# 实验与分析

## 实验目的

无

2017.12.4

Forest CoverType

# 数据集介绍

特征维度：54
10+40+4
数量：581012

特征维度：47
10+35+2
数量：487113

Elevation / Elevation in meters
Aspect / Aspect in degrees azimuth
Slope / Slope in degrees
Horizontal_Distance_To_Hydrology /Horz Dist to nearest surface water features
Vertical_Distance_To_Hydrology / Vert Dist to nearest surface water features
Horizontal_Distance_To_Roadways / Horz Dist to nearest roadway
Hillshade_9am / quantitative / 0 to 255 index / Hillshade index at 9am, summer solstice
Hillshade_Noon / Hillshade index at noon, summer soltice
Hillshade_3pm / Hillshade index at 3pm, summer solstice
Horizontal_Distance_To_Fire_Points /Horz Dist to nearest wildfire ignition points
Wilderness_Area /Wilderness area designation
Soil_Type / Soil Type designation
Cover_Type (7 types)

LR---acc:75.2

RF---acc:96.1

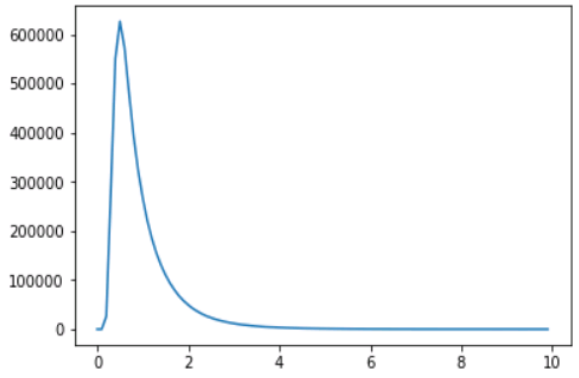
SUSY Data

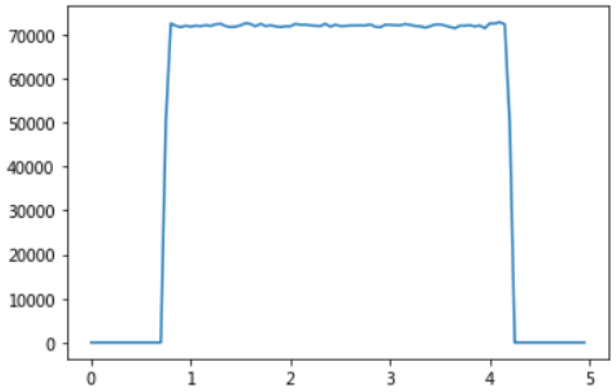
Classification： distinguish between a signal process which produces supersymmetric particles and a background process which does not.

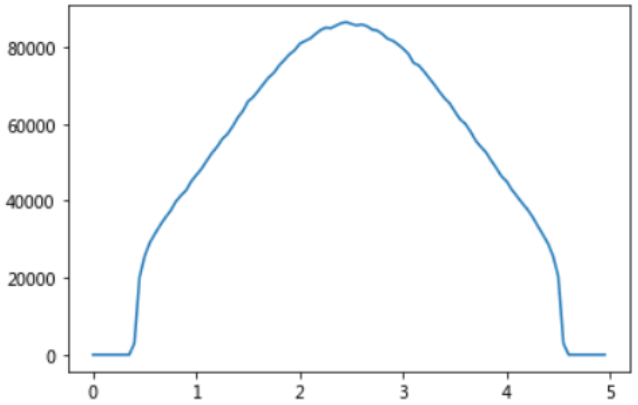
特征维度：54
数量：581012

RF:
Acc:78.18
Auc:84.89
LR:
Acc:76.85
Auc:82.97

18 features (8 low-level features then 10 high-level features):: lepton 1 pT, lepton 1 eta, lepton 1 phi, lepton 2 pT, lepton 2 eta, lepton 2 phi, missing energy magnitude, missing energy phi, MET_rel, axial MET, M_R, M_TR_2, R, MT2, S_R, M_Delta_R, dPhi_r_b, cos(theta_r1)。



HIGGS

Classification: distinguish between a signal process which produces Higgs bosons and a background process which does not.

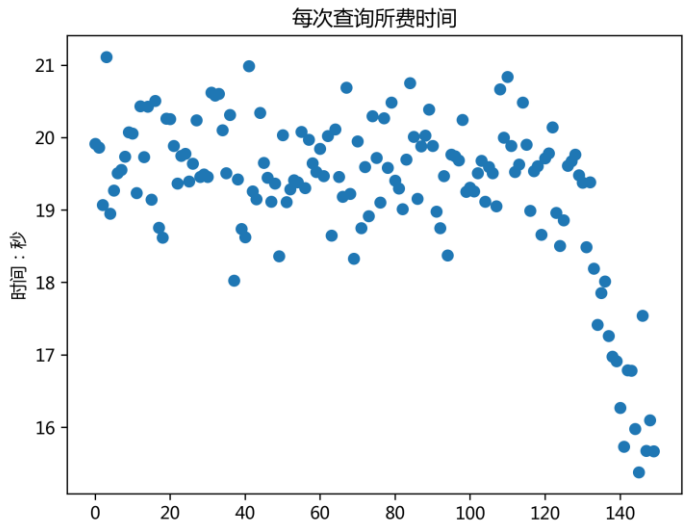
The first 21 features (columns 2-22) are kinematic properties measured by the particle detectors in the accelerator. The last seven features are functions of the first 21 features;

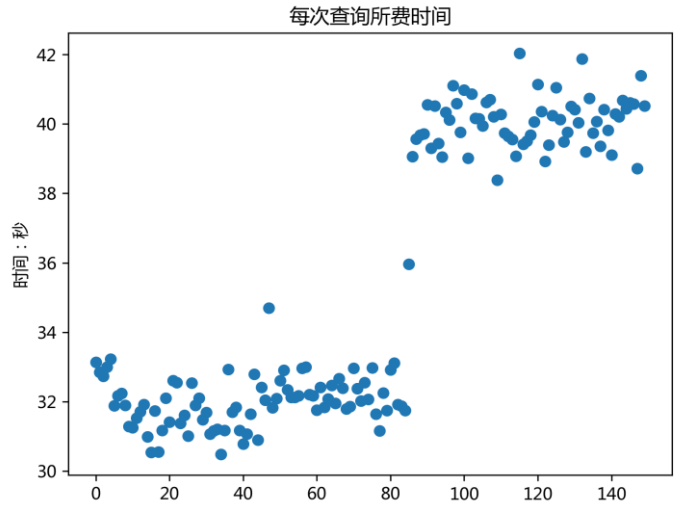
数量：11000000
特征：28维（21+7）

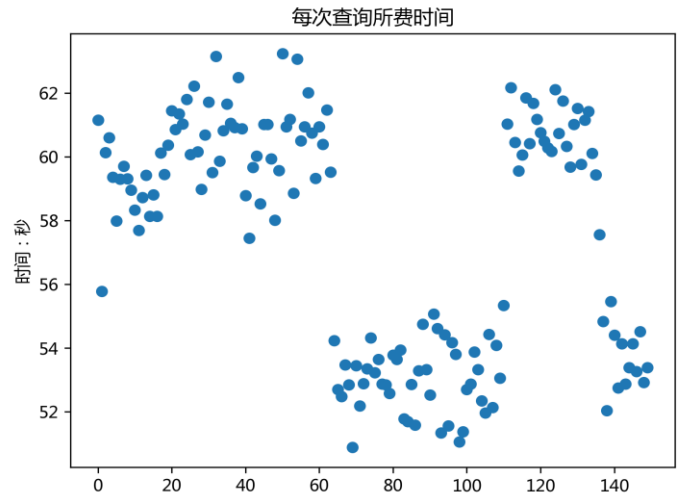
TABLE II: **Performance comparison for the SUSY benchmark.** Each model was trained five times with different weight initializations. The mean AUC is shown with standard deviations in parentheses as well as the expected significance of a discovery (in units of Gaussian $\sigma$) for 100 signal events and $1000 \pm 50$ background events.

