Невозможно воспользоваться формулой $\beta^* = (X^T \cdot X)^{-1} X^T \cdot Y$ (невозможно вычислить обратную матрицу).

Линейная зависимость

Рассмотрим линейную зависимость на примере векторов.

Система векторов является линейно-зависимой, если хотя бы один из векторов можно выразить как линейную комбинацию остальных.



Примеры линейной зависимости векторов

Так как мы работаем с признаками как с векторами, когда среди них присутствуют такие, которые можно выразить через другие признаки, появляется проблема мультиколлинеарности.

Мультиколлинеарность можно также обозначить как избыточность признаков, то есть убрав один из таких признаков мы не лишимся какой-либо важной информации.

По этой же причине мы применяли One Code Encoding, избавляясь от лишней колонки с валютой, которую мы можем легко восстановить, зная остальные.

Другой способ - использовать объединение коррелирующих признаков в один, например, посчитав какой-либо индекс по ним.

Статья о мультиколлинеарности

Теперь мы вооружены необходимым пониманием для борьбы с переобучением, остается научиться применять эти знания на практике!