Machine Learning HW 6

Kevin Zen

May 31, 2016

Collaborated with Malika Aubakirova, and Geoffrey Wang

Problem 1

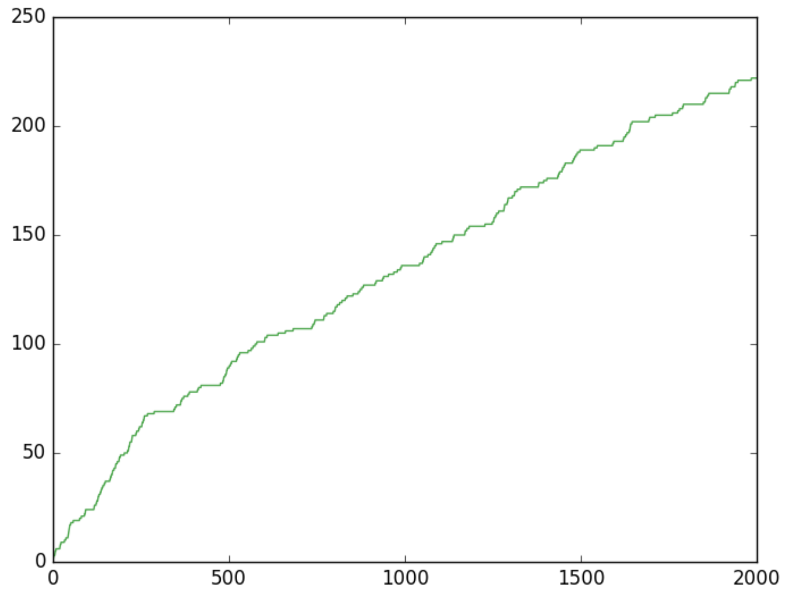
Uploading files to SVN:
To loade .npy files:

```
> import numpy as np
> labels = np.load('filename.npy')
```

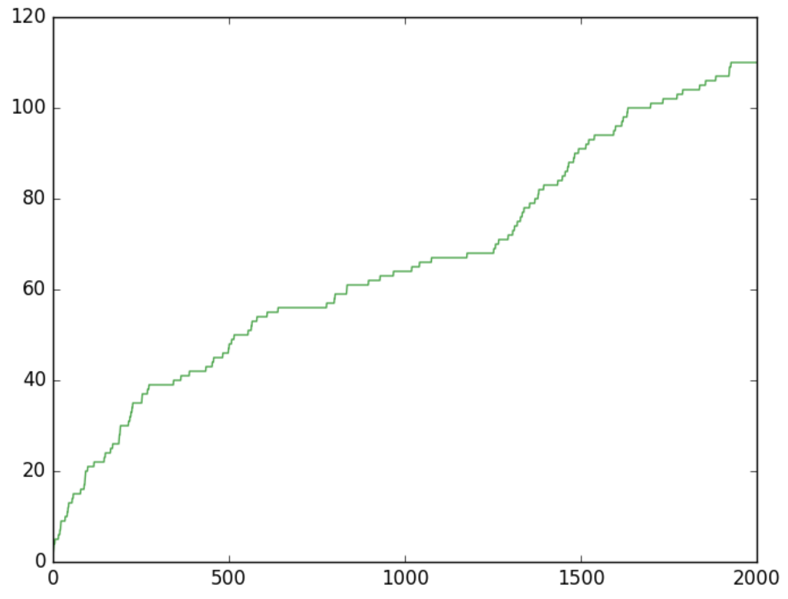
a.
lin_perceptron.py
b.
kernal_perceptron.py
c.
output/online_linear.npy
output/batch_linear.npy
d.
output/online_kernal.npy
output/batch_kernal.npy

