Machine Learning in Computational Biology - First assignment
Mastoraki Aikaterina

1 – Analytical estimation of the mean of maximum likelihood

We know that $\{X_1,...,X_N\}$ are N random variables that follow a multidimensional Normal distribution, $p(x) \sim N(\theta, \Sigma)$. We want to derive analytically the Maximum Likelihood estimate μ_{ML} for the distribution's mean. The probability density function for a multidimensional Normal distribution is the following :

$$p(x) = \frac{1}{(2\pi)^{l/_2}} \frac{1}{(det\Sigma)^{1/_2}} \exp\left[-\frac{1}{2}(x \quad -\mu)^T \Sigma^{-1}(x-\mu)\right]$$

Where *l* = number of dimensions.

We want to estimate θ, so we can write:

$$p(x) \equiv p(x\backslash\theta)$$

From probability theory, we know that the probability of multiple independent events all happening is termed *joint probability* and we can apply joint probability density as follows:

$$p(X_1,...,X_N\backslash\theta) = p(X_1\backslash\theta) * ... * p(X_N\backslash\theta) = \prod_{i=1}^{N} p(X_i\backslash\theta)$$

Our goal is to maximize the above probability density by finding the optimal θ ≡ μ_{ML}.
To do that we should take the derivative of the probability density with respect to θ and equate it to zero. We will do that, however we will take the log likelihood.

$$\frac{d}{d\theta}\ln\left[\prod_{i=1}^{N} p(X_i\backslash\theta)\right] = \frac{d}{d\theta}\sum_{i=1}^{N}\ln[p(X_i\backslash\theta)]$$
$$= \sum_{i=1}^{N}\frac{d}{d\theta}\ln\left[\frac{1}{(2\pi)^{l/_2}}\frac{1}{(det\Sigma)^{1/_2}}\exp\left[-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right]\right]$$

- $\ln\left[\frac{1}{(2\pi)^{l/_2}}\frac{1}{(det\Sigma)^{1/_2}}\exp\left[-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right]\right] = \ln[\frac{1}{(2\pi)^{l/_2}}\frac{1}{(det\Sigma)^{1/_2}}] +$ $\ln\left[\exp\left[-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right]\right] = \ln[\frac{1}{(2\pi)^{l/_2}}\frac{1}{(det\Sigma)^{1/_2}}] - \frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$

- $\frac{d}{d\theta}\ln\left[\frac{1}{(2\pi)^{l/_2}}\frac{1}{(det\Sigma)^{1/_2}}\exp\left[-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)\right]\right] = \frac{d}{d\theta}\ln[\frac{1}{(2\pi)^{l/_2}}\frac{1}{(det\Sigma)^{1/_2}}] +$ $\frac{d}{d\theta}[-\frac{1}{2}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)]$ = 0 - $\frac{1}{2}\frac{d}{d\theta}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$

- $\frac{d}{d\theta}\ln\left[\prod_{i=1}^{N} p(X_i \backslash \theta)\right] = -\frac{1}{2}\Sigma^{-1}\sum_{i=1}^{N}\frac{d}{d\theta}(x_i-\mu)^T(x_i-\mu)$ = - $\Sigma^{-1}\sum_{i=1}^{N}(x_i-\mu)$

- Now we will set the last sum equal to zero, so we can find the optimal mean value.
  - $\Sigma^{-1}\sum_{i=1}^{N}(x_i-\mu)$ = 0

- We will multiply both sides of the previous equation with the covariance matrix:
  $-\sum_{i=1}^{N}(x_i-\mu)$ = 0 $\Rightarrow -\sum_{i=1}^{N}(x_i) - N\mu = 0 \Rightarrow$

$$\boldsymbol{\mu_{ML}} = \frac{\mathbf{1}}{\mathbf{N}}\sum_{\mathbf{i=1}}^{\mathbf{N}}(\mathbf{x_i}) \Rightarrow \boldsymbol{\mu_{ML}} = \bar{\mathbf{x}}$$

2 – Binomial distribution

The probability density function of the binomial distribution is the following :

$$p(x \setminus N,\mu) = \binom{N}{x}\mu^x(1-\mu)^{N-x} \text{ , x=0, 1, … , n}$$

We want to prove that :

- $E[x] = \mu = Nx$
- $Var[x] = \sigma^2 = N\mu(1-\mu)$

To achieve that we need to take advantage of the binomial theorem :

$$(\alpha+\beta)^m = \sum_{y=0}^{m}\binom{m}{y}\alpha^y\beta^{m-y}$$

<u>Proof of the mean :</u>

$$E[x] = \sum_x xp(x) = \sum_{x=0}^{n}\binom{N}{x}\mu^x(1-\mu)^{N-x} = \sum_{x=0}^{n} x\frac{N!}{x(N-x)!}\mu^x(1-\mu)^{N-x}$$

- x! = x*(x-1)*…*2*1
- $\frac{x}{x!} = (x-1)!$
- When x = 0, the last sum is equal to 0

Thus :

$$E[x] = \sum_{x=0}^{n} x\frac{N!}{x(N-x)!}\mu^x(1-\mu)^{N-x} = \sum_{x=1}^{n}\frac{N!}{(x-1)!\,(N-x)!}\mu^x(1-\mu)^{N-x}$$

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

At this point we have managed to extract the wanted quantity (Nμ) outside of the sum and we have to prove that the quantity inside the sum is equal to one. To do that we will make the following alteration : n-x = (n-1)(x-1). Thus the previous equation is now:

$$E[x] = N\mu \sum_{x=1}^{n} \frac{(N-1)!}{(x-1)!\,((N-1)-(x-1))!} \mu^{x} (1-\mu)^{N-x}$$

I will set m=(N-1) and y = (x-1). This alteration will impact also the limits of the sum

$$E[x] = N\mu \sum_{y=0}^{m} \frac{m!}{y!\,(m-y)!} \mu^{v} (1-\mu)^{m-y} = \; N\mu \sum_{y=0}^{m} \binom{m}{y} \mu^{v} (1-\mu)^{m-y}$$

We can use the binomial theorem on the last sum with α = μ and β = 1-μ.

