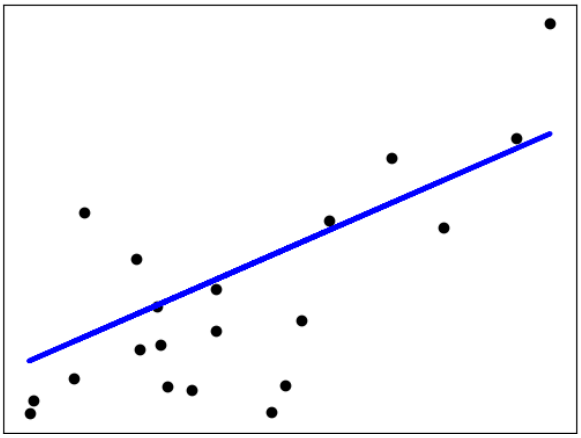
Задание 8. Линейная регрессия и метод главных компонент (применение сингулярного разложения)

I. Многомерная линейная регрессия. Использование сингулярного разложения (подробней см. в [1, стр. 55–71], [4, 5])

(документация: sklearn.linear_model.Ridge — scikit-learn 1.1.3 documentation https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model)

Пусть есть обучающая выборка $X = (x_i, y_i)_{i=1}^{l}$, где $x_i \in R^n$ – набор из l объектов, у каждого такого объекта x_i есть n признаков $f_j(x_i)$ $(j = 1..n)$, и $y_i \in R$ – известные ответы обучающей выборки.



В линейной модели **восстановления регрессии** решение ищется в виде линейной функции: суммы всех признаков $f_j(x)$ с коэффициентами w_j (взвешенная сумма всех признаков):

$$a(x, \alpha) = \langle x, w \rangle = \sum_{j=1}^{n} w_j f_j(x).$$

Используем следующие матричные обозначения. Матрица «объектов–признаков» размера $l \times$ n, где l – количество объектов (строк матрицы), а n – количество признаков (столбцов матрицы):

$$F_{l \times n} = \begin{pmatrix} f_1(x_1), \dots, f_n(x_1) \\ \dots \\ f_1(x_l), \dots, f_n(x_l) \end{pmatrix}.$$

Запишем также вектор известных ответов и вектор коэффициентов в решении:

$$y_{l \times 1} = \begin{pmatrix} y_1 \\ \dots \\ y_l \end{pmatrix}, \qquad w_{n \times 1} = \begin{pmatrix} w_1 \\ \dots \\ w_n \end{pmatrix}.$$

Тогда вектор ответов y приближается методом по формуле $y \approx Fw$:

$$y_{l\times 1} \approx a(x,\alpha) = F_{l\times n} w_{n\times 1}.$$

Функционал качества метода наименьших квадратов тоже можно записать в матричной форме

$$Q(w) = \sum_{i=1}^{l} (a(x_i, w) - y_i)^2 = \|Fw - y\| \to \min_w.$$

Необходимое условие минимума в матричном виде имеет вид:

$$\frac{\partial Q(w)}{\partial w} = 2F^T(Fw - y) = 0.$$

Откуда получается система метода наименьших квадратов в матричной форме:

$$F^T F w = F^T y,$$

где $F^T F$ – известная матрица размерности $n \times n$. Из этой системы линейных алгебраических уравнений (СЛАУ) находится неизвестный вектор коэффициентов w:

$$w = (F^T F)^{-1} F^T y = F^+ y,$$

где обозначили матрицу $F^+ = (F^T F)^{-1} F^T$. И решение имеет вид:

$$a(x,\alpha) = Fw = FF^+ y = F(F^T F)^{-1} F^T y.$$

Но в задачах машинного обучения метод наименьших квадратов в чистом виде часто не дает хорошего результата, т.к. матрица может быть большого порядка и плохо обусловлена или вообще вырождена из-за линейной зависимости признаков в задаче (мультиколлинеарности признаков).

Для решения этой задачи используем **сингулярное разложение** матрицы (SVD-разложение), которое может быть найдено и для плохо обусловленных матриц и для вырожденных. Произвольная $l \times n$ – матрица может быть представлена в виде сингулярного разложения (Singular Value Decomposition, SVD):

$$F = VDU^T,$$

где $l \times l$ – матрица $V = (v_1, \dots, v_l)$ ортогональна ($V^T V = I_l$), столбцы v_j – собственные векторы матрицы FF^T;

$n \times n$ – матрица $U = (u_1, \dots, u_n)$ ортогональна ($U^T U = I_n$), столбцы u_j – собственные векторы матрицы $F^T F$;

$l \times n$ – матрица D, у которой элементы, лежащие на главной диагонали – это сингулярные числа, а все элементы, не лежащие на главной диагонали, являются нулевыми. $D = \mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$, где $\lambda_j \geq 0$ – собственные значения матриц $F^T F$ и FF^T, $\sqrt{\lambda_j}$ – сингулярные числа.

Если в формулы метода наименьших квадратов подставить вместо цы F ее сингулярное разложение, то получим:

1) матрица F^+:

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

2) вектор коэффициентов w:

$$w = F^+ y = \sum_{j=1}^{n} \frac{1}{\sqrt{\lambda_j}} u_j ({v_j}^T y);$$

3) решение $a(x, \alpha)$:

$$a(x, \alpha) = Fw = (VDU^T) UD^{-1} V^T y = VV^T y = \sum_{j=1}^{n} v_j ({v_j}^T y);$$

4) норма вектора коэффициентов:

$$\|w\| = \sum_{j=1}^{n} \frac{1}{\lambda_j} \left({v_j}^T y_j\right)^2.$$

Видно, что если есть сингулярные числа λ_j близкие к нулю, то вектор коэффициентов w становится большим. Это говорит о том, что в задаче есть мультиколлинеарность признаков. В этом методе можно бороться с мультиколлинеарностью немного модифицировав метод, и ход решения не сильно меняется.

