IN4320 Machine Learning Exercise 2

Xiang Teng (4574060)

March 13, 2018

1 Question a

1.1 Method 1 (Generative methods)

This algorithm is a combination of Maximum likelihood (ML) and Bayes classifier (BC). BC is used to search the unlabeled dataset and find the one that has the highest probability to belong to one of the classes. So it could be labeled. ML is used to updated the parameters θ according to the new labeled dataset. Therefore, the whole process could be divided into two steps. S step stands for the candidate selection and U step stands for the model parameters update. Now the algorithm could be described as follows:

1. **Initialize θ:** $\{p(c_1), p(c_2), \mu_1, \mu_2, \Sigma_1, \Sigma_2\}$ from labeled data

Repeat step 2 and 3 until all unlabeled data are labeled

2. **S-step:** Select the candidate with the highest probability to belong to the one of our classes and give it the same label with that class. The equation 1 is used to find the posterior probability for each unlabeled sample.

$$p(\Theta = i|x_u) = \frac{\alpha_i \cdot p(x_u|\mu_i, \Sigma_i)}{\sum\limits_{i=1}^{N} \alpha_i \cdot p(x_u|\mu_i, \Sigma_i)} \tag{1}$$

3. **U-step:** Update the model parameters θ with taking the new labeled dataset into consideration.

1.2 Method 2(TSVM)

At the beginning, A SVM classifier is trained by using the available labeled dataset. And then, all unlabeled dataset are assigned with labels by using this classifier. Now, we can use all labeled dataset to train our SVM classifier again. After, the labels of the dataset with a high probability to be classified wrong are switched. For example,

$$\begin{aligned} &\textit{while} \quad \exists\{i,j|(\hat{y_i}\hat{y_j} < 0) \wedge (\xi_i > 0) \wedge (\xi_j > 0) \wedge (\xi_i + \xi_j > 2)\} \quad \textit{do} \\ &\qquad \hat{y_i} = -\hat{y_i} \\ &\qquad \hat{y_j} = -\hat{y_j} \end{aligned}$$

where $\hat{y}$ are the assigned labels for unlabeled dataset and ξ is the relax parameter. Here, we assume that the larger the relax parameter ξ is, the closer the unlabeled dataset from the hyperplane, and so, the more easy to make mistake. In addition, two other compromise parameters C_l, C_u are introduced. At the beginning the value of C_l is far more larger than C_u and the function of labeled data is far more important than the unlabeled data in this period. And then, increasing C_u until it has the same value with C_l. The whole algorithm could be described as follows:

$$\begin{aligned} \min_{\omega,b,\hat{y},\xi} \quad & \frac{1}{2}||\omega||_2^2 + \mathrm{C_l}\sum_{\mathrm{i=1}}^{\mathrm{l}} \mathrm{\xi_i} + \mathrm{C_u}\sum_{\mathrm{i=l+1}}^{\mathrm{m}} \mathrm{\xi_i} \\ \text{subject to} \quad & y_i(\omega^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, ..., l, \\ & \hat{y_i}(\omega^T x_i + b) \geq 1 - \xi_i, \quad i = l+1, l+2, ..., m, \\ & \xi_i \geq 0, \quad i = 1, 2, ...m, \end{aligned}$$

1. **Setp 1:** Train SVM_l using available labeled dataset D_l
2. **Step 2:** Assign labels for all unlabeled data according to the current SVM_l.
3. **Step 3:** Initialize $C_u \ll C_l$

Repeat the following steps until $C_u \geq C_l$

4. **Step 4:** Correct the dataset which are more likely to be misclassified.
5. **Step 5:** Update the value of $C_u = min\{2C_u, C_l\}$
6. **Step 6:** Retrain the SVM classifier, obtain the new value for parameters $(\omega, b), \xi$

2 Question b & c

Two methods were implement in this section and the relevant results are shown. Figure 1 shows the learning curves while two methods were implemented. Is is shown that the TSVM method has a better result than generative method for this dataset. Also, the error rate for both semi-classified methods decrease while the number of unlabeled data used for training increase. The test dataset for both semi-classifier are the whole `MAGIC Gamma Telescope Data Set`. Besides, we can see that there is an increasing in error rate while a small of unlabeled data is introduced for training for TSVM.

