Shivani_gowda_ps2

October 3, 2021

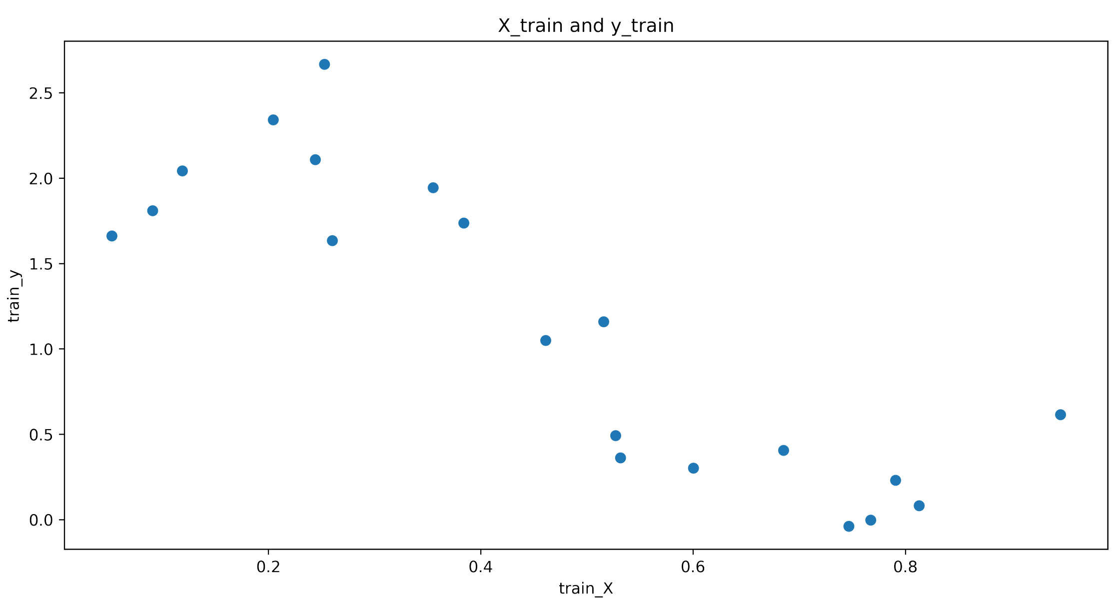
[1]:
```
%load_ext autoreload
%autoreload 2
%matplotlib inline
```

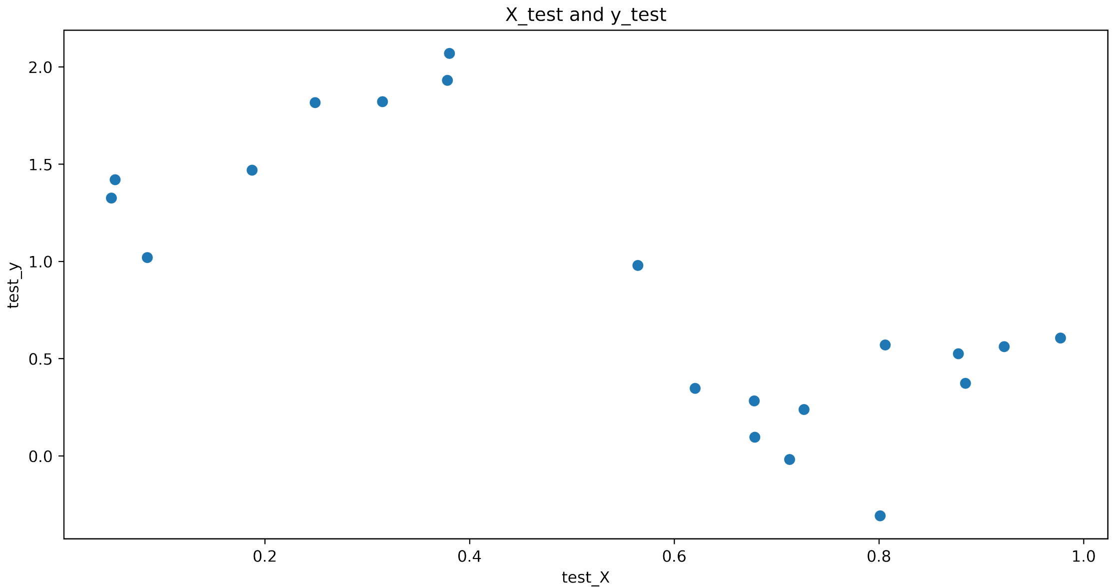
[5]:
```
import sys
from pathlib import Path
import matplotlib.pyplot as plt


plt.rcParams["figure.figsize"] = (12, 6)
plt.rcParams['figure.dpi'] = 600
if 'source' not in sys.path:
    sys.path.append('source')
```

