HELLENIC REPUBLIC
National and Kapodistrian
University of Athens

Postgraduate Program: «Data Science and Information Technologies»

Machine Learning in Computational Biology

Assignment #1

Postgraduate Student: Gavrielatos Marios
Registration Number: 7115152100023

Problem 1

Assume we have N random vectors $D = \{x_1, x_2, \dots, x_N\}$ that follow the multidimensional normal distribution: $x_{(i)} \sim N(x|\mu, \Sigma)$ where μ is a n×1 vector and Σ is a n×n symmetric matrix. The density function of the multivariate gaussian distribution:

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} \cdot |\Sigma|^{\frac{1}{2}}} \cdot \exp\left\{-\frac{1}{2}(x_n - \mu)^T \Sigma^{-1}(x_n - \mu)\right\}$$

The logarithm of the aforementioned density function gives us the log likelihood function:

$$A = ln(x|\mu, \Sigma) = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2} ln|\Sigma| - \frac{1}{2}\sum_{n=1}^{N}(x_n - \mu)^T \Sigma^{-1}(x_n - \mu)$$

will use the following matrix identity: $\frac{\partial w^T \Sigma w}{\partial w} = 2\Sigma w$,where w does not depend on Σ and Σ is symmetric.

Taking the derivative of A with respect to μ and equating it to zero:

$$\frac{\partial A}{\partial \mu} = 0 \Rightarrow -\frac{1}{2}\sum_{n=1}^{N} 2\Sigma^{-1}(x_n - \mu) = 0 \Rightarrow \sum_{n=1}^{N} \Sigma^{-1}(x_n - \mu) = 0 \Rightarrow N\mu - \sum_{n=1}^{N} x_n = 0 \Rightarrow \boldsymbol{\mu = \frac{1}{N}\sum_{n=1}^{N} x_n}$$

Problem 2

N is the number of trials and m the number of "positives" in N trials. The probability of getting m positive trials in N trial is given by the following probability mass function:

$$Bin(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m},\ m = 0,1,2,\dots,N \text{ and } \binom{N}{m} = \frac{N!}{m!(N-m)!}$$

i. Mean:

We have:

$$E[m] = \sum_{m=0}^{N} mBin(m|N,\mu) = \sum_{m=0}^{N} m\binom{N}{m}\mu^m(1-\mu)^{N-m} = \sum_{m=1}^{N} m\binom{N}{m}\mu^m(1-\mu)^{N-m}$$

, since the term with $m = 0$ will be equal to 0 and therefore we can skip it.

Moreover:

$$m\binom{N}{m} = m\frac{N!}{m!\,(N-m)!} = \frac{N\cdot(N-1)!}{(m-1)!\,(N-m)!} = N\frac{(N-1)!}{(m-1)!\,[(N-1)-(m-1)]!} = N\binom{N-1}{m-1}$$

By combining (1) and (2):

$$E[m] = \sum_{m=1}^{N} N\binom{N-1}{m-1}\mu^m(1-\mu)^{N-m} = N\mu\sum_{m=1}^{N}\binom{N-1}{m-1}\mu^{m-1}(1-\mu)^{N-m}$$

From the Binomial theorem: $\big(x+(1-x)\big)^n = \sum_{k=0}^{n}\binom{n}{k}x^k(1-x)^{n-k}$

we can derive that: $\sum_{k=1}^{n}\binom{n-1}{k-1}x^{k-1}(1-x)^{n-k} = (x+(1-x))^{n-1}$

By combining (3) and (4):

$$\boldsymbol{E[m]} = N\mu[\mu+(1-\mu)]^m \Rightarrow E[m] = \boldsymbol{N\mu}$$

ii. Variance

We know:

$$var[m] = \sum_{m=0}^{N}(m-E[m])^2Bin(m|N,\mu) = E[m^2] - \big[E[m]\big]^2$$

$E[m]$ is known. We need to calculate $E[m^2]$:

$$E[m^2] = E[m^2 - m + m] = E[m(m-1)+m] = E[m(m-1)] + N\mu$$

Calculate $E[m(m-1)]$:

$$E[m(m-1)] = \sum_{m=0}^{N} m(m-1)\binom{N}{m}\mu^m(1-\mu)^{N-m} = \sum_{m=1}^{N} m(m-1)\binom{N}{m}\mu^m(1-\mu)^{N-m} \;\; E[m(m-1)]$$