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

Thus, we proved that **E[x] = Nμ**

<u>Proof of the variance :</u>

$$Var(X) = E[(X-\mu)^2] = E[X^2] - (E[X])^2$$

We have already proved that $E[X] = N\mu$ so, $(E[X])^2$ = $(N\mu)^2$. Thus we only have to calculate the $E[X^2]$.

$$E[X^2] = \sum_{x=0}^{N} x^2 \frac{N!}{x!\,(N-x)!} \mu^{x} (1-\mu)^{N-x}$$

It is not very easy to calculate $E[X^2]$ directly from the previous equation, so we will calculate $E[X^2 - X]$. $E[X^2 - X] = \; E[X^2] - \; E[X]$. Then $E[X^2]$ = $E[X^2 - X] + \; E[X]$.

$$E[X^2 - X] = E[X(X-1)] = \sum_{x=0}^{N} x(x-1) \frac{N!}{x!\,(N-x)!} \mu^{x} (1-\mu)^{N-x}$$

- The last sum is equal to 0 for x=0 and x=1
- x! = x(x-1)(x-2)!

$$E[X^2 - X] = \sum_{x=2}^{N} \frac{N!}{(x-2)!\,(N-x)!} \mu^{x} (1-\mu)^{N-x}$$

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

- Set n-x = (n-2)-(x-2)

$$E[X^2 - X] = N(N-1)\mu^2 \sum_{x=2}^{N} \frac{(N-2)!}{(x-2)!\,((N-2)-(x-2))!} \mu^{x-2} (1-\mu)^{(N-2)-(x-2)}$$

- Set y = x-2 and m = n-2

$$E[X^2 - X] = N(N-1)\mu^2 \sum_{y=0}^{m} \frac{m!}{y!\,(m-y)!} \mu^y (1-\mu)^{m-y}$$

The last equation can once again be solved using the binomial theorem. So :

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

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

$$\begin{aligned} Var(X) = \; E[X^2] - (E[X])^2 = \; N(N-1)\mu^2 + N\mu - \; (N\mu)^2 = N\mu\,[(N-1)\mu + 1 - N\mu] \\ = N\mu\,[N\mu - \mu + 1 - N\mu] = N\mu\,(1-\mu) \end{aligned}$$

Thus, we proved that $\boldsymbol{Var(X)}$= $\boldsymbol{N\mu\,(1-\mu)}$

3 – Posterior distribution

3.1 – Proof of mean and variance

We want to maximize p(**θ**\x). From Bayes theorem :

p(**θ**\x) = p(x\θ**)** p(**θ**)

- p(x\θ**) :** likelihood
- p(**θ**) : prior knowledge – Gaussian

$$\mathrm{p}(\boldsymbol{\theta}\backslash \mathrm{x}) = \frac{1}{2\pi\sqrt{\sigma^N \sigma_0}} \exp\left[-\sum_{i=1}^{N} \frac{(x_i-\theta)^2}{2\sigma^2}\right] \exp\left[-\frac{(\theta-\theta_0)^2}{2\sigma_0^2}\right] = \frac{1}{2\pi\sqrt{\sigma^N \sigma_0}} \exp\left[-\sum_{i=1}^{N} \frac{(x_i-\theta)^2}{2\sigma^2} - \frac{(\theta-\theta_0)^2}{2\sigma_0^2}\right]$$

- Set A = $-\sum_{i=1}^{N} \frac{(x_i-\theta)^2}{2\sigma^2} - \frac{(\theta-\theta_0)^2}{2\sigma_0^2}$. We will gather all the similar terms together.

$$A = \frac{-\theta^2 + 2\theta\theta_0 - \theta_0^2}{2\sigma_0^2} - \sum_{i=1}^{N} \frac{x_i^2 - 2\theta x_i + \theta^2}{2\sigma^2} = \frac{-\theta^2 + 2\theta\theta_0\sigma^2 - \; \theta_0^2\sigma^2 - \; \sum_{i=1}^{N}[x_i^2\sigma_0^2 - 2\theta x_i\sigma_0^2 + \theta^2\sigma_0^2]}{2\sigma_0^2\sigma^2} = \frac{-\theta^2 + \dfrac{2\theta(\theta_0\;\sigma^2 + \sigma_0^2\sum_{i=1}^{N} x_i)}{\sigma^2 + N\sigma_0^2} + C}{\dfrac{2\sigma_0^2\sigma^2}{\sigma^2 + N\sigma_0^2}}$$

We will add and subtract to the numerator of A the following quantity :

$$\frac{\theta_0 \ \sigma^2 + \sigma_0^2 \sum_{i=1}^{N} x_i}{(\sigma^2 + N\sigma_0^2)^2}$$

Now A will be :

$$A = -\frac{\left(\theta - \frac{\theta_0 \ \sigma^2 + \sigma_0^2 \sum_{i=1}^{N} x_i}{\sigma^2 + N\sigma_0^2}\right)^2}{\frac{2\sigma_0^2\sigma^2}{\sigma^2 + N\sigma_0^2}} + C$$

Thus we have a new Gaussian with :

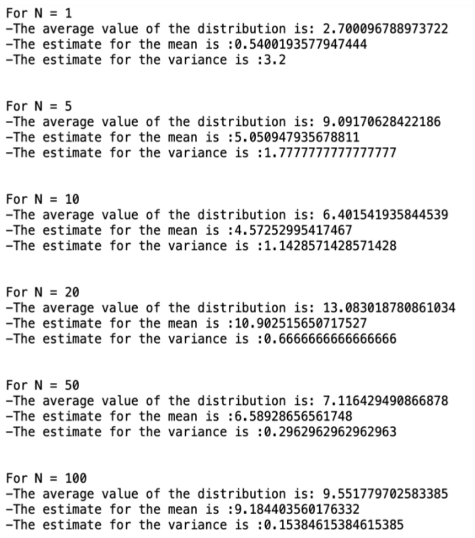
- mean μ_N = $\frac{\theta_0 \ \sigma^2 + \sigma_0^2 \sum_{i=1}^{N} x_i}{\sigma^2 + N\sigma_0^2} = \frac{\theta_0 \ \sigma^2 + \sigma_0^2 + N\sigma_0^2 \bar{x}}{\sigma^2 + N\sigma_0^2}$ where $\bar{x} = \sum_{i=1}^{N} x_i$
- variance $\sigma_N^2 = \frac{\sigma_0^2\sigma^2}{\sigma^2 + N\sigma_0^2}$

3.2.a – Mean and variance calculation using Jupyter Notebook

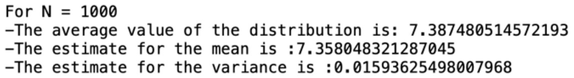
Using Jupyter Notebook and python I calculated the mean and the variance (using the analytical solution that I proved earlier) of the distribution for different values of N. Before the presentation of my results, let's have a look at what we expect to happen as we generate more points into our data set.

- N → ∞ $N\sigma_0^2\bar{x}$ and $N\sigma_0^2 \gg \theta_0 \ \sigma^2$ and $\sigma^2 \Rightarrow \mu_N \approx \frac{N\sigma_0^2\bar{x}}{N\sigma_0^2} \approx \bar{x}$. Thus, the estimation is getting closer to the average of the distribution.
- N → $0 \ \Rightarrow \mu_N \approx \frac{\theta_0 \ \sigma^2}{\sigma^2} \approx \theta_0$. Thus, the estimation is getting closer to our prior knowledge about θ, in our case $\theta_0 \ = 0$.

The next figure shows the different estimations of the mean and the variance of my distribution



```
For N = 1
-The average value of the distribution is: 2.700096788973722
-The estimate for the mean is :0.5400193577947444
-The estimate for the variance is :3.2


For N = 5
-The average value of the distribution is: 9.09170628422186
-The estimate for the mean is :5.050947935678811
-The estimate for the variance is :1.7777777777777777


For N = 10
-The average value of the distribution is: 6.401541935844539
-The estimate for the mean is :4.57252995417467
-The estimate for the variance is :1.1428571428571428


For N = 20
-The average value of the distribution is: 13.083018780861034
-The estimate for the mean is :10.902515650717527
-The estimate for the variance is :0.6666666666666666


For N = 50
-The average value of the distribution is: 7.116429490866878
-The estimate for the mean is :6.58928656561748
-The estimate for the variance is :0.2962962962962963


For N = 100
-The average value of the distribution is: 9.551779702583385
-The estimate for the mean is :9.184403560176332
-The estimate for the variance is :0.15384615384615385
```



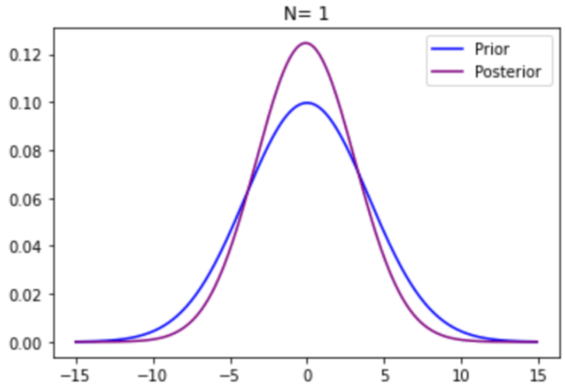
```
For N = 1000
-The average value of the distribution is: 7.387480514572193
-The estimate for the mean is :7.358048321287045
-The estimate for the variance is :0.01593625498007968
```

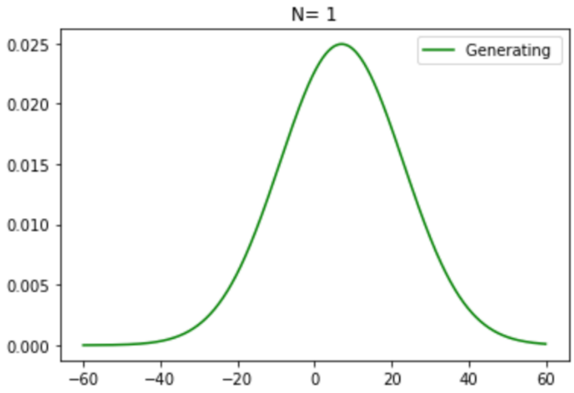
As we can see from the above results, when we have only one point in our data set, we get a mean that is closer to θ_0 (in our case $\theta_0 = 0$ and the estimation is 0.54), than to the actual mean or the average value of the distribution. We also observe that the variance is high, as we don't have a lot of points in the data set, so we are very uncertain about our calculation. However, as we generate more points, the mean value is augmented and the variance is decreasing. Thus, our calculation about the mean is getting closer to the average value of the distribution and the variance decreases, as we have more confidence that the estimation of the mean value is close to the real one. When we have 1,000 points, the estimation of the mean value is almost identical to the average and the true mean of the distribution and the variance is very low.

3.2.b – Distributions' plots

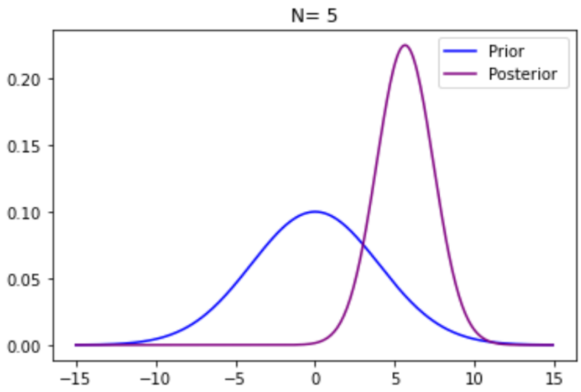
Below are presented the plots of the prior, the posterior and the distribution that generates the data for the various observations :

