Solutions to the Homework in Lecture 9
Fundamentals of Observational Cosmology (III)

These notes provide worked solutions to the homework problems in Lecture 9. Throughout, we use the supernova residual vector

$$\boldsymbol{\Delta} \equiv \boldsymbol{\mu}_{\rm th} - \boldsymbol{\mu}_{\rm obs},$$

and the Gaussian Fisher matrix for a parameter-dependent mean and covariance,

$$F_{ij} = \frac{1}{2}\,\mathrm{Tr}\big(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,i}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,j}\big) + \boldsymbol{\mu}_{,i}^{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{,j}.$$

Problem 1. Matrix χ^2

Question. Write the supernova χ^2 first for diagonal covariance and then for a full covariance matrix. Explain what physical effects can create off-diagonal covariance.

Solution. If the supernova errors are treated as independent, the covariance matrix is diagonal:

$$C_{ij} = \sigma_i^2\,\delta_{ij}.$$

Then the usual least-squares expression is

$$\chi^2 = \sum_i \frac{\big[\mu_{\rm th}(z_i) - \mu_{{\rm obs},i}\big]^2}{\sigma_i^2}.$$

This is the formula students usually meet first.

For a realistic dataset, however, the errors of different supernovae need not be independent. Then we must use the full covariance matrix:

$$\chi^2 = \boldsymbol{\Delta}^T\mathbf{C}^{-1}\boldsymbol{\Delta} = \sum_{i,j}\Delta_i\,(\mathbf{C}^{-1})_{ij}\,\Delta_j, \qquad \Delta_i \equiv \mu_{\rm th}(z_i) - \mu_{{\rm obs},i}.$$

The off-diagonal terms encode correlations between different entries of the data vector.

What causes off-diagonal covariance? Several physical or observational effects can correlate supernova measurements:

- **Photometric calibration errors.** A zero-point shift in one filter can move many supernova magnitudes together.
- **Light-curve model systematics.** Uncertainty in the training of the light-curve fitter can affect many objects in a correlated way.
- **Selection effects and Malmquist bias corrections.** The same correction model may enter many supernovae, again creating correlated uncertainties.

- **Peculiar-velocity or bulk-flow corrections at low redshift.** Nearby supernovae can share correlated velocity-induced distance errors.
- **Host-galaxy or population corrections.** If one nuisance model is used across the sample, it can correlate the inferred distance moduli.

So the full covariance matrix is the correct description whenever common systematics affect more than one supernova at once.

Problem 2. H_0–M degeneracy

Question. Starting from the definition of the distance modulus, explain why supernovae alone do not determine both H_0 and the absolute magnitude M without additional calibration information.

Solution. The observed apparent magnitude is

$$m = M + \mu, \qquad \mu = 5\log_{10}\left(\frac{D_L}{\mathrm{Mpc}}\right) + 25.$$

For any FRW background,

$$D_L(z) = \frac{c}{H_0}\, d_L(z; \Omega_\mathrm{m}, \Omega_\Lambda, \ldots),$$

where d_L is a *dimensionless* function of redshift and the density parameters. Substituting this into the distance modulus gives

$$\mu(z) = 5\log_{10}\left(\frac{c\, d_L(z)}{H_0\,\mathrm{Mpc}}\right) + 25.$$

Therefore the apparent magnitude becomes

$$m(z) = M - 5\log_{10} H_0 + \underbrace{5\log_{10}\left(\frac{c\, d_L(z)}{\mathrm{Mpc}}\right) + 25}_{\text{depends on } z \text{ and cosmology, but not separately on } M \text{ or } H_0}.$$

The key point is that the dependence on the absolute calibration enters through the *combination*

$$\mathcal{M} \equiv M - 5\log_{10} H_0 + \text{constant}.$$

So supernova data alone measure the shape of the Hubble diagram very well, but they cannot split this combination into an independent value of M and an independent value of H_0.

To see the degeneracy directly, let $H_0 \to \alpha H_0$. Then

$$\mu \to \mu - 5\log_{10}\alpha.$$

This change can be absorbed by shifting the absolute magnitude as

$$M \to M + 5\log_{10}\alpha,$$

so that $m = M + \mu$ is unchanged. Hence a supernova Hubble diagram without external calibration constrains relative distances and cosmological shape parameters much better than the absolute distance scale.

How is the degeneracy broken? We need some external calibration: for example Cepheid calibration, tip-of-the-red-giant-branch distances, masers, or an external prior on H_0. Without such information, SNe Ia alone do not determine both M and H_0 separately.

Problem 3. Grid fitting

Question. Design a toy grid-based fit for $(\Omega_{\rm m}, \Omega_\Lambda)$ using five mock supernovae. You may describe the algorithm in pseudocode or implement it in a language of your choice.

