Machine Learning: Homework 2

Overfitting

Ana Mestre

Problem

We set up an experimental framework to study various aspects of overfitting. The input space is X = [-1,1] with uniform input probability density, P(x) = ½. We consider the two models H2 and H10. The target function is a polynomial of degree Qf, which we write as $f(x) = \Sigma\, a_q * L_q(x)$, where Lq (x) are the Legendre polynomials.

We use the Legendre polynomials because they are a convenient orthogonal basis for the polynomials on [-1,1].

The data set is D = (x1, y1), . . . , (xN, yN), where $y_n = f(x) + \sigma\varepsilon_n$ and ε_n are i.i.d. Standard Normal random variables.

For a single experiment, with specified values for Qf , N, σ, generate a random degree-Qf target function by selecting coefficients aq independently from a standard Normal distribution, rescaling them so that Ex[f(x)^2] = 1. Generate a data set, selecting x1, . . . ,xN independently from P(x) and $y_n = f(x) + \sigma\varepsilon_n$. Let g2 and g10 be the best fit hypotheses to the data from H2 and H10, respectively, with respective out-of-sample errors Eout (g2) and Eout (g10).

(a) Why do we normalize f ? [Hint: how would you interpret σ?]

Knowing the following statements:

- Our function is: $f(x) = \Sigma a_q * L_q(x)$ and our output is: $y_n = f(x) + \sigma\varepsilon_n$, from this we can interpret σ as the noise value of the function. The noise increases as the σ increases.
- Var[f] = E[f^2] – E[f]^2
- All Legendre polynomials of degree = 0 have mean 0 on [-1,1], which means that the mean of the function f(x) on [-1,1] is E[f(x)] = a0.
- The variance of f(x) on [-1, 1] is given by:

$$Var(f(x)) = \Sigma \frac{a\_q^2}{2q+1}$$

Taking these statements into consideration and knowing that we want to get a function really close to a normal distribution, we have to normalize the coefficients such that $E(f^2) = 1$. The way of doing so is by dividing a by:

$$2\sqrt{\Sigma \frac{a_q^2}{2q+1}}$$

```python
# Get rescaled aq
def norm_aqs(Qf):
    aqs = np.random.standard_normal(Qf + 1)
    summ = sum([aqs[q]**2/(2 * q + 1) for q in range(Qf + 1)])
    return [aqs / math.sqrt(2 * summ)]
```

(b) How can we obtain g2, g10? [Hint: pose the problem as linear regression]

We build polynomials models of order 2 and 10. We use them along with the dataset D to fit a linear regression model (H2 and H10, respectively).

```
def polRegression(x, y, Qf, color, label, title="", plot = True):
    x_reshape = x.reshape(-1, 1)

    poly_feat = PolynomialFeatures(degree = Qf, include_bias = True)
    lin_reg = LinearRegression()
    x_ = poly_feat.fit_transform(x_reshape)
    lin_reg.fit(x_, y)
    y_pred = lin_reg.predict(x_)

    if plot:
        plotResults(x, y, y_pred, Qf, label, title, color)

    return x, y_pred, lin_reg, poly_feat
```

(c) How can we compute E_out analytically for a given g10?

We have to calculate: E_out(g10) = E[(g10(x) - f(x) ^{2}], which can be analyzed by the bias-variance decomposition.

Finding an f that generalizes to points outside of the training set can be done with any of the countless algorithms used for supervised learning. It turns out that whichever function f we select, we can decompose its expected error on an unseen sample x as follows:

$$\mathrm{E}\left[(y-\hat{f}(x))^2\right] = \left(\mathrm{Bias}\left[\hat{f}(x)\right]\right)^2 + \mathrm{Var}\left[\hat{f}(x)\right] + \sigma^2$$

where

$$\mathrm{Bias}\left[\hat{f}(x)\right] = \mathrm{E}\left[\hat{f}(x)\right] - \mathrm{E}\left[f(x)\right]$$

and

$$\mathrm{Var}\left[\hat{f}(x)\right] = \mathrm{E}[\hat{f}(x)^2] - \mathrm{E}[\hat{f}(x)]^2.$$

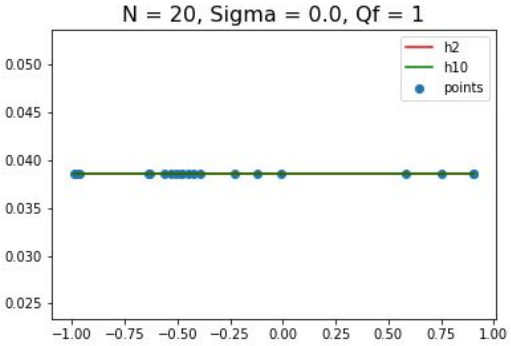
(d) Vary Qf , N, σ and for each combination of parameters, run a large number of experiments, each time computing E_out(g2) and E_out(g10). Averaging these out-of-sample errors estimates of the expected out-of-sample error for the given learning scenario (Qf ,N, σ) using H2 and H10. Let:

