Latent Variable Models

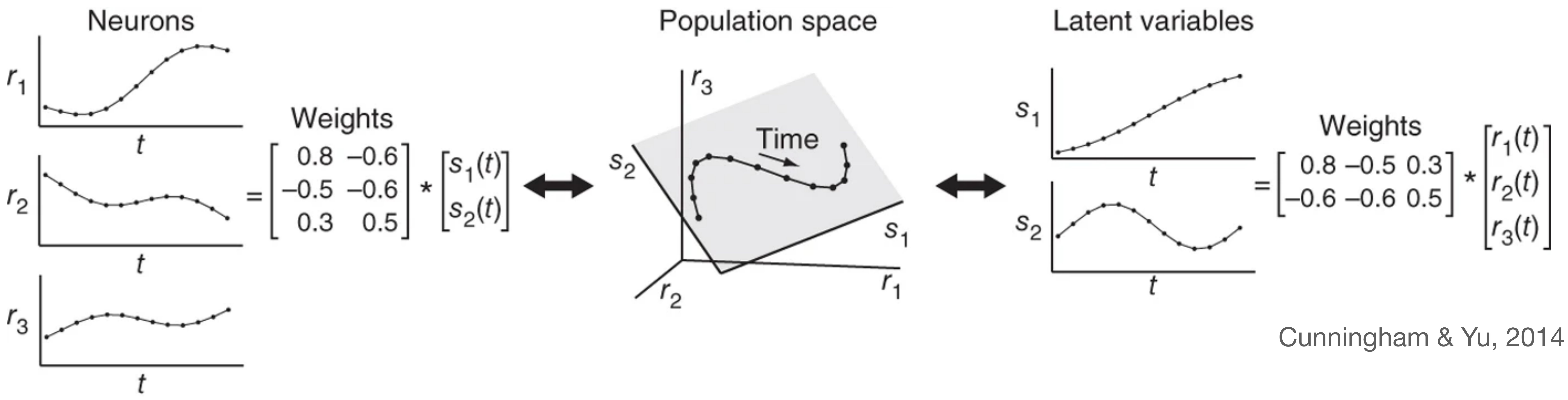
Overview of Latent Variable Models

- **Motivation**: Find “latent” (unobserved) structure in neural population activity

- Latents can be discrete or continuous

Overview of Latent Variable Models

- **Motivation**: Find “latent” (unobserved) structure in neural population activity
- Latents can be discrete or **continuous**

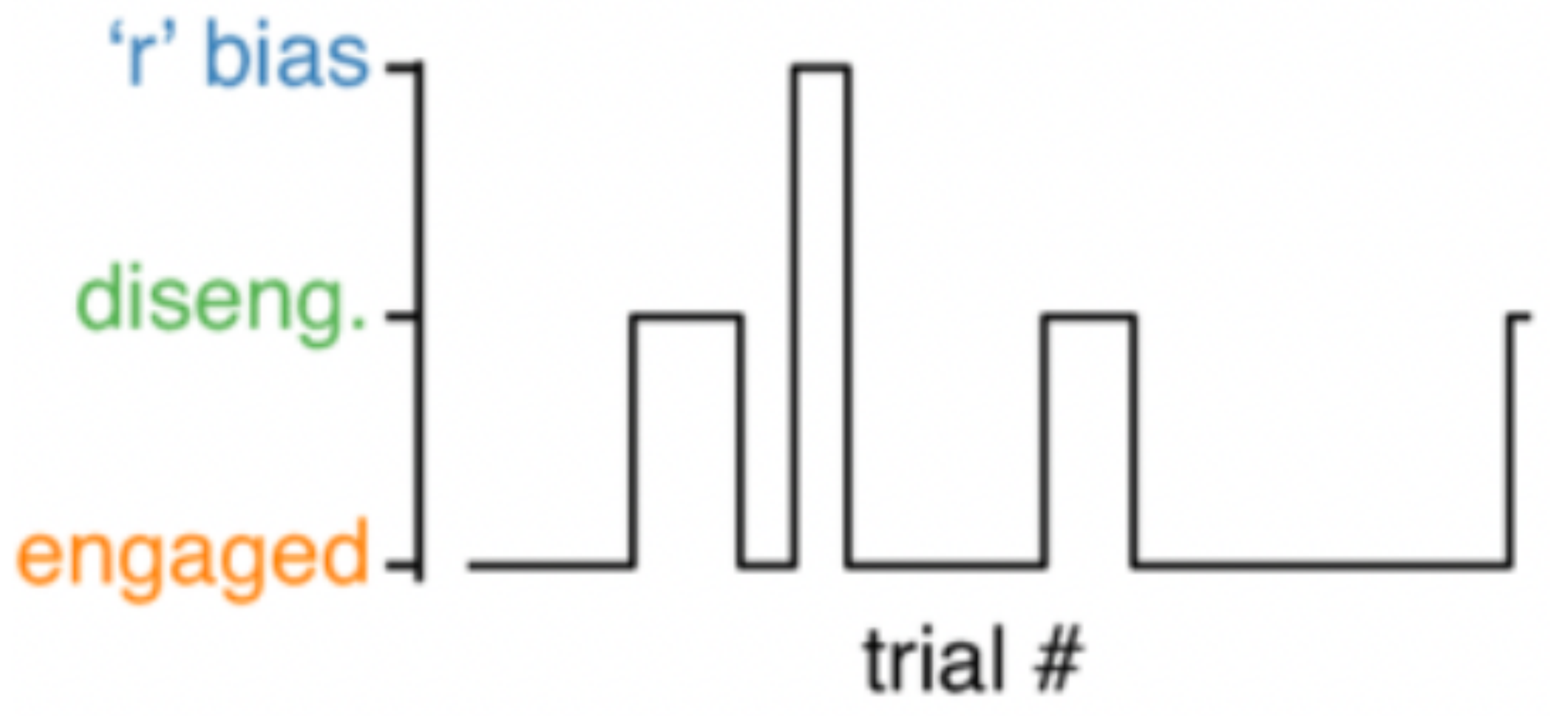


Cunningham & Yu, 2014

Overview of Latent Variable Models

- **Motivation**: Find “latent” (unobserved) structure in neural population activity

