

ОБУЧЕНИЕ МОДЕЛЕЙ ЛИНЕЙНОЙ РЕГРЕССИИ ДЛЯ ПЕРЕМЕННЫХ, СОДЕРЖАЩИХ КОМПЛЕКСНЫЕ ЧИСЛА, С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМОВ SCIKIT-LEARN И НЕЙРОННЫХ СЕТЕЙ KERAS

Актуальность задачи

Комплексные числа широко используются для вычислений в науке, технике, экономических показателях. Примером их использования являются задачи электротехники, радиофизики, схемотехники, обработки сигналов, теории управления, эконометрики и т. д.

Алгоритмы Scikit-learn и Keras не поддерживают работу с комплексными числами, что не позволяет построить непосредственные модели регрессии для данных, содержащих комплексные числа, с помощью стандартных алгоритмов линейной регрессии Scikit-learn или нейронных сетей Keras.

Попытки обойти это препятствие, как правило, сводятся к разделению исходного датасета с комплексными числами на два вещественных датасета, содержащих, соответственно, реальную и мнимую части исходного датасета с последующим построением независимых моделей регрессии для каждого выделенного датасета. Окончательная модель для прогноза целевой переменной, являющейся комплексным числом, представляет собой объединение прогнозов обеих моделей регрессии для реальной и мнимой частей в одно комплексное число.

Такой подход к решению этой задачи не имеет математического обоснования, так как в общем случае значение реальной части целевой переменной зависит от мнимой части исходного датасета и, наоборот, значение мнимой части целевой переменной зависит от реальной части исходного датасета. Очевидно, что при описанном подходе такие зависимости игнорируются, что приводит к недостоверным результатам.

Таким образом возникает актуальная задача обоснованного преобразования исходного датасета с комплексными числами в датасет с вещественными числами, учитывающего взаимные зависимости реальных и мнимых частей целевой переменной и исходного датасета, что позволит для построения **достоверной** модели регрессии комплексных чисел воспользоваться стандартными алгоритмами линейной регрессии Scikit-learn или нейронными сетями Keras.

Решение задачи

В общем случае операторное уравнение для определения неизвестных параметров регрессии имеет следующий вид (подчеркивание означает, что величина относится к множеству \mathbb{C} комплексных чисел)

$$\underline{A} \underline{z} = \underline{b}, \quad (1)$$

где \underline{A} – заданный оператор значений признаков;

\underline{z} – неизвестная функция параметров регрессии;

\underline{b} – заданная функция значений целевой переменной.

В нашем случае значения признаков и целевой переменной заданы в виде датасета размером $M \times N$ и, следовательно, оператор \underline{A} является матрицей заданных значений признаков, а функция \underline{b} – вектором заданных значений целевой переменной. При этом матрица \underline{A} имеет размер $M \times (N - 1)$, а вектор \underline{b} имеет размер $M \times 1$.

Представим операторное уравнение (1) в следующем виде

$$(A_x + jA_y)(z_x + jz_y) = b_x + jb_y, \quad (2)$$

где индексами x и y обозначены реальные и мнимые части комплексных величин, соответственно.

Предположим, что оператор $\underline{A} = A_x + jA_y$ является линейным. Тогда уравнение (2) можно записать в следующем виде

$$A_x z_x - A_y z_y + j(A_y z_x + A_x z_y) = b_x + jb_y. \quad (3)$$

Далее разделяя действительную и мнимую части (3), получаем систему из двух вещественных операторных уравнений

$$\begin{cases} A_x z_x - A_y z_y = b_x \\ A_y z_x + A_x z_y = b_y \end{cases}, \quad (4)$$

которую можно представить в компактной форме

$$Az = b, \quad (5)$$

где A – расширенная вещественная матрица размером $2M \times 2(N - 1)$, элементами которой являются вещественные матрицы A_x и A_y заданных значений признаков

$$A = \begin{pmatrix} A_x & -A_y \\ A_y & A_x \end{pmatrix}; \quad (6)$$

z – расширенный вещественный вектор размером $2(N - 1) \times 1$, элементами которого являются векторы z_x и z_y неизвестных параметров регрессии

$$z = \begin{pmatrix} z_x \\ z_y \end{pmatrix}; \quad (7)$$

b – расширенный вещественный вектор размером $2M \times 1$, элементами которого являются векторы b_x и b_y заданных значений целевой переменной

$$b = \begin{pmatrix} b_x \\ b_y \end{pmatrix}. \quad (8)$$

Таким образом модель регрессии комплексных чисел, параметры которой определяются операторным уравнением (1), сводится к модели регрессии вещественных чисел, параметры которой определяются уравнением (5). Для обучения этой модели регрессии вещественных чисел необходимо вместо исходного датасета размером $M \times N$ с комплексными числами использовать преобразованный датасет размером $2M \times (2N - 1)$ с вещественными числами, полученный путем объединения вещественных матрицы A и вектора b .

Прогноз b_{pred} целевой переменной для некоторого заданного вектора размером $(N - 1) \times 1$ комплексных значений признаков $\underline{a}_{pred} = a_{pred_x} + ja_{pred_y}$ описывается операторным выражением

$$b_{pred} = A_{pred}^T z, \quad (9)$$

где A_{pred} – матрица размером $2(N - 1) \times 2$

$$A_{pred} = \begin{pmatrix} a_{pred_x} & a_{pred_y} \\ -a_{pred_y} & a_{pred_x} \end{pmatrix}. \quad (10)$$

Для получения прогноза b_{pred} целевой переменной необходимо в обученную модель передать матрицу A_{pred}^T (верхний индекс T означает транспонирование матрицы). Результат b_{pred} прогноза целевой переменной будет представлен в виде вектора размером 2×1 , в котором первый его элемент является реальной частью комплексного числа \underline{b}_{pred} , а второй элемент – мнимой частью \underline{b}_{pred} .

Выводы

1. Алгоритмы библиотек Scikit-learn и Keras не поддерживают работу с комплексными числами, что не позволяет построить непосредственные модели линейной регрессии для данных, содержащих комплексные числа, с помощью стандартных алгоритмов линейной регрессии Scikit-learn или нейронных сетей Keras.

2. Разработан метод преобразования исходного датасета размером $M \times N$ с комплексными числами в датасет размером $2M \times (2N - 1)$ с вещественными числами, что позволяет использовать для построения **достоверных** моделей регрессии комплексных чисел стандартные алгоритмы линейной регрессии Scikit-learn или нейронные сети Keras.