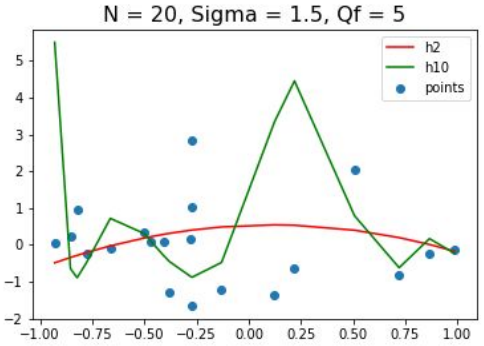
Eout (H2) = average over experiments(Eout (g2)),

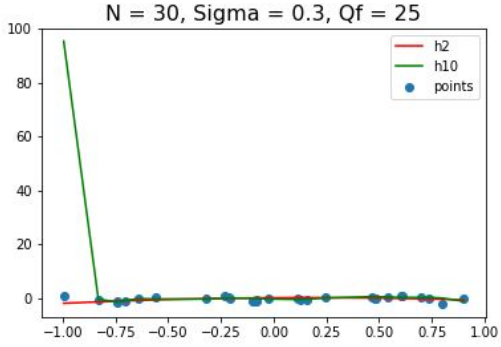
Eout (H10) = average over experiments(Eout (g10)).

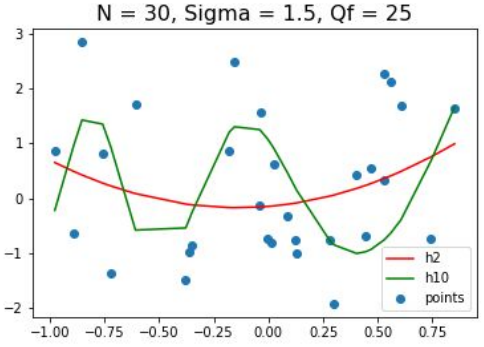
Define the overfit measure E_out(H10) - E_out(H2). When is the overfit measure significantly positive (i.e. overfitting is serious) as opposed to significantly negative?
Try the choices Qf ε{1,2, . . . ,100}, N ε {20,25,. . . ,120}, σ^2 ε {0,0.05,0.1, . . . , 2}. Explain your observations.

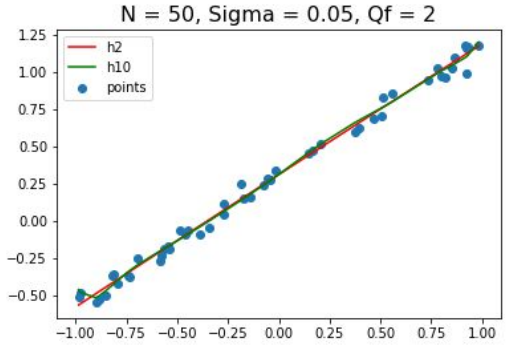
Here are some results (find more attached to the assignment's submission):

