Assignment #4

Dr. Rajati

Muxin Liang

Contents

Exercise # 1

Active Learning Using SVM

(a) Dataset

Download banknote data. Randomly selected 472 as test data.

(b) Passive Learning And Active Learning

I used R for this homework and the package I used for generating SVM are *caret* and *LiblineaR*.

The best cost was chosen from the list:

$$C \in \{1000, 100, 10, 1, 0.1, 0.01, 0.001\}$$

by 10-fold Cross-validation.
The test error in every iteration will be stored in 1 block of a 90×50 dataframe.(named as df1.csv and df2.csv for passive and active learning, respectively).

For active learning, to get the closest 10 samples to the margin, I access the coefficients of the model and calculate the inner product between them and every piece of data and then add up with the intersect and then take the absolute value:

$$\text{Minimum 10 values of } |\sum \beta_{1,2,3,4...} \times x_{1,2,3,4...} + \beta_0|$$

Detailed code in Jupyter Notebook.

(c) Monte Carlo Simulation

The following figures are the generated results from Monte Carlo Simulation.

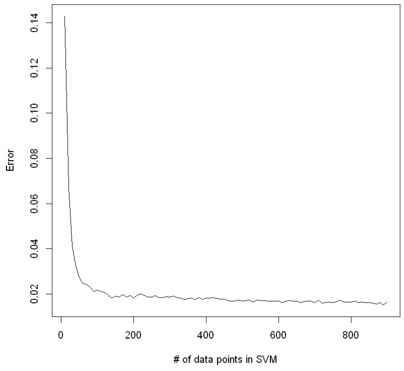

Figure 1: Passive Learning result

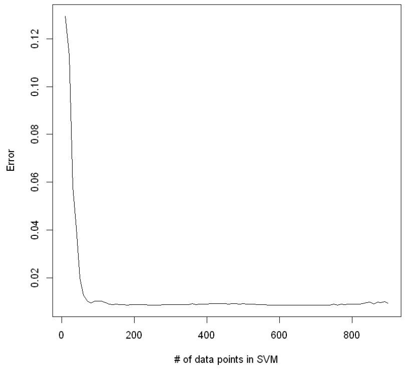

Figure 2: Active Learning result

From observation, we can see that active learning will give out a better accuracy when the size of dataset is getting bigger and bigger. It also performs better than passive given small dataset. It is the same as my assumption.

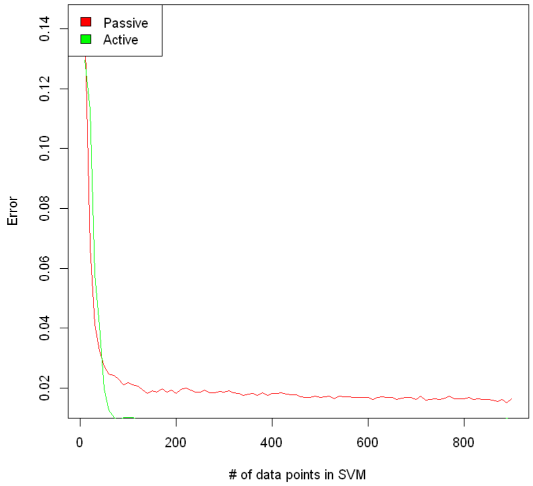

Figure 3: Combination of two curves

Problem 2

Multi-class Classification Using SVM

(a) Dataset

Downloaded dataset and choose 70% of data randomly as train.

(b) Multilabel

1. Exact Match and Hamming Loss:
   Exact Match: We consider the partially correct predictions as incorrect and only treat those that are completely correct predictions as correct and calculate the exact match rate:

   $$R = \frac{1}{n}\sum_{i=1}^{n} I(Y_i = Z_i)$$

   Where $I(Y_i + Z_i)$ states that an observation in prediction that is completely correct.

   Hamming Loss: Hamming Loss reports how many times on average that the relevance of an example to a class label is incorrectly predicted. It takes prediction error and missing error into account over total number of classes and total number of examples:

   $$HL = \frac{1}{kn}\sum_{i=1}^{n}\sum_{i=1}^{n} I(Y_i \neq Z_i)$$

Where k is the number of class and n is number of data.

2. Train a SVM for each of the labels, using Gaussian kernels and OVA.
   Since we are doing One-versus-All, we will generate classifier for each label, the generated result will be shown as charts. It shows the best-tuned results and their test error rates for each labels (Width is calculated by $\gamma = \frac{1}{2\sigma^2}$).

   The best cost was chosen from the list:

$$C \in \{1000, 100, 10, 1, 0.1, 0.01, 0.001\}$$

   by 10-fold Cross-validation.

	Gaussian(Not normalized)		
	Width	Cost	Error Rate
Family	184.85	1	0.012
Genus	212.98	1	0.018
Species	190.65	1	0.017

	Gaussian(Normalized)		
	Width	Cost	Error Rate
Family	182.81	1	0.011
Genus	217.73	1	0.018
Species	194.04	1	0.016

   The result shows that Normalization will give a little boost in accuracy of Gaussian Kernel SVM.

3. L1-penalized SVM using normalized data.
   The chart shows best tuned parameter and error rate.
   The best cost was chosen from the list:

$$C \in \{1000, 100, 10, 1, 0.1, 0.01, 0.001\}$$

   by 10-fold Cross-validation.

	L1-penalized	
	Cost	Error Rate
Family	100	0.011
Genus	10	0.004
Species	100	0.003

   The result shows that L1-penalized SVM gives better accuracy comparing to Gaussian.

4. SMOTE and Gaussian on Normalized Data:
   I used package DMwR to do SMOTE for normalized data and put then into Gaussian kernel.

   The chart below shows that best-tuned results for each label.

	Gaussian(Normalized)		
	Width	Cost	Error Rate
Family	212.45	1	0.014
Genus	200.74	1	0.022
Species	210.23	1	0.013

We can see there is a small decline in accuracy after SMOTE is applied. But we can not say it is worse because SMOTE resampled the data and the model's performance may dependent from what kind of data is given.