Home Work - 3: Machine Learning COMS4721

- Vinayak Bakshi, vb2424

Out[1]:

Click here to toggle on/off the raw code.

1 (b): RMSE Table

Out[6]:

	b = 5.0	b = 7.0	b = 9.0	b = 11.0	b = 13.0	b = 15.0
σ^2 = 0.1	1.966276	1.920163	1.897649	1.890507	1.895849	1.909603
σ^2 = 0.2	1.933135	1.904877	1.902519	1.914981	1.935586	1.959549
σ^2 = 0.3	1.923420	1.908080	1.917648	1.938849	1.964597	1.990804
σ^2 = 0.4	1.922198	1.915902	1.932514	1.957936	1.985502	2.011915
σ^2 = 0.5	1.924769	1.924804	1.945699	1.973216	2.001314	2.027370
σ^2 = 0.6	1.929213	1.933701	1.957235	1.985764	2.013878	2.039465
σ^2 = 0.7	1.934634	1.942254	1.967403	1.996375	2.024310	2.049463
σ^2 = 0.8	1.940583	1.950380	1.976492	2.005603	2.033307	2.058105
σ^2 = 0.9	1.946820	1.958093	1.984741	2.013835	2.041317	2.065845
σ^2 = 1.0	1.953213	1.965438	1.992341	2.021345	2.048642	2.072976

1 (c): Best RMSE value = 1.8905, for b=11, sigma = 0.1

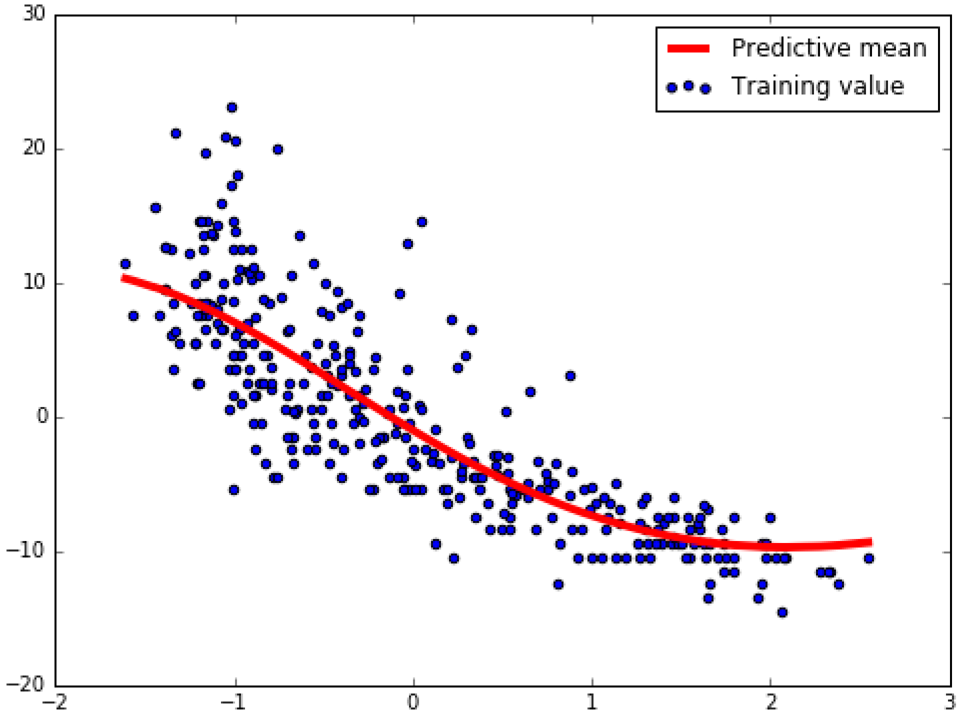
Compared to the first homework where we use ridge regression, we obtain a minimum RMSE of **2.192** for hyper-parameter =21 and polynomial of degree 2. By using Gaussian process we obtain a minimum RMSE of **1.8905**. Therefore this approach is better in terms of accuracy.

Drawback: We do not obtain feature weights as output from the Gaussian process implementation. Hence we do not know the importance of features in the prediction of output since they are being mapped to a higher dimensional space.

