S1: Principals of Data Science - Coursework

William Knottenbelt, wdk24

December 10, 2023

Section A

Part (a)

The total probability density function, p, is normalised over $M \in [-\infty, +\infty]$ if it integrates to unity. The integral can be written as a sum of signal and background terms:

$$\int_{-\infty}^{+\infty} p \, dM = f \int_{-\infty}^{+\infty} s \, dM + (1-f) \int_{0}^{+\infty} b \, dM. \tag{1}$$

We require the identity:

$$\int_{-\infty}^{+\infty} e^{-ax^2} = \sqrt{\frac{\pi}{a}}, \tag{2}$$

For the signal term, we have:

$$\int_{-\infty}^{+\infty} s \, dM = \int_{-\infty}^{+\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(M-\mu)^2}{2\sigma^2}\right) dM = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx,$$

where we used the substitution $x = M - \mu$. This integral has the form of identity (2) with $a = \frac{1}{2\sigma^2}$, hence:

$$\int_{-\infty}^{+\infty} s \, dM = \frac{1}{\sigma\sqrt{2\pi}} \sqrt{2\sigma^2\pi} = 1.$$

For the background term we have:

$$\int_{0}^{+\infty} b \, dM = \int_{0}^{+\infty} \lambda e^{-\lambda M} \, dM = (-e^{-\lambda M})|_{0}^{+\infty} = 1.$$

Hence, both the signal and background terms integrate to unity. We can plug this into equation (1) and we see that the normalisation condition is satisfied:

$$\int_{-\infty}^{+\infty} p \, dM = f + (1-f) = 1.$$

Part (b)

In general, any probability density function, $g(X)$, can be normalised over an arbitrary range $[\alpha, \beta]$ by multiplying it by the factor $\frac{1}{F(\beta)-F(\alpha)}$, where F is its cumulative density function, CDF. To see proof of this, we add normalisation factor, A, to g. We have:

$$1 = \int_{\alpha}^{\beta} Ag(M)\, dM = A\left(\int_{-\infty}^{\beta} g(M)\, dM - \int_{-\infty}^{\alpha} g(M)\, dM\right) = A(F(\beta) - F(\alpha)).$$

Hence:

$$A = \frac{1}{F(\beta) - F(\alpha)}.$$

To ensure that the signal and background distributions contribute the correct fractions (f and $(1-f)$) to the total probability, they must be normalised individually before summing them. To normalise the signal, we add the factor $\frac{1}{\Phi(\frac{\beta-\mu}{\sigma})-\Phi(\frac{\alpha-\mu}{\sigma})}$, where Φ is the CDF of the standard normal distribution. The CDF of the exponential decay distribution is:

$$F(X) = \begin{cases} 1 - e^{-\lambda X} & \text{for } X \geq 0, \\ 0 & \text{for } X < 0. \end{cases}$$

Hence, provided that $\alpha, \beta > 0$, the normalisation factor for the background distribution is:

$$\frac{1}{e^{-\lambda\alpha} - e^{-\lambda\beta}}.$$

Normalisation of the signal and background guarantees the total PDF to be normalised:

$$\int_{\alpha}^{\beta} p\, dM = f \int_{\alpha}^{\beta} s\, dM + (1-f) \int_{\alpha}^{\beta} b\, dM = f + (1-f) = 1.$$

Hence, assuming $\alpha, \beta > 0$, the full expression is:

$$p(M; \boldsymbol{\theta}) = \frac{f}{\Phi(\frac{\beta-\mu}{\sigma}) - \Phi(\frac{\alpha-\mu}{\sigma})} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(M-\mu)^2}{2\sigma^2}\right) + (1-f)\frac{\lambda e^{-\lambda M}}{e^{-\lambda\alpha} - e^{-\lambda\beta}}.$$

Part (c)

We implemented the normalised signal, background and total PDFs as found in part (b) using the `scipy.stats` package. We then integrated the total PDF over $[5, 5.6]$ for 1000 random combinations of the parameters, $\boldsymbol{\theta}$, and all integrals came out to unity as expected.

Part (d)

We created two plots visualising the distributions with the true parameters. Fig. 1 includes the weights f and $(1-f)$ on the signal and background distributions. Fig. 2 shows the distributions properly normalised (ie. without the weights).

