Model Order Selection. In-class Exercise 1 Solution

EL-GY 6143 Intro Machine Learning. Prof. Sundeep Rangan

Question

Complete the following items in the demo, demo_polyfit.ipynb.

In-class Exercise To inspire you think about a possible way to find the correct model order, modify the above code to do the following:

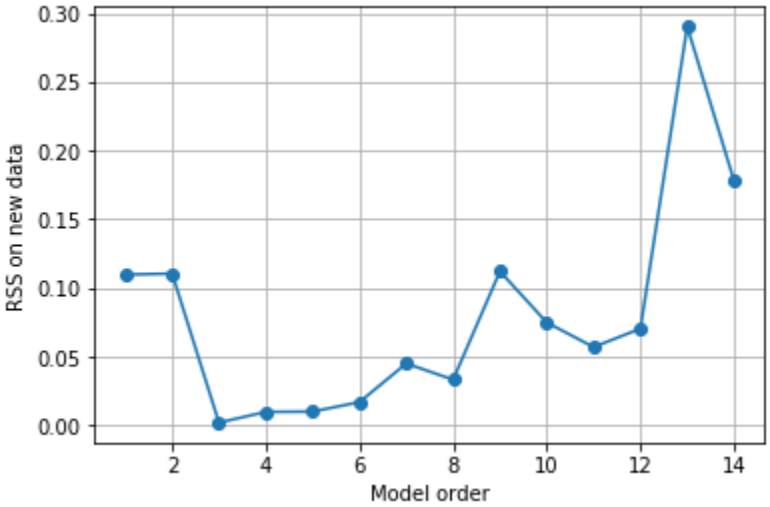
- Generate a *new* set of 100 data points, `xnew`, in the interval `[-1,1]`. You can use the `np.random.uniform()` `np.linspace()` function. You will need to look up their syntax.
- Compute corresponding labels for the points, `ynew = poly.polyval(xnew,beta)`. These would be the labels of the *true* function without noise corresponding to the inputs `xnew`.
- For each model order, `d`: find the parameters `beta_hat` from `(xdat,ydat)` as before.
- But, measure the RSS on the new data `(xnew,ynew)`.
- Plot the RSS on the new data.
- Which model order produces the minimum RSS when measured on the new data?

Of course, you cannot implement this procedure in reality, since it requires that you get another 100 data points without noise. But, we will see how to implement a good approximation of this procedure.

Solution

Solution The modified code is below. When you run it, you see that the RSS on the new data increases as you select model orders `d > 3`. So, one way to select the correct model order is to somehow use *new* data not part of training.

```
# TODO

# Generate new test data
xnew = np.linspace(-1,1,100)
ynew = poly.polyval(xnew, beta)

dtest = np.array(range(1,15))
RSS_new = []

for d in dtest:

    # Fit data on the training data
    beta_hat = poly.polyfit(xdat,ydat,d)

    # Measure RSS on the *new* data
    yhat = poly.polyval(xnew,beta_hat)
    RSSd = np.mean((yhat-ynew)**2)
    RSS_new.append(RSSd)

# Plot the RSS on the new data
plt.plot(dtest,RSS_new,'o-')
plt.xlabel('Model order')
plt.ylabel('RSS on new data')
plt.grid()
```


0.30
0.25
0.20
0.15
0.10
0.05
0.00
RSS on new data
2
4
6
8
10
12
14
Model order