Delft University of Technology

Machine Learning
IN4320

Final Project

Author:
Yuan Tian (4716159)
Netid: ytian4

July 7, 2018

1 Abstract

In the E Corp Challenge, we meet three main problems : extremely low training data, relative high dimension, corrupted test data.
To deal with these challenge, my idea is building a classifier which is robust on p-norm noise. By generating some data with different p-norm noise to increase the number of training set as well as improve the classifier's robustness.
As the classifier I choose the L2-SVM due to SVM is quite good for high dimension binary classification problem and L2 regularisation could beat overfitting and also good for improving model robustness. And the learning method, I choose Self-training semi-supervised learning which could use as much high confidence unlabeled data as possible.
The estimation of the error rate may vary between 25% to 45%,according to different r and p of the noise on the data.And on average, the error rate may around 35%.

2 Normalization

Due to every dimension has different scale and SVM is not scale invariance and we will add noise later. So, I normalize the dataset at beginning.

3 Dataset augmentation

Firstly, there is a very large unlabeled dataset with p-norm noise and a very small labeled dataset without noise. In this condition, I am trying to add p-norm noise on the labeled data to get more labeled data. Meanwhile, by training a classifier with these data could also improve the classifier's robustness with noise.

3.1 Parameters setting

To get as more as possible informative dataset, I try to add different r,p combination. And I do an experiment to find reasonable parameter range.

(i) The SVM setting in this experiment is L2 regularisation with polynomial kernel and the Outlier Fraction is 0.01.

(ii) I use leave one out cross validation on the original 200 samples dataset, using 199 samples for training and add p-norm noise to the one test data.See Fig 1 and 2 below, which shows when r and p over 20, the data will become very difficult to classify.In other words, I think when r and 20 are over 20, the data will be another different data.

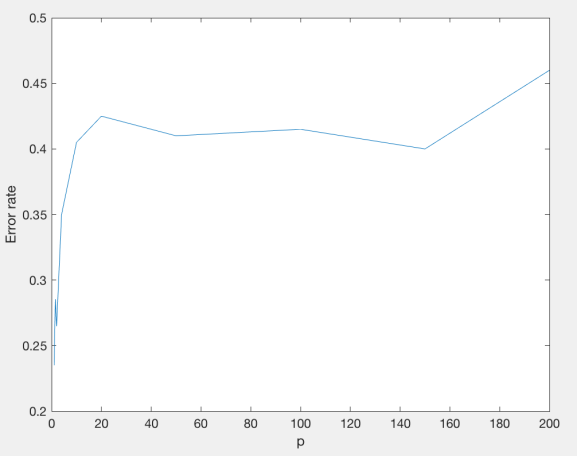

Figure 1: p-norm with r=10

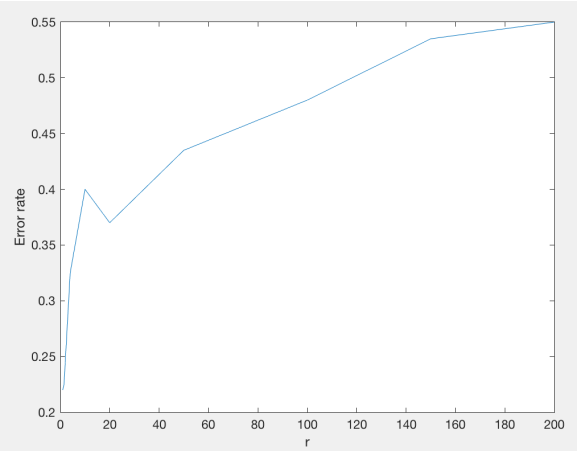

Figure 2: 10-norm different p

(iii) So, I do the dataset augmentation with p = [1,1.5,2,5,10] and r = [1,2,5,10,15,20]. With all the r,p combinations,the training set expand to 6200 samples.

3.2 Conclusion

Then I compare the error rate on add (r,p)=[(1,1),(2,2),(5,5),(8,8),(10,10),(12,12),(15,15),(18,18),(20,20)] to the one test data between the training set without data augmentation and the training set with data augmentation.See Fig 3(5 times average), which shows that the data augmentation could slightly improved the noise robustness of the SVM classifier.

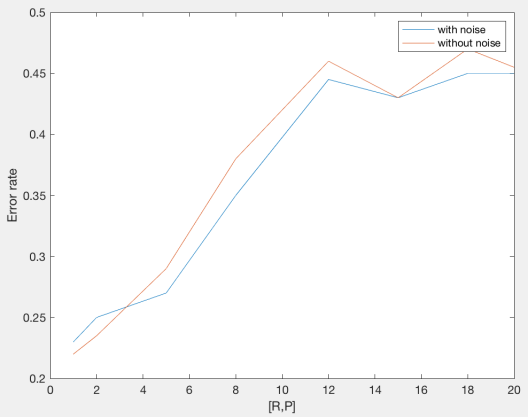

Figure 3: Comparison

3.3 p-norm noise generator matlab code

```
function p_norm_data=p_norm(p,r,data)
%%% p-norm noise generator
%%% 'p' is the real number p ti difine the p-norm
%%% 'r' is the real number r to difine the p-norm-ball's radius
%%% 'data' is the input data
%%% 'p_norm_data' is the output data with additional p-norm noise
 [m,n]=size(data);
p_norm_data=data;
for i=1:m
    list=randperm(n);
    range=r;
    for t=1:n
        noise =-range + 2*range*rand;
        p_norm_data(m,list(t))=p_norm_data(m,list(t))+noise;
        range=(range^p-abs(noise^p))^(1/p);
    end
end
end
```

4 SVM Setting

To optimize the robustness and performance for SVM, I do three experiments to find the optimal Box constraint, Polynomial kernel function order and Expected proportion of outliers in the training data.
The initial setting of these parameters are [100,3,0.01] and all training data are with data augmentation and the test data add a 10-norm noise with radius 10.

