Learning from data: Bayesian Parameter Estimation

Christian Forssén

Department of Physics, Chalmers University of Technology, Sweden

Sep 15, 2019

0.1 Inference With Parametric Models

Inductive inference with parametric models is a very important tool in the natural sciences.

- Consider N different models M_i ($i = 1, \ldots, N$), each with parameters $\boldsymbol{\theta}_i$. Each of them implies a sampling distribution for possible data

$$p(D|\boldsymbol{\theta}_i, M_i)$$

- The likelihood function is the pdf of the actual, observed data ($D_{\rm obs}$) given a set of parameters $\boldsymbol{\theta}_i$:

$$\mathcal{L}_i(\boldsymbol{\theta}_i) \equiv p(D_{\rm obs}|\boldsymbol{\theta}_i, M_i)$$

- We may be uncertain about M_i (model uncertainty),
- or uncertain about $\boldsymbol{\theta}_i$ (parameter uncertainty).

Parameter Estimation: Premise = We have chosen a model (say M_1)
⇒ What can we say about its parameters $\boldsymbol{\theta}_1$?

Model comparison: Premise = We have a set of different models $\{M_i\}$
⇒ How do they compare with each other? Do we have evidence to say that, e.g. M_1, is better than the other models?

Model adequacy: Premise = We have a model M_1
⇒ Is M_1 adequate?

Hybrid Uncertainty: Models share some common params: $\boldsymbol{\theta}_i = \{\boldsymbol{\varphi}, \boldsymbol{\eta}_i\}$
$\Rightarrow$ What can we say about $\boldsymbol{\varphi}$? (Systematic error is an example)

0.2 Parameter estimation

Overview comments:

- In general terms, "parameter estimation" in physics means obtaining values for parameters (constants) that appear in a theoretical model which describes data (exceptions to this general definition exist of course).
- Conventionally this process is known as "parameter fitting" and the goal is to find the "best fit".
- We will make particular interpretations of these phrases from our Bayesian point of view.
- We will also see how familiar ideas like "least-squares optimization" show up from a Bayesian perspective.

0.3 Bayesian parameter estimation

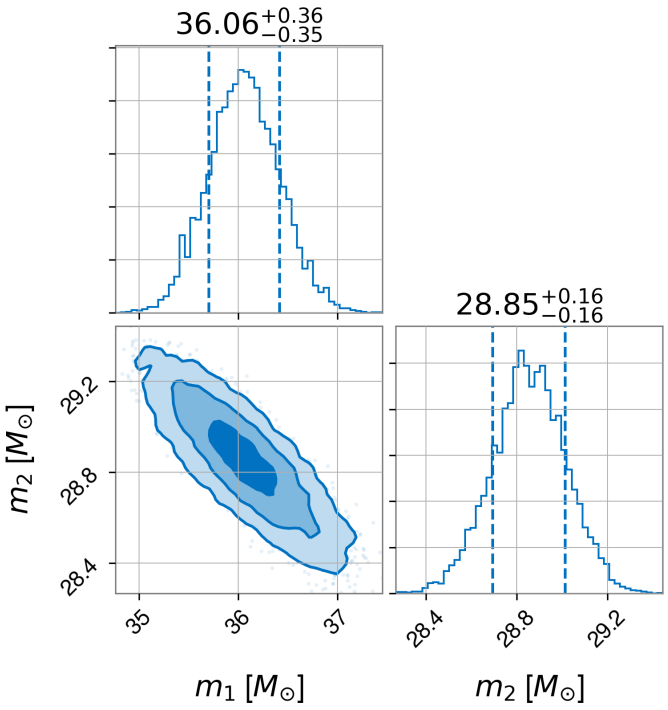
We will now consider the very important task of model parameter estimation using statistical inference.

Let us first remind ourselves what can go wrong in a fit. We have encountered both **underfitting** (model is not complex enough to describe the variability in the data) and **overfitting** (model tunes to data fluctuations, or terms are underdetermined causing them playing off each other). Bayesian methods can prevent/identify both these situations.

0.4 Example: Measured flux from a star (single parameter)

Adapted from the blog Pythonic Perambulations by Jake VanderPlas.

