ELL-409 (Machine Learning)

Assignment-2

~Harikesh

**PART-2**: In part 2, we are given an extended version of the dataset and for that, we have given 8000 instances i.e. 8000 rows of the dataset and the same 25-dimensional feature space of that dataset. We have to train a multiclass SVM model for predicting labels on a target of 2000 instances with labels hidden.

Training Set: http://web.iitd.ac.in/˜sumeet/A2/train\_set.CSV

By downloading the file from here we can see it has a training set of 8000 samples with 25 features followed by a target label of each sample. Hence the training set is 8000\*26

Target Set: http://web.iitd.ac.in/˜sumeet/A2/test\_set.CSV

It is our target set where we have to predict the labels using our multiclass SVM model. The target set has dimensionality 2000\*25.

We have to use concepts like Hyperparameter tuning, model or feature selection, cross-validation to find the optimal SVM and then optimal accuracy.

First of all, we have to read the training set file and then will try to train our model using that file.

Below is a code snippet to show the reading of the two CSV files from their name.

Text

Description automatically generated

And initially, we are just putting 0.0 in the label of our dataset and will finally try to write it off in the CSV file we want to write. On the other hand for the features of our test set, we are just adding all the features to our array which we have defined and are going to use.

Now, below is a code snippet showing the program to train our SVM using the train set data which we have and then try to classify the labels of the test data set according to the features which are given in our testing dataset. We will try to vary the hyperparameters like the type of kernel, degree of the polynomial which we are going to use if it is a polynomial kernel, etc. We will also change the value of gamma which will also affect the accuracy of our model as it affects all types of kernels except only linear kernel i.e. linear kernel will give the same results irrespective of the value of gamma. We have also done 10 fold cross-validation on our training set data to improve the accuracy of our test set predictions.

Apart from these, we are also checking whether some feature is a good classification of our data or not i.e. we are trying to vary the feature selection by taking only some rows from our training set data in our feature selection and all of that may be helpful in better classification of our testing dataset. Below is the code for the prediction of the features of our testing dataset:

Text

Description automatically generated

Now, we have to submit our testing datset csv file on Kaggle for the ranking. So, finally we have to write all the predicted labels back to a new CSV file and for that we have to submit that file on Kaggle. Below is a snippet of code for writing the class label in our CSV file and we have to write it as the columns with titles ID and Class which are for the features of the testing dataset. For that either we can direct copy paste the values of the class id or we can use code to add it to excel sheet for the Kaggle competition. We will then play with the values of the various parameters and then will try to find the optimal accurate score.

Text

Description automatically generated

After that I have made three tries to submit it onto Kaggle and the best one is giving nearly 96.625% accuracy.

Graphical user interface, text, application

Description automatically generated

**PART 1: (PART 1A)**

In this subproblem, we have to experiment with the use of SVMs for both binary and multiclass classification problems and then will try to make our model accurate by altering various parameters and drawing various plots for the accuracy predicted by our model.

We have been provided we a personalized dataset which contains 3000 rows and there are 25 per data point in a row and the last column contains a class label for that particular data point. We are also told that each data point or each row in the given file represents an image. Hence, each of these 25 features are the festures of the image.

One important thing given is that there are only 10 class labels and hence the last column of the input data contains only integers in range of [0,9] both included. We have to take the last column as input only in form of double and then try to train our data basically.

Input file is given in the link: http: //web.iitd.ac.in/~sumeet/A2/2019CS10355.csv

Now we will be trying to actually form a SVM classifier for these images (given as a dataset in the rows of the input CSV file given to us).Also we have to access the usefulness of different features. For the beginning we are going to use the standard SVM library which is LIBSVM and we will be programming basically in python. We will be basically experimenting on various hyperparaeters such as value of C for soft margin SVM, which kernel function to use, and the kernel parameters such as gamma (if not linear) or the degree of polynomial if it is a polynomial kernel. After that we will start working with our dataset which is given to us and then will try to train it using the standard library which is LIBSVM.