4.1 Polynomial kernel function order

Do the leave one out cross validation on training set with different Polynomial kernel function order. See Fig 4, which shows that the 2 could be a good Polynomial kernel function order.

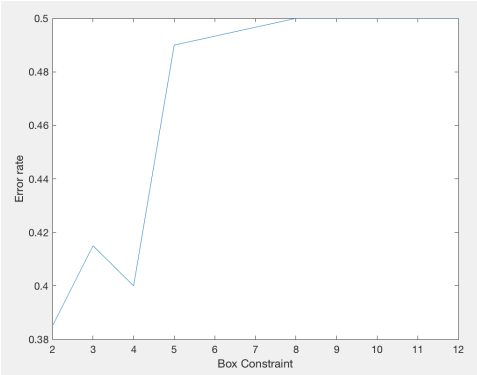

Figure 4: Polynomial kernel function order

4.2 Box constraint

Do the leave one out cross validation on training set with different Box constraint. See Fig 5, which shows that the 600 could be a good value of Box constraint.

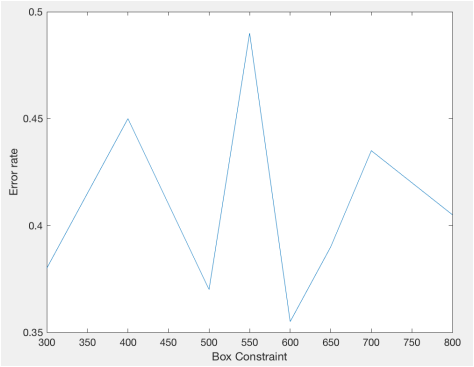

Figure 5: Box constraint

4.3 Expected proportion of outliers

Do the leave one out cross validation on training set with different Expected proportion of outliers. See Fig 6, which shows that the 0.01 could be a good proportion of outliers.

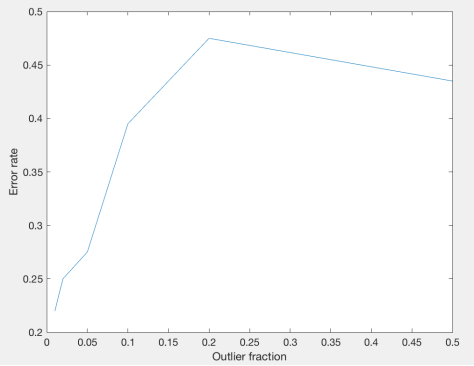

Figure 6: proportion of outliers

4.4 Conclusion

So, with all the result, I choose Polynomial kernel function order = 2, Box constraint = 600 , and Expected proportion of outliers = 0.01 for my SVM classifier.

5 Semi-supervised learning

Using the self training on the unlabeled data,and every iteration replace the test samples which have highest confidence into the training set to update a new SVM classifier.Iteration until we get 600(I believe more will be better, but it will take a extremely long time) additional training samples.
To verify the performance of adding additional samples from self learning to the training set.I do the leave one out cross validation on the original training set.For every 199 training samples from self learning I will put additional sample into it,and then train a classifier and test on the one test sample.See Fig 7, which shows that more additional data could improve the classifier's performance.

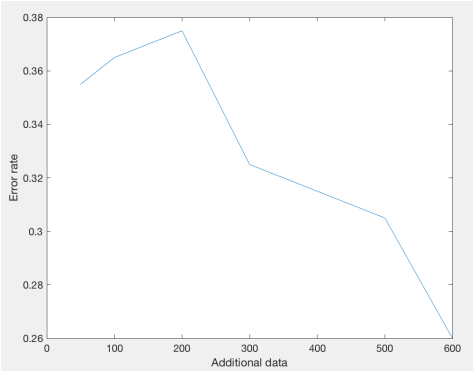

Figure 7: Polynomial kernel function order

6 Estimation of the error rate

Doing the leave one out cross validation on the original training set. For every 199 training samples,I will do data augmentation on them and put 1000 additional sample from self learning into the training set,and then learn a SVM with the parameter we get in previous section.
Then, the estimation of the error rate could be done by adding different noise on the one test sample. See Fig 8, which shows the range of error rate with different noise(I compare the error rate on add (r,p)=[(1,1),(2,2),(5,5),(8,8),(10,10),(12 to the one test data). So the error rate should be in 25% -45% according to different r and p of the noise on the data.And on average, the error rate may around 35%.

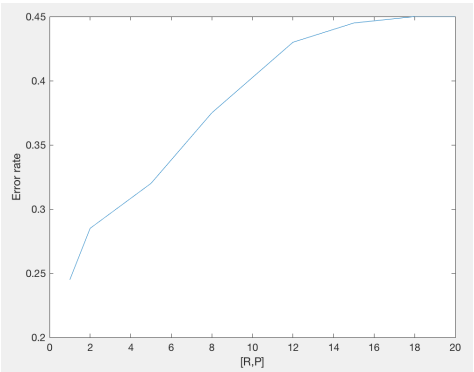

Figure 8: Error rate