Module-3

Antara Paul, Labani Roy

Activity 1: Maximum Likelihood estimator

100 random variables are generated from binomial distribution, where number of trials is 1 and size is 100 (1 binoimal trial is equivalent to 100 bernoulli trials). To get the maximum likelihood estimator of head probability (p), we need to maximize the likelihood function w.r.t p. In this question, likelihood function is a binomial distribution with $N = 100$ bernoulli trials.
Probability of getting x from a bernoulli distribution is, $P(X = x) = p^x(1-p)^{(1-x)}$
So, Likelihood, $L(p) = \prod_{n_i=1}^{N} p^{n_i}(1-p)^{(1-n_i)}$ The log-likelihood becomes, $\log L(p) = \sum_{n_i=1}^{N}(n_i \log p + (1-n_i)\log(1-p))$.

Maximization of $L(p)$ in terms of p will give us, $\frac{\partial \log L(p)}{\partial p} = \sum_{n_i=1}^{N}(\frac{n_i}{p} - \frac{1-n}{1-p}) = 0$. So, the maximum likelihood estimator of p, $\hat{p} = \frac{\sum_{n_i=1}^{N} n_i}{N}$

Suppose, k = number of heads, likelihood, $L(k,p) = \binom{N}{k}p^k(1-p)^{(N-k)}$ and the log-likelihood, $\log L(k,p) = \log\left(\frac{N!}{k!(N-k)!}\right) + k\log p + (N-k)\log(1-p)$.

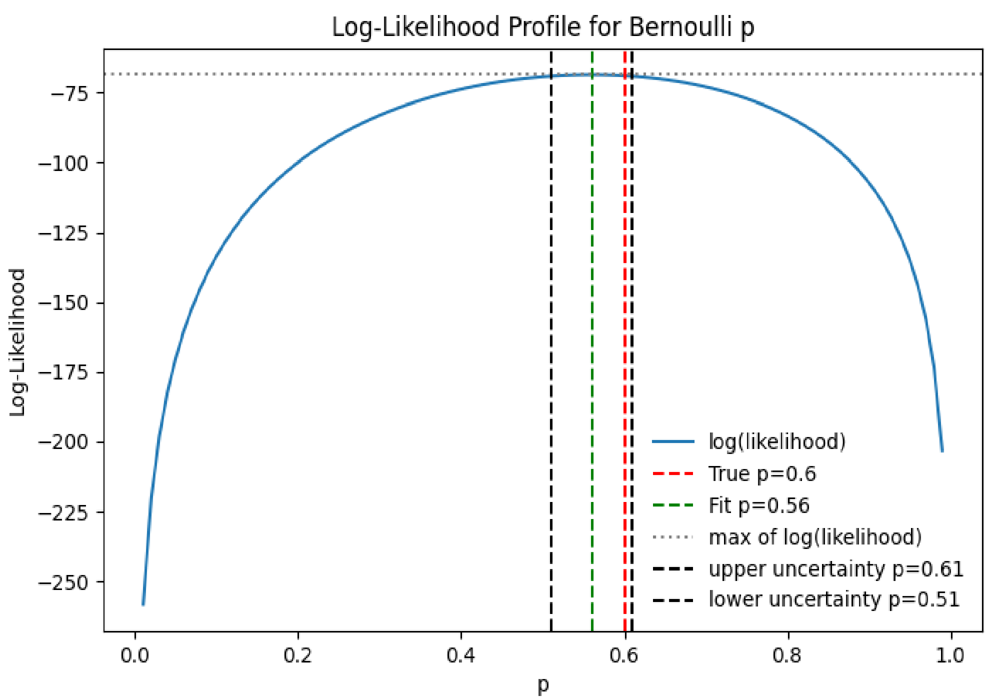

Figure 1: log-likelihood vs p (prob. of head outcome) plot

We considered true probability (p_{true}) of head outcome is 0.6. From the maximum likelihood estimation of p we get p_{MLE} or p_{fit}. Figure(1) shows p_{fit} is not properly p_{true}. If the size of binomial trial (number of bernoulli trial) increases then p_{fit} converges to it's true value p_{true}. To find out the uncertainty on p_{fit}, we look for the range of p for which log-likelihood function does not drop too much from it's maximum value. Black dashed vertical lines show the roots of this equation $\log(L(p)) = max(\log(L(p))) - 1/2$, which corresponds to 1σ confidence interval. So $p = 0.51$ and $p = 0.61$ define the 1σ confidence interval.