$$= \sum_{m=1}^{N} m(m-1)\frac{N!}{m!\,(N-m)!}\mu^m(1-\mu)^{N-m}$$

$$= \sum_{m=1}^{N} \frac{N!}{(m-2)!\,(N-m)!}\mu^m(1-\mu)^{N-m}$$

$$= N(N-1)\mu^2 \sum_{m=1}^{N} \frac{(N-2)!}{(m-2)!\,(N-m)!}\,\mu^{m-2}(1-\mu)^{N-m} = N(N-1)\mu^2(\mu+(1+\mu))^{N-2}$$

$$= N(N-1)\mu^2$$

By combining (1), (2) and (3):

$$\boldsymbol{var}[\boldsymbol{m}] = E[m^2] - \left[E[m]\right]^2 = E[m(m-1)] + N\mu - N^2\mu^2 = N(N-1)\mu^2 + N\mu - N^2\mu^2$$

$$= N^2\mu^2 - N\mu^2 + N\mu - N^2\mu^2 = \boldsymbol{N\mu(1-\mu)}$$

Problem 3

x is a random variable following the Gaussian Distribution $N(\mu, \sigma^2)$ with a known variance σ^2 but unknown mean μ which follows a prior distribution $N(\mu_0, \sigma_0^2)$, as in the Bayesian framework. Given a dataset of N independent observations $X = \{x_1, x_2, \dots, x_N\}$:

3.1

Show that the posterior distribution of the mean $p(\mu|X)$ is also a Gaussian with mean $\mu_N = \frac{N\sigma_0^2\bar{x}+\sigma^2\mu_0}{N\sigma_0^2+\sigma^2}$ where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, and variance $\sigma_N^2 = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2+\sigma^2}$:

We want to maximize: $p(\mu|\boldsymbol{X}) = p(\boldsymbol{X}|\mu) \cdot p(\mu)$

$$p(\mu|\boldsymbol{X}) = \frac{1}{(2\pi\sigma^2)^{N/2}} exp\left[-\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}\right] \cdot \frac{1}{(2\pi\sigma_0^2)^{\frac{1}{2}}} exp\left[-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right]$$

$$= \frac{1}{2\pi\sqrt{\sigma^N\sigma_0}} exp\left[-\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2}\right] \cdot exp\left[-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right]$$

$$= \frac{1}{2\pi\sqrt{\sigma^N\sigma_0}} exp\left[-\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}\right]$$

We set:

$$I = -\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{2\sigma^2} - \frac{(\mu-\mu_0)^2}{2\sigma_0^2}$$

$$= -\sum_{i=1}^{N}\frac{x_i^2-2\mu x_i+\mu^2}{2\sigma^2} - \frac{-\mu^2+2\mu\mu_0-\mu_0^2}{2\sigma_0^2}$$

$$= \frac{-\mu^2\sigma^2+2\mu\mu_0\sigma^2-\mu_0^2\sigma^2-\sum_{i=1}^{N}\left[x_i^2\sigma_0^2-2\mu x_i\sigma_0^2+\mu^2\sigma_0^2\right]}{2\sigma^2\sigma_0^2}$$

$$= \frac{-\mu^2\sigma^2+2\mu\mu_0\sigma^2-\mu_0^2\sigma^2-\sigma_0^2\sum_{i=1}^{N}x_i^2+2\mu\sigma_0^2\sum_{i=1}^{N}x_i-N\mu^2\sigma_0^2}{2\sigma^2\sigma_0^2}$$

$$= \frac{-\mu^2(\sigma^2+N\sigma_0^2)+2\mu(\mu_0\sigma^2+\sigma_0^2\sum_{i=1}^{N}x_i)-(\mu_0^2\sigma^2+\sigma_0^2\sum_{i=1}^{N}x_i^2)}{2\sigma^2\sigma_0^2}$$

We divide I with $(\sigma^2+N\sigma_0^2)$:

$$I = \frac{-\mu^2+\frac{2\mu(\mu_0\sigma^2+\sigma_0^2\sum_{i=1}^{N}x_i)}{(\sigma^2+N\sigma_0^2)} - \frac{(\mu_0^2\sigma^2+\sigma_0^2\sum_{i=1}^{N}x_i^2)}{(\sigma^2+N\sigma_0^2)}}{\frac{2\sigma^2\sigma_0^2}{(\sigma^2+N\sigma_0^2)}}$$

