Линейные модели

- Линейные модели сводятся к суммированию значений признаков с некоторыми весами:

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

- Параметрами модели являются веса или коэффициенты w_j .
- Вес w_0 также называется свободным коэффициентом или сдвигом (bias).
- Линейную модель можно представить в более компактном виде (через скалярное произведение векторов):

$$a(x) = w_0 + \langle w, x \rangle$$

- Часто запись упрощают введением дополнительного признака, всегда равного единице:

$$a(x) = \langle w, x \rangle$$

Линейные модели

- За счёт простой формы линейные модели достаточно быстро и легко обучаются, и поэтому популярны при работе с большими объёмами данных.
- Также у них мало параметров, благодаря чему удаётся контролировать риск переобучения и использовать их для работы с зашумлёнными данными и с небольшими выборками

Измерение ошибки в задачах регрессии

- Чтобы обучать регрессионные модели, нужно определить, каким образом измеряется качество предсказаний.
- Как правило, используются разные формы $L(y, a)$ оценки отклонений прогноза *a* от истинного ответа *y*

- Среднеквадратичное отклонение (mean-squared error, MSE):

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

- Корень среднеквадратичного отклонения (rooted mean-squared error, RMSE):

$$MSE(a, X) = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(a(x_i) - y_i)^2}$$

Коэффициент R2

- Среднеквадратичная ошибка подходит для сравнения двух моделей или для контроля качества во время обучения, но не позволяет сделать выводы том, насколько хорошо данная модель решает задачу.
- Коэффициент детерминации (нормированная среднеквадратичная ошибка) :

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^{l}(a(x_i) - y_i)^2}{\sum_{i=1}^{l}(y_i - \overline{y})^2}$$

$\overline{y} = \frac{1}{l}\sum_{i=1}^{l} y_i$ среднее значение целевой переменной

MAE

- Среднее абсолютное отклонение (mean absolute error, MAE):

$$MAE(a, X) = \frac{1}{l} \sum_{i=1}^{l} \left| a(x_i) - y_i \right|$$

- Модуль отклонения не является дифференцируемым, но при этом менее чувствителен к выбросам. Квадрат отклонения, по сути, делает особый акцент на объектах с сильной ошибкой, и метод обучения будет в первую очередь стараться уменьшить отклонения на таких объектах.

Обучение линейной регрессии

- Чаще всего линейная регрессия обучается с использованием среднеквадратичной ошибки. В этом случае получаем задачу оптимизации:

$$\frac{1}{l}\sum_{i=1}^{l}(\langle w, x_i \rangle - y_i)^2 \to \min_w$$

- Задачу можно переписать в матричном виде:

$$\frac{1}{l}\|Xw - y\|^2 \to \min_w$$

- Если продифференцировать данный функционал по вектору w, приравнять к нулю и решить уравнение, то получим явную формулу для решения:

$$w = (X^T X)^{-1} X^T y$$

Градиент

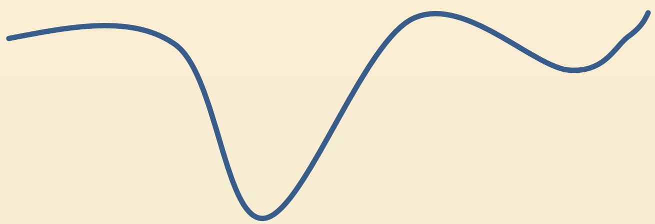
- Оптимизационные задачи можно решать итерационно с помощью градиентных методов.
- Градиентом функции $f : R_d \to R$ называется вектор его частных производных:

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

- Градиент является направлением наискорейшего роста функции, а антиградиент (т.е. $-\nabla f$) направлением наискорейшего убывания. Это ключевое свойство градиента, обосновывающее его использование в методах оптимизации.
- Задача оптимизации решается итеративно: из некоторой точки необходимо сдвинуться в сторону антиградиента, пересчитать антиградиент и снова сдвинуться в его сторону и т.д.

Градиентный спуск

- Градиентный спуск состоит в повторении следующих шагов до сходимости:



$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q(w^{(k-1)})$$

$Q(w)$ - значение функционала ошибки для набора параметров w

η_k - длина шага, которая нужна для контроля скорости движения.

