Final Report

By: Huyunting Huang

Based on the result in the Pre-report, for kernel SVM I then constrain the hyper-parameter within range (0,0.02) with K=5(in cross validation), sample number=2000~3000 to estimate a good beta (which is V2&V3 for). And the result is as fig.2 & fig.3 shown.

Also, the evaluation metric change from 0-1 loss to AMS metric.

$$\mathrm{AMS} = \sqrt{2\left((s + b + br)\log\left(1 + \frac{s}{b + br}\right)\right)}$$

Where s,b : unnormalized true **prediction** and false **prediction** rates, respectively.

$$s = \sum_{i=1}^{n} \delta\,(y[i],\ y'[i])$$

$$b = n - s$$

$$\delta\,(y[i],\ y'[i]) = 1\ (\text{if}\ y[i] = y'[i])$$

$$\delta\,(y[i],\ y'[i]) = 0\ (\text{if}\ y[i]\ != y'[i])$$

Where y[i] is the label of prediction and y'[i] is the label of test set.

br=10 is the constant regularization term.

log is the natural log.

(PS: in the original web(URL is in the project plan) s,b is the unnormalized true **positive** and false **positive** rates, respectively)

Which means:

$$s = \sum_{i=1}^{n} \delta\,(y[i],\ 1)\ \delta\,(y'[i],\ 1)$$

$$b = \sum_{i=1}^{n} \delta\,(y[i],\ 0)\ \delta\,(y'[i],\ 1)$$

I used a **‘wrong’** evaluation metric when I was training the data.The difference between this two metric is that for the **‘wrong’** metric I should choose the beta that has **minimal** score while for the **‘right’** metric I should choose the beta that has **maximal** score. Because the **‘right’** metric focuses on its power of correctness and the ‘wrong’ metric focuses on model’s precision.

So I here just used the ‘right’ metric as the 2nd version of evaluation of prediction which is stored as kernelAMSscorev1b, 2b, 3b, etc .

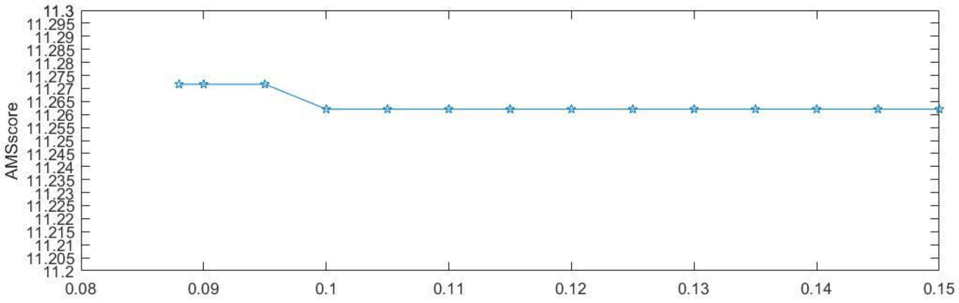

Fig.1 2nd training CV record (K=5, #of training sample =4000, beta = 0.88, and then goes from 0.09 to 0.15 with interval 0.005)