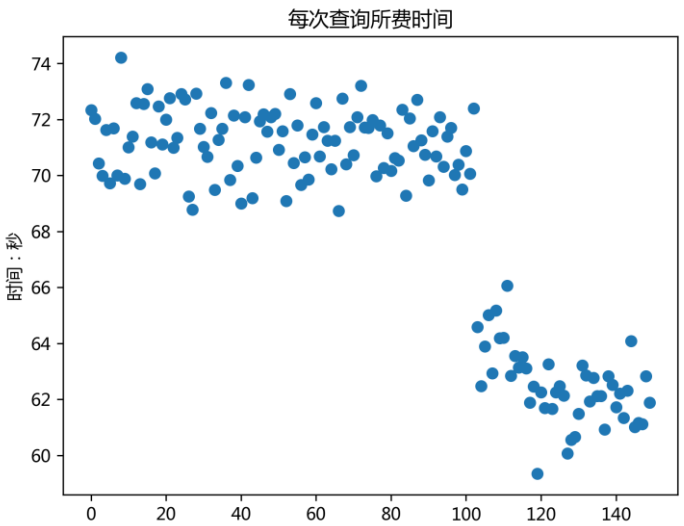
| | AUC | | |
|---|---|---|---|
| Technique | Low-level | High-level | Complete |
| BDT | 0.850 (0.003) | 0.835 (0.003) | 0.863 (0.003) |
| NN | 0.867 (0.002) | 0.863 (0.001) | 0.875 ($< 0.001$) |
| $\text{NN}_{dropout}$ | 0.856 ($< 0.001$) | 0.859 ($< 0.001$) | 0.873 ($< 0.001$) |
| DN | 0.872 (0.001) | 0.865 (0.001) | 0.876 ($< 0.001$) |
| $\text{DN}_{dropout}$ | 0.876 ($< 0.001$) | 0.869 ($< 0.001$) | 0.879 ($< 0.001$) |
| | Discovery significance | | |
| Technique | Low-level | High-level | Complete |
| NN | $6.5\sigma$ | $6.2\sigma$ | $6.9\sigma$ |
| DN | $7.5\sigma$ | $7.3\sigma$ | $7.6\sigma$ |

