Computational Physics – Lecture 25

(18 May 2020)

Recap

In the previous lecture, we learnt two Monte Carlo methods: (a) MCMC and (b) Simulated Annealing.

Today's plan

In today's class we want to learn probabilistic inference.

Data

Suppose an experiment has measured a quantity y as a function of another quantity x.

The experimentalist has also estimated the uncertainty σ_y in each measurement.

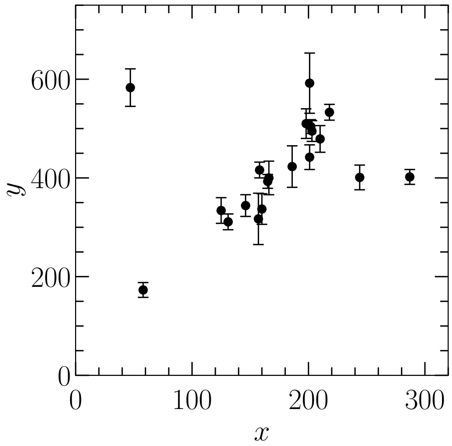
The measurements reported by the experiment is our *data*. We would like to use it to verify a physics *model*.

Data

Here is the data given to us by the experimentalist:

index	x	y	σ_y
1	201	592	61
2	244	401	25
3	47	583	38
4	287	402	15
5	203	495	21
6	58	173	15
7	210	479	27
8	202	504	14
9	198	510	30
10	158	416	16
11	165	393	14
12	201	442	25
13	157	317	52
14	131	311	16
15	166	400	34
16	160	337	31
17	186	423	42
18	125	334	26
19	218	533	16
20	146	344	22

Data



A visualisation of our data with points and errorbars.

Model

Now, given this data, we want to find a function of the form

$$y \equiv f(x) = mx + b \tag{1}$$

that *best fits* the data.

We may want to do this because

- We are prejudiced towards this $y = mx + b$ form ("theory"), or
- We want to agnostically check if a linear model works ("phenomenology")

This functional form is our *model*.

Probabilistic inference

We now want to know the best-fit values of the model parameters m and b from our data. This is called *inference*.

One way of doing inference is to produce best-fit values of the model parameters with an associated probability distribution. This is called *probabilistic inference*.

This is also known as “fitting models to data”.

Note that for given data, you can fit any model with its own set of model parameters. (For our given data, nobody can stop us from fitting $y = ax^2 + bx + c$ and inferring a, b and c.)

Probabilistic inference

We now want to understand how to infer m and b probabilistically, given our data.

This is done in two steps:

1. Find an *objective function* that represents the quality of the fit of the model to the data
2. Optimise the objective function to get the best-fit parameter values with their PDFs

Probabilistic inference

How do we get the objective function?

We imagine that the data actually do come from a line of the form $y = mx + b$.

We also assume that the only reason a point deviates from this straight line is because to the true value of y is added an offset drawn from a Gaussian distribution with zero mean and known variance σ_y^2.

Probabilistic inference

These assumptions imply that the probability that the i^{th} point has value y_i is given by

$$p(y_i|x_i,\sigma_{yi},m,b) = \frac{1}{\sqrt{2\,\pi\,\sigma_{yi}^2}}\,\exp\left(-\frac{[y_i - m\,x_i - b]^2}{2\,\sigma_{yi}^2}\right) \quad . \qquad (2)$$

This is called a *generative model* for the objective function.

Probabilistic inference

We can now write down the probability for our data given our model as

$$\mathscr{L} = \prod_{i=1}^{N} p(y_i | x_i, \sigma_{yi}, m, b) \quad . \tag{3}$$

Taking the logarithm, we can write

$$\begin{aligned} \ln \mathscr{L} &= K - \sum_{i=1}^{N} \frac{[y_i - m\,x_i - b]^2}{2\,\sigma_{yi}^2} \\ &= K - \frac{1}{2}\,\chi^2, \end{aligned} \tag{4}$$

where K is some constant and we encounter the χ^2 statistic again.

$\mathscr{L}$ is called the *Likelihood*.

Probabilistic inference

Now how do we do our inference from the above $\ln \mathscr{L}$ function?

