1. Before training, we firstly use the standardization function to standardize the training data. We use the equation that: (data sample - mean) / variance.

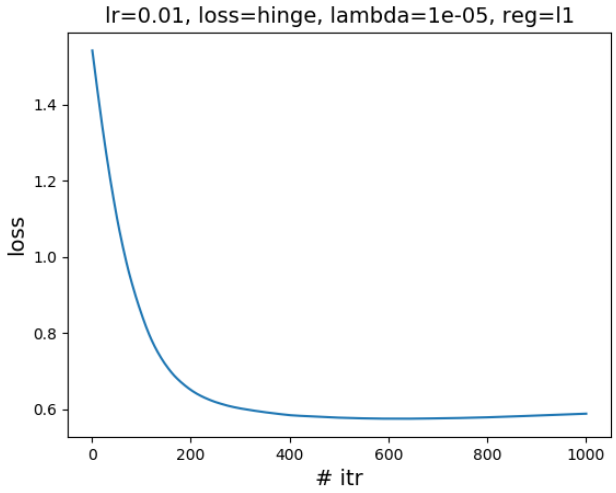
   During the training stage, we use the 5-fold cross validation method with two learning rates: 0.01 and 0.05. (training iteration = 1000) We can then get the intermediate training results. There are two loss functions, one is logistic loss and the other is hinge loss.

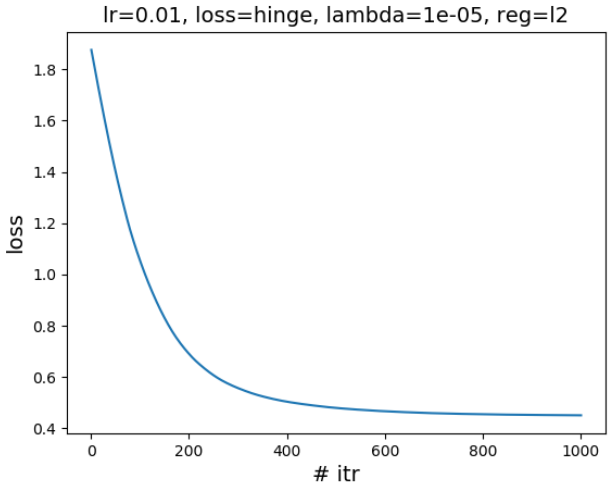
   In the test phase, (1) at the end of each training, the validation set is used; (2) test set is used for testing. We evaluate by checking the accuracy rate (ACC). At the end of each round, the results will be printed.

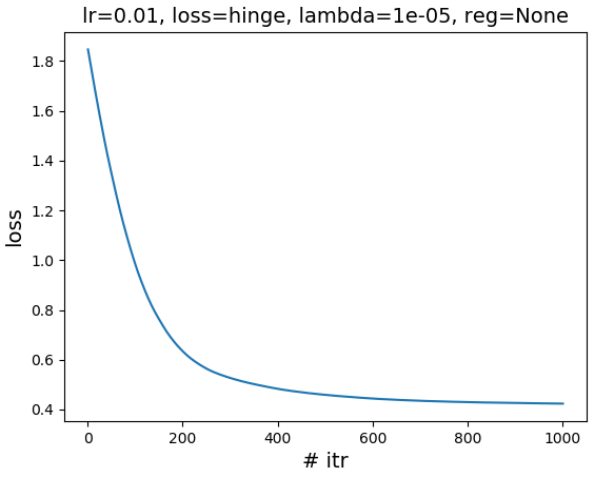
2. Through the experiment, we find that the training speed is faster when the learning rate is 0.05 than when it is 0.01. Hinge loss is more stable than logistic loss, and the training speed is faster. At the same time, adding L1 or L2 regularities may make the training unstable.

