Statistical Machine Learning: Assignment 1

Joris van Vugt, s4279859
Luc Nies, s4136748

September 19, 2016

Exercise 1

1

```
def f(x):
    return 1 + np.sin(6 * (x - 2))

def noisy_f(x):
    noise = np.random.normal(0, 0.3)
    return noise + f(x)

# Generate data
D = [noisy_f(x) for x in np.linspace(0, 1, 10)]
T = [noisy_f(x) for x in np.linspace(0, 1, 100)]
```

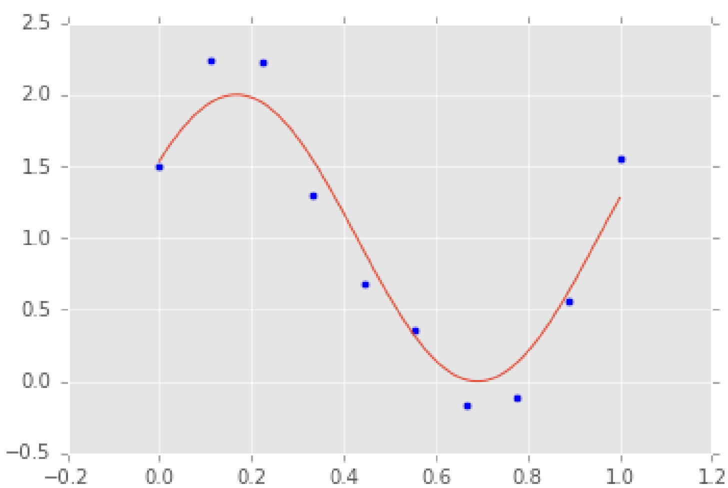

Figure 1: Plot of $f(x)$ and the 10 noisy observations in the training data

2

```
def pol_cur_fit(D, M):
    x = D[0, :]
```

```python
    t = D[1, :]
    A = np.zeros((M, M))
    for i in range(M):
        for j in range(M):
            A[i, j] = np.sum(x ** (i+j))
    T = np.zeros(M)
    for i in range(M):
        T[i] = np.sum(t * x**i)
    w = np.linalg.solve(A, T)
    return w
```

3

```python
def polynomial(X, w):
    return np.polyval(list(reversed(w)), X)

def RMSE(observed, target):
    error = 0.5 * np.sum((observed - target)**2)
    return np.sqrt(2*error / len(observed))
```

