IN4320: Machine Learning Assignment 4

Zheng Liu (4798406)

May 2, 2019

1 a

I considered that using data both unlabelled and labelled could build better learners than using single one alone. My methods of two semi-supervised learning methods are described as follow:

The first method is Self-training model [1, 2]. The idea of self-training is starting from labelled data to train the original model, to use the model to predict unlabelled data, assign best k unlabelled data labels and add into labelled data set, training new model and repeat until there is no unlabelled data.

- Input: labelled data: X_l, unlabelled data: X_u
  Repeat
- Train classifier f by X_l
- Use f to predict X_u and assign labels for top-k best-fitting samples $X_{predict}$
- $X_u = X_u - X_{predict}$, $X_l = X_l + X_{predict}$
  End until there is no unlabelled data: $X_u = \emptyset$
- Output: f

The second method is Co-training [2]. The assumption of co-training is $x = [x^{(1)}; x^{(2)}]$ exists, which is nature, and both features are good enough to train a classifier and are conditionally independent given the class. There are the steps of the algorithm. Each classifier is retrained with the additional training examples given by the other classifier, and the process repeats.

- Input: feature set $x^{(1)}, x^{(2)}$, labelled data: X_l, unlabelled data: X_u
  Repeat
- Train classifier $f^{(1)}$, $f^{(2)}$ by X_l based on $X^{(1)}$, $x^{(2)}$ respectively
- Use $f^{(1)}$ and $f^{(2)}$ to classify X_u
- Select k most confident predictions from X_u and assign labels
- Add $f^{(1)}$'s prediction with label $(x, f^{(1)}(x))$ to $f^{(2)}$'s labelled data, $f^{(2)}$'s prediction with label $(x, f^{(2)}(x))$ to $f^{(1)}$'s labelled data
- Delete these data from unlabelled data X_u
  End until there is no unlabelled data

- Select the unlabelled data commonly agreed by two classifiers and add them to final training set (with their agreed labels as labelled data)
- Train the LRC with final X_l
- Output: f

2 b

2.1 Experiment setting

The experiment dataset consists 10000 samples on 11 dimensions. The number of unlabelled data is set as $\{0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096\}$ with 8 labelled data per class in training process. I repeated my experiment 200 times on both semi-supervised learning method mentioned above and the supervised learning method I applied as baseline. On testing phase, 500 samples per class were chosen which are untouched and fixed (each repeating keeps same test data excluding training data). The error rate of LRC (Linear Regression Classifier) is defined as:

$$error\_rate = \frac{1}{m}\sum_{i=1}^{m}\frac{number\ of\ misclassified\ test\ samples}{total\ number\ of\ test\ samples} \tag{1}$$

where m is the experiment repeating times, fixed in 200 in experiment. The Expected squared Loss function is defined as:

$$\frac{1}{N}\sum_{i=1}^{N}(< w, x_i > +w_0 - y_i)^2 \tag{2}$$

where N is number of total testing samples, w and w_0 are trained parameter of linear function, x_i and y_i are feature vectors and labels of samples respectively.

2.2 Result analysis

Figure 1 illustrates the expected true error rate curve. The trend for all these three methods is that they perform well with small training data, but with the increment of training data size, all these methods performs worse in some certain phase, or say, there is significant changes from range $[32, 128]$. When training size is larger than 128, the errors for all methods hold on a high level (over 0.45). Another thing to say is that all curves' trend is not monotonous. Error rate of self-training model, supervised learning model and co-training model decrease in range $[4, 32]$ $[2, 4]$ and $[4, 16]$ respectively. Last but not least, in small training size, co-training model performs best well self-training model performs worst, our baseline supervised learning model is in between, well in large training size, supervised learning model performs better than both semi-supervised learning methods although itself has a high error rate (0.47), yet error rate of two semi-supervised learning reach around 0.49 after training size is larger than 256.

The reason why two semi-supervised learning methods perform badly, in my point of view is that in linear regression progress, once model start to have bias from randomly selected data, the situation become worse and worse, because best fitting points were selected and added into model, bias model would add bias samples as preference, after long iteration, the model becomes worse and

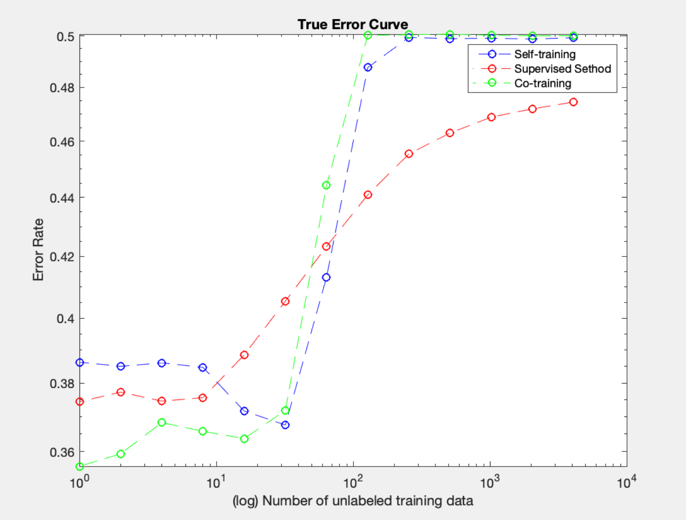

Figure 1: True error rate curve

worse. And the reason for all models have increasing error rate is that with adding more and more training samples linear function would have more and more outliers which influences the models, the more data there are, the less accurate it is.

3 c

The expected squared loss function and experiment setting is described in part b above.