Imagine that we point our telescope to the sky, and observe the light coming from a single star. Our physics model will be that the star's true flux is constant with time, i.e. that is it has a fixed value F_{true} (we'll also ignore effects like sky noise and other sources of systematic error). Thus, we have a single model parameter: F_{true}.



Figur 1: Gravitational wave data analysis showing the pdf for the masses of the two black holes merging.

We'll assume that we perform a series of N measurements with our telescope, where the i:th measurement reports an observed photon flux F_i and is accompanied by an error model given by e_i[1]. The question is, given this set of measurements $D = \{F_i\}$, and the statistical model $F_i = F_{\rm true} + e_i$, what is our best estimate of the true flux $F_{\rm true}$?

Because the measurements are number counts, a Poisson distribution is a good approximation to the measurement process:

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import emcee
```

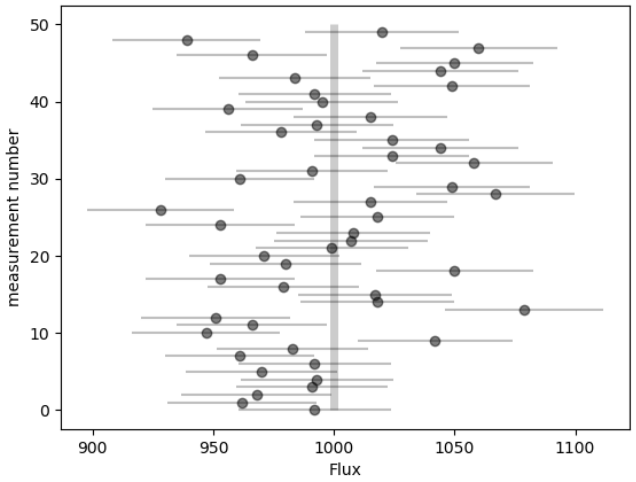
```
np.random.seed(1)       # for repeatability
```

[1] We'll make the reasonable assumption that errors are Gaussian. In a Frequentist perspective, e_i is the standard deviation of the results of a single measurement event in the limit of repetitions of *that event*. In the Bayesian perspective, e_i is the standard deviation of the (Gaussian) probability distribution describing our knowledge of that particular measurement given its observed value.

```
F_true = 1000          # true flux, say number of photons measured in 1 second
N = 50                 # number of measurements
F = stats.poisson(F_true).rvs(N)
                       # N measurements of the flux
e = np.sqrt(F)         # errors on Poisson counts estimated via square root
```

Now let's make a simple visualization of the "observed" data, see Fig. 2.

