Raymond Duncan
CSCI 5266
SVM Writeup

After running a grid search on several parameters for an SVM model, I get the following output, which suggests that for this task (differentiating between 3 and 8 in the MNIST dataset), a polynomial kernel with margin 100 and degree 2 performs the bests.

Rank: Parameters
1: {'kernel': 'poly', 'C': 100, 'degree': 2}
2: {'kernel': 'poly', 'C': 50, 'degree': 2}
3: {'kernel': 'poly', 'C': 20, 'degree': 2}
4: {'kernel': 'rbf', 'C': 50, 'gamma': 0.0001}
5: {'kernel': 'poly', 'C': 10, 'degree': 2}
6: {'kernel': 'rbf', 'C': 20, 'gamma': 0.0001}
7: {'kernel': 'rbf', 'C': 100, 'gamma': 0.0001}
8: {'kernel': 'rbf', 'C': 10, 'gamma': 0.0001}
9: {'kernel': 'poly', 'C': 100, 'degree': 3}
10: {'kernel': 'rbf', 'C': 5, 'gamma': 0.0001}
11: {'kernel': 'poly', 'C': 5, 'degree': 2}
12: {'kernel': 'linear', 'C': 1, 'degree': 1}
13: {'kernel': 'linear', 'C': 50, 'degree': 1}
14: {'kernel': 'linear', 'C': 20, 'degree': 1}
15: {'kernel': 'linear', 'C': 10, 'degree': 1}
16: {'kernel': 'linear', 'C': 5, 'degree': 1}
17: {'kernel': 'linear', 'C': 100, 'degree': 1}
18: {'kernel': 'poly', 'C': 50, 'degree': 3}
19: {'kernel': 'rbf', 'C': 1, 'gamma': 0.0001}
20: {'kernel': 'rbf', 'C': 100, 'gamma': 1e-06}
21: {'kernel': 'poly', 'C': 20, 'degree': 3}
22: {'kernel': 'poly', 'C': 1, 'degree': 2}
23: {'kernel': 'rbf', 'C': 50, 'gamma': 1e-06}
24: {'kernel': 'poly', 'C': 10, 'degree': 3}
25: {'kernel': 'poly', 'C': 100, 'degree': 4}
26: {'kernel': 'poly', 'C': 5, 'degree': 3}
27: {'kernel': 'poly', 'C': 50, 'degree': 4}
28: {'kernel': 'poly', 'C': 20, 'degree': 4}
29: {'kernel': 'poly', 'C': 100, 'degree': 5}
30: {'kernel': 'poly', 'C': 10, 'degree': 4}
31: {'kernel': 'poly', 'C': 50, 'degree': 5}
32: {'kernel': 'poly', 'C': 5, 'degree': 5}
33: {'kernel': 'rbf', 'C': 1, 'gamma': 1e-06}
34: {'kernel': 'poly', 'C': 1, 'degree': 5}
35: {'kernel': 'rbf', 'C': 5, 'gamma': 1e-06}
36: {'kernel': 'poly', 'C': 1, 'degree': 4}
37: {'kernel': 'rbf', 'C': 10, 'gamma': 1e-06}
38: {'kernel': 'poly', 'C': 1, 'degree': 3}
39: {'kernel': 'rbf', 'C': 20, 'gamma': 1e-06}

40: {'kernel': 'poly', 'C': 20, 'degree': 5} 41: {'kernel': 'poly', 'C': 5, 'degree': 4} 42: {'kernel': 'poly', 'C': 10, 'degree': 5}

Using the above information, it appears that for the linear kernel there is not a noticeable effect on accuracy in changing the size of the margin, for the polynomial kernel degree is inversely proportional to accuracy while the margin size seems to be proportional, and for the rbf kernel there seems to be no proportionality between the parameters and the accuracy. I should note that the lack of proportionality stated above does not have a relationship, it just means the parameter variation is not granular enough to reveal any patterns.

As it would be impractical to perform an analysis of the accuracy of each model, I instead calculated the accuracy of the best set of parameters for each kernel type (linear, polynomial, rbf) using the classification_report() function of the sklearn.metrics module:

# 1: {'kernel': 'poly', 'C': 100, 'degree': 2}	precision recall f1-score support 3 0.96 0.99 0.97 204 8 0.98 0.96 0.97 192 avg / total 0.97 0.97 0.97 396
# 4: {'kernel': 'rbf', 'C': 50, 'gamma': 0.0001}	precision recall f1-score support 3 0.95 0.98 0.96 204 8 0.97 0.94 0.96 192 avg / total 0.96 0.96 0.96 396
# 12: {'kernel': 'linear', 'C': 1, 'degree': 1}	precision recall f1-score support 3 0.92 0.96 0.94 204 8 0.95 0.92 0.93 192 avg / total 0.94 0.94 0.94 396

The next page shows some picture examples of support vectors of the above linear kernel: