Assignment 2

EECS 4404

Anton Sitkovets
212168048

(1) $E(\mu, \Sigma) = \sum_{i=1}^{N} \log p(X|\mu, \Sigma) = * \frac{N}{2} \log|2\pi\Sigma| - \frac{1}{2} \sum_{i}^{N} (X_i - \mu)^T \Sigma^{-1} (X_i - \mu)$

$$\frac{\partial E}{\partial \mu} = -\frac{1}{2} \sum_{i}^{N} -2\,\Sigma^{-1}(X_i - \mu) = 0$$

$$\sum_{i=1}^{N} \Sigma^{-1} X_i = \sum_{i}^{N} \Sigma^{-1} \mu$$

$$\hat{\mu}_{MLE} = \frac{1}{N} \sum_{i}^{N} X_i$$

To find Σ_{MLE} rewrite the log likelihood

$$= \frac{N}{2} \log|2\pi\Sigma^{-1}| - \frac{1}{2} \sum_{i} \mathrm{tr}\left[(X_i - \mu)(X_i - \mu)^T \Sigma^{-1}\right]$$

$$= \frac{N}{2} \log|2\pi\Sigma^{-1}| - \frac{1}{2} \mathrm{trace}\left[S_\mu \Sigma^{-1}\right]$$

where $S_\mu = \sum_{i=1}^{N} (X_i - \mu)(X_i - \mu)^T$

$$\frac{\partial E}{\partial \Sigma^{-1}} = \frac{N}{2} \Sigma^{-T} - \frac{1}{2} S_\mu^T = 0 \qquad * \Sigma^{-T} = \Sigma^{-1} = \Sigma$$

$$\frac{N}{2} \Sigma^{-1} = \frac{1}{2} S_\mu^T$$

$$\Sigma_{MLE}^{-1} = \frac{1}{N} \sum_{i=1}^{N} (X_i - \mu)(X_i - \mu)^T$$

$(X_i - \mu)$: D×1 matrix; $(X_i - \mu)^T$: $(D \times 1 \text{ matrix})^T = 1 \times D$

Hence Σ_{MLE} is a D×D matrix
within the matrix each element i, j is the covariance ~~of~~ between the i^{th} and j^{th} element of the random vector.
Since we have the Naive Bayes assumption, and we know that the variables are independent from each other, the covariance matrix will be a ~~diag~~ diagonal matrix, since non diagonal values in the matrix will have 0 values because they are independent from each other.

(2) $X \in R^D$ [D×1] $\quad \Sigma = \sigma^2 I$

$Y \in R^M$ [M×1]

$W \in R^{M \times D}$ [M×D]

a) $\Sigma \in R^{M \times M}$ [M×M]

$$p(y|x,W) = \mathcal{N}(y|Wx, \Sigma)$$

$$E(w) = \prod_{i=1}^{N} (2\pi)^{-\frac{D}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y^{(i)} - Wx^{(i)})^T \Sigma^{-1} (y^{(i)} - Wx^{(i)})\right)$$

$$= -\log \mathcal{N}(y|Wx, \Sigma) = -\sum_{i=1}^{N} \left[-\frac{1}{2}\log 2\pi|\Sigma| - \frac{1}{2}(y^{(i)} - Wx^{(i)})^T \Sigma^{-1} (y^{(i)} - Wx^{(i)})\right]$$

$$= \sum_{i=1}^{N} \left[\frac{1}{2}\log 2\pi|\Sigma| + \frac{1}{2}(y^{(i)} - Wx^{(i)})^T \Sigma^{-1} (y^{(i)} - Wx^{(i)})\right]$$

$$\star \quad \frac{\partial}{\partial W}\left[(y-Wx)^T \Sigma^{-1} (y-Wx)\right] = -2\Sigma^{-1}(y-Wx)x^T$$

$$\Rightarrow \frac{\partial E(w)}{\partial W} = \sum_{i=1}^{N} -\Sigma^{-1}(y^{(i)} - Wx^{(i)})x^{(i)T} = 0$$

$$\sum_{i=1}^{N} -y^{(i)}x^{(i)T} + Wx^{(i)}x^{(i)T} = 0$$

$$-\sum_{i=1}^{N} y^{(i)}x^{(i)T} + \sum_{i=1}^{N} Wx^{(i)}x^{(i)T} = 0$$

$$W \cdot \underbrace{\sum_{i=1}^{N} x^{(i)}x^{(i)T}}_{XX^T} = \underbrace{\sum_{i=1}^{N} y^{(i)}x^{(i)T}}_{YX^T}$$

Substitute:

$$WXX^T = YX^T$$

$$W = YX^T(XX^T)^{-1}$$

Dimensions: $(M\times 1)(1\times D)(D\times 1)(1\times D)$ → $(D\times D)$ → $(1\times D)$ → $M\times D$

Hence we get W as a M×D matrix and this means that the ~~only~~ rows i of W are only dependent on the corresponding ith row from Y.