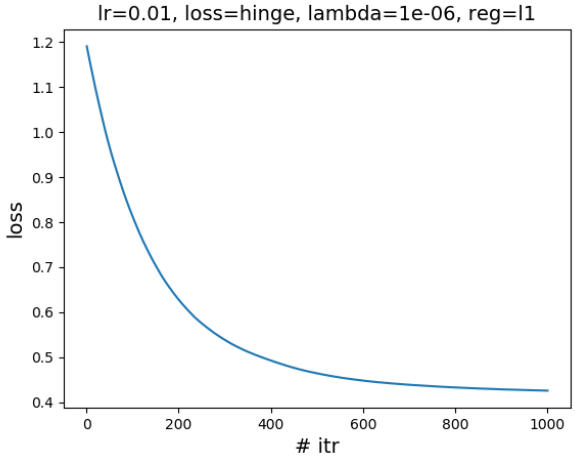
3. I have attached the plots of the loss as a function of the number of iterations below. Please check them out, thanks.

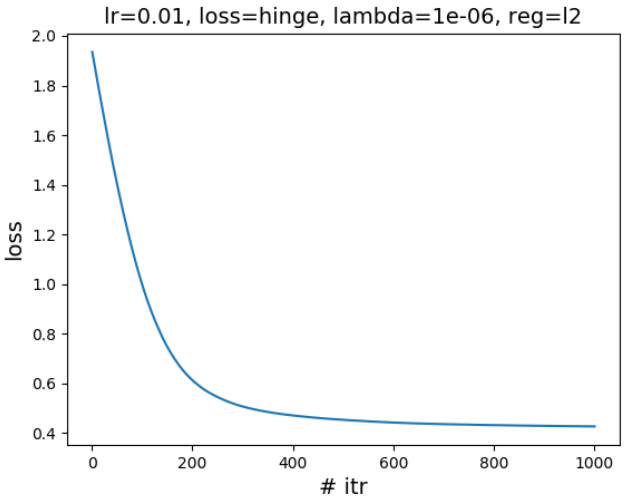
0.01:

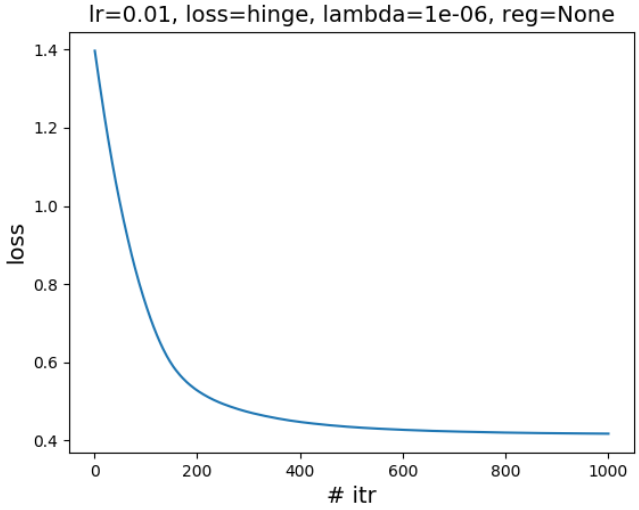

lr=0.01, loss=hinge, lambda=1e-05, reg=l1
loss
itr
1.4
1.2
1.0
0.8
0.6
0
200
400
600
800
1000

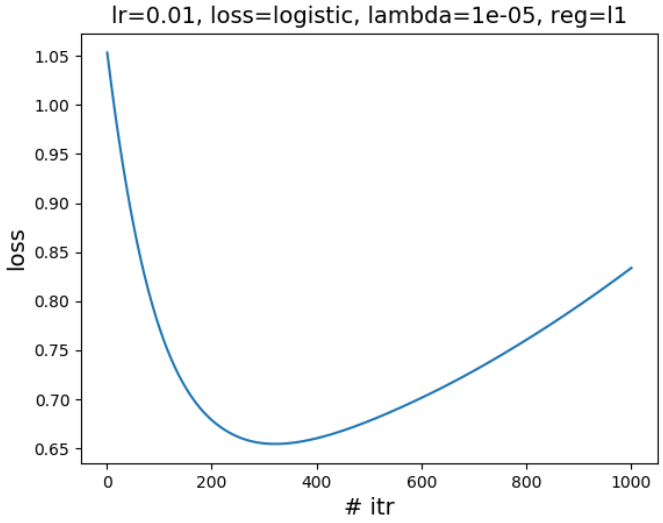

lr=0.01, loss=hinge, lambda=1e-05, reg=l2
loss
itr
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

