**BENG 420/590: Homework #5 SVM**

**Due: Friday, 12/7/2018, by 11:59pm**

*Same as previous homework, you can work in a group of two if you prefer. Submit this assignment as a Word or PDF document through Blackboard. Please also submit your commented Matlab code.*

1. **SVM classifier on microarray data**

The Notterman Carcinoma Data in “CarcinomaNormalDataset.xls” contains microarray data from 18 tumor patients and 18 healthy controls. The data were used for the research presented in Notterman et al, “Transcriptional Gene Expression Profiles of Colorectal Adenoma, Adenocarcinoma, and Normal Tissue Examined by Oligonucleotide Arrays”, *Cancer Res* April 1, 2001 *61;* 3124.

1. The precompiled Windows library files of *libsvm* can be found on blackboard. Depending on your system config as either being 32-bits or 64-bits, you need to download the proper files to use the svm library. Save the files in the folder where you will program this homework assignment. More information on the libsvm function utilization can be found here: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

If you have a Mac, you may have to google proper ways to install the package.

1. Download “CarcinomaNormalDataset.xls” and open it in Excel. There are a lot of features in this microarray data. **(1 point)** Pick 10 features (as rows) that show **strong (small) p-values.** The p-value of each feature is given in the last column “T-Test tumor vs. normal” in the spreadsheet. These features have stronger capability in distinguishing patient instances from healthy instances. **(2 points)** Train two SVM classifiers, a linear one and a nonlinear one with Gaussian kernel, to distinguish between patients and healthy controls. You can either use the default values for the Gaussian kernel parameters (gamma and C) or optimize them.
2. **(6 points)** Given the limited number of instances, use cross validation to evaluate performance of your classifiers. Experiment with the leave-1-out strategy and leave-10-out strategy.
3. **(2 points)** Discuss the performance difference between the linear SVM classifier and the Gaussian SVM classifier.

When using low p-values, linear SVM classifier performs better with a significantly lower sum-squared-error than gaussian SVM. One of the possible reasons for this is because gaussian might be overfitting. In this case, a simple kernel such as linear is preferred. In addition, with higher p-values, both linear and gaussian kernels performed poorly. The reason for this is because features with higher p-values have weaker capability in distinguishing patient instances from healthy instances.

1. **(2 points)** Discuss what you observe on the classifier performance using the two cross validation choices (leave-1-out and leave-10-out). Whether there is any difference between the two or not, explain whether that is what you expected and why.

When using low p-values, leave-1-out and leave-10-out did not change the sum-square-error in the linear kernel. Gaussian kernel, however, saw a large rise in sum-square-error when using leave-10-out. This is expected as leave-1-out gives a more precise classification due to only having to classify one data point at a time.

1. **(4 points)** Now pick ten features that show weak (large) p-values larger than 0.1 and repeat (b)-(d). Compare and discuss classification using these features is different from classification using features of strong p-values.

The same is seen in high p-values as sum-square-error increases when using leave-10-out as oppose to leave-1-out. High p-values doesn’t have as high of a capability to distinguish between healthy instances and unhealthy instances as compared to low p-values. Therefore, that results in a higher sum-square-error.

1. Summarize your results in the following table. Submit your **discussions** and Matlab code.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Low p-values (sum-square-error) | | High p-values (sum-square-error) | |
|  | Leave-1-out | Leave-10-out | Leave-1-out | Leave-10-out |
| Linear SVM | 0 | 0 | 1 | 5.9444 |
| Gaussian SVM | 1 | 5.6944 | 1 | 5.6389 |

1. **SVM classifier on diabetes**

The paper “Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes” describes the application of SVM on diabetes prediction. The authors were kindly to share the data for us to use in this assignment.

1. A subset of the training and testing data used in the paper for Classification Scheme I were given in ‘train.txt’ and ‘test.txt’. They are already in the libsvm format that can be loaded into Matlab by using the “libsvmread” command.
2. **(4 points)** Experiment with four different kernels, linear kernel, polynomial kernel, radial basis function (Gaussian) kernel, and sigmoid kernel. Build a classifier using each of the four kernels and report training accuracy and testing accuracy in a table. You can use the default kernel parameters. Table 2 in the paper presented their comparison and concluded that Gaussian kernel performs the best and polynomial performs the worst. Sigmoid and linear kernels were just a little bit behind Gaussian kernel. **(2 points)** Do you observe the same order of performance with your predicted testing accuracy outcome? Please explain.

|  |  |  |
| --- | --- | --- |
|  | training accuracy (%) | testing accuracy (%) |
| linear | 74.6 | 73.95 |
| poly | 71.2 | 66.59 |
| gaussian | 73.9 | 72.13 |
| sigmoid | 66.3 | 70.15 |

Unlike in the literature, the linear kernel slightly outperformed the gaussian kernel, followed by the polynomial kernel and the sigmoid kernel. However, in concurrence with the literature, the performance of linear, gaussian, and sigmoid are fairly close to each other (with regards to the testing accuracy).

1. **(3 points)** Now let us examine the Gaussian kernel. Build two classifiers with the following two sets of parameters: (1) gamma=10000, C = 1, (2) gamma = 10, C = 2000. Describe what you observe from comparing these two classifiers and explain why one is better than the other.

Classifier1: 30.72% for gamma = 10000 & C = 1

Classifier2: 70.07% for gamma = 10 & C = 2000

Classifier1 is overfitted with the training data, hence why it did not have a high testing accuracy. This is evident in the training accuracy being 100% and 90.55% for Classifier1 and Classifier2 respectively. The overfitting is caused by the high gamma value in Classifier1.

1. **(5 points)** Optimize the Gaussian kernel parameters. Use a double loop to find the “optimal” gamma and C within these ranges: gamma = 0.1:0.5:10, C = 0.1:0.1:2. The optimal gamma and C will lead to the most accurate classification on the testing data. Report your gamma and C. Describe how sensitive the training accuracy and the testing accuracies are to gamma and C for this data and explain why. (Hint: think about the performance of the linear classifier)

Optimal gamma = 9.6

Optimal C = 2

Training accuracy = 79.03%

Testing accuracy = 72.92%

Training accuracy and testing accuracy is fairly sensitive to gamma and C. The reason for this is because gaussian kernels can be overfitted, which results in over 6% difference between the training and testing accuracy.

1. **(5 points)** Train a neural network using the training data and test the classifier with the testing data. You can decide what neural network architecture to use. Report the accuracies of the classifier on the training data and the testing data in two confusion matrices. How does your neural network classifier perform compared to SVM? Please discuss.

The neural network performed much worse than the SVM. This is probably due to the data set being linearly inseparable. If the data set is linearly inseparable (which it most likely is), then it is hard for the neural network to properly classify the data points because the neural network is a linear classifier. However, since the SVM is a nonlinear classifier, it can classify the data points much better than the neural network.