Write up

a.
Linear Online Example Error Plot

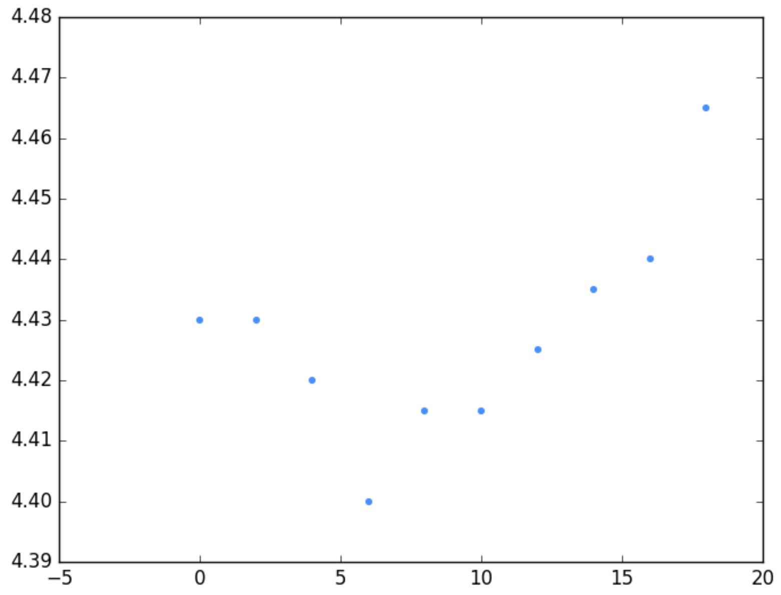


Kernal Online Example Error Plot

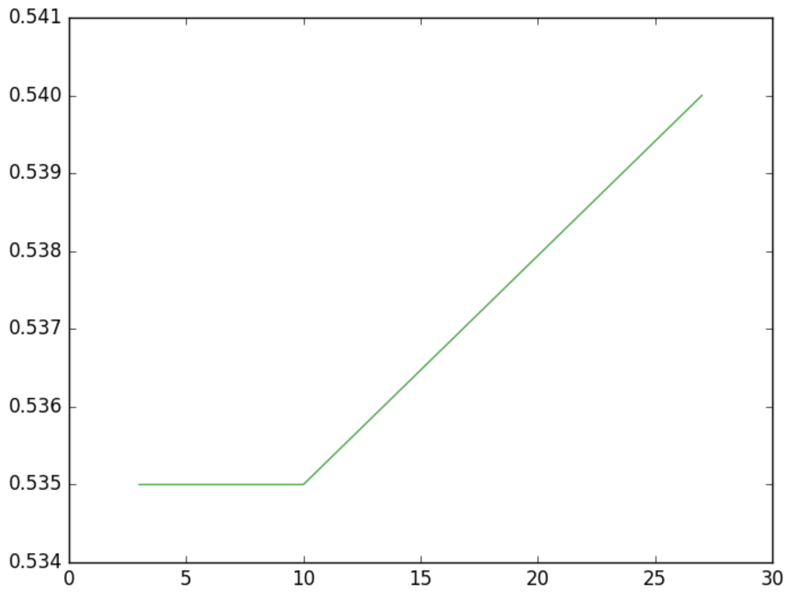


b.

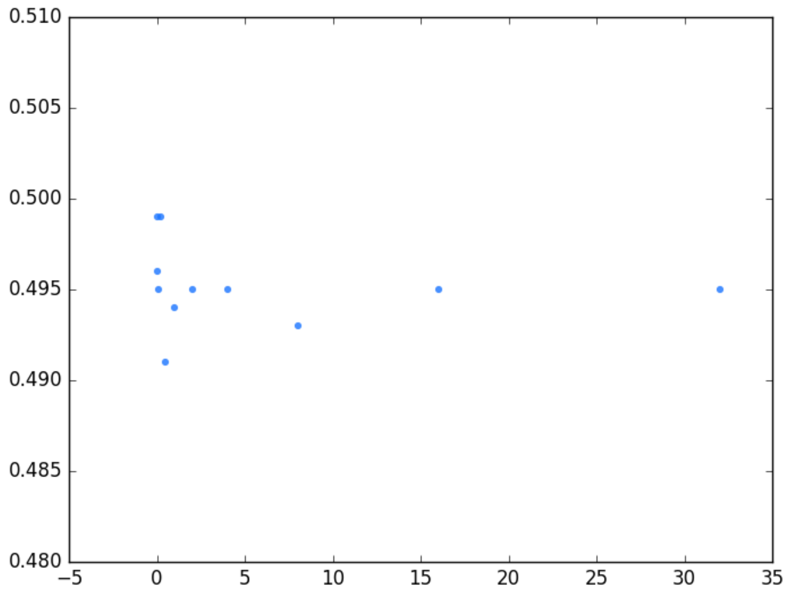
Linear Cross Validation(Thresh):



RBF Kernal Cross Validation (Thresh):



RBF Kernal Cross Validation:



c. We ran grid search along with k-fold cross validation to determine the optimal error rate for our linear batched perceptron. This is shown with under the 'Linear Cross Validation' graph, and we see that with $k = 10$, such that we use 200 of our training set as our validation set, that a threshold of 6 or 0.03 gives us the lowest error rate in our batched solution. Running k-fold, we split our 2k training set into 10 different training (size 1800) and validation sets (200) in order to get our results.

We implemented the same k-fold cross validation to determine the optimal gamma value to calculate the rbf kernal $exp(-\gamma * |x_i - x_t|^2)$ We used a range of possible gammas ranging from $2^(- 6), .., 2^(5)$ in exponential intervals. We found that our gamma that gave us the best value is $2^(- 6)$

We also ran grid search to find the best threshold for the rbf kernal perceptron in the same manner we did for the linear cross validation. This is depicted in the graph above under RBF Kernal Cross Validation (Threshold). We see that selecting any range from 0 to 10 has the same low error rate, and translated into percentages, those are values from 0 to 0.02. Essentially for both kernal and linear batched perceptrons, the number of cycles we run are dependent on the error, being below 0.03.

d.

After running our perceptrons on our the test data, and manually checking if the test data was correct or not, we attained the following values. Note these may be off, as some of the test data were difficult to categorize from a human point of view.

