

## Problem 1: Conditional Gaussian Distributions

We are given a partitioned Gaussian vector  $\mathbf{x}$  with mean  $\boldsymbol{\mu}$  and covariance  $\boldsymbol{\Sigma}$ :

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \\ \mathbf{x}_c \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \\ \boldsymbol{\mu}_c \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} & \Sigma_{ac} \\ \Sigma_{ba} & \Sigma_{bb} & \Sigma_{bc} \\ \Sigma_{ca} & \Sigma_{cb} & \Sigma_{cc} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} & \Lambda_{ac} \\ \Lambda_{ba} & \Lambda_{bb} & \Lambda_{bc} \\ \Lambda_{ca} & \Lambda_{cb} & \Lambda_{cc} \end{pmatrix}.$$

Since  $\mathbf{x}_c$  is marginalized out, the joint distribution of  $(\mathbf{x}_a, \mathbf{x}_b)$  depends only on the corresponding sub-blocks of  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ . The marginal is:

$$\begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \sim \mathcal{N} \left( \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix}, \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \right).$$

However, to find the conditional  $p(\mathbf{x}_a | \mathbf{x}_b)$ , we work with the precision matrix of this marginal distribution. Note that the inverse of the marginal covariance matrix is the corresponding sub-block of the joint precision matrix  $\boldsymbol{\Lambda}$  only if the remaining variables are independent, which is not generally true.

Let us expand the quadratic form in the exponent of the Gaussian for the joint  $(\mathbf{x}_a, \mathbf{x}_b)$  distribution. Let the precision of this marginal be denoted by the block matrix  $\tilde{\boldsymbol{\Lambda}}$ :

$$\tilde{\boldsymbol{\Lambda}} = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}^{-1}.$$

For notation in this specific derivation, we assume the problem implies working directly with the partitions of the precision matrix provided. We expand the exponent:

$$\begin{aligned} Q &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\mathbf{x} - \boldsymbol{\mu}) \\ &= -\frac{1}{2} \begin{pmatrix} \mathbf{x}_a - \boldsymbol{\mu}_a \\ \mathbf{x}_b - \boldsymbol{\mu}_b \end{pmatrix}^\top \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix} \begin{pmatrix} \mathbf{x}_a - \boldsymbol{\mu}_a \\ \mathbf{x}_b - \boldsymbol{\mu}_b \end{pmatrix} \\ &= -\frac{1}{2}(\mathbf{x}_a - \boldsymbol{\mu}_a)^\top \Lambda_{aa}(\mathbf{x}_a - \boldsymbol{\mu}_a) - (\mathbf{x}_a - \boldsymbol{\mu}_a)^\top \Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) - \frac{1}{2}(\mathbf{x}_b - \boldsymbol{\mu}_b)^\top \Lambda_{bb}(\mathbf{x}_b - \boldsymbol{\mu}_b). \end{aligned}$$

To find  $p(\mathbf{x}_a | \mathbf{x}_b)$ , we treat  $\mathbf{x}_b$  as constant and complete the square for  $\mathbf{x}_a$ . The terms involving  $\mathbf{x}_a$  are:

$$\begin{aligned} Q(\mathbf{x}_a) &= -\frac{1}{2}\mathbf{x}_a^\top \Lambda_{aa}\mathbf{x}_a + \mathbf{x}_a^\top \Lambda_{aa}\boldsymbol{\mu}_a - \mathbf{x}_a^\top \Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) + \dots \\ &= -\frac{1}{2}\mathbf{x}_a^\top \Lambda_{aa}\mathbf{x}_a + \mathbf{x}_a^\top (\Lambda_{aa}\boldsymbol{\mu}_a - \Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)) + \text{const.} \end{aligned}$$

Comparing this to the standard quadratic form  $-\frac{1}{2}(\mathbf{x}_a - \boldsymbol{\mu}_{a|b})^\top \boldsymbol{\Sigma}_{a|b}^{-1}(\mathbf{x}_a - \boldsymbol{\mu}_{a|b})$ , we identify:

$$\boldsymbol{\Sigma}_{a|b}^{-1} = \Lambda_{aa} \implies \boldsymbol{\Sigma}_{a|b} = \Lambda_{aa}^{-1}.$$

