Abdul Rehman Liaqat

Lab Course Machine Learning

Exercise Sheet 6

Report

Libraries and methods used:

A snapshot of libraries and methods.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn as skl
from sklearn import linear_model
from sklearn.preprocessing import PolynomialFeatures
from sklearn.grid_search import GridSearchCV
from sklearn import cross_validation
from sklearn.cross_validation import cross_val_score
import sklearn.metrics as sklm
import random as rd
```

Data Sets:

Wine quality dataset was used for task 1. For task 2 a random uniform data array was generated using following code:

```
mean = 1
sigma= 0.05
x= [rd.normalvariate(mean,sigma ) for i in range(100)]
```

Abdul Rehman Liaqat

Exercise 1: Generalized Linear Models with Scikit Learn

1- Data was split into train and test using user defined separate() function. Data was standardized by converting non-numeric values to numeric, normalizing it and shuffling the whole data once.

2- Following function would return the RMSE for training dataset and RMSE for test dataset using SGDRegressor function. For all type of models SGDRegressor function was used.

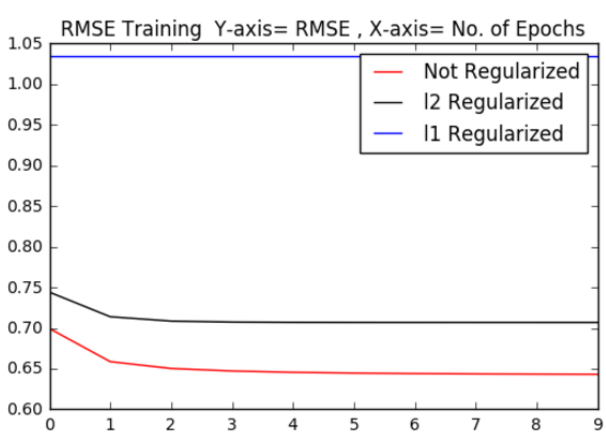
```
# rmseTrain,rmseTest=model(xtrain,xtest,ytrain,ytest,Epochs max,StepSize0,StepSizeUpdate,Regularizaiton Type,Lambda)
def model(xtrain,xtest,ytrain,ytest,itr,StepSize0,SSChange,regularizationType,lamb):
    rmseTrain=np.zeros(itr)
    rmseTest=np.zeros(itr)
    for i in range(itr):
        clf=skl.linear_model.SGDRegressor(penalty=regularizationType,
                                          n_iter=i+1,alpha=lamb,eta0=StepSize0,learning_rate=SSChange,shuffle=False)
        clf.fit(xtrain,ytrain)
        rmseTrain[i]=skl.metrics.mean_squared_error(np.dot(xtrain,clf.coef_),ytrain)
        rmseTest[i]=skl.metrics.mean_squared_error(np.dot(xtest,clf.coef_),ytest)
    return rmseTrain,rmseTest
```

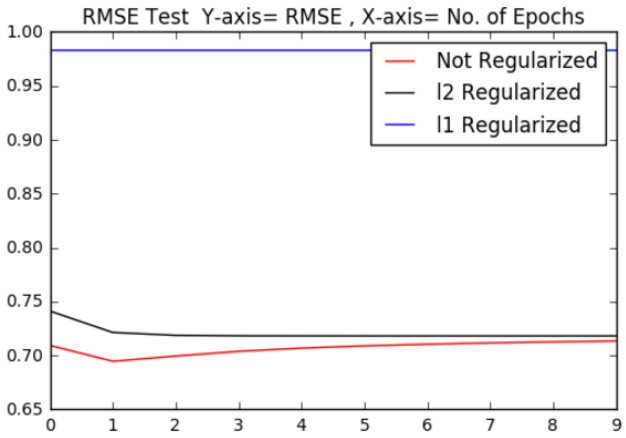
Same Hyperparameter Different models:

First same hyperparameters for different models were used and resultant RMSEs had following shape. (Note: in the figures below y-axis represent RMSE while x-axis represent epochs. To generate many epochs SGDRegressor function was called by varying the value of epochs and then plotting the respective RMSEs)
For the following plots initial step size is kept constant (0.0005) and regularization constant is varied.