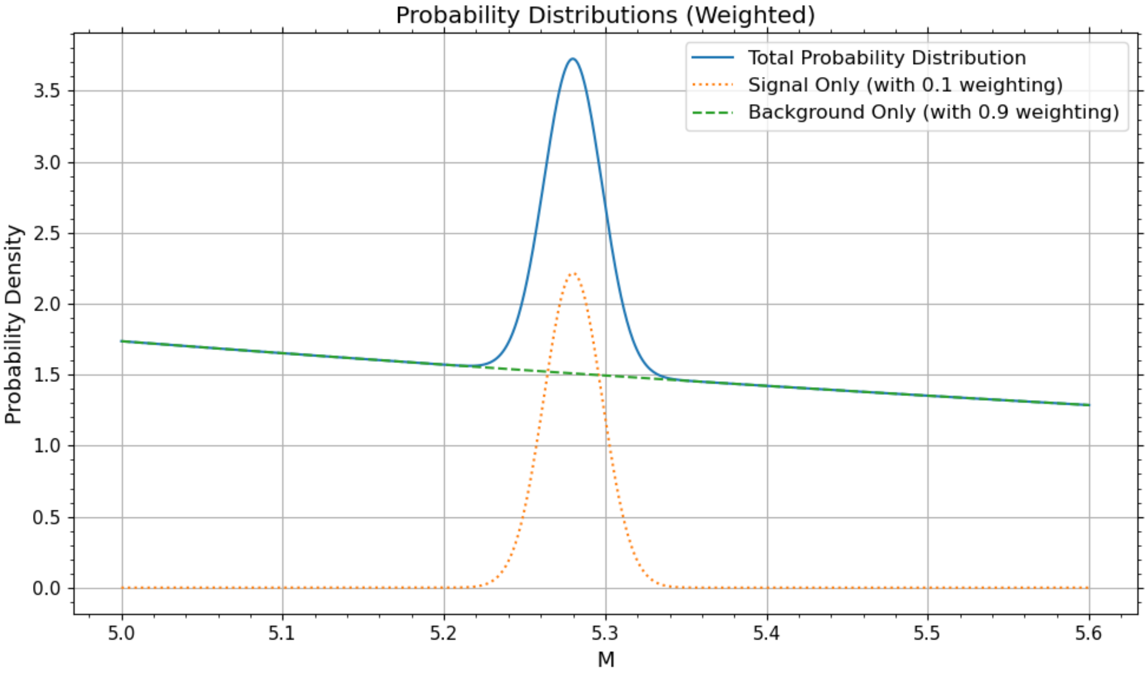

Figure 1: Visualisation of the total, signal and background PDFs with the weights applied to the signal and background models.

Part (e)

To generate samples from the total PDF, we implemented the following algorithm: for each event, choose whether to use the signal-only or background-only distribution with probability 0.1 and 0.9, then generate the event using the inverse CDF method, which works by taking uniform random numbers in [0, 1] and passing them into the percentage point function, PPF, to generate correctly-distributed events. This method is very efficient as there are no wasted samples. We used the PPFs of the normal and exponential distributions which are not normalised over $[\alpha, \beta]$, hence rather than passing uniform random numbers from the interval [0, 1] into the PPFs, we used the interval $[F(\alpha), F(\beta)]$, where F is the CDF of the distribution we are generating from. This guarantees that data will only be generated in the desired range, and it will be distributed by the correctly normalised PDF.

We generated a sample of 100K events, then fitted the total PDF to this data using maximum likelihood estimation. That is, we estimated the parameters by minimising the negative log likelihood:

$$l = -ln\left(\prod_i p(M_i;\boldsymbol{\theta})\right) = -\sum_i ln(p(M_i;\boldsymbol{\theta})). \tag{3}$$

In the limit of large samples ($N \to \infty$), maximum likelihood estimates are consistent, unbiased and normally distributed, and their variance is given by the minimum variance bound:

$$V(\hat{\theta}) = \left(NE\left[\frac{\partial^2 l}{\partial \theta^2}\right]\right)^{-1}, \tag{4}$$