**Binary Classification**: In binary classification we are just going to use two classes lets say 1 and 9 and then are going to find out various relationships. We will be leaving some data aside for validation i.e. we are using cross validation and are taking 80% of the data for training basically. Now we will be forming the graph by varying the parameter gamma and the accuracy we got in all cases. (gamma is just the measure of how far influence of any single training example reaches (inverse of the radius of influence) and we will be trying to find the optimal value of gamma in almost all the kernels. There is another parameter C which trade offs correct classification of training examples against maximization of the decision function’s margin (it is basically how much we want to avoid misclassify each training example).Below are some of the plots:

Binary Classification (Classes 1 and 9) all 25 features included

Plots between gamma and accuracy for different kernels(with all other values same):

For kernel type 0 : As we know that the kernel type 0 i.e. linear kernel doesn’t depend on value of gamma for classification. And hence the training accuracy and cross-validation accuracy will remain same even by changing the value of gamma. Plot is shown below:

Classes used 1 and 9, kernel type 0 (linear), All 25 features used

We can clearly see that here the accuracy of both cross validation remain same. And varying gamma doesn’t change accuracy.

Classes used 1 and 9, kernel type 1 (quadratic polynomial), All 25 features used

Classes used 1 and 9, kernel type 1 (tertiary polynomial), All 25 features used

Classes used 1 and 9, kernel type 2, All 25 features used

Classes used 1 and 9, kernel type 3, All 25 features used

Now we will try to change the classes and then will try to see if the same parameters give the max accuracy or does it highly depends on the classes chosen.

Classes used 4 and 7, kernel type 0, All 25 features used

Classes used 4 and 7, kernel type 1 (quadratic polynomial), All 25 features used

Classes used 4 and 7, kernel type 1 (three degree polynomial), All 25 features used

Classes used 4 and 7, kernel type 2, All 25 features used

Classes used 4 and 7, kernel type 3, All 25 features used

Now we will try to change the classes and then will try to see if the same parameters give the max accuracy or does it highly depends on the classes chosen.

Classes used 0 and 5, kernel type 0, All 25 features used

Classes used 0 and 5, kernel type 1(quadratic polynomial), All 25 features used

Classes used 0 and 5, kernel type 1(3 degree polynomial), All 25 features used

Classes used 0 and 5, kernel type 2, All 25 features used

Classes used 0 and 5, kernel type 3, All 25 features used

Note: To distinguish the case of underfitting and overfitting the parameter gamma play an important role. The parameter gamma plays an important role in overfitting. The higher the value of gamma, the more is the tendency of hyperplane to match the training data. And if we took gamma to be ver very less than that will be case of underfitting. So, our role is to choose gamma to avoid both overfitting and underfitting. Hence in any graph we can see that initially the accuracy is less and than it increases to a certain extent and then again decreases. So, left side is the case of underfitting in any graph and right side is the case of overfitting. One another aspect from which we can see overfitting is that higher value of our misclassification error parameter i.e. C lead to overfitting of results and hance will have a low bias and high variance. As the value of C approaches towards infinity there is no room for error as penalty for misclassification is enormous and it results in heavy overfitting as the model pushes for the best possible accuracy.

Now choosing the type of kernel function affects accuracy as was seen earlier too. This plot shows the accuracy classification for the best value of gamma as was shown earlier by plots:

Classes used : 1 and 9, gamma = 0.02.

Here 0,1,2,3,4 respective kernels which are linear, quadratic, three degree polynomial, Radial bias and sigmoid kernels. In binary classification the classifier accuracy is coming nearly 100% almost always because there are only 2 classes and it can be like both the classes are pole apart in the two cases and hence it will be normally classifying every datapoint correctly.

Now taking two more pairs of classes we can see that for varying various hyperparameters we can clearly see that the accuracy in all the three pairs of classes is not coming at the same hyperparameters. Lets took the case of classes 4 and 7 we can see that mostly we are getting a good accuracy near the region of gamma = 0.02 while on the other hand in classes like 0 and 5 the accuracy is coming very good at points like 0.06. Talking about for different kernels here is the graph showing the accuracy for different kernels for same gamma(0.02) for different classes :

Classes 1 and 9:

Classes 4 and 7:

Classes 0 and 5:

Now we can see that for same value of gamma, for different classes we have different accuracies and if we compare we are getting the max accuracy at the 3rd kernel that is fine but for third kernel if we see gammas at which those differences occur we can easily conclude that there are different gammas at which the accuracy is maximum and hence we are getting the best results for different hyperparameter settings and they are varying a lot as pair of classes are varying. This is because different pair of classes use different features heavily i.e. have high weight of different features and hence during classification it may occur that we need different gammas for different classifications and hence there can be different kernels at which best results occur as if those two characterstics which are highly bounded are segretable by linear kernel easily then it is fine. Hence, for changing the classes one can see that the hyperparameters with best results are varying a lot.

