Factor Analysis: Posterior Mean Function

October 2025

1 Posterior Expectation of Latent Factors

In Factor Analysis (FA), the observed variable $\mathbf{x} \in \mathbb{R}^d$ is modeled as

$$\mathbf{x} = W^\top \mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \Psi), \quad \mathbf{z} \sim \mathcal{N}(0, I).$$

Here W is the loading matrix, Ψ is a diagonal noise covariance, and $\boldsymbol{\mu}$ is the mean vector. The posterior distribution of the latent variable $\mathbf{z}$ given $\mathbf{x}$ is Gaussian:

$$p(\mathbf{z} \,|\, \mathbf{x}) = \mathcal{N}\big(E[\mathbf{z} \,|\, \mathbf{x}],\, (I + W\Psi^{-1}W^\top)^{-1}\big),$$

where the posterior mean is

$$E[\mathbf{z} \,|\, \mathbf{x}] = (I + W\Psi^{-1}W^\top)^{-1}(\mathbf{x} - \boldsymbol{\mu})\Psi^{-1}W^\top.$$

For a dataset $X \in \mathbb{R}^{n \times d}$, the corresponding matrix form is:

$$F = (X - \boldsymbol{\mu})\Psi^{-1}W^\top(I + W\Psi^{-1}W^\top)^{-1},$$

where $F \in \mathbb{R}^{n \times k}$ contains the latent factor scores for each of the n samples.

2 Implementation in Python

```
W   = fa.components_          # (k, d)
Psi = fa.noise_variance_      # (d,)
mu  = fa.mean_                # (d,)
Wpsi = W / Psi
I    = np.eye(W.shape[0])
G    = np.linalg.inv(I + Wpsi @ W.T)
Xc   = X - mu
F_manual = (Xc @ Wpsi.T) @ G
```

This implementation computes the same posterior mean $E[\mathbf{z}|\mathbf{x}]$ as derived above, producing the $n \times k$ factor score matrix.