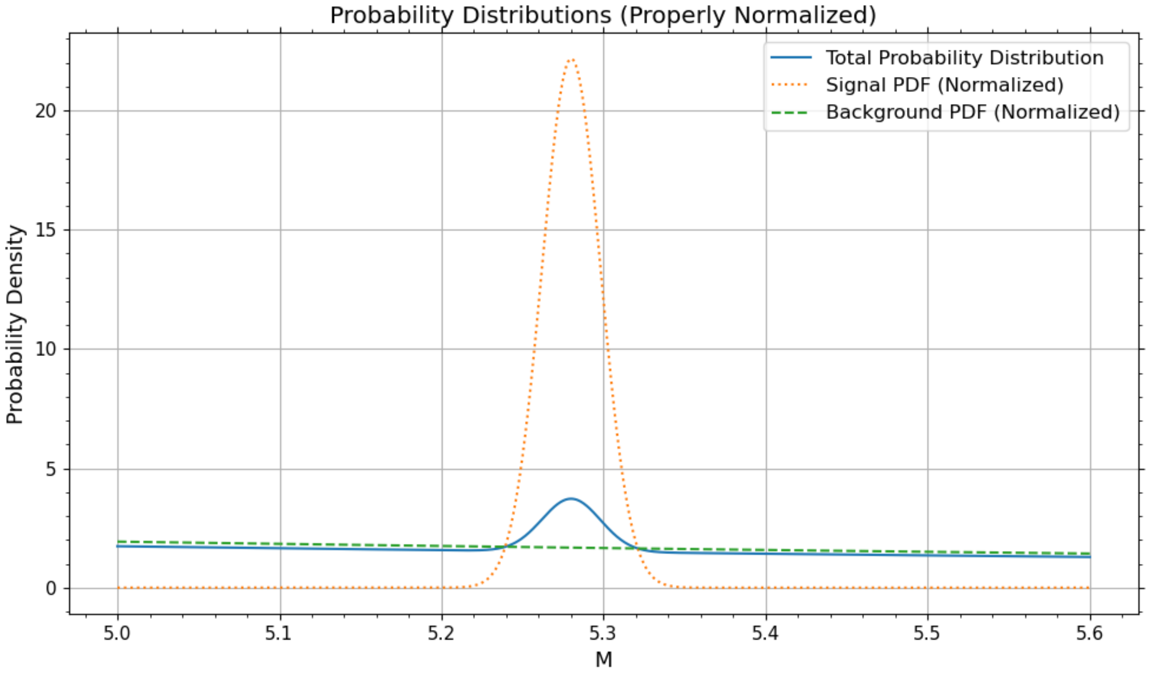

Figure 2: Visualisation of the total, signal and background PDFs, with each distribution properly normalised.

where n is the sample size, $E[\cdot]$ is the expectation operator, l is the negative log likelihood. Since we are dealing with a sample of 100K events, this asymptotic limit is a reasonable assumption. We obtained the estimates using `iminuit`, which calculates the uncertainties using equation (4) and the property that $E\left[\frac{\partial^2 l}{\partial\theta^2}\right] = \left[\frac{\partial^2 l}{\partial\theta^2}\right]_{\theta=\hat{\theta}}$.

$$\hat{f} = 0.0998 \pm 0.0016, \quad \hat{\lambda} = 0.470 \pm 0.019, \quad \hat{\mu} = 5.27980 \pm 0.00033, \quad \hat{\sigma} = 17.99 \times 10^{-3} \pm 0.32 \times 10^{-3}.$$

For all estimates, the true values of the parameters lie within the uncertainties of the estimates. We then binned and plotted the sample, and overlaid the fitted PDF, as can be seen in Fig. 3. The fitted PDF clearly matches the data very well. Since bin counts are Poisson distributed, their uncertainty estimates are given by the square root of the bin count (then scaled for bin density).

