In this session you learn how to use MCMC methods to place constraints on model parameters. The key question today is, "How do we estimate the correct parameters (input) of a simulation if we only get to observe the results (output) of the simulation?" To put this in a context you have seen before, how would you estimate the temperature parameter of the Ising model if you got to observe one or more samples from the model state?

This is far from a trivial application: applications range from financial engineering, climate forecasting, computational biology and cosmology. During this session, we will discuss two main examples:

1. constraining the ingredients of our universe, and
2. determining the half-life of a radioactive isotope.

To be prepared for class, you will need to:

- have a high-level understanding of how cosmologists use MCMC methods to place constraints on dark-energy and other constituents of our universe.
- know how to compute an isotope's half-life given experimental data on the amount of isotope left over time—this is Eq. (2) below.
- understand how the Metropolis–Hastings acceptance criteria bypasses the need to know $P(\text{data})$ (the probability that the data was generated by the fundamental model we are constraining/testing). In other words, why Eq. (3) does not contain $P(\text{data})$.

The material included below is meant to provide you with extra guidance towards these points.

1 Some background knowledge on our application examples

1.1 Ingredients of our universe

The CMBR (Cosmic Microwave Background Radiation) is the oldest memory of the 13.7 billion year-old universe we live in. It is indeed the only fossilized record of microphysics in the early universe which shows evidence of a young, vibrant universe. We measure the anisotropies in the CMBR as fluctuations in temperature, against an almost uniform temperature field. The statistics in the temperature field can in turn be used to trace back what mechanism generated them.

To understand the origin of these small differences in the temperature field, physicists have conjectured a period of dramatically accelerated expansion when the universe was just born. This period is technically referred to as *inflation*, during which space itself expanded at a speed greater than that of the light and quantum fluctuations of the primitive microphysics became imprinted in the CMBR. The same fluctuations have also become the seeds for structure formation in our Cosmos. Hence, by closely examining the CMBR today, we can get a glimpse of what happened in the primordial universe but also at the current constituents in our universe.

Though inflation is the most popular paradigm, other proposals for our universe have also been put forward. All models that have been proposed thus far have suffered the scrutiny of last decade's data-driven cosmology. Never before had we had access to cosmological surveys with such high precision at different scales. With precise measurements, many models have been severely constrained or even completely ruled-out by MCMC methods.

Pulling out different datasets, our best estimate for the universe's budget is as follows:

- the largest component is **dark energy** with a proportion of about 69% of all there is. This is sometimes referred to as the cosmological constant, denoted by Λ, which was a (constant) term that Einstein introduced as an extra parameter in his General Relativity equations (these describe gravity in the presence of matter). Constraint analysis suggests not only that this term is non-zero (contrary to what Einstein believed), but that it is also overwhelmingly dominating the later universe's history.

- regular matter, i.e., **atoms** (or technically, **baryonic matter**) builds up only 5%.
- the remainder 26% is made up of (cold) **dark matter**, which we can't see but again is a conjectured contribution inferred from data.

The concordance cosmological model goes by the name of ΛCDM, which stands for Λ and Cold Dark Matter (CDM). In the simulation on WMAP's website[1], there are three additional parameters you can play with:

- the **Hubble's constant**, usually represented by H_0, which quantifies the rate of growth of the space expansion.
- the **reionization redshift** that translates into the redshift[2] when the first stars in our universe were born.
- the **spectral index**, which is related to the scaling of the power spectrum, i.e., the two-point correlation function of the primordial fluctuation. Our primordial power spectrum is essentially scale free (similarly to what we have seen in the Barabási-Albert network model!), and only slightly red-tilted (correlations are larger at larger scales).

Whilst the physical significance of these parameters is not important for understanding the MCMC approach to constraining different universes, the above should provide with enough context for running the simulation. It should also emphasize how complicated solving the universe is: it is a multi-parametrised system (we have 6 parameters here) subject to 13 billion years of time evolution. Constraining the ΛCDM parameters is a tour de force!

1.2 Radioactive decay

Half-life is a concept which is central to survival analysis studies. In the context of radioactive decay, we can parametrise the amount of isotope left by the exponential, radioactive decay as follows

$$N(t) = N_0 \, \exp\left(-\lambda \, t\right) \, , \tag{1}$$

where N_0 is the initial quantity of the isotope and t the amount of time that has occurred since that initial stage. The constant λ parametrises how fast or slow the decay process happens. This parameter can be estimated through experimental data. Applying the logarithm to both sides of the previous equations, gives

$$\ln N = -\lambda \, t + \ln N_0 \, .$$

The slope of the straight line in the $(\ln N, t)$ plane is $-\lambda$. This parameter relates to the element's half-life. If we set

$$\frac{N}{N_0} = \frac{1}{2}$$

in Eq. (1), we get

$$t_{\mathrm{half-life}} = \frac{\ln 2}{\lambda} \, . \tag{2}$$

You will use this formula to solve Exercise 2 of the pre-class work. Note that, even though applying the logarithm to both sides of Eq. (1) simplifies the fitting of the parameters to finding the slope of a simple straight line, it is not necessary. Indeed the optimization techniques offered by the scipy module overcome this extra step.

[1] The simulation uses the WMAP constraints. There have been more recent missions, most notably Planck's, which has found slightly different values for these parameters. However, the simulation offered on their website is far less playful, so we will use WMAP's instead.