Now we will be trying to use the first 10 features instead of all the features for the same pair of classes which are 1 and 9 and we will try to conclude some results from it. We will be finding the same plots as founded above and will try to use something to find the accuracies.

Classes 1 and 9, 10 features used, kernel type 0

Classes 1 and 9, 10 features used, kernel type 1(quadratic)

Classes 1 and 9, 10 features used, kernel type 2

Now from the above graphs and the graphs obtained earlier using all the 25 features one can easily conclude that the accuracy in case of using 10 features is less than using all the features as it may be possible that the image is uniquely identied/classified using the features other than the one taken among those 10 and by taking only 10 features we are giving a path to take many misclassifications in our training data and it then be used against us on our test data by giving less accuracy. More and More constraints we will be having. More and more it will be good for us to increase accuracy.

**Multiclass classification**: Now we will be training a classifier for all the 10 classes and (only by using standard SVM library i.e. LIBSVM). LIBSVM uses one vs one method for classification. We will be evaluating it for cross validation. Now we will be seeing the effect of changing various hyperparameters and kernel functions. After that we will be trying to finetune them and obtain best possible accuracy. Now we will be comparing these tuned values to that of binary classification.

Multiclass, kernel type 0 :

Multiclass, quadratic kernel:

Multiclass, three degree kernel:

Multiclass, kernel type =2

Multiclass kernel type = 3

Now as above we have plotted graphs between accuracies and gamma for various kernel type and we can see that we are getting max accuracy i.e. best possible results in case when the value of gamma is nearly 0.06 which was not the case when we were doing the binary class classification. These are different then those obtained in binary case and that is kind of obvious because here we are doing for all classes and there may be the case that those 2 classes are separatable by a kernel which was quite fine there but not here as LIBSVM uses one vs one approach for these cases. Hence, there are major differences as all the classes can have a different classifier hyperplane as compared to just two classes out there. Below is a plot showing the accuracies vs the type of kernel used and this is also different because there can be basic kernel which may be useful there but it is not the case here

Multiclass, gamma value = 0.06,

Now we will be using first 10 features only instead of all 25 features and see the results. Also, we will be deducting what does the results conclude about the usefulness of different features basically.

Multiclass, kernel=0 (type linear), features used = 10

Multiclass, kernel =1 (quadratic), features used = 10

Multiclass, kernel = 2, featured used = 10

Multiclass, kernel = 3, features used = 10

Now from the above graphs we can clearly see that we are not getting good accuracy whatever we try and this is the case because while using 25 features we were guaranteeing the group of images using some other classes also which was the scope as even if few of the features are misclassifying we were anyways able to classify it very finely using the other features and at last see the combination of other images. Now, if we are decreasing the number of features we can clearly see that we will not be classifying everything perfectly as there will be a lot of misclassifications while doing cross validation in our data. Hence, all the features are required for making such groups of images as there can be some features which can easily divide the group of images explicitly. Hence the usefulness of each and every feature is necessary for any image classification.

**PART 1B :**

Here in this part of the assignment we have to use the same datset which we were using while calculating the binary classification and here we will be using the simplified SMO algorithms. And then we will be comparing our results which we get using the previous two algorithms i.e. using our standard library and CVX.

First lets compare the runtime of two algorithms: For that we will be using time library of python and will be calculating the time taken by two algorithms to calculate time.

The below code snippets show the runtime of two of the algorithms and we can clearly see that Simplified SMO is taking more time to execute the program and it is the case because SMO’s take the sequential minimum algorithm to just reach to most accurate classification. Also time taken is relative and depends on various parameters like the number of iterations in case of SMO’s while in case of LIBSVM’s it depends on the type of kernel and misclassification penalty. But mostly by running we got the idea that SMO is taking more time than SVM. One theoretical thing we know is that for real world sparse data set, SMO id too much faster than LIBSVM.

