Pattern Recognition Homework 2

Lev Kolezhuk

June 2017

1 Task 2.1

[60%] Consider the dataset contained in the file hw1data.mat available on the course site. It contains 8,000 samples coming from a two–class problem, each made of 10 numerical features and a binary label (1). Use the Matlab function inmodel = sequentialfs(fun,X,y) to perform a sequential feature selection by using a filter ap-proach. Split the data into training and test sets by randomly selecting 25% of the examples from each class for the test set and use the training set to select the features. Assume as the objective function for the feature selection the accuracy obtained by a 1-NN classifier.

Consider the following tasks:

(a) Adopting a Sequential Forward Selection, select the subsets made of 5 and 8 features.

(b) For each feature subset, split the data into a training set (40%), validation set (40%) and test set (20%).

(c) Use the training set for building a Support Vector Machine and the validation set for choosing the optimal value of its parameters (i.e. those maximizing the AUC).

(d) Evaluate the AUC on the test set. Execute the steps above with two different kernels of the SVM (linear and Polynomial of degree 2). For each test, consider at least 5 different splits and evaluate average AUC and standard deviation. For the sake of comparison, perform the same steps also with all the original features of the data set. Discuss the results obtained.

Hint: The objective function used by sequentialfs should be implemented by a Matlab function of the form criterion = fun(XTRAIN,ytrain,XTEST,ytest), where XTRAIN and XTEST contain feature vectors and ytrain and ytest contain the corresponding class labels. The function should implement a 1-NN classifier that uses the XTRAIN,ytrain as reference test for classifying the samples in XTEST. The value that the function should return is the accuracy evaluated by comparing the predicted labels with the true labels contained in ytest.

1.1 Solution

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

In order to implement the algorithm in MATLAB, we are using the set of SVM functions for training the machine and for actual classification of samples : "svmtrain", "svmclassify".

However, in order to achieve the stated goal, we have to build a function that splits our dataset randomly into 2 sets(training and test) and a similar function that splits into 3 subsets(training, validation and test sets). These functions accept as input the initial dataset as well as the required percentage for each subset.

The second step is to select only some of the existing features from the acquired subsets. This feature selection is needed to work with the most significant features of the subset. This means that it can happen that some features are completely random and don't really contribute to the accuracy of classification, so there is no need to use them for comparison of samples. In order to perform this a matlab function "sequentialfs" is used, that determines the most significant n features from the data based on a provided classifier. In our case we use the KNN classifier for $n = 1$. As mentioned in the task, we extract 5 and 9 features from our initial dataset.

For both polynomial and linear kernel we validated our results on validation datasets, after training the SVM on the test set for different number of features. For each of the cases optimal parameters have been selected such as the box-constraint parameter of the SVMTrain is a soft margin, which allows us to change the rate of error we accept. The biggest problem is that if we set it really small, the algorithm will not be able to converge within a normal number of iterations. It means we have either to set the maximum limit for the iterations really high or change the parameters, that correspond to the optimization of the algorithm such as 'kktviolationlevel', 'tolkkt'. The results for the AUC depending on the kernel type and the number of selected features can be seen in the table below:

Table 1:

Polynomial kernel		
Features	**AUC**	**Standard deviation**
5	0.8305	0.0081
8	0.828	0.0044
10	0.8351	0.0039

Table 2:

Linear kernel		
Features	**AUC**	**Standard deviation**
5	0.774	0.0076
8	0.78564	0.0060
10	0.78	0.0054

The result was also validated on the test set with the best chosen configuration, that allowed us to achieve AUC of 83.4%.

2 Task 2.2

Problem 2.2 [40%] Use the dataset of Problem 2.1 and adopt the same approach (several splits into a training set, validation set and a test set) to determine which combination of SVM model and feature subset has the best AUC among the ones you considered in Problem 2.1. Prepare a Matlab fuction called test.m (function y=test(A)) that implements your best choice combination. The function will accept a matrix A and return a vector y having the same number of rows in A. The matrix A will contain several samples (one for each row) organized in the same way described in Problem 2.1, but without label (i.e. each row will contain containing a sample with 10 numerical features). For each of the samples A(i,:), the function will select the features according to the subset chosen and provide the predicted class in y(i). Your function must be submitted together with your report and will be run on a separate matrix containing new test data. Your grade will be based on the performance of your classifier on the new test data, which will contain a very large number of examples generated from the same distribution.

2.1 Solution

During the testing of the algorithm, we have discovered that the optimal configuration of the classifier lets us obtain an accuracy of 83.4%. This configuration involves setting the following parameters for the SVM :

- boxconstraint : 2.4
- kernelfunction : polynomial
- polyorder : 2
- kktviolationlevel : 0.1

It was decided to use the featureset of 5: [1 2 3 7 9] because it allows us to obtain an AUC close to the one in case of considering the entire dataset, and even though it has a slightly bigger standard deviation, it is still more rational to use because of the speedup that corresponds to the reduction of the number of features.

A function that implements this configuration is implemented in test.m file. It is containing training of the SVM with the selected parameters. It could be possible to save the configuration of the SVM in a matlab file, however in the case of dataset change it may be more difficult to change the settings. That is why the training is also included inside the function.

3 Conclusion

During this work an SVM classifier was implemented. The training parameters have been discovered and optimized. While choosing the best performing configuration we considered the stability of the result (preferably standard deviation of AUC has to be low, so that we can be more sure of the AUC value that we obtain with every run of the classifier), the AUC itself as well as the speed and the number of processed features. It was clear that the case of selecting half of the features was performing rather well, so that it is not necessary in general to use the full feature space just for improving the overall AUC by less than 1%.