Question 0: SVM questions from the lecture

Download the MNIST dataset from the link: http://yann.lecun.com/exdb/mnist/

For a 2-class classification between 0 and 1, split the training set into 80:20 ratio for training and validation in a stratified manner. Using the training set, perform 5-fold stratified cross-validation such that:

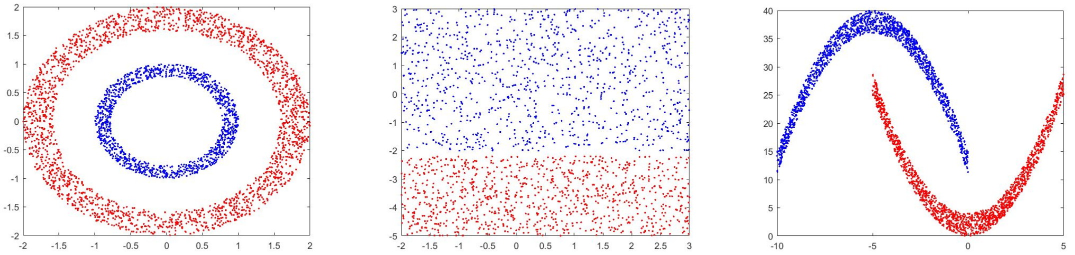
(a) All training samples (80% data) corresponding to 0 and 1 are used for training SVM.

(b) Use the above-trained model to obtain support vectors. Use these support vectors (corresponding to 80% train data) to train another SVM classifier from scratch.

Report mean $\pm$ std on the testing and validation set for both the cases and draw ROC curve for the 2 class classification. Draw your inferences from the observed results on why one approach performs same/worse/better than the other.

Question 1: Basics of SVM

(a) Use the codes provided to generate the data distributions shown in the images below. If you are using any other language other than MATLAB, generate the distribution using the details provided in the code. For each of the distributions shown below, perform the following experiments (b-g). You may reduce the sample points for all the following parts if the experiment is taking time. Use the script twice (with 2 different seeds for generating random points) to generate training and testing samples.



(b) Using a linear kernel and hinge loss, perform 2 class classification for each of the distributions to obtain baseline accuracy for SVM with default parameters.

(c) Using a grid search mechanism, find the optimal accuracy by the varying value of regularization parameter (C) and the gamma parameter simultaneously. Vary value of C as follows:

10^i, $where\ i = [-2, 5]$

And the value of gamma varies as follows:

10^i, $where\ i = [-5, 2]$

Grid search in part (c) should only be done by changing C and gamma parameters over what you had done in part (b). The assignment does not require you to perform a grid search experiment for upcoming part (d) and part (e).

(d) Change the type of loss used in part(b) to squared hinge loss and report the performance.

(e) Report classification accuracy by changing kernels to polynomial, RBF, and Sigmoid.

(f) Analysis of results:
   (i) Why is a linear kernel able to classify/misclassify on the 3 distributions in part (b).
   (ii) Why and how does the accuracy change on varying C and gamma parameter in part(c).
   (iii) Why and how does the accuracy change on changing the loss in part(d).
   (iv) Why and how does the accuracy change on changing the kernels in part(e). How does TP, TN, FP, and FN vary on changing the kernels?

(g) Visualize the decision boundary obtained for:
   (i) Part (b)
   (ii) The best results obtained from the grid search in (c)
   (iii) For the new loss in part (d)
   (iv) Each of the kernel in part (e)

For each of the part of question 1, report the classification accuracy along with the confusion matrix, reporting TP, TN, FP, FN. Draw a combined ROC for parts: 1(b), the best performance of 1(c), each of the kernels of 1(e).

Question 2: Multi-class SVM

CIFAR 10 is an image database for classification of images into 10 classes. The database can be downloaded from: https://www.cs.toronto.edu/~kriz/cifar.html . Use any one training set out of 5 to train the SVM and use the testing set to evaluate the model.

Use Multi-class SVM by:

(a) One-vs-one
(b) One-vs-rest

mechanism to classify images into 10 classes. Explain the difference in the results for 10-class classification, explaining why one approach outperforms the other. For each of the part of question 2, report the classification accuracy along with the confusion matrix.

Question 3: Incremental SVM

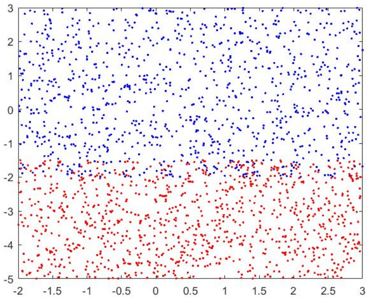
CIFAR 10 is an image database for classification of images into 10 classes. The database can be downloaded from: https://www.cs.toronto.edu/~kriz/cifar.html . However, instead of 10 class classification, we would perform 2 class classification in this question. For training, consider only data_batch_1, data_batch_2, and data_batch_3. Combine the three training sets. For this combined training set and the only testing set, prune the dataset to get samples corresponding to only cats and dogs. Perform incremental learning of SVM by presenting 100 training samples at a time, i.e, update trained SVM incrementally by providing it 100 samples each time. After training of each 100 samples of the training set, report:

(a) Classification accuracy on the testing set after each incremental learning. How does the testing accuracy vary before and after the incremental learning?
(b) Classification accuracy on the total observed training set (i.e, 100 samples after first training, 200 samples after second training, 300 samples after the third training an so on).
(c) Classification accuracy of the most recently trained training set (i.e, 1-100 samples after the first training, 101-200 samples after second training, and so on).
(d) How does the number of support vectors vary after each incremental learning?

Draw a plot of the above accuracies (part a,b, c) after each incremental learning. Clearly mention the algorithm you followed for incremental SVM. Report your understanding of the experimental observations.

Question 4: Advanced SVM

Similar to Q1, use the code provided to generate the following distribution. Use the script twice (with 2 different seeds for generating random points) to generate training and testing samples. You may reduce the number of training samples, however, make sure there is some overlap of samples belonging to different classes in the feature space and the generated samples are same for both the parts.



Using the generated training data, train a:

(a) nu-SVM

(b) C-SVM

to learn a classification model. Test the model using the second set of generated points.

Compare and contrast the 2 algorithms on basis of:

1. Time taken to train the model
2. Model performance
3. The number of FP and FN in the testing and training set.
4. Comparison with simple SVM with a linear kernel
5. The number of support vectors of nu-SVM, C-SVM and simple SVM with a linear kernel

Submission Policy and Requirements

1. Any kind of plagiarism is not accepted. We will strictly follow institute policies for plagiarism.
2. Recommended programming languages: MATLAB, python.
3. You may use any external libraries or GitHub codes. However, the evaluation will test your knowledge of the algorithm and the choice of hyperparameters. Do cite the libraries/codes.
4. **Submission should include:** Working code for each of the part separately and a report to show the analysis of results in each of the parts.

Assessment criterion

The assessment will be done on basis of the following components:

1. Working codes
2. Analysis and clarity of results (drawing comparisons across different parts) & clarity of the report
3. Understanding the theoretical concepts and the choice of hyperparameters.