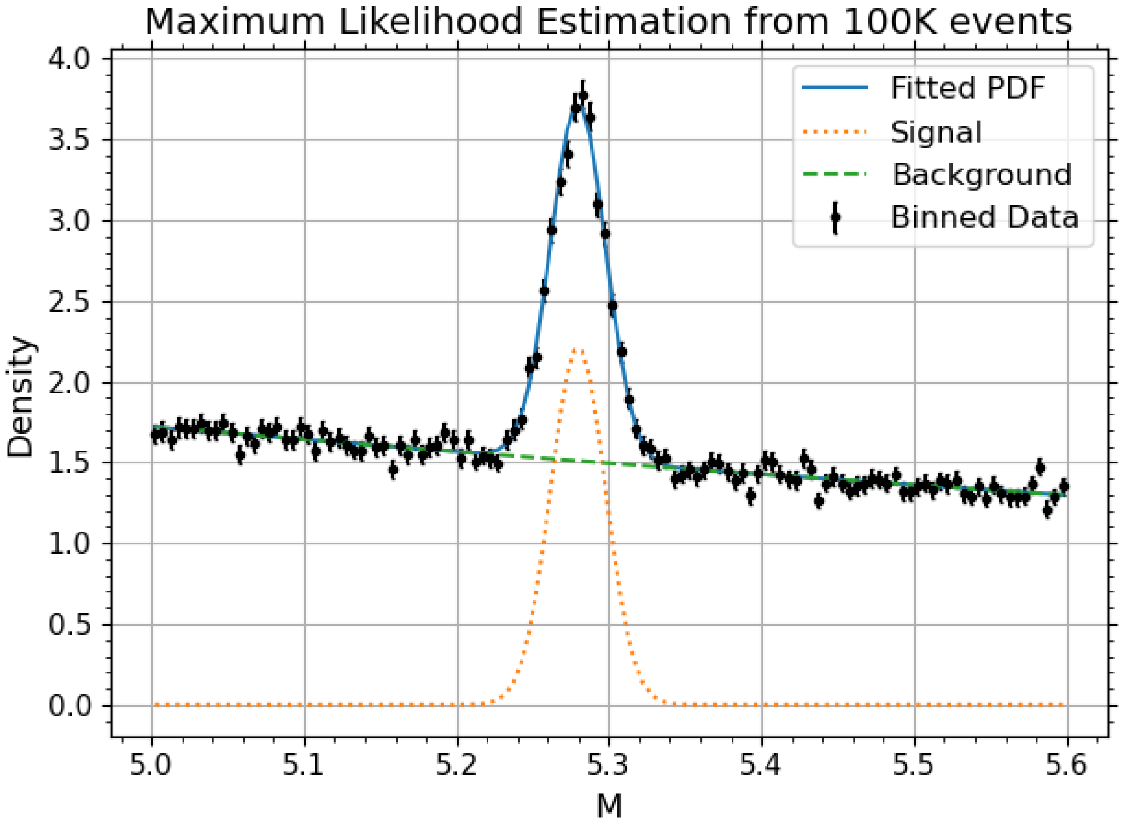

Figure 3: Visualisation of the our 100K sample generated from the true PDF, overlaid by the fitted PDFs of the total, signal and background distributions.

Section B

Introduction

Section B consists of parts (f) and (g). The goal of these parts was to find the critical sizes of the dataset needed to 'discover' (f) a signal and (g) two distinct signals at least 90% of the time. To do this, we performed Neyman-Pearson hypothesis tests on many example datasets, which allowed us to estimate the probability of discovery, p, at a range of different sizes, N. We then fitted predictive models to this data to estimate the critical dataset size, and used a Monte Carlo simulation to estimate its uncertainty.

Methodology

Neyman-Pearson Test

The power, $1-\beta$, of a hypothesis test is the probability of correctly rejecting the null hypothesis when it is false. The Neyman-Pearson Lemma states that the test statistic which maximises the power is given by the ratio of the likelihoods under each hypothesis:

$$T = -2ln\left(\frac{L(\vec{X}|H_0)}{L(\vec{X}|H_1)}\right). \tag{5}$$

According to Wilk's theorem, in the limit of large sample sizes, this test statistic under the null hypothesis is χ^2 distributed with 1 degree of freedom. This allows us to find the p value of the hypothesis test via:

$$p = 1 - F_{\chi^2,1}(T), \tag{6}$$

where $F_{\chi^2,1}$ is the CDF of the χ^2 distribution with 1 degree of freedom. Since we cannot have a 'negative' presence of signal ($f \geq 0$), this is a one-sided test, and we calculate the significance via the formula:

$$Z = \Phi^{-1}(1-p), \tag{7}$$

where Φ is the CDF of the normal distribution, and p is the p-value of the test. The threshold for a 'discovery' is $Z \geq 5$

Binomial Distribution

On each dataset size, N_i, we conduct hypothesis tests on n generated datasets and find k_i discoveries. Each hypothesis test has two possible outcomes (discovery or no discovery) and the probability of discovery is constant for constant dataset size, hence the number of discoveries is binomially distributed. Let p_i be the true probability of discovery for dataset size N_i. The mean of the binomial distribution is $\mu = E[k_i] = p_i n$ and the variance is $V(k_i) = np_i(1-p_i)$.