We use the Bayes Theorem of conditional probabilities

$$p(A|B) = \frac{p(B|A)\,p(A)}{p(B)} \tag{5}$$

to write

$$p(m, b|\{y_i\}_{i=1}^{N}, I) = \frac{p(\{y_i\}_{i=1}^{N}|m, b, I)\,p(m, b|I)}{p(\{y_i\}_{i=1}^{N}|I)}. \tag{6}$$

Probabilistic inference

Here,

- $\{y_i\}_{i=1}^N$ is our data
- I is all the prior knowledge of the x_i and the σ_{yi} and everything else about the problem.
- $p(\{y_i\}_{i=1}^N|m, b, I)$ is the likelihood $\mathscr{L}$
- $p(m, b|I)$ is the *prior* probability distribution for the parameters that represents all knowledge *except* the data $\{y_i\}_{i=1}^N$
- $p(m, b|\{y_i\}_{i=1}^N, I)$ is the *posterior* probability distribution for the parameters given the data and the prior knowledge
- The denominator $p(\{y_i\}_{i=1}^N|I)$ is just a normalisation constant.

Probabilistic inference

So the expression

$$p(m, b|\{y_i\}_{i=1}^N, I) = \frac{p(\{y_i\}_{i=1}^N|m, b, I)\, p(m, b|I)}{p(\{y_i\}_{i=1}^N|I)}. \tag{7}$$

is updating our knowledge of $p(m, b)$ from the *prior* to the *posterior* by taking new data $\{y_i\}_{i=1}^N$ into account.

You can think of this is

$$p(\text{model}|\text{data}) = \frac{p(\text{data}|\text{model})\, p(\text{model})}{p(\text{data})} \tag{8}$$

This way of doing probabilistic inference is called Bayesian statistics.

Probabilistic inference

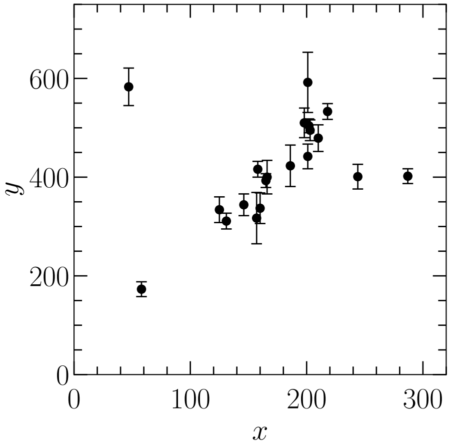
Given the posterior PDF $p(m, b|\{y_i\}_{i=1}^N, I)$, we can do our inference by noticing that

- The best-fit values of m and b are the values at which $p(m, b|\{y_i\}_{i=1}^N, I)$ peaks
- The uncertainties on the values of m and b are just the moments of $p(m, b|\{y_i\}_{i=1}^N, I)$

The above can be computed by MCMC.

Let us do this for our data and our model.

Probabilistic inference



Probabilistic inference

Let us begin by coding up our $\ln \mathscr{L}$ function:

```
def log_likelihood(theta, x, y, yerr):
    m, b = theta
    model = m*x + b
    sigma2 = yerr**2

    # actually negative ln L
    return 0.5 * np.sum((y - model) ** 2 / sigma2 +
                        np.log(2 * np.pi * sigma2))
```

Probabilistic inference

Now the prior distribution $p(m, b|I)$.

We are *a priori* ignorant about the parameters so we choose a uniform prior:

```
def log_prior(theta):
    m, b = theta
    if -500.0 < m < 500 and 0.0 < b < 1000.0:
        return 0.0
    return -np.inf
```

Probabilistic inference

With the prior and likelihood coded up, we can now code our posterior PDF:

```
def log_probability(theta, x, y, yerr):

    lp = log_prior(theta)
    if not np.isfinite(lp):
        return -np.inf
    return lp - log_likelihood(theta, x, y, yerr)
```

Probabilistic inference

Now we can sample our posterior PDF using MCMC.

Let us use 32 Markov chains.

Where do we initialise them? Anywhere we want but a common idea to start near the optimum of the likelihood.

```
from scipy.optimize import minimize

guess = (1.0, 1.0)
soln = minimize(log_likelihood,
                guess,
                args=(x, y, sigma_y))
```

Probabilistic inference

We now initialise each of our 32 Markov chains near the optimum reported by the `minimize` function.

```
nwalkers, ndim = 32, 2
pos = soln.x + 1e-4 * np.random.randn(nwalkers, ndim)
```

Probabilistic inference

We now use the `emcee` library to do the MCMC so that each Markov chain takes 5,000 steps.

