Неоднородные СЛАУ

Совокупность уравнений первой степени, в которых каждая переменная и коэффициенты в ней являются вещественными числами, называется **системой линейных алгебраических уравнений (СЛАУ)** и в общем случае записывается как:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2m}x_m = b_2 \\ \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nm}x_m = b_n \end{cases} \quad (1)$$

где n — количество уравнений,
m — количество переменных,
x_i — неизвестные переменные системы,
a_{ij} — коэффициенты системы,
b_i — свободные члены системы.

СЛАУ (1) называется **однородной**, если все свободные члены системы равны 0 $b_1 = b_2 = \ldots = b_n = 0$:

$$\text{СЛАУ} - \text{однородная, если } \forall b_i = 0$$

СЛАУ (1) называется **неоднородной**, если хотя бы один из свободных членов системы отличен от 0:

$$\text{СЛАУ} - \text{однородная, если } \exists b_i \neq 0$$

СЛАУ в матричном виде:

$$A\vec{w} = \vec{b}$$

$$\begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \ldots \\ w_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_n \end{pmatrix}$$

где A — матрица системы, $\vec{w}$ — вектор неизвестных коэффициентов, а $\vec{b}$ — вектор свободных коэффициентов.

Расширенная матрица системы

Расширенной матрицей системы $(A|\vec{b})$ неоднородных СЛАУ называется матрица, составленная из исходной матрицы и вектора свободных коэффициентов:

$$(A|\vec{b}) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1m} & b_1 \\ a_{21} & a_{22} & \dots & a_{2m} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} & b_n \end{array}\right)$$

Над расширенной матрицей неоднородной СЛАУ можно производить все те же действия, что и над обычной, а именно:

- складывать/вычитать между собой строки/столбцы матрицы;
- умножать строки/столбцы на константу;
- менять строки/столбцы местами.

Случаи решения СЛАУ

Существует три случая при решении неоднородных СЛАУ:

1. **"Идеальная пара"**

Это так называемые определённые системы линейных уравнений, имеющие единственные решения.

2. **"В активном поиске"**

Неопределённые системы, имеющие бесконечно много решений.

3. **"Всё сложно"**

Переопределённые системы, которые не имеют точных решений.

Случай "Идеальная пара"

Системы, имеющие только одно решение, называются **совместными**.

Теорема Кронекера — Капелли:

Неоднородная система линейный алгебраических уравнений $A\vec{w} = \vec{b}$ является совместной тогда и только тогда, когда ранг матрицы системы A **равен** рангу

расширенной матрицы системы $(A|\vec{b})$ и **равен** количеству независимых переменных m:

$$rk(A) = rk(A|\vec{b}) = m \leftrightarrow \exists!\vec{w} = (w_1, w_2, \dots w_m)^T$$

Причём решение системы будет равно:

$$\vec{w} = A^{-1}\vec{b}$$

Случай "В активном поиске"

Следствие №1 из теоремы Кронекера — Капелли:

Если ранг матрицы системы A **равен** рангу расширенной матрицы системы $(A|\vec{b})$, но **меньше**, чем количество неизвестных m, то система имеет бесконечное множество решений:

$$rk(A) = rk\left(A|\vec{b}\right) < m \leftrightarrow \infty \text{ решений}$$

Случай "Всё сложно"

Следствие №2 из теоремы Кронекера — Капелли:

Если ранг матрицы системы A меньше, чем ранг расширенной матрицы системы $(A|\vec{b})$, то система несовместна, то есть не имеет точных решений.

$$rk(A) < (A|\vec{b}) \leftrightarrow \nexists \text{ решений}$$

Можно попробовать найти приблизительное решение — вопрос лишь в том, какое из всех этих решений лучшее.
На этот вопрос отвечает **метод наименьших квадратов**. Согласно ему, наилучшее приблизительное решение вычисляется по формуле:

$$\hat{w} = (A^T A)^{-1} \cdot A^T b$$

Решением является ортогональная проекция вектора b на столбцы матрицы A.