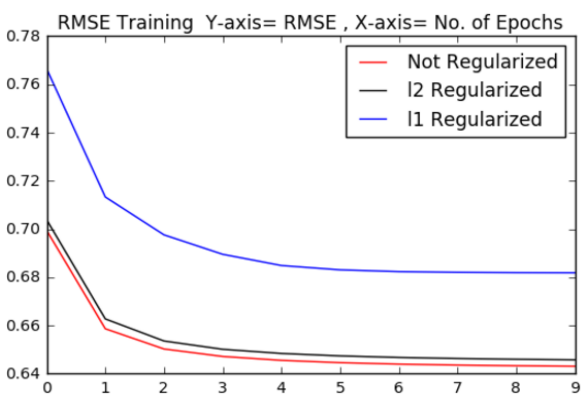
Abdul Rehman Liaqat

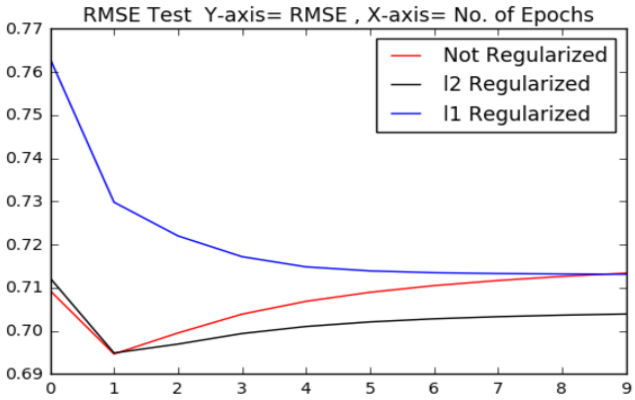
Regularization Constant= 1



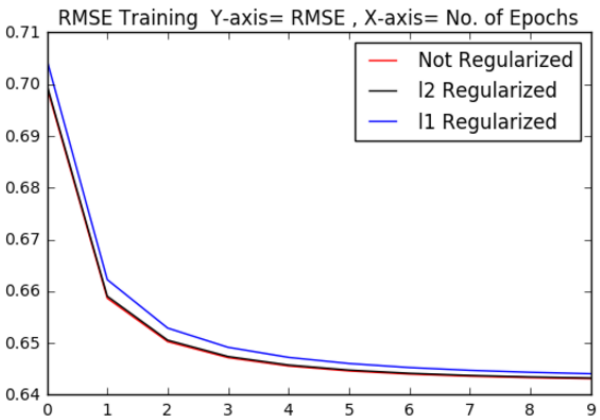


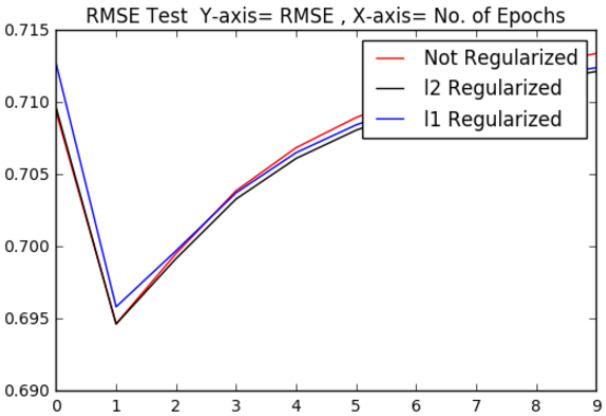
Regularization Constant= 0.1





Regularization Constant=0.01





Abdul Rehman Liaqat

There are following results that can be deduced from the given plots:

- Model performance is highly dependent upon hyperparameters. Which is why after a $1/10^{th}$ change in a one hyper parameter while keeping other fixed has caused one model to outperform other.
- For big regularization constants models with regularization perform worse than non-regularized model i.e. when regularization constant was 1.
- With very small regularization constant models with regularization perform almost the same as non-regularized model i.e. when regularization constant was 0.01. Also, as we decrease the value this similarity will keep on increasing until regularization constant value has approached zero.
