EE6550 Machine Learning, Spring 2016

Homework Assignment #2 Report

102061210 王尊玄

1. Iris_set_ver :

    This sample can be well classified with both inner product kernel and Gaussian kernel. Also, since the sample size is solely 67, which is pretty small, I didn't apply cross validation over this sample.

    Inner product provide 100% correctness over testing data, and so does Gaussian kernel with different variance.

inner product kernel correctness			
C	1.2	1.7	2.1
	1	1	1

Gaussian kernel correctness					
C\sigma	1.4	1.7	2.1	2.5	2.7
1.2	1	1	1	1	1
1.7	1	1	1	1	1
2.1	1	1	1	1	1

2. Adult :

    Size of adult training sample is pretty large, 30162. Cross validation is applied over this sample. Besides, It takes too long to implement k'th fold cross validation. Thus, I adopted alternative method for cross validation.

    I randomly pick a subset of entire sample as training data, with data outside the subset as validation data. Also, I further break the training subset into m mini-sets in order to speed up my program. I apply SVM training algorithm over each mini-sets, for each is validated according to entire validation data, to obtain m correctness. Those correctness is averaged to produce final correctness, representing a particular training data (subset of entire sample). This method allows number of iteration while doing cross validation can be dramatically reduced, and randomly picking training data (and validation data as well) can, in some degree, represses bias over certain dataset. Note that I think k'th fold cross validation still provides better guarantee.

    With subset size = 10000, size of a mini-set = 1000, and number of mini-set = 10, the following table shows result of cross validation,

inner product kernel correctness		
C	1.2	1.7
	0.8209	0.8202

With subset size = 20000, size of a mini-set = 2000, and number of mini-set = 10,

C\sigma	1.4	1.7	2.1	2.5	2.7
1.2	0.8248	0.8247	0.8228	0.8217	0.824
1.7	0.821	0.8233	0.8227	0.8294	0.8251

After I've tried over all pairs of free parameters (slack penalty C and variance of Gaussian kernel sigma), I can then use certain combination of C and sigma (or C only if using inner product kernel) to run SVM training algorithm over entire training sample, generating final Lagrange multipliers λ and shifting value b of SVM method. Finally, we can apply λ and b to predict on testing data, and check effectiveness of our SVM classifier over adult sample.

I have tried C = 1.7, sigma = 2.5 with 10000 samples (adult_training(1:10000)) on my SVM training algorithm, generating Lagrange multiplier and shifting value, which can be obtained from file best.m and provides 83.65% correctness over testing data.