Линейная регрессия по МНК

Рассматривается задача регрессии:

y — целевая переменная

$x_1, x_2, \dots, x_k$ — признаки/ факторы/ регрессоры

В задаче регрессии есть N наблюдений — это обучающая выборка или датасет, представленный в виде таблицы. В столбцах таблицы располагаются векторы признаков $\vec{x}_i$.

$$\vec{y} \in \mathbb{R}^N$$

$$\overrightarrow{x_1}, \overrightarrow{x_2}, \dots, \overrightarrow{x_k} \in \mathbb{R}^N$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix}, \quad \begin{pmatrix} x_{11} \\ x_{12} \\ \dots \\ x_{1N} \end{pmatrix}, \dots, \begin{pmatrix} x_{k1} \\ x_{k2} \\ \dots \\ x_{kN} \end{pmatrix}$$

В качестве регрессионной модели будем использовать модель линейной регрессии:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_k x_k,$$

или

$$y = (\vec{w}, \vec{x})$$

$\vec{w} = (w_0, w_1, \dots, w_k)^T$ — веса (коэффициенты уравнения линейной регрессии), а $\vec{x} = (1, x_1, x_2, \dots, x_k)^T$.

Мы пытаемся найти такие веса w, чтобы для каждого наблюдения наше равенство было выполнено. Таким образом получается N уравнений на $k + 1$ неизвестную.

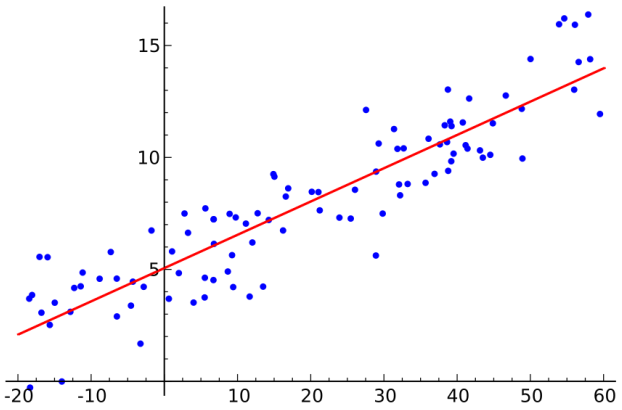
$$\begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} = w_0 \begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \end{pmatrix} + w_1 \begin{pmatrix} x_{11} \\ x_{12} \\ \dots \\ x_{1N} \end{pmatrix} + \dots + w_k \begin{pmatrix} x_{k1} \\ x_{k2} \\ \dots \\ x_{kN} \end{pmatrix}$$

Или в привычном виде систем уравнений:

$$\begin{cases} w_0 1 + w_1 x_{11} + \dots + w_k x_{k1} = y_1 \\ w_0 1 + w_1 x_{12} + \dots + w_k x_{k2} = y_2 \\ \dots \\ w_0 1 + w_1 x_{1N} + \dots + w_k x_{kN} = y_N \end{cases}$$

Как правило, N гораздо больше k (количество строк в таблице с данными намного больше количества столбцов таблицы), и система переопределена, значит точного решения не имеется, поэтому можно найти только приближённое.

Полученной СЛАУ можно дать **геометрическую интерпретацию**. Если представить каждое наблюдение на графике в виде точки (см. рисунок ниже), то уравнение линейной регрессии будет задавать прямую. Приравняв уравнение прямой к целевому признаку, мы требуем, чтобы эта прямая проходила через все точки в нашем наборе данных. Конечно же, это условие не может быть выполнено полностью, так как в данных всегда присутствует какой-то шум и идеальной прямой (гиперплоскости) не получится, но зато можно построить приближённое решение.



Составим матрицу системы A (матрицу наблюдений), записав в столбцы все наши регрессоры, включая регрессор константу:

$$A = \begin{pmatrix} 1 & x_{11} & \dots & x_{k1} \\ 1 & x_{12} & \dots & x_{k2} \\ \dots & \dots & \dots & \dots \\ 1 & x_{1N} & \dots & x_{kN} \end{pmatrix} \quad N \text{ строк}$$

$k+1$ столбец

Итоговая формула для оценки коэффициентов модели линейной регрессии:

$$\hat{\vec{w}} = \left(A^T A\right)^{-1} A^T \vec{y}$$

Предсказание будет строиться следующим образом:

$$\hat{y}_{NEW} = \hat{w}_0 + \hat{w}_1 x_{1,NEW} + \dots + \hat{w}_k x_{k,NEW}$$

или

$$\hat{y}_{NEW} = (\hat{\vec{w}}, \vec{x}_{NEW})$$

Проблемы в классической МНК-модели

Если матрица A^TA вырождена или близка к вырожденной, то хорошего решения у классической модели не получится. Такие данные называют **плохо обусловленными**.