And the linear term coefficient is  $\boldsymbol{\Sigma}_{a|b}^{-1}\boldsymbol{\mu}_{a|b}$ . Thus:

$$\begin{aligned} \Lambda_{aa}\boldsymbol{\mu}_{a|b} &= \Lambda_{aa}\boldsymbol{\mu}_a - \Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ \boldsymbol{\mu}_{a|b} &= \boldsymbol{\mu}_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b). \end{aligned}$$

Thus, the conditional distribution is:

$$p(\mathbf{x}_a | \mathbf{x}_b) = \mathcal{N} \left( \mathbf{x}_a | \boldsymbol{\mu}_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b), \Lambda_{aa}^{-1} \right).$$

## Problem 2: The Woodbury Matrix Identity

We wish to verify the identity:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}.$$

Let  $X = (A + BCD)$ . We prove this by showing  $X \cdot X^{-1} = I$ . Let the RHS be  $Y$ .

$$\begin{aligned} XY &= (A + BCD) \left[ A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \right] \\ &= (A + BCD)A^{-1} - (A + BCD)A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ &= (I + BCDA^{-1}) - (B + BCDA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}DA^{-1}. \end{aligned}$$

Factor out  $B$  from the second term:

$$\begin{aligned} &= I + BCDA^{-1} - B(I + CDA^{-1}B)(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ &= I + BCDA^{-1} - B \left[ C(C^{-1} + DA^{-1}B) \right] (C^{-1} + DA^{-1}B)^{-1}DA^{-1} \\ &= I + BCDA^{-1} - BC \underbrace{\left[ (C^{-1} + DA^{-1}B)(C^{-1} + DA^{-1}B)^{-1} \right]}_I DA^{-1} \\ &= I + BCDA^{-1} - BCDA^{-1} \\ &= I. \end{aligned}$$

This confirms the identity.

**Computational Efficiency:** Consider  $\mathbf{A}$  as an  $n \times n$  diagonal matrix,  $\mathbf{B}$  is  $n \times k$ ,  $\mathbf{C}$  is  $k \times k$ , and  $\mathbf{D}$  is  $k \times n$ , where  $k \ll n$ .

- Naively inverting  $(A + BCD)$  requires  $O(n^3)$  operations.
- Using Woodbury, we only invert  $(C^{-1} + DA^{-1}B)$ , which is a  $k \times k$  matrix. This costs  $O(k^3)$ .
- Since  $A$  is diagonal,  $A^{-1}$  is trivial  $O(n)$ .
- The matrix multiplications cost approx  $O(nk^2)$ .

Thus, when  $k \ll n$ , the complexity drops from cubic in  $n$  to linear in  $n$ .

## Problem 3: Bayesian Inference for Gaussian (Unknown Mean and Precision)

Likelihood for  $N$  i.i.d samples with precision  $\lambda = 1/\sigma^2$ :

$$p(\mathbf{X} \mid \mu, \lambda) \propto \lambda^{N/2} \exp \left( -\frac{\lambda}{2} \sum_{n=1}^N (x_n - \mu)^2 \right).$$

The conjugate prior is Normal-Gamma:

$$p(\mu, \lambda) = \mathcal{N}(\mu \mid \mu_0, (\beta\lambda)^{-1}) \text{Gam}(\lambda \mid a, b) \propto \lambda^{1/2} e^{-\frac{\beta\lambda}{2}(\mu - \mu_0)^2} \lambda^{a-1} e^{-b\lambda}.$$

Combining terms in the prior:

$$p(\mu, \lambda) \propto \lambda^{a-1/2} \exp \left( -\frac{\lambda}{2} [\beta(\mu - \mu_0)^2 + 2b] \right).$$