Figure 2 shows the relation between log likelihood and the number of unlabeled dataset. As the more the number of unlabeled data introduced, the value of the log likelihood increase. Figure 3 shows the log likelihood for TSVM. The same situation happens. The reason why I am not putting them together is because their difference in scale. There is a decrease in log likelihood while the number of unlabeled data is small which is matched with the increase in the error rate for TSVM.

If we compare with the results of the experiments under the supervised error rates which are 0.3053 and 0.2698 for GM and TSVM respectively, it less than the error rate while only a small amount of unlabeled data introduced for both classifier.

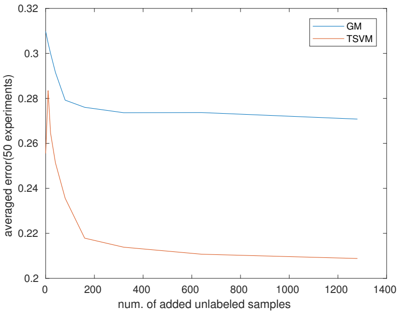


Figure 1: learning curves against the number of unlabeled samples for a total of 25 labeled samples in the training set

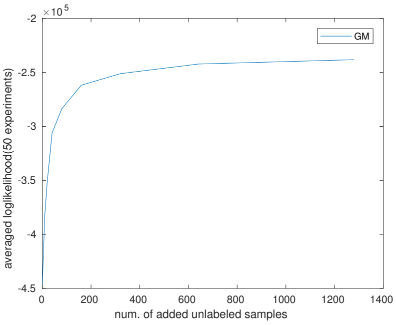


Figure 2: the averaged log likelihood versus the number of unlabeled samples for a total of 25 labeled samples in the training set for generative method

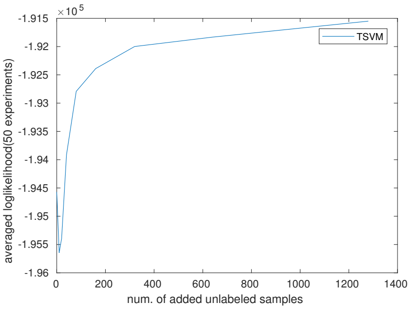


Figure 3: the averaged log likelihood versus the number of unlabeled samples for a total of 25 labeled samples in the training set for TSVM

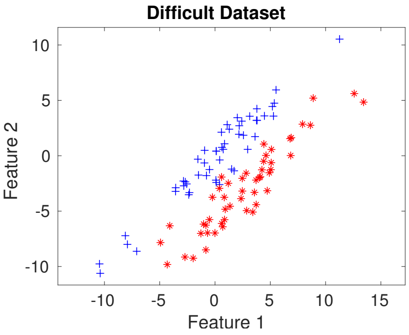
3 Question d

Figure 4 and 5 shows two simple data sets in 2D, The dataset in Figure 4 are in Gaussian distribution and Figure 5 shows the data set in a banana shape distribution. This is just an example in 2D for 2 data sets each. It could also be generalized in a higher dimensions and in more than 2 classes.

GM assumes that the datasets follow a certain kind of distribution. In this case, we assume the class-conditional distributions to be Gaussian with the same covariance matrix. Therefore, if the dataset is really follow this distribution such as the one in Figure 4. The GM method will perform better than TSVM. However, if the assumption is false, such as the distribution in Figure 5. TSVM will perform better.

In more detail, for GM method, the condition about which class an unlabeled data should belong depends on its prior probability related to the classes' mean and covariance. So the dataset in one of the class mostly will have a higher probability if the calculation is based on the class' mean and covariance. However, this is not true for TSVM. It finds the gap with the lowest density to separate the data. It is most likely for it to classify the dataset in Figure 4 as left-down and up-right parts which is obviously wrong.

Nevertheless, if the dataset is not normal distribution such as the situation in Figure 5. TSVM performs may performs better.



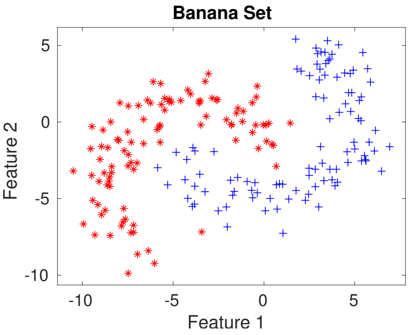


Figure 4: data sets with Gaussian distribution

Figure 5: data sets with Banana shape distribution