Activity 2: Decay fit

For our fit, we have used the `iminuit` package and its `LeastSquares` method, correctly accounting for the different *yerr* provided in the data file. The fit equation used was:

$$\begin{aligned} y &= a_1 e^{-a_2 t} + b_1 e^{-b_2 t} + c_1 t \\ t &= log(x) \quad \text{or} \\ x &= e^{t/100} \quad \text{(to rescale so } e^{800} \text{ doesn't return inf)} \\ y &= a_1 x^{-100 a_2} + b_1 x^{-100 b_2} + 100 c_1 log(x) + c_2 \end{aligned} \tag{1}$$

Calculating the lifetime values from the fit parameters a_2 and b_2, and propagating the errors accordingly, the final values are shown Table 1.

Isotope	**Known lifetime (s)**	**Lifetime from fit with error (s)**
Ag^{47}_{108}	207	247.52 ± 31.25
Ag^{47}_{110}	35.5	35.09 ± 2.54

Table 1: Lifetime values compared to fit parameters

We have plotted our data with a $\log(y)$ scale to better distinguish the decay of the two Ag isotopes. As can be seen from both the plot and the table, our model describes the data quite well, with errors taken into account.
Furthermore, our pull plot, defined as $\frac{y_{obs} - y_{model}}{y_{obs}}$, shows a good agreement with most of the points lying within (or close to) the $\pm 1\sigma$ band.

The $p-$value of our fit was found to be 0.08. Considering that the null hypothesis is taken that the data and model have no significant statistical differences, this $p-$value (> 0.05) is an acceptable one as it supports the null hypothesis.

Activity 3: Bayesian Linear regression

a. For the first fitting here, we have used `scipy.optimize.curve_fit` as its default method is least squares. We used the same data as Module 1 ($y = \sin 2\pi x + \mathcal{N}(0, 0.3)$), but could not fit for N=10 data, as the number of parameters in all the three models is more than 10. This can be fixed in the next case when we use the bayesian linear regression method. The least squares fit, along with the reduced χ^2 is shown in Figs. 3 and 4 below. Initial parameters here were taken as all 1; even tuning them to different values brought the fit to the same state. For N=100, the gaussian function gives the best fit, with its reduced $\chi^2 = 1$. For N=1000, all fits look similar, giving the same χ^2 value.

b. The bayesian predictive distribution takes the form:

$$p(t \mid \mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}\left(t \mid \mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x})\right) \tag{2}$$