[2] The redshift z is related to the scale factor a via $a = (1+z)^{-1}$.

2 MCMC parameter estimation as a random walk

Let's assume we have some data and we would like to use a certain model to explain that data. This model is defined by a set of parameters, θ, which, in general, is an n-vector mapping n model parameters, $\boldsymbol{\theta}$.

MCMC parameter estimation relies on a random-walk implementation of the Metropolis–Hastings algorithm. The starting point is Bayes's formula, according to which we can determine the posterior, i.e., the probability that $\boldsymbol{\theta}$ is supported by the data via

$$P(\boldsymbol{\theta}|\text{data}) = \frac{P(\text{data}|\boldsymbol{\theta}) \times P(\boldsymbol{\theta})}{P(\text{data})} \ .$$

The likelihood is represented by the term $P(\text{data}|\boldsymbol{\theta})$, whereas the prior is $P(\boldsymbol{\theta})$. Choosing a likelihood and prior mostly depend on the application we are focusing on, and different scenarios will have different optimal priors. The evidence that the data were generated by the model with parameters $\boldsymbol{\theta}$ is denoted by $P(\text{data})$ and it's remarkably difficult, if not impossible, to compute.

This is where the Metropolis–Hastings comes to the rescue. Rather than computing $P(\boldsymbol{\theta}|\text{data})$ by itself, we can compute the relative magnitude of this probability as new proposals come forward. In particular, in

$$\frac{P(\boldsymbol{\theta}_p|\text{data})}{P(\boldsymbol{\theta}_c|\text{data})} = \frac{\dfrac{P(\text{data}|\boldsymbol{\theta}_p) \times P(\boldsymbol{\theta}_p)}{P(\text{data})}}{\dfrac{P(\text{data}|\boldsymbol{\theta}_c) \times P(\boldsymbol{\theta}_c)}{P(\text{data})}} = \frac{P(\text{data}|\boldsymbol{\theta}_p) \times P(\boldsymbol{\theta}_p)}{P(\text{data}|\boldsymbol{\theta}_c) \times P(\boldsymbol{\theta}_c)} \ ,$$

where $\boldsymbol{\theta}_p$ is the proposal for the $\boldsymbol{\theta}$ set of parameters and $\boldsymbol{\theta}_c$ is the current guess, there is no dependence on $P(\text{data})$ (or, more rigorously, it cancels out). For simplicity, we will choose uninformative priors, so that $P(\boldsymbol{\theta}_p) = P(\boldsymbol{\theta}_c)$. The criterion selection then simplifies to a ratio of likelihoods:

$$r = \frac{P(\boldsymbol{\theta}_p|\text{data})}{P(\boldsymbol{\theta}_c|\text{data})} = \frac{P(\text{data}|\boldsymbol{\theta}_p)}{P(\text{data}|\boldsymbol{\theta}_c)} \equiv \frac{\text{likelihood}(\boldsymbol{\theta}_p)}{\text{likelihood}(\boldsymbol{\theta}_c)}. \tag{3}$$

Therefore, the acceptance criterion uniquely depends on the likelihood function for the model parameters, $\boldsymbol{\theta}$, which will measure how the likelihood will change in response to a different set of parameters. The likelihood is simply the plausibility that the parameters have a certain value for the given data. Choosing a likelihood function is a non-trivial task. Assuming the data follow a Gaussian distribution, we can use the least squares (χ^2) to feed into the likelihood, cf.

$$\text{likelihood} \sim \exp\left(-\chi^2\right) \ . \tag{4}$$

In simulations, it is quite frequent to obtain incredibly small numbers for the likelihood, so as a means of avoiding numerical overflow, we usually compute the (natural) logarithm of the likelihood, rather than the likelihood itself.

2.1 The simplified algorithm

As outlined in Rein (from the pre-class reading), to implement MCMC for parameter estimation we need to follow the steps:

1. choose a likelihood function. For simplicity, we will make the choice (4).
2. start with an initial guess for the model parameters, $\boldsymbol{\theta}_{\text{guess}} = (\theta_{1,\text{guess}}, \theta_{2,\text{guess}}, \cdots, \theta_{\text{n,guess}})$, and assume as above uninformative priors.
3. perform a large number of trials for which you add a Gaussian, random deviation from the previously chosen values for $\boldsymbol{\theta}$. These will be the new guesses for our parameters set.

4. for these new parameters $\boldsymbol{\theta}$, compute the least squared errors, and therefore the (log-)likelihood for those parameters.

5. compute the likelihood ratio

$$r = \exp\left(\text{new log likelihood - current log likelihood}\right) , \tag{5}$$

   which gives the probability of accepting the new guess for $\boldsymbol{\theta}$.

6. compare this ratio with a random sample, s, of the uniform distribution, following the Metropolis–Hastings acceptance criterion. If $s < r$, then take the new, proposed values $\boldsymbol{\theta}_p$ as the current guess, and the proposed value of likelihood as the current likelihood. Otherwise, keep the current guess, $\boldsymbol{\theta}_c$.

7. record the different guesses for theta as the simulation proceeds.

Refer to the code in the readings to see how these steps are implemented in Python. To observe how MCMC methods, based on the Metropolis-Hastings algorithm, explore the parameter phase space, one can plot the evolution (over the model iterations) of the different model estimates.