Lecture 9: Fundamentals of Observational Cosmology (III)

Gong-Bo Zhao

Abstract

This lecture combines two ideas that students often meet for the first time in observational cosmology: parameter estimation from supernova data and the Fisher information matrix. We first discuss how a full χ^2 fit works for Type Ia supernovae, then derive the general Fisher-matrix formula and explain the difference between fitting real data and forecasting future constraints.

Learning goals

After this lecture, students should be able to:

- write the supernova χ^2 with a full covariance matrix;
- understand why covariance matrices matter for real datasets;
- explain the formal definition of the Fisher matrix;
- specialize the general Fisher formula to supernova distance-modulus data;
- distinguish a full likelihood analysis from a local Gaussian forecast.

1 Parameter estimation with Type Ia supernovae

A supernova compilation usually provides

- redshifts z_i,
- observed distance moduli $\mu_{\mathrm{obs},i}$,
- a covariance matrix $\mathbf{C}$.

Given a cosmological model with parameters $\boldsymbol{\theta}$, we compute

$$\mu_{\mathrm{th},i} = \mu_{\mathrm{th}}(z_i; \boldsymbol{\theta}).$$

The natural residual vector is

$$\boldsymbol{\Delta} \equiv \boldsymbol{\mu}_{\mathrm{th}} - \boldsymbol{\mu}_{\mathrm{obs}}.$$

For Gaussian errors,

$$\chi^2(\boldsymbol{\theta}) = \boldsymbol{\Delta}^T \mathbf{C}^{-1} \boldsymbol{\Delta}.$$

The maximum-likelihood estimate is obtained by minimizing χ^2.

A practical warning. Students often begin with diagonal error bars only. That is useful for learning, but real supernova datasets include correlated systematics, so the full covariance matrix should be used whenever it is provided.

2 The role of nuisance parameters

In realistic supernova work, the absolute magnitude M is a nuisance parameter, and it is partly degenerate with H_0. This is because the observable is the distance modulus

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

Changing H_0 rescales the luminosity distance, while changing M shifts the zero point of the Hubble diagram. Without external calibration, supernovae alone constrain relative distances much better than they constrain an absolute scale.

3 Grid-based fitting

For a two-parameter example such as $(\Omega_\mathrm{m}, \Omega_\Lambda)$, a grid-based fit is conceptually transparent:

1. choose a grid of trial values;
2. compute $\mu_\mathrm{th}(z_i)$ at each point;
3. evaluate χ^2;
4. find the minimum χ^2_min;
5. convert $\Delta\chi^2 \equiv \chi^2 - \chi^2_\mathrm{min}$ into confidence contours.

This approach becomes slow in high dimensions, but it is excellent for teaching because every step is visible.

4 Confidence regions for two parameters

For an approximately Gaussian likelihood in two parameters, the standard confidence levels are

$$\Delta\chi^2 = 2.30, \qquad 6.18, \qquad 11.83$$

for 68.3%, 95.4%, and 99.7% confidence, respectively. The resulting contours are often elongated because distance data constrain some parameter combinations much better than others.

5 The Fisher information matrix

The Fisher matrix technique is widely used in cosmology for forecasting the uncertainty of cosmological parameters. In essence, it propagates the uncertainty in the observables into expected parameter errors around a chosen fiducial model. A classic reference is Tegmark, Taylor, and Heavens [1].

The formal definition of the Fisher matrix is

$$F_{ij} \equiv -\left\langle \frac{\partial^2 \ln L}{\partial\theta_i \, \partial\theta_j} \right\rangle.$$

Equivalently, under the usual regularity conditions,

$$F_{ij} = \left\langle \frac{\partial \ln L}{\partial\theta_i} \frac{\partial \ln L}{\partial\theta_j} \right\rangle.$$

It measures how sharply the likelihood changes with the parameters. A sharper likelihood means more information and therefore smaller expected parameter errors.