- Длина шага может быть постоянной. Если длина шага слишком большая, то есть риск постоянно перепрыгивать через точку минимума, а если шаг слишком маленький, то движение к минимуму может занять слишком много итераций. Часто длину шага монотонно уменьшают по мере движения
- Останавливать итерационный процесс можно, например, при близости градиента к нулю или при слишком малом изменении вектора весов на последней итерации.

Оценивание градиента

- Как правило, в задачах машинного обучения функционал Q(w) представим в виде суммы ℓ функций:

$$Q(w) = \sum_{i=1}^{l} q_i(w)$$

- Проблема метода градиентного спуска состоит в том, что на каждом шаге необходимо вычислять градиент всей суммы («полный градиент»):

$$\nabla_w Q(w) = \sum_{i=1}^{l} \nabla_w q_i(w)$$

- Это может быть очень трудоёмко при больших размерах выборки.

Оценивание градиента: SGD

- Оценить градиент суммы функций можно градиентом одного случайно взятого слагаемого. В этом случае мы получим метод стохастического градиентного спуска (stochastic gradient descent, SGD):

$$w^{(k)} = w^{(k-1)} - \eta_k \nabla q_{i_k}(w^{(k-1)})$$

i_k - случайно выбранный номер слагаемого из функционала.

- Таким образом, метод стохастического градиента имеет менее трудоемкие итерации по сравнению с полным градиентом, но и скорость сходимости у него существенно меньше.
- Для выполнения одного шага в данном методе требуется вычислить градиент лишь одного слагаемого
- На каждом шаге необходимо держать в памяти всего один объект из выборки.

Регуляризация в LR

- Построение полиномиальной регрессии, например, в виде:

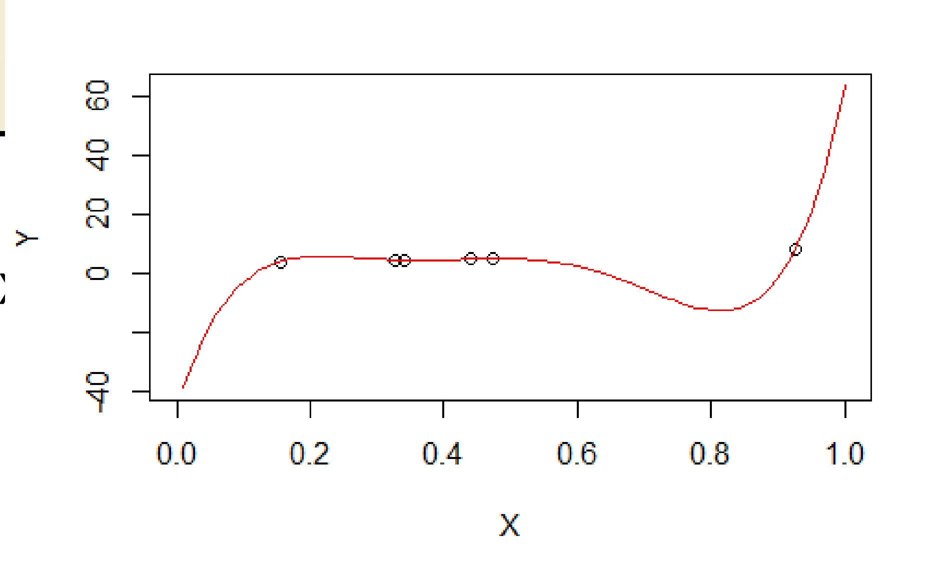
$$a(x) = w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_1^3 + w_4 x_1^4$$

может приводить к получению достаточно больших коэффициентов w_i

Такие модели являются «переобученными» (overfitted)

```
Call:
lm(formula = Y ~ X + I(X^2) + I(X^3) + I(X^4) + I(

Coefficients:
(Intercept)          X       I(X^2)       I(X^3)
    -16.22       535.03     -4696.89     13889.26
    I(X^4)       I(X^5)
 -16466.27      6811.28
```



