Linear Regression

Модель многомерной линейной регрессии:

$$f(x,\theta) = \sum_{j=1}^{m} \theta_j f_j(x),\ \theta \in \mathbb{R}^n$$

Матричные обозначения:

$$F = \begin{pmatrix} f_1(x_1) & \dots & f_m(x_1) \\ \dots & \dots & \dots \\ f_1(x_n) & \dots & f_m(x_n) \end{pmatrix},\ y = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix},\ \theta = \begin{pmatrix} \theta_1 \\ \dots \\ \theta_n \end{pmatrix}$$

Эмпирический риск в матричной записи:

$$\mathcal{L}(\theta, \mathcal{D}) = \sum_{i=1}^{n} (f(x_i, \theta) - y_i)^2 = \|F\theta - y\|^2 \to \min_{\theta \in \mathbb{R}}$$

Сумма квадратов в качестве ошибки удобна тем, что так мы можем свести задачу в матричным операциям (типа $X^2 = X^T X$).
Для линейной регрессии МНК позволяет найти точное решение.

Условие минимума:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = 2F^\top (F\theta - y) = 0$$
$$\theta^* = (F^\top F)^{-1} F^\top y$$

$F^+ = (F^\top F)^{-1} F^\top$ **— псевдообратная матрица (обратное преобразование Мура-Пенроуза)**
$P_F = FF^+$ **— проекционная матрица**

Решение:

$$\theta^* = F^+ y$$

Минимальное приближение:

$$\mathcal{L}(\theta^*) = \|P_F y - y\|^2$$

Почему **проекционная** матрица?
Мы пытаемся посчитать минимальное расстояние от y до линейной комбинации (умножаем на столбец θ) строк F^T. А точнее, найти ближайшую точку до y на плоскости, задаваемой векторами F^T, которая параметризуется компонентами вектора θ.

Регуляризация

$$\sum_i (F_i\theta - y_i)^2 + \tau \sum_i \theta_i^2 - \text{гребневая регуляризация, метод Тихонова}$$

К F мы добавляли вектор $(1, 1, ..., 1)$. Если $\mathbb{E}[F_j] = 0, \mathbb{E}[y] = 0$, то можно про него забыть. Важно понимать, что если нормализованы и входные, и выходные данные, то модель обучилась именно на них, и θ тоже будут найдены для нормализованных данных. Нормализация - линейная операция, представляющая собой просто умножение на какие-то матрицы. А значит, если в алгоритм подавать и забирать данные не нормализованные, то надо выполнять для θ_{norm} ассоциативно слева и справа умножение на матрицу нормализации и обратную к ней.
Если входные данные нормализованы и декоррелированы, $\mathbb{D}[F_j] = 1, \mathbb{D}[y] = 1$?

$$\theta = (F^T F)^{-1} F^T y = (\Sigma \cdot n)^{-1} F^T y = diag(n^{-1}) F^T y$$
$$\theta_j = \frac{F_j y}{n} = cov < F_j, y >= corr < F_j, y >$$

Если есть веса w_i для объектов?

$$\sum_i w_i (x_i a - y_i)^2 = \sum_i (\sqrt{w_i} x_i a - \sqrt{w_i} y_i)^2 = \sum_i (x_i^{'} a - y_i^{'})^2$$

С такой заменой годятся все выше описанные методы.

Но вернёмся к регуляризации.

$$\sum_i (F_i\theta - y_i)^2 + \tau \sum_i \theta_i^2$$
$$2F^T (F\theta - y) + \tau \cdot 2 \cdot \theta = 0$$
$$\theta = (F^T F + diag(\tau))^{-1} F^T y$$

Это также решает проблему, когда F ЛЗ и обратная не берется. Мы прибавляем диагональную матрицу, тем самым увеличивая детерминант.

МНК при SVD

$$F^+ = (UDV^\top VDU^\top)^{-1}UDV^\top = UD^{-1}V^\top = \sum_{j=1}^{m} \frac{1}{\sqrt{\lambda_j}} u_j v_j^\top;$$

$$\theta^* = F^+ y = UD^{-1}V^\top y = \sum_{j=1}^{m} \frac{1}{\sqrt{\lambda_j}} u_j (v_j^\top y);$$

$$F\theta^* = P_F y = (VDU^\top)UD^{-1}V^\top y = VV^\top y = \sum_{j=1}^{m} v_j (v_j^\top y);$$

$$\|\theta^*\|^2 = \left\|D^{-1}V^\top y\right\|^2 = \sum_{j=1}^{m} \frac{1}{\lambda_j} (v_j^\top y)^2.$$

- Когда мы можем вычислить сингулярное разложение, мы можем легко найти решение для МНК.
- Сингулярное разложение вычисляется за $\mathcal{O}(nm^2 + m^3)$
- Сингулярное разложение — важный инструмент во многих других задачах машинного обучения, в первую очередь, в снижении размерности.

$$(F^T F)^{-1} F^T y \xrightarrow{time} nm^2 + m^3 + nm^2 + nm$$

Если предсказываем не одно значение y, а целый вектор, то есть $y = [n \times k]$? Можно предсказывать по отдельности каждую координату. Но тогда суммарное время умножится на k. А можно предпосчитать одинаковое для всех координат $(F^T F)^{-1} F^T$:)