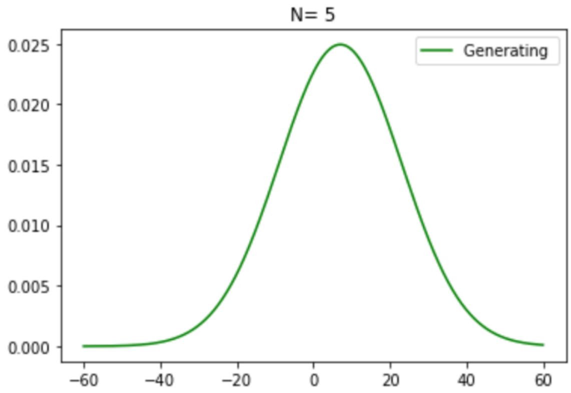
- **N = 1**



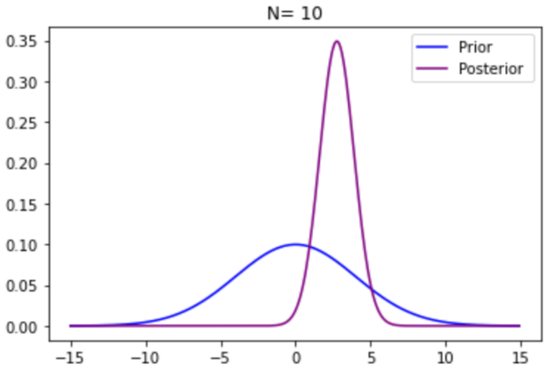


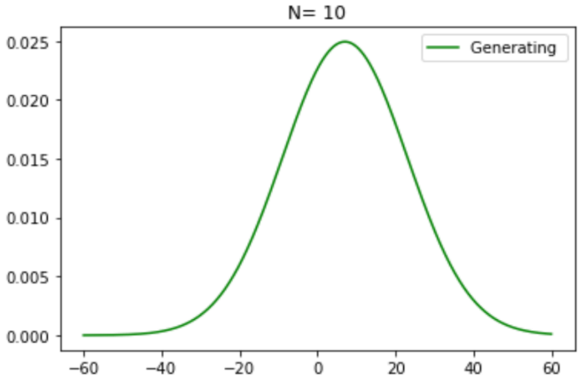
- **N = 5**



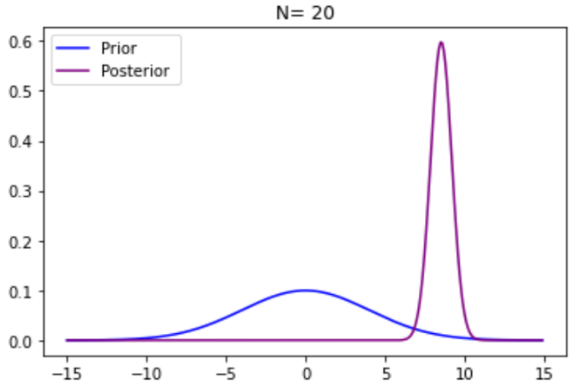


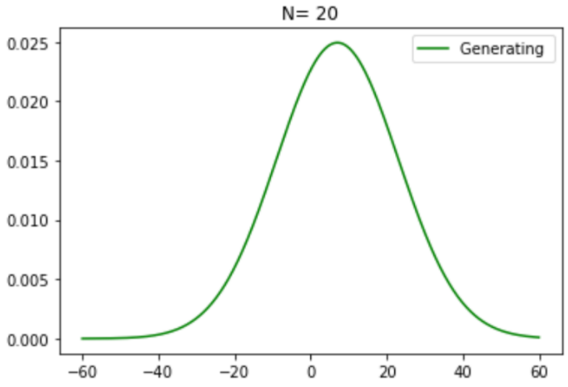
- **N = 10**


N= 10
Prior
Posterior
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−15
−10
−5
0
5
10
15

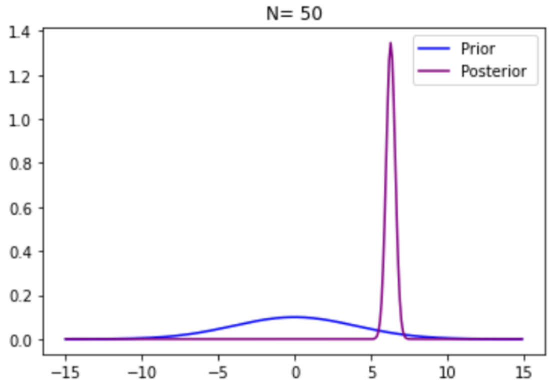

N= 10
Generating
0.025
0.020
0.015
0.010
0.005
0.000
−60
−40
−20
0
20
40
60

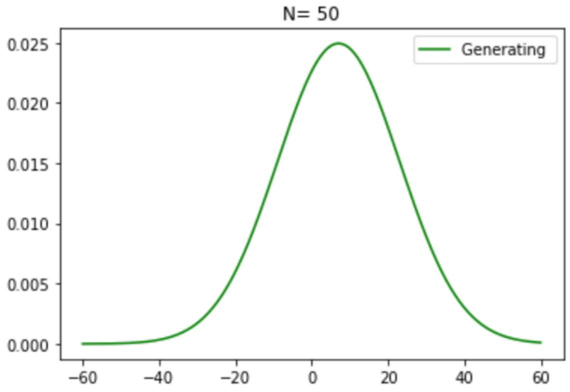
- **N = 20**


N= 20
Prior
Posterior
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−15
−10
−5
0
5
10
15

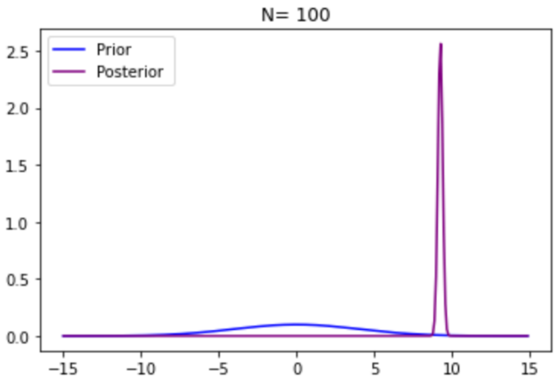

N= 20
Generating
0.025
0.020
0.015
0.010
0.005
0.000
−60
−40
−20
0
20
40
60

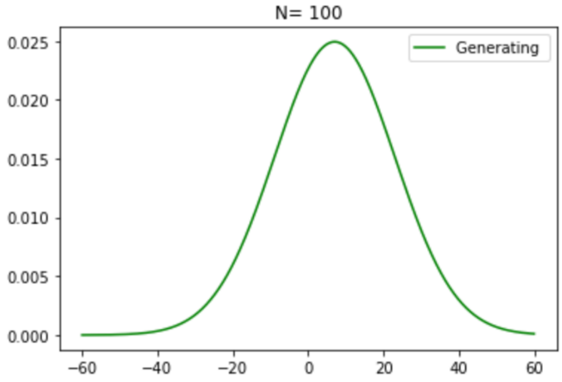
- **N = 50**


N= 50
Prior
Posterior
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−15
−10
−5
0
5
10
15

