Homework 05

Tianyang Chen

1 An Experiment on Synthetic Data

The code for this problem is in directory ./code/Synthetic_data.ipynb

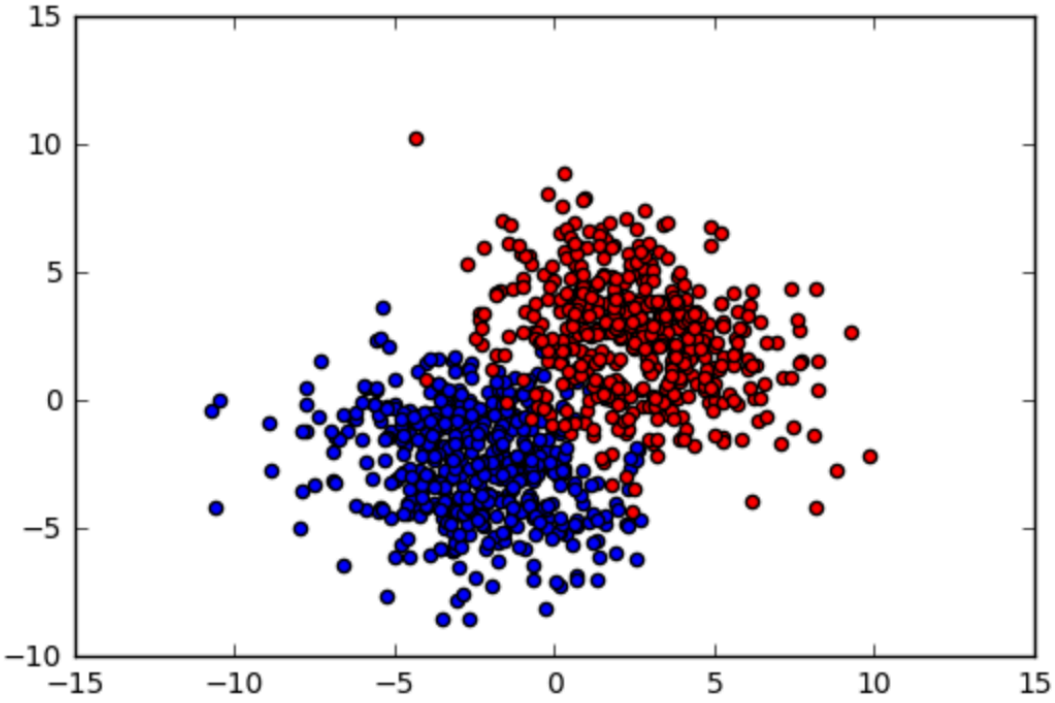

Fig.1 Data set for problem 1

In this problem, I use Gaussian Naive Bayes as the classifier, and set the probability threshold to be 0.9999.

Table 1 Testing error with self-training rounds

Round	Percentage of Labeled Data	Error
0	10.0%	0.0411
1	21.6%	0.0411
2	23.9%	0.0422

3	24.9%	0.0411
4	25.6%	0.0411
5	26.4%	0.0422
6	28.9%	0.0411
7	32.4%	0.0389
8	36.0%	0.0411
9	39.9%	0.0411
10	42.5%	0.0433
11	43.8%	0.0433
12	44.5%	0.0433
13	44.6%	0.0444

In this table, round 0 means the classifier is trained using only the labeled data, The last round means the self-training is completed, the other rounds between round 0 and the last round means some pseudo labels are used.

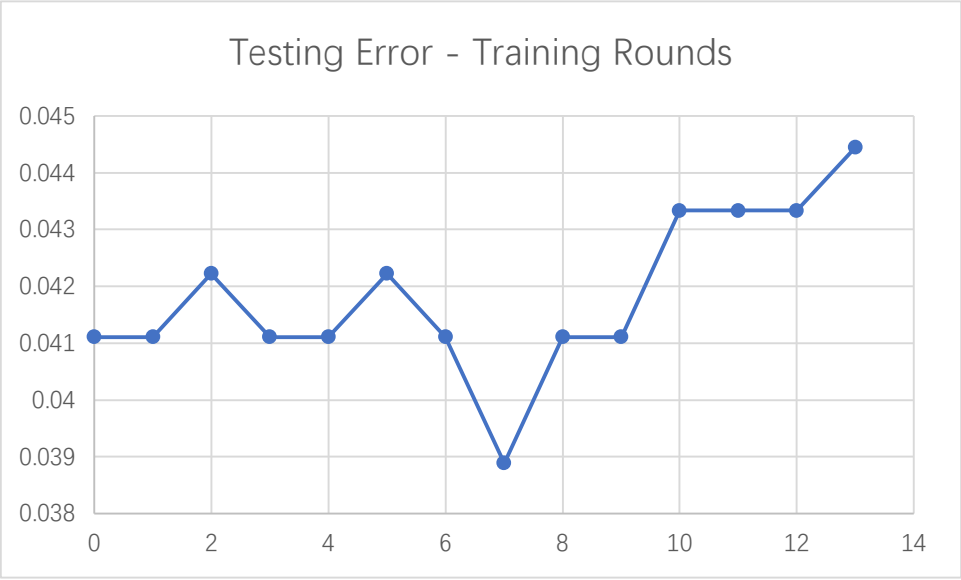

Fig.2 Testing error with training rounds

According to the results, the error slightly fluctuates during the self-training process, and the final error is usually bigger than beginning. This is because the self-training algorithm depends on the original labeled data,

if the original labeled data was sampled improperly, self-training will reinforce such mistakes.

2 An Experiment on Real World Data

The code for this problem is in directory ./code/ Real_world_data.ipynb

In this problem, I use Gaussian Naive Bayes as the classifier. I not only report the 5-fold cross validation error with self-training algorithm, but also the 5-fold cross validation error without semi-supervised learning.

Table 2 5-fold cross validation error on real world data

	Error	
Data Set	With SSL	Without SSL
abalone.csv	0.4333	0.4242
acute-inflammation.csv	0.1667	0.1750
acute-nephritis.csv	0.0667	0.0417
adult_train.csv	0.2154	0.1880
annealing_train.csv	0.2231	0.4022
arrhythmia.csv	0.8318	0.8694
audiology-std_train.csv	0.7063	0.3196
balance-scale.csv	0.2624	0.1056
bank_train.csv	0.7146	0.1784
blood_train.csv	0.2559	0.2475

According to these 10 data sets, it is easy to tell that the error of self-training algorithm is usually bigger than supervised learning. For the data sets “audiology-std_train.csv” and “bank_train.csv”, we can see that semi-supervised learning greatly increased the error, however, except for these two, the error of the other data sets increased to some tolerable extent.

Then, I look at the data in “audiology-std_train.csv” and “bank_train.csv”,

I find that these two data sets have greatly unbalanced classes. Because I randomly selected 15% of data as labeled, it is very likely that most of these labeled data are in the same class, such that violently reinforce the mistake.

In a word, if the real-word data has balanced classes, semi-supervised leaning helps; if not, semi-supervised may lead to disaster.