b) In the first part (a) we used $\Sigma = \sigma^2 I = \begin{bmatrix} \sigma^2 & & \\ & \sigma^2 & \\ & & \ddots \sigma^2 \end{bmatrix}$ While here we use $\Sigma = \begin{bmatrix} \sigma_1^2 & & \\ & \sigma_2^2 & \\ & & \ddots \sigma_m^2 \end{bmatrix}$. From the derivation above, for W we do not use Σ in the derivation. Hence, W is independent on the value of Σ and the derivation of W will remain the same and continues to show that the estimate of row i of W only uses the values of ith dimension of output vectors y.

$$\mu_N = \frac{1}{N}\sum_{i}^{N} X_i \qquad \Sigma_N = \frac{1}{N}\sum_{i}^{N}(X_i-\mu_N)(X_i-\mu_N)^T$$

(3) a) $\mu_N = \mu_{N-1} + \frac{1}{N}(X_N - \mu_{N-1})$

$$= \frac{1}{N-1}\sum_{i=1}^{N-1} X_i + \frac{1}{N}\left(X_N - \frac{1}{N-1}\sum_{i}^{N-1} X_i\right)$$

$$= \frac{1}{N-1}\sum_{i=1}^{N-1} X_i + \frac{X_N}{N} - \frac{1}{N(N-1)}\sum_{i}^{N-1} X_i$$

$$= \frac{N}{N(N-1)}\sum_{i=1}^{N-1} X_i + \frac{(N-1)X_N}{N(N-1)} - \frac{\sum_{i}^{N-1} X_i}{N(N-1)} = \frac{N}{N(N-1)}\sum_{i=1}^{N-1} X_i - \frac{\sum_{i}^{N-1} X_i}{N(N-1)} + \frac{(N-1)X_N}{N(N-1)}$$

$$= \frac{(N-1)\sum_{i}^{N-1} X_i}{N(N-1)} + \frac{(N-1)}{N(N-1)}X_N = \frac{(N-1)\left[\sum_{i}^{N-1} X_i + X_N\right]}{N(N-1)} = \frac{1}{N}\sum_{i}^{N} X_i$$

b)

~~$\Sigma_{N-1} = E(X_{N-1}X_{N-1}^T) - \frac{N^2}{(N-1)^2}\left[E(X_N X_N^T) - \Sigma_N\right]$~~

~~$+ \frac{N}{(N-1)^2}X_N\mu_N^T + \frac{N}{(N-1)^2}\mu_N X$~~

~~$\mu_N\mu_N^T = E(X_N X_N^T) - \Sigma_N$~~

$\lambda_N = E(XX^T) = \alpha\sum_{i=1}^{N} X_i X_i^T$ where $\alpha = \frac{1}{N}$

$= \alpha\sum_{i=1}^{N-1} X_i X_i^T + \alpha X_N X_N^T$

$= \alpha(N-1)\lambda_{N-1} + \alpha X_N X_N^T = \boxed{(1-\alpha)\lambda_{N-1} + \alpha X_N X_N^T}$

$\Sigma_N = \lambda_N - \mu_N\mu_N^T = \alpha(N-1)\lambda_{N-1} + \alpha X_N X_N^T - \mu_N\mu_N^T$

From last a use recursive defn of μ_N

$\mu_N = \mu_{N-1} + \alpha(X_N - \mu_{N-1}) = \mu_{N-1} + \alpha X_N - \alpha\mu_{N-1}$

$= \boxed{(1-\alpha)\mu_{N-1} + \alpha X_N}$

$= \lambda_N - \mu_N\mu_N^T = \lambda_N - \left[(1-\alpha)\mu_{N-1} + \alpha X_N\right]\left[(1-\alpha)\mu_{N-1} + \alpha X_N\right]^T$

$= \lambda_N - \left[(1-\alpha)^2\mu_{N-1}\mu_{N-1}^T + \alpha(1-\alpha)\left[\mu_{N-1}X_N^T + X_N\mu_{N-1}^T\right] + \alpha^2 X_N X_N^T\right]$

$= \lambda_N - (1-\alpha)^2\mu_{N-1}\mu_{N-1}^T - \alpha(1-\alpha)\left[\mu_{N-1}X_N^T + X_N\mu_{N-1}^T\right] - \alpha^2 X_N X_N^T$

$= (1-\alpha)\lambda_{N-1} + \alpha X_N X_N^T - (1-\alpha)^2\mu_{N-1}\mu_{N-1}^T - \alpha(1-\alpha)\left[\mu_{N-1}X_N^T + X_N\mu_{N-1}^T\right] - \alpha^2 X_N X_N^T$

$= (1-\alpha)\left[\lambda_{N-1} - \mu_{N-1}\mu_{N-1}^T + \alpha\left[X_N X_N^T - \mu_{N-1}X_N^T - X_N\mu_{N-1}^T + \mu_{N-1}\mu_{N-1}^T\right]\right]$

($\lambda_{N-1} - \mu_{N-1}\mu_{N-1}^T = \Sigma_{N-1}$; $X_N X_N^T - \mu_{N-1}X_N^T - X_N\mu_{N-1}^T + \mu_{N-1}\mu_{N-1}^T = (X_N-\mu_{N-1})(X_N-\mu_{N-1})^T$; $\alpha = 1/N$)