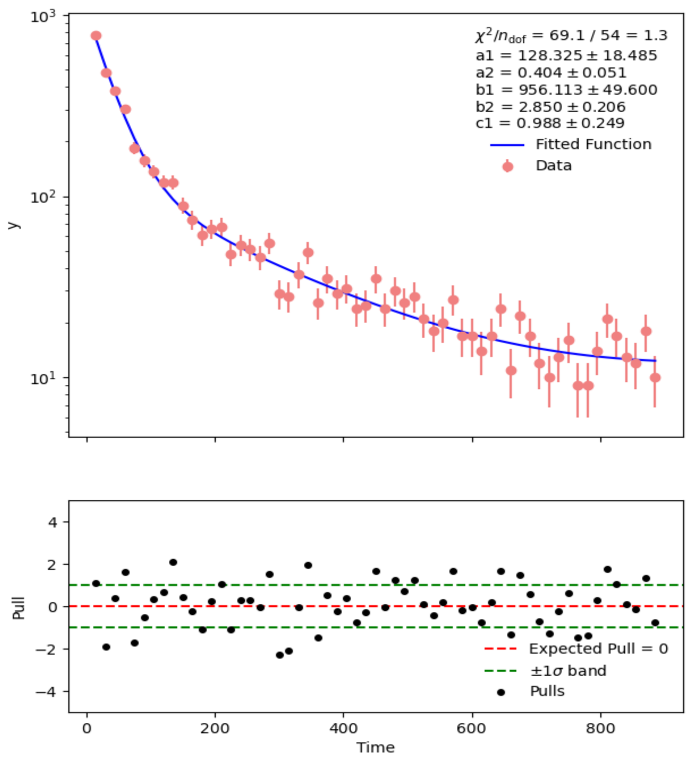

Figure 2: Above: Data with our fit. Below: Pull plot

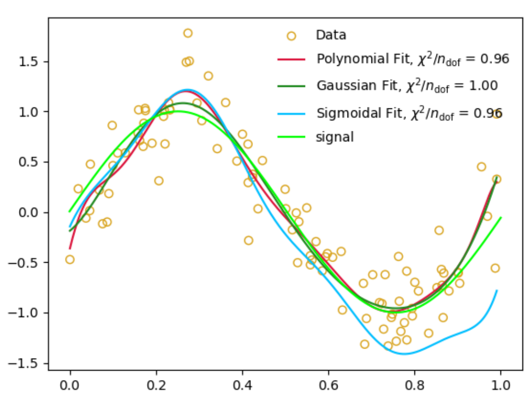

Figure 3: N = 100

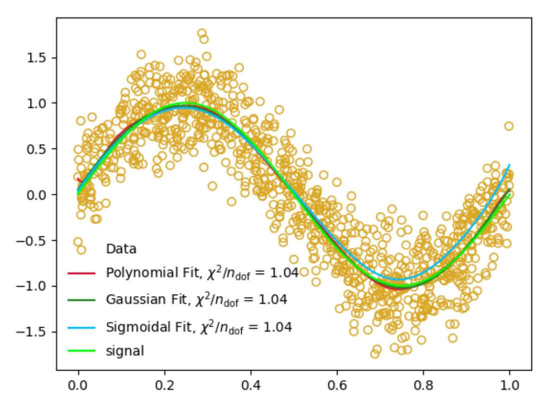

Figure 4: N = 1000

where, the mean is $\mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$ and the variance $\sigma_N^2(\mathbf{x})$ is given by $\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x})$. The parameters $\mathbf{m}_N$ and $\mathbf{S}_N$ are the mean and variance for the posterior distribution, and $\boldsymbol{\phi}(\mathbf{x})$ is the *design matrix*, whose elements define the basis functions. For us, the basis functions were polynomial of 11th degree (Fig 5), 11 gaussian pdfs (Fig 6) and 11 sigmoidal pdfs (Fig 7).

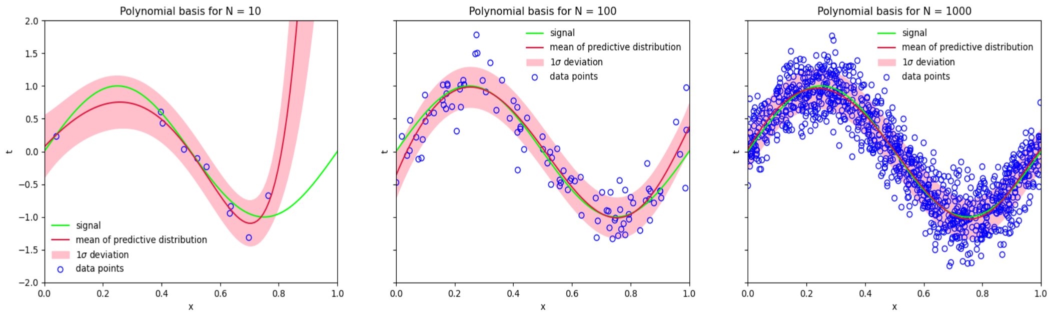

Figure 5: Predictive distribution for polynomial of 11th degree

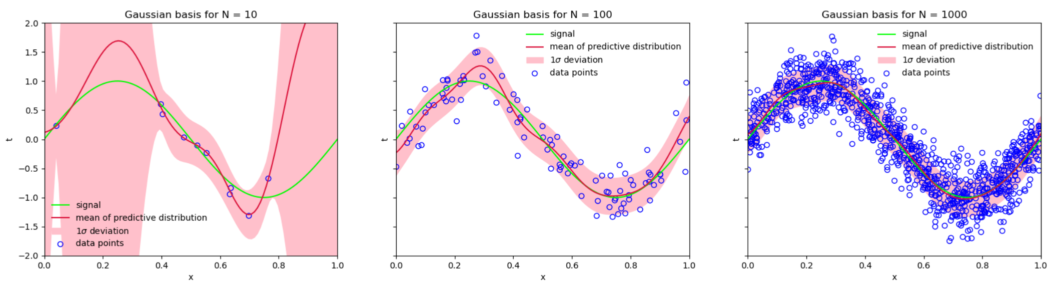

Figure 6: Predictive distribution for 11 Gaussian pdf

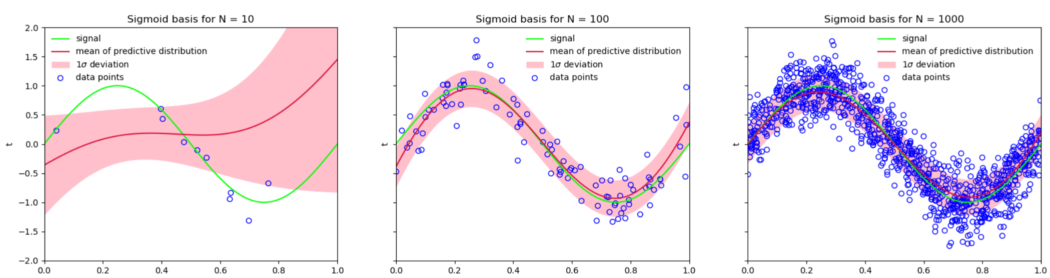

Figure 7: Predictive distribution for 11 sigmoidal pdf