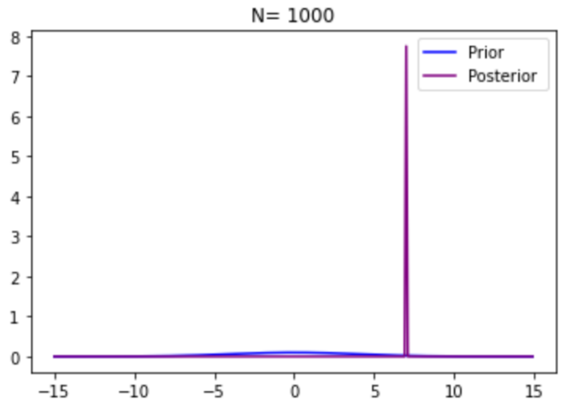

N= 50
Generating
0.025
0.020
0.015
0.010
0.005
0.000
−60
−40
−20
0
20
40
60

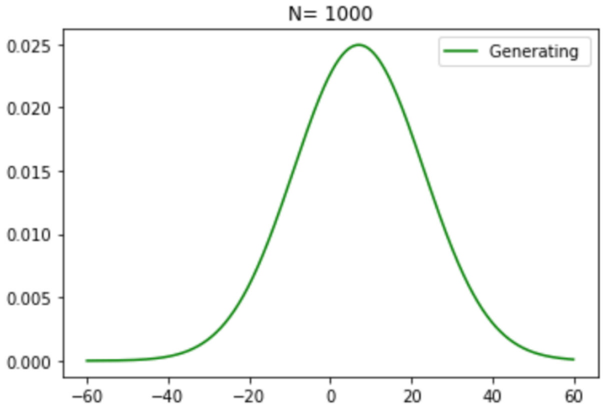
- **N = 100**





- **N = 1000**

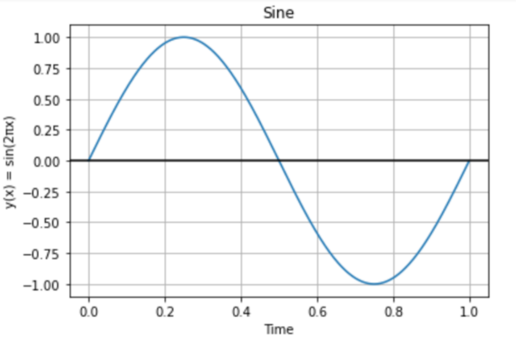




The mean and the variance values for each N are changing once the python code is re executed, so the mean and the variance of the plotted distributions may differ from the presented mean and variance values of the first figure. From the plots we can infer the same conclusion as we did once we saw the estimated mean and the estimated variance values for each N.

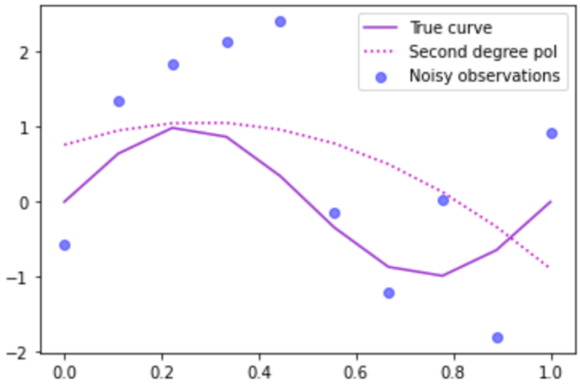
4 – Polynomial curve fitting

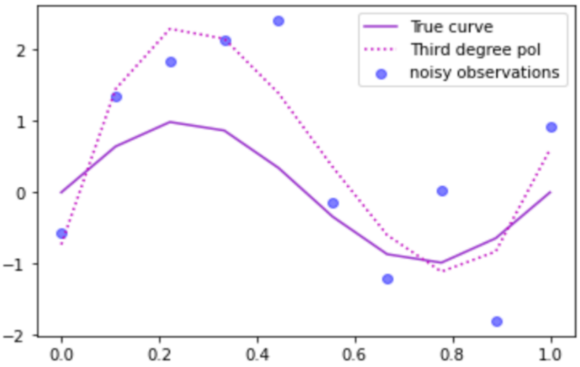
For this experiment, firstly I plotted a period of the sinusoidal function y(x)=sin(2πx) as we can see on the next plot :

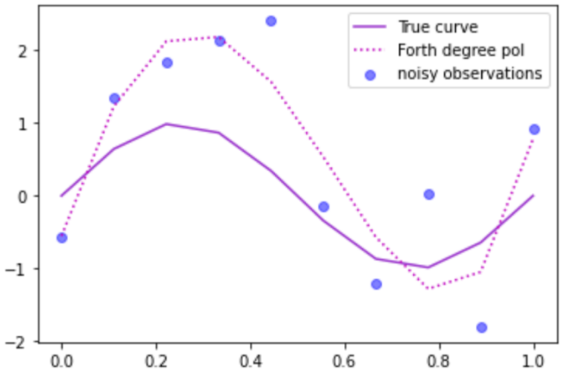


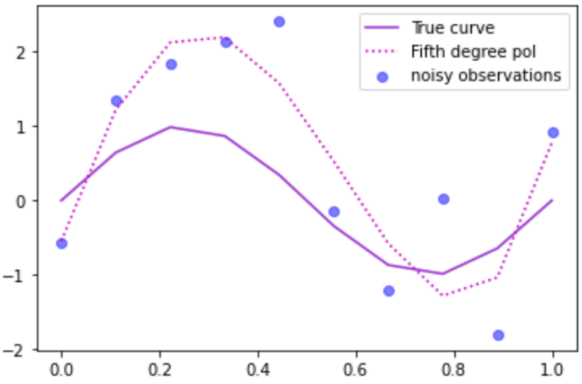
Then, I generated the observations for the sinusoidal function and I added white Gaussian noise to each observation. After that, I used the polyfit function to fit to the noisy data various polynomials of different degrees. I repeated the previous process for two different set of observations, one set with 10 points and another set with 100 points. Below you can see the plots, the coefficients and the RMSE for each polynomial :