Multiplying Prior  $\times$  Likelihood:

$$p(\mu, \lambda | \mathbf{X}) \propto \lambda^{a-1/2+N/2} \exp \left( -\frac{\lambda}{2} [\beta(\mu - \mu_0)^2 + 2b + \sum (x_n - \mu)^2] \right).$$

We complete the square for  $\mu$  inside the exponent. Let  $\bar{x} = \frac{1}{N} \sum x_n$ .

$$\sum (x_n - \mu)^2 = \sum x_n^2 - 2N\bar{x}\mu + N\mu^2.$$

Total term in bracket:

$$\begin{aligned} [\dots] &= \beta(\mu^2 - 2\mu\mu_0 + \mu_0^2) + 2b + \sum x_n^2 - 2N\bar{x}\mu + N\mu^2 \\ &= (\beta + N)\mu^2 - 2(\beta\mu_0 + N\bar{x})\mu + (\beta\mu_0^2 + 2b + \sum x_n^2). \end{aligned}$$

This is quadratic in  $\mu$ , implying a new Gaussian component with precision  $\lambda(\beta + N)$  and mean:

$$\mu_N = \frac{\beta\mu_0 + N\bar{x}}{\beta + N}.$$

The remaining terms form the new Gamma parameters.

$$a_N = a + \frac{N}{2}, \quad b_N = b + \frac{1}{2} \sum x_n^2 + \frac{\beta}{2} \mu_0^2 - \frac{\beta + N}{2} \mu_N^2.$$

This confirms the posterior is Normal-Gamma.

## Problem 4: The Wishart Distribution

### 4a. Conjugacy Proof

The Wishart prior for precision matrix  $\Lambda$  is:

$$\mathcal{W}(\Lambda | \mathbf{W}, \nu) \propto |\Lambda|^{(\nu-D-1)/2} \exp \left( -\frac{1}{2} \text{Tr}(\mathbf{W}^{-1}\Lambda) \right).$$

The likelihood for Multivariate Gaussian data  $\mathbf{X}$  is:

$$p(\mathbf{X} | \mu, \Lambda) \propto |\Lambda|^{N/2} \exp \left( -\frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \mu)^\top \Lambda (\mathbf{x}_n - \mu) \right).$$

Using the trace trick  $\mathbf{x}^\top \mathbf{A} \mathbf{x} = \text{Tr}(\mathbf{A} \mathbf{x} \mathbf{x}^\top)$ :

$$\sum (\mathbf{x}_n - \mu)^\top \Lambda (\mathbf{x}_n - \mu) = \text{Tr} \left( \Lambda \sum (\mathbf{x}_n - \mu) (\mathbf{x}_n - \mu)^\top \right) = \text{Tr}(\Lambda \cdot \mathbf{NS}).$$

Multiplying Prior  $\times$  Likelihood:

$$\begin{aligned} p(\Lambda | \mathbf{X}) &\propto |\Lambda|^{(\nu-D-1)/2} |\Lambda|^{N/2} \exp \left( -\frac{1}{2} \text{Tr}(\mathbf{W}^{-1}\Lambda) - \frac{1}{2} \text{Tr}(\mathbf{NS}\Lambda) \right) \\ &= |\Lambda|^{(\nu+N-D-1)/2} \exp \left( -\frac{1}{2} \text{Tr}([\mathbf{W}^{-1} + \mathbf{NS}]\Lambda) \right). \end{aligned}$$

This is functionally identical to the Wishart form with updated parameters:

$$\nu_{\text{new}} = \nu + N, \quad \mathbf{W}_{\text{new}}^{-1} = \mathbf{W}^{-1} + \mathbf{NS}.$$