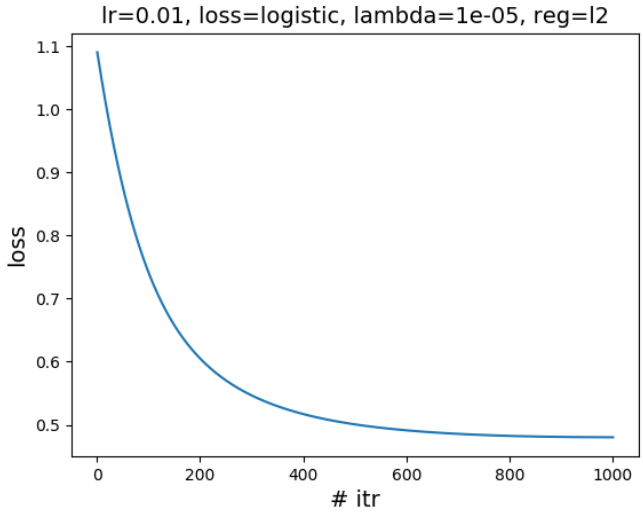

lr=0.01, loss=hinge, lambda=1e-05, reg=None
loss
itr
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

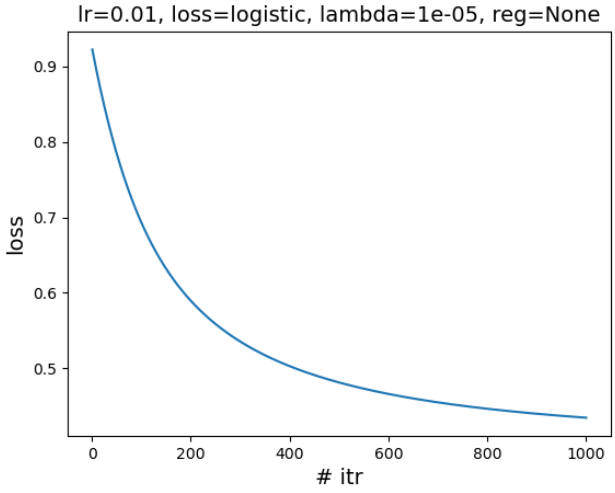

lr=0.01, loss=hinge, lambda=1e-06, reg=l1
loss
itr

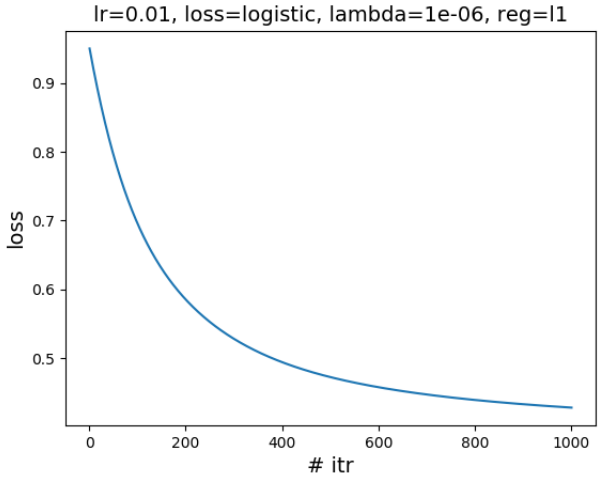

lr=0.01, loss=hinge, lambda=1e-06, reg=l2
loss
itr

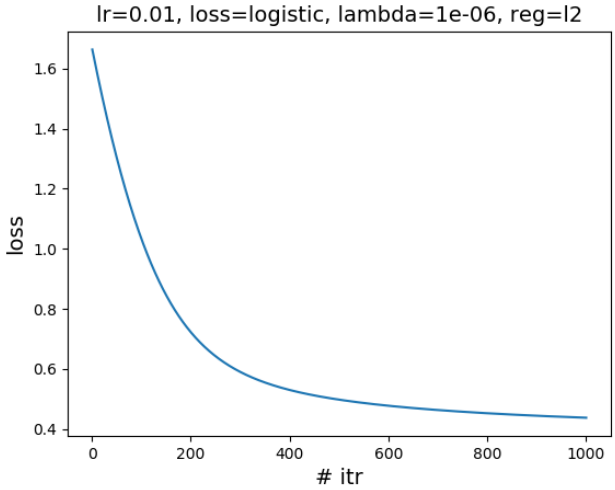

lr=0.01, loss=hinge, lambda=1e-06, reg=None
loss
itr

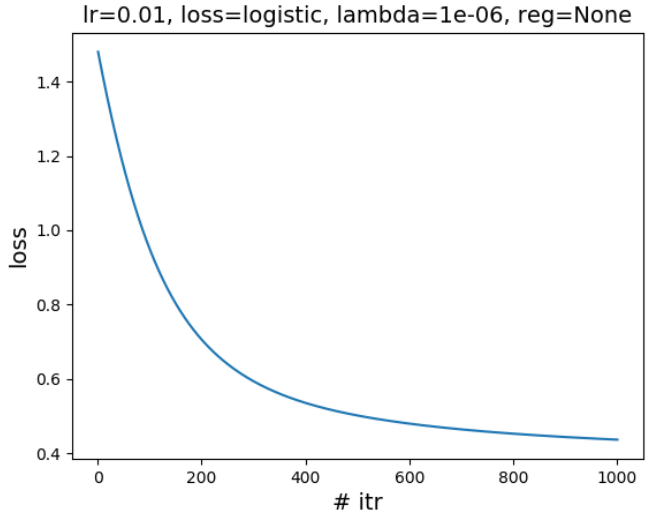

lr=0.01, loss=logistic, lambda=1e-05, reg=l1