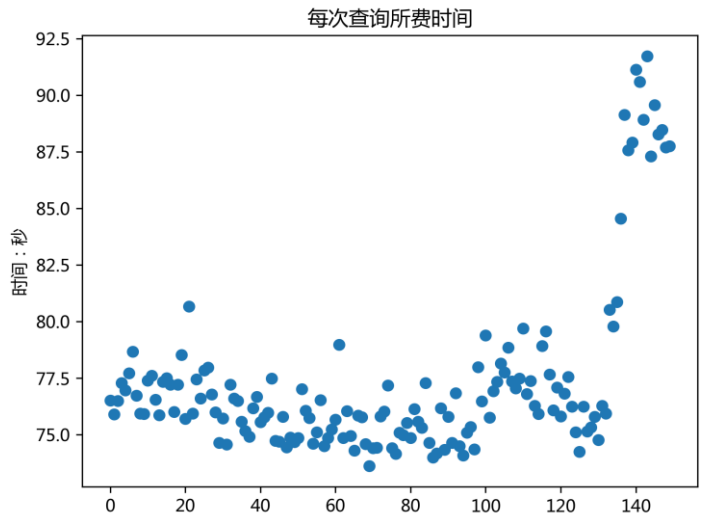
# Linear 时间（10 20 30 40 50万）

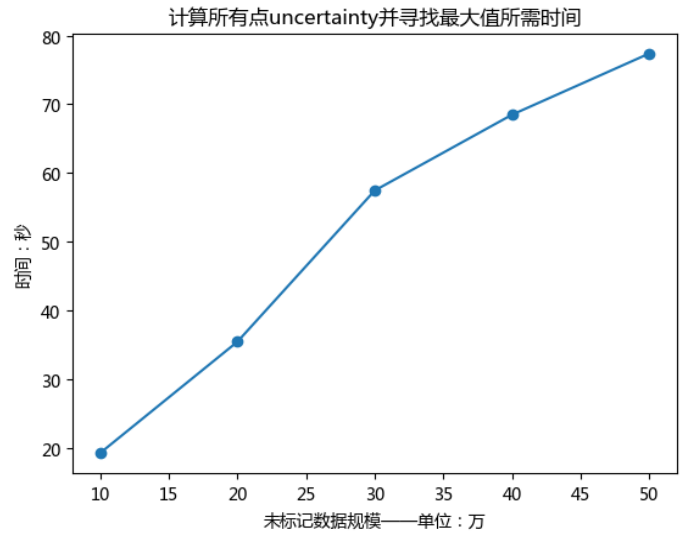


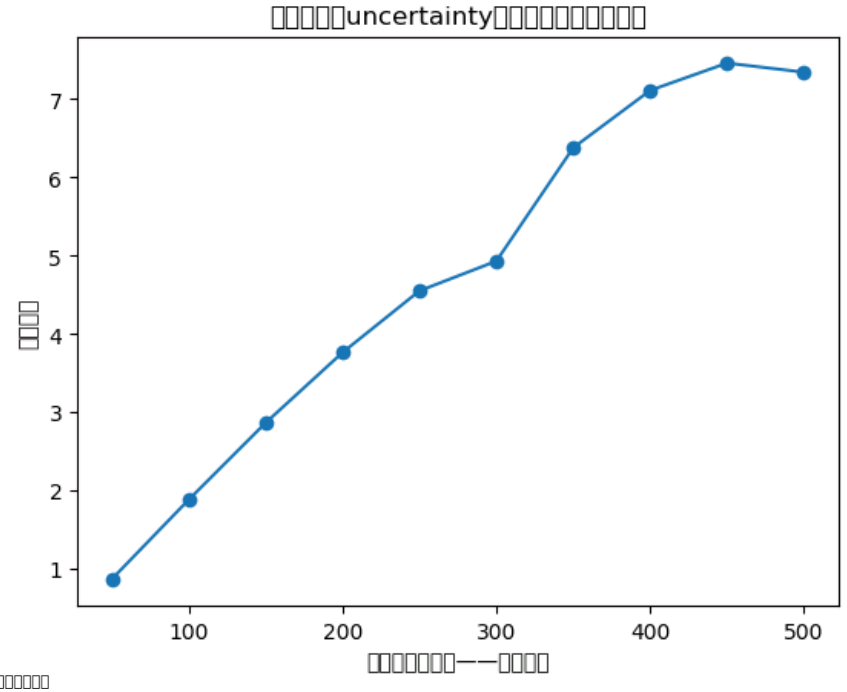



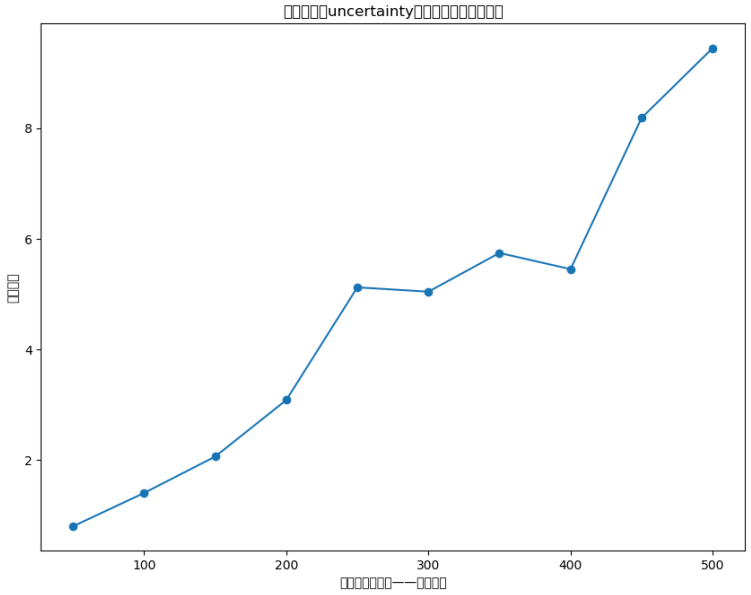



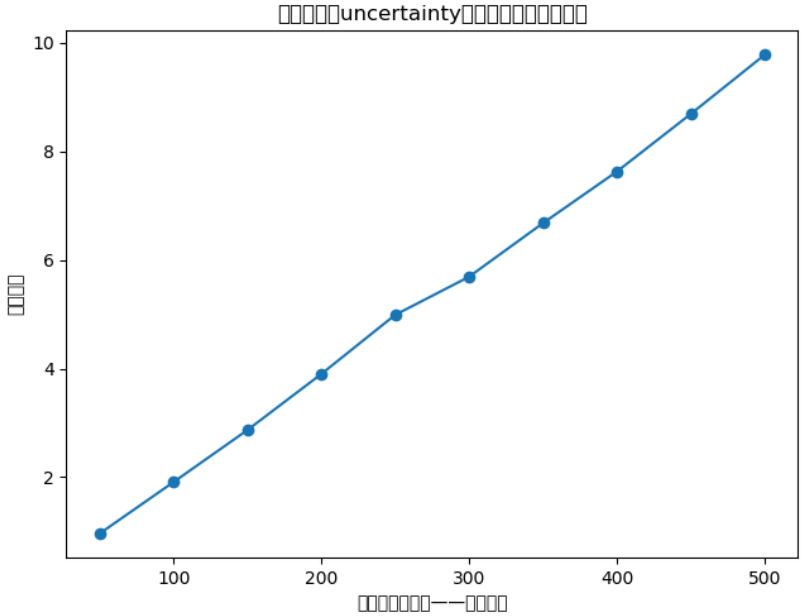



Linear:查询时间

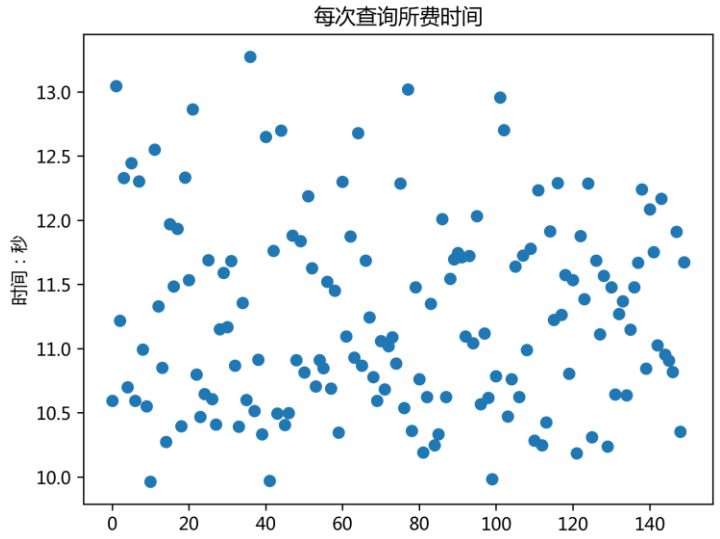
随规模增长关系

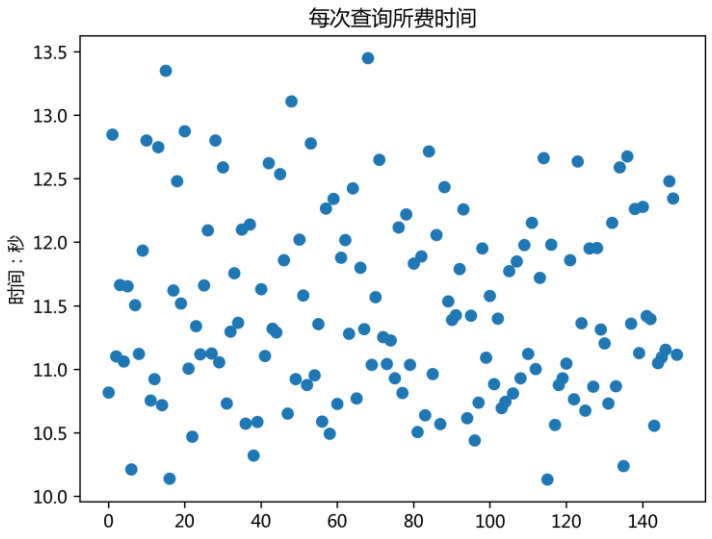


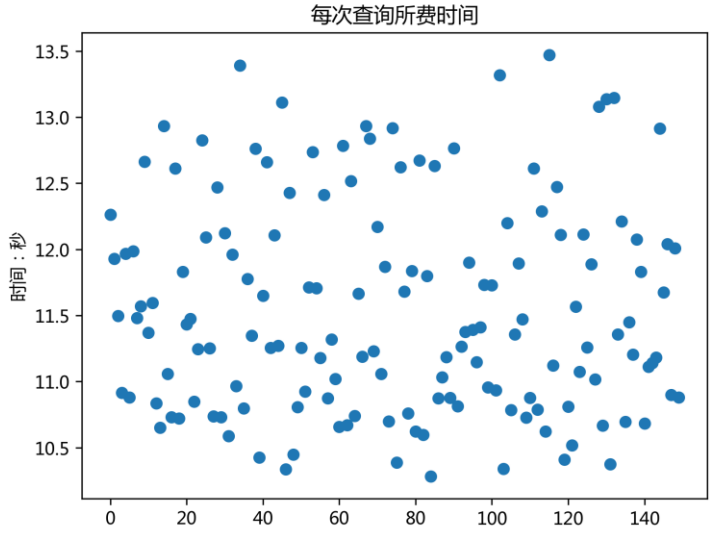



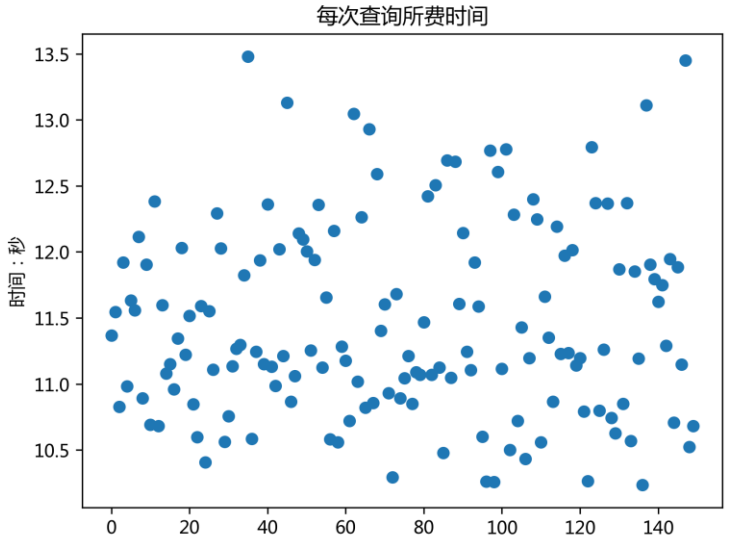



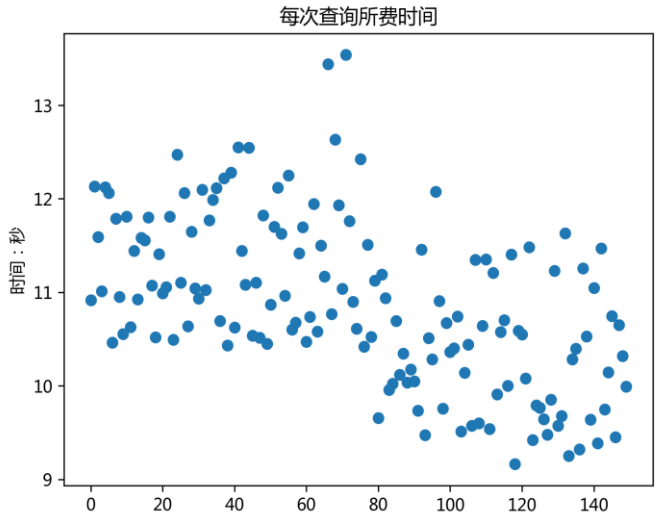
# Net_查询时间


每次查询所费时间
时间：秒
13.0
12.5
12.0
11.5
11.0
10.5
10.0
0
20
40
60
80
100
120
140

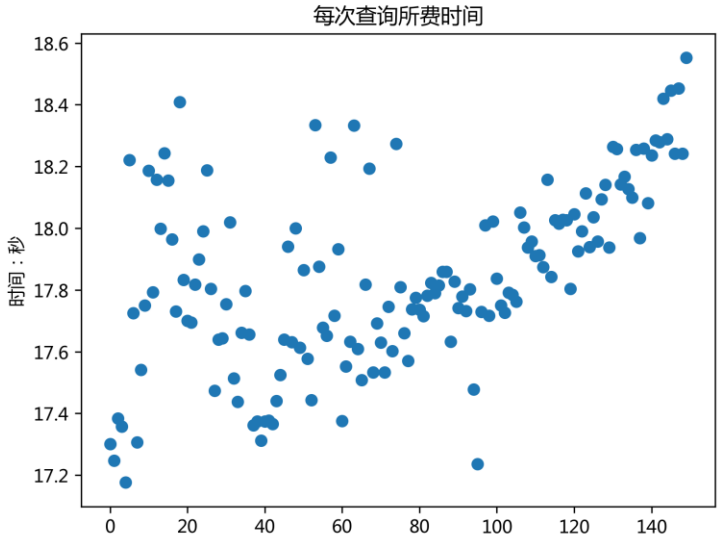

每次查询所费时间
时间：秒
13.5
13.0
12.5
12.0
11.5
11.0
