دانشگاه صنعتی امیرکبیر(پلی‌تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش تمرین سری ۴ درس بهینه سازی محدب

دکتر امیرمزلقانی

غلامرضا دار ۴۰۰۱۳۱۰۱۸

ابتدا داده‌های مربوطه را فراخوانی می‌کنیم. این دیتاست شامل ۸ فیچر و ۲ لیبل است. در ادامه می‌خواهیم با استفاده از الگوریتم ADMM به دو شکل موازی و غیرموازی و هم چنین LassoCVRegression به پیش بینی لیبل‌های این دیتاست بپردازیم.

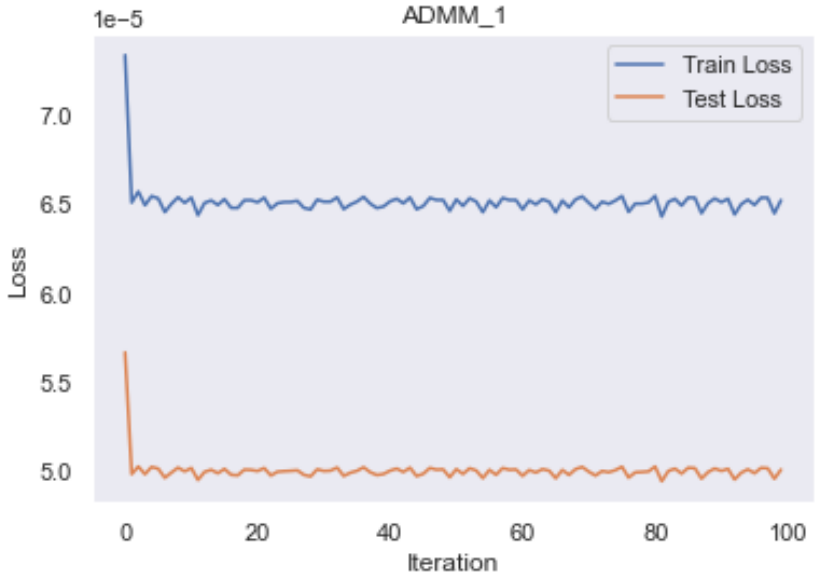
ADMM ; y_1

```
[219]
history1 = ADMM(
    X_train, y1_train,
    X_test, y1_test,
    dim=8,
    iterations=100,
    lambd=0.01)
```

```
Initial Value: 821.6779411830228

100%|██████████| 100/100 [00:00<00:00, 675.67it/s]

Iteration 0/100 | Value: 0.0247 | train_loss: 7.345e-05 | test_loss: 5.665e-05
Iteration 10/100 | Value: 0.0232 | train_loss: 6.54e-05 | test_loss: 5.013e-05
Iteration 20/100 | Value: 0.0231 | train_loss: 6.514e-05 | test_loss: 4.996e-05
Iteration 30/100 | Value: 0.0232 | train_loss: 6.517e-05 | test_loss: 4.996e-05
Iteration 40/100 | Value: 0.0231 | train_loss: 6.519e-05 | test_loss: 4.998e-05
Iteration 50/100 | Value: 0.0231 | train_loss: 6.529e-05 | test_loss: 5.006e-05
Iteration 60/100 | Value: 0.0232 | train_loss: 6.473e-05 | test_loss: 4.968e-05
Iteration 70/100 | Value: 0.0233 | train_loss: 6.51e-05 | test_loss: 4.992e-05
Iteration 80/100 | Value: 0.0231 | train_loss: 6.55e-05 | test_loss: 5.021e-05
Iteration 90/100 | Value: 0.0232 | train_loss: 6.514e-05 | test_loss: 4.997e-05
```



ADMM ; y_2

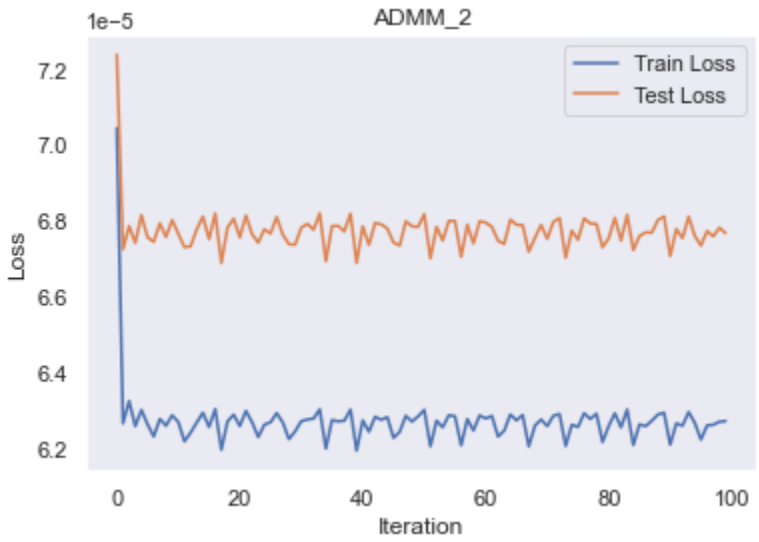
```
history2 = ADMM(
    X_train, y2_train,
    X_test, y2_test,
    dim=8,
    iterations=100,
    lambd=0.01)
```

