Astronomy 8824: Problem Set 4
Due Tuesday, October 17

Multi-variate Gaussians and Simple MCMC

References

For bivariate and multivariate Gaussians, section 3.5 of Ivezic et al.

For MCMC, sections 5.8.1 and 5.8.2 of Ivezic et al., and section 15.8 of the 3rd edition of Numerical Recipes, though this topic wasn't in the 1st or 2nd edition. Useful journal article references are Dunkley et al. 2005, MNRAS, 356, 925 and the more comprehensive review of Sharma 2017, ARAA 55, 213, arXiv:1706.01629.

For those of you who have learned to love `sm` (or at least to tolerate it), I have put my plotting scripts for this solution on the web page.

1. Bivariate Gaussian

Generate 5000 random data pairs (p_1, p_2) where p_1 and p_2 are drawn independently from Gaussians of standard deviation $\sigma_1 = 2$ and $\sigma_2 = 0.5$, respectively (with mean zero). (Use the python routine `np.random.normal`.)

Compute new data pairs (x, y) with $x = p_1 \cos\alpha - p_2 \sin\alpha$ and $y = p_1 \sin\alpha + p_2 \cos\alpha$ for $\alpha = \pi/6$.

Plot these two distributions (e.g., as tiny dots with different colors) over the top of each other, and plot as x and y-axis histograms the marginal distributions of p_1, p_2, x, and y.

Using equations (3.85)-(3.87) of Ivezic et al., compute the expected values of σ_x, σ_y, and σ_{xy}. Are the marginal distributions for x and y in your plot Gaussians with the expected widths? (Overplot Gaussians for comparison.)

What is the covariance matrix of (x, y)? (Compute this analytically, though it may be useful to compare it to your numerical estimate.)

Draw 5000 random data pairs from a bivariate Gaussian with this covariance matrix using `np.random.multivariate_normal`.
Compare this distribution and the marginal distributions of x and y to the ones you got by your previous procedure and comment on the result.

If you want an example of using `np.random.multivariate_normal`, you can start from my code `gauss2d_example.py` on the web page.

2. MCMC realization of a 2-d probability distribution

The probability distribution for the bivariate Gaussian distribution in Part 1 is:

$$p(\mathbf{x}) = \frac{1}{2\pi\sqrt{\det(\mathbf{C})}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x}\right),$$

where $\mathbf{x} = (x, y)$, $\mathbf{C} = \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}$, and $\mathbf{H} = \mathbf{C}^{-1}$.

Implement a simple Markov Chain Monte Carlo (MCMC) routine:

1. Start at a user-specified location x_0, y_0.

2. At each iteration generate a trial point (x_{i+1}, y_{i+1}) with

$$x_{i+1} = x_i + h\sigma_x \mathcal{N}(0,1)$$
$$y_{i+1} = y_i + h\sigma_y \mathcal{N}(0,1)$$

where $\mathcal{N}(0,1)$ is a Gaussian random variable of zero mean and unit dispersion (chosen separately for x and y) and h is a user-specified scaling of step size.

3. If $p(x_{i+1}, y_{i+1}) > p(x_i, y_i)$ accept the step, i.e., add the new pair to the chain and take your next trial step from this new position.

4. If $p(x_{i+1}, y_{i+1}) < p(x_i, y_i)$ then accept the step with probability $p(x_{i+1}, y_{i+1})/p(x_i, y_i)$ (draw a uniform random deviate and compare it to this ratio). If the step is not accepted, *do not add the new point to the chain*, and go back to step 2 to choose a new trial point.

5. Output (or just plot within your program) the final distribution of the chain. Also keep track of and report the fraction of trial steps that are accepted, i.e., the ratio of the final length of the chain to the total number of steps needed to produce it.

Use this program to generate a 5000-element chain starting from $(x, y) = (1, 1)$ with step scaling $h = 1$. Plot the distribution of points from this chain, and the corresponding marginal distributions, over the bivariate Gaussian distribution from Part 1. If your programs are working, you should get good agreement.

Try several different starting points and compare the results. You can just describe this comparison in words.

Change h from 1 to 0.1. Compare the distribution to that for $h = 1$ (with a plot), and compare the fraction of steps that are accepted.