Solution. A simple teaching example is enough here. Choose five mock redshifts, for example

$$z = (0.05,\ 0.20,\ 0.40,\ 0.80,\ 1.10),$$

and construct mock distance moduli from a fiducial cosmology, say $(\Omega_{\rm m}, \Omega_\Lambda) = (0.3, 0.7)$, plus small Gaussian noise. For a toy exercise it is fine to assume diagonal errors, for example

$$\sigma_i = 0.15 \text{ mag for all } i.$$

The grid-based fitting algorithm is then:

1. Choose a grid, for example $\Omega_{\rm m} \in [0, 1]$ and $\Omega_\Lambda \in [0, 1.5]$.
2. For each grid point, compute the theoretical luminosity distance $D_L(z_i; \Omega_{\rm m}, \Omega_\Lambda)$.
3. Convert D_L to theoretical distance moduli $\mu_{\rm th}(z_i)$.
4. Evaluate
$$\chi^2(\Omega_{\rm m}, \Omega_\Lambda) = \sum_{i=1}^{5} \frac{\left[\mu_{\rm th}(z_i; \Omega_{\rm m}, \Omega_\Lambda) - \mu_{{\rm obs},i}\right]^2}{\sigma_i^2}.$$
5. Store the resulting χ^2 value.
6. Find the minimum $\chi^2_{\rm min}$ over the grid.
7. Draw contours of
$$\Delta\chi^2 = \chi^2 - \chi^2_{\rm min}$$
at 2.30, 6.18, and 11.83 for the usual 68%, 95%, and 99.7% two-parameter confidence regions.

A concise pseudocode version is:

```
choose redshifts z[i], observed mu_obs[i], and errors sigma[i]
for Om in grid_Om
    for OL in grid_OL
        chi2 = 0
        for i in 1:5
            mu_th = distance_modulus(z[i], Om, OL)
            chi2 += (mu_th - mu_obs[i])^2 / sigma[i]^2
        end
        store (Om, OL, chi2)
    end
end
find chi2_min
plot contours of DeltaChi2 = chi2 - chi2_min
```

If a full covariance matrix is available, replace the sum by the matrix expression

$$\chi^2 = \mathbf{\Delta}^T \mathbf{C}^{-1} \mathbf{\Delta}.$$

The point of the exercise is that every step is transparent: students can see how the theoretical model is turned into a surface in parameter space.

Problem 4. General Fisher formula

Question. Explain in words the meaning of the two terms in the Gaussian Fisher matrix. Give one example of a cosmological observable for which each term is important.

Solution. The Gaussian Fisher matrix is

$$F_{ij} = \frac{1}{2}\,\mathrm{Tr}\big(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,i}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,j}\big) + \boldsymbol{\mu}_{,i}^{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{,j}.$$

The two terms answer two different questions.

First term: information from the covariance.

$$\frac{1}{2}\,\mathrm{Tr}\big(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,i}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,j}\big)$$

measures how the *width and shape of the data distribution* change when the parameters change. Even if the mean of the data vector stayed fixed, parameters could still be constrained if they altered the covariance. In other words, the experiment can learn from how the variance or correlations respond to the cosmology.

A standard cosmological example is the forecasting of **CMB angular power spectra** or **weak-lensing shear spectra** when the parameter dependence is carried mainly by the covariance of the harmonic modes. There the information is tied to changes in the power spectrum and therefore to changes in the covariance of the Gaussian modes.

Second term: information from the mean signal.

$$\boldsymbol{\mu}_{,i}^{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{,j}$$

measures how the *predicted mean observable* shifts when the parameters change. If the covariance is fixed but the model prediction moves in data space, that motion constrains the parameters. This is the term most students first encounter.

A standard example is **Type Ia supernova distance-modulus data**. The covariance is often treated as fixed, while the theoretical mean vector $\boldsymbol{\mu}(z_i;\boldsymbol{\theta})$ changes with $(\Omega_{\mathrm{m}},\Omega_{\Lambda})$. The same logic applies to many BAO distance points or $f\sigma_8$ measurements.

Bottom line. The trace term says *the noise model itself carries information.* The mean term says *the predicted signal carries information.* Depending on the observable, one term may dominate, or both may matter.

Problem 5. Forecast versus inference

Question. In one page or less, compare a Fisher forecast with a full likelihood analysis. State one advantage and one limitation of each.

Solution. A **full likelihood analysis** and a **Fisher forecast** are related, but they answer different questions.

Full likelihood analysis. Here we take actual data and evaluate the likelihood over parameter space, often with a grid, optimization, or MCMC. The goal is inference: what do the data really prefer? A full analysis can capture non-Gaussian posteriors, curved degeneracies, nuisance parameters, parameter bounds, and multimodality.