Therefore!

$$\Sigma_N = (1-\alpha)\left[\Sigma_{N-1} + 2(X_N-\mu_{N-1})(X_N-\mu_{N-1})^T\right]$$

(4) a) $x \in \mathbb{R}^N$ $p(x) = \mathcal{N}(x \mid \mu_x, \Sigma_x)$

$n \in \mathbb{R}^M$ $p(n) = \mathcal{N}(n \mid 0, \Sigma_n)$

$y = Ax + b + n$ $A \in \mathbb{R}^{M \times N}$ & $b \in \mathbb{R}^M$ are constants

$$\mu_y = E[y] = E[Ax + b + n] = AE[x] + AE[b] + E[n]$$
$$= A\mu_x + b + 0 = A\mu_x + b$$

$$\Sigma_y = E[(y - \mu_y)(y - \mu_y)^T] = E[(Ax + b + n)(Ax + b + n)^T]$$

$$= E\left[\left([A\ I_m]\begin{bmatrix} x \\ n \end{bmatrix} + b - [A\ I_m]\begin{bmatrix} \mu_x \\ \mu_n \end{bmatrix} - b\right)\left([A\ I_m]\begin{bmatrix} x \\ n \end{bmatrix} + b - [A\ I_m]\begin{bmatrix} \mu_x \\ \mu_n \end{bmatrix} - b\right)^T\right]$$

$$= \left[[A\ I_m]\left(\begin{bmatrix} x - \mu_x \\ n - \mu_n \end{bmatrix}\right)\left(\begin{bmatrix} x - \mu_x \\ n - \mu_n \end{bmatrix}\right)^T \begin{bmatrix} A^T \\ I_m \end{bmatrix}\right]$$

$$= [A\ I_m] E\left[\begin{bmatrix} (x-\mu_x)(x-\mu_x)^T & (x-\mu_x)(n-\mu_n)^T \\ (n-\mu_n)(x-\mu_x)^T & (n-\mu_n)(n-\mu_n)^T \end{bmatrix}\right]\begin{bmatrix} A^T \\ I_m \end{bmatrix}$$

$$= [A\ I_m]\begin{bmatrix} \Sigma_x & \Sigma_{xn} \\ \Sigma_{xn}^T & \Sigma_n \end{bmatrix}\begin{bmatrix} A^T \\ I_m \end{bmatrix} = [A\ I_m]\begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_n \end{bmatrix}\begin{bmatrix} A^T \\ I_m \end{bmatrix}$$

$$= [A\Sigma_x + \Sigma_n]\begin{bmatrix} A^T \\ I_m \end{bmatrix} = A\Sigma_x A^T + \Sigma_n$$

* $\Sigma_{xn} = 0$ because x and n are independent

b) The mean is the same; $\mu_y = A\mu_x + b$

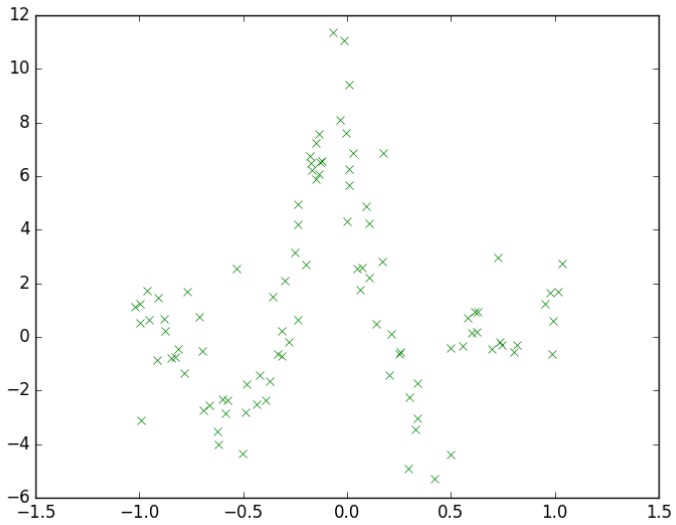
Since $\Sigma_{xn} \neq 0$, because x & n are not independent:

$$\Sigma_y = [A\ I_m]\begin{bmatrix} \Sigma_x & \Sigma_{xn} \\ \Sigma_{xn}^T & \Sigma_n \end{bmatrix}\begin{bmatrix} A^T \\ I_m \end{bmatrix} = [A\Sigma_x + \Sigma_{xn}^T + A\Sigma_{xn} + \Sigma_n]\begin{bmatrix} A^T \\ I_m \end{bmatrix}$$