[8]:
```
import regression
from regression import main
```

[19]:
```
main()
```

```
Visualizing data…
```

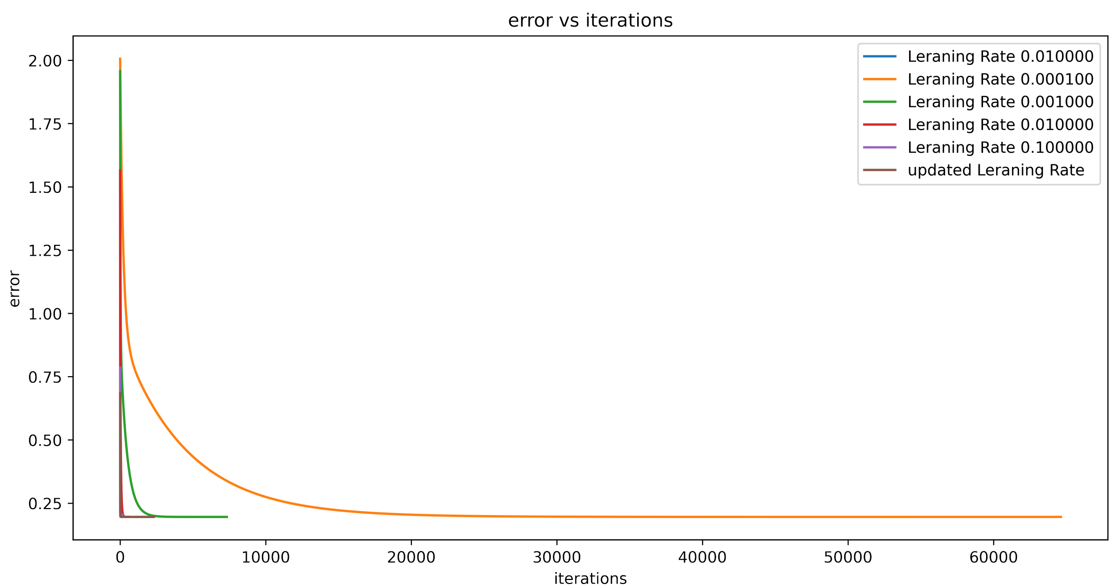


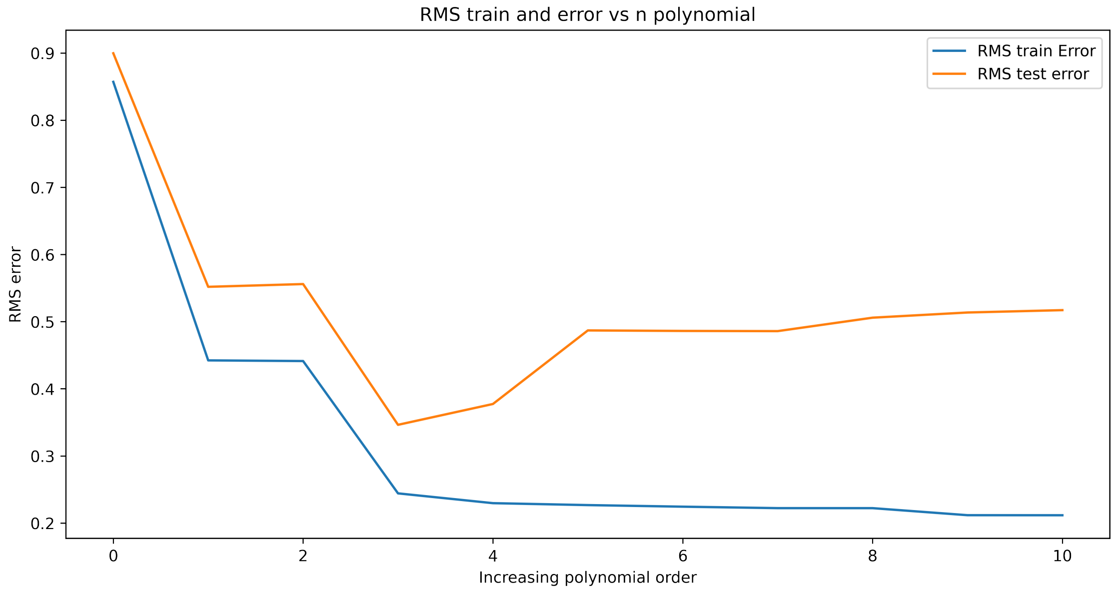



```
Investigating linear regression…
[[1.        0.515773]
 [1.        0.790645]
 [1.        0.685289]
 [1.        0.946009]
 [1.        0.53169 ]
 [1.        0.118853]
 [1.        0.600345]
 [1.        0.090962]
 [1.        0.355033]
 [1.        0.204493]
 [1.        0.812833]
 [1.        0.252854]
 [1.        0.527025]
 [1.        0.260206]
 [1.        0.244096]
 [1.        0.383914]
 [1.        0.767244]
 [1.        0.461121]
 [1.        0.746685]
 [1.        0.052486]]


coef:  [6.578865 7.953225 7.426445 8.730045 6.65845  4.594265 7.001725 4.45481
 5.775165 5.022465 8.064165 5.26427  6.635125 5.30103  5.22048  5.91957
```

```
 7.83622  6.305605 7.733425 4.26243 ]


cost function: 314.5730363712805


number of iterations: 616
sgd solution: [ 2.44078184 -2.81863861]
Time for sgd solution:  0.09516255598282441


table for varying alpha size
number of iterations: 64620
number of iterations: 7330
number of iterations: 616
number of iterations: 152


    alpha  iterations                       coef
0  0.0001       64620  [ 2.44509281 -2.81382739]
1  0.0010        7330  [ 2.44555307 -2.81594059]
2  0.0100         616  [ 2.44078184 -2.81863861]
3  0.1000         152  [ 2.38405089 -2.87906028]


closed_form solution: [ 2.44640709 -2.81635359]
Time for closed_form solution:  0.00021810800535604358


Results with eta = None
number of iterations: 2320
sgd solution after predicting tmax: [ 2.44652801 -2.81718713]


Investigating polynomial regression…
Polynomial feature: [[1.         0.515773   0.26602179]
 [1.         0.790645   0.62511952]
 [1.         0.685289   0.46962101]
 [1.         0.946009   0.89493303]
 [1.         0.53169    0.28269426]
 [1.         0.118853   0.01412604]
 [1.         0.600345   0.36041412]
 [1.         0.090962   0.00827409]
 [1.         0.355033   0.12604843]
 [1.         0.204493   0.04181739]
 [1.         0.812833   0.66069749]
 [1.         0.252854   0.06393515]