Регуляризация

- Решение проблемы «больших коэффициентов» обеспечивает «регуляризация» введение в функцию потерь дополнительного слагаемого – «штраф» *R(w)* :

$$Q_{\alpha}(w) = \frac{1}{l}\sum_{i=1}^{l}(a(x_i) - y_i)^2 + \alpha R(w)$$

- Распространены две формы регуляризации:

L1-регуляризация (Lasso): $$R(w) = \|w\|_1 = \sum_{i=1}^{l}|w_i|$$

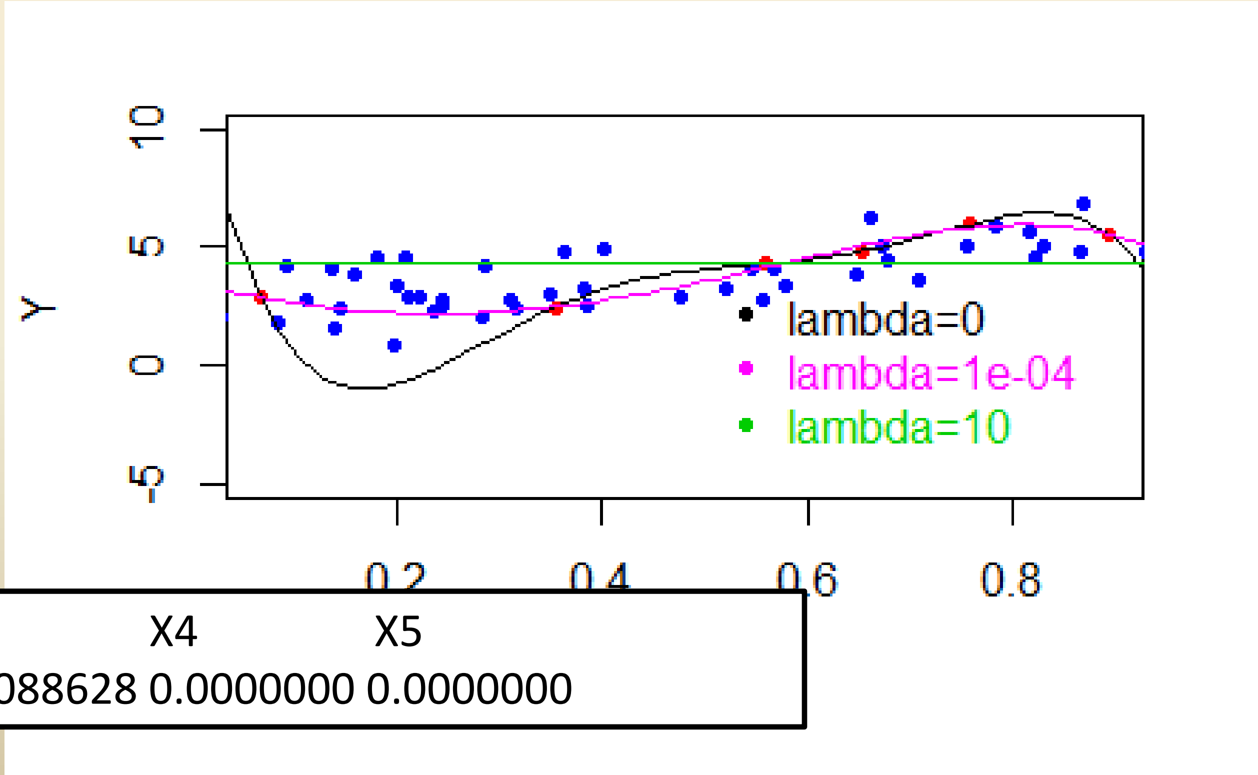
L2-регуляризация
(Ridge или регуляризация по Тихонову) $$R(w) = \|w\|_2 = \sum_{i=1}^{l} w_i^2$$

Регуляризация

- В R поддерживается регуляризация для регрессионных моделей в пакетах lars, mars

```
lasso <- lars(train, Y, type = "lasso")
```

- Коэффициент при дополнительном слагаемом в функции ошибки определяет величину «штрафа» за большие коэффициенты.



X	X2	X3	X4	X5
0.0000000	3.3393577	0.5088628	0.0000000	0.0000000