loss
itr
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
200
400
600
800
1000


lr=0.01, loss=logistic, lambda=1e-05, reg=l2
loss
itr
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0
200
400
600
800
1000

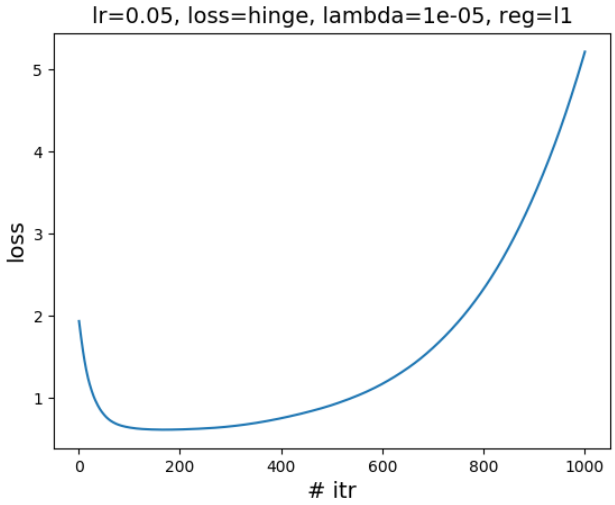

lr=0.01, loss=logistic, lambda=1e-05, reg=None
loss
itr
0.9
0.8
0.7
0.6
0.5
0
200
400
600
800
1000

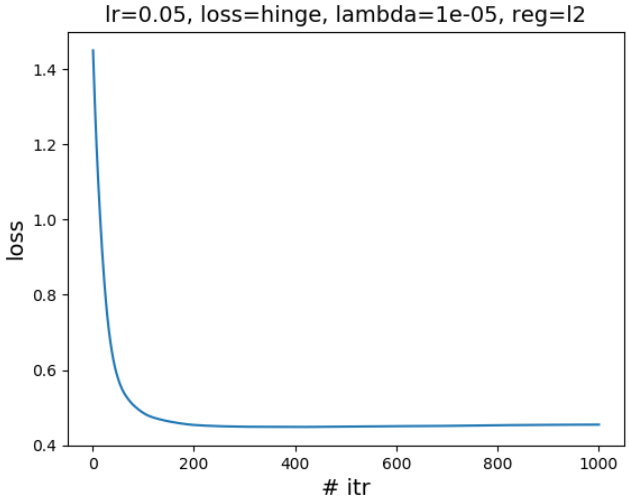

lr=0.01, loss=logistic, lambda=1e-06, reg=l1
loss
itr
0.9
0.8
0.7
0.6
0.5
0
200
400
600
800
1000

