

CEE 616: Probabilistic Machine Learning
M5 Unsupervised Learning:
5B: Factor Analysis and Autoencoders

Jimi Oke

UMass**Amherst**

College of Engineering

Dec 4, 2025

Outline

- ① Factor analysis
- ② FA Estimation
- ③ Autoencoders
- ④ AE variants
- ⑤ Outlook

Factor analysis model

- Basic idea: there are latent (hidden) **common factors** \mathbf{z} underlying some multivariate observations $\mathbf{x}_n \in \mathbb{R}^D$

Factor analysis (FA) is a latent variable generative model specified as:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \quad (1)$$

$$p(\mathbf{x} | \mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x} | \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (2)$$

where:

- \mathbf{z} : latent vector
- \mathbf{W} : factor loading matrix, $D \times L$
- $\boldsymbol{\Psi}$: diagonal covariance matrix, $D \times D$

Induced marginal distribution

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)d\mathbf{z} \quad (3)$$

$$= \mathcal{N}(\mathbf{x}|\underbrace{\mathbf{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}}_{\text{mean}}, \underbrace{\boldsymbol{\Psi} + \mathbf{W}\boldsymbol{\Sigma}_0\mathbf{W}^\top}_{\text{variance}}) \quad (4)$$

Simplifications:

- $\boldsymbol{\mu}_0 \rightarrow \mathbf{0}$
- $\boldsymbol{\Sigma}_0 \rightarrow \mathbf{I}$

The simplified marginal distribution then becomes:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi}) \quad (5)$$

Generative model simplified

$$\text{[Prior]} \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}) \quad (6)$$

$$\text{[Likelihood]} \quad p(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\mathbf{x} | \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (7)$$

$$\text{[Evidence/marginal]} \quad p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi}) \quad (8)$$

What FA does

It approximates the covariance matrix of the visible/observed vector \mathbf{x} using a low-rank decomposition:

$$\mathbf{C} = \text{Cov}[\mathbf{x}] = \underbrace{\mathbf{W}\mathbf{W}^\top}_{\text{low-rank decomp}} + \mathbf{\Psi} \quad (9)$$

- $\mathbf{W}\mathbf{W}^\top$ is $D \times D$ (recall: $\mathbf{W} \in \mathbb{R}^{D \times L}$)
- $\mathbf{\Psi}$ is $D \times D$ (restricted to be diagonal)
- For each variable x_d , the **marginal variance** (each diagonal term in \mathbf{C}) is given by:

$$\mathbb{V}[x_d] = \sum_{k=1}^L \underbrace{w_{dk}^2}_{\text{common}} + \underbrace{\psi_d}_{\text{unique}} \quad (10)$$

Parameters to be estimated

The unknown parameters in FA are:

- \mathbf{W} : factor loading matrix
- Ψ : covariance matrix

These can be estimated via:

- Maximum likelihood estimation (MLE)
- Expectation-maximization (EM) algorithm
- Bayesian methods (e.g., variational inference, MCMC)

Once estimated, the **posterior** of latent embeddings is given by:

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{W}^\top \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}), \mathbf{I} - \mathbf{W}^\top \mathbf{C}^{-1} \mathbf{W}) \quad (11)$$

- Posterior has closed-form solution under Gaussian distribution

FA model summary

$$\text{[Prior]} \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}) \quad (12)$$

$$\text{[Likelihood]} \quad p(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\mathbf{x} | \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (13)$$

$$\text{[Evidence/marginal]} \quad p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{C}) \quad (14)$$

$$\text{[Posterior]} \quad p(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z} | \mathbf{W}^\top \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}), \mathbf{I} - \mathbf{W}^\top \mathbf{C}^{-1} \mathbf{W}) \quad (15)$$

- \mathbf{z} : latent vector, length L (assumed to be zero-mean, unit variance)
- \mathbf{x} : observed vector
- \mathbf{W} : $D \times L$ factor loading matrix
- $\boldsymbol{\Psi}$: $D \times D$ diagonal covariance matrix or **matrix of unique variances**
- $\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \boldsymbol{\Psi}$

Unidentifiability of FA parameters

The parameters \mathbf{W} and Ψ are unidentifiable. This can be addressed via:

- Constraining \mathbf{W} to have orthonormal columns [PCA]
- Constraining \mathbf{W} to be lower triangular
- Informative rotation: $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$, where \mathbf{R} is the rotation matrix
 - Commonly used rotations: Varimax, Promax, Oblimin, Geomin, Thurstone, Equamax
- Sparsity-promoting priors on \mathbf{W}
- Non-Gaussian priors for latent factors

PCA as a special case of FA

Principal components analysis (PCA) is a special case of FA with:

$$\Psi = \sigma^2 I \quad (16)$$

where σ^2 is the isotropic noise variance. In PCA, the covariance of the observed vector is given by:

$$\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2 I \quad (17)$$

Autoencoders as nonlinear PCA/FA

In PCA/FA, we learn a **linear mapping** from a high-dimensional observed space $\mathbf{x} \in \mathbb{R}^D$ to a low-dimensional latent space $\mathbf{z} \in \mathbb{R}^L$ and vice-versa.