We can estimate the probability of discovery as the frequency of discovery in the experiment:

$$\hat{p}_i = \frac{k_i}{n}. \tag{8}$$

The variance of this estimate is $V(\hat{p}_i) = V(\frac{k_i}{n}) = \frac{V(k_i)}{n^2} = \frac{p_i(1-p_i)}{n}$, hence we can estimate the uncertainty using:

$$\hat{\sigma_i} = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{n}}. \tag{9}$$

The estimate $\hat{p}_i$ is consistent since the law of large numbers states that the frequency converges to the true probability in the asymptotic limit: $\hat{p}_i = \frac{k_i}{n} \rightarrow p_i$ as $n \rightarrow \infty$. It is also unbiased since the expectation is $E[\hat{p}_i] = E[\frac{k_i}{n}] = \frac{E[k_i]}{n} = \frac{\mu}{n} = p_i$. Since $\hat{p}_i$ is consistent, we see that as $n \rightarrow \infty$, $\hat{\sigma_i} = \sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{n}} \rightarrow \sqrt{\frac{p_i(1-p_i)}{n}} = \sigma_i$, hence $\hat{\sigma_i}$ is also consistent.

According to the central limit theorem, our estimate $\hat{p}_i = \frac{k_i}{n}$ is normally distributed in the limit of large n.

Least Squares Fitting

We can fit some model, $f(N;\boldsymbol{\theta})$, to the dataset $\{(N_i, \hat{p}_i \pm \hat{\sigma_i})\}$, by minimising the 'chi-squared' with respect to $\boldsymbol{\theta}$, given by:

$$\chi^2 = \sum_i \frac{(\hat{p}_i - f(N_i|\boldsymbol{\theta}))^2}{\hat{\sigma_i}^2}. \tag{10}$$

Assuming $\hat{p}_i$ is normally distributed with mean $f(N_i|\hat{\boldsymbol{\theta}})$ and standard deviation $\hat{\sigma_i}$, then χ^2 is proportional to the negative log likelihood (3) and least squares fitting is equivalent to maximum likelihood estimation. As discussed in part (e), maximum likelihood estimates are maximally efficient in the asymptotic limit, so we can estimate the errors of the fitted model parameters using the minimum variance bound (4). Additionally, if the Gaussian assumption is valid, then the quoted uncertainties represent 1 standard deviation, and the fitted model should make predictions that lie within these uncertainties 68.3% of the time.

χ^2 test

Assuming $\hat{p}_i$ is normally distributed with mean $f(N_i|\hat{\boldsymbol{\theta}})$ and standard deviation $\hat{\sigma_i}$, then χ^2 (10) is χ^2 distributed with $k = n_{obs} - n_{params}$ degrees of freedom, where n_{obs} is the number of observations and n_{params} is the number of parameters of the fitted model. The expectation value of the χ^2 distribution is k, hence we expect $\chi^2/d.o.f \approx 1$. The p-value for this test is the probability that we obtain a χ^2 as large or larger than the value we measured. This can obtained using the cumulative density function:

$$p = 1 - F_{\chi^2,k}(\chi^2). \tag{11}$$

The higher the p-value, the better the fitted model agrees with the data. Conventionally, a p-value of less than 5% is considered statistically significant to suggest the model is a poor fit. A p-value which is too high may indicate that the model has over-fitted to the data.

Monte Carlo Simulation

Once we have a fitted model $f(N|\hat{\boldsymbol{\theta}})$ with parameter estimates $\hat{\boldsymbol{\theta}}$ and uncertainties on those parameter estimates, we can estimate the critical dataset size by finding N_{90} for which $f(N_{90}|\hat{\boldsymbol{\theta}}) = 0.9$. We can estimate the uncertainty of N_{90} by repeatedly sampling the parameters θ_i from a normal distribution (with mean as the parameter estimate, and standard deviation as the error), then solving for N_{90}. This gives us a sample of measurements of N_{90}, from which we can estimate the error on N_{90} as the sample standard deviation. This analysis is only valid when the parameter estimates are normally distributed, which is true for maximum likelihood estimates in the asymptotic limit.

Part (f)

We determined the range of dataset sizes over which to conduct the experiment via a trial and error approach of repeatedly generating datasets for a range of sizes, performing hypothesis tests, and using the outcomes to estimate the probability of discovery. We chose the range $N \in [500, 900]$ as it appeared to contain the true N_{90}, and it is narrow enough to make a precise measurement N_{90}. For 50 equally-spaced sizes in this range, we generated $n = 500$ example datasets.

For each dataset, we constructed a null (H_0) and alternate (H_1) hypothesis by fitting a 'background-only model' (for H_0) and a 'signal plus background' model (for H_1) to the dataset using maximum likelihood estimation (as described in part (e)). We conducted a Neyman-Pearson test (5) as it is the most powerful test statistic. The outcomes of the tests allowed us to estimate the probability of discovery, $\hat{p}_i$, for each size, N_i, and the corresponding uncertainty, $\hat{\sigma_i}$, using equations (8) and (9) respectively.