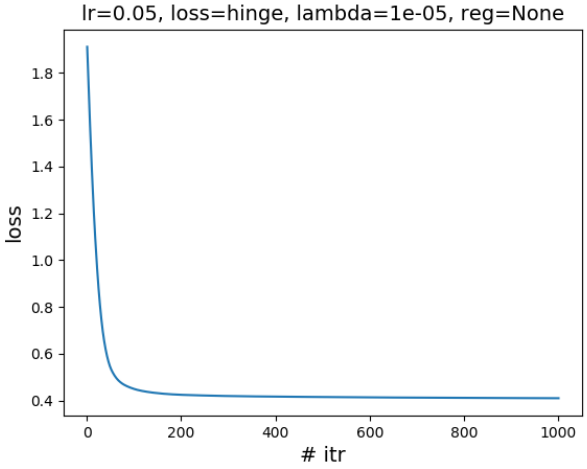

lr=0.01, loss=logistic, lambda=1e-06, reg=l2
loss
itr
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

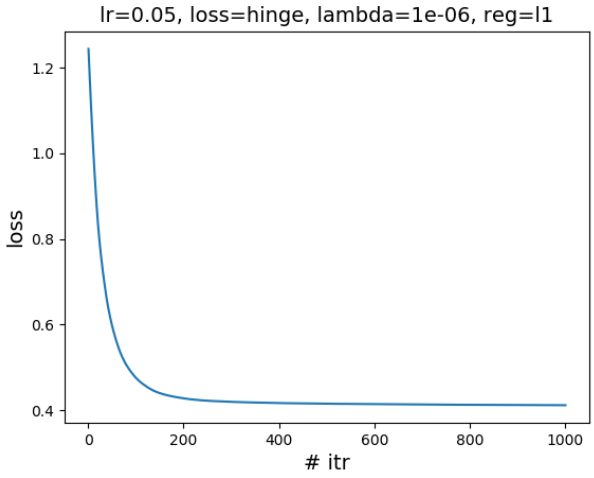

lr=0.01, loss=logistic, lambda=1e-06, reg=None
loss
itr
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

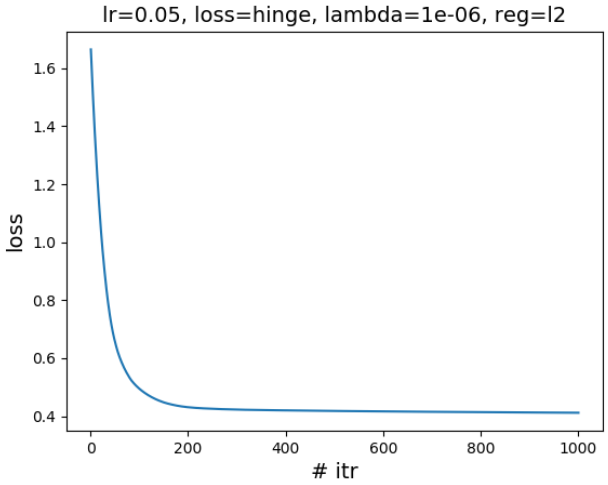
0.05:

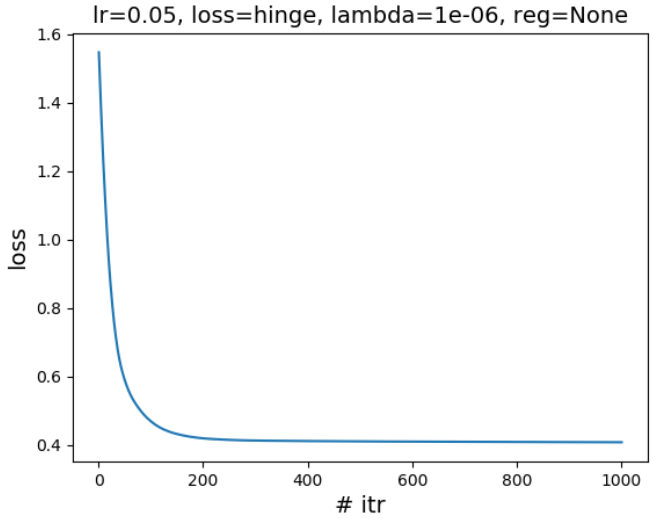

lr=0.05, loss=hinge, lambda=1e-05, reg=l1
loss
itr
1
2
3
4
5
0
200
400
600
800
1000