Борьба с вырожденностью матрицы A^TA часто сводится к устранению «плохих» (зависимых) признаков. Для этого анализируют корреляционную матрицу признаков или матрицу их значений.

Особенности Linear Regression из sklearn

В реализации линейной регрессии в sklearn предусмотрена борьба с плохо определёнными (близкими к вырожденным и вырожденными) матрицами.

Для этого используется метод под названием **сингулярное разложение (SVD)**. Суть метода заключается в том, что в OLS-формуле используется не сама матрица A, а её диагональное представление из сингулярного разложения, которое гарантированно является невырожденным.

Однако данный метод только оберегает от ошибки при обращении плохо обусловленных и вырожденных матриц, но не гарантирует получения корректных коэффициентов линейной регрессии.

Стандартизация векторов

В линейной алгебре под стандартизацией вектора $\vec{x} \in R^n$ понимается операция, которая проходит в два этапа:

1) Центрирование вектора — операция приведения среднего к 0:

$$\vec{x}_{cent} = \vec{x} - \vec{x}_{mean}$$

2) Нормирование вектора — операция приведения диапазона вектора к масштабу от -1 до 1 путём деления центрированного вектора на его длину:

$$\vec{x}_{st} = \frac{\vec{x}_{cent}}{||\vec{x}_{cent}||}$$

где $\vec{x}_{mean}$ — вектор, составленный из среднего значения вектора $\vec{x}$, а $||\vec{x}_{cent}||$ — длина вектора $\vec{x}_{cent}$.

В результате стандартизации вектора всегда получается новый вектор, длина которого равна 1:

$$||\vec{x}_{st}|| = 1$$

До стандартизации мы прогоняли регрессию y на регрессоры x_1, x_2, ..., x_k и константу. Всего получалось $k + 1$ коэффициентов:

$$\vec{y} = w_0 + w_1\vec{x}_1 + w_2\vec{x}_2 + \ldots + w_k\vec{x}_k,$$

После стандартизации мы прогоняем регрессию стандартизованного y на стандартизованные регрессоры БЕЗ константы:

$$\vec{y}_{st} = w_{1_{st}}\vec{x}_{1_{st}} + w_{2_{st}}\vec{x}_{2_{st}} + \ldots + w_{k_{st}}\vec{x}_{k_{st}}$$

Для того чтобы проинтерпретировать оценки коэффициентов линейной регрессии (понять, каков будет прирост целевой переменной при изменении фактора на 1 условную единицу), нам достаточно построить линейную регрессию в обычном виде без стандартизации и получить обычный вектор $\hat{\vec{w}}$. Однако, чтобы корректно говорить о том, какой фактор оказывает на прогноз большее влияние, нам нужно рассматривать стандартизированную оценку вектора коэффициентов $\hat{\vec{w}}_{st}$.

Корреляционная матрица

Корреляционная матрица C — это матрица выборочных корреляций между факторами регрессий.

$$C = corr(X)$$

Выборочная корреляция — это корреляция, вычисленная на ограниченной выборке.

Выборочная корреляция отражает линейную взаимосвязь между факторами $\vec{x}_i$ и $\vec{x}_j$, реализации которых представлены в выборке.

$$c_{ij} = corr(\vec{x}_i, \vec{x}_j) = \frac{\sum_{l=1}^{n}(\vec{x}_{il} - x_{i_{mean}})(\vec{x}_{jl} - x_{j_{mean}})}{\sqrt{\sum_{l=1}^{n}(\vec{x}_{il} - x_{i_{mean}})^2 \cdot \sum_{l=1}^{n}(\vec{x}_{jl} - x_{j_{mean}})^2}}$$

Из вычисленных c_{ij} составляется матрица корреляций C. Если факторов k штук, то матрица C будет квадратной размера $dim\,(C) = (k, k)$:

$$C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1k} \\ c_{21} & c_{22} & \dots & c_{2k} \\ \dots & \dots & \dots & \dots \\ c_{k1} & c_{k2} & \dots & c_{kk} \end{pmatrix}$$