- Latents can be **discrete** or continuous

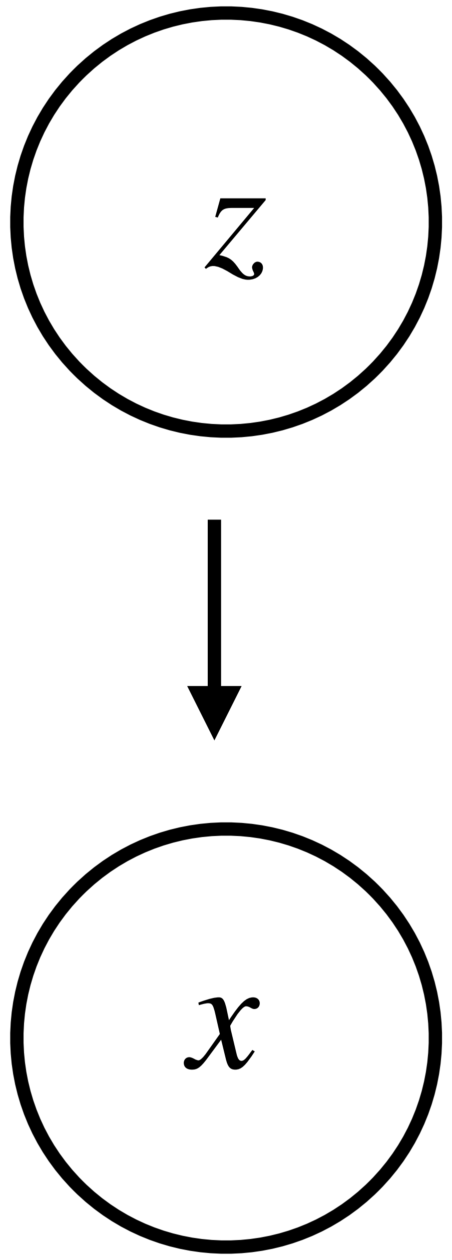


Ashwood et al, 2020

Overview of Latent Variable Models

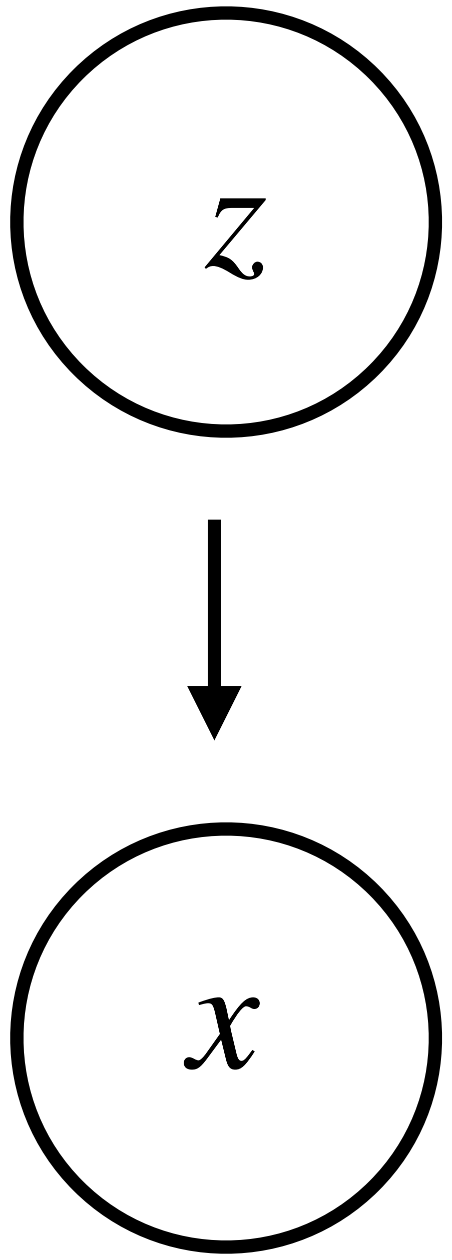
- Class 1: Intro to LVMs, Factor Analysis (FA), Gaussian Processes (GPs)
- Class 2: Hidden Markov Models (HMMs), Linear Dynamical Systems (LDS)
- Class 3: JC on GLM-HMMs and GPFA
- Class 4: Statistical inference: Expectation-Maximization, Markov Chain Monte Carlo (MCMC), Variational Inference
- Class 5: Switching Dynamical Systems, Poisson Linear Dynamical Systems, Variational Autoencoders (VAEs)
- Class 6: JC on LFADS (sequential VAEs), RSLDSs in neural data
- Anything else you'd like covered?

Intro to Latent Variable Models



https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

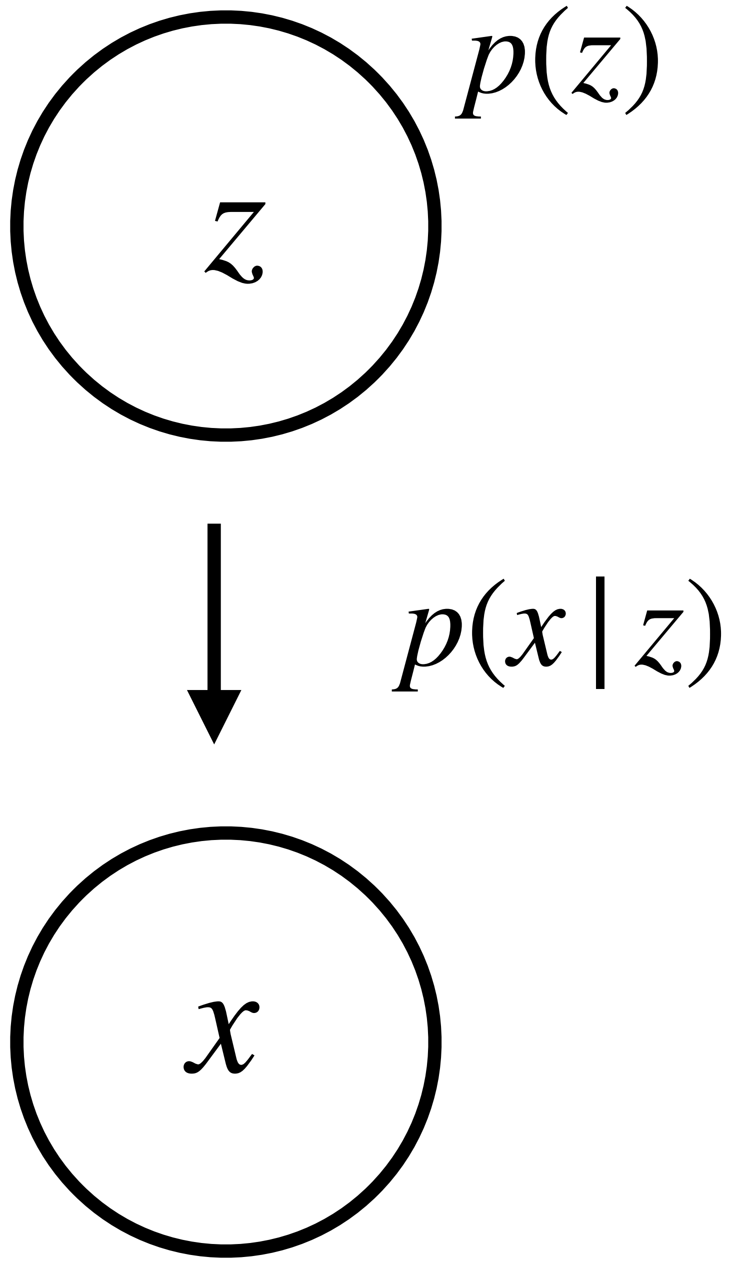
Intro to Latent Variable Models



- Two parts of a LVM
  - Prior: $z \sim p(z)$
  - Conditional probability of observed data: $x \,|\, z \sim p(x \,|\, z)$

