EECS545 - Homework 4

February 26th, 2018

Instructions

- This homework is Due Monday, March the 12th at 5pm. No late submissions will be accepted.
- As always, submit a write-up and your code for this homework.
- You are expected to use Python for the programming questions in this homework. Use of Python 3 is recommended.
- **Submit your write-up to Gradescope under the assignment titled** `Homework 4`. Make sure to tag the pages with the corresponding problems numbers. Please check our recent announcement on Canvas for submission instructions.
- Submit your plots and answers as a pdf on Canvas under filename `uniqname_hw4.pdf`.
- Submit all your python code to Canvas in a compressed zip file named `uniqname_hw4_code.zip`. Your zip file should contain `prob1.py, prob2.py`.

1. In this problem you'll use `sklearn.svm.SVC` to learn a SVM classifier with different kernels and report plots of your results. You may find the following links are useful:

   - `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
   - `http://scikit-learn.org/stable/auto_examples/svm/plot_iris.html`
   - `http://scikit-learn.org/0.17/auto_examples/svm/plot_separating_hyperplane.html`

   The dataset is generated in starter code `prob1.py`. The dataset consists of 30 samples, each of them has two features and a label of 1 or -1. The input features are stored in matrix 'X', and labels in vector 't'. Please do not change the generation code, and in particular do not normalize the dataset.

   (a) Use the identity kernel, for each of the following values of C in $\{1.0, 100.0\}$, report the number of support vectors and construct and submit a plot showing:

   - A scatter plot of given inputs colored based on the corresponding labels. Please use x_i's as X-coordinate and y_i's as Y-coordinate.
   - The separating hyperplane learned by your algorithm.

   (b) Use the Gaussian kernel, for each of the following values of C in $\{1.0, 3.0\}$, report the number of support vectors and construct and submit a plot showing:

   - A scatter plot of given elements colored based on the corresponding labels. Please use x_i's as X-coordinates and y_i's as Y-coordinates.

- The decision boundary learned by your algorithm.

Hint: For the SVC learner, you need to set the parameter "C" to the penalty coefficient you choose, and make sure to set the "kernel" to "linear" when you want to use identity kernel, or to "rbf" for Gaussian kernel with default kernel width. For plotting, you will find the functions `contour` and `scatter` helpful. We suggest you to set the data limits for x-axis to [-4,6], and the limits for y-axis to [-6,3].

2. We will train an SVM classifier to recognize images of hand-written digits in this problem. You will train models on the widely used MNIST dataset. The provided starter code downloads and loads the MNIST dataset and constructs subsets of 10k images each for training and testing. Normalized pixel values of input images are considered to be the input features. The classifier predicts an output label $\in \{0, 1, ..., 9\}$ corresponding to the digit identity.

   (a) What is the accuracy of a classifier that guesses the class label uniformly at random ? 10%

   (b) Use the SVM package from scikit-learn to train a classifier. Use the hyperparameter values $C = 1, \gamma = 1$ with an RBF kernel (Eg: `classifier = svm.SVC(C=1,gamma=1)`). Report the accuracy of the trained model on the test set.

   (c) The RBF kernel for a given γ parameter is given by $k(x, x') = \exp(-\gamma||x - x'||^2)$. The gamma parameter controls the extent of influence of a particular training example. Where does a classifier with high/low γ respectively, fit in the bias-variance spectrum ? (Eg: A classifier with high γ has ____ bias and ____ variance.). Justify your answer **briefly**.

   (d) Perform 5-fold cross-validation to choose better parameter values for C, γ from the following range of values $C \in \{1, 3, 5\}$, $\gamma \in \{0.05, 0.1, 0.5, 1.0\}$. Report the best values of C, γ identified and report the corresponding test accuracy.

   (e) (**Ungraded**) The best SVM models achieve an error rate of under 1.5% on MNIST [1]), on the standard train/test split. Explore techniques to bring down the error rate you found in part (d) above. Feel free to use a larger training set for this purpose.

[1] `http://yann.lecun.com/exdb/mnist/`