It is important to note that the following part of the equation: $\frac{(\mu_0^2\sigma^2+\sigma_0^2\sum_{i=1}^{N}x_i^2)}{(\sigma^2+N\sigma_0^2)}$ is a constant. We want to bring I in the following form: $-\frac{(\Theta-K)^2}{2\Lambda^2}$. This will show me that the a-posteriori probability has a Gaussian form and we will be able to derive the mean (K) and the variance (Λ^2) of this probability. In order to do that we will perform "completing the square" technique by adding and subtracting the following: $\frac{(\mu_0\sigma^2+\sigma_0^2\sum_{i=1}^{N}x_i)^2}{(\sigma^2+N\sigma_0^2)^2}$ to the numerator of I in order to complete the square. Finally, I will have the following form:

$$I = \frac{-\left[\mu - \frac{(\mu_0\sigma^2 + \sigma_0^2 \sum_{i=1}^{N} x_i)}{(\sigma^2 + N\sigma_0^2)}\right]^2}{2\frac{\sigma^2\sigma_0^2}{(\sigma^2 + N\sigma_0^2)}}$$

From this equation we can derive that $\boldsymbol{\mu_N} = \frac{\boldsymbol{N\sigma_0^2\bar{x}+\sigma^2\mu_0}}{\boldsymbol{N\sigma_0^2+\sigma^2}}$ and $\boldsymbol{\sigma_N^2} = \frac{\boldsymbol{\sigma^2\sigma_0^2}}{\boldsymbol{N\sigma_0^2+\sigma^2}}$.

3.2

Python Notebook: Problem_3_2.ipynb

x follows the distribution $x \sim N(\mu, 16)$, and as a Bayesian, we assume a prior for the mean $\mu \sim N(0, 4)$. Finally, we use the distribution $N(7, 16)$ to generate N observations for x.

(a) Estimate the posterior distribution's $N(\mu|X)$ mean and variance for N = 1, 5, 10, 20, 50, 100, 1000. (b) Provide a diagram that shows the prior distribution, the distribution generating the data, and the estimated posterior distribution for every value of N.

Above, we showed that the posterior distribution follows a Gaussian distribution with mean $\mu_N = \frac{N\sigma_0^2\bar{x}+\sigma^2\mu_0}{N\sigma_0^2+\sigma^2}$ where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$, and variance $\sigma_N^2 = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2+\sigma^2}$. Moreover, from the problem's description we have: $\sigma^2 = 16$, $\mu_0 = 0$ and $\sigma_0^2 = 4$.

The following table and plot summarize the results:

Table 1: *Values for the mean and variance for different values of N*

N	Mean	Variance
1	2.020	3.200
5	4.109	1.778
10	5.811	1.143
20	5.133	0.667
50	6.130	0.296
100	6.464	0.154
1000	6.840	0.016

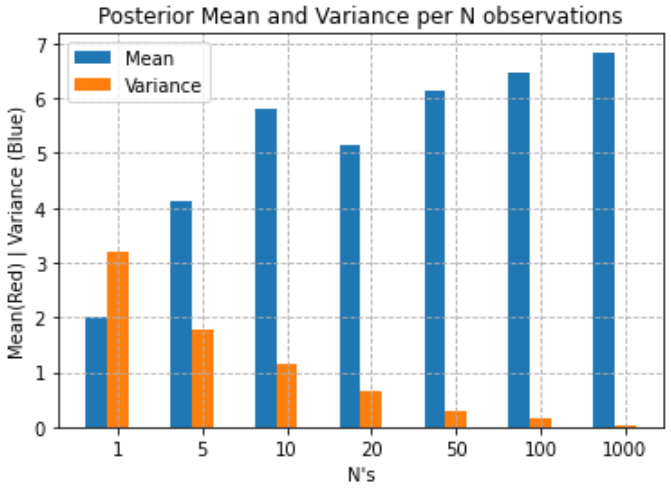

Figure 1: *Posterior Mean and Variance per N observations*

As the number of observations increases, we observe that the Mean value approaches the mean value of the distribution we used to generate the observations (true mean). At the same time, the variance decreases and goes to 0. Therefore, the precision increases and the uncertainty decreases.