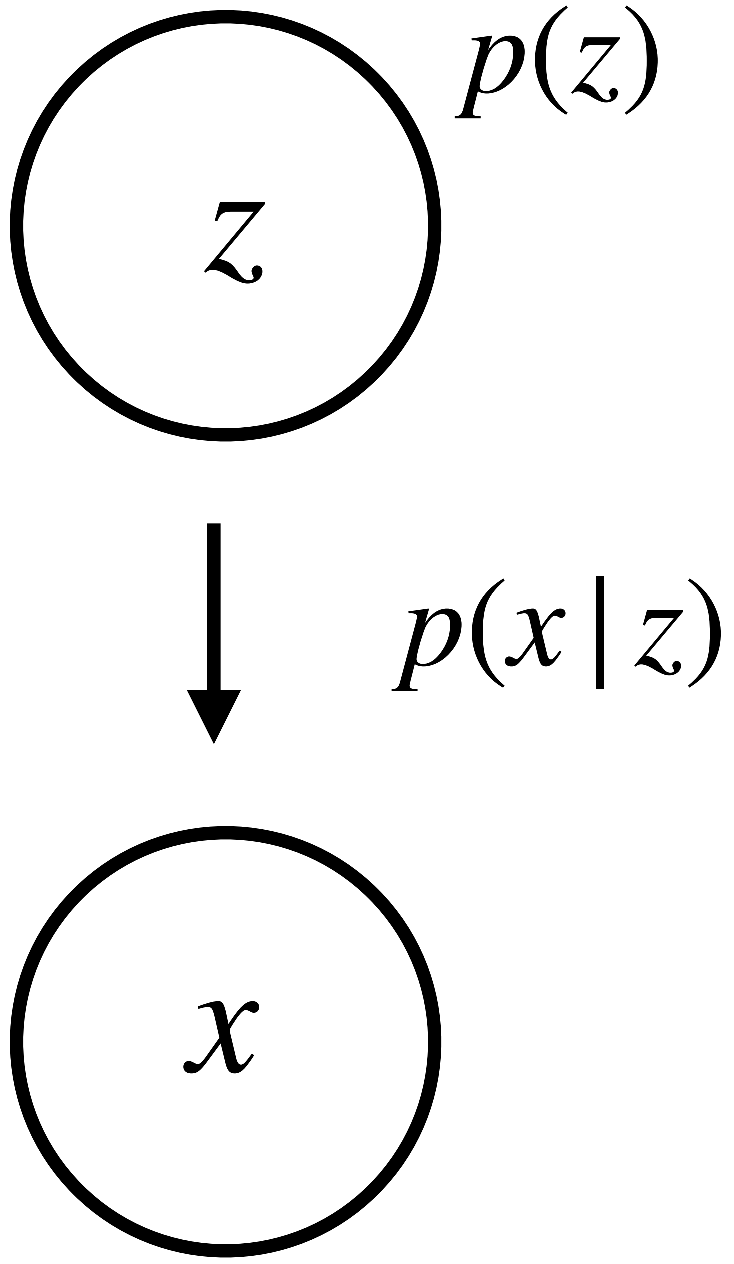
https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

Intro to Latent Variable Models



https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

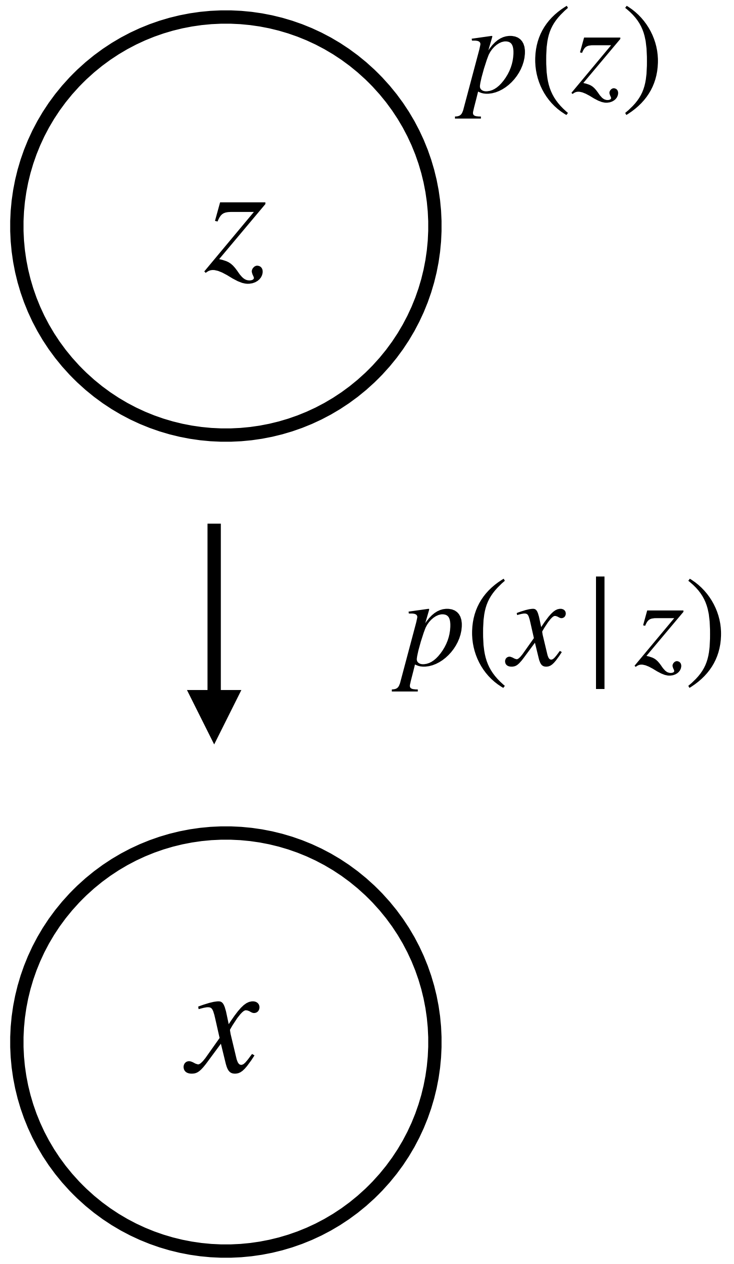
Intro to Latent Variable Models



- Probability of observed data, $p(x)$, is:

https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

Intro to Latent Variable Models

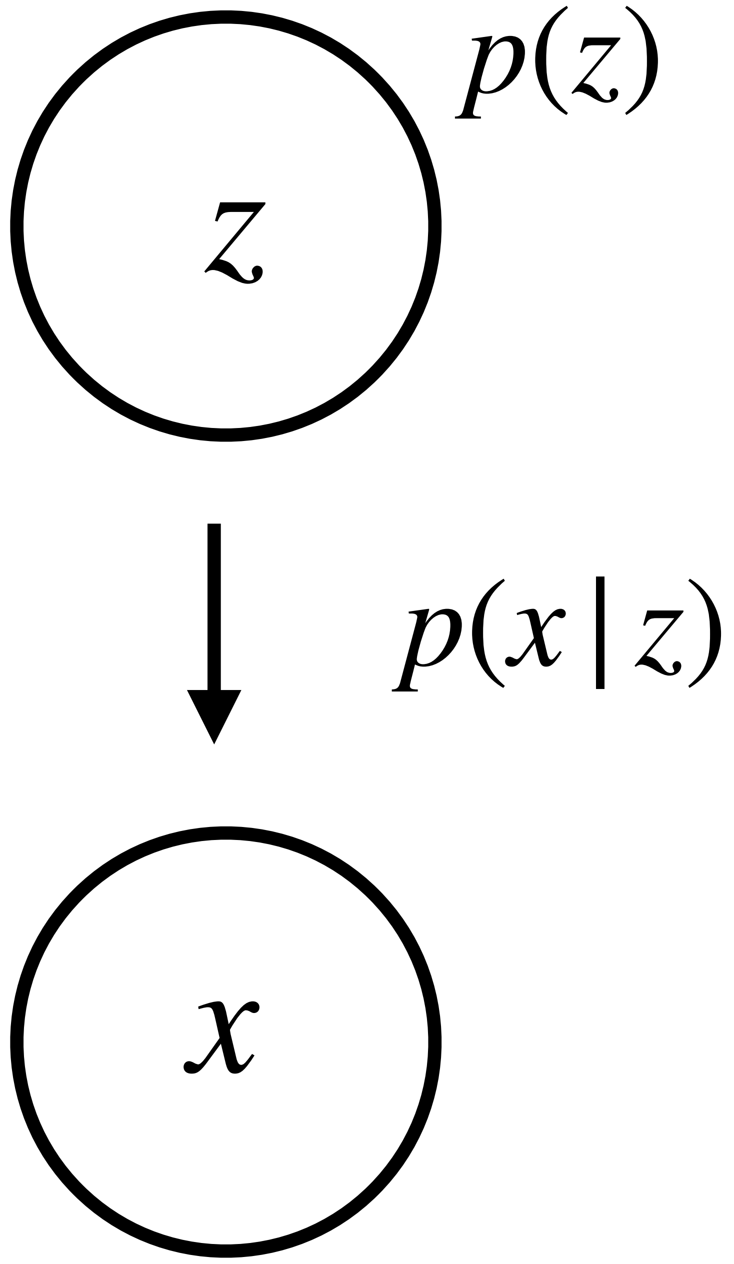


- Probability of observed data, $p(x)$, is:
  - For discrete latents:

$$p(x) = \sum_{i=1}^{m} p(x|z = z_i)p(z = z_i)$$

https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

Intro to Latent Variable Models



- Probability of observed data, $p(x)$, is:
  - For discrete latents:

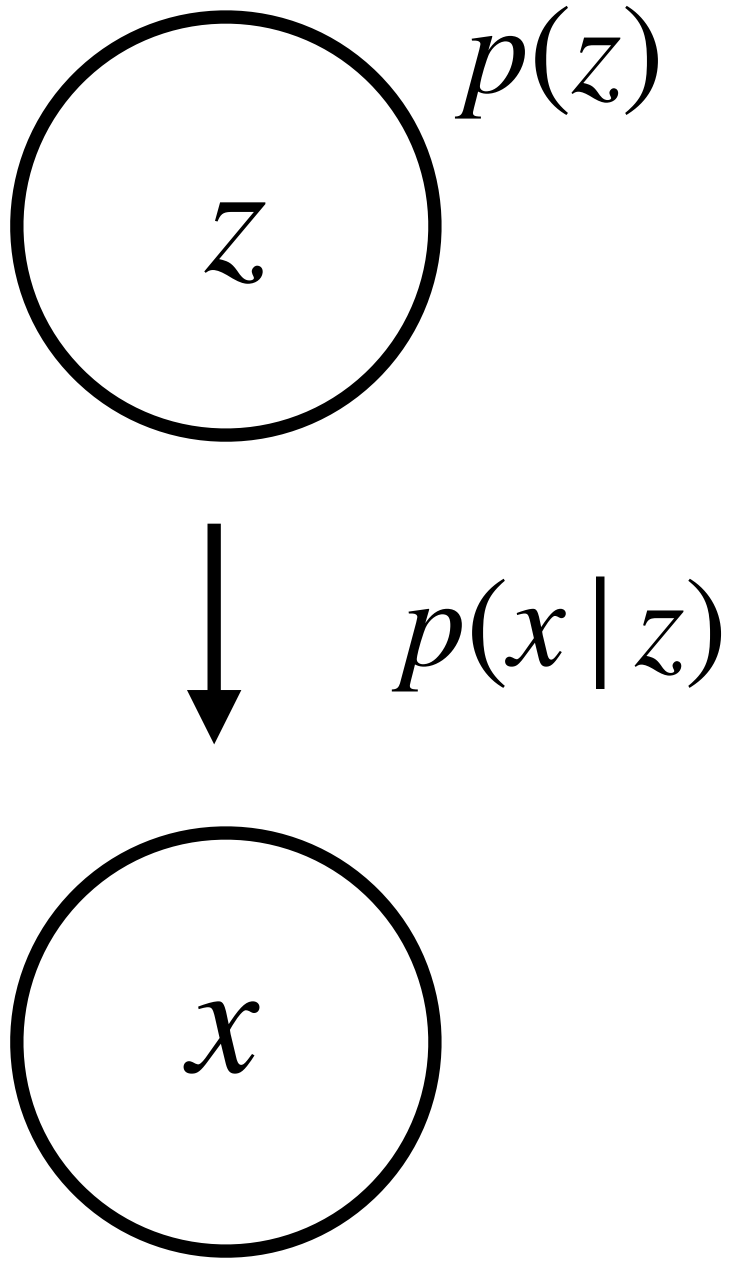
$$p(x) = \sum_{i=1}^{m} p(x|z = z_i)p(z = z_i)$$

  - For continuous latents:

$$p(x) = \int p(x|z)p(z)dz$$

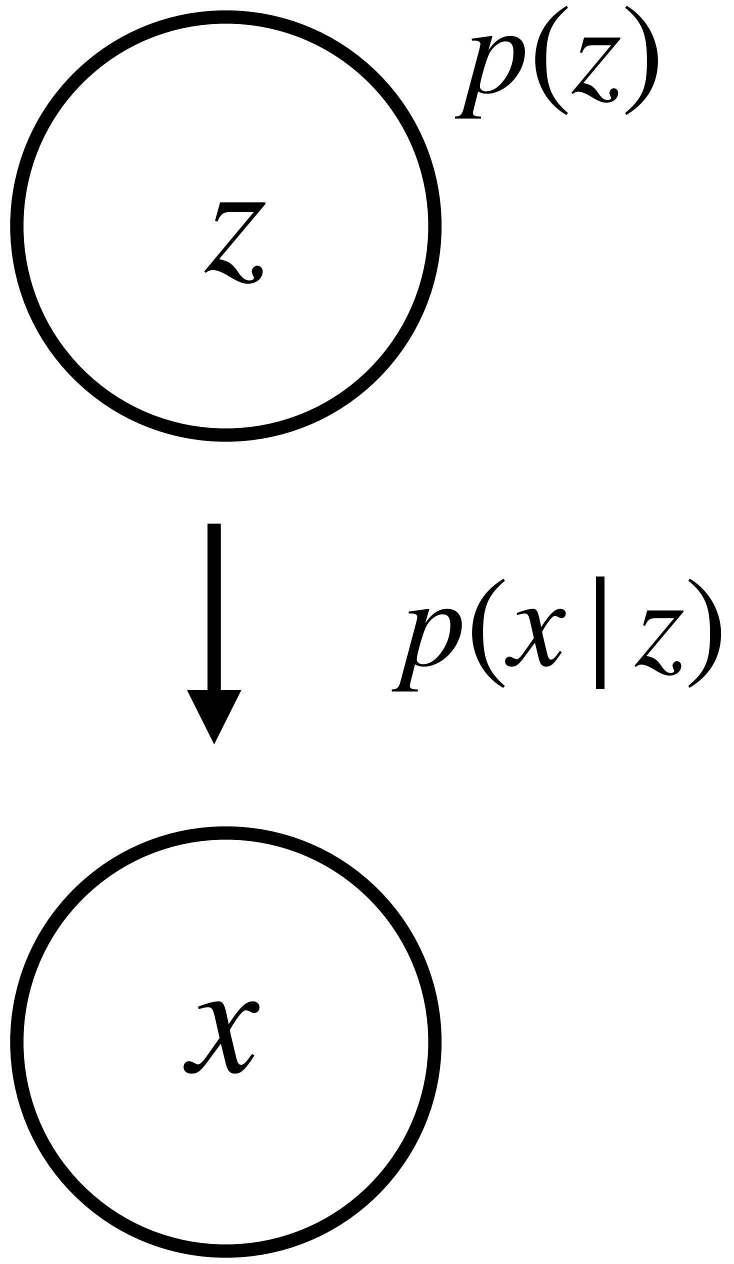
https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

Intro to Latent Variable Models: Goals



https://pillowlab.princeton.edu/teaching/statneuro2020/notes/notes18_LatentVariableModels.pdf

