1. **已经跑出来的算法**
   1. **单目标-线性模型**

|  |  |
| --- | --- |
| 算法 | 算法改进 |
| CoDE |  |
| CoDE\_toZero | 绝对值小于1的置为0 |
| CoDE\_10p\_toZero | 绝对值小于当前染色体最大绝对值的10%置为0. |
| CoDE\_20p\_toZero, | 绝对值小于当前染色体最大绝对值的20%置为0. |
| CoDE\_10p\_lr\_toZero, | 对于正数，小于当前染色体最大正数的10%置为0.；对于负数，大于当前染色体最小负数的10%置为0 |
| CoDE\_20p\_lr\_toZero, | 对于正数，小于当前染色体最大正数的20%置为0.；对于负数，大于当前染色体最小负数的20%置为0 |

* 1. **单目标-非线性模型(MLP)**

|  |  |
| --- | --- |
| 算法 | 算法改进 |
| CoDE |  |
| CoDE\_toZero | 绝对值小于1的置为0 |
| CoDE\_10p\_toZero | 绝对值小于当前染色体最大绝对值的10%置为0. |
| CoDE\_20p\_toZero, | 绝对值小于当前染色体最大绝对值的20%置为0. |
| CoDE\_10p\_lr\_toZero, | 对于正数，小于当前染色体最大正数的10%置为0.；对于负数，大于当前染色体最小负数的10%置为0 |
| CoDE\_20p\_lr\_toZero, | 对于正数，小于当前染色体最大正数的20%置为0.；对于负数，大于当前染色体最小负数的20%置为0 |

* 1. **多目标-线性模型**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 优化目标\算法 | NSGA2 | NSGA2-toZero | NSGA2-10p-toZero | NSGA2-20p-toZero | NSGA2-30p-toZero |
| FPA, AAE | √ | √ | √ | √ | √ |
| FPA, NNZ | √ | √ | √ | √ | √ |
| FPA, AAE, NNZ | √ | √ | √ | √ | √ |
| FPA, MSE | √ | √ | √ | √ | √ |
| FPA, MSE, NNZ | √ | √ | √ | √ | √ |
| FPA, L1 | √ | √ | √ | √ | √ |
| FPA, L1, AAE | √ | √ | √ | √ | √ |
| FPA, L1, MSE | √ | √ | √ | √ | √ |

* 1. **多目标-非线性模型(MLP)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 优化目标\算法 | NSGA2 | NSGA2-toZero | NSGA2-10p-toZero | NSGA2-20p-toZero | NSGA2-30p-toZero |
| FPA, AAE | √ | √ | √ | √ | √ |
| FPA, NNZ | √ | √ | √ | √ | √ |
| FPA, AAE, NNZ | √ | √ | √ | √ | √ |
| FPA, MSE | √ | √ | √ | √ | √ |
| FPA, MSE, NNZ | √ | √ | √ | √ | √ |

* 1. **传统机器学习算法**

|  |  |
| --- | --- |
| 算法 | 参数 |
| LinearRegression | 默认参数 |
| RidgeCV | 留一法，alpha = list(np.linspace(start=0.1, stop=1000, num=10000)) |
| LassoLarsCV | cv=10 |
| MLPRegressor | 三层,Hidden\_node = 3，隐藏层激活函数sigmoid函数,max\_iter=1000 |

1. **实验结果**
   1. **单目标算法**

**绘图结果见【plot/单目标算法对比】**

* + 1. **单目标算法-linear**

(1)结合CoDE\_toZero的算法获得相对较高的FPA，

(2)结合CoDE\_20p\_toZero的算法获得相对较低的MSE和NNZ

* + 1. **单目标算法-mlp**

差别不大，其中:

(1)结合CoDE\_10p\_lr\_toZero的算法获得相对较高的FPA，

(2)结合CoDE的算法获得相对较低的MSE

(3)结合CoDE\_20p\_toZero的算法获得相对较低NNZ

* + 1. **单目标算法-linear-vs-mlp**

以linear模型为基础建模，获得更优的FPA和较差的MSE；

以MLP模型为基础建模，获得较差的FPA和较优的MSE

* 1. **多目标算法**
     1. **多目标算法-linear**

1. 总体来说，FPA+L1+MSE获性能更优，FPA+NNZ+MSE获得的性能相对较优,偶尔有FPA+MSE,FPA+NNZ更优的情况。**（见【plot/多目标/linear/同一算法不同优化目标】）**
2. 总体来说，感觉不同数据集不同的算法会有更优的结果，往往置零法结果更好。（见【多目标/linear/同优化一目标不同算法】）
   * 1. **多目标算法-mlp**
3. 总体来说，FPA+L1+MSE获性能更优，FPA+NNZ+MSE获得的性能相对较优,偶尔有FPA+MSE,FPA+NNZ更优的情况。（**见【plot/多目标/mlp/同一算法不同优化目标】）**
4. 总体来说，感觉不同数据集不同的算法会有更优的结果，往往置零法结果更好。**（见【plot/多目标/mlp/同优化一目标不同算法】）**
   * 1. **多目标算法-linear-vs-mlp**
5. 对比图显示，线性模型的整体FPA更优。需要注意的是，在训练集上，MLP表现出更好的性能，但是在测试集上性能变差。结合mlp模型通过置零法最后模型简化效果不好(参数并没有减少很多)，考虑置零法是否可以进行调整。（**见【plot/多目标/linear\_vs\_mlp/】）**

**2.3 单目标vs多目标**

**见【plot/单目标\_vs\_多目标/】**

线性模型和非线性模型的优化三个目标尤其是FPA+L1+MSE的结果要优于CoDE的单目标模型。

1. **反思与改进**
   1. **置零法的改进：**

(原算法：置零时，小于最大值的某一个比例r，置为0，或者小于某一个确定数置为0)

有两种思路:

1. 随机法: 对于子代的一个染色体，对其每一个基因，设定一个置零率pz，随机一个数，小于pz时，置为0。将生成的新的染色体与原子代染色体进行对比，更优的个体进入下一代。
2. 进化法：将比例r当成染色体的一条基因，跟随种群的迭代而进行优化
   1. **MLP的改进：**

原MLP：固定了模型的架构，即所有模型都具有一层隐藏层，隐藏层具有三个隐藏结点，隐藏层激活函数是sigmoid函数，输出为直接输出。

改进方法：将激活函数选择，隐藏结点个数都作为染色体的一个基因跟随种群进化