Матрица корреляций совпадает с матрицей Грама, вычисленной для стандартизированных векторов $\vec{x}_{1_{st}}$ и $\vec{x}_{2_{st}}$:

$$C = G(\vec{x}_{1_{st}}, \vec{x}_{2_{st}})$$

По свойству скалярного произведения корреляция является косинусом угла между центрированными векторами $\vec{x}_{i_{cent}}$ и $\vec{x}_{j_{cent}}$:

$$c_{ij} = corr(\vec{x}_i, \vec{x}_j) = \cos(\widehat{\vec{x}_{i_{cent}}, \vec{x}_{Jcent}}) = \frac{(\vec{x}_{i_{cent}}, \vec{x}_{jcent})}{\|\vec{x}_{i_{cent}}\| \cdot \|\vec{x}_{jcent}\|}$$

В NumPy матрица корреляций вычисляется функцией `np.corrcoef()`.

В Pandas матрица корреляций вычисляется методом `corr()`, вызванным от имени DataFrame.

Свойства корреляции:

➔ Если корреляция $c_{ij} = 1$, значит векторы $\vec{x}_i$ и $\vec{x}_j$ пропорциональны и сонаправлены.

➔ Если корреляция $c_{ij} = -1$, значит векторы $\vec{x}_i$ и $\vec{x}_j$ пропорциональны и противонаправлены.

➔ Если корреляция $c_{ij} = 0$, значит векторы $\vec{x}_i$ и $\vec{x}_j$ ортогональны друг другу и, следовательно, являются линейно независимыми.

Во всех остальных случаях между факторами $\vec{x}_i$ и $\vec{x}_j$ существует какая-то линейная взаимосвязь, причём чем ближе модуль коэффициента корреляции к 1, тем сильнее эта взаимосвязь.

Сила связи	Значение коэффициента корреляции
Отсутствие связи или очень слабая связь	0...+/- 0.3
Слабая связь	+/- 0.3...+/- 0.5
Средняя связь	+/- 0.5...+/- 0.7
Сильная связь	+/- 0.7...+/- 0.9
Очень сильная или абсолютная связь	+/- 0.9...+/-1

Коллинеарность и мультиколлинеарность

➔ **Чистая коллинеарность**

Некоторые факторы являются линейно зависимыми между собой. Это ведёт к уменьшению ранга матрицы факторов. Корреляции между зависимыми факторами близки к +1 или -1. Матрица корреляции вырождена.
Такие случаи очень редко встречаются на практике, но если вы таковые заметите, можете смело избавляться от одного из факторов.

➔ **Мультиколлинеарность**

Формально линейной зависимости между факторами нет, и матрица факторов имеет максимальный ранг. Однако корреляции между мультиколлинеарными факторами по-прежнему близки к +1 или -1, и матрица корреляции практически вырождена, несмотря на то что имеет максимальный ранг.

Полиномиальная регрессия

Полином (многочлен) от k переменных $x_1, x_2, ..., x_k$ — это выражение (функция) вида:

$$P(x_1, x_2, ..., x_k) = \sum_I w_i x_1^{i_1} x_2^{i_2} ... x_k^{i_k},$$

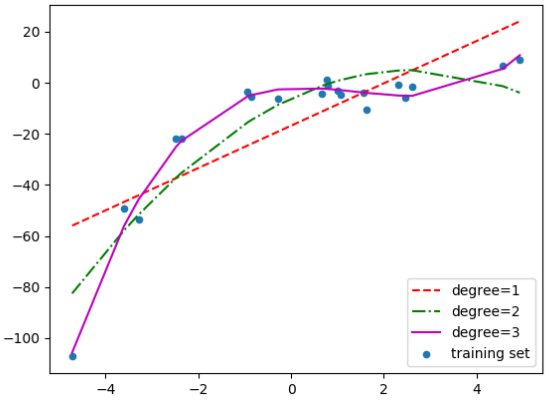
где $I = (i_1, i_2, ..., i_k)$ — набор из k целых неотрицательных чисел (степеней полинома), w_I — числа, называемые коэффициентами полинома.