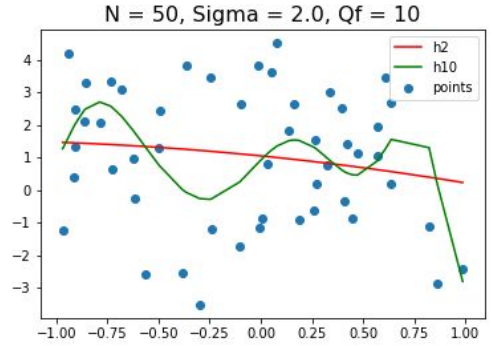


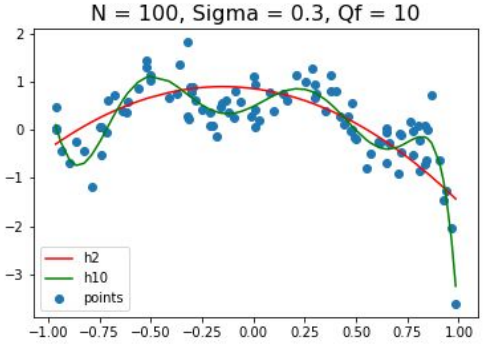


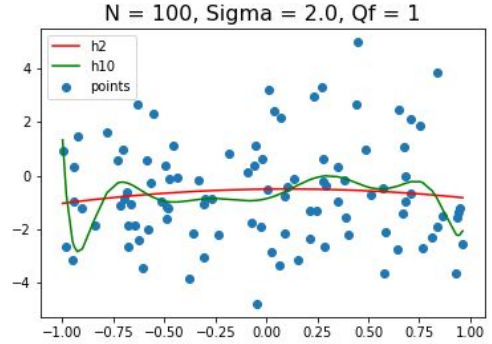


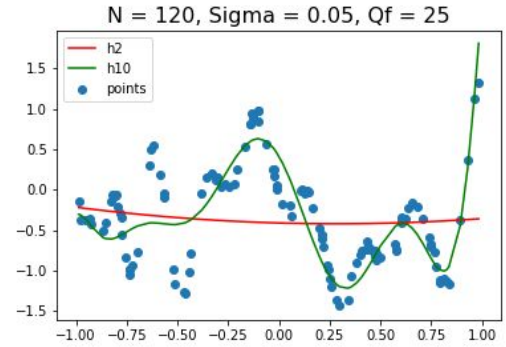


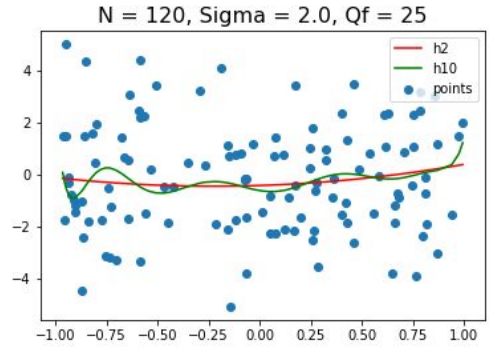

N = 50, Sigma = 0.05, Qf = 2
h2
h10
points


N = 50, Sigma = 2.0, Qf = 10
h2
h10
points


N = 100, Sigma = 0.3, Qf = 10
h2
h10
points


N = 100, Sigma = 2.0, Qf = 1
h2
h10
points


N = 120, Sigma = 0.05, Qf = 25
h2
h10
points


N = 120, Sigma = 2.0, Qf = 25
h2
h10
points

Example of results for N = 30 (more to be found attached on this assignment's submission)

N	Qf	Sigma	Mean E2	Mean E10	Overfit
30	1	0.0	0.0	0.0	0.0
30	1	0.05	0.0022	0.0026	0.0004
30	1	0.3	0.0718	0.6541	0.5823
30	1	1.0	1.1326	148.4168	147.2842
30	1	1.5	1968	56.4844	54.5164
30	1	2.0	4.3613	283.6016	279.2403
30	2	0.0	0.0	0.0	0.0
30	2	0.05	2	0.0087	0.0067
30	2	0.3	0.0863	0.5751	0.4888
30	2	1.0	1.0646	21.2056	20141
30	2	1.5	2.0266	190137	188.1104
30	2	2.0	2.1119	8.4165	6.3046
30	5	0.0	158	0.0	-158
30	5	0.05	0.0999	0.0058	-0.0941
30	5	0.3	229	0.7638	0.5348
30	5	1.0	1.0857	10.9593	9.8736
30	5	1.5	2327	10.9999	8.6729
30	5	2.0	3884	5.2269	1.3429
30	10	0.0	0.8404	0.0	-0.8404
30	10	0.05	0.2276	0.0323	-0.1953
30	10	0.3	0.2623	0.0991	-0.1632
30	10	1.0	1.0807	180.9269	179.8462
30	10	1.5	2293	2.3501	0.0571
30	10	2.0	3.2291	13.0701	9841
30	25	0.0	0.5303	6.9601	6.4298

30	25	0.05	0.2686	8.4549	8.1863
30	25	0.3	0.1809	0.2357	0.0548
30	25	1.0	1.3466	11.9003	10.5537
30	25	1.5	2.1861	5.2774	3.0913
30	25	2.0	4.5349	15.8969	11362

By taking a look at the results obtained, but especially to the table of output errors, we can observe that usually the 2nd-degree model is the one that obtains better results. This can be seen notably when the noise level is increased, that is when our sigma value is higher. Therefore, we can verify what we have studied in class, the simpler hypothesis is the one that obtains better results.

In addition, we can observe that also for higher noise values (1.5, 2 in this case), we obtain a overfit measure significantly positive and it tends to be significantly negative when the noise values are low (0, 0.05). However, there are more cases of positive overfitting than negative overfitting.

(e) Why do we take the average over many experiments? Use the variance to select an acceptable number of experiments to average over.

As we are using random variables in order to create our datasets for training, we need to perform different experiments for the sake of approximating better to the real value. We can see that for some tests using the same parameters, the output error is different. This specially happens when the amount of data is small. Therefore, after executing a different number of times every experiment, we compute the mean of these errors for getting a more realistic error.