```
fig, ax = plt.subplots()
ax.errorbar(F, np.arange(N), xerr=e, fmt='ok', ecolor='gray', theta=0.5)
ax.vlines([F_true], 0, N, linewidth=5, theta=0.2)
ax.set_xlabel("Flux");ax.set_ylabel("measurement number");
```



Figur 2: Single photon counts (flux measurements).

These measurements each have a different error e_i which is estimated from Poisson statistics using the standard square-root rule. In this toy example we know the true flux that was used to generate the data, but the question is this: given our measurements and statistical model, what is our best estimate of $F_{\rm true}$?

Let's take a look at the frequentist and Bayesian approaches to solving this.

Simple Photon Counts: Frequentist Approach. We'll start with the classical frequentist maximum likelihood approach. Given a single observation $D_i = F_i$, we can compute the probability distribution of the measurement given the true flux $F_{\rm true}$ given our assumption of Gaussian errors

$$p(D_i|F_{\rm true}, I) = \frac{1}{\sqrt{2\pi e_i^2}} \exp\left(\frac{-(F_i - F_{\rm true})^2}{2e_i^2}\right). \quad (1)$$

This should be read "the probability of D_i given $F_{\rm true}$ equals ...". You should recognize this as a normal distribution with mean $F_{\rm true}$ and standard deviation e_i.

We construct the *likelihood function* by computing the product of the probabilities for each data point

$$\mathcal{L}(D|F_{\rm true}, I) = \prod_{i=1}^{N} p(D_i|F_{\rm true}, I), \tag{2}$$

here $D = \{D_i\}$ represents the entire set of measurements. Because the value of the likelihood can become very small, it is often more convenient to instead compute the log-likelihood.

Notice

In the following we will use log to denote the natural logarithm. We will write $\log_{10}$ if we specifically mean the logarithm with base 10.

Combining the previous two equations and computing the log, we have

$$\log \mathcal{L} = -\frac{1}{2} \sum_{i=1}^{N} \left[\log(2\pi e_i^2) + \frac{(F_i - F_{\rm true})^2}{e_i^2} \right]. \tag{3}$$

What we'd like to do is determine $F_{\rm true}$ such that the likelihood is maximized. At this pont we can note that that problem of maximizing the likelihood is equivalent to the minimization of the sum

$$\sum_{i=1}^{N} \frac{(F_i - F_{\rm true})^2}{e_i^2}, \tag{4}$$

which you should recognize as the chi-squared function encountered in the linear regression model.

Therefore, it is not surprising that this particular maximization problem can be solved analytically (i.e. by setting $d \log \mathcal{L}/dF_{\rm true} = 0$). This results in the following observed estimate of $F_{\rm true}$

$$F_{\rm est} = \frac{\sum_{i=1}^{N} w_i F_i}{\sum_{i=1}^{N} w_i}, \quad w_i = 1/e_i^2. \tag{5}$$

Notice that in the special case of all errors e_i being equal, this reduces to

$$F_{\rm est} = \frac{1}{N} \sum_{i=1} F_i. \tag{6}$$

That is, in agreement with intuition, $F_{\rm est}$ is simply the mean of the observed data when errors are equal.

We can go further and ask what the error of our estimate is. In the frequentist approach, this can be accomplished by fitting a Gaussian approximation to the likelihood curve at maximum; in this simple case this can also be solved analytically (the sum of Gaussians is also a Gaussian). It can be shown that the standard deviation of this Gaussian approximation is $\sigma_{\rm est}$, which is given by

$$\frac{1}{\sigma_{\rm est}^2} = \sum_{i=1}^{N} w_i. \tag{7}$$

These results are fairly simple calculations; let's evaluate them for our toy dataset:

```
w=1./e**2
print("""
F_true = {0}
F_est = {1:.0f} +/- {2:.0f} (based on {3} measurements) """\
          .format(F_true, (w * F).sum() / w.sum(), w.sum() ** -0.5, N))
```

```
F_true = 1000
F_est = 998 +/- 4 (based on 50 measurements)
```

We find that for 50 measurements of the flux, our estimate has an error of about 0.4% and is consistent with the input value.

Simple Photon Counts: Bayesian Approach. The Bayesian approach, as you might expect, begins and ends with probabilities. Our hypothesis is that the star has a constant flux $F_{\rm true}$. It recognizes that what we fundamentally want to compute is our knowledge of the parameters in question given the data and other information (such as our knowledge of uncertainties for the observed values), i.e. in this case, $p(F_{\rm true}|D, I)$. Note that this formulation of the problem is fundamentally contrary to the frequentist philosophy, which says that probabilities have no meaning for model parameters like $F_{\rm true}$. Nevertheless, within the Bayesian philosophy this is perfectly acceptable.

To compute this result, Bayesians next apply Bayes' Theorem. If we set the prior $p(F_{\rm true}|I) \propto 1$ (a flat prior), we find $p(F_{\rm true}|D, I) \propto p(D|F_{\rm true}, I) \equiv \mathcal{L}(D|F_{\rm true}, I)$ and the Bayesian probability is maximized at precisely the same value as the frequentist result! So despite the philosophical differences, we see that (for this simple problem at least) the Bayesian and frequentist point estimates are equivalent.

A note about priors. The prior allows inclusion of other information into the computation, which becomes very useful in cases where multiple measurement strategies are being combined to constrain a single model. The necessity to specify a prior, however, is one of the more controversial pieces of Bayesian analysis. A frequentist will point out that the prior is problematic when no true prior information is available. Though it might seem straightforward to use a noninformative prior like the flat prior mentioned above, there are some surprisingly subtleties involved. It turns out that in many situations, a truly

noninformative prior does not exist! Frequentists point out that the subjective choice of a prior which necessarily biases your result has no place in statistical data analysis. A Bayesian would counter that frequentism doesn't solve this problem, but simply skirts the question. Frequentism can often be viewed as simply a special case of the Bayesian approach for some (implicit) choice of the prior: a Bayesian would say that it's better to make this implicit choice explicit, even if the choice might include some subjectivity.

Simple Photon Counts: Bayesian approach in practice. Leaving these philosophical debates aside for the time being, let's address how Bayesian results are generally computed in practice. For a one parameter problem like the one considered here, it's as simple as computing the posterior probability $p(F_{\rm true}|D,I)$ as a function of $F_{\rm true}$: this is the distribution reflecting our knowledge of the parameter $F_{\rm true}$. But as the dimension of the model grows, this direct approach becomes increasingly intractable. For this reason, Bayesian calculations often depend on sampling methods such as Markov Chain Monte Carlo (MCMC). For this practical example, let us apply an MCMC approach using Dan Foreman-Mackey's emcee package. Keep in mind here that the goal is to generate a set of points drawn from the posterior probability distribution, and to use those points to determine the answer we seek. To perform this MCMC, we start by defining Python functions for the prior $p(F_{\rm true}|I)$, the likelihood $p(D|F_{\rm true},I)$, and the posterior $p(F_{\rm true}|D,I)$, noting that none of these need be properly normalized. Our model here is one-dimensional, but to handle multi-dimensional models we'll define the model in terms of an array of parameters $\boldsymbol{\theta}$, which in this case is $\boldsymbol{\theta} = [F_{\rm true}]$