Figure 2 illustrates the expected squared loss versus the number of unlabelled data. Firstly and briefly give my result that there are significant changes. The main trend is that the expected squared loss decrease significantly in certain ranges. For Self-training, it decrease on $[4, 64]$ and for both supervised method and co-training model, they decrease in the beginning. For two semi-supervised learning methods, they have significant decrease but not monotonous, both end when training size is 64, and then increase slightly and converge to a certain value about 2. For supervised learning, the value decreasing is monotonous until it converges to 1.

The reason for all model have decreasing loss, in my point of view is that with the increment of training size, more data correspond to better linear regression, leading that the increment scale of loss is lower than increment scale of number of samples, so it leads to decreasing of expected loss.

4 d

4.1 Self-train is better than Co-train

I constructed a Gaussian dataset with two class, 1000 samples per classes with $\mu_1 = (0, 0)$ and $\mu_2 = (3, 0)$ with equal covariance $\sigma = 1$, shown in Figure 3. The result is that self-training model is better than supervised learning and supervised learning is better than co-training model.

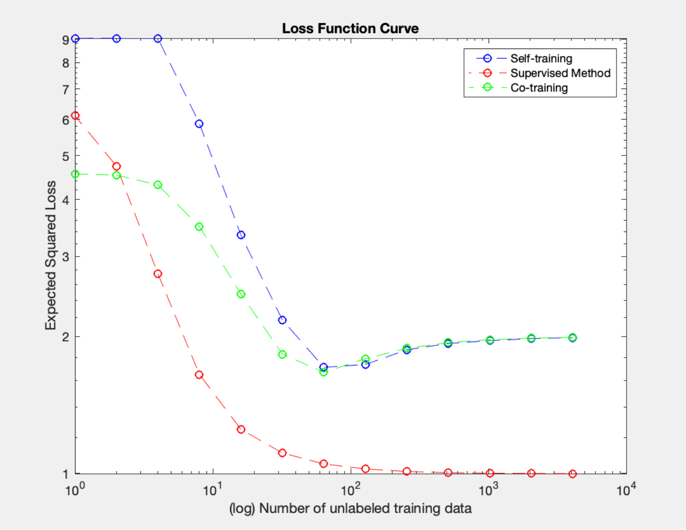

Figure 2: Loss function curve

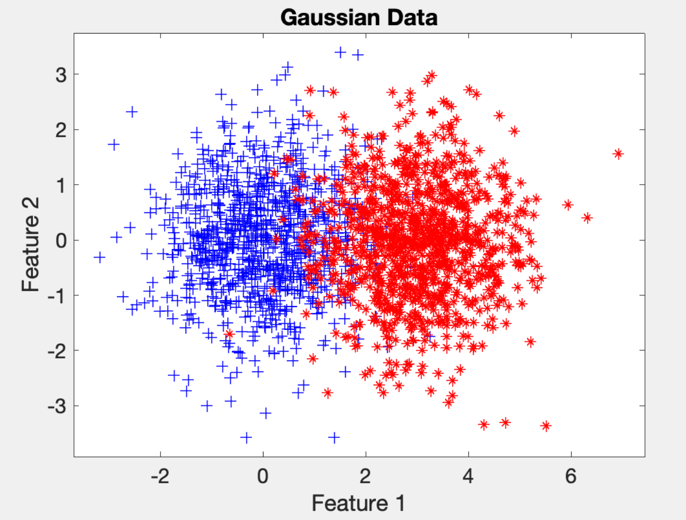

Figure 3: Loss function curve

The reason is that there there is an inseparable dimension in this dataset, thus for co-training model, when features are separated, there is a feature set which is inseparable. On this dimension (feature), one of the co-training classifier would work randomly, the result on this feature is rather bad and it influences the label assigning process. Much few data are selected for the final training set, in which more wrongly labelled data (compare with self-training) are included. For another reason, the assumption is that there are multiple view of co-training, which is not fulfilled by my dataset.

4.2 Co-train is better than Self-train

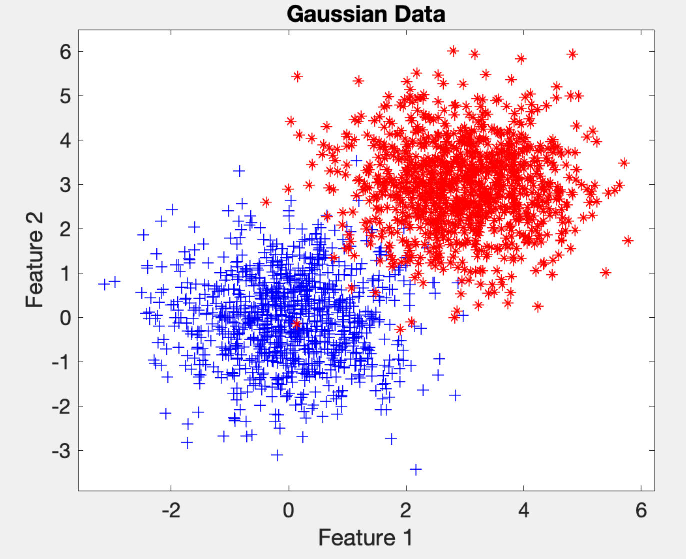

Figure 4: Loss function curve

Then I constructed another Gaussian dataset with two class, only changing parameter $\mu_2 = (2,2)$, shown in Figure 4. Here it fulfilled that co-training is better than self-training. Because two features are separable which convinces each classifiers and better samples are picked out, where constrained by the ”bias leads more bias” mentioned in part 2, self-training works worse than co-training.

4.3 A short conclusion

Generally, based on the property of self-training and co-training algorithm, for 2 dimension situation, when at least one dimension is inseparable, self-training works better than co-training, on the contrary, when two dimension are more separable, co-training works better than self-training.

References

[1] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, 1995.

[2] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.