Machine Learning
Homework 7 : SVM
(due Midnight April 9)

Instructions

1. The deadline for full score is Midnight April 9. You can get 50% credit for late submission (Midnight April 10).
2. Total marks = 8
3. You have to type the assignment using a word processing engine, create a pdf and upload on the form. Please note that only pdf files will be accepted.
4. All code/Jupyter notebooks must be put up as secret gists and linked in the created pdf submission. Again, only secret gists. Not public ones.
5. Any instances of cheating/plagiarism will not be tolerated at all.
6. Cite all the pertinent references in IEEE format.
7. The least count of grading would be 0.5 marks.

1. (a) Implement a function for hard margin SVM in primal form using cvxpy. For keeping this task simple assume w is two dimensional, i.e. $\hat{y(x)} = SIGN(w_1x_1 + w_2x_2 + b)$ where x and w are both two dimensional vectors **[2 marks]**

   (b) Show the usage of your implementation on the IRIS dataset. For all our experiments, we will only be making use of sepal-length and petal-width as the two features. We have only two classes - Setosa and Not-Setosa. This problem is linearly separable.

   i. For showing your implementation, train the hard margin SVM using all the data. **[1 marks]**

   ii. cvxpy also allows you to see the dual values. Can you compute the dual values? What do they mean? **[1 marks]**

   iii. Plot the decision boundary in dark black and the margins in dotted line. Encircle the support vector points. **[1 marks]**

   (c) Implement hard margin SVM in primal form using cvxpy.
   First, randomly shuffle the dataset. Then, Use the first 70% of the dataset for training and report the test performance on the remaining 30%.**[2 marks]**.

   (d) We will now use active learning to improve the performance of training over the IRIS dataset. We will use the first 10% of the samples of the shuffled IRIS dataset for training set, and the last 30% as the test set. The remaining 60% of the samples will serve as the pool set which we can query to obtain new labeled instances.

   i. We will retrain our above Gaussian Naive bayes model on the new train set (consisting of 10% of the samples). As before, we will only be making use of sepal-length and petal-width as the two features. Next, we will do 10 iterations of active learning where we will query on point from the pool set, acquire its labels, add that instance to the train set and remove from the pool set. For each time a new instance is added to the train set, we retrain our model and compute the test accuracy. The querying strategy is to choose the instance with least confidence (Naive Bayes is well suited for this application since it directly gives us the probabilities of different classes given the observed features) **[3 marks]**

   ii. Use the notebook shown in the lectures as a base to show how active learning changes the decision surface (different colours are assigned to the 3 different classes over the 2d space of sepal-length and petal-width) as a function of number of queries. This can be demonstrated via an animation exported as a GIF. **[1 mark]**

iii. Compare the accuracy of the above active learning strategy with random sampling where we query a sample at random from the pool set. Repeat this experiment over 5 different random seeds and make a plot with x-axis showing the number of queries, y-axis shows the accuracy on the test set. There are two lines to show - Naive Bayes with least confidence estimation, and a line showing mean and standard deviation for the random estimate. What can you infer from this plot? **[2 marks]**