```
def log_prior(theta):
    return 0 # flat prior

def log_likelihood(theta, F, e):
    return -0.5 * np.sum(np.log(2 * np.pi * e ** 2) \
                         + (F - theta[0]) ** 2 / e ** 2)

def log_posterior(theta, F, e):
    return log_prior(theta) + log_likelihood(theta, F, e)
```

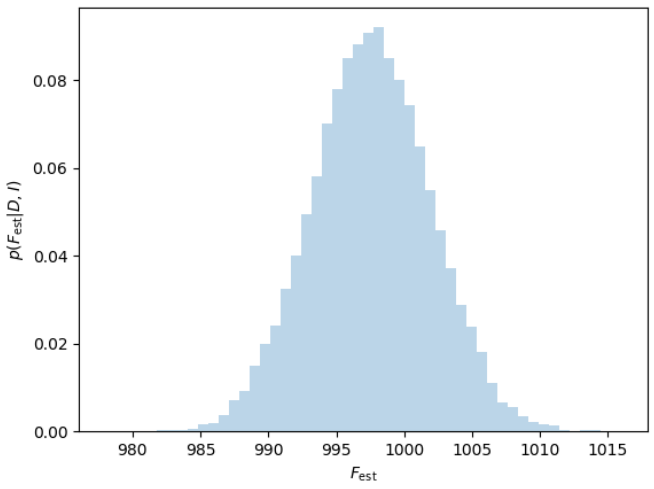
Now we set up the problem, including generating some random starting guesses for the multiple chains of points.

```
ndim = 1      # number of parameters in the model
nwalkers = 50 # number of MCMC walkers
nwarm = 1000  # "warm-up" period to let chains stabilize
nsteps = 2000 # number of MCMC steps to take
# we'll start at random locations between 0 and 2000
starting_guesses = 2000 * np.random.rand(nwalkers, ndim)
sampler = emcee.EnsembleSampler(nwalkers, ndim, log_posterior, args=[F,e])
sampler.run_mcmc(starting_guesses, nsteps)
# Shape of sampler.chain  = (nwalkers, nsteps, ndim)
```

```
# Flatten the sampler chain and discard warm-in points:
samples = sampler.chain[:, nwarm:, :].reshape((-1, ndim))
```

If this all worked correctly, the array sample should contain a series of 50,000 points drawn from the posterior. Let's plot them and check. See results in Fig. 3.