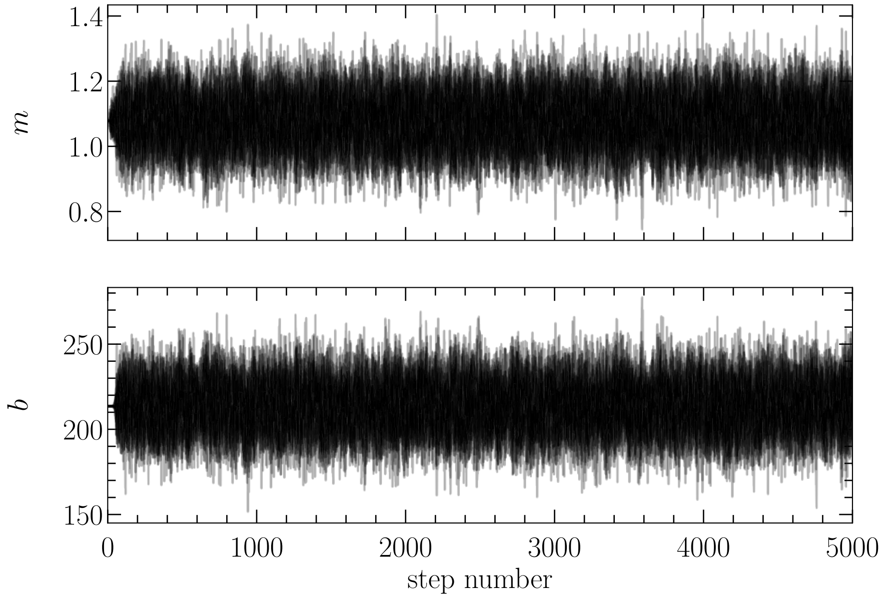
```
import emcee

sampler = emcee.EnsembleSampler(nwalkers,
                                ndim,
                                log_probability,
                                args=(x, y, yerr))
sampler.run_mcmc(pos, 5000)
```

Probabilistic inference

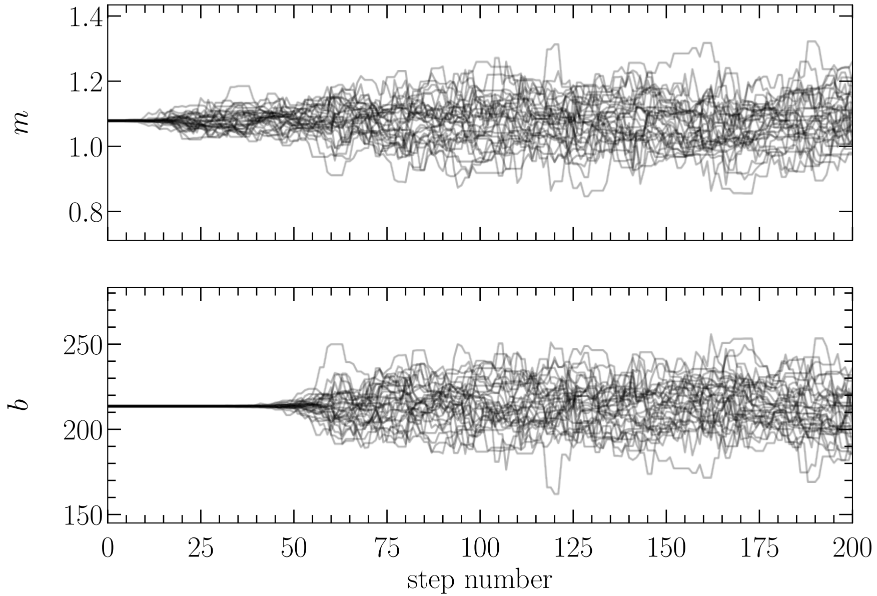
We can look at the chains by plotting them:

```
samples = sampler.get_chain()
plt.plot(samples[:, :, 0]) # m values
plt.plot(samples[:, :, 1]) # b values
```

Markov Chains



"Burn in"