Когда переменная всего одна, полином будет записываться так:

$$P(x) = \sum_I w_i x^i = w_0 + w_1 x^1 + w_2 x^2 + ... + w_k x^k$$

Полином степени k способен описать абсолютно любую зависимость. Для этого ему достаточно задать набор наблюдений — точек, через которые он должен пройти (либо приблизительно пройти). Вопрос стоит только в степени

этого полинома — k. Например, на рисунке ниже представлены три полинома: первой степени — линейная регрессия, второй степени — квадратичная регрессия и третьей степени — кубическая регрессия.



Цель обучения модели полиномиальной регрессии всё та же, что и для линейной регрессии: найти такие коэффициенты w_i, при которых ошибка между построенной функцией и обучающей выборкой была бы наименьшей из возможных.

Тогда мы хотим, чтобы для полинома степени k (от одной переменной) выполнялась система уравнений:

$$\begin{cases} w_0 + w_1 x + w_2 x^2 + \cdots + w_k x^k = y_1 \\ w_0 + w_1 x + w_2 x^2 + \cdots + w_k x^k = y_2 \\ \cdots \\ w_0 + w_1 x + w_2 x^2 + \cdots + w_k x^k = y_N \end{cases}$$

Обычно количество точек в обучающей выборке N значительно больше, чем степень полинома k, а значит перед нами переопределённая СЛАУ относительно с $k + 1$ неизвестной — w_i. Точных решений у системы практически никогда не будет, но зато мы умеем решать её приближенно:

$$\vec{w} = \left(A^T A\right)^{-1} A^T \vec{y}$$

Для создания полиномиальных признаков в библиотеке sklearn используется класс `PolynomialFeatures`.

Возможна ситуация, когда какие-то сгенерированные полиномиальные факторы могут линейно выражаться через другие факторы. Тогда ранг корреляционной матрицы будет меньше числа факторов и поиск по классическому МНК алгоритму не будет успешным.

В sklearn для решения последней проблемы предусмотрена защита — **использование сингулярного разложения матрицы** A. Однако данная защита не решает проблемы неустойчивости коэффициентов регрессии.

Полиномиальная регрессия очень склонна к переобучению: чем выше степень полинома, тем сложнее модель и тем выше риск переобучения.

Регуляризация

Регуляризация — это способ уменьшения переобучения моделей машинного обучения путём намеренного увеличения смещения модели, чтобы уменьшить её разброс.

Цели регуляризации:

- Предотвратить переобучение модели.
- Включить в функцию потерь штраф за переобучение.
- Обеспечить существование обратной матрицы $\left(A^T A\right)^{-1}$.
- Не допустить огромных коэффициентов модели.

Идея регуляризации состоит в наложении ограничения на вектор весов. Часто также говорят "наложение штрафа за высокие веса". В качестве штрафа принято использовать норму вектора весов.

$$\begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \left(\|\vec{w}\|_{L_p}\right)^p \le b \end{cases}$$

где $\|\vec{w}\|_{L_p}$ — норма вектора порядка $p > 1$, которая определяется как:

$$\|\vec{w}\|_{L_p} = \sqrt[p]{\sum_{i=0}^{k} |w_i|^p}$$

Записанная система эквивалентна:

$$L\left(\vec{w}, \alpha\right) = \|\vec{y} - A\vec{w}\|^2 + \alpha(\|\vec{w}\|_{L_p})^p \to min$$

где $L\left(\vec{w}, \alpha\right)$ — функция Лагранжа, которая зависит не только от вектора весов модели $\vec{w}$, но и от некоторой константы $\alpha \geq 0$ — множителя Лагранжа (коэффициент регуляризации).

L_2-регуляризация

> **L_2-регуляризация (Ridge)**, или **регуляризация по Тихонову** — это регуляризация, в которой порядок нормы $p = 2$.