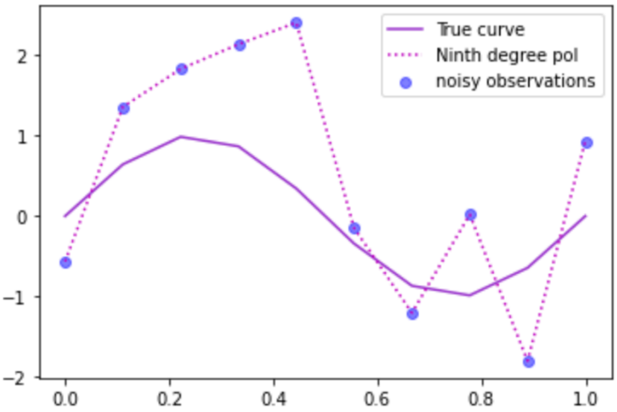
- **N = 10**

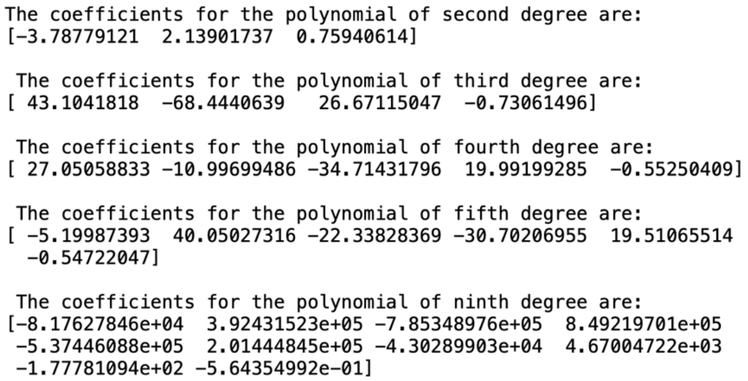




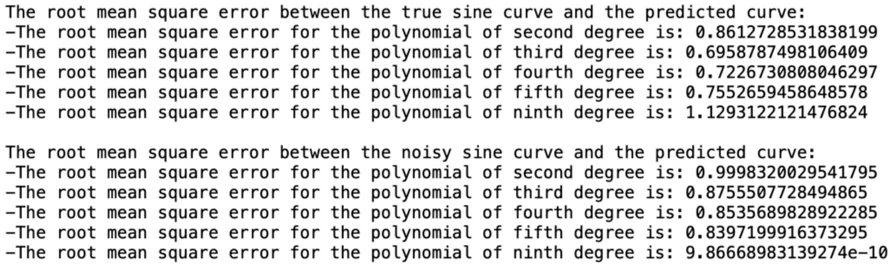








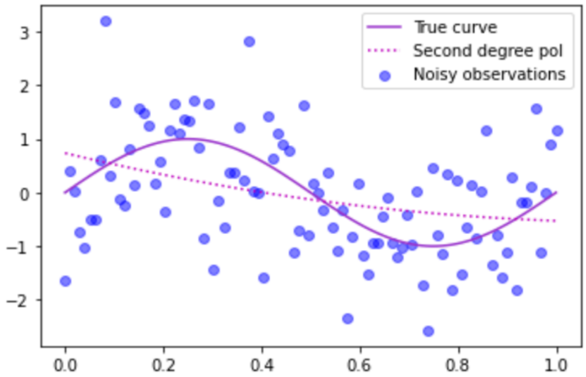
```
The coefficients for the polynomial of second degree are:
[-3.78779121  2.13901737  0.75940614]

 The coefficients for the polynomial of third degree are:
[ 43.1041818  -68.4440639   26.67115047  -0.73061496]

 The coefficients for the polynomial of fourth degree are:
[ 27.05058833 -10.99699486 -34.71431796  19.99199285  -0.55250409]

 The coefficients for the polynomial of fifth degree are:
[ -5.19987393  40.05027316 -22.33828369 -30.70206955  19.51065514
  -0.54722047]

 The coefficients for the polynomial of ninth degree are:
[-8.17627846e+04  3.92431523e+05 -7.85348976e+05  8.49219701e+05
 -5.37446088e+05  2.01444845e+05 -4.30289903e+04  4.67004722e+03
 -1.77781094e+02 -5.64354992e-01]
```

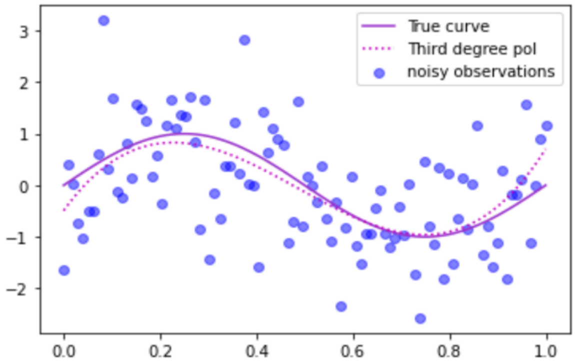


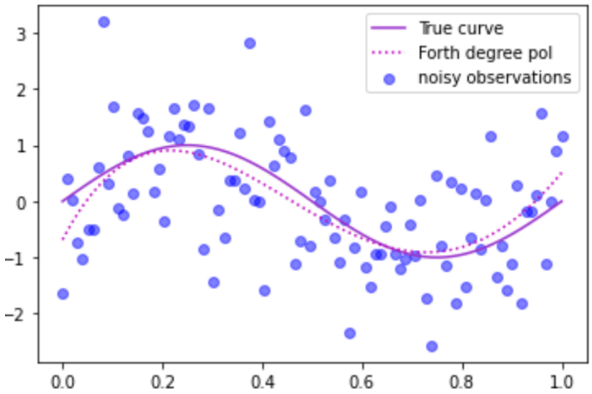
```
The root mean square error between the true sine curve and the predicted curve:
-The root mean square error for the polynomial of second degree is: 0.8612728531838199
-The root mean square error for the polynomial of third degree is: 0.6958787498106409
-The root mean square error for the polynomial of fourth degree is: 0.7226730808046297
-The root mean square error for the polynomial of fifth degree is: 0.7552659458648578
-The root mean square error for the polynomial of ninth degree is: 1.1293122121476824

The root mean square error between the noisy sine curve and the predicted curve:
-The root mean square error for the polynomial of second degree is: 0.9998320029541795
-The root mean square error for the polynomial of third degree is: 0.8755507728494865
-The root mean square error for the polynomial of fourth degree is: 0.8535689828922285
-The root mean square error for the polynomial of fifth degree is: 0.8397199916373295
-The root mean square error for the polynomial of ninth degree is: 9.86668983139274e-10
```

