IN4320 Machine Learning Exercise: Semi-Supervised Learning

Yadong Li(4608283)

March 28, 2017

1 Two ways of semi-supervised learning

1.1 Expectation Maximization LDA

This algorithm basically consists two steps: E step and M step. In E step, we calculate the posterior probability for unlabeled data and assign soft labels accordingly. In M step, we update the parameters using the estimated soft label obtained in E step. And we repeat E step and M step until converge. The algorithm can be described as follows:

1. **Initialize** $\theta : \{p(c_1), p(c_2), \mu_1, \mu_2, \Sigma\}$ from labeled data

   Repeat the following steps until Log Likelihood doesn't change:

2. **E-step**: calculate the posterior probability for unlabeled data and assign soft labels

   $p_\theta(c_1|x^u) \leftarrow \frac{p(x^u|c_1)p(c_1)}{p(x^u)}$

   $p_\theta(c_2|x^u) \leftarrow \frac{p(x^u|c_2)p(c_2)}{p(x^u)}$

3. **M-step**: update θ

   update $p(c_1)$, $p(c_2)$

   update μ_1, μ_1

   update μ_2, μ_2

   update Σ

1.2 Co-training

The basic idea behind this algorithm is that we can let two classifiers to teach other. We assume that there exists a feature split $x = [x^{(1)}; x^{(2)}]$ which can provide two different views of the data, and these two views are both capable of successfully performing the classification task individually. After these assumption, here is how it works:

1. Use cross validation to separate the feature into two views f^1, f^2.

   Repeat until there is no unlabeled data

2. Train two classifiers using labeled data.

3. Use these two trained classifiers to classify the unlabeled data.

4. Select the K most confident predictions in each classifier and assign labels to these unlabeled data.

5. Add these predicted unlabeled data to each other's training set.

6. Delete them from unlabeled data.

   End Repeat

7. Select the unlabeled data whose predicted label form two classifiers agrees and add these samples with predicted labels to the training set.

8. Train a LDC classifier using the enriched training set.

It's worth to mention that I have chosen the AdaBoost classifier I built for last assignment as base classifiers here. The reason to choose it is that first it is written by myself so I can easily mortify it to suit my need for this assignment. Also, it can provide confident level when predicting, and this feature is great for this co-training algorithm.

2 Testing on *spambase* dataset

2.1 Experiment setup

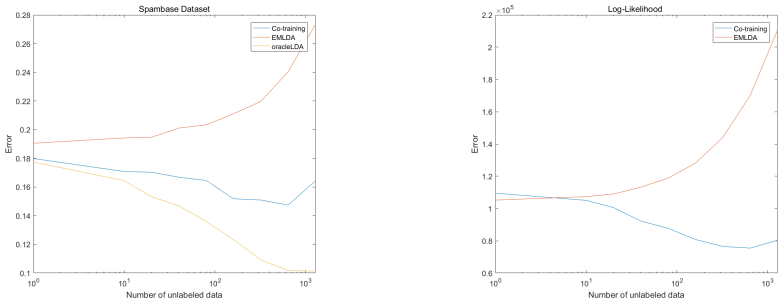
The *spambase* dataset consists 4601 observations with 57 features. In this experiment, a total of 75 labeled samples are in the training set per class. And 0, 10, 20, 40, 80, 160, 320, 640, 1280 unlabeled samples per class were added in different trails. In the experiment, a total number of 450 testing samples per class were kept untouched and fixed, which means that all the experiments were tested using the same untouched testing set.This procedure was repeated 30 times and the average error and standard deviations were obtained. To compare the result, a supervised LDA trained on the 150 labeled samples and a supervised LDA trained on both labeled and unlabeled samples() were tested on the same testing set described before. This classifier is referred as "OracleLDA".
A log-likelihood versus the number of unlabeled data was conducted as well. The log-likelihood is calculated as follow,

$$LnL(\theta|x_1, x_2, ..., x_n) = \sum Lnp(x_k; \theta)$$

where $x_1, x_2, ..., x_n$ are testing samples and θ represents all the parameters obtained in training phase.

2.2 Results

The learning curve for EMLDA and Co-training can be found in Figure 1a.
It reveals that EMLDA didn't perform well on this dataset, as it error rate went up as more unlabeled data is added. On the contrary, Co-training performed better on it, as the error rate went down at first, and when the number of unlabeled data reached 640 it went up a little. The learning curve for increasing amount of unlabeled data against the (negative) Log-Likelihood is shown in Figure 1b.



(a) against the error rate. (b) against the (negative) Log-Likelihood.

Figure 1: Learning curves for increasing amount of unlabeled data against the error rate and against the (negative) Log-Likelihood.

Table 1 shows the standard deviation of error rate for different semi-supervised classifiers over 30 REPEATS.

#unlabeled data	0	10	20	40	80	160	320	640	1280
EMLDA	0.024	0.023	0.025	0.024	0.031	0.029	0.028	0.032	0.034
Co-training	0.024	0.023	0.025	0.021	0.027	0.014	0.023	0.022	0.017

Table 1: Standard deviation of error rate for different semi-supervised classifiers over 30 REPEATS.