- Models RMSE for test data or real-life predictions will not necessarily be the same.
- Non-regulated model doesn't perform well on test data but has good performance for training data which means it is suffering from the problem of over-fitting which is why regularization is necessary.
- Since the value of regularization constant varies from zero to maximum and (hence model will vary from overfitting to underfitting respectively), hyperparameters such as this should be optimized. Until an optimum solution is achieved which is performing best for test data.

Abdul Rehman Liaqat

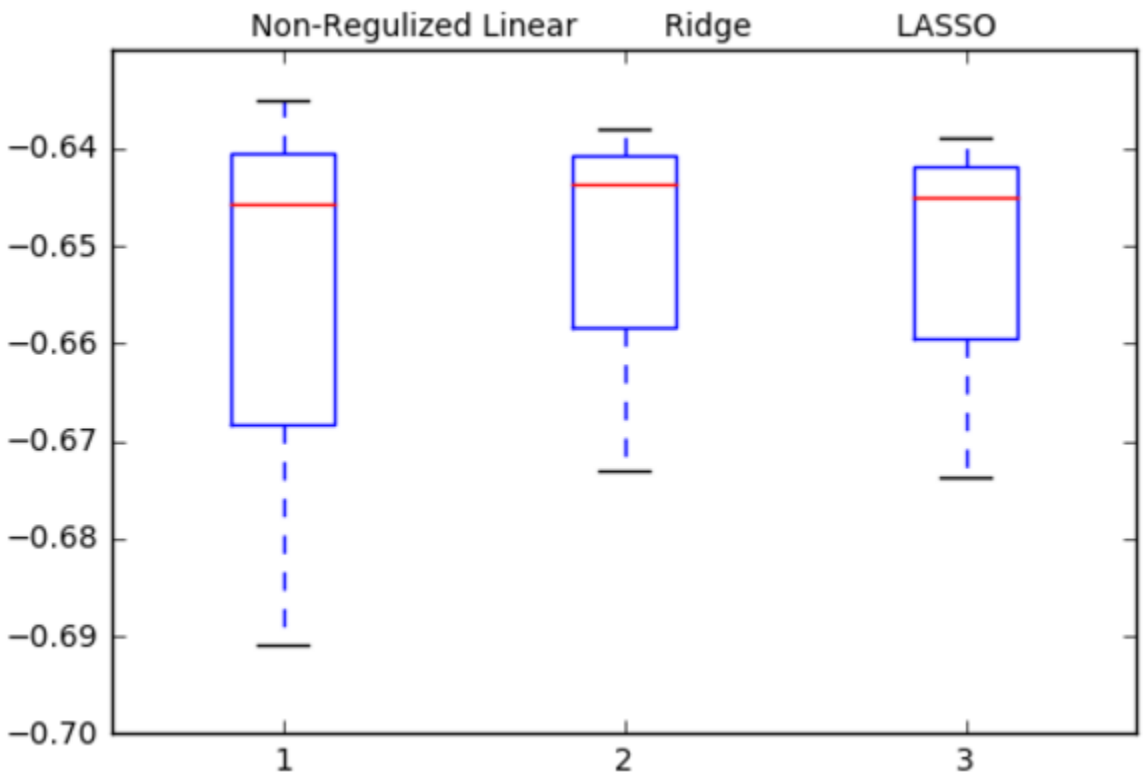
3- Each model was tuned for Initial step size and regularization constant (not for non-regulared model). Grid search output a best set of parameters which is has best k-fold performance mean. Each grid search for each model will output only one best solution which are following:

Model Name	RMSE Test	RMSE Train K-fold mean	RMSE Train whole	Best combination
Ordinary Least Square	0.733	0.657	0.648	Step=0.01 Lambda=0
Ridge Regression	0.711	0.651	0.648	Step=0.01 Lambda=0.1
LASSO	0.714	0.652	0.647	Step=0.01 Lambda=0.01

This table also proves the preposition proposed in the previous part.