The following diagrams shows the prior distribution, the distribution generating the data and the estimated posterior distribution for every value of N:

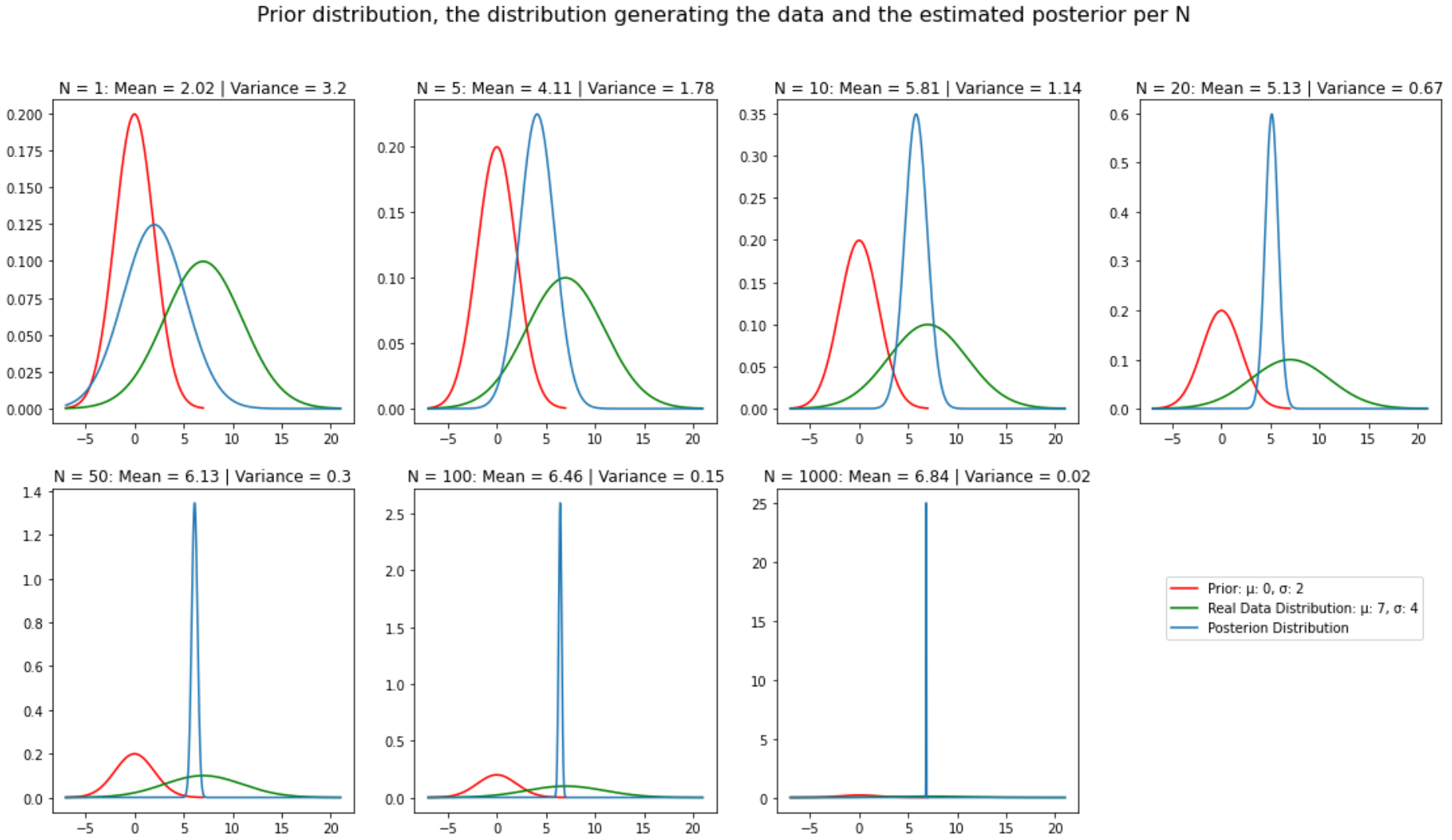

Figure 2: *Prior distribution (red line), the distribution generating the data (green line) and the estimated posterior distribution (blue line) per N observations. We can observe that as N increases the mean of the posterior distribution approaches the mean of the real distribution while the variance goes to zero.*

Problem 4

Python Notebook: Problem_4.ipynb

In this problem we will use the sinusoid function $y(x) = \sin(2\pi x)$ to produce N samples for x uniformly distributed in the interval $[0,1]$. A period of the sinusoid function is the following:

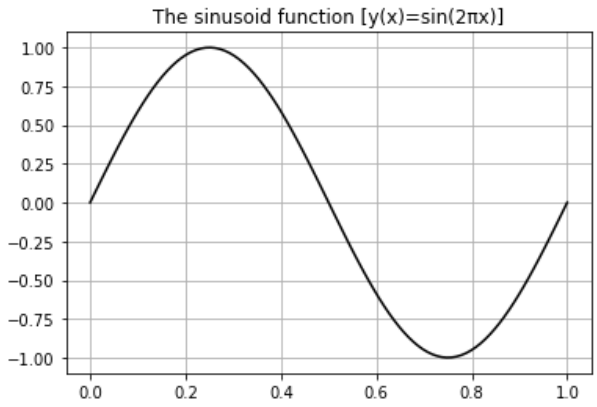

Figure 3: *A period of the sinusoid function*

After producing the N points, we added white Gaussian noise distributed as $N(0,1)$ in order to create a set of noisy observations. Afterwards, using these noisy observations, we fitted polynomial models of various degrees (M = 2, 3, 4, 5, 9) using the least-squares method and calculated the RMSE statistic. We repeated this process for $N = 10$ and $N = 100$. In Figure 4 we can observe the true model curve (black line), the observations drawn with added noise (red dots) and the estimates y's (blue asterisks) with the estimated curve (green line). In Table 2 we present the coefficients of the best least-squares fit model and the achieved RMSE, for a given number of observations and polynomial degree.

The first column of the figure represents experiments in which we used a low degree of polynomial (M = 2), thus our model has lower complexity that the true model. For this reason, these models present a higher RMSE value that the rest of the models with higher polynomial degrees. These models are underfitting (high bias). To fix the **underfitting** problem we must increase the variance and as a result the bias will decrease (bias-variance dilemma). This can be done by *increasing the number of parameters* in the model, the complexity of the model.

As we increase the polynomial degree, the complexity of the model increases. This way the model learns the characteristics of the dataset too well and for this reason it cannot generalize well. This is called overfitting (high variance). To fix the **overfitting** problem we must decrease the variance and introduce bias in our model. This can be done by *reducing the complexity of the model* by introducing bias (i.e., Ridge Regression) or by decreasing the number of parameters.

From Table 2 and Figure 4 we can also infer that by increasing the number of points in the dataset, we are able to reduce overfitting. From the RMSE values we can see that the model with N=10 and M = 9 is highly overfitting. However, the model with N=100 and M = 9 is overfitting in a much lower degree. If the noise variance was smaller the overfitting problem would be reduced in a much higher degree. Therefore, **another way to reduce overfitting is by increasing the size of the dataset**. We must note that we use a model that is different to the true sinusoid model. Finally, we observe **large size coefficients as M increases**. This also indicates overfitting.

In conclusion, the models with 3 ≤ M ≤ 5 seem to give the best fit to the sinusoid function. For M=9 we get an excellent fit to the training data as the polynomial curve passes from every data point and thus gives us a bad representation of the sinusoid and for this reason it cannot generalize well. By increasing the number of observations, the effects of the overfitting decrease.