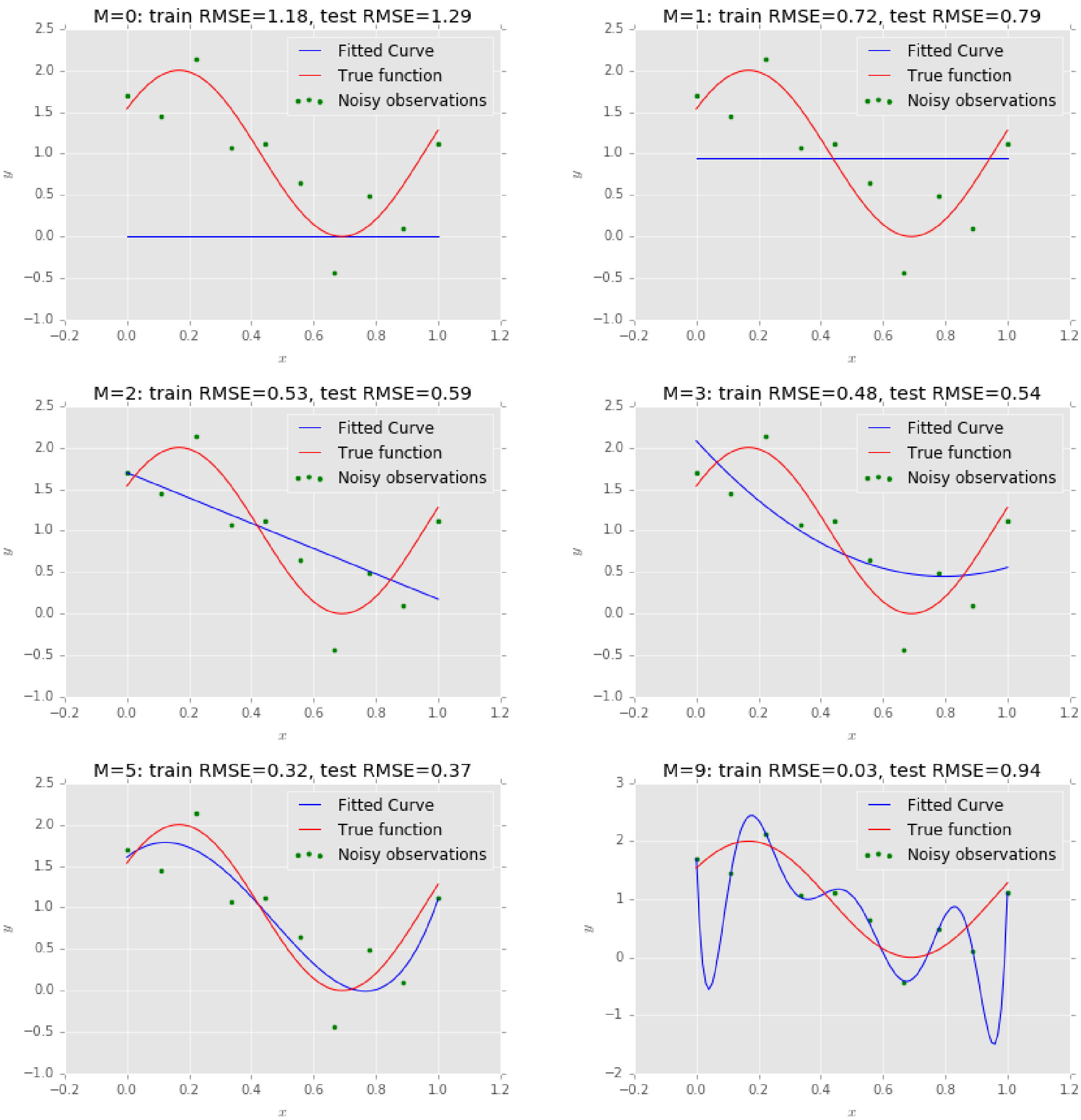

Figure 2: Fitted curves for different values of M

When $M = 2$, the polynomial is linear. From $M = 4$ to $M = 8$, the polynomial fits the underlying sine wave quite well. When $M = 9$, the polynomial is clearly overfitted on the training data. This can also be concluded by comparing the root mean squared errors on the training and test sets.

4

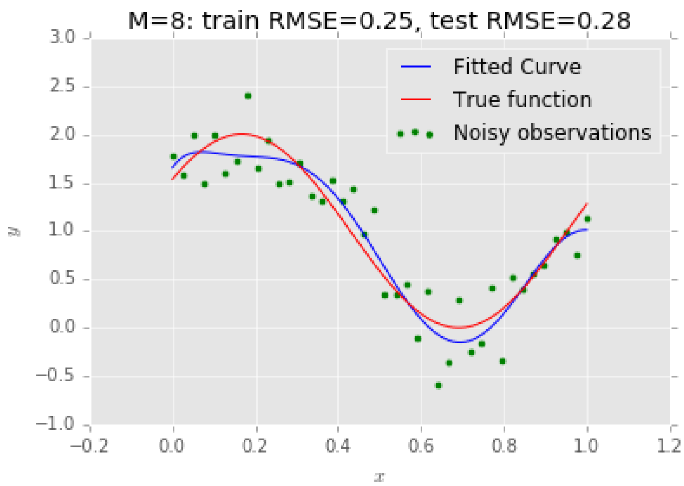

Figure 3: Fitted curve with $M = 8$ and $N = 40$

The fitted curves show less signs of overfitting overall and have lower errors. However for $M \geq 8$ there is still some overfitting to the noise, but since $N > M + 1$, the curve does not have enough degrees of freedom to fit all points exactly.

5

$\mathbf{w}$	$\lambda = 0$	$\lambda = 0.1$
w_0^*	1.37	1.78
w_1^*	61.59	-1.19
w_2^*	-1022.49	-1.29
w_3^*	7096.50	-0.62
w_4^*	-25604.26	-0.05
w_5^*	51562.88	0.34
w_6^*	-58476.55	0.60
w_7^*	34924.10	0.77
w_8^*	-8541.82	0.88

Table 1: Optimal weights for $M = 9$ and $N = 10$ with ($\lambda = 0.1$)and without ($\lambda = 0$) regularization

The weigths with regularization are a lot smaller than the unregularized weights. There is also a smaller difference between the error on the test and the training set when using regularization.

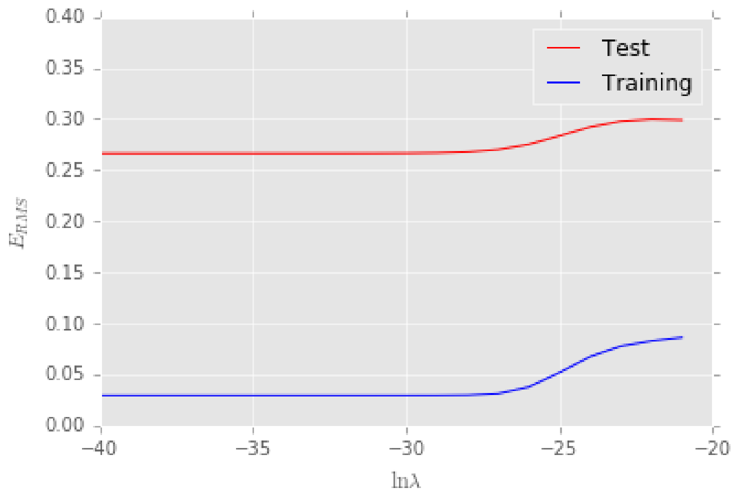

Figure 4: The error on the training and test set versus the regularization strength

The reproduction of Figure 1.8 in Bishop looks slightly different for small values of $\ln \lambda$. We assume that the figure in Bishop is not on scale for $\ln \lambda < 35$ for educational purposes.

Exercise 2

1

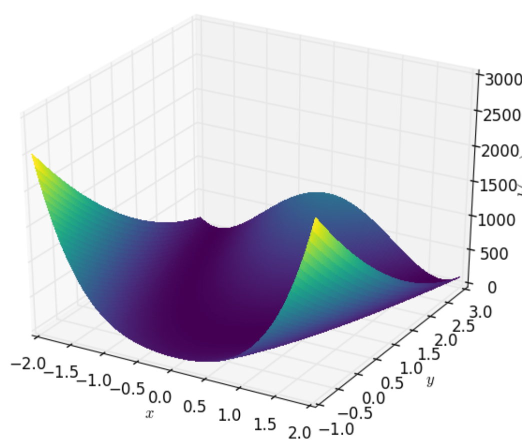

Figure 5: Surface plot of $h(x, y)$

```
def h(x, y):
    return 100 * (y - x**2)**2 + (1 - x)**2
```