We then fitted a third order polynomial, $f(N)$, to the dataset $\{(N_i, \hat{p}_i \pm \hat{\sigma_i}) | i = 1, ..., 50\}$ using least squares fitting (10). We chose a third order polynomial as it is sufficient to approximate the relationship between p and N within the range of interest. The uncertainties on the fitted parameters were estimated using the minimum variance bound (4). As discussed in the methodology section, this is valid under the assumptions that $\hat{p}_i$ is normally distributed (which is valid when n is large) and the number of data-points is large, since then least squares is equivalent to maximum likelihood estimation and the asymptotic limit holds. Since $n = 500$ and we have only 50 data-points, it is not clear that these assumptions are correct. To assess the goodness-of-fit, we calculated that coverage = 70% and performed a χ^2 test, in which we calculated $\chi^2/d.o.f = 1.15$ and p-value = 21.9%. The coverage is very close 68.3% and the $\chi^2/d.o.f$ is close to 1, indicating that the model strongly agrees with the data, and our Gaussian assumption on $\hat{p}_i$ is reasonable. The p value is high enough that we can accept the fitted polynomial as the true relationship between p and N in the range of interest, but not so high as to be suspicious.

In Fig. 4, we plotted the probability of discovery against the size of the dataset, with the fitted model overlaid. Below this plot, there is a plot of the pulls, which are given by:

$$\text{pull} = \frac{\hat{p}_i - f(N_i)}{\hat{\sigma_i}}, \tag{12}$$

where f is the fitted polynomial. If it is true that $\hat{p}_i$ is normally distributed with mean $f(N_i)$ and standard deviation $\hat{\sigma_i}$, then the pulls are distributed by the standard normal distribution. On the right hand side of the plot is a density histogram of the pulls, with a standard normal distribution overlaid (the red line). The density histogram appears to agree fairly well with the standard normal distribution, reinforcing the validity of the fitted model and the Gaussian assumption.

We estimated N_{90} and its uncertainty using the Monte Carlo simulation discussed in the methodology section, with 1000 parameter samples.

$$N_{90} = 603 \pm 3.$$

However, the error on this result may not be reliable, as it was derived by taking the fitted parameter uncertainties to be the minimum variance bound. Since this is only valid in the asymptotic limit and our sample size was only 50, it is not clear that this assumption is reasonable.

