Radial Basis Functions

We load the diabetes data from sklearn package, and use the code provided to implement a Radial Basis Function model. **Figure 1** shows the comparison between prediction and the true target.

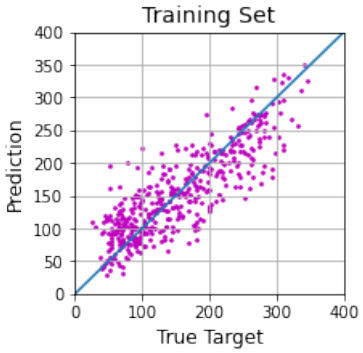

Figure 1: Prediction

Improvements

We normalize the data by using *preprocessing.scale*() from sklearn package to change the value of data to the same scale.
Then we change the width parameter to the average of several pairwise distances, this is better than just choosing the distance between two random points probably because two randomly sampled data couldn't represent the feature of the whole dataset because of the noise.
After that we use the $KMeans$ function from sklearn package to cluster the data and set the basis function location to the cluster centers.
Finally we estimate the model on training set and test its performance on training set and test set. **Figure 2** shows the prediction on training set and test set.

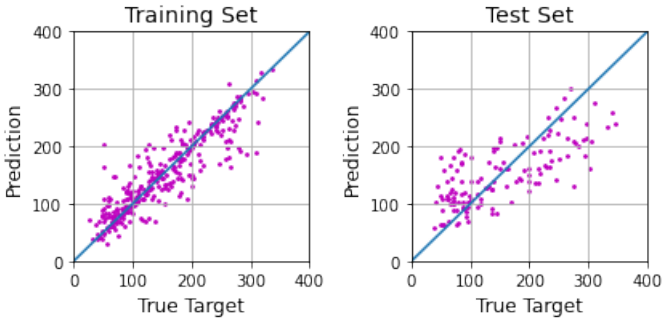

Figure 2: Prediction on training set and test set

Ten-fold cross validation

$KFold()$ from sklearn package is used to implement ten-fold cross validation. The Mean of the 10 errors computed by using ten-fold on training set is 1430.5 while the mean error on test set is 3362.1($M = 200$).

Overtraining

By changing the number of basis functions used M, we train the model on training set(training set : test set = 0.8 : 0.2), and compute errors on training set and test set for each M. **Figure 3** shows the errors with different M. It is clear that the model is overtrained because the error on training set keeps decreasing while the error on test set slightly goes up which shows the sign of overfitting.

Figure 3: Errors with different M

Boxplot display

We use ten-fold cross validation to train the Linear Regression predictor, and get 10 test errors for each validation. **Figure 4** shows the test errors on RBF model we trained before and the test errors on Linear model. It seems like in this case the RBF model is not as good as the Linear model.

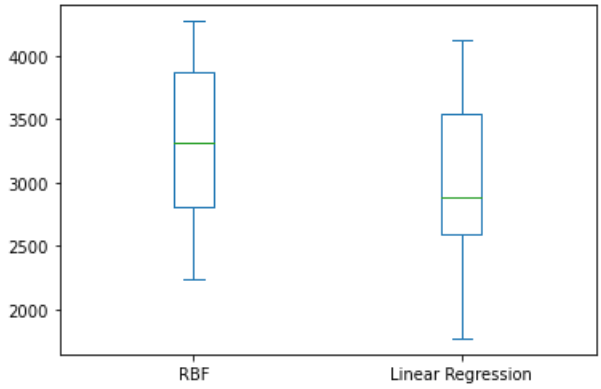

Figure 4: Regularization path