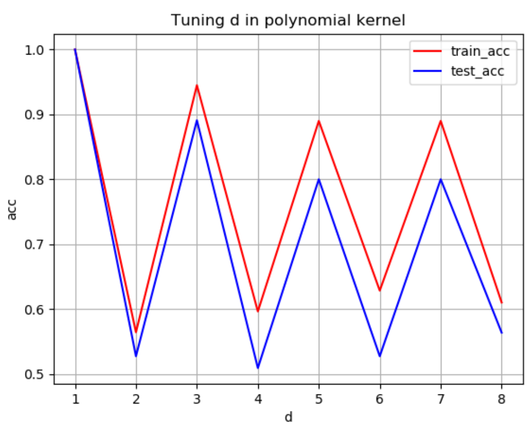
Problem 1

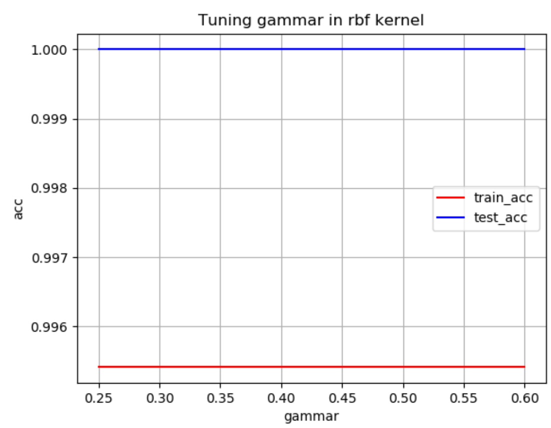
	linear	polynomial(d=2)	rbf
train_acc	97.5%	99.38%	95%
test_acc	88.75%	82.5%	91.25%

We could see that the test accuracy of **rbf kernel** is the highest among three kernels because it can produce more sophisticated boundaries in a multi-dimensional feature space. And **linear kernel** has a very high train accuracy which reflects the model is overfitting and the training data is highly separable with severl outliners. And I set parameter d in **polynomial kernel** to 2 in order to prevent it from degrading to a linear kernel. But surprisingly the result accuracy of polynomial kernel is even lower than a linear kernel.

Problem 2

Figure 1: Tuning parameters in polynomial kernel and rbf kernel on **fake-data1**





As we can see on the left graph, the accuracy of polynomial kernel is heavily affected by the parity of parameter d. Since **fake-data1** is a linear separable, $d = 1$ turns the polynomial kernel into a linear kernel and thus getting the best result. On the contrary, polynomial kernel with d is a even number fits perfectly with **fake-data2** whose boundary is a circle.

For some reason, rbf kernel does not behave differently when γ changes.

Problem 3

Figure 2: Decision boundaries on **fake-data1**

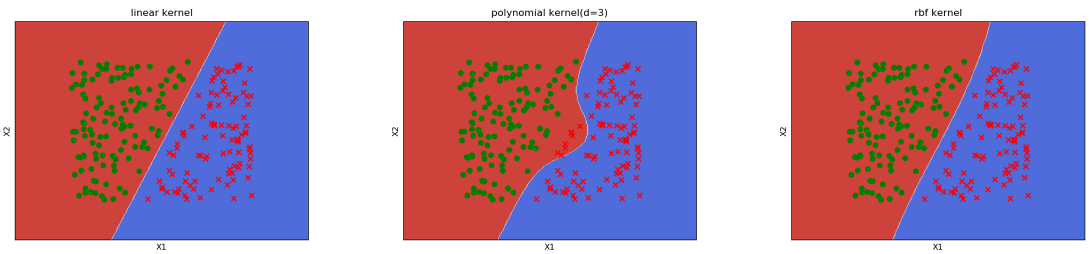


Figure 3: Decision boundaries on **fake-data2**