Тогда оптимизационная задача в случае L_2-регуляризации будет иметь вид:

$$\|\vec{w}\|_{L_2} = \sqrt[2]{\sum_{i=0}^{k} |w_i|^2} = \sqrt{\sum_{i=0}^{k} (w_i)^2}$$

$$\begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \left(\|\vec{w}\|_{L_2}\right)^2 \leq b \end{cases} \leftrightarrow \begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \sum_{i=0}^{k} (w_i)^2 \leq b \end{cases}$$

$$\|\vec{y} - A\vec{w}\|^2 + \alpha \sum_{i=0}^{k} (w_i)^2 \to min$$

У данной задачи даже есть аналитическое решение, полученное математиком Тихоновым:

$$\hat{\vec{w}}_{ridge} = \left(A^T A + \alpha E\right)^{-1} A\vec{y}$$

где E — единичная матрица размера $dim\,(I) = (k + 1, k + 1)$ вида:

$$E = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

В sklearn L_2-регуляризация реализована в классе `Ridge` из модуля linear_model.

Перед использованием рекомендуется произвести стандартизацию/нормализацию данных.

L_1-регуляризация

> **L_1-регуляризацией**, или **Lasso (Least Absolute Shrinkage and Selection Operator)**, называется регуляризация, в которой порядок нормы $p = 1$.

Тогда оптимизационная задача в случае L_1-регуляризации будет иметь вид:

$$\|\vec{w}\|_{L_1} = \sqrt[1]{\sum_{i=0}^{k} |w_i|^1} = \sum_{i=0}^{k} |w_i|$$

$$\begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ (\|\vec{w}\|_1)^1 \le b \end{cases} \leftrightarrow \begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \sum_{i=0}^{k} |w_i| \le b \end{cases}$$

$$\|\vec{y} - A\vec{w}\|^2 + \alpha \sum_{i=0}^{k} |w_i| \to min$$

Можно показать, что данная задача имеет аналитическое решение, однако в реализации sklearn оно даже не заявлено как возможное для использования в связи с нестабильностью взятия производной от функции модуля, поэтому мы не будем его рассматривать.

В sklearn L_1-регуляризация реализована в классе **Lasso**, а заданная выше оптимизационная задача решается алгоритмом координатного спуска (Coordinate Descent).

Перед использованием рекомендуется произвести стандартизацию/нормализацию данных.

Elastic-Net

> **Elastic-Net** — это комбинация из L_1- и L_2-регуляризации.

Идея Elastic-Net состоит в том, что мы вводим ограничение как на норму весов порядка $p = 1$, так и на норму порядка $p = 2$.

Тогда оптимизационная задача будет иметь вид:

$$\begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \left(\|\vec{w}\|_{L_1}\right)^1 \le b_1 \\ \left(\|\vec{w}\|_{L_2}\right)^2 \le b_2 \end{cases} \leftrightarrow \begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \sum_{i=0}^{k} |w_i| \le b_1 \\ \sum_{i=0}^{k} (w_i)^2 \le b_2 \end{cases}$$

$$\|\vec{y} - A\vec{w}\|^2 + \alpha{\cdot}\lambda \sum_{i=0}^{k} |w_i| + \frac{\alpha{\cdot}(1-\lambda)}{2} \sum_{i=0}^{k} (w_i)^2 \to min$$

Аналитического решения у этой задачи нет, поэтому для её решения в sklearn используется координатный спуск, как и для модели Lasso.

В sklearn эластичная сетка реализована в классе `ElasticNet` из пакета с линейными моделями — linear_model. За коэффициент α отвечает параметр `alpha`, а за коэффициент λ — `l1_ratio`.

Перед использованием рекомендуется произвести стандартизацию/нормализацию данных.

Геометрическая интерпретация регуляризации

Задача условной оптимизации:

$$\begin{cases} \|\vec{y} - A\vec{w}\|^2 \to \min \\ \left(\|\vec{w}\|_{L_p}\right)^p \le b \end{cases}$$

геометрически означает поиск минимума функции $L(\vec{w}) = \|\vec{y} - A\vec{w}\|^2 = \sum_{i=1}^{N} (y_i - (\vec{x_i}, \vec{w}))^2$, которая отражает выпуклую поверхность, на пересечении с фигурой, которая образуется функцией $\psi(\vec{w}) = (\|\vec{w}\|_{L_p})^p$, ограниченной некоторым числом b.

