Course › Week 7... › Progra... › Progra...

Programming Assignment 7

Click this link to download the Kernel SVM and Perceptron notebook and then complete problems 1-5.

As part of this notebook, you are asked to: implement the kernel Perceptron algorithm in such a way that it allows either the quadratic or RBF kernel; train a classifier using this algorithm on 2-d data; and then display the resulting nonlinear decision boundaries. It will be helpful to think of this in two parts:

1. Implementing kernel Perceptron.

The algorithm is short and fully spelled out in lecture. The two kernel functions you will be using are the quadratic kernel

$$k\left(x, z\right) = \left(1 + x \cdot z\right)^2$$

and the RBF kernel

$$k\left(x, z\right) = \exp\left(-\|x - z\|^2 / s^2\right)$$

2. Running the learning algorithm and displaying the results.

Part of the notebook shows how to do this for kernel SVM, and you can pretty much use this same subroutine. If you are having trouble understanding the code, the idea is this: it first chooses a particular region of the 2-d plane to display, based on the bounding box of the data set; then it puts a very fine grid on this rectangular region (sort of pixelizing it); it evaluates the classifier at every grid point (pixel); and then it shows the decision regions by coloring each pixel in the rectangle according to the prediction at that location.

Problem 1

1/1 point (graded)

Were you able to correctly implement the kernel Perceptron algorithm? You will be given full points regardless of your answer.

- ● Yes, and it was fairly easy.
- ○ Yes, after some trouble.
- ○ Almost: I was able to implement one of the two kernel functions, but not both.
- ○ I had a lot of trouble with this, and did not eventually get a working function.

✓

Submit

Problem 2

1/1 point (graded)

On which of the five data sets, if any, did your quadratic-kernel Perceptron algorithm fail to converge?

- □ `data1.txt`
- □ `data2.txt`
- ☑ `data3.txt`
- □ `data4.txt`
- □ `data5.txt`
- □ None of them

✓

Submit

Problem 3

1/1 point (graded)

On which of the five data sets, if any, did your RBF-kernel Perceptron algorithm, with scaling parameter $s = 1.0$, fail to converge?

□ `data1.txt`

□ `data2.txt`

□ `data3.txt`

□ `data4.txt`

□ `data5.txt`

☑ None of them

✓

Submit

In week 3's assignment *Gaussian generative models for handwritten digit recognition*, you saw how to load the MNIST data set of handwritten digits and build a basic classifier for it. It is now time to try using an SVM for this problem.

Create a new Jupyter notebook to do the following.

Load in the MNIST data (the training set of $60{,}000$ points and the test set of $10{,}000$ points).

Learn a linear SVM classifier using $\mathtt{sklearn.svm.LinearSVC}$, setting $\mathtt{loss} =' \mathtt{hinge}'$. Try different values $C = 0.01, 0.1, 1.0, 10.0, 100.0$. Find the training error and test error in each case.

Then try kernel SVM with a quadratic kernel. For this you can use $\mathtt{sklearn.svm.SVC}$, setting $\mathtt{kernel} =' \mathtt{poly}'$ and $\mathtt{degree} = 2$. Just try the setting $C = 1.0$.

Then answer the questions below.

Problem 4

1/1 point (graded)

Is the MNIST training set linearly separable?

○ Yes, a training error of zero is obtainable.

○ No, the best possible training error seems to be around 5%

● No, the best possible training error seems to be around 10%

○ No, the best possible training error seems to be around 15%

○ No, the best possible training error seems to be around 20%

✓

Submit

Problem 5

1/1 point (graded)

What is the effect of using the quadratic kernel? Selet all that apply.

☐ The training error drops to between 5% and 10%.

☑ The training error becomes zero, or close to zero.

☐ The test error drops to about 5%.

☑ The test error drops below 2%.

✓

Submit