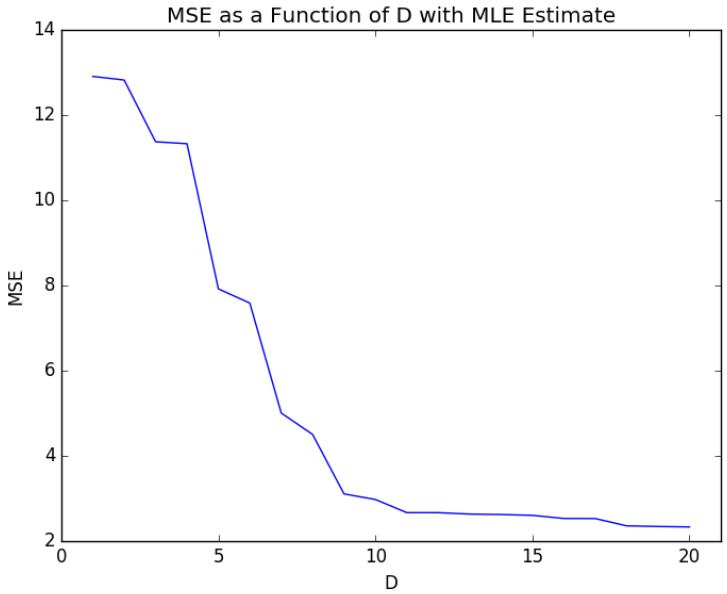
$$= A\Sigma_x A^T + \Sigma_{xn}^T A^T + A\Sigma_{xn} + \Sigma_n$$

Assignment 2 Report
Anton Sitkovets

Step 1



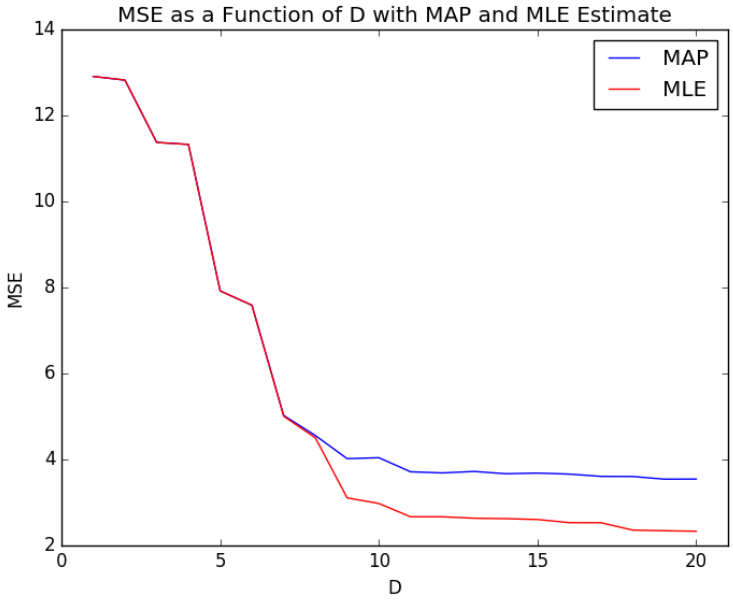
Step 2



Based on the values we can see that when D = 20 gives the best result as it has the smallest error when comparing the predicted values to the actual values. Although the data shows that this is the best value, I would say that using a polynomial of (20 – 1) = 19, to be very complicated and could potentially cause overfitting. As discussed in class, a good way to avoid overfitting is to use simpler basis functions, so I think the best bet would be to use D = 11 to 16

because although this doesn't give the best MSE it is very close to the best, while still remaining a simple model.