Runtime of LIBSVM algorithm:

Text

Description automatically generated

Runtime of Simplified SMO algorithm:

Text

Description automatically generated with medium confidence

Accuracy of Simplified SMO’s:

Now to check the accuracy of simplified SMO’s we have to plot some graphs and check how it’s accuracy is depending on various parameters and then we will see it’s maximum accuracy.

Now, we are taking the two same classes as we were taking during our binary classification and are plotting some of the graphs:

Classes 0 and 5, number of iterations 2000, regularization parameter = 4

The below is the plot between Numerical tolerance and accuracy of Simplified SMO algorithm:

Classes 0 and 5, regularization parameter = 4, numerical tolerance = 0.001

The below plot is the plot between Accuracy of our model and the number of iterations required:

It is also noted that accuracy in case of SMO’s doesn’t depend on Regularization parameter that much and hence we are always getting almost similar accuracy in case regularization parameter is increased slightly.Now, finally as for the two classes we have seen both of the above methods i.e. Simplified SMO and LIBSVM we can now clearly deduce that although both of them have high accuracy but in case of SMO it is even higher than the other one and this is the case because in SMO’s even by varying the parameters we are getting pretty good accuracy but in the case of LIBSVM’s we were getting low accuracy on almost similar parameters. As of the question of maximum accuracy it is the case that maximum accuracy of both are almost same but it is slightly better in case of SMO’s. So, on an average SMO’s are having better accuracy than LIBSVM’s.

**PART 1C:**

We have to now build full SMO on our simplified SMO by adding some more things. After that we have to compare our results to all of the preceding algorithms/proesses we have used. Apart from these we have to also tell which of the langrange multipliers we have to optimize.

Now, regarding runtime of the algorithm we know that SMO is practically the fastest algorithm to train support vector machines because for training a SVM we have to solve a Quadratic Programming problem and SMO break this into a set of smaller such problems. The smallest possible optimization involves two lagrange multipliers because lagrange multipliers must obey a linear equality constraint. Hence, at every point SMO chooses two lagrange multipliers, find the optimal values for these multipliers and updates SVM’s to reflect the new optimal values. Now, SMO uses a heuristic for choosing which multipliers to optimize. Choice of first heuristic provides the first outer loop of SMO algorithm. And in each iterations it is checked which multipliers are violating KKT conditions and those are actually eligible for optimization. Once the first multiplier is chosen now SMO chooses the second lagrange multiplier to maximize the size of step taken during joint optimization. For comparing the result with previous processes we can clearly see that this is faster algorithm comparing to the other two but the main issue here is that the accuracy can not be that much better although it can become better but for that the algo will take a long period of time.

The algorithm sketch is shown below:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Convex Optimization Package:**

While doing this we first have to tune all hyperparameters using LIBSVM i.e. we don’t have to tune hyperparameters using this package and then we have to train tuned values using CVX. Then we have to compare the results obtained through other methods to this method.

In order to use CVX to train an algorithm we have to first know how to express SVM dual problem in a form which can be fed into package. And when the CVX runs an gives the output we have to figure out how to take the output from the CVX andconstruct the actual classifier to run on test data. So, the basic procedure of taking the input and output we have to show. We can try it using skylearn, pandas and numpy library maybe but it is not possible here.

Text

Description automatically generated

References for the assignment: [smo.pdf (iitd.ac.in)](https://web.iitd.ac.in/~sumeet/smo.pdf)

[Microsoft Word - smoTR.doc (iitd.ac.in)](https://web.iitd.ac.in/~sumeet/tr-98-14.pdf)

<https://www.bytefish.de/blog/using_libsvm.html#:~:text=In%20this%20experiment%20I%20want%20to%20show%20you,in%20reader%3A%20classes.append%28int%28row%29%29%20data.append%28%5Bfloat%28num%29%20for%20num%20in%20row%5B1%3A%5D%5D%29>

<https://www.codeproject.com/Articles/1267445/An-Introduction-to-Support-Vector-Machine-SVM-and>

<https://github.com/LasseRegin/SVM-w-SMO/blob/master/SVM.py>