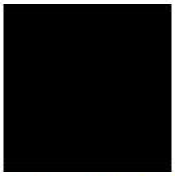
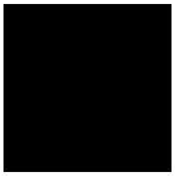
Question 3

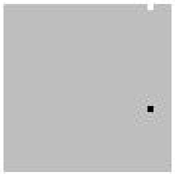
L1 Regularization



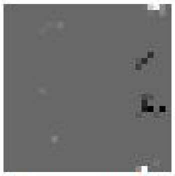
lambda = 10000000.0
accuracy = 0.492882562278



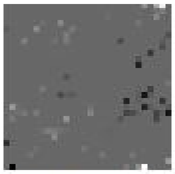
lambda = 1000000.0
accuracy = 0.492882562278



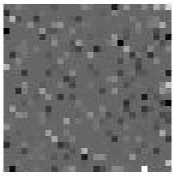
lambda = 100000.0
accuracy = 0.891459074733



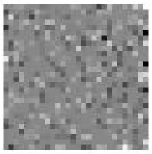
lambda = 10000.0
accuracy = 0.945729537367



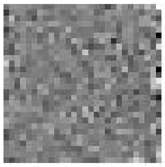
lambda = 1000.0
accuracy = 0.955516014235



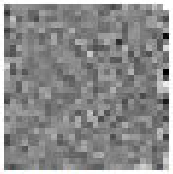
lambda = 100.0
accuracy = 0.953736654804



lambda = 10.0
accuracy = 0.940391459075

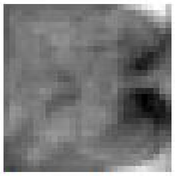


lambda = 1.0
accuracy = 0.940391459075

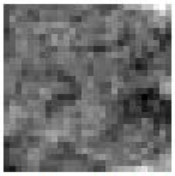


lambda = 0.1
accuracy = 0.937722419929

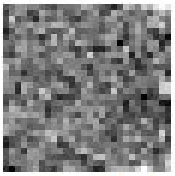
L2 Regularization



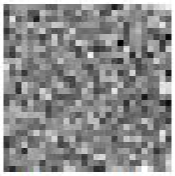
lambda = 10000000.0
accuracy = 0.955516014235



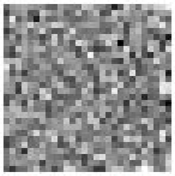
lambda = 1000000.0
accuracy = 0.960854092527



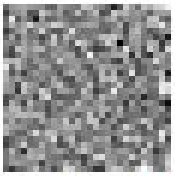
lambda = 100000.0
accuracy = 0.952846975089



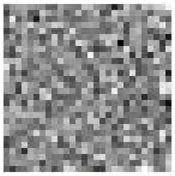
lambda = 10000.0
accuracy = 0.940391459075



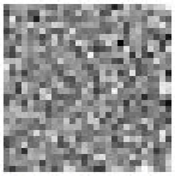
lambda = 1000.0
accuracy = 0.936832740214



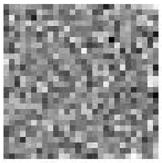
lambda = 100.0
accuracy = 0.936832740214



lambda = 10.0
accuracy = 0.935943060498



lambda = 1.0
accuracy = 0.936832740214



lambda = 0.1
accuracy = 0.934163701068

The images we printed are of the weights vectors. The images which appear “smoother” i.e. evenly spread out, have weights which are evenly distributed. The more “grainy” images on the other hand have weights which are more sparsely distributed i.e. some weights have very high value and some have very low values.

Observations

1. Maximum accuracy on test data is achieved with L1 loss with gamma = 1000 (C=0.001) and L2 loss with gamma = 10^7.
2. Lower the value of C, more strict the regularization. Changes clearly seen on tinkering with the value of C while considering L2 loss.

Conclusion

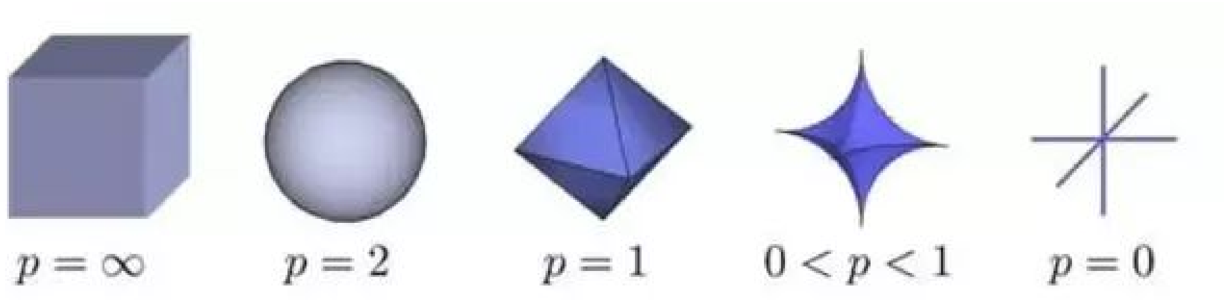
In general, lower the value of C, higher the degree of overfitting. This results in good results on training data and bad results on test data. As a result. Decreasing Gamma beyond a certain point (Gamma=0.1 for L1 and Gamma=1000 for L2) would lead to poor results.

Inferences

1. As we vary lambda, we see a change in distribution of the weights.
    a. Increasing the lambda means that we’re giving more weightage to the regularization term in the loss function, which will tend to make the weights more evenly distributed, which theoretically should make the classifier more general at cost of wrongly classifying a few of the data points.

    b. Decreasing lambda on the other hand means we're giving more weightage to correctly classifying the training data w.r.t. to the regularization term, which may sometimes lead to overfitting.

2. We observed that our weights were in the range [-1, 1], so, between L1 and L2 we see:
    a. Since L2 squares the weights, the regularization loss would be smaller. Hence, weights would react slower to change in lambda.
    b. Whereas L1 directly takes sum of the absolute values of the weights, so, the regularization loss would be higher, hence, the change in weights would be more prominent with changing lambda.



Lasso forces the sum of the absolute value of the regression coefficients to be less than a fixed value, which forces certain coefficients to be set to zero, effectively choosing a simpler model that does not include those coefficients. In other words, if there is a group of highly correlated variables, then lasso tends to select one variable from a group and ignore the others.

Ridge Regularization, as opposed to Lasso Regularization, gives preference to solutions with smaller norms, and does not force them to be set to 0.
All possible vectors of some L2-norm, say 1/2, form a unit hypersphere. Putting an L2-norm of parameters in the objective function constraints the optimised theta to a specific limited length. That is the reason, when in ridge linear regression with polynomial features of a high degree, the coefficients do not blow up, because the *feasible set* of solutions for theta is restricted to a ball.

Since L2 norm allows for a bit more "freedom" than L1 norm, we see that the values obtained here are a bit more dispersed than what is obtained in L1 norm. Typically ridge penalties are much better for minimizing prediction error rather than l1 penalties. The reason for this is that when two predictors are highly correlated, l1 regularizer will simply pick one of the two predictors. In contrast, the l2 regularizer will keep both of them and jointly shrink the corresponding coefficients a little bit.