С проблемой **мультиколлинеарности и переобучения** можно бороться разными способами:

- отбор признаков (заранее убираем признаки, которые могут оказаться линейно зависимыми);
- регуляризация (добавляем ограничение на норму $\|w\|$, как и в других методах ранее);
- преобразование признаков (например, с помощью метода главных компонент).

Гребневая регрессия получается при применении **регуляризации** L_2:

$$Q(w) = \|Fw - y\| + \frac{\tau}{2} \|w\|^2 \to \min_w.$$

Тогда вектор коэффициентов w регуляризованного метода наименьших квадратов имеет вид:

$$w_\tau = \sum_{j=1}^{n} \frac{\sqrt{\lambda_j}}{\lambda_j + \tau} u_j ({v_j}^T y).$$

Благодаря τ нет деления на близкие к нулю значения. Преимущество метода состоит в том, что можно один раз вычислить SVD-разложение матрицы и использовать его для всех τ, не надо каждый раз его пересчитывать.

Метод LASSO получается при применении **регуляризации** L_1:

$$Q(w) = \|Fw - y\| + \tau \sum_{j=1}^{n} |w_j| \to \min_w.$$

Особенностью метода является то, что метод обнуляет коэффициенты w_j, т.е. он фактически выбрасывает менее важные признаки или осуществляет отбор признаков.

Комбинация этих двух регуляризаций приводит к методу ElasticNet:

$$Q(w) = \|Fw - y\| + \frac{\tau_1}{2}\|w\|^2 + \tau_2 \sum_{j=1}^{n} |w_j| \to \min_w.$$

II. Метод главных компонент

Использование сингулярного разложения

(подробней см. в [1, стр. 157–173], [2, стр. 185–232], [3–6])

(документация: sklearn.decomposition.PCA — scikit-learn 1.2.0 documentation https://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition)

Метод главных компонент позволяет перейти от исходных признаков к новым линейно независимым признакам, уменьшив их количество и оставив из них самые важные. Переход к новым признакам получается с использованием той же самой матрицы SVD-разложения.

Постановка задачи:

Имеются исходные числовые признаки: $f_1(x), \ldots, f_n(x)$.

Переходим к новым числовым признакам: $g_1(x), \ldots, g_m(x), m \le n$.

Их количество меньше исходных, но так, чтобы старые признаки можно было восстановить по новым признакам

$$\hat{f}_j(x) = \sum_{s=1}^{m} g_s(x) u_{js}$$

как можно точнее на обучающей выборке:

$$\sum_{i=1}^{l} \sum_{j=1}^{n} (\hat{f}_j(x_i) - f_j(x_i))^2 \to \min_{g,u}.$$

В матричных обозначениях матрицы “объекты-признаки”:

$$F_{l\times n} = \begin{pmatrix} f_1(x_1), \dots, f_n(x_1) \\ \dots \\ f_1(x_l), \dots, f_n(x_l) \end{pmatrix}; \qquad G_{l\times m} = \begin{pmatrix} g_1(x_1), \dots, g_m(x_1) \\ \dots \\ g_1(x_l), \dots, g_m(x_l) \end{pmatrix}.$$

Матрица преобразований U переводит признаки G в приближение к старым $\hat{F}$:

$$U_{n\times m} = \begin{pmatrix} u_{11}, \dots, u_{1m} \\ \dots \\ u_{n1}, \dots, u_{nm} \end{pmatrix}; \qquad \hat{F} = GU^T \approx F.$$

Есть теорема о том, что если $m \leq \operatorname{rang} F$, то минимум $\|GU^T - F\|^2$ достигается, когда столбцы матрицы $U = (u_1, \dots, u_m)$ – это собственные векторы матрицы $F^T F$, соответствующие максимальным m собственным значениям λ_j матрицы $F^T F$. В частном случае при $m = n$ это разложение совпадает с сингулярным разложением. Т.е. получается урезанное SVD-разложение, из которого убрали минимальные собственные значения. Число m заранее неизвестно, его можно выбирать, например, основываясь на величинах собственных значений λ_j матрицы $F^T F$.

Задания

I. Прочитайте пункты “ I. Многомерная линейная регрессия ”, “II. Метод главных компонент” и рекомендуемую литературу.

II. Выполните задание из в файла "statement-linreg.pdf".

III. Выполните задание из в файла "statement-pca.pdf".

Литература

[1] Андреас Мюллер, Сара Гвидо Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. 393 с.

[2] Рашка Себастьян, Мирджалили Вахид Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2, 3-е изд.: Пер. с англ. СПб. : ООО "Диалектика", 2020. 848 с.

[3] Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. Пер. с англ. СпБ.: ООО Альфа-книга: 2018. 688 с.

[4] К.В. Воронцов Математические методы обучения по прецедентам (теория обучения машин). 141 с. (Voron-ML-1.pdf)

[5] Машинное обучение (курс лекций, К.В.Воронцов): Многомерная линейная регрессия Машинное обучение (курс лекций, К.В.Воронцов) (machinelearning.ru) (“Многомерная линейная регрессия. Метод главных компонент”: https://www.youtube.com/watch?v=tCE_vnPoU44)

[6] Крис Элбон Машинное обучение с использованием Python. Сборник рецептов. Пер. с англ. СПб. : БХВ-Петербург, 2019. 384 с.