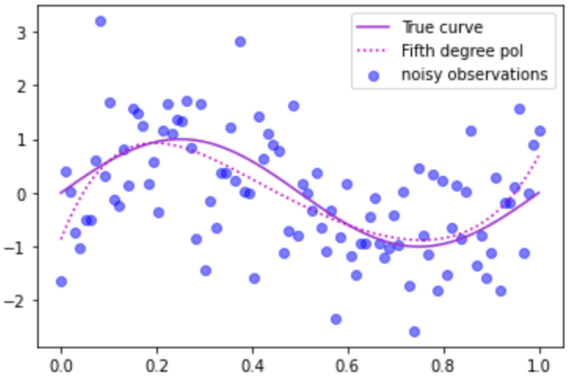
For the first set, that contains only 10 data points, we can see that by increasing the complexity of the model, each model approaches even better the noisy observations than the previous one and the ninth-degree polynomial is able to fit perfectly the noisy observations. We can also see the coefficients that minimizes the squared error for each degree and two types of RMSE. The first one is the RMSE between the true curve and the predicted polynomial with the best coefficients and the second one is the RMSE between the noisy curve and the same predicted curve. As we can observe, the RMSE between the noisy observations and the predicted curve, decreases as the complexity (degree) of each polynomial increases. We can see that the RMSE plummets by changing the degree of the curve from 5 to 9. This makes sense, as the noisy data are perfectly fitted by the ninth-degree polynomial, while all the other curves just approach the data. On the other hand, the RMSE between the true model and the predicted curves, drops for the first two polynomials and then starts rising again for the last three curves. This observation is also reasonable, as the third-degree polynomial approaches the best the true sine curve. However, using polynomials with higher degree, leads to overfitting of the noisy data, thus the predicted curves start once again to diverge from the true curve.

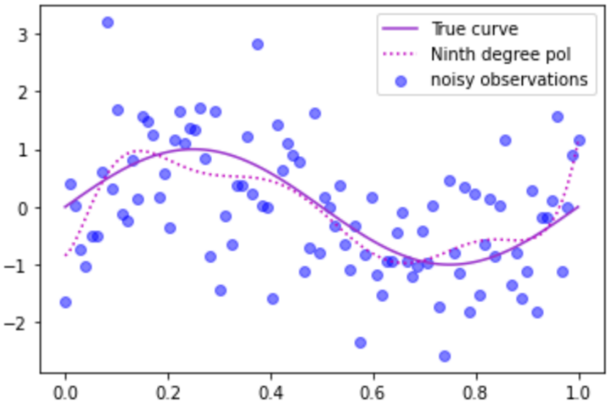
- **N = 100**

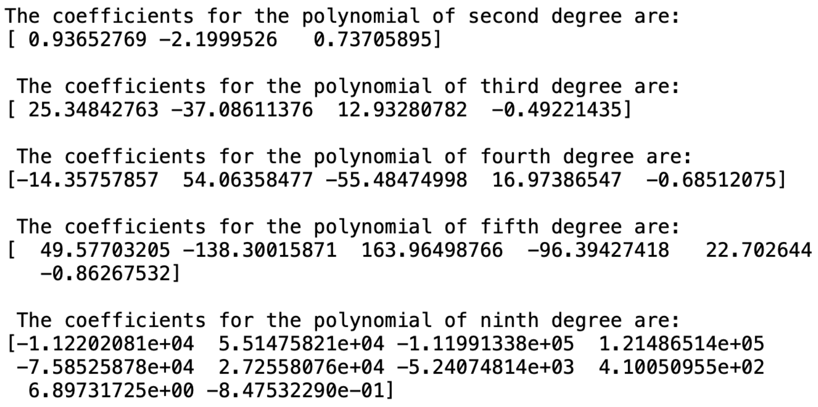




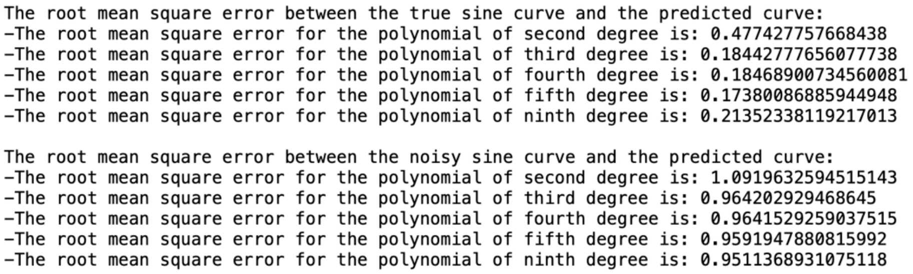








```
The coefficients for the polynomial of second degree are:
[ 0.93652769 -2.1999526   0.73705895]

 The coefficients for the polynomial of third degree are:
[ 25.34842763 -37.08611376  12.93280782  -0.49221435]

 The coefficients for the polynomial of fourth degree are:
[-14.35757857  54.06358477 -55.48474998  16.97386547  -0.68512075]

 The coefficients for the polynomial of fifth degree are:
[  49.57703205 -138.30015871  163.96498766  -96.39427418   22.702644
   -0.86267532]

 The coefficients for the polynomial of ninth degree are:
[-1.12202081e+04  5.51475821e+04 -1.11991338e+05  1.21486514e+05
 -7.58525878e+04  2.72558076e+04 -5.24074814e+03  4.10050955e+02
  6.89731725e+00 -8.47532290e-01]
```



```
The root mean square error between the true sine curve and the predicted curve:
-The root mean square error for the polynomial of second degree is: 0.477427757668438
-The root mean square error for the polynomial of third degree is: 0.18442777656077738
-The root mean square error for the polynomial of fourth degree is: 0.18468900734560081
-The root mean square error for the polynomial of fifth degree is: 0.17380086885944948
-The root mean square error for the polynomial of ninth degree is: 0.21352338119217013

The root mean square error between the noisy sine curve and the predicted curve:
-The root mean square error for the polynomial of second degree is: 1.0919632594515143
-The root mean square error for the polynomial of third degree is: 0.964202929468645
-The root mean square error for the polynomial of fourth degree is: 0.9641529259037515
-The root mean square error for the polynomial of fifth degree is: 0.9591947880815992
-The root mean square error for the polynomial of ninth degree is: 0.9511368931075118
```