[221]

```
Initial Value: 211.344327670127

100%|██████████| 100/100 [00:00<00:00, 704.22it/s]

Iteration 0/100 | Value: 0.0247 | train_loss: 7.374e-05 | test_loss: 6.035e-05
Iteration 10/100 | Value: 0.0234 | train_loss: 6.636e-05 | test_loss: 5.401e-05
Iteration 20/100 | Value: 0.0233 | train_loss: 6.614e-05 | test_loss: 5.383e-05
Iteration 30/100 | Value: 0.0233 | train_loss: 6.654e-05 | test_loss: 5.415e-05
Iteration 40/100 | Value: 0.0232 | train_loss: 6.636e-05 | test_loss: 5.401e-05
Iteration 50/100 | Value: 0.0234 | train_loss: 6.577e-05 | test_loss: 5.356e-05
Iteration 60/100 | Value: 0.0233 | train_loss: 6.652e-05 | test_loss: 5.413e-05
Iteration 70/100 | Value: 0.0232 | train_loss: 6.634e-05 | test_loss: 5.399e-05
Iteration 80/100 | Value: 0.0235 | train_loss: 6.53e-05 | test_loss: 5.32e-05
Iteration 90/100 | Value: 0.0234 | train_loss: 6.604e-05 | test_loss: 5.377e-05
```

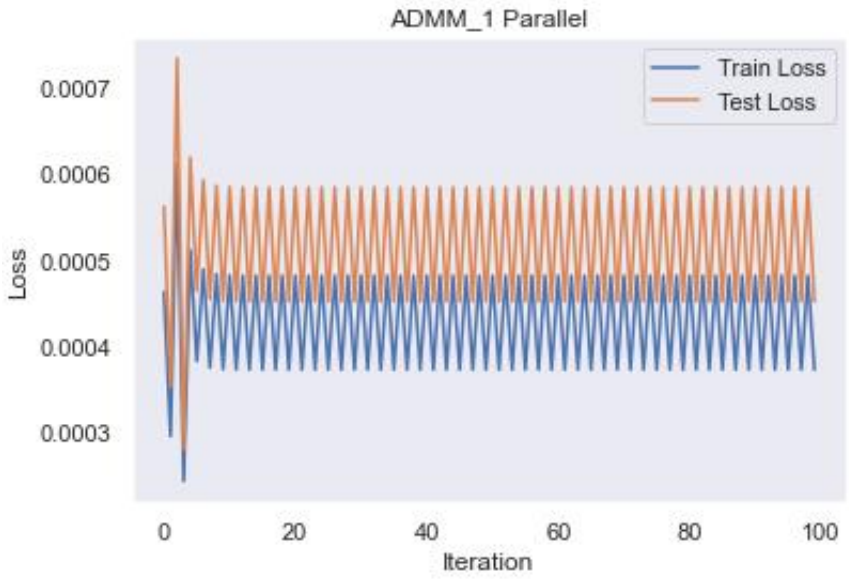


Parallel ADMM ; y_1

```
[43]
1  history3 = Parallel_ADMM(
2      X_train, y1_train,
3      X_test, y1_test,
4      dim=8,
5      iterations=100,
6      lambd=0.01)
```

✓ 6.7s

```
Iteration 0/100 | Value: 0.1426 | train_loss: 0.00046373 | test_loss: 0.0005629
Iteration 10/100 | Value: 0.149 | train_loss: 0.00048309 | test_loss: 0.00058573
Iteration 20/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 30/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 40/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 50/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 60/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 70/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 80/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
Iteration 90/100 | Value: 0.1489 | train_loss: 0.00048264 | test_loss: 0.00058519
```

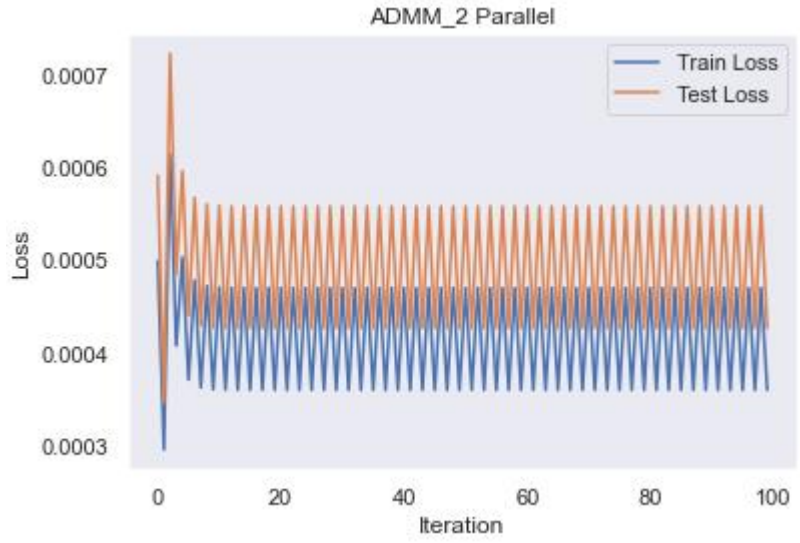


Parallel ADMM ; y_2

```
history4 = Parallel_ADMM(
    X_train, y2_train,
    X_test, y2_test,
    dim=8,
    iterations=100,
    lambd=0.01)
```

[45] ✓ 6.4s

```
Iteration 0/100 | Value: 0.1537 | train_loss: 0.00049966 | test_loss: 0.00059219
Iteration 10/100 | Value: 0.1455 | train_loss: 0.00047141 | test_loss: 0.00055933
Iteration 20/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 30/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 40/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 50/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 60/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 70/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 80/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
Iteration 90/100 | Value: 0.1453 | train_loss: 0.00047091 | test_loss: 0.00055875
```



مقایسه با روش Lasso CV Regression

مشاهده می‌شود که Lasso Regression از خطای کمتری برخوردار است.