Онлайн-курс "Быстрый старт в искусственный интеллект"
Модуль 1. Машинное обучение
Дополнительные материалы к
уроку 1.2 "Линейные алгоритмы в машинном обучении"

Линейная регрессия

Линейная регрессия — это алгоритм регрессии, предполагающий, что целевая переменная y имеет линейную зависимость от признаков $(x_1, x_2, \ldots, x_k)$. Иными словами, алгоритм a представляется в виде

$$a(x) = w_1 x_1 + w_2 x_2 + \ldots + w_k x_k + b = \langle w, x \rangle + b,$$

где $w = (w_1, w_2, \ldots, w_k)$ называется *вектором весов* линейной модели, а b — *свободным членом*.

Прежде всего давайте обратим внимание, что свободный член b не играет определяющей роли для линейной закономерности. В самом деле, давайте добавим к каждому объекту фиктивный признак x_{k+1}, тождественно равный единице. Обозначим

$$\tilde{x} = (x_1, x_2, \ldots, x_k, 1), \ \tilde{w} = (w_1, w_2, \ldots, w_k, b).$$

Тогда имеем

$$\langle x, w \rangle + b = w_1 x_1 + \ldots + w_k x_k + b = \langle \tilde{x}, \tilde{w} \rangle.$$

Таким образом, задача линейной регрессии со свободным членом свелась к новой задаче линейной регрессии, уже без свободного члена. Итак, мы решаем задачу поиска оптимального алгоритма a в виде

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

Алгоритм линейной регрессии основан на методе минимизации эмпирического риска для квадратичной функции потерь. А именно, пусть дана обучающая выборка $X = \left\{x^1, x^2, \ldots, x^\ell\right\}$, а также вектор правильных ответов $y = (y^1, y^2, \ldots, y^\ell)$. Тогда задача минимизации для линейной регрессии, как мы обсуждали в лекции, с поправкой на отсутствие коэффициента b, записывается в виде

$$\sum_{i=1}^{\ell} \left(\langle x^i, w \rangle - y^i\right)^2 \to \min_w \tag{1}$$

Прежде чем предъявить решение данной задачи в явном виде, перепишем её в матричном виде. Положим

$$X = \begin{pmatrix} x_1^1 & x_2^1 & \ldots & x_k^1 \\ x_1^2 & x_2^2 & \ldots & x_k^2 \\ \ldots & \ldots & \ldots & \ldots \\ x_1^\ell & x_2^\ell & \ldots & x_k^\ell \end{pmatrix}, \ w = \begin{pmatrix} w_1 \\ w_2 \\ \ldots \\ w_k \end{pmatrix}$$

— соответственно матрица "объекты-признаки": в строчках записаны векторы признаков каждого объекта, и вектор-столбец весов алгоритма.

Тогда рассмотрим произведение Xw по правилам перемножения матриц. Имеем

$$Xw = \begin{pmatrix} x_1^1 & x_2^1 & \ldots & x_k^1 \\ x_1^2 & x_2^2 & \ldots & x_k^2 \\ \ldots & \ldots & \ldots & \ldots \\ x_1^\ell & x_2^\ell & \ldots & x_k^\ell \end{pmatrix} \times \begin{pmatrix} w_1 \\ w_2 \\ \ldots \\ w_k \end{pmatrix} = \begin{pmatrix} \langle x^1, w \rangle \\ \langle x^2, w \rangle \\ \ldots \\ \langle x^\ell, w \rangle \end{pmatrix}.$$

Наконец, имеем

$$Xw - y = \begin{pmatrix} \langle x^1, w \rangle - y^1 \\ \langle x^2, w \rangle - y^2 \\ \ldots \\ \langle x^\ell, w \rangle - y^\ell \end{pmatrix}.$$

Таким образом, в компонентах вектора $Xw - y$ записаны значения отклонений алгоритма от правильных ответов. Тогда по теореме Пифагора квадрат длины вектора $Xw - y$ равен сумме квадратов отклонений или, иными словами,

$$||Xw + b - y||^2 = \sum_{i=1}^{\ell} \big(\langle x^i, w\rangle - y^i\big)^2,$$

что в точности совпадает с выражением (1). Таким образом, задача минимизации переписывается в компактном виде

$$||Xw - y||^2 \to \min_w. \tag{2}$$

Получившуюся задачу минимизации называют *задачей наименьших квадратов*.

Утверждение (без доказательства). Задача (2) имеет точное решение, которое записывается в виде

$$w^* = (X^T X)^{-1} X^T y. \tag{3}$$

Таким образом, мы решили поставленную нами задачу минимизации суммы квадратов отклонений. Формула из Утверждения собственно и является алгоритмом линейной регрессии.