## 4b. Python Simulation

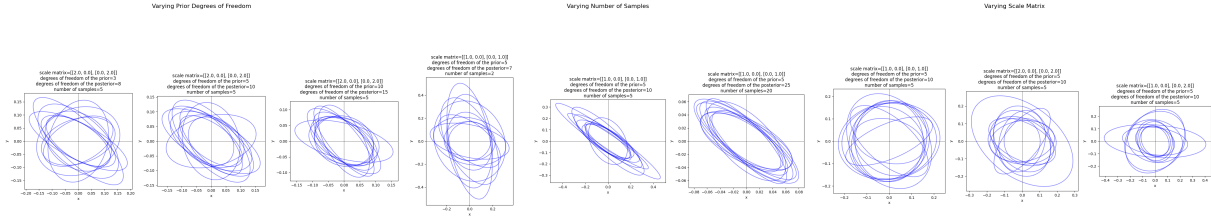


Figure 1: Wishart distributions showing precision ellipses for varying degrees of freedom, sample sizes, and scale matrices.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.stats import wishart
4
5 def generate_data(n, cov):
6     return np.random.multivariate_normal(mean=[0, 0], cov=cov, size=n)
7
8 def sample_wishart(df, scale, num_samples=5):
9     samples = []
10    for _ in range(num_samples):
11        W = wishart.rvs(df=df, scale=scale)
12        samples.append(W)
13    return samples
14
15 def plot_precision_ellipses(precisions, ax, title):
16    for p in precisions:
17        eigvals, eigvecs = np.linalg.eigh(p)
18        if np.any(eigvals <= 0):
19            continue
20
21        angles = np.linspace(0, 2*np.pi, 200)
22        circle = np.stack([np.cos(angles), np.sin(angles)], axis=1)
23        scale_matrix = np.diag(1.0 / np.sqrt(eigvals))
24        ellipse_y = circle @ scale_matrix
25        ellipse_x = ellipse_y @ eigvecs.T
26        ax.plot(ellipse_x[:, 0], ellipse_x[:, 1], 'b', alpha=0.5)
27
28    ax.set_aspect('equal', 'box')
29    ax.set_title(title)
30    ax.set_xlabel('x')
31    ax.set_ylabel('y')
32    ax.axhline(0, color='black', linewidth=0.5)
33    ax.axvline(0, color='black', linewidth=0.5)
34
35 def variation_1_distribution():
36    n = 5
37    scale_matrix = np.eye(2) * 2.0
38    cov = np.array([[1.0, 0.5], [0.5, 1.0]])
39    X = generate_data(n, cov)
40    sum_T = X.T @ X
41    dfs = [3, 5, 10]
42
43    fig, axes = plt.subplots(1, 3, figsize=(15, 4))

```

```

44
45     for i, df_prior in enumerate(dfs):
46         df_post = df_prior + n
47         scale_post = scale_matrix + sum_T
48         posterior_samples = sample_wishart(df_post, scale_post, num_samples=10)
49
50         title = (
51             f"scale_matrix={scale_matrix.tolist()}\n"
52             f"degrees_of_freedom_of_the_prior={df_prior}\n"
53             f"degrees_of_freedom_of_the_posterior={df_post}\n"
54             f"number_of_samples={n}"
55         )
56         plot_precision_ellipses(posterior_samples, axes[i], title)
57
58     fig.suptitle("Varying Prior Degrees of Freedom", fontsize=16)
59     plt.tight_layout()
60     plt.show()
61
62 def variation_2_distribution():
63     df_prior = 5
64     scale_matrix = np.eye(2)
65     cov = np.array([[1.0, 0.5], [0.5, 1.0]])
66     ns = [2, 5, 20]
67     fig, axes = plt.subplots(1, 3, figsize=(15, 4))
68
69     for i, n in enumerate(ns):
70         X = generate_data(n, cov)
71         sum_T = X.T @ X
72         df_post = df_prior + n
73         scale_post = scale_matrix + sum_T
74         posterior_samples = sample_wishart(df_post, scale_post, num_samples=10)
75
76         title = (
77             f"scale_matrix={scale_matrix.tolist()}\n"
78             f"degrees_of_freedom_of_the_prior={df_prior}\n"
79             f"degrees_of_freedom_of_the_posterior={df_post}\n"
80             f"number_of_samples={n}"
81         )
82         plot_precision_ellipses(posterior_samples, axes[i], title)
83
84     fig.suptitle("Varying Number of Samples", fontsize=16)
85     plt.tight_layout()
86     plt.show()
87
88 def variation_3_distribution():
89     df_prior = 5
90     n = 5
91     scale_matrix_list = [np.eye(2), 2.0 * np.eye(2), np.diag([1.0, 2.0])]
92     cov = np.array([[1.0, 0.5], [0.5, 1.0]])
93     X = generate_data(n, cov)
94     sum_T = X.T @ X
95
96     fig, axes = plt.subplots(1, 3, figsize=(15, 4))
97
98     for i, scale_matrix in enumerate(scale_matrix_list):
99         df_post = df_prior + n
100         scale_post = scale_matrix + sum_T
101         posterior_samples = sample_wishart(df_post, scale_post, num_samples=10)
102

```

