HW 4

Ques 2a

1 vs 1 (I used the gausian kernel)

using lambda of 0.2 I got an accuracy of about 90% and the confusion matrix as shown below. The program runs in about 6-7 minutes. It is named main\_1vs1\_rbf.m. It achieves a good amount of accuracy without taking an excessive amount of time like the one Vs all case does. For this case y program compares each testing sample in all possible binary classification cases and makes a voting matrix in which each row of the matrix shows the voting values of each testing sample. And the column of this matrix exemplifies the classification label. In the following confusion matrix and for any further use. Label 10 is used for labeling 0 while for other numbers the label is the number itself. It can be seen in the matrix below that the algorithm makes the highest mistake in figuring out the difference between 5 and 3.

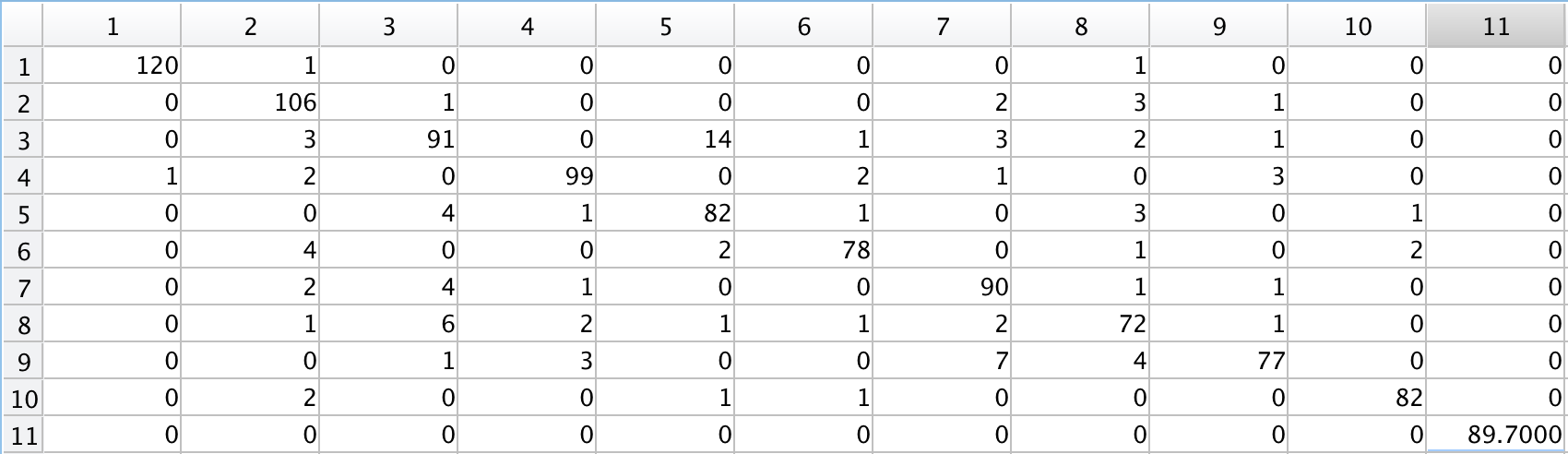


fig 1. The columns are the predictions and the rows are the actual values.

1 vs Rest (I used the gausian kernel)

using lambda of 0.2 I got an accuracy of about 88% and the confusion matrix as shown below. The program runs in about 50 minutes. It is named main\_1vsAll\_rbf.m. This is not the recommended voting scheme for the time it takes is too long and the results are not an improvement on that of 1 vs 1 scheme. For this method I recoded the maximum values of the y calculated for each of the 10 cases for all testing samples and whichever one was the highest was selected as the label. The following confusion matrix is received.

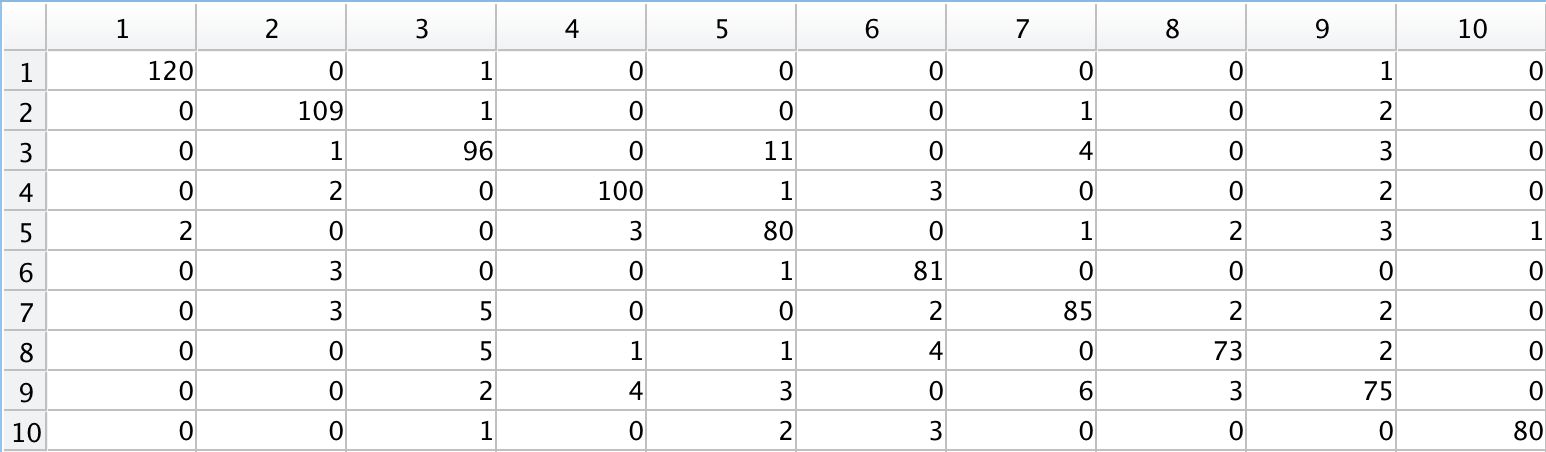


fig 1. The columns are the predictions and the rows are the actual values.

Ques 2b

For DAGSVM. I used my same code from 1 vs 1 as mentioned above for training. For testing used a 3 dimensional matrix where the z coordinate was related to the first label, y coordinate(columns) to the second label and rows to each testing sample. This comprised of my voting matrix for each possible case. Then using the value of |w| I found the labels with the highest separation between them. Using the value of the labels and the votes from the voting matrix I removed the discarded label for each sample. Finally leaving only 2 sample one of them being the predicted label. My code similar to my 1 vs 1 code has an accuracy of 90% and takes 15 minutes to run. Unfortunately I didn’t have enough time to improve my code to get 95% accuracy or more. Although a very clever idea as opposed 1 vs 1 algorithm it doesn’t improve a lot on the 1vs1 algorithm and while taking the same amount of time (6 minutes). As cane be seen in the confusion matrix below. It is very similar to that of simple 1 vs 1 kernel SVM. 