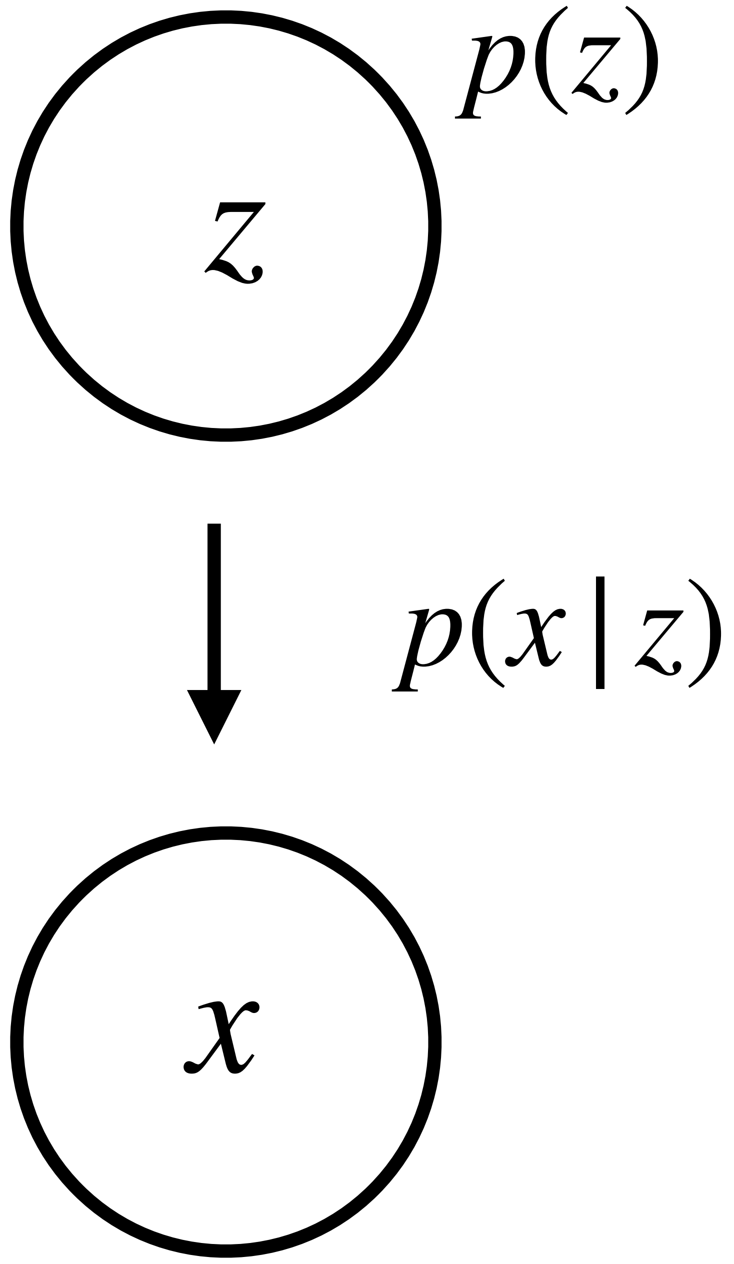
Intro to Latent Variable Models: Goals



- **Recognition/Inference**

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Intro to Latent Variable Models: Goals



- **Recognition/Inference**

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

- **Model Fitting**
  - Model including parameters is actually:

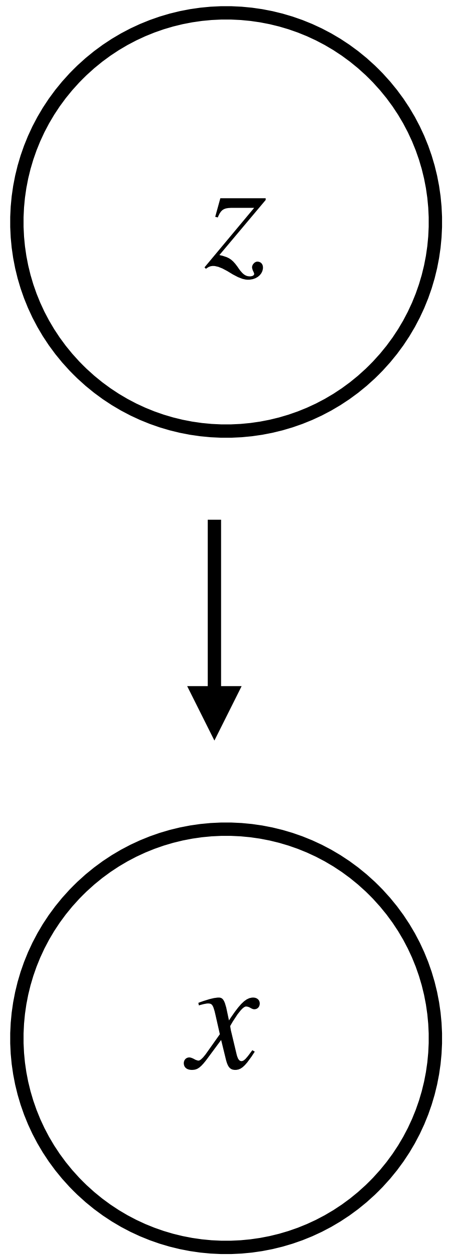
$$p(x, z|\theta) = p(x|z, \theta)p(z|\theta)$$

  - Learning parameters by maximum likelihood:

$$\hat{\theta} = \arg\max_{\theta} p(x|\theta) = \arg\max_{\theta} \int p(x, z|\theta)dz.$$

Factor Analysis

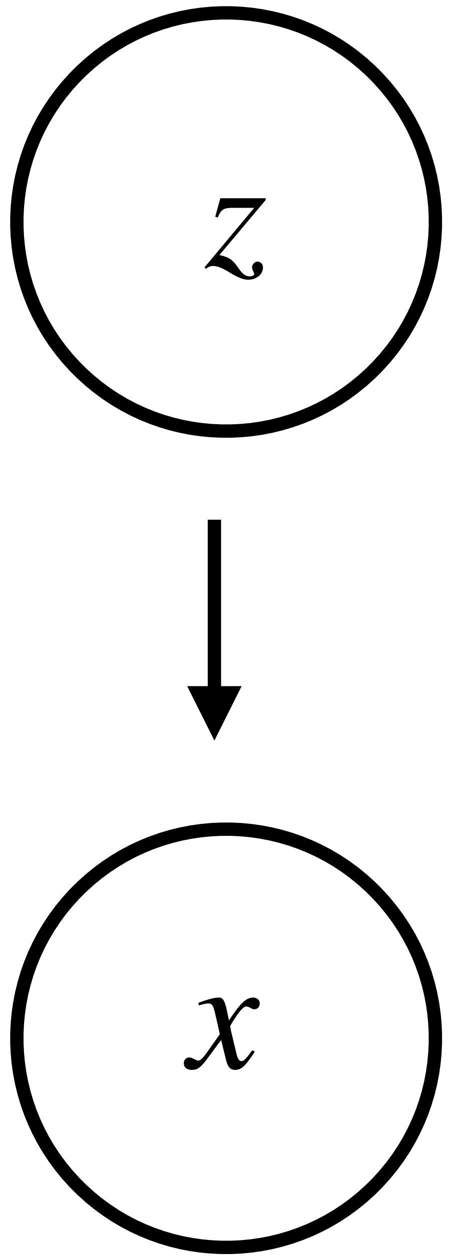
Factor Analysis: Generative Model



$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$$

- Can be any Gaussian (see Murphy, book 1, section 20.2)