Linear Perceptron:

Online = 0.875 Batched = 0.91

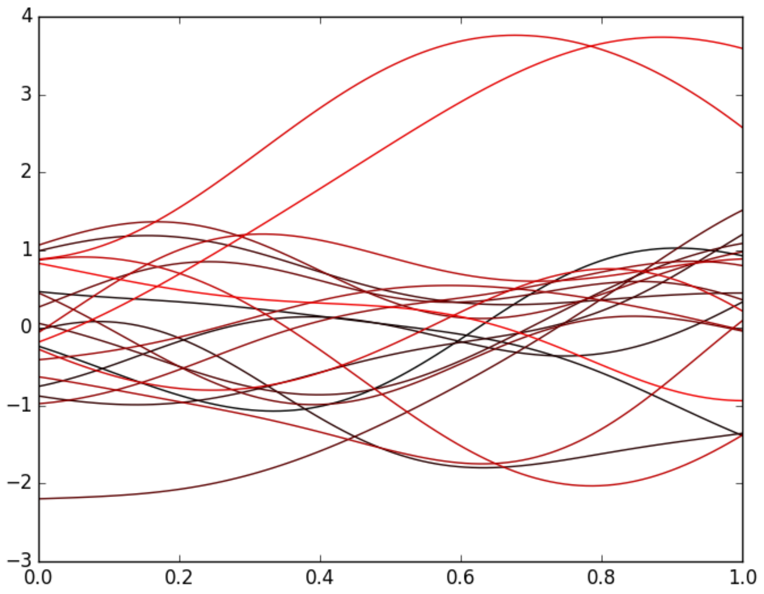
RBF Kernal Perceptron:

Online = 0.92 Batched = 0.92

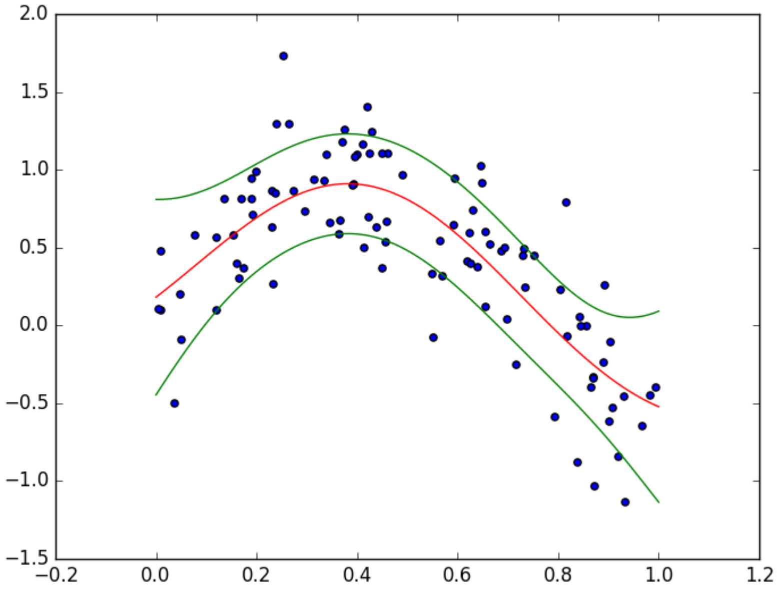
We see that chosing the correct gamma, our optimal RBF kernal perceptron does better than the linear perceptron. However the online and batched rbf kernal perceptron don't have different value, at least for this test set. The worst results were the online linear perceptron.

Problem 2

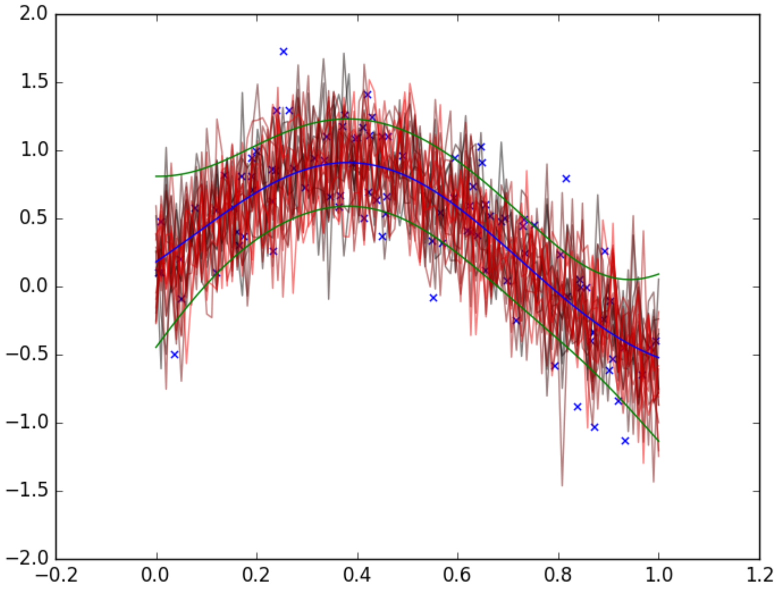
2a



2b


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

2c


2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2