1 (d): Gaussian Process for 4th dimension

Out[9]:

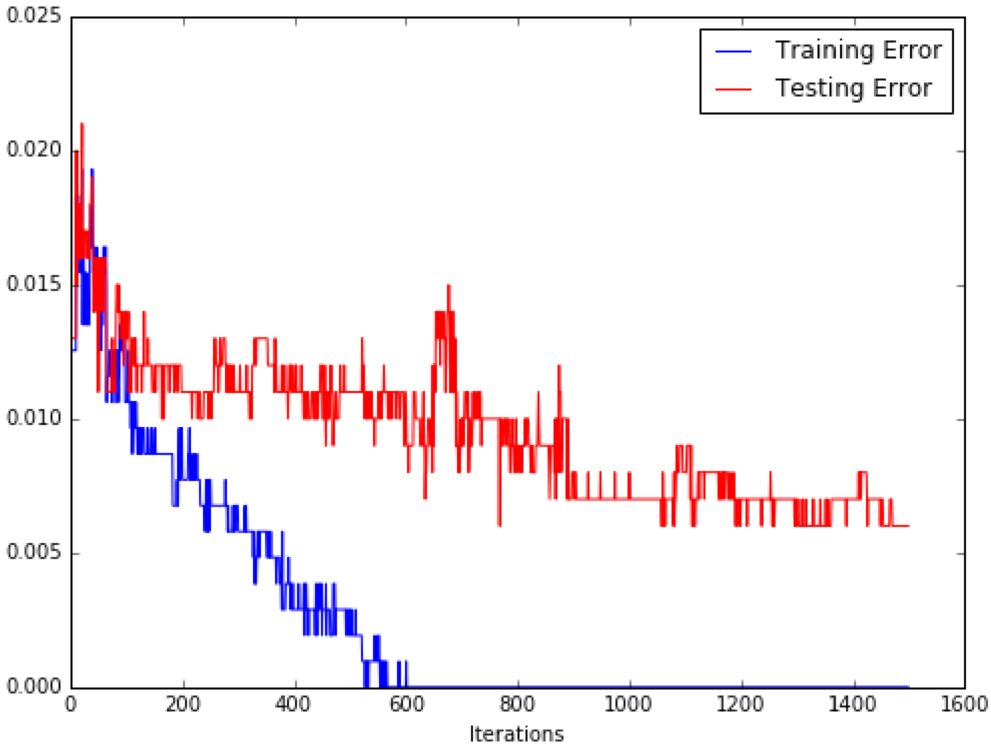
<matplotlib.legend.Legend at 0x858b320>



2(a): Implementation of boosting and training test error plot

Out[12]:

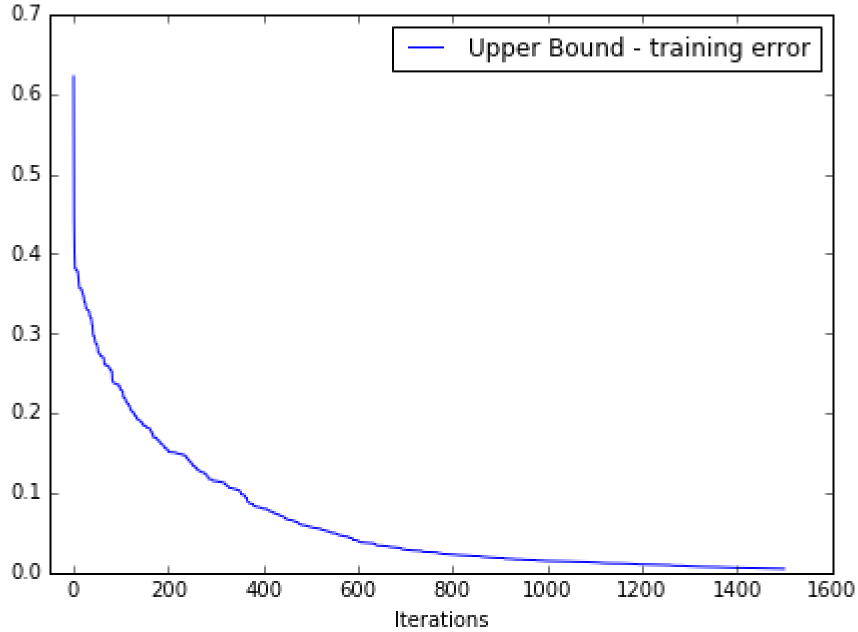
<matplotlib.legend.Legend at 0xa68a7f0>



2(b): Upper Bound on the training error

Out[18]:

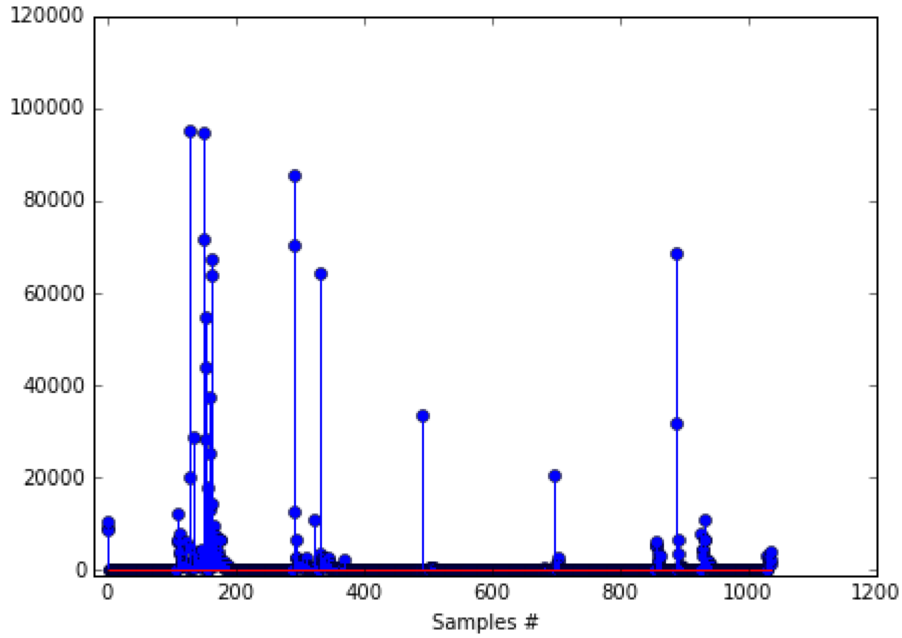
`<matplotlib.legend.Legend at 0x16fbb4e0>`



2(c): Histogram of samples

Out[20]:

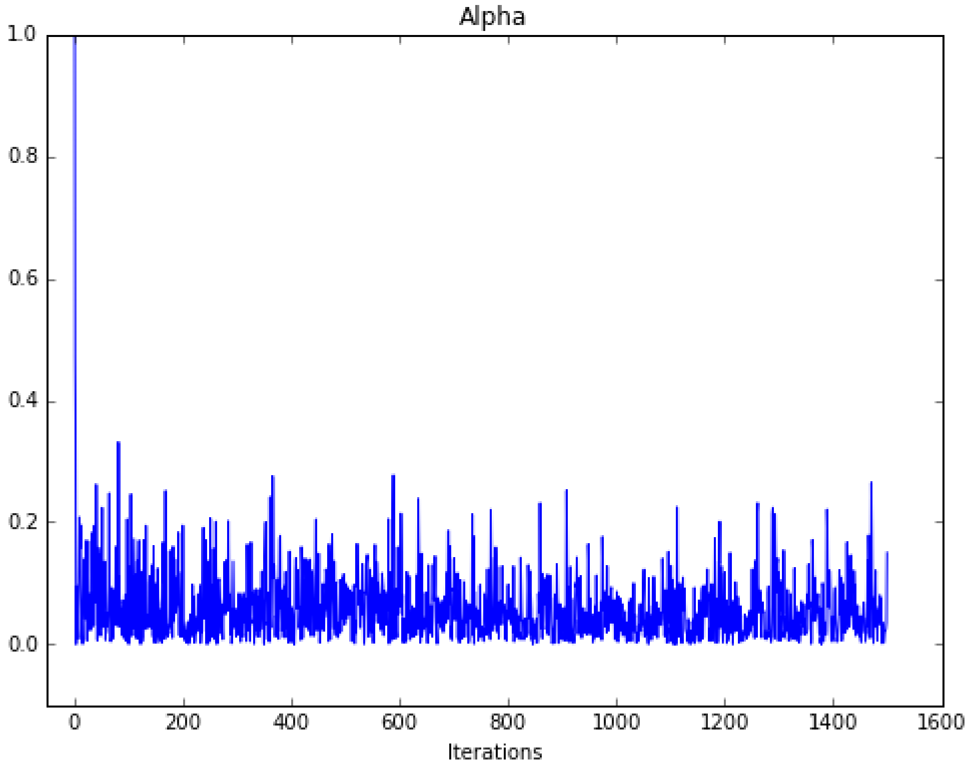
(-20, 1200)



2(d): Epsilon, Alpha as a function of iterations

Out[15]:

<matplotlib.text.Text at 0x16fa7ac8>



Out[16]:

<matplotlib.text.Text at 0x17196208>