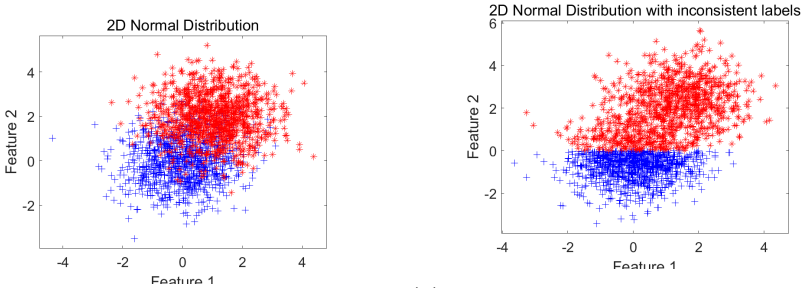
3 Toy dataset and discussions

3.1 Experiment setup

To illustrate the behavior of these two different semi-supervised learning algorithms, two toy dataset were built and they behaved differently.
The settings for the experiment is very similar to the previous one. A total of 75 labeled samples are in the training set per class. And 0, 10, 80, 160, 320, 640, 1280 unlabeled samples per class were added in different trails. In the experiment, a total number of 1000 testing samples per class were kept untouched and fixed, which means that all the experiments were tested using the same untouched testing set. All experiments were repeated 20 times to get reliable results.

3.2 2D Gaussian distribution

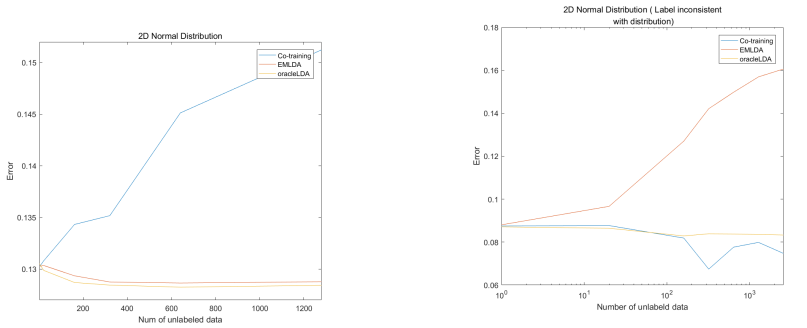
In the first case, we have a multivariate normal distribution with $\mu_1 = [0, 0]^T$ and $\mu_1 = [1, 2]^T$ and equal covariance $\Sigma = 1$. The scatter plot for this dataset is shown in Figure 2a. For this dataset, the EMLDA algorithm is expected to perform better than a supervised classifier, and the Co-training algorithm is expected to perform worse than the supervised. The experiment results matched with our expectation, as shown in Figure 3a. Detailed discussion will shown in Discussion section.



(a) Dataset1: 2D Gaussian distribution.

(b) 2D Gaussian distribution with labels inconsistent with distributions.

Figure 2: Two toy datasets.



(a) Dataset1: 2D Gaussian distribution.

(b) 2D Gaussian distribution with labels inconsistent with distributions.

Figure 3: Learning curve for two toy datasets.

3.3 2D Gaussian distribution with labels inconsistent with distributions

Here two Gaussian Distributions are created as well: $\mu_1 = [0, 0]^T$ and $\mu_1 = [1.5, 2.5]^T$ and equal covariance $\Sigma = 1$. But here, labels were assigned not consistent with distributions. We assigned all samples whose feature 2(y-axis) greater than 0 as class 1, and less or equal to 0 as class 0, as shown in Figure 2b. For this dataset, we are expected to observe EMLDA going worse and Co-training going better. The experiment results verified our expectation, as shown in Figure 3b. In next section, we will discuss the results.

3.4 Discussion

3.4.1 EMLDA

For EMLDA algorithm, it assumes that the data in each class actually comes from a Gaussian distribution. unlabeled data is assigned with soft labels to help the classifier to better estimate the class means and covariance. Therefore, if the assumption holds true and the data in a certain class really comes from a a certain Gaussian distribution, then of course, adding more unlabeled data does provide more "correct" information, and it can help to improve performance. This explains why in the first dataset, EMLDA helps to improve performance with unlabeled data.
However, if the assumption is wrong, the unlabeled data will be in high danger of wrongly labeled during EM steps, and these "wrongly" labeled samples will certainly harm the classifier and make the prediction worse. This may explain why in the second dataset, where the assumption is totally not true, the performance of EMLDA gets worse and worse when adding more unlabeled data.

3.4.2 Co-training

For co-training algorithm here, it also makes several assumptions. First, it assumes that there exists a feature split $x = [x^{(1)}; x^{(2)}]$ which can provide two different views of the data. Second, it assumes that these two views are both capable of successfully performing the classification task individually. For the first dataset, I noticed that a large proportion of the labels predicted by two co-trained classifiers were wrong during the experiment. This might be explained the overlaps between two classes, which makes classifiers hard to separate two classes. If the predicted "hard" labels are wrong at a large scale, it is not surprising to observe worse performance when adding unlabeled data.
However, if the dataset is separable, like the second dataset, the two classifiers in co-training algorithm can be very confident to predict labels and teach each other to learn from unlabeled data. Therefore, in the second dataset, co-training performs much better when adding unlabeled data.

3.4.3 Final thoughts

I have to admit that this is a very challenging exercise. I've spent a lot of time exploring, experimenting, tuning parameters and debugging. And I also learned a lot from it. The core message for supervised learning I learned is that unlabeled data can be useful, but we need to look closely at the dataset first and find out a suitable algorithm. A suitable algorithm can improve the result, but a unsuitable one may cause worse result.