- **Encoder** f_e : mapping from $\mathbf{x} \rightarrow \mathbf{z}$
- **Decoder** f_d : mapping from $\mathbf{z} \rightarrow \mathbf{x}$

In PCA, for example, f_e is given by:

$$\mathbf{z} = \mathbf{W}^\top \mathbf{x} \equiv f_e(\mathbf{x}) \quad (18)$$

And f_d is given by:

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{z} \equiv f_d(\mathbf{z}) \quad (19)$$

To introduce flexibility, we can specify f_e and f_d are nonlinear/more complex functions. This is best accomplished via neural network, resulting in an **autoencoder** (AE).

Reconstruction loss

The reconstruction function is the approximation of the observation from the decoder:

$$\hat{\mathbf{x}} \equiv r(\mathbf{x}) = f_d(f_e(\mathbf{x})) \quad (20)$$

An autoencoder is thus trained to minimize the reconstruction loss

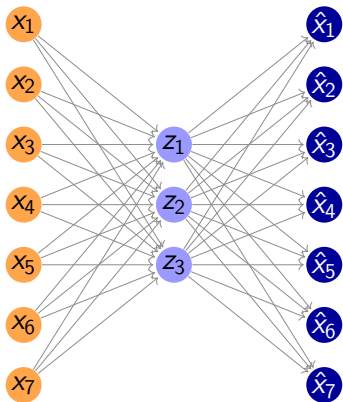
$$\mathcal{L}(\boldsymbol{\theta}) = ||r(\mathbf{x}) - \mathbf{x}||_2^2 \quad (21)$$

or equivalently, the negative log-likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\mathbf{x}|r(\mathbf{x})) \quad (22)$$

Basic autoencoder (AE) architecture

Autoencoder with 2 single-layer MLPS: input layer, hidden layer (latent representation) and output layer (reconstruction)

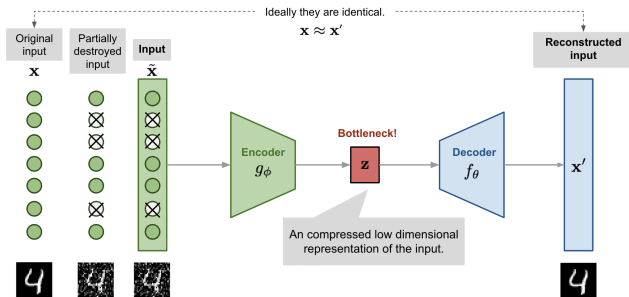


- Hidden layer (size L) is a low-dimensional **bottleneck** between input and reconstruction
- $L \ll D$: undercomplete representation
- $L \gg D$: overcomplete representation (regularize to prevent identity learning)

Denoising autoencoders

In denoising autoencoders (DAEs), the input is corrupted ($\tilde{\mathbf{x}}$) by:

- Gaussian noise: $p_c(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}|\mathbf{x}, \sigma^2 I)$
- Bernoulli dropout: randomly setting a proportion of input nodes to zero



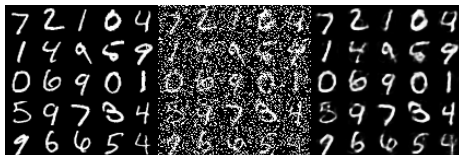
Schematic of a DAE.

Source: <https://lilianweng.github.io/posts/2018-08-12-vae/>

The model is then trained to minimize the loss between the reconstructed input $r(\tilde{\mathbf{x}})$ and its uncorrupted version \mathbf{x}

Uses of DAE

- DAEs are used for denoising images



Original, corrupted and reconstructed images from MNIST dataset.

Source: <http://www.opendeep.org/v0.0.5/docs/tutorial-your-first-model>

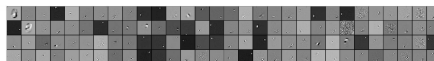
- They can also learn vector fields of input data
- Popular for their simplicity

Sparse autoencoder (SAE)

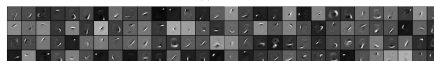
Sparse autoencoder (SAE): sparsity penalty on latent activations

$$\Omega(\mathbf{z}) = \lambda \|\mathbf{z}\|_1 \quad (23)$$

- k*-Sparse autoencoder:** use only *k* largest activations in training



(a) $k = 70$



(b) $k = 40$



(c) $k = 25$



(d) $k = 10$

Filters of the *k*-sparse autoencoder for different sparsity levels *k*, learnt from MNIST with 1000 hidden units.

Source: <https://arxiv.org/pdf/1312.5663.pdf>

Useful for interpretability

Other AEs

- **Contractive autoencoder** (CAE): regularizes via penalty on reconstruction loss

$$\Omega(\mathbf{z}, \mathbf{x}) = \lambda \left\| \left\| \frac{\partial f_e(\mathbf{x})}{\partial \mathbf{x}} \right\|_F \right\|^2 \quad (24)$$

- Variational autoencoder (VAE): probabilistic version of AE/generative model

Reading

- **PMLI 20.3**
- **DL 20**