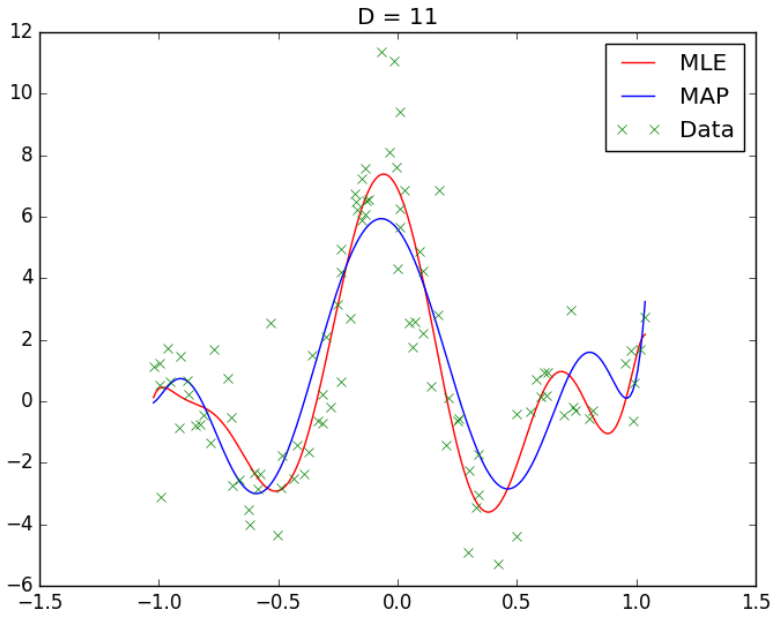
Step 3



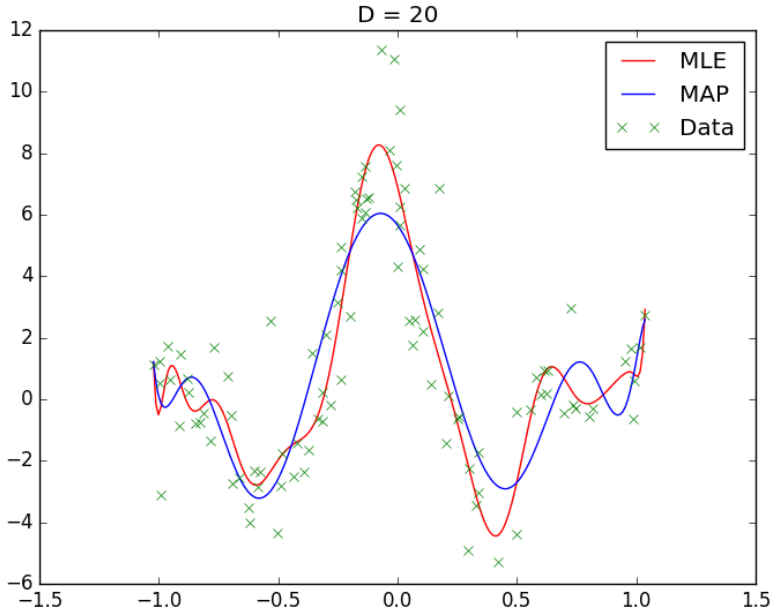
The MSE for the MAP estimate stops decreasing after a point while the ML case continues to decrease because the purpose of the MAP estimate is to add a prior to encourage smooth models and penalize more complex models. So, as D increases, the prior begins to have more and more effect on the quantities that one wants to estimate. Meaning that as we increase D, the resulting estimates begin to rely more on the prior rather than the actual distribution of data, making its predictions less accurate and its mean squared error large. Hence as can be seen from the result, the MAP estimate discourages models with high complexity as unlike the ML estimate, the mean squared error doesn't continue to decrease as rapidly after a certain value of D.

Step 4

Since the MAP estimate function is used to penalize complex models, using the results from step 3 I would say that D = 11 would give the best result for the MAP estimate. After looking at the results I can see this is the case, as when D increases after 11, there is only minor improvements. But we need to consider whether these minor improvements are worth the added complexity. Also, we can assume that real world phenomena are mostly smooth, so we can say that adding a prior for the MAP estimate adds those smoothness assumptions. Since the data should be smooth, the plot should not have all these short curves as you can see in higher order MLE functions. Below we can see the MAP estimate fits the data well.

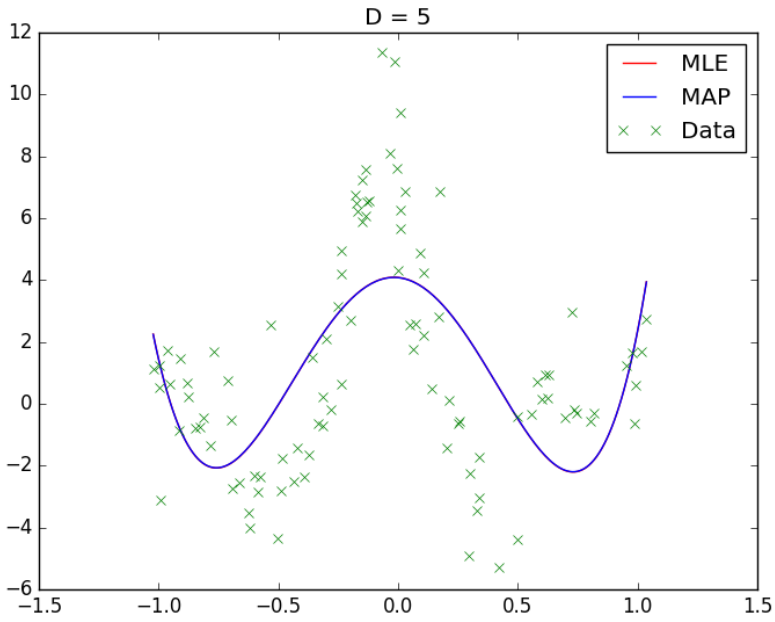


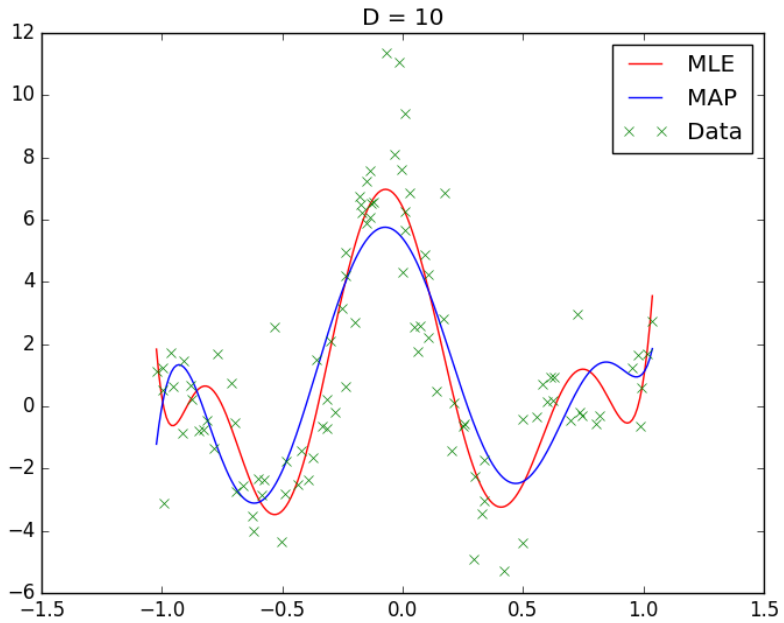
From Step 2, I found that when D = 20, the MSE is the lowest for the MLE and looking at the plot we can see that the results are quite good for the dataset. From the plot below we can see that the plot is a good fit for the MLE as well as the MAP estimate. But although this seems like the best fit, as mentioned in step 2, to prevent over fitting the data we should use a less complex model as real world data is generally smooth and doesn't need this complexity of curves as seen below for the MLE.