```

103     title = (
104         f"scale_matrix={scale_matrix.tolist()}\n"
105         f"degrees_of_freedom_of_the_prior={df_prior}\n"
106         f"degrees_of_freedom_of_the_posterior={df_post}\n"
107         f"number_of_samples={n}"
108     )
109     plot_precision_ellipses(posterior_samples, axes[i], title)
110
111 fig.suptitle("Varying Scale Matrix", fontsize=16)
112 plt.tight_layout()
113 plt.show()
114
115 def main():
116     variation_1_distribution()
117     variation_2_distribution()
118     variation_3_distribution()
119
120 if __name__ == "__main__":
121     main()

```

Listing 1: Simulation Code

## Problem 5: Exponential Family Form of Multivariate Gaussian

The Multivariate Gaussian density is:

$$\begin{aligned}
 p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= (2\pi)^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \\
 &= (2\pi)^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right).
 \end{aligned}$$

Standard Exponential Family form:

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta}) \exp(\boldsymbol{\eta}^\top \mathbf{u}(\mathbf{x})).$$

By inspection:

- Sufficient statistics:  $\mathbf{u}(\mathbf{x}) = \begin{pmatrix} \mathbf{x} \\ \text{vec}(\mathbf{x}\mathbf{x}^\top) \end{pmatrix}$ .
- Natural parameters:  $\boldsymbol{\eta} = \begin{pmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \\ -\frac{1}{2}\boldsymbol{\Sigma}^{-1} \end{pmatrix}$ .

Base measure  $h(\mathbf{x}) = (2\pi)^{-D/2}$ .

**Recovering Standard Parameters:** From  $\boldsymbol{\eta}_2 = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}$ , we get  $\boldsymbol{\Sigma}^{-1} = -2\boldsymbol{\eta}_2 \implies \boldsymbol{\Sigma} = -\frac{1}{2}\boldsymbol{\eta}_2^{-1}$ .  
 From  $\boldsymbol{\eta}_1 = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ , we get  $\boldsymbol{\mu} = \boldsymbol{\Sigma}\boldsymbol{\eta}_1 = (-\frac{1}{2}\boldsymbol{\eta}_2^{-1})\boldsymbol{\eta}_1$ .

**Normalization Factor  $g(\boldsymbol{\eta})$ :** The log-partition function logic leads to:

$$\begin{aligned}
 g(\boldsymbol{\eta}) &= |\boldsymbol{\Sigma}|^{-1/2} \exp \left( -\frac{1}{2} \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} \right) \\
 &= | -2\boldsymbol{\eta}_2 |^{D/2} \exp \left( -\frac{1}{2} (\boldsymbol{\Sigma} \boldsymbol{\eta}_1)^\top (-2\boldsymbol{\eta}_2) (\boldsymbol{\Sigma} \boldsymbol{\eta}_1) \right) \\
 &= | -2\boldsymbol{\eta}_2 |^{D/2} \exp \left( \frac{1}{4} \boldsymbol{\eta}_1^\top \boldsymbol{\eta}_2^{-1} \boldsymbol{\eta}_1 \right).
 \end{aligned}$$