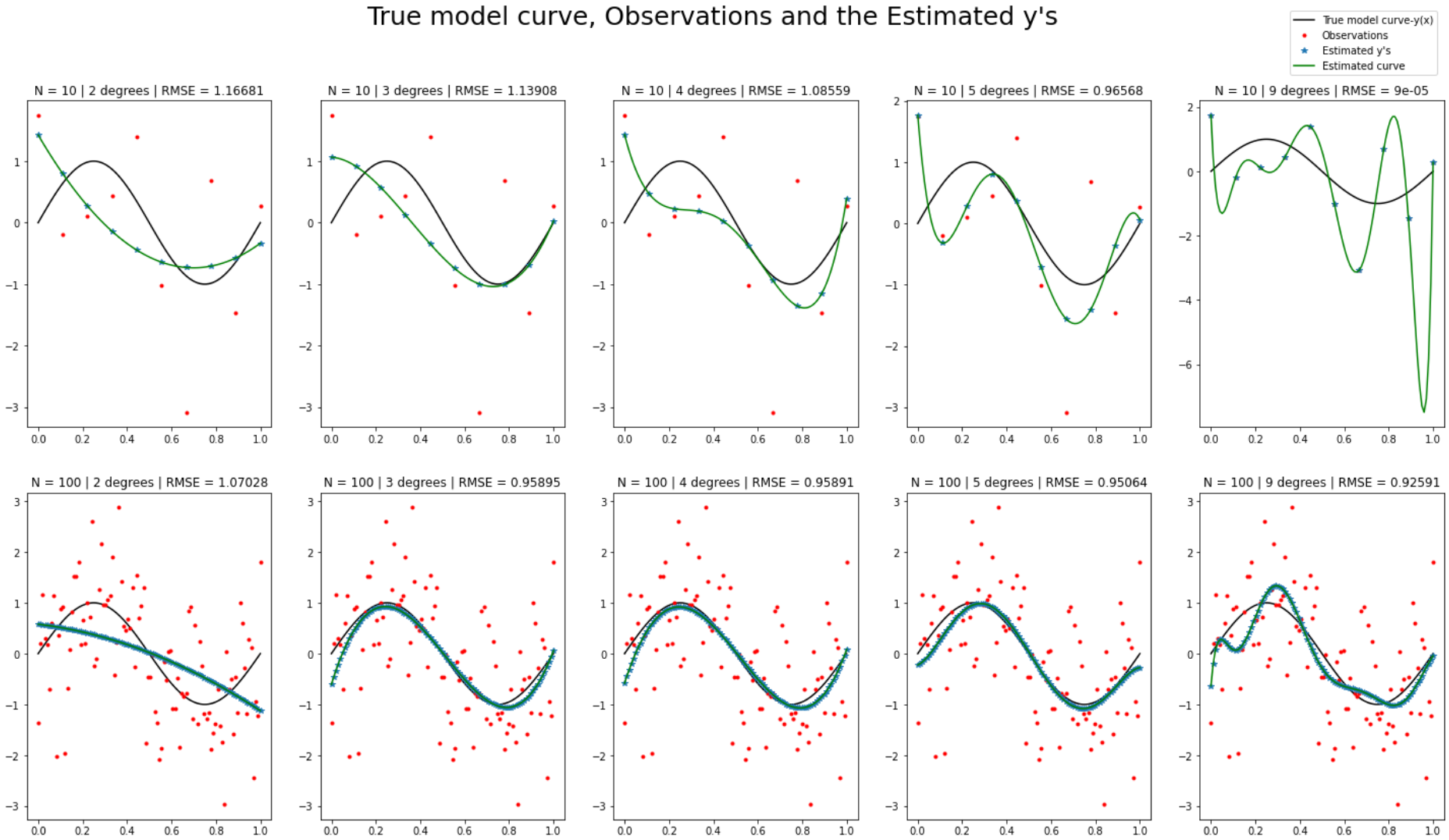

Figure 4: *The True model curve (black curve) plotted against the observations (red points) and the estimated points (blue asterisks) for 5 different polynomial degrees and 2 different number of observations*

Table 2: *Coefficients of the best least-squares fit model and the corresponding RMSE value*

		Coefficients					
Number of observations (N)	**Polynomial degree (M)**	**θ_0**	**θ_1**	**θ_2**	**θ_3**	**θ_4**	**θ_5**
10	**2**	1.426	-6.177	4.416			
10	**3**	1.063	-0.207	-11.316	10.488		
10	**4**	1.431	-13.976	58.214	-101.035	55.761	
10	**5**	1.768	-44.670	314.072	-824.262	884.744	-331.593
10	**9**	1.748	-158.111	2798.152	-20648.413	76482.698	-147204.489
100	**2**	0.579	-0.529	-1.174			
100	**3**	-0.606	14.050	-37.805	24.421		
100	**4**	-0.581	13.536	-35.463	20.766	1.827	
100	**5**	-0.222	1.964	47.169	-201.222	252.177	-100.140
100	**9**	-0.634	53.275	-1086.117	9556.856	-42130.075	103914.282

		Coefficients					
Number of observations (N)	**Polynomial degree (M)**	**θ_6**	**θ_7**	**θ_8**	**θ_9**	**RMSE**
10	2					1.16681
10	3					1.13908
10	4					1.08559
10	5					0.96568
10	9	132007.347	-17198.245	-50120.582	24040.169	0.00009
100	2					1.07028
100	3					0.95895
100	4					0.95891
100	5					0.95064
100	9	-150788.944	128094.032	-58994.200	11381.481	0.92591

Problem 5

Python Notebook: Problem_5.ipynb

The observations are generated as $t = y(x) + \eta$, where $y(x) = sin(2\pi x)$ and the Gaussian noise η is distributed by $N(0, \beta^{-1})$ with known $\beta = 11.1$. We generate a dataset with $N = 10$ samples $(\boldsymbol{x}, \boldsymbol{t})$ where $0 < x < 1$. We want to fit to the data a regression model $t = g(x, \boldsymbol{w}) + \eta$ where $g(x, \boldsymbol{w})$ is a 9th degree polynomial (M=9) with coefficient vector $\boldsymbol{w}$ which follows a normal prior distribution with precision $a = 0.005$.