Change h from 1 to 2.5. Compare the distribution to that for $h = 1$ (with a plot), and compare the fraction of steps that are accepted.

In generating the initial bivariate Gaussian and computing the covariance matrix for the MCMC, change σ_2 from 0.5 to 0.1. Compare to your previous results, for $h = 1$.

Comment on issues of efficiency and accuracy in MCMC computations and strategies that could improve the efficiency for the $\sigma_2 = 0.1$ case.

3. Cosmic MCMC: Parameters of the Universe

Here we will do a simplified version of the statistical analysis in Aubourg et al. (`arXiv:1411.1074`).

You'll need to adapt the program `cosmodist.py` that I provided for PS 3, or your own code that does the equivalent. This time, we will use its ability to compute distances for $\Omega_k \neq 0$ and $w = -1$. Refer back to PS 3 for the relevant equations. Because you'll evaluate this integral many times, I recommend adopting a tolerance of 3×10^{-5}, which is adequate given the uncertainties of our observational constraints.

As the cosmological constraints, take the following (the first two are from the CMB, and others are from BAO measurements):

$\Omega_m h^2 = 0.1386 \pm 0.0027$.

$D_M(z = 1090) = 13962 \pm 10$ Mpc.

$D_M(z = 2.34) = 5381 \pm 170$ Mpc.

$H(z = 2.34) = 222 \pm 5\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$.

$D_M(z = 0.57) = 2204 \pm 31$ Mpc.

$H(z = 0.57) = 98 \pm 3\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$.

$D_M(z = 0.32) = 1249 \pm 25$ Mpc.

Compute the likelihood of the data for a given set of cosmological parameters as $L \propto e^{-\chi^2/2}$, where χ^2 is computed from the above data values ignoring any error covariances (i.e., $\chi^2 = \sum (y_i - y_{\mathrm{mod},i})^2/\sigma_i^2$).

Adapt your MCMC code to create a chain for cosmological parameter values. You should set it up to allow steps in 4 parameters: Ω_m, h, w, and Ω_k.

First consider a flat universe, with $\Omega_k = 0$, but allowing free w. Create a 2000-point, 3-d MCMC using the parameters Ω_m, h, and w, where $h \equiv H_0/100\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$. (Use your 4-d code but set the step size in the Ω_k dimension to zero.) For a starting point I suggest $\Omega_m = 0.3$, $h = 0.68$, $w = -1$, and for initial step sizes I suggest trying $\Delta = 0.03$ in each parameter.

Note that Δ here refers to the actual steps in Ω_m, h, and w, and I've chosen it because I know that these data give parameter errors that are roughly in this ballpark. Don't further multiply 0.03 by the expected standard deviations of these parameters — that would be like taking $h = 0.03$ in Part 2, and you already saw (I hope) that $h = 0.1$ leads to chains that don't explore the likelihood surface very well. I warn you in advance that with $\Delta = 0.03$ your acceptance fraction in the MCMC will be low, $\sim 1\%$, but if you take a much smaller step then you will not get good likelihood sampling.

Plot the distribution of your points in the planes w vs. Ω_m, w vs. h, and Ω_m vs. h.

Now consider a universe, with $w = -1$ and free Ω_k. Create a 2000-point, 3-d MCMC using the parameters Ω_m, h, and Ω_k. (This time set the step size in the w dimension to zero.) For a starting point I suggest $\Omega_m = 0.3$, $h = 0.68$, $\Omega_k = 0$, and for initial step sizes I suggest trying $\Delta = 0.03$ in the first two parameters and $\Delta = 0.003$ in Ω_k.

Plot the distribution of your points in the planes Ω_k vs. Ω_m, Ω_k vs. h, and Ω_m vs. h.

For reference, you may want to look at Figure 8 of Aubourg et al. (the wCDM and oΛCDM panels), but you shouldn't expect to get exactly the same results. The main simplifications are that you are not including covariances of the errors and that I have converted the BAO measurements to absolute units using the best-fit value of the sound horizon r_d, which is well known (to 0.4%) but not perfectly known; a full calculation would consider its dependence on cosmological parameters.