Factor Analysis: Generative Model

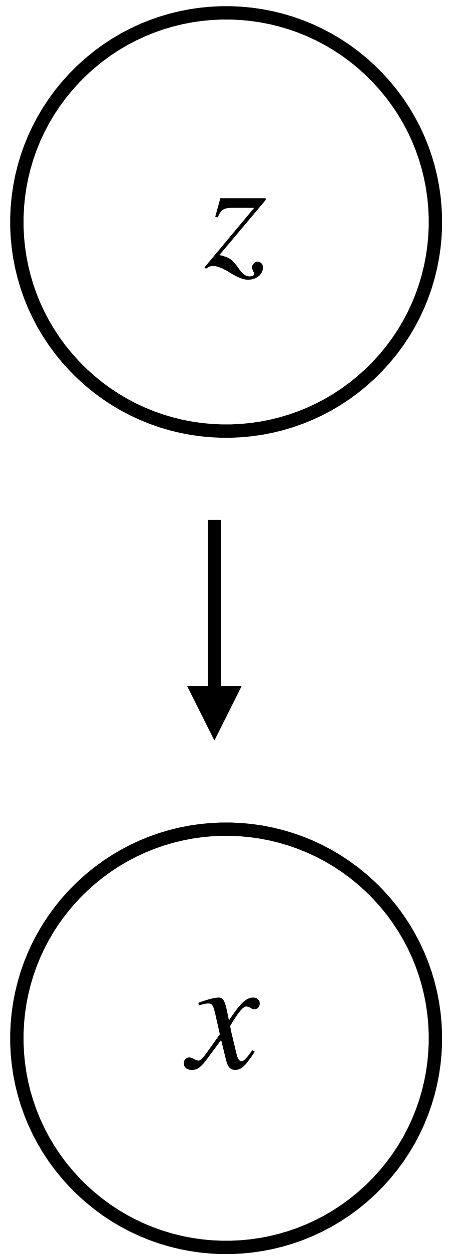


$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$$

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

- Can be any Gaussian (see Murphy, book 1, section 20.2)
- Linear Gaussian model
- z : $D \times T$ latent. dim x samples (timepoints)
- x : $N \times T$ obs. dim (neurons) x samples (timepoints)
- $\mathbf{W}$: $N \times D$ obs. dim. (neurons) x latent dim.
- $\boldsymbol{\Psi}$: $D \times D$ diagonal covariance matrix

Factor Analysis: Generative Model



$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$$

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

$$p(x) = \int p(x|z)p(z)dz$$

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathsf{T}} + \boldsymbol{\Psi})$$

- Can be any Gaussian (see Murphy, book 1, section 20.2)
- Linear Gaussian model
- z : $D \times T$ latent. dim x samples (timepoints)
- x : $N \times T$ obs. dim (neurons) x samples (timepoints)
- $\mathbf{W}$: $N \times D$ obs. dim. (neurons) x latent dim.
- $\boldsymbol{\Psi}$: $D \times D$ diagonal covariance matrix

Factor Analysis: Generative Model

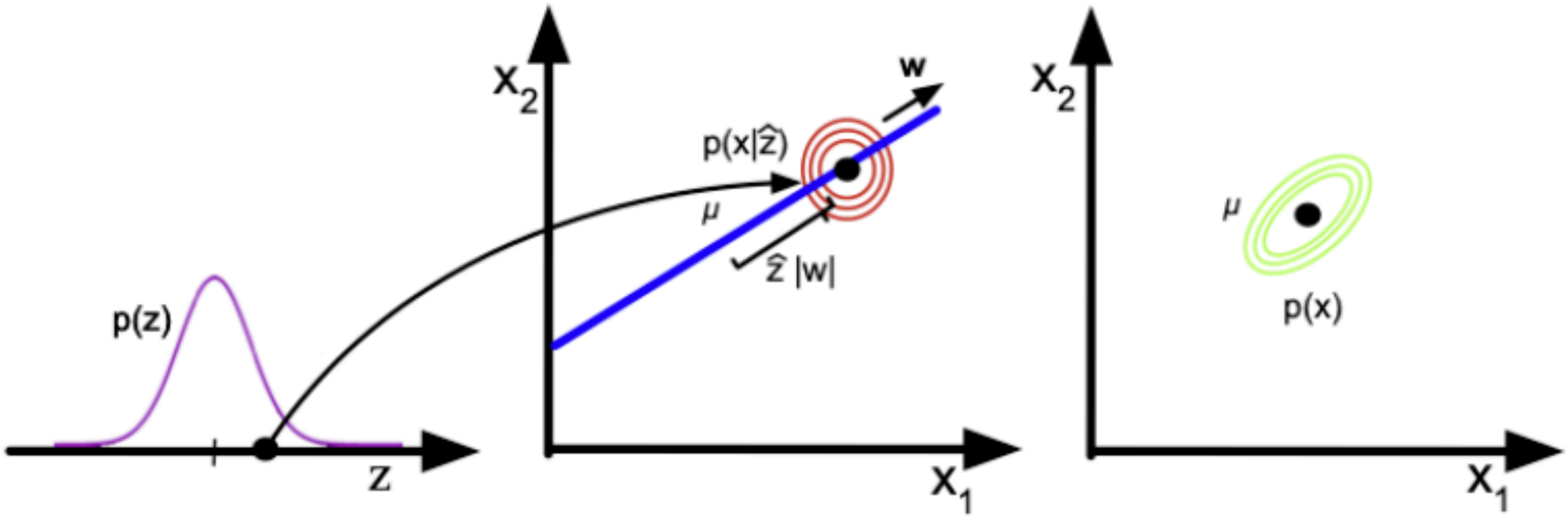
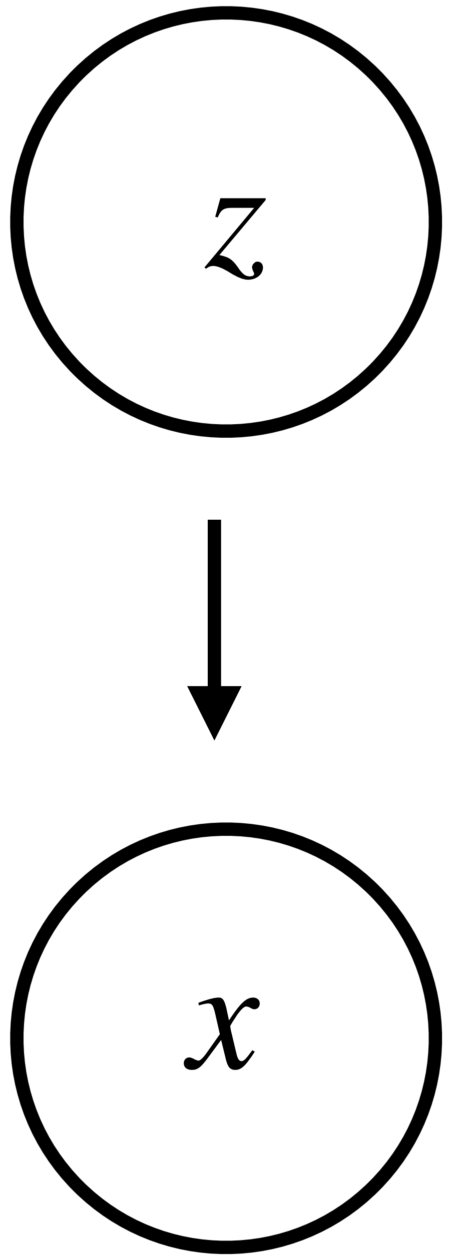


Figure 20.9: Illustration of the FA generative process, where we have $L = 1$ latent dimension generating $D = 2$ observed dimensions; we assume $\boldsymbol{\Psi} = \sigma^2\mathbf{I}$. The latent factor has value $z \in \mathbb{R}$, sampled from $p(z)$; this gets mapped to a 2d offset $\boldsymbol{\delta} = z\boldsymbol{w}$, where $\boldsymbol{w} \in \mathbb{R}^2$, which gets added to $\boldsymbol{\mu}$ to define a Gaussian $p(\boldsymbol{x}|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu} + \boldsymbol{\delta}, \sigma^2\mathbf{I})$. By integrating over z, we "slide" this circular Gaussian "spray can" along the principal component axis $\boldsymbol{w}$, which induces elliptical Gaussian contours in $\boldsymbol{x}$ space centered on $\boldsymbol{\mu}$. Adapted from Figure 12.9 of [Bis06].