Its main **advantage** is realism: if the posterior is not well approximated by a Gaussian, the full likelihood still describes it correctly. Its main **limitation** is cost: exploring a high-dimensional likelihood can be computationally expensive, especially when each model evaluation is slow.

Fisher forecast. The Fisher matrix instead approximates the likelihood locally around a chosen fiducial model. It is therefore mainly a forecasting tool: if an experiment behaves as expected near this fiducial cosmology, how well should the parameters be measured? The result is a covariance matrix estimate,

$$\mathbf{C}_\theta \approx \mathbf{F}^{-1},$$

which is quick to compute and easy to combine across independent datasets.

Its main **advantage** is speed and transparency. It is excellent for survey design, intuition about parameter degeneracies, and rapid comparison of experimental setups. Its main **limitation** is the Gaussian approximation: if the true posterior is highly curved, bounded, skewed, or multimodal, the Fisher ellipse can be misleading.

Comparison. A good summary is:

- **Full fit:** inference from actual data; slower, but more faithful to the true posterior.
- **Fisher forecast:** local Gaussian prediction near a fiducial point; very fast, but only approximate.

In practice, cosmologists often use Fisher matrices early for survey planning, then switch to full likelihood methods once real data or strong non-Gaussianities matter.

Problem 6. Optional coding task (Julia)

Question. Build a toy likelihood in two parameters, evaluate it on a grid, and compare the exact contour shape with the ellipse implied by the Fisher matrix at the best-fit point.

Solution. A useful toy model is a mildly curved ("banana-shaped") likelihood,

$$\chi^2(x, y) = x^2 + \frac{\left(y - 0.45x^2\right)^2}{0.6^2}.$$

Its minimum is at $(x, y) = (0, 0)$, but away from the minimum the contours bend upward because of the x^2 term inside the second factor. A Fisher approximation only uses the local curvature at the minimum, so it produces an *ellipse* there. The comparison shows exactly what the lecture emphasized: near the best fit the ellipse is reasonable, while at larger $\Delta\chi^2$ the true contour becomes non-elliptical.

The following Julia script evaluates the toy likelihood on a grid. It then computes the numerical Hessian at the best-fit point and converts it into a Fisher matrix. Finally, it overlays the Fisher ellipses on the exact contours. The same code is also saved separately as

`lecture9_toy_likelihood_fisher.jl`

```
using LinearAlgebra
using Plots

chi2(x, y) = x^2 + ((y - 0.45 * x^2) / 0.6)^2
nll(x, y) = 0.5 * chi2(x, y)

x0 = 0.0
y0 = 0.0
h = 1.0e-3

# Numerical Hessian of chi^2 / 2 at the best-fit point.
F11 = (nll(x0 + h, y0) - 2nll(x0, y0) + nll(x0 - h, y0)) / h^2
F22 = (nll(x0, y0 + h) - 2nll(x0, y0) + nll(x0, y0 - h)) / h^2
```

```
F12 = (nll(x0 + h, y0 + h) - nll(x0 + h, y0 - h)
     - nll(x0 - h, y0 + h) + nll(x0 - h, y0 - h)) / (4h^2)

F = [F11 F12; F12 F22]
C = inv(F)

function ellipse_points(C, deltachi2; n = 400)
    eig = eigen(Symmetric(C))
    order = sortperm(eig.values; rev = true)
    vals = eig.values[order]
    vecs = eig.vectors[:, order]

    t = range(0, 2pi; length = n)
    circle = [cos.(t)'; sin.(t)']
    axes = Diagonal(sqrt.(vals .* deltachi2))
    pts = vecs * axes * circle
    return pts[1, :], pts[2, :]
end

xs = range(-2.0, 2.0; length = 401)
ys = range(-1.2, 2.8; length = 401)
chi = [chi2(x, y) for y in ys, x in xs]

x68, y68 = ellipse_points(C, 2.30)
x95, y95 = ellipse_points(C, 6.18)

plt = contour(xs, ys, chi;
    levels = [2.30, 6.18],
    aspect_ratio = :equal,
    xlabel = "x",
    ylabel = "y",
    linewidth = 2,
    label = false)
plot!(plt, x68, y68; linewidth = 2, linestyle = :dash, label = "Fisher 68%")
plot!(plt, x95, y95; linewidth = 2, linestyle = :dashdot, label = "Fisher 95%")
scatter!(plt, [x0], [y0]; markersize = 4, label = "best fit")

savefig(plt, "lecture9_toy_likelihood_fisher.pdf")
```

Near the origin, the exact contour and Fisher ellipse are similar because every smooth likelihood looks quadratic sufficiently close to its minimum. Farther away, however, the exact contours bend while the Fisher result stays elliptical. That is the whole lesson of the exercise: the Fisher matrix is a *local* approximation, not a guarantee of global accuracy.