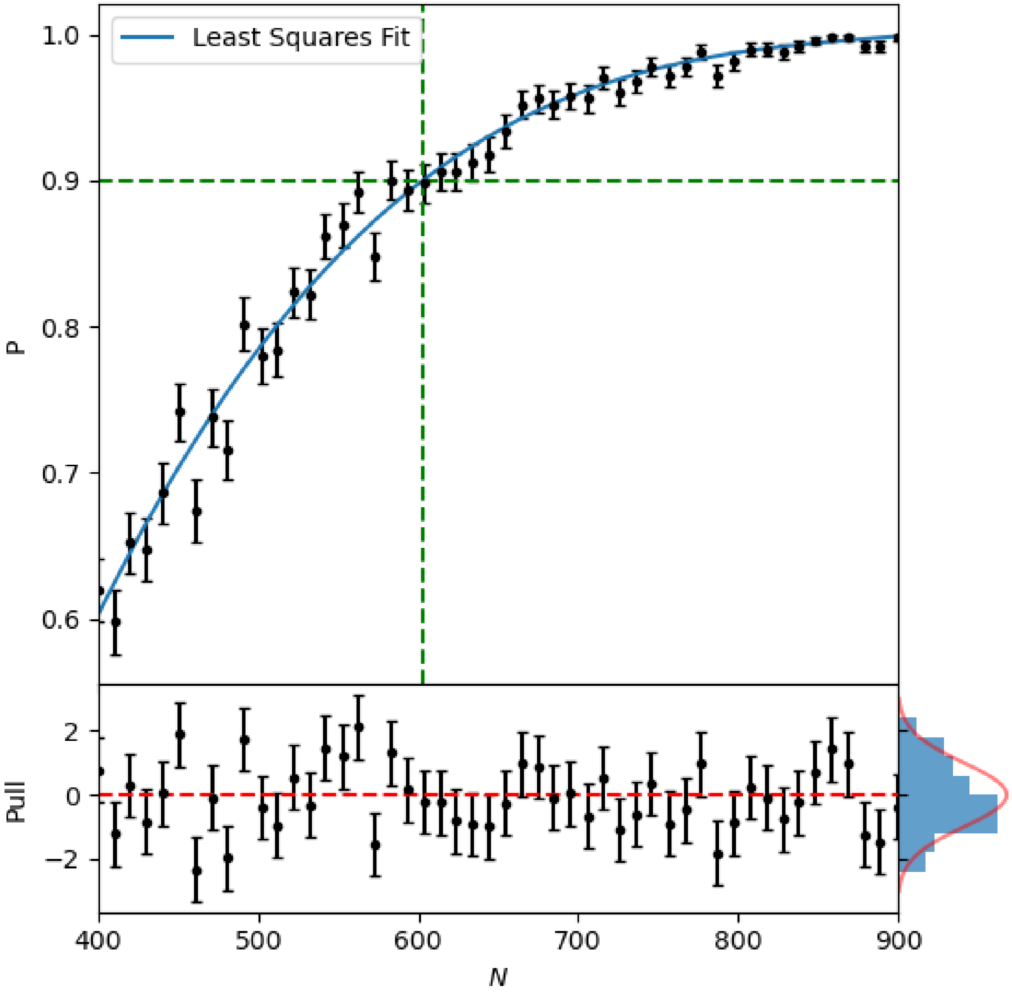

Figure 4: Plot of probability of discovering a signal vs the size of the dataset. The critical dataset size (90% discovery rate) is indicated by green axis lines. The bottom plot shows the pulls, with a density histogram plotted on the right hand edge and the standard normal distribution overlaid.

Part (g)

Using the same trial and error procedure as in part (f), we decided to conduct the experiment on 50 dataset sizes, $\{N_i\}$, in the range $[1500, 3000]$, and generated $n = 300$ example datasets at each size.

For each dataset, we conducted a hypothesis test for the presence of two distinct signals using the Neyman-Pearson test statistic (equation (5)), yielding k_i discoveries. The null and alternate hypotheses were constructed by fitting a 'signal plus background' model (H_0) and a 'two signals plus background' model (H_1) to the dataset using maximum likelihood estimation. We estimated the probability of discovery with equation (8) and its uncertainty with equation (9).

We then fitted a third order polynomial, $f(N)$, to the dataset $\{(N_i, \hat{p}_i \pm \hat{\sigma_i})\}$ using least squares fitting (10). Similarly to part (f), we calculated the coverage and performing a χ^2 test. The coverage was 60%, which is fairly close to the target of 68.3%, indicating that the Gaussian assumption on $\hat{p}_i$ is reasonable, and the fitted model agrees with the data. Furthermore, the χ^2 test revealed $\chi^2/d.o.f = 1.197$ and p-value = 16.96%. These metrics indicate that the model is a good fit of the data, since $\chi^2/d.o.f$ is close to 1 and the p value is high enough that we can say the model agrees with the data. In Fig. 5 we plot the probability of discovery vs size of dataset with the fitted model overlaid, and the pulls (equation (12)) visualised directly below. We can see that the density histogram of the pulls agrees well with the standard normal distribution, which suggests indicates agreement between the data and the model.

We estimated the dataset size to discover two distinct signals 90% of the time, N_{90}, and it's uncertainty using the Monte Carlo simulation method:

$$N_{90} = 2570 \pm 35.$$

Similarly to part (f), the uncertainty on this result may not be trustworthy, as it assumed the predictive model's parameter estimates were maximally efficient (4), which is only valid with asymptotic sample sizes. Since we had only 50 data-points, this assumption may not be correct.