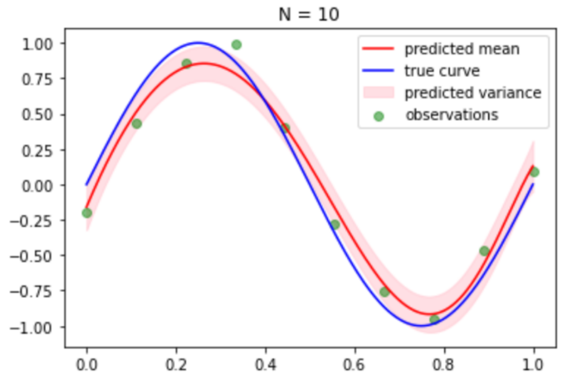
The second set contains 100 noisy observations. As we can see from the plots, there are three polynomials that approach the best the true curve, the third, the fourth and the fifth-degree polynomials. We can also see that none of the models is able to fit perfectly the noisy data, as we have more observations than the previous time. We can see that the calculated RMSE values between the predicted and the noisy curve are continuously decreasing. However, the difference in the RMSE between the last two polynomials is very low. On the other hand, the RMSE between the true curve and the predicted curves is decreasing for the first four polynomials, but increases a bit for the last one. As we observed from the plots, the third, the fourth and the fifth-degree polynomials can approach better the sine function. Their RMSE values are very low and very close to each other's, with the fifth-degree polynomial having a slightly best RMSE value than the two previous curves. This means that the fifth-degree polynomial approaches better the sine curve. Finally, the RMSE of the ninth-degree polynomial is increased compared to the value of the fifth-degree polynomial. This is maybe resulted by the overfitting of the higher degree polynomial to the noisy data (we can also observe that in the plots, as the ninth-degree polynomial has more curved parts, as it tries to fit better the noisy observations).

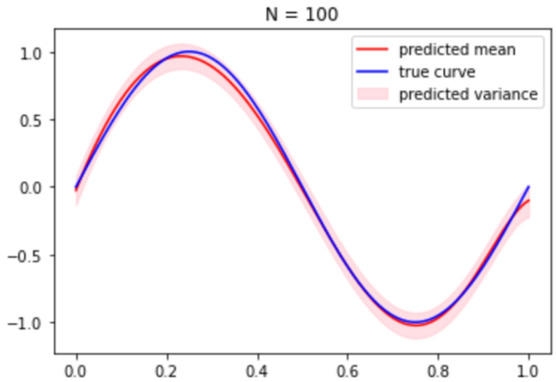
5 – Bayesian curve fitting

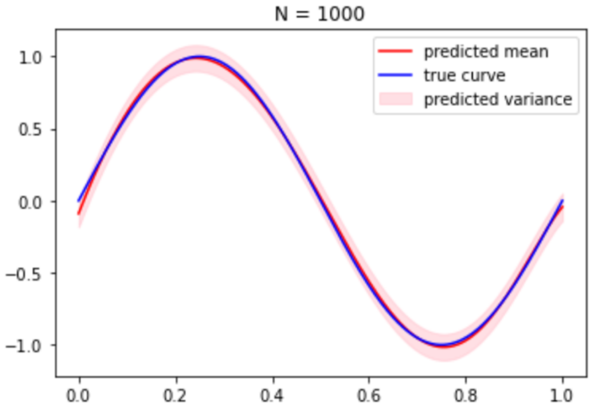
Starting this experiment, I generated 10 points of the sinusoidal function and I added to them Gaussian noise with 0 mean and β^{-1} = 1/11.1 variance. Then I calculated the ϕ(x) matrix. The ϕ(x) is a matrix with N (number of observations x) rows and M+1 columns, where M is the degree of the polynomial we want to fit to our data. For every row, the elements of the fist column are all equal to 1, denoting the fixed term of the polynomial. The element of row i and column j is equal to ϕ[i][j] = x_i^j, where x_i is the ith observation in our data set. Then I calculated the S, the mean and the variance of the predictive model:

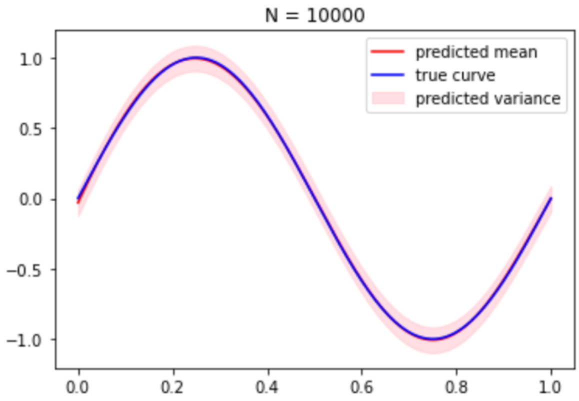
- $S^{-1} = \alpha I + \beta \sum_{n=1}^{N} \varphi(x_n)\varphi(x_n)^T$
- m(x) = $\beta\varphi(x_n)^T S \sum_{n=1}^{N} \varphi(x_n)\, t_n$
- $s(x)^2 = \beta^{-1} + \varphi(x_n)^T S\, \varphi(x_n)$

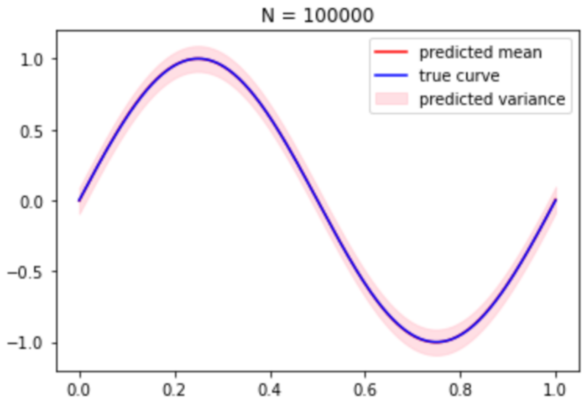
Below you can see the plots of the mean and the variance of the predictive model and the true curve for different sets with various numbers (N) of observations :











As we can infer from the previous plots, as we acquire more points in the data set, the variance is decreasing and the mean of the predictive distribution approaches better and better the true curve. When we have only 10 points to the data set, we have a greater variance and the predictive mean is far from the true curve. However, when the data set includes 100,000 points, the predicted mean is identical to the true curve and the variance is low.