Factor Analysis: Generative Model



$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$$

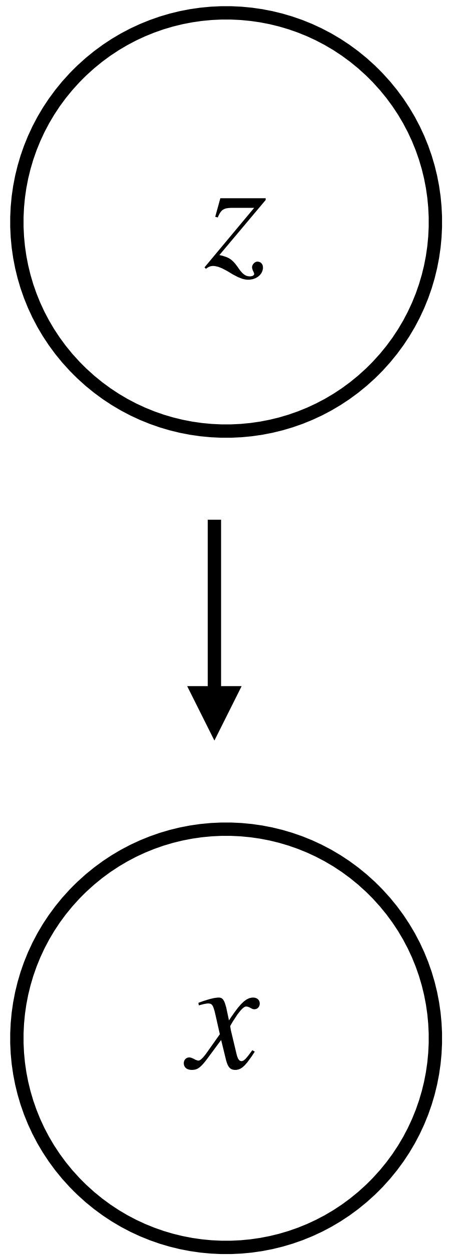
$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\mathbf{W}\boldsymbol{z} + \boldsymbol{\mu}, \boldsymbol{\Psi})$$

$$p(x) = \int p(x|z)p(z)dz$$

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathsf{T}} + \boldsymbol{\Psi})$$

- Can be any Gaussian (see Murphy, book 1, section 20.2)
- Linear Gaussian model
- z : $D \times T$ latent. dim x samples (timepoints)
- x : $N \times T$ obs. dim (neurons) x samples (timepoints
- $\mathbf{W}$: $N \times D$ obs. dim. (neurons) x latent dim.
- $\boldsymbol{\Psi}$: $D \times D$ diagonal covariance matrix

Factor Analysis: Generative Model



$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$$

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

$$p(x) = \int p(x|z)p(z)dz$$

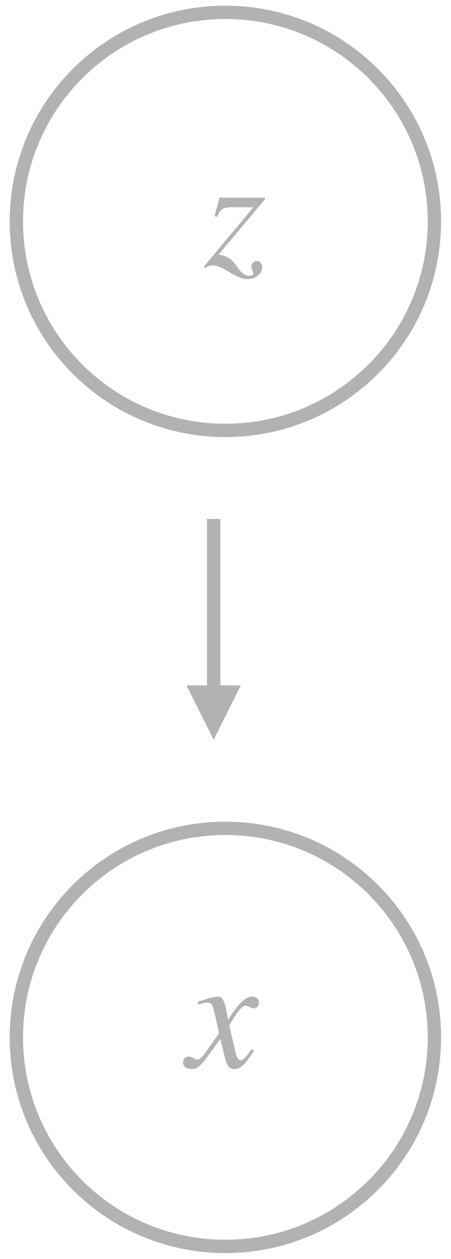
$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathsf{T}} + \boldsymbol{\Psi})$$

Low Rank + Noise

Shared and Unique

- Can be any Gaussian (see Murphy, book 1, section 20.2)
- Linear Gaussian model
- z : $D \times T$ latent. dim x samples (timepoints)
- x : $N \times T$ obs. dim (neurons) x samples (timepoints
- $\mathbf{W}$: $N \times D$ obs. dim. (neurons) x latent dim.
- $\boldsymbol{\Psi}$: $D \times D$ diagonal covariance matrix