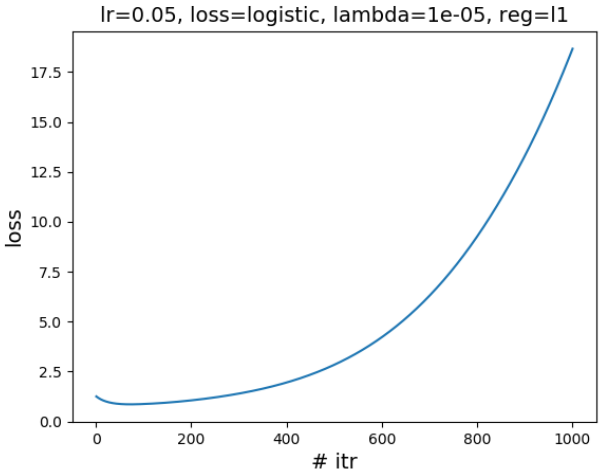

lr=0.05, loss=hinge, lambda=1e-05, reg=l2
loss
itr
0.4
0.6
0.8
1.0
1.2
1.4
0
200
400
600
800
1000

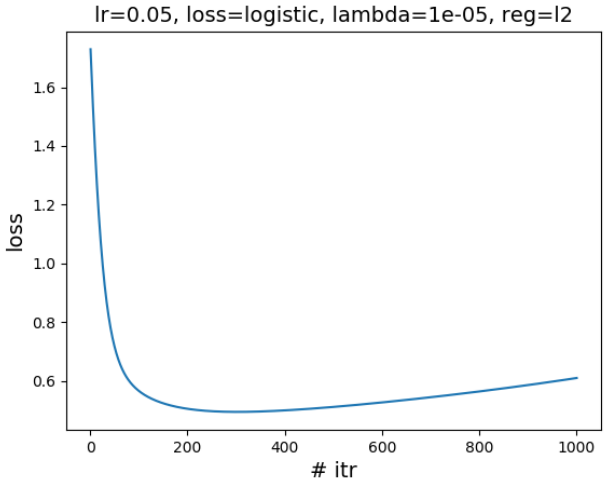

lr=0.05, loss=hinge, lambda=1e-05, reg=None
loss
itr
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
0
200
400
600
800
1000

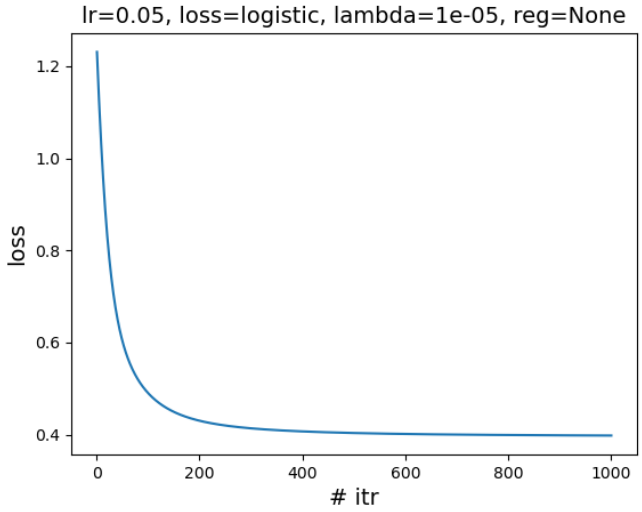

lr=0.05, loss=hinge, lambda=1e-06, reg=l1
loss
itr
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