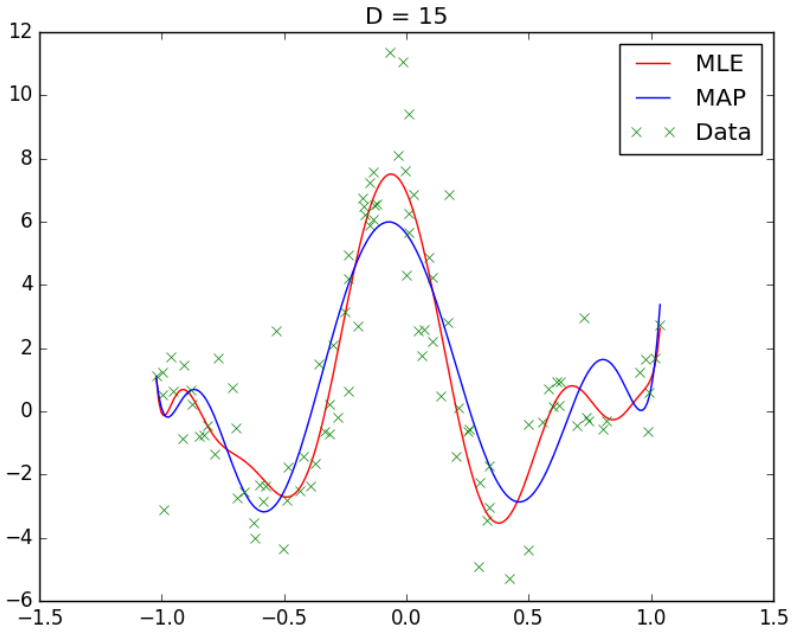


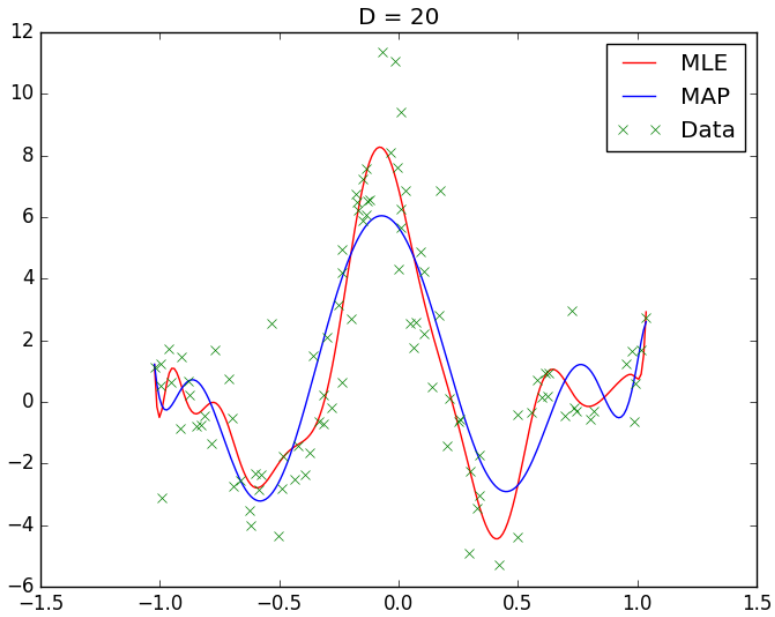
The differences between the MAP and ML estimated functions as D increases can be seen clearly from the results of step 3. As D increases, the MAP and ML estimates tend to decrease in the mean squared error, but the MAP will cut off at a certain point and will stop decreasing. This can be attributed to the prior knowledge added to the model with the MAP, as we are trying to enforce the use of smoother and simpler models to simulate the phenomena that real-world situations are typically smooth. This is all done to prevent overfitting the data and making sure that we are not using a very complex model that fits our training data perfectly, but poorly with test data. Hence, MAP estimates should be used over ML estimates to prevent the use of complex models and overfitting.