In the python notebook we implement the following steps:

1. Generate $N = 10$ points with equal distance in the interval x = [0,1] and random white Gaussian noise $N(0, \beta^{-1})$.
2. Calculate:
   a. $m(x) = \beta\varphi(x)^T \boldsymbol{S} \sum_{n=1}^{N} \varphi(x_n) t_n$,
   b. $s^2(x) = \beta^{-1} + \varphi(x)^T \boldsymbol{S}\, \varphi(x)$ and
   c. $S^{-1} = \beta \sum_{n=1}^{N} \varphi(x_n)\varphi(x_n)^T$
   where $\varphi(x_n) = ({x_n}^0, {x_n}^1, \dots, {x_n}^M)^T$

   Using the MuCoeff_Sigma function we calculate the parameters of the posterior probability $p(\theta|y) \rightarrow N(\theta|\mu_{\theta|y}, \Sigma_{\theta|y})$:
   a. $\mu_{\theta|y} = \beta \boldsymbol{S} \sum_{n=1}^{N} \varphi(x_n) t_n$ and
   b. $S = [\beta \sum_{n=1}^{N} \varphi(x_n)\varphi(x_n)^T]^{-1}$

   Using the mu_sigma_y function we calculate the mean $m(x)$ and variance $s^2(x)$ of the predictive Gaussian model:
   a. $m(x) = \varphi(x)^T \mu_{\theta|y}$ and
   b. $s^2(x) = \beta^{-1} + \varphi(x)^T \boldsymbol{S}\, \varphi(x)$

   We split the $m(x)$ function like this in order to create the right plots in the following figures. The estimated curves of the right plots have the same coefficients as the estimated curves of the left plots and it was created using a test set of 100 points.

In Figure 5 we can observe the true model curve (black line) ± one SD, the noisy observations (green circles) and the estimated points (blue points/asterisks) with their variance (red error bar). The left image plots the estimated points using the training points and the right image shows the estimated curve (green line) ± one SD with the estimated y's.

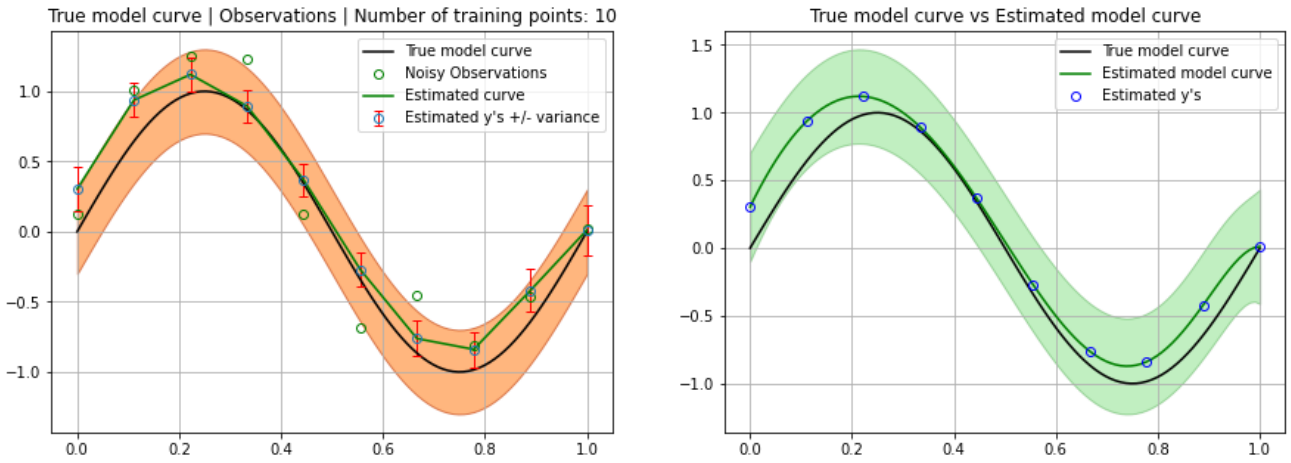

Figure 5: *Left plot: True model curve (± SD) plotted against the noisy observations, the estimated curve and the estimated*

points (N=10). Right plot: True model curve plotted against the estimated model curve (± SD) with the estimated points.