10.5
10.0
0
20
40
60
80
100
120
140

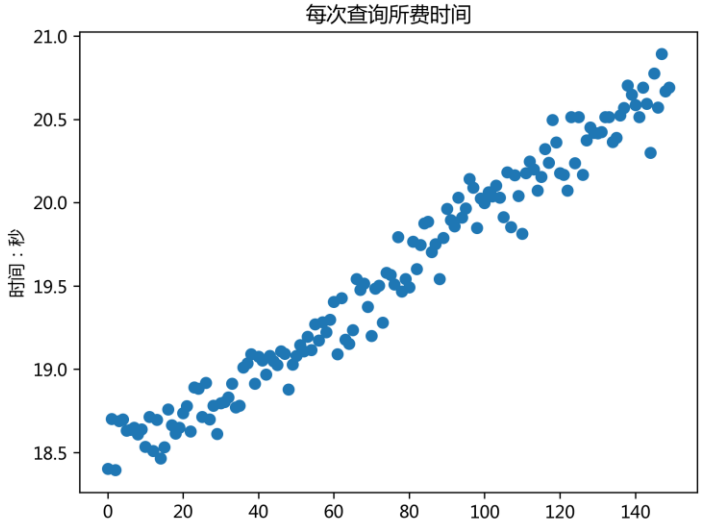

每次查询所费时间
时间：秒
13.5
13.0
12.5
12.0
11.5
11.0
10.5
0
20
40
60
80
100
120
140

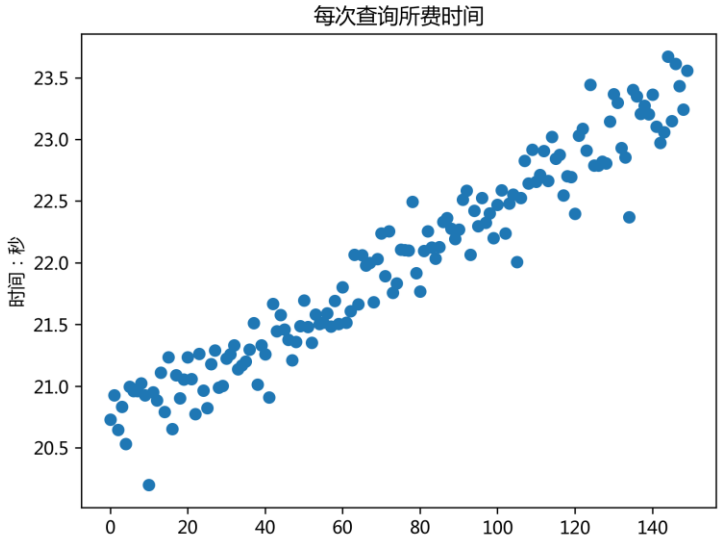

每次查询所费时间
时间：秒
13.5
13.0
12.5
12.0
11.5
11.0
10.5
0
20
40
60
80
100
120
140

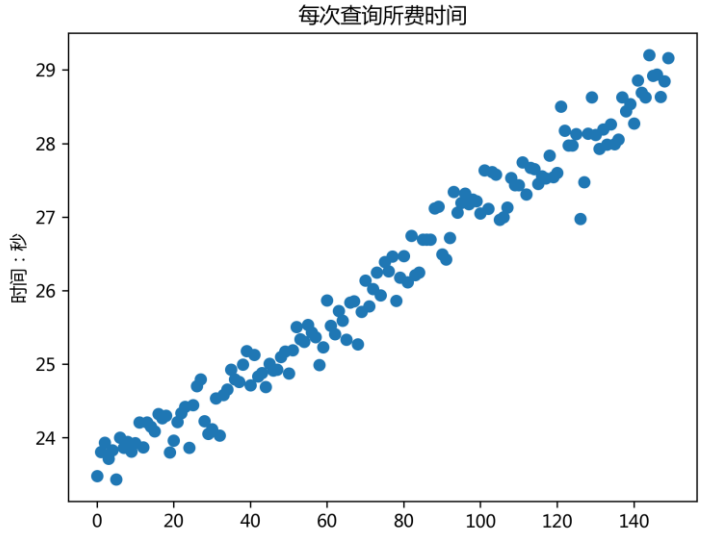

每次查询所费时间
时间：秒
13
12
11
10
9
0
20
40
60
80
100
120
140

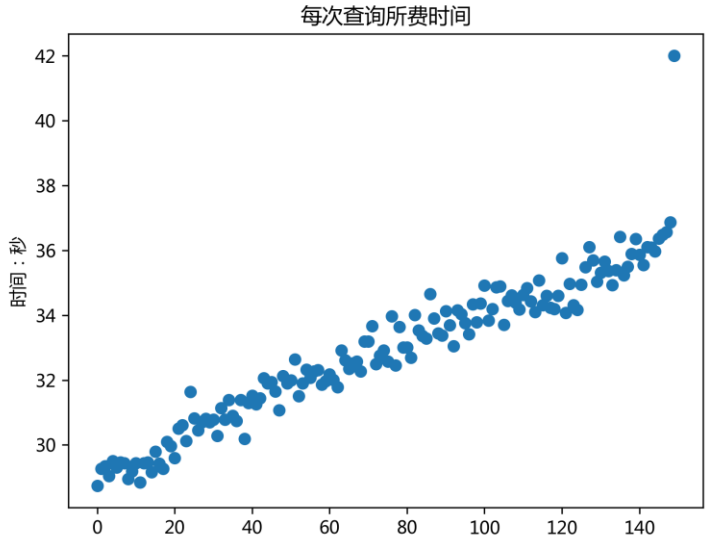
Net_查询时间


每次查询所费时间
时间：秒
18.6
18.4
18.2
18.0
17.8
17.6
17.4
17.2
0
20
40
60
80
100
120
140

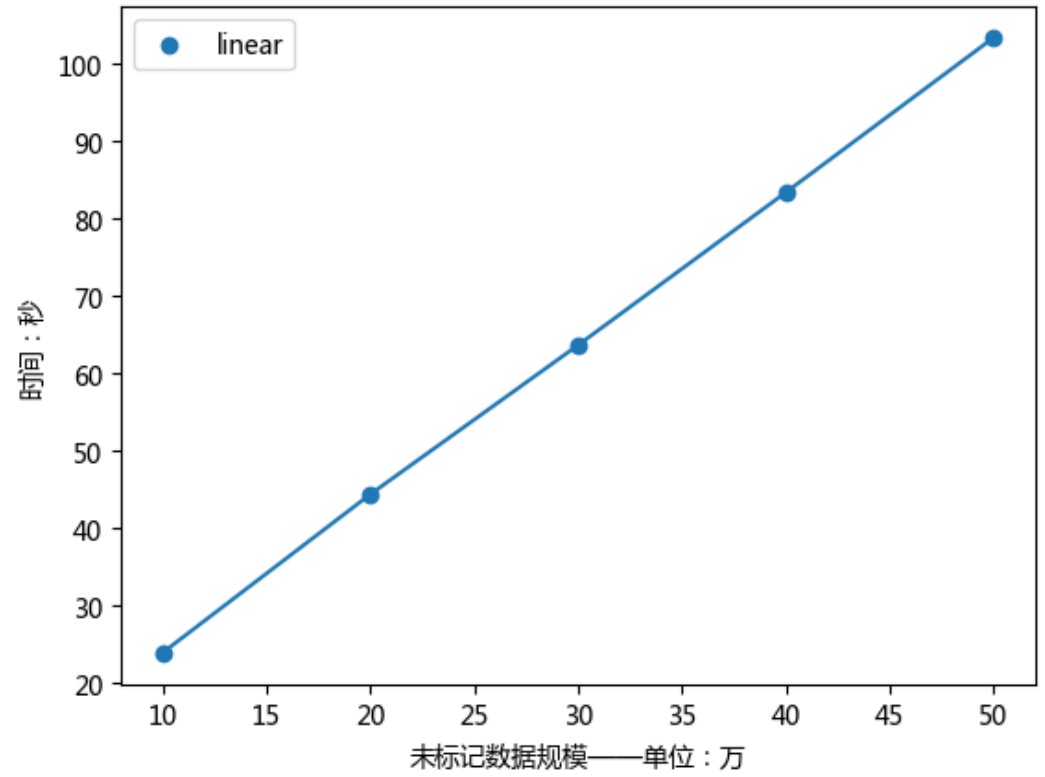

每次查询所费时间
时间：秒
21.0
20.5
20.0
19.5
19.0
18.5
0
20
40
60
80
100
120
140

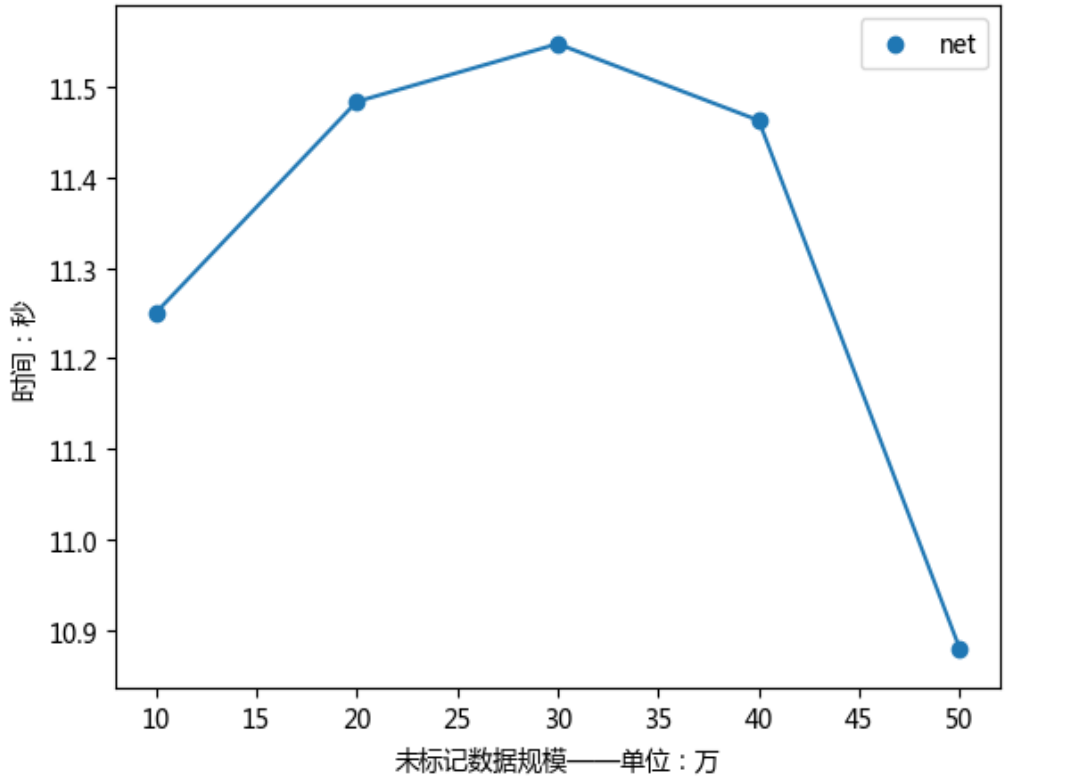