Below are plots for D e {5, 10, 15, 20}:

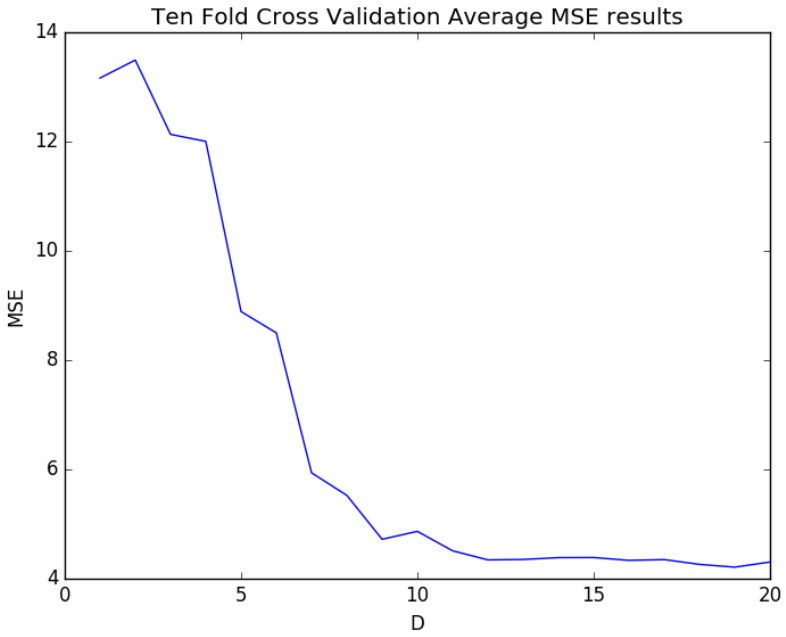



D = 10
MLE
MAP
Data
12
10
8
6
4
2
0
−2
−4
−6
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

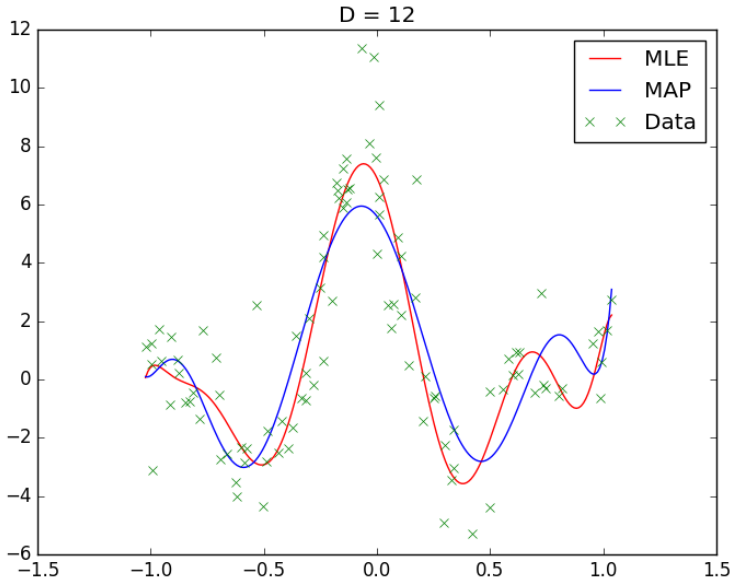

D = 15
MLE
MAP
Data
12
10
8
6
4
2
0
−2
−4
−6
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5