Замечание. Как правило, на практике оказывается целесообразнее решать задачу (2) с помощью метода стохастического градиентного спуска. Во-первых, операция взятия обратной матрицы, фигурирующая в формуле (3), слишком дорогая по времени. Во-вторых, если матрица $X^T X$ не является обратимой (или по крайней мере близка к необратимой), то взятие обратной матрицы оказывается сопряжено с проблемами вычислительной стабильности. Впрочем, в библиотеках машинного обучения все необходимые вычисления реализованы программно.

Замечание. Для большей вычислительной стабильности и борьбы с переобучением задачу (2) видоизменяют следующим образом:

$$\frac{1}{\ell}||Xw - y||^2 + C||w||^p \to \min_w,$$

где C — фиксированная положительная константа. Такая модификация линейной регрессии называется Ridge-регрессией при $p = 2$ и Lasso-регрессией для $p = 1$. Алгоритмы Ridge и Lasso также реализованы в библиотеке scikit-learn.

Логистическая регрессия

Логистическая регрессия — это алгоритм линейной классификации, то есть алгоритм классификации на классы $\{+1, -1\}$, имеющий вид $a(x) = \text{sign}(\langle w, x\rangle)$, где

$$\text{sign}(x) = \begin{cases} +1, & x \geqslant 0; \\ -1, & x < 0. \end{cases}$$

Как и в алгоритме линейной регрессии, мы можем считать, что свободный член b тождественно равен 0.

Отличительной особенностью логистической регрессии является то, что основным объектом рассмотрения является не класс объекта, а *вероятность принадлежности объекта классу* $+1$. Как упоминалось в видео, искомая вероятность вычисляется по формуле

$$\text{P}_w(y(x) = +1) = \sigma(\langle w, x\rangle),$$

где

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Очевидно, вероятность принадлежности объекта классу -1 высчитывается как $1 - \mathrm{P}_w(y(x^i) = +1)$.

Итак, необходимо подобрать такие коэффициенты w, чтобы вероятность $\mathrm{P}_w(y(x^i) = +1)$ была как можно больше на объектах, для которых $y^i = +1$, а вероятность $1 - \mathrm{P}_w(y(x^i) = +1)$ — как можно больше на объектах, для которых $y^i = -1$.

Тогда логично записать произведение максимизируемых вероятностей по всем объектам обучающей выборки и максимизировать его по вектору весов w. Таким образом, имеем задачу оптимизации

$$\prod_{y^i=+1} \mathrm{P}_w(y(x^i) = +1) \cdot \prod_{y^i=-1} \big(1 - \mathrm{P}_w(y(x^i) = +1)\big) \to \max_w .$$

Оптимизировать произведение большого количества сомножителей сложно, поэтому логарифмируем произведение и будем оптимизировать сумму логарифмов. Также домножим обе части на -1, чтобы иметь задачу минимизации (для единообразия с задачей линейной регрессии). Имеем

$$\sum_{y^i=+1} -\ln \mathrm{P}_w(y(x^i) = +1) + \sum_{y^i=-1} -\ln\big(1 - \mathrm{P}_w(y(x^i) = +1)\big) \to \min_w . \tag{4}$$

Преобразуем выражение $-\ln(P_w(y(x) = +1))$ для объектов класса $+1$. Имеем

$$-\ln(P_w(y(x^i) = +1)) = -\ln\left(\frac{1}{1 + e^{-\langle w, x^i\rangle}}\right) = \ln\left(1 + e^{-\langle w, x^i\rangle}\right) = \ln\left(1 + e^{-y^i\langle w, x^i\rangle}\right).$$

Здесь мы используем то, что $y^i = +1$. Аналогично, если $y^i = -1$, то

$$-\ln(P_w(y(x^i) = -1)) = \ln\left(1 + e^{\langle w, x^i\rangle}\right) = \ln\left(1 + e^{-y^i\langle w, x^i\rangle}\right).$$

Таким образом, задачу (4) можно переписать в виде

$$\sum_{i=1}^{\ell} \ln\left(1 + e^{-y^i\langle w, x^i\rangle}\right) \to \min_w . \tag{5}$$

Определение. Функция $\ln\left(1 + e^{-y\langle w, x\rangle}\right)$ называется *логистической функцией потерь*.

Определение. *Логистическая регрессия* — это алгоритм линейной классификации, который находит вектор параметров w, который является решением задачи минимизации (5).

Замечание. Задача минимизации логистической функции потерь не имеет аналитического решения, в отличие, например, от задачи минимизации квадратичной функции потерь в задаче линейной регрессии. Для решения этой задачи используют метод стохастического градиентного спуска.

Для большей вычислительной стабильности и борьбы с переобучением задачу (5) видоизменяют следующим образом:

$$\frac{1}{\ell}\sum_{i=1}^{\ell} \ln\left(1 + e^{-y^i\langle w, x^i\rangle}\right) + C||w||^2 \to \min_w .$$

где C — фиксированная положительная константа. Такая модификация логистической регрессии реализована в библиотеке scikit-learn с параметром по умолчанию $C = 1$.