每次查询所费时间
时间：秒
23.5
23.0
22.5
22.0
21.5
21.0
20.5
0
20
40
60
80
100
120
140


每次查询所费时间
时间：秒
29
28
27
26
25
24
0
20
40
60
80
100
120
140


每次查询所费时间
时间：秒
42
40
38
36
34
32
30
0
20
40
60
80
100
120
140

# 查询一个点所需平均时间



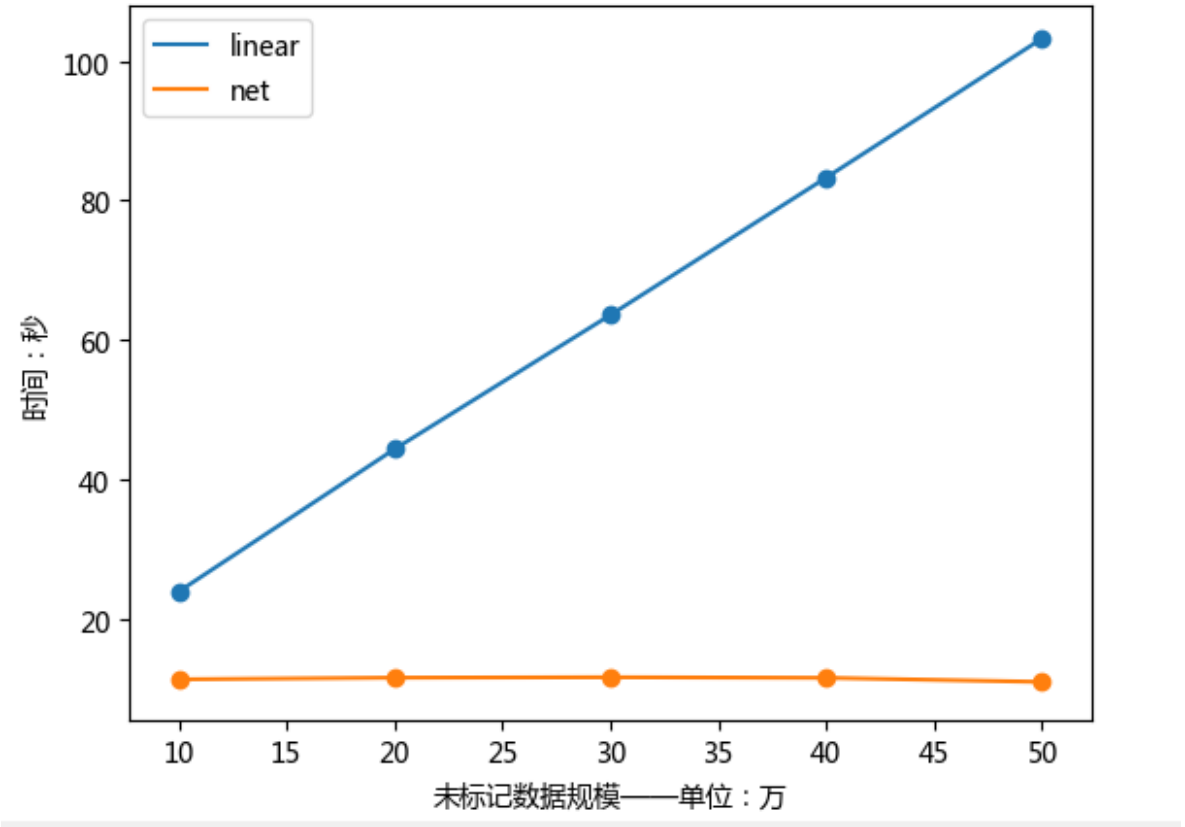



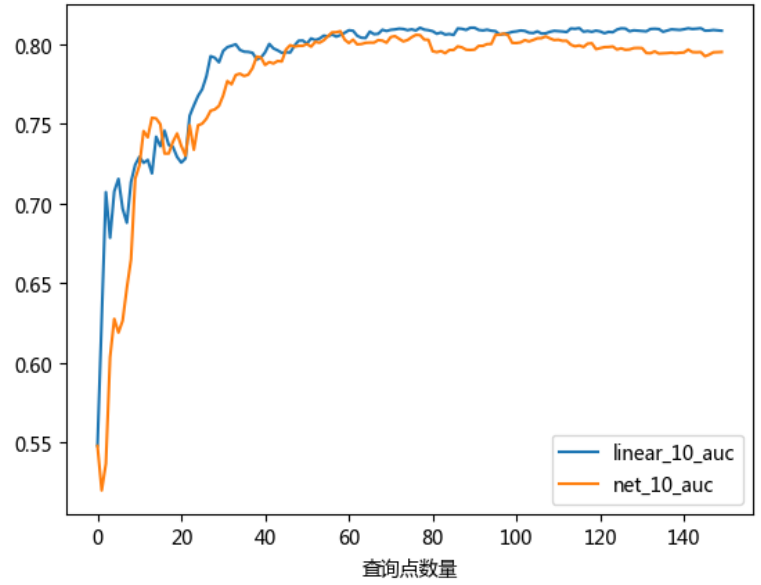
```python
def max_uncer(coef, intercept, train_x):
    max = 1
    id = -1
    coef = coef.flatten()
    for i in range(len(train_x)):
        mid = np.dot(train_x.iloc[i], coef) + intercept
        p1 = mid / (1 + mid)
        p2 = 1 / (1 + mid)
        # uncer = -1 * np.multiply(p1, np.log(p1)) - np.multiply(p2, np.log(p2))
        uncer = (p1-p2)*(p1-p2)