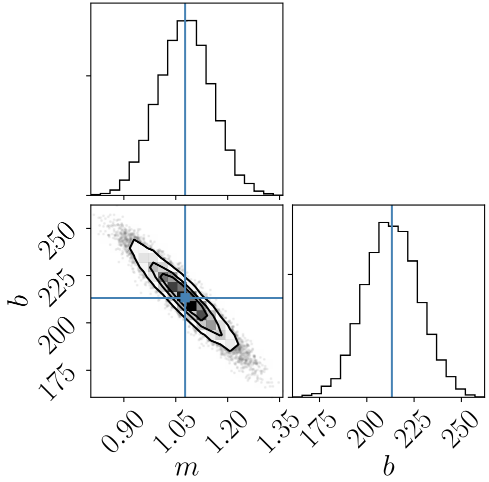
m
1.4
1.2
1.0
0.8
b
250
200
150
0
25
50
75
100
125
150
175
200
step number

Posterior distributions

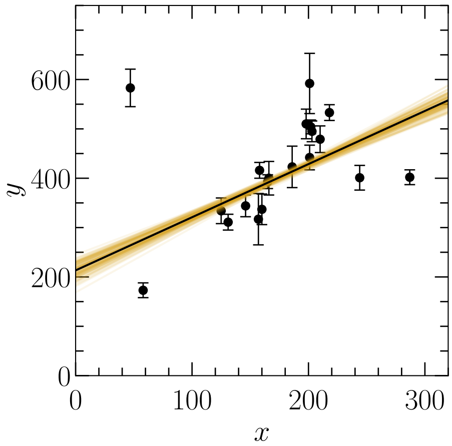
We can plot the posterior PDF using the `corner` library.

```
import corner

medians = np.median(samples, axis=0)
m_true, b_true = medians

fig = corner.corner(samples,
                    labels=labels,
                    truths=[m_true, b_true])
```

Posterior distributions



Probabilistic inference



Our "best-fit" model with a sample from the model posterior distribution.

Probabilistic inference

We can summarise our findings by using the uncertainties based on the 16th, 50th, and 84th percentiles of the samples in the marginalized distributions.

So our inference is

$$m = 1.077025^{+0.076651}_{-0.079228} \tag{9}$$

and

$$b = 213.258205^{+14.673500}_{-14.434788}. \tag{10}$$

(The uncertainty based on the 16th and 84th percentiles is sometimes called the "1-sigma uncertainty", for historical reasons.)

Probabilistic inference

Note that our result depends on the prior.

Also, the data did not tell us if $mx + b$ is a good model.

However, note the simplicity of the inference procedure. This makes Bayesian statistics easy to generalise to complex cases.

Probabilistic inference

Let us summarise what we did:

- For a given data, construct a model
- Given the model, write a likelihood
- Consider your prior belief in the model and write a prior
- Combine the prior and likelihood using the Bayes theorem to get the posterior PDF
- Sample the posterior PDF using MCMC
- Use the sample to get quantities such as best-fit parameter values, one-sigma uncertainty ranges, etc.

Wrap-up

What have we learnt so far:

- Uniform random number generation
- Non-uniform random number generation
- Tests of randomness
- Monte Carlo integration
- Monte Carlo sampling
- Monte Carlo optimization
- Monte Carlo inference

Wrap-up

In these 25 lectures, we have learnt techniques to solve various kinds of problems on a computer:

- Matrix inversion using decomposition and iteration
- Eigenvalue problems using decomposition and iteration
- Singular Value Decomposition
- ODEs using forward and backward Taylor-type methods
- ODEs as IVPs and BVPs
- Fourier transforms
- Convolution and spectral analyses
- Random numbers and Monte Carlo techniques
- Software engineering: libraries, documentation, Python/C
- Computer science: analysis of algorithms, error analysis

Wrap-up

How to get better at all this?

- Embrace computational techniques
- Practice
- Engage
- Discuss
- Read (literature *and* code!)

Wrap-up

Learn to distinguish between operational capability and theoretical understanding.

For example, understanding DFT with $n \neq 2^m$ is an interesting theoretical question, but it should not distract you from mastering DFT for $n = 2^m$, which covers 99.9% cases.

Some results in computing are serendipitous and theoretically ill-founded (e.g., some results in random number theory). What is important is that they are still very useful to solve real physics problems.

Next

https://forms.gle/GpyNDMqMLFb4kpTs9

Next

We will have an Assignment 4.

There will be an exam in the last week of May.