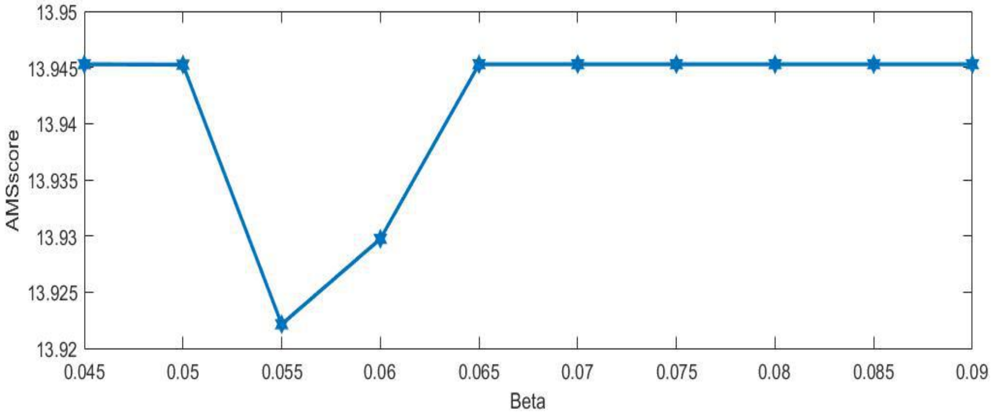

Fig.2 3rd training CV record(K=5, #of training sample =6000, beta goes from 0.045 to 0.09 with interval 0.005)

This two plot shows that as the number of training sample change the beta of minimal score will also change. But what they all have in common is that it tends to have a limit where the score will not change as the beta continues getting bigger.

But because the number of point is relatively small, based on the result of 2nd training I used a smaller interval and increase the K into 8 to do the 4th training. The CV record is as fig.3

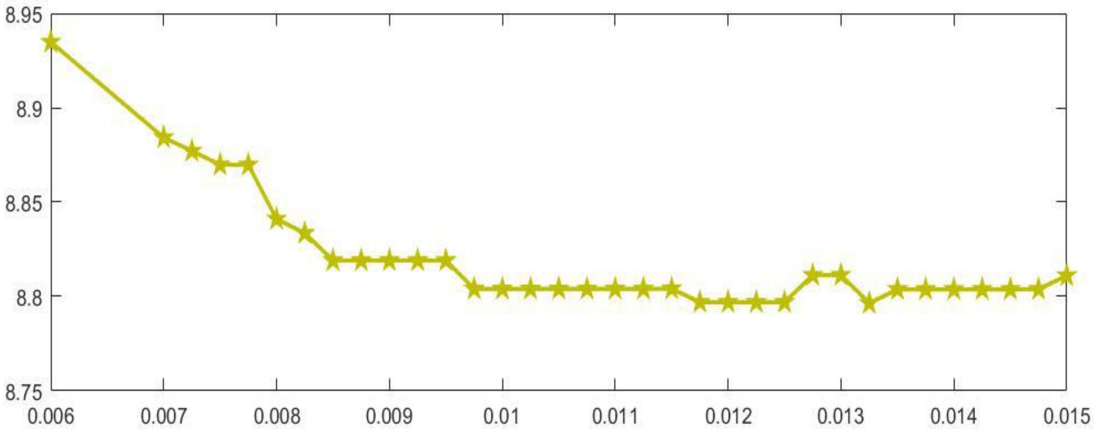

Fig.3 4th training CV record (# of training sample = 4000, K for CV = 8,beta = 0.006

and then goes from 0.007 to 0.015 with interval = 0.00025)

This 4th training record shows that as the beta goes, the AMS score could still has some kind of perturbation but in all it won't change drastically.

The prediction result (calculated by function calculator(or v2 for **'right'** **metric**).m)is shown in Table.1

Version	**'wrong'** AMS score	**'right'** AMS score
2	57.4483	79.4434
3	57.2808	80.3937
4	57.1805	81.7712

Table 1. the evaluation of each training model.

From the table we can see that the performance of model is somehow consistent, the model with lower '**wrong**' AMS score also has a higher '**right**' AMS score. Also the change of beta and # of sample will not influence the performance of model too much as the score is bounded around 57(for **'wrong'**) or 80(for the **'right'**).

Besides radial-basis kernel SVM, I also tried linear SVM model to do a comparison and the result is in the work_history_for_linear.pdf file. (I forget to store the data when I input clear all commands). I here just change the function into linear version in the kfoldcv.m and svmtrain.m(and then change back to the kernel one). The record in the file(at the page 5 --- 6) shows that the linear model might be systematically worse than Kernel model because it gets a somehow obvious higher **‘wrong’** AMS score than Kernel (around 2). As a result I finally switch back to kernel model.