Линейная регрессия

Зотиков Дима <dmitry.zotikov@ungrund.org>

30 сентября 2022 г.

1 Модель линейной регрессии

Понятие *линейной регрессии* имеет несколько интерпретаций. С наиболее общими из них можно будет познакомиться на старших курсах в рамках соответствующих предметов. Для данной работы имеет смысл рассмотреть простую постановку, в которой линейная регрессия—это метод, позволяющий устанавливать *линейную* зависимость между «*признаками*» («предикторами», «features») и «*наблюдениями*» («response», «outcome»). В частности, пусть $\mathcal{D} = \{(\mathbf{x}_i, y_i)_{i=1}^n\}$ есть множество данных, каждый элемент которого есть пара, состоящая из значений признаков $\mathbf{x}_i \in \mathbb{R}^d$ и наблюдений $y_i \in \mathbb{R}$. Подразумевается, что признаки и наблюдения связаны *моделью* $f(\mathbf{x}_i)$, причем в данных значения аддитивно «*зашумлены*» при помощи случайных величин ϵ_i; таким образом

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad i \in 1 : n.$$

Будем считать, что модель линейна, т.е. является линейной комбинацией *вектора параметров* $\boldsymbol{\beta} \in \mathbb{R}^{d+1}$ и вектора признаков $\mathbf{x}_i$, $i \in 1 : n$ а шум имеет нормальное распределение:

$$f(\mathbf{x}_i \mid \boldsymbol{\beta}) = \beta_0 + \sum_{j=1}^{d} \beta_j x_{ij}, \ \epsilon_i \sim \mathrm{N}(0, \sigma^2), \quad i \in 1 : n$$

«Настоящий» вектор $\boldsymbol{\beta}^*$ неизвестен. Представляет интерес нахождение такого оптимального $\hat{\boldsymbol{\beta}}$ («*оценки*»), что он максимально «близок» к истинному $\boldsymbol{\beta}$. Тогда можно было бы, к примеру, «предсказывать» наблюдения по одним лишь признакам или анализировать, насколько сильно каждый предиктор влияет на наблюдения.

Существует несколько способов оптимизации $\boldsymbol{\beta}$; мы будем использовать *метод градиентного спуска.* К его плюсам можно отнести универсальность (работает не только для линейной регрессии, а для большого класса моделей) и применимость в задачах, где данных настолько много, что другие методы перестают работать (не помещаются в оперативную память, например).

2 Метод наименьших квадратов

В курсе статистического обучения показывается, что нахождение оптимальной оценки $\hat{\boldsymbol{\beta}}$ эквивалентно нахождению такого $\hat{\boldsymbol{\beta}}$, что расхождения между $f(\mathbf{x}_i \mid \hat{\boldsymbol{\beta}})$ и y_i были бы минимально возможные для всех $i \in 1 : n$. В частности, величина

$$e_i = f(\mathbf{x}_i \mid \boldsymbol{\beta}) - y_i, \quad i \in 1 : n$$

называется *остаток; сумма квадратов* таких *остатков* по всем данным («*Residual Sum of Squares*») есть

$$\mathrm{RSS}(\boldsymbol{\beta} \mid \mathcal{D}) = \sum_{i=1}^{n} \left(f(\mathbf{x}_i \mid \boldsymbol{\beta}) - y_i\right)^2.$$

Квадрат здесь играет роль функции модуля, но проще в оптимизации. Ясно, что для нахождения оптимальной оценки $\hat{\boldsymbol{\beta}}$, следует минимизировать RSS:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{d+1}} \mathrm{RSS}(\boldsymbol{\beta} \mid \mathcal{D}).$$

Вычислительно удобнее минимизировать не RSS, а среднее остатков, т.е.

$$R(\boldsymbol{\beta} \mid \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i \mid \boldsymbol{\beta}) - y_i)^2 = \frac{1}{n} \mathrm{RSS}(\boldsymbol{\beta} \mid \mathcal{D}),$$

также называемое функцией оценки *риска* в контексте статистической оптимизации.

Данный метод имеет название «*Метод наименьших квадратов*» («*Least Squares*») и может применяться для решения ряда других оптимизационных проблем.

3 Градиентный спуск

Градиентный спуск является итеративной процедурой, на каждом шаге которого происходит движение к минимуму объектной функции по её градиенту:

1. Пусть на шаге t есть текущее приближение $\hat{\boldsymbol{\beta}}^{(t)}$
2. Вычисляется градиент $\nabla R(\hat{\boldsymbol{\beta}}^{(t)} \mid \mathcal{D})$
3. Следующее лучшее приближение есть $\hat{\boldsymbol{\beta}}^{(t+1)} = \hat{\boldsymbol{\beta}}^{(t)} - \alpha \nabla R(\hat{\boldsymbol{\beta}}^{(t)} \mid \mathcal{D})$. α—предзаданный параметр, по смыслу задающий «*скорость оптимизации*» («*learning rate*»).
4. Алгоритм завершается, когда выполняется некоторый критерий останова.

4 Параллелизация алгоритма

Для действительно больших объемов данных и при наличии вычислительных мощностей алгоритм имеет смысл параллелизовать. Конкретнее, аналитическая форма градиента $\nabla R(\hat{\boldsymbol{\beta}}^{(t)} \mid \mathcal{D})$ позволяет посчитать его «по частям» в параллели на рабочих вычислительных нодах, а затем суммировать полученные значения на главной ноде.