        if uncer < max:
            max = uncer
            id = i
    return id
```

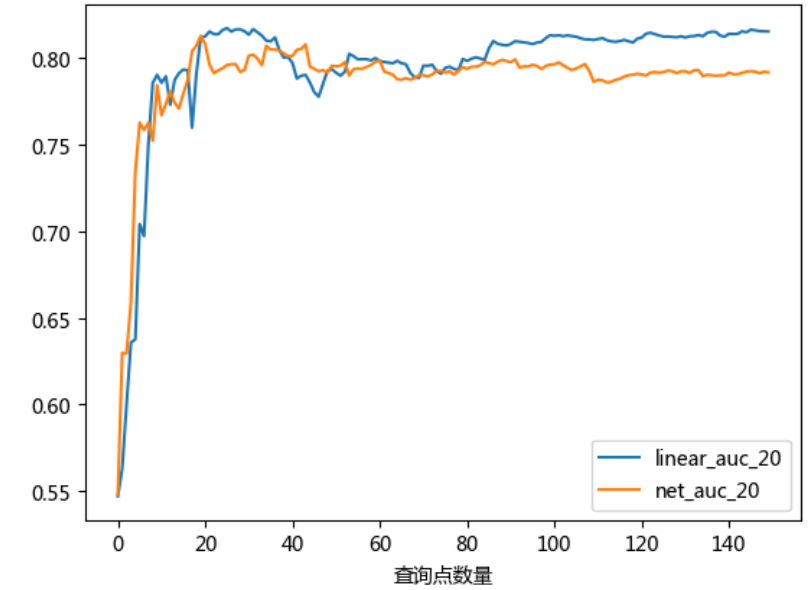
```python
start1 = time.clock()
for i in range(num_train):
    x_data = np.random.normal(loc=0, scale=1, size=[300, 4])
    sess.run(train_step, feed_dict={xs: x_data})
    if i == num_train - 1:
        prediction_value = sess.run(prediction, feed_dict={xs: x_data})
        dist, ind = tree_copy.query(prediction_value, k=3)
        ind = np.unique(ind)
        train_candidates = train_x.iloc[ind]
        id = max_uncer(clf, train_candidates)
        mid_time = time.clock() - start1
        time_cost.append(mid_time)
```