One thing to note is that Grid Search will select the best model based on mean performance of a model with a given set of parameters on k-folds of the data. Hence RMSE or score provided by the Grid Search will be the mean of all these performances and always more than the original RMSE of the same set of parameters when calculated on the data as a whole. Because the smaller the sample is the most difficult it will be to generalize the whole data. Hence if we break down a data into smaller parts, predict on these parts and average their performance, their average is bound to be more than the one learned on the whole data.
Same is the reason all values in “RMSE Train whole” is less than the respective “RMSE Train k-fold mean”.
Second thing is the performance of non-regulated (Ordinary Least Square) model is more similar to other models for training data as compared to test data. This again proves that reducing super fitting will generalize the model as a whole hence we will get better performance for test data and in real life.

Abdul Rehman Liaqat

4- After Grid Search now, we have access to best parameter set for each model. Validation score of k-folds for each parameter set has following box plot:



Mean and K-folds RMSE of each model is as following:

```
Ordinary Least Squares 3-folds RMSE Train =  [ 0.690827    0.64569828  0.63513492]
Ordinary Least Squares 3-folds RMSE Train mean =  0.657220064103
Ridge Regression 3-folds RMSE Train =  [ 0.67300761  0.64366772  0.6379588 ]
Ridge Regression 3-folds RMSE Train mean =  0.65154470879
LASSO 3-folds RMSE Train =  [ 0.67377594  0.64504159  0.63889836]
LASSO 3-folds RMSE Train mean =  0.65257195998
```

If we compare the mean values of achieved through Cross_val_Score they are exactly the same obtained from the previous (Grid Search) method.
Moving on to the box plot given above, it shows the variance of RMSE around the mean. In other words, for Non-Regulated model we should expect more fluctuations of test data prediction and less fluctuations in regulated model. This is the very reason non-regulated model doesn't perform better when tested on test data because they have more error range hence more chances of the predictions to go wrong.
This is what regulation do. Reduce the overfit. Reduce the effect of outliers for linear models specifically.

Abdul Rehman Liaqat

Exercise 2: Polynomial Regression

Task A:

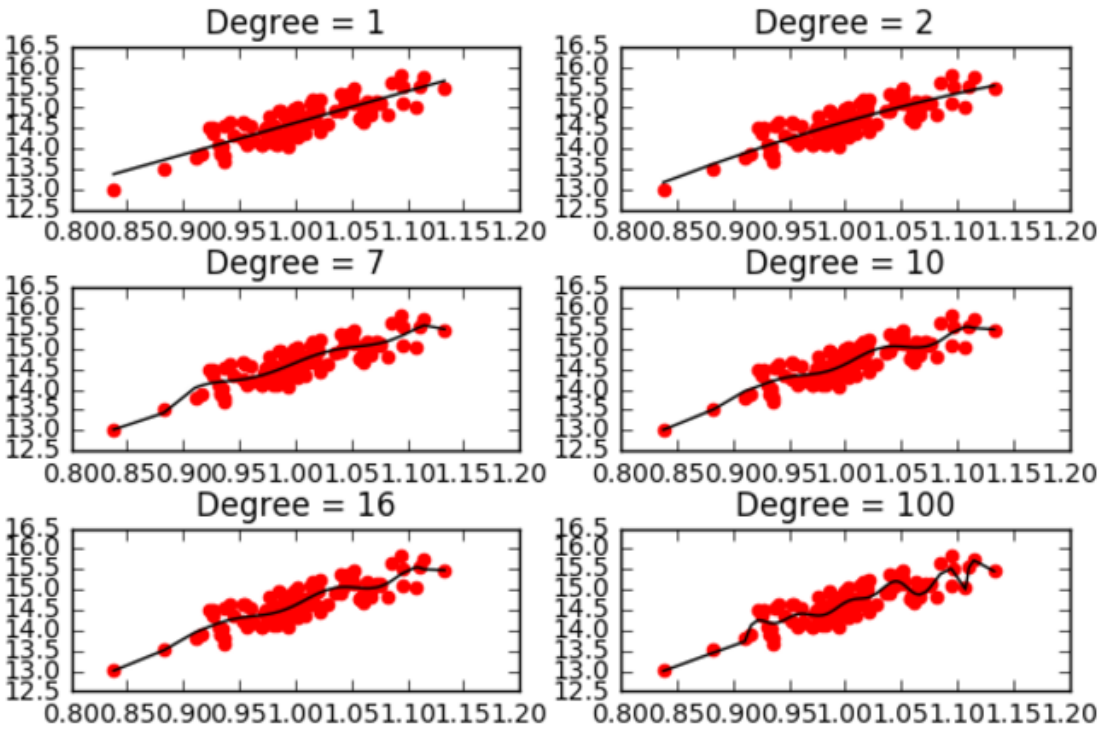
Degrees of Polynomial equations: 1,2,7,10,16,100

Note: To make the plots more visually understandable, box x and y were sorted with respect to x at initiation.