6 General Gaussian Fisher matrix

For a Gaussian data vector $\boldsymbol{x}$ with mean $\boldsymbol{\mu}(\boldsymbol{\theta})$ and covariance $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, the likelihood is

$$L = \frac{\exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]}{\sqrt{(2\pi)^k\,|\boldsymbol{\Sigma}|}}.$$

Dropping the additive constant associated with the $(2\pi)^k$ factor, one may write

$$-2\ln L = \ln\det\boldsymbol{\Sigma} + (\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) + \text{constant}.$$

Define

$$\boldsymbol{\Sigma}_{,i} \equiv \frac{\partial\boldsymbol{\Sigma}}{\partial\theta_i}, \qquad \boldsymbol{\mu}_{,i} \equiv \frac{\partial\boldsymbol{\mu}}{\partial\theta_i}.$$

Then the Fisher matrix is

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

Why are there two terms?

- The first term is present when the covariance depends on the parameters.
- The second term is present when the mean prediction depends on the parameters.

7 Which term matters for which observable?

For many low-dimensional background probes such as supernova distances, BAO distance measurements, or $f\sigma_8$ points, the covariance is often treated as fixed and the mean carries the parameter dependence. Then the Fisher matrix reduces to

$$F_{ij} = \boldsymbol{\mu}_{,i}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{,j}.$$

For observables such as power spectra or CMB angular spectra, the covariance itself may contain much of the parameter dependence. So the first term can be important there.

The following table summarizes the cases most commonly encountered in cosmological data analysis.

Experiment	Observable	$\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,i}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{,j}\right)$	$\boldsymbol{\mu}_{,i}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{,j}$
SNe	Luminosity distance	0	✓
BAO	$H(z), D_A(z)$	0	✓
RSD	$f\sigma_8(z)$	0	✓
CMB	C_ℓ angular power spectra	✓	0
Galaxy clustering	$P(k)$	✓	0
Weak lensing	$C_\ell^{\gamma\gamma}$ shear spectra	✓	0

8 Fisher matrix versus full inference

The Fisher matrix is mainly a **forecast** tool. It approximates the likelihood as locally Gaussian near a chosen fiducial model. By contrast, a full χ^2 or MCMC analysis is an **inference** tool applied to actual data.

So the two methods answer slightly different questions:

- **Full fit:** What do the current data prefer?
- **Fisher forecast:** If the experiment behaves as expected near a fiducial cosmology, how well should it constrain the parameters?

9 A minimal computational workflow

A simple supernova fitting script can be summarized as

```
read z_i, mu_obs_i, covariance C
invert C once
for each parameter point theta:
    compute mu_th(z_i; theta)
    diff = mu_th - mu_obs
    chi2 = diff^T C^{-1} diff
store chi2(theta)
find chi2_min and draw Delta-chi2 contours
```

The Fisher calculation then adds numerical derivatives of μ_{th} with respect to the parameters.

10 Summary

This lecture introduced two connected but distinct ideas. The full χ^2 surface is the correct starting point for fitting supernova data, while the Fisher matrix is a local quadratic approximation useful for forecasts. Both will reappear repeatedly in later cosmological data analyses.

Suggested reading

- Dodelson and Schmidt on likelihood-based inference.
- Tegmark, Taylor, and Heavens on the Fisher matrix.
- Introductory supernova-cosmology notes emphasizing covariance matrices.

Homework

1. **Matrix χ^2.** 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.
2. **H_0–M degeneracy.** 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.
3. **Grid fitting.** Design a toy grid-based fit for $(\Omega_{\mathrm{m}}, \Omega_\Lambda)$ using five mock supernovae. You may describe the algorithm in pseudocode or implement it in a language of your choice.

4. **General Fisher formula.** 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.

5. **Forecast versus inference.** In one page or less, compare a Fisher forecast with a full likelihood analysis. State one advantage and one limitation of each.

6. **Optional coding task.** 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.