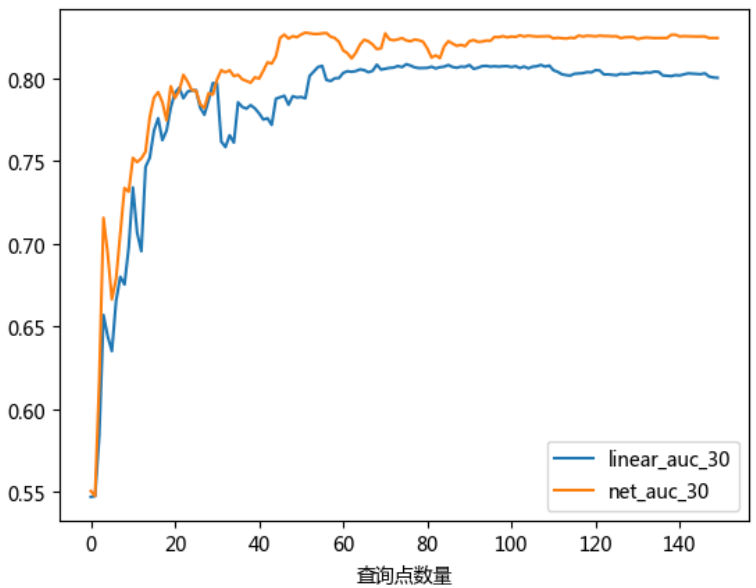
对比

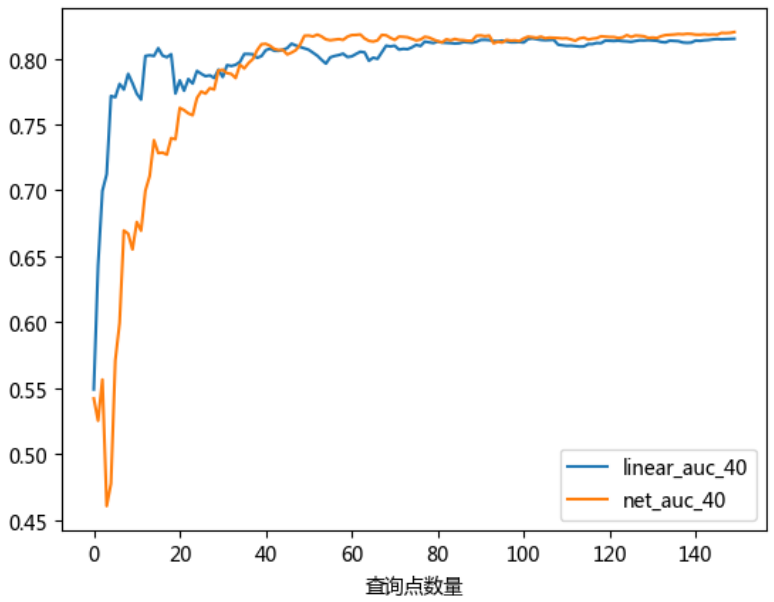
总时间：2.5h(net)——10.9h(linear)