 [1.         0.527025   0.27775535]
```

```
 [1.         0.260206   0.06770716]
 [1.         0.244096   0.05958286]
 [1.         0.383914   0.14738996]
 [1.         0.767244   0.58866336]
 [1.         0.461121   0.21263258]
 [1.         0.746685   0.55753849]
 [1.         0.052486   0.00275478]]


RMS_error =  [7.79451604]


Investigating regularized regression…
RMS train minimum error: %f [0.211693]
RMS test minimum error: %f [0.34635263]
```






```
3rd order is giving the lowerst RMS error
Done!
```

0.1 Visualization:

0.1.1 1.

The scatter plot X-train vs y-train and x-test and y-test has many raise and dips, hence cannot be eaily fitted into a linear solution. And hence using a polynomial function would give more accuracy.

0.2 Linear Regression:

0.2.1 4(c).

The coffecients are alomst the same except for the small change in decemials but for the one with learning rate 0.1, there is comparitively more chnage observed. After computing closed form solution I have observed that both extreamly high and low rates are yeilding coeffecients that are slighlty different than closed forms. As the learning rate increases, the time taken for convergence is small.

0.2.2 5(b).

I have obtained alomst same coffecients for both the SGD and closed-form solution, except for some minor changes in the least significant decemials. ### 5(c). The runtime for SGD is 0.1300 seconds and for closed solution it is 0.0002 seconds and hence it can be conculded that SGD takes more time while comapred to closed form solution.

0.2.3 6.

With the prososed learning rate the algorithm is taking more time to converge but the coffecients values are almost the same as previous.

0.2.4 9.

from the graph, RMS train and error vs n polynomial, it can be observed that, degree 3 fits the best. Because, at degree 3, we can observe that RMS test error is minimum and later it starts increasing. Overfitting can be observed in the above graph because, the training error after degree 3 decreases and remains constant, but thr RMS test error increases. This show the overfitting of the model to train data.