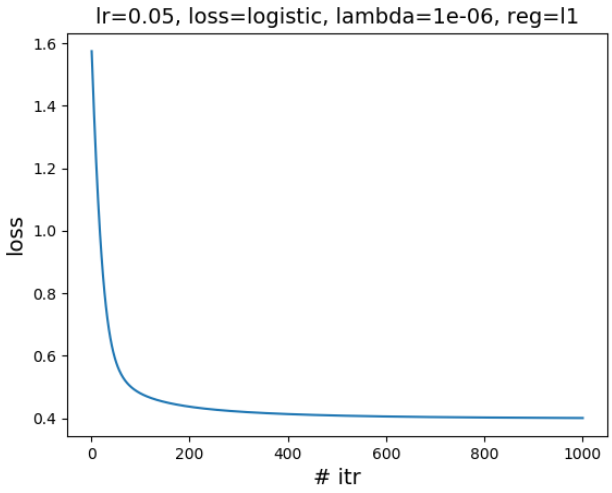

lr=0.05, loss=hinge, lambda=1e-06, reg=l2
loss
itr
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

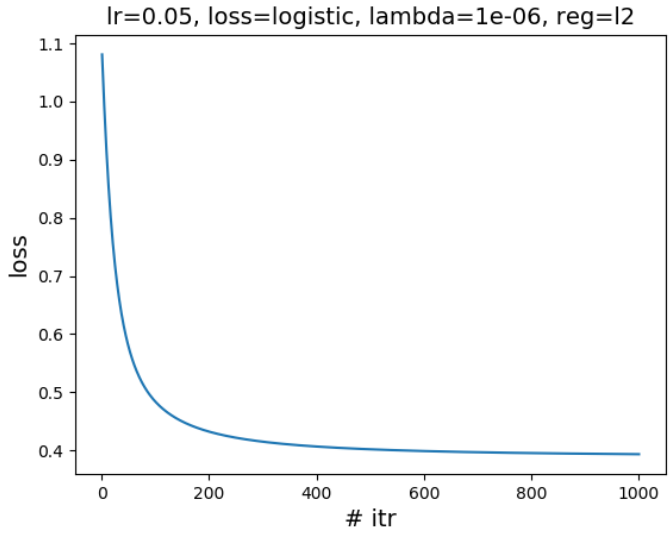

lr=0.05, loss=hinge, lambda=1e-06, reg=None
loss
itr
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

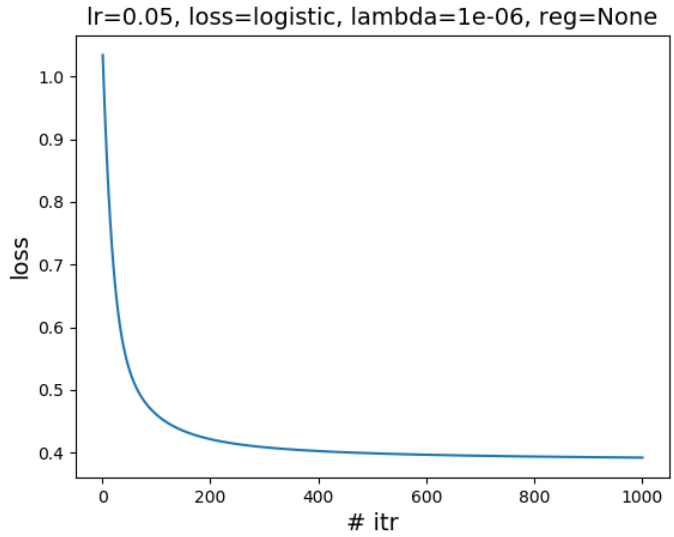

lr=0.05, loss=logistic, lambda=1e-05, reg=l1
loss
itr
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
200
400
600
800
1000


lr=0.05, loss=logistic, lambda=1e-05, reg=l2
loss
itr
0.6
0.8
1.0
1.2
1.4
1.6
0
200
400
600
800
1000


lr=0.05, loss=logistic, lambda=1e-05, reg=None
loss
itr
0.4
0.6
0.8
1.0
1.2
0
200
400
600
800
1000


lr=0.05, loss=logistic, lambda=1e-06, reg=l1
loss
itr
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000


lr=0.05, loss=logistic, lambda=1e-06, reg=l2
loss
itr
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
200
400
600
800
1000


lr=0.05, loss=logistic, lambda=1e-06, reg=None
loss
itr
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
200
400
600
800
1000