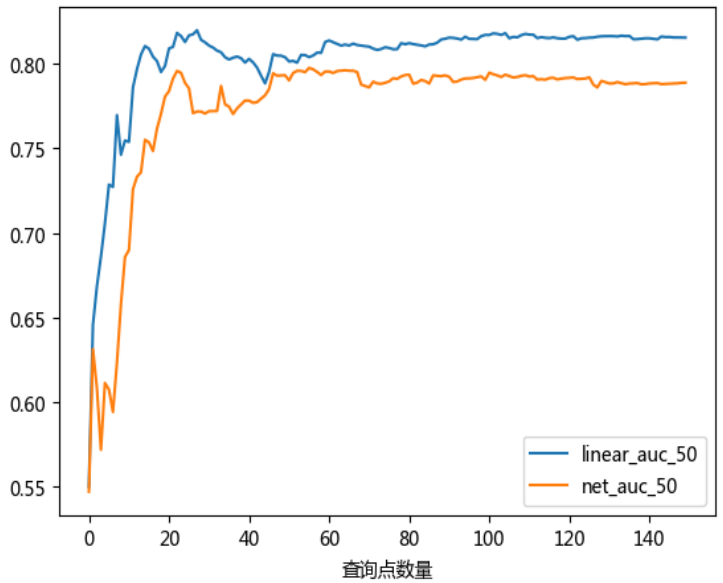


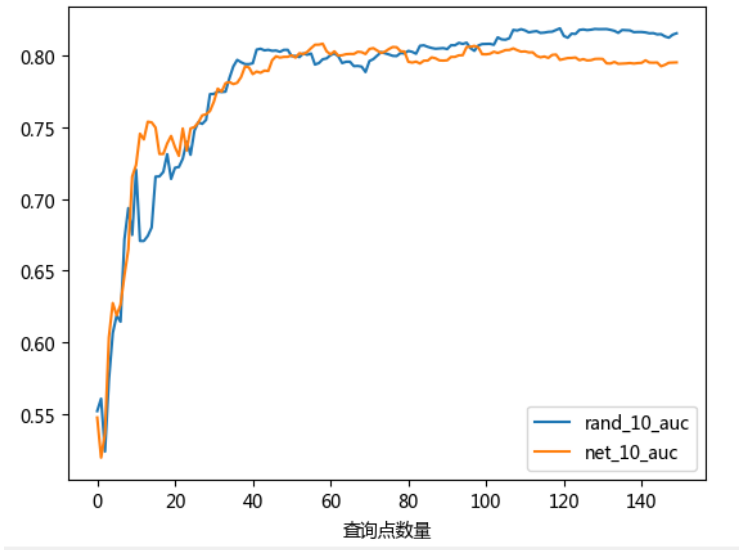

0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
linear_10_auc
net_10_auc

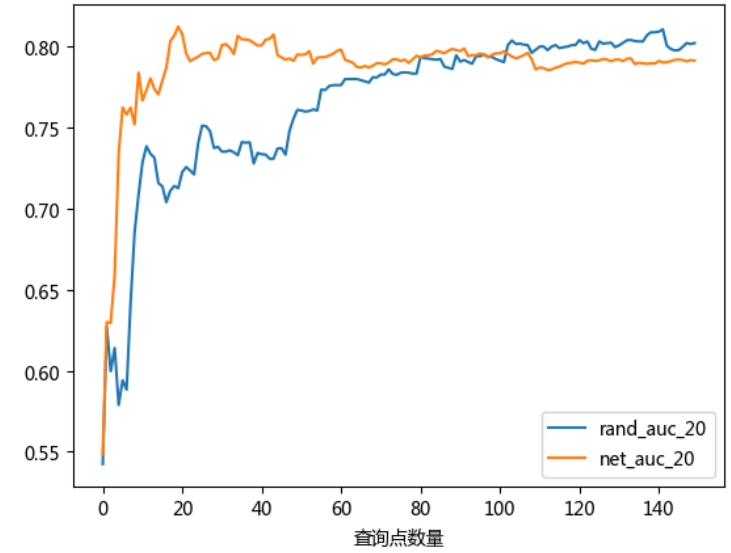

0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
linear_auc_20
net_auc_20

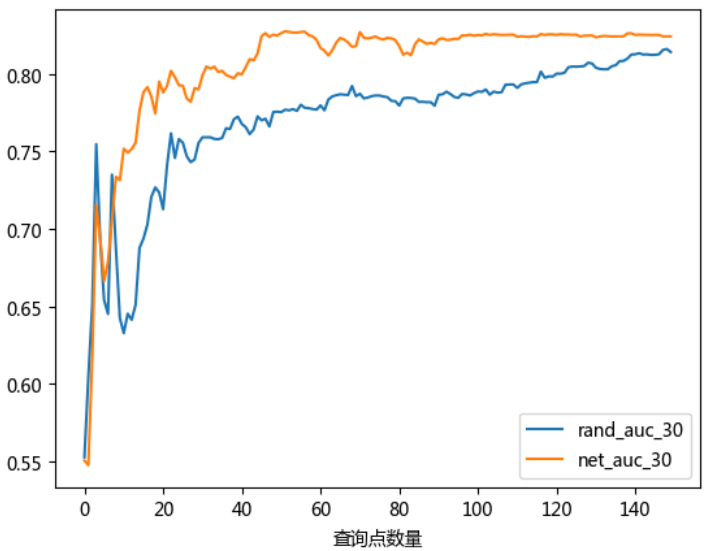

0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
linear_auc_30
net_auc_30

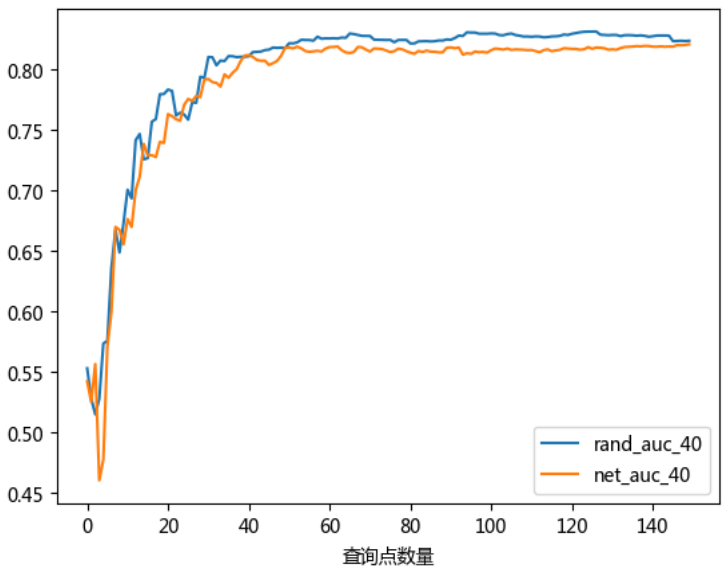

0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0
20
40
60
80
100
120
140
查询点数量
linear_auc_40
net_auc_40

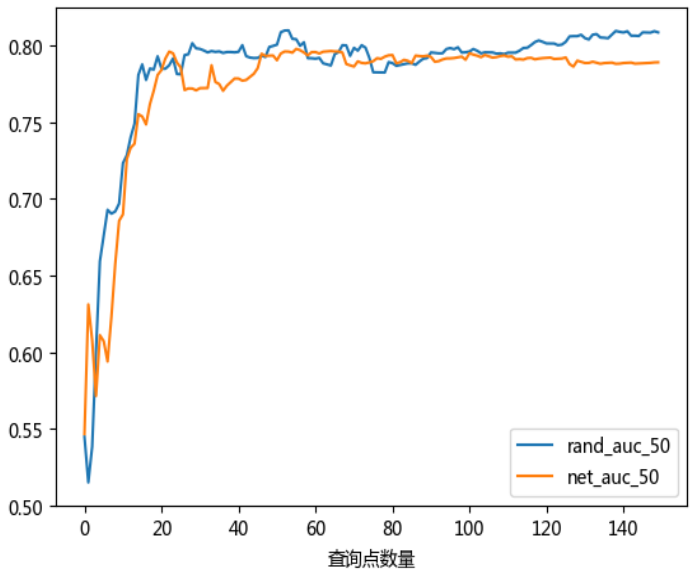

0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
linear_auc_50
net_auc_50


0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
rand_10_auc
net_10_auc


0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
rand_auc_20
net_auc_20


0.80
0.75
0.70
0.65
0.60
0.55
0
20
40
60
80
100
120
140
查询点数量
rand_auc_30
net_auc_30


0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0
20
40
60
80
100
120
140
查询点数量
rand_auc_40
net_auc_40


0.80
0.75
0.70
0.65
0.60
0.55
0.50
0
20
40
60
80
100
120
140
查询点数量
rand_auc_50
net_auc_50

$$\arg\min_{x} P_D(x; D_k, D_k^- \cup \{x\}) + \lambda_1 P_1(x, D_k) + \lambda_2 P_2(x, D_k^-)$$

$$P_1(x, D_k) = \max\{0, \arg\min_{x' \in D_k} dist(x, x') - C_1\} \qquad P_2(x, D_k^-) = \max\{0, C_2 - \arg\min_{x' \in D_k^-} dist(x, x')\}$$

优化目标

$$\min_{X \subset R^d} \quad -1 * \text{uncertainty}(X_{generate}) + \text{MMD}(Y_{rand\_select}, X_{generate})$$

$$\min \quad \frac{-1}{N_{batch\_size}} \sum_{i=1}^{batch\_size} \sum_{k=1}^{K=2} P_k(x_i) \log(P_k(x_i)) + \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) +$$

$$\frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j)$$

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2 * sigma^2}\right) \qquad m = n = batch_{size} = 300$$

$$x^*_{LC} = \operatorname*{argmax}_x \; 1 - P_\theta(\hat{y}|x) \qquad x^*_M = \operatorname*{argmin}_x P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x)$$

$$x^*_H = \operatorname*{argmax}_x - \sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$$

$$x^*_{EGL} = \operatorname*{argmax}_x \sum_i P_\theta(y_i|x) \left\| \nabla \ell_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \right\| \qquad x^*_{0/1} = \operatorname*{argmin}_x \sum_i P_\theta(y_i|x) \left( \sum_{u=1}^{U} 1 - P_{\theta^+\langle x, y_i \rangle}(\hat{y}|x^{(u)}) \right)$$

$$x^*_{\log} = \operatorname*{argmin}_x \sum_i P_\theta(y_i|x) \left( -\sum_{u=1}^{U} \sum_j P_{\theta^+\langle x, y_i \rangle}(y_j|x^{(u)}) \log P_{\theta^+\langle x, y_i \rangle}(y_j|x^{(u)}) \right)$$

$$Total\_query\_time = N_{unlabel} \times Time_{find\_x^*}$$

large