As we can see from Figure 6, on the first two plots where we have a small number of data points, the uncertainty is less in areas where data points are present. As we increase the number of points in the training set, the estimated curve approaches the true curve without reaching it because we do not use the same model (9th degree polynomial function) as the true model (sinusoid function). Therefore, as we increase the number of points in the training set, the bias decreases and uncertainty decreases.

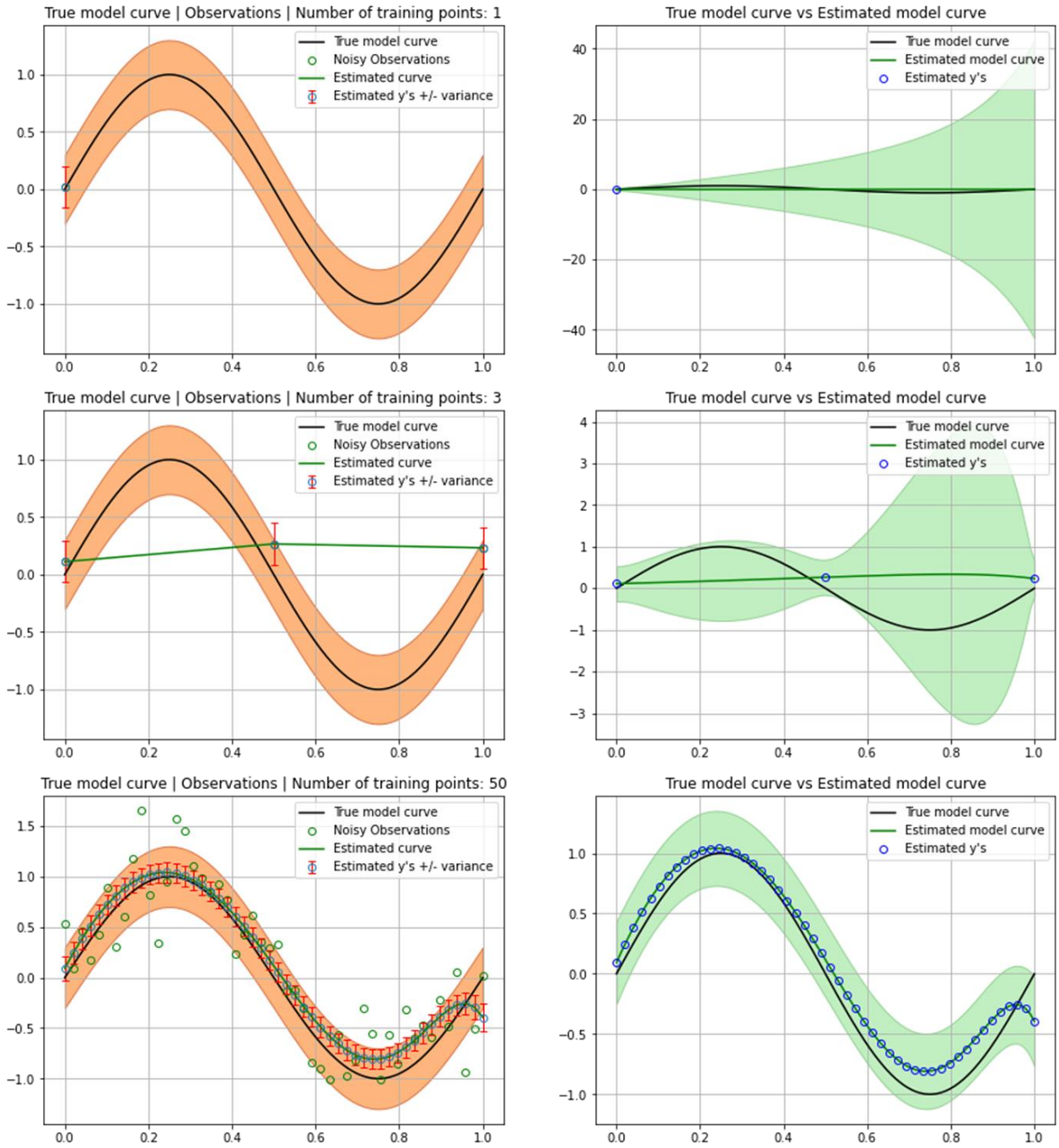

Figure 6: *Left plots: True model curve (± SD) plotted against the noisy observations, the estimated curve and the estimated points (N=1, 3, 50). Right plots: True model curve plotted against the estimated model curve (± SD) with the estimated points.*