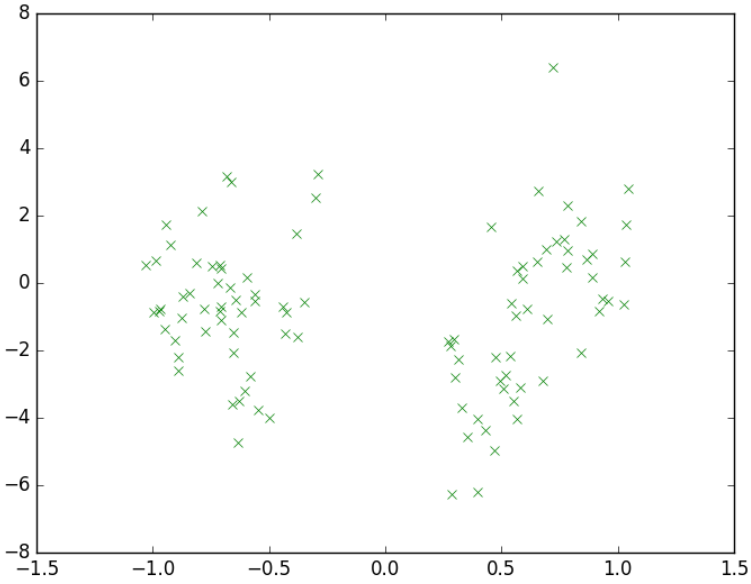
Step 5



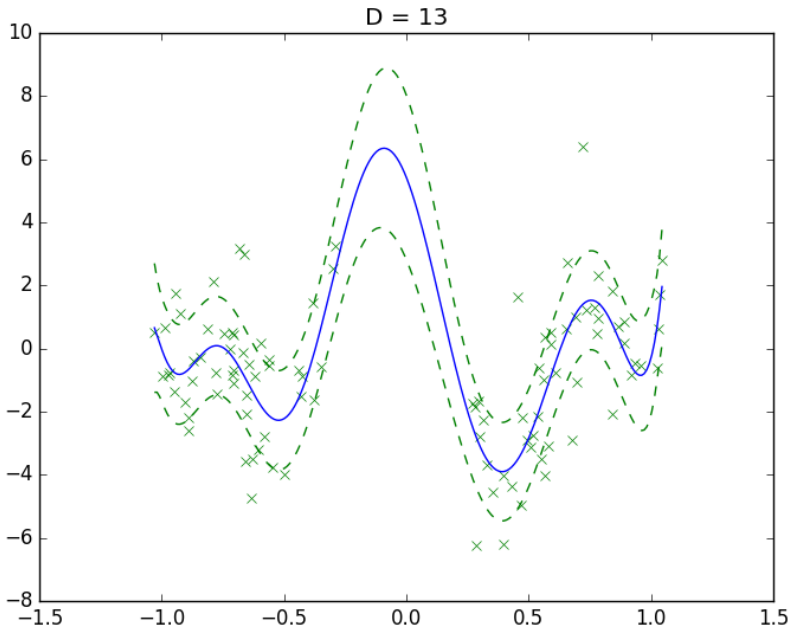
Based on the curve I would pick D of about 11, 12 or 13. This was an obvious choice as it aligns perfectly to what I have been saying previously about MAP estimates enforce simpler models. So, although 20 might be the lowest error, the difference in MSE between using a D of 12 vs 20 is insignificant, whereas the risk of overfitting is much greater with the complex model of D = 20. Although it is not obvious which choice is the best, since values 12 to 20 are around the same value, it is the safest route to just choose D= 12 because of the danger of overfitting.

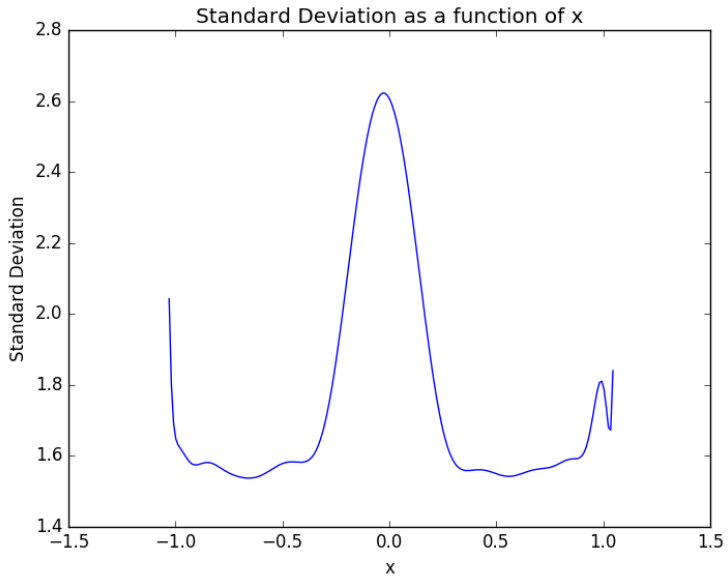

D = 12
MLE
MAP
Data
12
10
8
6
4
2
0
−2
−4
−6
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Step 6


8
6
4
2
0
−2
−4
−6
−8
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Step 7





The second graph here shows the standard deviation as a function of x. The range here is from [min(x), max(x)], where x is the values from the dataset2 input file. This specific range was taken to show a closer view of the effects of standard deviation with respect to x, whereas the graph below depicts the specified range of [-1.1, 1.1]. Looking at the second plot in step 7, the standard deviation correlates to the amount of points present at that value of x in the given dataset. Comparing the first and second plots we can see that there are no data points between the range [-0.25, 0.25], which leads to a high uncertainty and high standard deviation. Whereas, in areas from [-1.4, -0.25] and [0.25, 1.4], the uncertainty and standard deviation of the model is low since we have lots of data points to fit the model to. The points -1.1 and 1.1 have very high levels of uncertainty as can be seen in the third plot because we do not have any values in

our dataset at these values of x, so the model is not sure whether the new points will fit the current model.

When the standard deviation is at a minimum this means that these values of x have many points that align along the model, so the certainty of the prediction and its location on the y axis is quite high, so the standard deviation is low. When we have a minimum value for the standard deviation, this means this is a point where there are a lot of dataset values along this value of x. Looking at the second graph we can see the minimum value for standard deviation is around -0.6, and looking at the first graph we see that this point has a lot of data, so this means areas with high data concentration mean low uncertainty.

If the y values were badly corrupted but the x values were unchanged it would not lead to a change in standard deviation but only a change in $mu_w(x)$. This makes sense as the standard deviation only cares about the frequency of data along a value of x, not the location of the points on y, whereas the mean value depends on y. So, the result would lead to the mean curve be shifted and possibly a different shape, but the standard deviation values would remain the same and just add uncertainty to the $mu_w(x)$ curve. So, if we look at the standard deviation function on its own, the plot would be the same as previously found. This result is reasonable since we are assuming our predicted outputs y are independent of the data given weights, w, and w is independent of new inputs being validated.