

У `.py` файлі, що додається до поточного звіту, реалізовано функцію генерації даних `data_gen`, що приймає на вхід

- двовимірний масив  $\mathbf{X}$  з ознаками (кожний стовпчик –  $n$  вимірів одного атрибута);
- вектор-стовпчик параметрів  $a$ ;
- `string` з назвою розподілу бажаного випадкового шуму: `"normal"` для нормального розподілу, `"laplace"` для лавласівського, все інше для узагальненого нормального розподілу;
- параметри розподілу в порядку: математичне очікування, дисперсія, а також  $\beta$  (для узагальненого нормального розподілу).

Функція повертає двовимірний масив  $y$  розмірності  $n \times 1$  зі значеннями відгуку, згенерованими за формулою

$$y = \sum_{i=0}^{d+1} a_i x_i + noise, \quad x_0 := 1.$$

Також, згідно з ТЗ реалізовано функцію `calcfg_Lp_criterion`, що за заданими значеннями ознак, відгуків та параметрів обчислює значення функції максимальної правдоподібності, МНМ для  $p = 1$ , МНК для  $p = 2$ ,  $L_p$ -критерій для інших ( $p \in (1, 2)$ ).

Для випробувань був використаний датасет з даними студентської успішності, що містить 5 ознак: 4 числових та одну бінарну, яку було конвертовано в 0/1 для зручності; та один стовпчик зі значеннями відгуку. Нижче представлені вибрані результати застосування метода еліпсоїдів. Як видно, за умови вірного вибору початкового радіуса  $r_0$  функція правдоподібності дійсно спадає для різних значень  $p$ .

Для порівняння на синтетичних даних згенеровано масив  $Y_{gen}$  за допомогою `data_gen`, та заготовлено стандартну функцію з `sklearn.linear_model`, проте повноцінного порівняння поки що провести не вдалося, отримані досі результати не піддаються інтерпретації. Якщо можливо, буду намагатися надіслати це доповнення пізніше.

```

In [192]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 1.5, w, 0.1, 1e-6, int(1e6), int(1e2))
itn = 0, f = 2268298.645297049
itn = 100, f = 2155089.797477003
itn = 200, f = 1917676.7089531904
itn = 300, f = 1043802.7915562693
itn = 400, f = 1004405.8083110831
itn = 500, f = 1005441.3585710251
itn = 600, f = 1005711.574497945
Result:
ist = 1, itn = 651, f = 1004434.8896942697

In [193]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 1.5, w, 0.2, 1e-6, int(1e6), int(1e2))
itn = 0, f = 2268298.645297049
itn = 100, f = 2025266.6224191634
itn = 200, f = 945329.7466169778
itn = 300, f = 905415.6638736197
itn = 400, f = 897335.3676682153
itn = 500, f = 898854.1791114621
itn = 600, f = 899054.2030789517
Result:
ist = 1, itn = 654, f = 898908.3487028824

```

```

In [173]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 1, w, 0.1, 1e-6, int(1e6), int(1e2))
itn = 0, f = 1870846.290501977
itn = 100, f = 1836431.2294447722
itn = 200, f = 1810919.2693693594
itn = 300, f = 1818688.257318819
itn = 400, f = 1784141.6346546859
itn = 500, f = 1775817.43449257
itn = 600, f = 1742537.8226520931
Result:
ist = 1, itn = 649, f = 1739876.8439787908

In [174]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 1, w, 0.4, 1e-6, int(1e6), int(1e2))
itn = 0, f = 1870846.290501977
itn = 100, f = 1745476.6230416354
itn = 200, f = 1701814.3522172053
itn = 300, f = 1645690.4630737214
itn = 400, f = 1471990.5385653463
itn = 500, f = 1369082.0002373157
itn = 600, f = 1213524.7506980742
itn = 700, f = 1050646.4942359778
Result:
ist = 1, itn = 724, f = 1028616.4426997621

```

```

In [168]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 2, w, 0.1, 1e-6, int(1e6), int(1e2))
itn = 0, f = 387802917.74314225
itn = 100, f = 374192228.51188827
itn = 200, f = 364212490.6914169
itn = 300, f = 367373636.53491443
itn = 400, f = 353947640.7001221
itn = 500, f = 350825733.0841676
itn = 600, f = 338201260.0927536
Result:
ist = 1, itn = 677, f = 336676259.3653164

In [169]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 2, w, 0.5, 1e-6, int(1e6), int(1e2))
itn = 0, f = 387802917.74314225
itn = 100, f = 308886991.2919237
itn = 200, f = 293450444.2084014
itn = 300, f = 291619549.19411093
itn = 400, f = 201840401.36078766
itn = 500, f = 173143722.6965632
itn = 600, f = 125342490.72261415
itn = 700, f = 90736026.39364864
Result:
ist = 1, itn = 749, f = 57597258.536619104

```

```

In [171]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 2, w, 1, 1e-6, int(1e6), int(1e2))
itn = 0, f = 387802917.74314225
itn = 100, f = 272156549.24792403
itn = 200, f = 232190879.61520392
itn = 300, f = 189691714.48020503
itn = 400, f = 96844949.13166144
itn = 500, f = 36349435.09357659
itn = 600, f = 6558215.748408149
itn = 700, f = 15527603.649955261
itn = 800, f = 4256197985.3942842
Result:
ist = 1, itn = 816, f = 3263396098.8911095

In [172]: w1, f1, itn1, ist1, w_values1, f_values1 = elipsoid_method(calcfg_Lp_criterion, Xtest,
Ytest, 2, w, 0.7, 1e-6, int(1e6), int(1e2))
itn = 0, f = 387802917.74314225
itn = 100, f = 305380684.90988195
itn = 200, f = 266710010.3139265
itn = 300, f = 255062743.75576916
itn = 400, f = 172215263.1138586
itn = 500, f = 135970828.57689074
itn = 600, f = 69677201.2220161
itn = 700, f = 24117940.328227464
Result:
ist = 1, itn = 762, f = 12966313.227803808

```