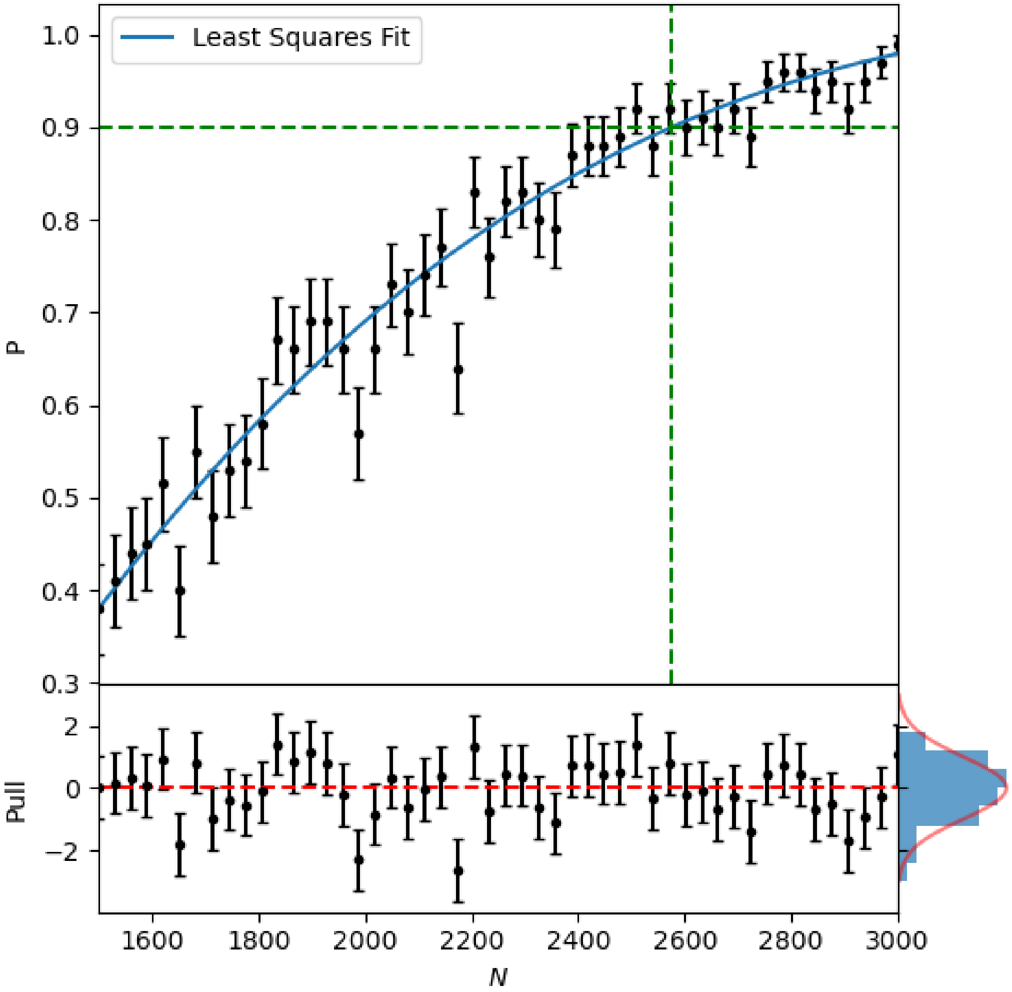

Figure 5: Plot of probability of discovering two distinct signals vs the size of the dataset. The critical dataset size (90% discovery rate) is indicated by green axis lines. The bottom plot shows the pulls, with a density histogram plotted on the right hand edge and the standard normal distribution overlaid.

Appendix

A README

This repository contains all code necessary to reproduce the results discussed in `report/main.pdf`.

A.0.1 Usage

To generate the docker image and run the container, navigate to the root directory and use:

```
$ docker build -t <image_name> .
$ docker run -ti <image_name>
```

To replicate the results and/or plots from parts c, d, e, f and g, run the following scripts:

```
$ python src/solve_part_c.py
$ python src/solve_part_d.py
$ python src/solve_part_e.py
$ python src/solve_part_f.py
$ python src/solve_part_g.py
```

Results will be printed to terminal and plots will be saved in `plots/`

A.0.2 Timing

I ran all scripts on my personal laptop with the following specifications:

- Chip: Apple M1 Pro
- Total Number of Cores: 8 (6 performance and 2 efficiency)
- Memory (RAM): 16 GB
- Operating System: macOS Sonoma v14.0

The scripts `solve_part_c.py`, `solve_part_d.py`, `solve_part_e.py` all ran in less than 10 seconds.

`solve_part_f.py` took roughly 15 minutes to run.

`solve_part_g.py` took roughly 50 minutes to run.

A.0.3 Project Structure

All code is contained inside the `src/` folder. The `solve_part_*` scripts are those which can be used to replicate the results in the report. The other modules in the `src/` folder contain functionality to assist the scripts.

- `NP_analysis.py` - Contains functions to assist with the analysis of the 'probability of discovery vs dataset size' data in parts (f) and (g)
- `discovery.py` - Contains function to calculate the probability of discovery for a given dataset size.
- `distributions.py` - Contains all probability density functions and cumulative density functions relevant to the project.
- `generation.py` - Contains functionality to generate datasets (from the 'signal plus background' model and the 'two signals plus background' model).
- `hypothesis_test.py` - Contains functionality to perform hypothesis tests on datasets.