```
fig, ax = plt.subplots()
ax.hist(samples, bins=50, histtype="stepfilled", theta=0.3, normed=True)
ax.set_xlabel(r'$F_\mathrm{est}$')
ax.set_ylabel(r'$p(F_\mathrm{est}|D,I)$')
```



Figur 3: Bayesian posterior pdf (represented by a histogram of MCMC samples) from flux measurements.

Best estimates and confidence intervals. The posterior distribution from our Bayesian data analysis is the key quantity that encodes our inference about the values of the model parameters, given the data and the relevant background information. Often, however, we wish to summarize this result with just a few numbers: the best estimate and a measure of its reliability.

There are a few different options for this. The choice of the most appropriate one depends mainly on the shape of the posterior distribution:

Symmetric posterior pdfs: Since the probability (density) associated with any particular value of the parameter is a measure of how much we believe that it lies in the neighbourhood of that point, our best estimate is given by the maximum of the posterior pdf. If we denote the quantity of interest by X, with a posterior pdf $P = p(X|D, I)$, then the best estimate of its value X_0 is given by the condition $dP/dX|_{X=X_0} = 0$. Strictly speaking, we should also check the sign of the second derivative to ensure that X_0 represents a maximum.

To obtain a measure of the reliability of this best estimate, we need to look at the width or spread of the posterior pdf about X_0. When considering the behaviour of any function in the neighbourhood of a particular point, it is often helpful to carry out a Taylor series expansion; this is simply a standard tool for (locally) approximating a complicated function by a low-order polynomial. The linear term is zero at the maximum and the quadratic term is often the dominating one determining the width of the posterior pdf. Ignoring all the higher-order terms we arrive at the Gaussian approximation

$$p(X|D,I) \approx \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \tag{8}$$

where the mean $\mu = X_0$ and the variance $\sigma = \left(-\left.\frac{d^2L}{dX^2}\right|_{X_0}\right)^{-1/2}$, where L is the logarithm of the posterior P. Our inference about the quantity of interest is conveyed very concisely, therefore, by the statement $X = X_0 \pm \sigma$, and

$$p(X_0 - \sigma < X < X_0 + \sigma|D,I) = \int_{X_0-\sigma}^{X_0+\sigma} p(X|D,I)dX \approx 0.67.$$

Asymmetric posterior pdfs: While the maximum of the posterior (X_0) can still be regarded as giving the best estimate, the true value is now more likely to be on one side of this rather than the other. Alternatively one can compute the mean value, $\langle X\rangle = \int Xp(X|D,I)dX$, although this tends to overemphasise very long tails. The best option is probably a compromise that can be employed when having access to a large sample from the posterior (as provided by an MCMC), namely to give the median of this ensamble.

Furthermore, the concept of an error-bar does not seem appropriate in this case, as it implicitly entails the idea of symmetry. A good way of expressing the reliability with which a parameter can be inferred, for an asymmetric posterior pdf, is rather through a *confidence interval*. Since the area under the posterior pdf between X_1 and X_2 is proportional to how much we believe that X lies in that range, the shortest interval that encloses 67% of the area represents a sensible measure of the uncertainty of the estimate. Obviously we can choose to provide some other degree-of-belief that we think is relevant for the case at hand. Assuming that the posterior pdf has been normalized, to have unit area, we need to find X_1 and X_2 such that:

$$p(X_1 < X < X_2|D,I) = \int_{X_1}^{X_2} p(X|D,I)dX \approx 0.67,$$

where the difference $X_2 - X_1$ is as small as possible. The region $X_1 < X < X_2$ is then called the shortest 67% confidence interval.

Multimodal posterior pdfs: We can sometimes obtain posteriors which are multimodal; i.e. contains several disconnected regions with large probabilities. There is no difficulty when one of the maxima is very much larger than the others: we can simply ignore the subsidiary solutions, to a good approximation, and

concentrate on the global maximum. The problem arises when there are several maxima of comparable magnitude. What do we now mean by a best estimate, and how should we quantify its reliability? The idea of a best estimate and an error-bar, or even a confidence interval, is merely an attempt to summarize the posterior with just two or three numbers; sometimes this just can't be done, and so these concepts are not valid. For the bimodal case we might be able to characterize the posterior in terms of a few numbers: two best estimates and their associated error-bars, or disjoint confidence intervals. For a general multimodal pdf, the most honest thing we can do is just display the posterior itself.

Simple Photon Counts: Best estimates and confidence intervals. To compute these numbers for our example, you would run:

```
sampper=np.percentile(samples, [2.5, 16.5, 50, 83.5, 97.5],axis=0).flatten()
print("""
F_true = {0}
Based on {1} measurements the posterior point estimates are:
...F_est = {2:.0f} +/- {3:.0f}
or using credibility intervals:
...F_est = {4:.0f}          (posterior median)
...F_est in [{5:.0f}, {6:.0f}] (67% credibility interval)
...F_est in [{7:.0f}, {8:.0f}] (95% credibility interval) """\
          .format(F_true, N, np.mean(samples), np.std(samples), \
                      sampper[2], sampper[1], sampper[3], sampper[0], sampper[4]))
```

```
F_true = 1000
Based on 50 measurements the posterior point estimates are:
...F_est = 998 +/- 4
or using credibility intervals:
...F_est = 998          (posterior median)
...F_est in [993, 1002] (67% credibility interval)
...F_est in [989, 1006] (95% credibility interval)
```

In this particular example, the posterior pdf is actually a Gaussian (since it is constructed as a product of Gaussians), and the mean and variance from the quadratic approximation will agree exactly with the frequentist approach.

From this final result you might come away with the impression that the Bayesian method is unnecessarily complicated, and in this case it certainly is. Using an MCMC sampler to characterize a one-dimensional normal distribution is a bit like using the Death Star to destroy a beach ball, but we did this here because it demonstrates an approach that can scale to complicated posteriors in many, many dimensions, and can provide nice results in more complicated situations where an analytic likelihood approach is not possible.

Furthermore, as data and models grow in complexity, the two approaches can diverge greatly.

0.5 Example: Gaussian noise and averages

The first example in the demonstration notebook is from Sivia's book. How do we infer the mean and standard deviation of a Gaussian distribution from M measurements $D \in \{x_k\}_{k=0}^{M-1}$ that should be distributed according to a normal distribution $p(D|\mu, \sigma, I)$?

Start from Bayes theorem

$$p(\mu, \sigma|D, I) = \frac{p(D|\mu, \sigma, I)p(\mu, \sigma|I)}{p(D|I)}$$

- Remind yourself about the names of the different terms.
- It should become intuitive.
- Bayes theorem tells you how to flip from (hard) $p(\mu, \sigma|D, I) \Leftrightarrow p(D|\mu, \sigma, I)$ (easy).

Aside on the denominator, which is known as the "data probability" or "marginalized likelihood" or "evidence".

- With θ denoting a general vector of parameters we must have

$$p(D|I) = \int d\theta p(D|\theta, I)p(\theta|I).$$

- This integration (or marginalization) over all parameters is often difficult to perform.
- Fortunately, for **parameter estimation** we don't need $p(D|I)$ since it doesn't depend on θ. We usually only need relative probabilities, or we can normalize the unnormalized posterior

$$p(\theta|D, I) \propto p(D|\theta, I)p(\theta|I)$$

when we have computed it.

If we use a uniform prior $p(\theta|I) \propto 1$ (in a finite volume), then the posterior is proportional to the **likelihood**

$$p(\theta|D, I) \propto p(D|\theta, I) = \mathcal{L}(D, \theta)$$

In this particular situation, the mode of the likelihood (which would correspond to the point estimate of maximum likelihood) is equivalent to the mode of the posterior pdf in the Bayesian analysis.

The real use of the prior, however, is to include additional information that you might have into the analysis. The prior makes these additional assumptions very explicit.

But how do we actually compute the posterior in practice. Most often we won't be able to get an analytical expression, but we can sample the distribution using a method known as Markov Chain Monte Carlo (MCMC).

0.6 Example: Fitting a straight line

The second example in the demonstration notebook is the fit of a straight line.

- Here the theoretical model is

$$y_{\rm th}(x;\theta) = mx + b,$$

with parameters $\theta = [b, m]$.

- The statistical model for the data is

$$y_{\rm exp} = y_{\rm th} + \delta y_{\rm exp},$$

where we often assume that the experimental errors are independent and normally distributed so that