X was converted into polynomial data using PolynomialFeatures function. For each degree of polynomial, we get following result. RMSE train of the above degrees is:

POLYNOMIAL DEGREE	RMSE
1	0.0933
2	0.093
7	0.0917
10	0.087
16	0.086
100	0.082

And we get the following plots:



Abdul Rehman Liaqat

As it can be seen from the plots that by increasing the polynomial degree RMSE training decreases and model becomes better fit for the training data but obviously issue of overfitting and complexity also start springing up.

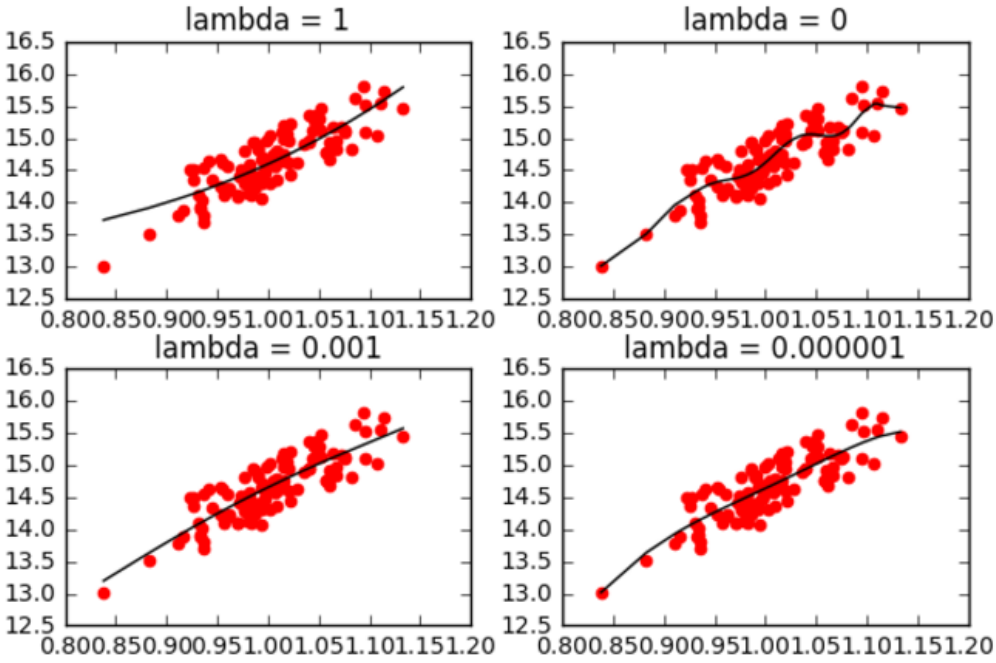
Task B:

Degree of the polynomial is fixed to 10.

After applying ridge regression, we get the following results:

REGULARIZATION CONSTAT VALUE	RMSE
1	0.08
0	0.066
0.001	0.0723
0.000001	0.071

And we get the following plots:



Abdul Rehman Liaqat

As discussed before non-regulated high degree polynomial have less RMSE for training data. It can be again proved by looking at the table given. When regularization constant has zero value (meaning no regularization) RMSE is minimum. Also, the values which are close to zero has least RMSE.
Similarly plots with less value is more closer to the one with 0 value meaning the plot is still overfit and won't generalize well.