A Julia code for SN Ia fitting

The following Julia example follows the same logic as the classroom fitting exercise, but in a more modern style. It reads the supernova redshifts and distance moduli, reads the inverse covariance matrix, scans a grid in $(\Omega_\mathrm{M}, \Omega_\Lambda)$, computes the theoretical distance modulus, and writes out the corresponding χ^2 values.

```
using LinearAlgebra
using DelimitedFiles
using QuadGK

const c = 3.0e5 # km/s
const H0 = 70.0 # km/s/Mpc
const DH = c / H0 # Mpc

const Om_min = 0.0
const Om_max = 3.0
const OL_min = 0.0
const OL_max = 3.0
const Np = 100

"""
Read the Union2.1-style SN file.
Assumes columns: name, z, mu, ..., ...
"""
function read_sn_data(filename::AbstractString)
    raw = readdlm(filename)
    z = Float64.(raw[:, 2])
    mu = Float64.(raw[:, 3])
    return z, mu
end

"""
Read the inverse covariance matrix.
"""
function read_invcov(filename::AbstractString)
    invcov = Float64.(readdlm(filename))
    return invcov
end

E2(z, Om, OL) = Om * (1 + z)^3 + OL + (1 - Om - OL) * (1 + z)^2

Einv(z, Om, OL) = 1.0 / sqrt(E2(z, Om, OL))

function DC(z, Om, OL; rtol=1e-6)
    integral, _ = quadgk(zp -> Einv(zp, Om, OL), 0.0, z, rtol=rtol)
```

```
    return DH * integral
end

function DA(z, Om, OL)
    Ok = 1.0 - Om - OL
    chi = DC(z, Om, OL) / DH

    if Ok > 0
        return DH / (1 + z) / sqrt(Ok) * sinh(sqrt(Ok) * chi)
    elseif Ok < 0
        return DH / (1 + z) / sqrt(abs(Ok)) * sin(sqrt(abs(Ok)) * chi)
    else
        return DC(z, Om, OL) / (1 + z)
    end
end

Ldis(z, Om, OL) = DA(z, Om, OL) * (1 + z)^2

mu_theory(z, Om, OL) = 5.0 * log10(Ldis(z, Om, OL)) + 25.0

function check_model(Om, OL; zmin=0.0, zmax=3.0, nz=500)
    zgrid = range(zmin, zmax; length=nz)
    for z in zgrid
        e2 = E2(z, Om, OL)
        if !(e2 > 0.0)
            return false
        end
        da = DA(z, Om, OL)
        if !(da > 0.0)
            return false
        end
    end
    return true
end

function fit_sn_grid(snfile::AbstractString, covfile::AbstractString;
                     outfile::AbstractString="chi2.dat")
    z_dat, mu_dat = read_sn_data(snfile)
    invcov = read_invcov(covfile)
    n = length(z_dat)

    open(outfile, "w") do io
        for iOm in 1:Np
            Om = Om_min + (Om_max - Om_min) * (iOm - 1) / (Np - 1)
            for iOL in 1:Np
                OL = OL_min + (OL_max - OL_min) * (iOL - 1) / (Np - 1)

                if check_model(Om, OL)
                    diff = [mu_theory(z_dat[i], Om, OL) - mu_dat[i] for i in 1:n]
                    chi2 = dot(diff, invcov * diff)
                else
                    chi2 = 1.0e10
                end

                println(io, Om, ' ', OL, ' ', chi2)
            end
        end
    end
```

```
end

# Example usage:
# fit_sn_grid("sn_z_mu_dmu_plow_union2.1.txt", "sn_wmat_nosys_union2.1.txt")
```

References

[1] M. Tegmark, A. Taylor and A. Heavens, "Karhunen-Loeve eigenvalue problems in cosmology: How should we tackle large data sets?," *Astrophysical Journal* **480**, 22 (1997).

[2] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical Recipes in FORTRAN: The Art of Scientific Computing."