$$y_i = \mathcal{N}\left(y_{\rm th}(x_i;\theta), e_i^2\right).$$

- Is independent errors a good approximation?

- An even better statistical model for finite resolution models would be

$$y_{\rm exp} = y_{\rm th} + \delta y_{\rm exp} + \delta y_{\rm th}.$$

Why normal distributions? Let us give a quick motivation why Gaussian distributions show up so often. Say that we have a pdf $p(\theta|D, I)$. Our best estimate from this pdf will be θ_0 where

$$\left.\frac{\partial p}{\partial \theta}\right|_{\theta_0} = 0, \qquad \left.\frac{\partial^2 p}{\partial \theta^2}\right|_{\theta_0} < 0.$$

The distribution usually varies very rapidly so we study $L(\theta) \equiv \log p$ instead. Near the peak, it behaves as

$$L(\theta) = L(\theta_0) + \frac{1}{2}\left.\frac{\partial^2 L}{\partial \theta^2}\right|_{\theta_0} (\theta - \theta_0)^2 + \ldots,$$

where the first-order term is zero since we are expanding around a maximum and $\partial L/\partial\theta = 0$.

If we neglect higher-order terms we find that

$$p(\theta|D, I) \approx A \exp\left[\frac{1}{2}\left.\frac{\partial^2 L}{\partial \theta^2}\right|_{\theta_0} (\theta - \theta_0)^2\right],$$

which is a Gaussian $\mathcal{N}(\mu, \sigma^2)$ with

$$\mu = \theta_0, \qquad \sigma^2 = \left(-\left.\frac{\partial^2 L}{\partial \theta^2}\right|_{\theta_0}\right)^{-1/2}.$$

0.7 Confidence intervals

Rather than point estimates it is better to provide a confidence interval for the parameter(s)

A Bayesian confidence interval, or credible interval, or degree-of-belief (DOB) interval is the following: Given this data and other information there is $d\%$ probability that this interval contains the true value of the parameter. E.g. a 95% DOB interval implies that the Baysian data analyser would bet 20-to-1 that the true result is inside the interval.

A frequentist 95% confidence interval should be understood as follows: "There is a 95% probability that when I compute a confidence interval from data of this sort that he true value of the parameter will fall within the (hypothetical) space of observations". So the parameter is fixed (no pdf) and the confidence interval is based on random sampling of data.

Let's try again to understand this: If we make a large number of repeated samples, then 95% of the intervals extracted in this way will include the true value of the parameter.

Some issues with confidence intervals:

If the distribution is symmetric, then it is clear how to define the interval: Just start from the center, then step outward adding probability area and stop when you have reached $d\%$.

What if the distribution is asymmetric or multimodal?

- Equal-tailed interval: the probability area above and below the interval are equal.
- Highest posterior density (HPD) interval: The posterior density for any point within the interval is larger than the posterior density for any point outside the interval.

0.8 Correlations

In the "fitting a straight-line" example you should have seen that the joint pdf for the slope and the intercept $[m, b]$ corresponds to a slanted ellipses. That means that the parameters are **correlated**.

A Taylor expansion for a pdf $p(x, y)$ around the mode (x_0, y_0) gives

$$p(x,y) \approx p(x_0,y_0) + \frac{1}{2}\begin{pmatrix} x-x_0 & y-y_0 \end{pmatrix} H \begin{pmatrix} x-x_0 \\ y-y_0 \end{pmatrix},$$

where H is the symmetric Hessian matrix

$$\begin{pmatrix} A & C \\ C & B \end{pmatrix}, \qquad A = \left.\frac{\partial^2 p}{\partial x^2}\right|_{x_0,y_0}, \quad B = \left.\frac{\partial^2 p}{\partial y^2}\right|_{x_0,y_0}, \quad C = \left.\frac{\partial^2 p}{\partial x \partial y}\right|_{x_0,y_0}.$$

- So in this quadratic approximation the contour is an ellipse centered at (x_0, y_0) with orientation and eccentricity determined by A, B, C.
- The principal axes are found from the eigenvectors of H.
- Depending on the skewness of the ellipse, the parameters are either not correlated, correlated, or anticorrelated.
- Take a minute to consider what that implies?