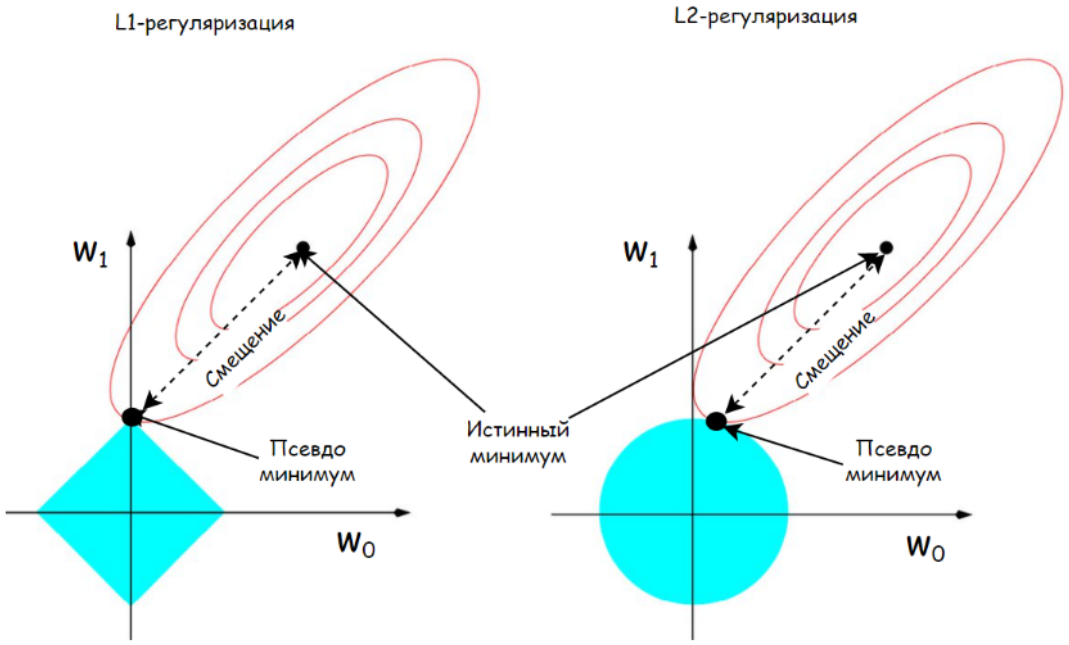
В случае L_1-регуляризации выражение $\sum_{i=0}^{k} |w_i| \le b$ задаёт в пространстве параметров w внутренность ромба с центром в начале координат:

$$|w_0| + |w_1| + \ldots + |w_k| = b \text{ — } \textit{уравнение ромба}$$

В случае L_2-регуляризации выражение $\sum_{i=0}^{k} (w_i)^2 \leq b$ задаёт окружность с центром в начале координат:

$$(w_0)^2 + (w_1)^2 + \ldots + (w_k)^2 = b \text{ — } \textit{уравнение окружности}$$

Рассмотрим случай, когда фактор всего один ($k = 1$), а в уравнении линейной регрессии присутствуют только два параметра w_0 и w_1.



Концентрическими кругами обозначены линии равного уровня функции $L(w)$. Для каждого конкретного набора данных она будет иметь разный вид, но смысл будет тем же. **Голубой** областью обозначены ромб и окружность, которые задаёт L_1- и L_2-норма вектора весов соответственно.

Если бы мы использовали классическую линейную регрессию, то МНК приводил бы нас в точку истинного минимума функции $L(w)$ — в центр, из которого исходят концентрические круги. Это была бы некоторая комбинация параметров w_0 и w_1.

В случае, когда мы используем модель линейной регрессии с регуляризацией, мы будем пытаться найти такую комбинацию w_0 и w_1, которая доставляет минимум функции $L(w)$, но при этом не выходит за границы ромба (или

окружности). Таким образом, вместо истинного минимума мы находим так называемый **псевдоминимум**.

Заметим, что у ромба вероятность коснуться концентрического круга одной из своих вершин больше, чем у окружности — своей верхней/нижней/правой/левой точкой. Точка касания в вершине ромба — это точка, в которой либо $w_0 = 0$, либо $w_1 = 0$. То есть L_1-регуляризация склонна с большей вероятностью занулять коэффициенты линейной регрессии, чем L_2 -регуляризация.

Величина диагонали ромба и радиуса окружности зависят от величины коэффициента регуляризации α: чем больше α, тем меньше ромб/окружность, а значит тем дальше псевдоминимум будет находиться от истинного минимума, и наоборот.