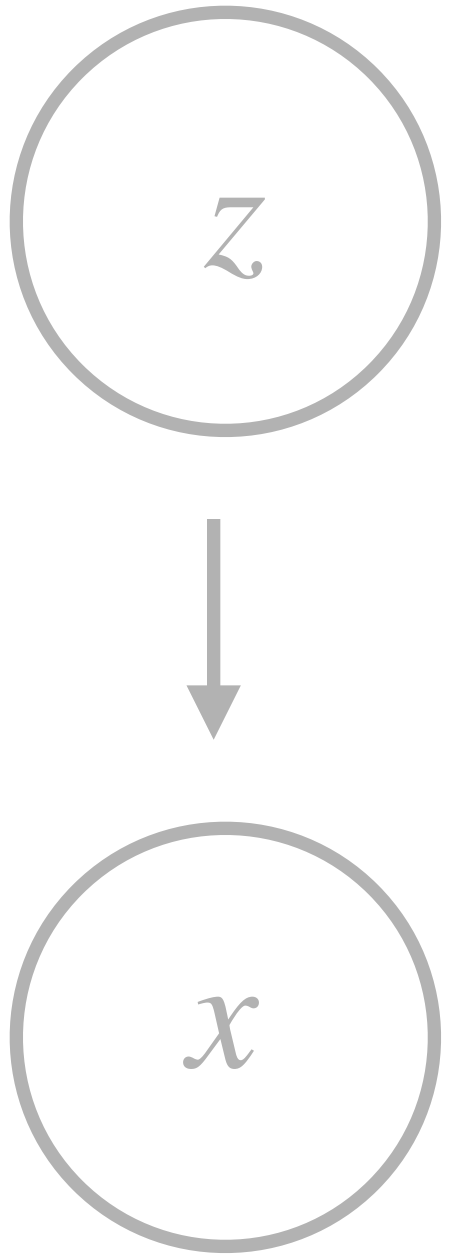
FA vs. Probabilistic PCA vs. PCA



$$p(z) = \mathcal{N}(z|\mathbf{0}, \mathbf{I})$$

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

FA vs. Probabilistic PCA vs. PCA

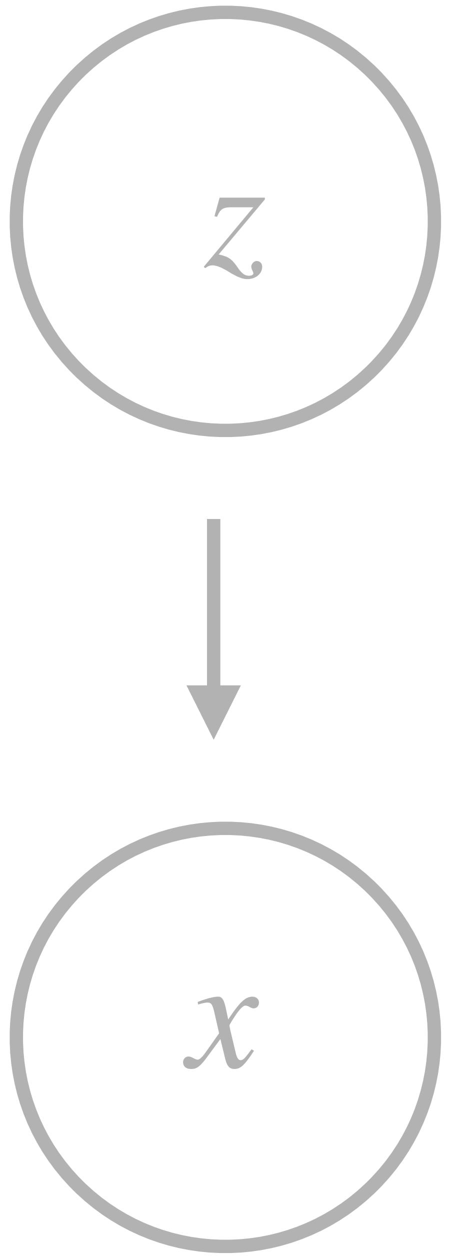


$$p(z) = \mathcal{N}(z|\mathbf{0}, \mathbf{I})$$

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

- Probabilistic PCA is Factor Analysis where $\boldsymbol{\Psi}$ is the identity matrix (all observations have the same independent noise)

FA vs. Probabilistic PCA vs. PCA

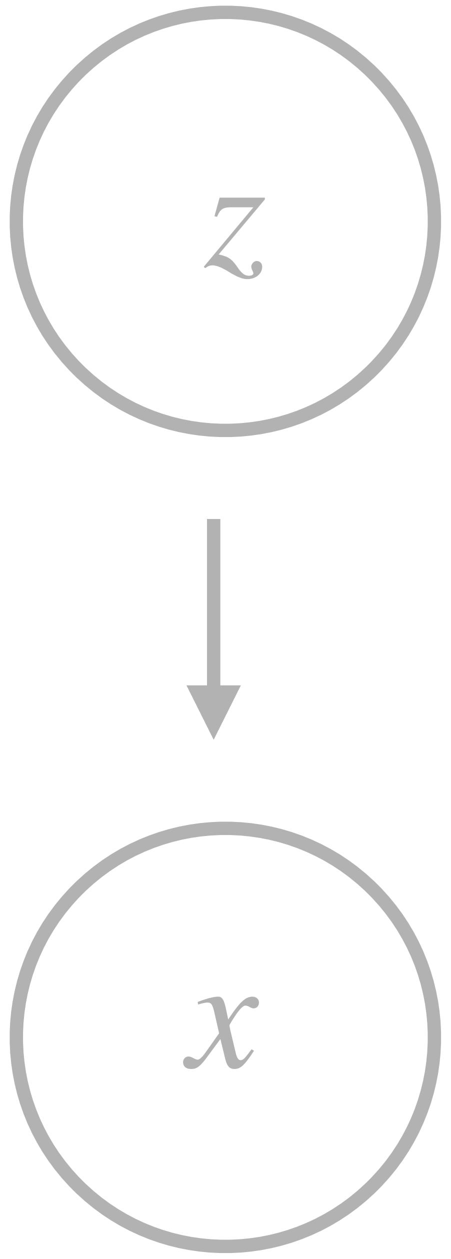


$$p(z) = \mathcal{N}(z|\mathbf{0}, \mathbf{I})$$

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

- Probabilistic PCA is Factor Analysis where $\boldsymbol{\Psi}$ is the identity matrix (all observations have the same independent noise)
- PPCA when $\boldsymbol{\Psi}$ -> 0 becomes PCA

FA vs. PCA: An Example

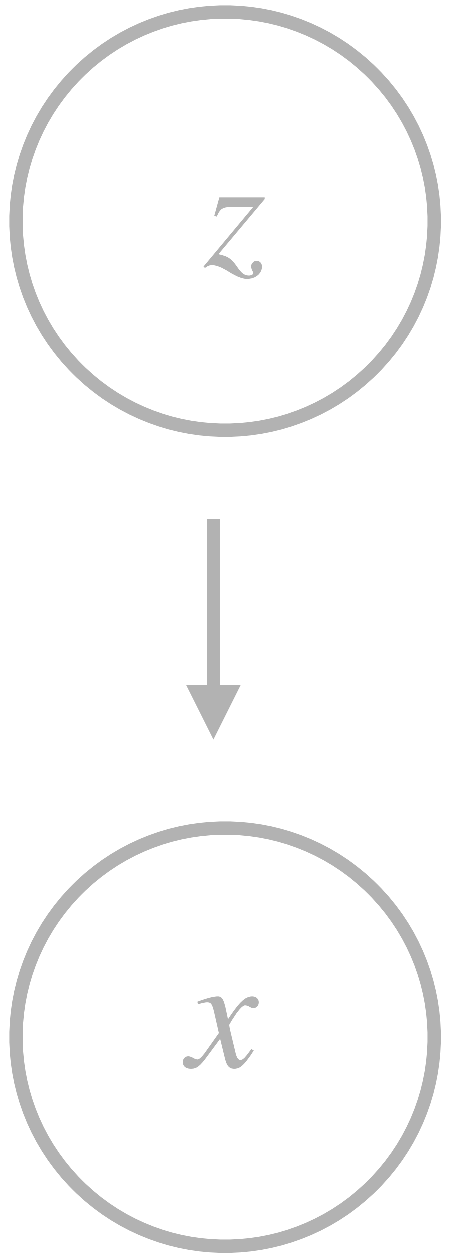


$$p(z) = \mathcal{N}(z|\mathbf{0}, \mathbf{I})$$

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

$$W = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \Psi = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$$

FA vs. PCA: An Example



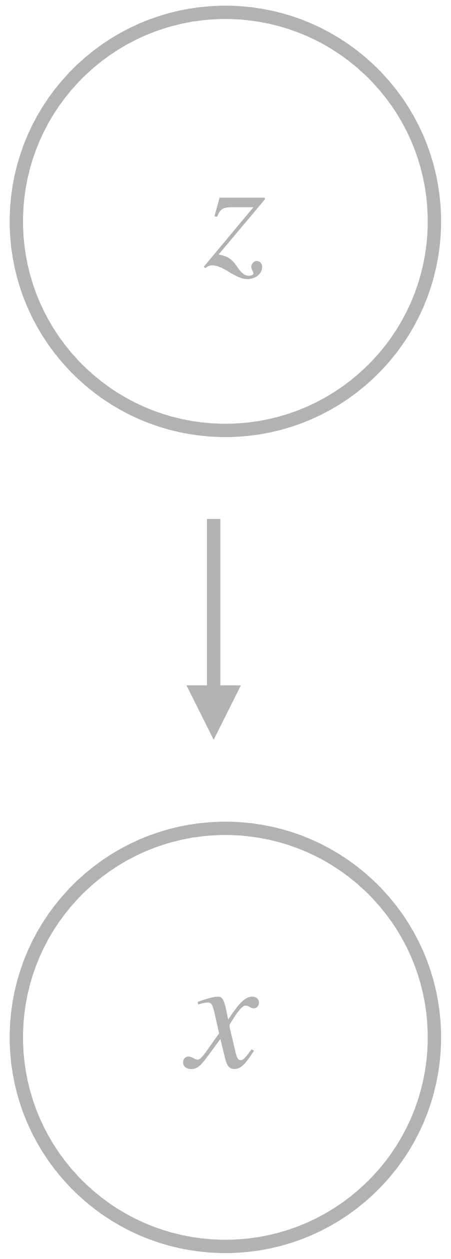
$$p(z) = \mathcal{N}(z|\mathbf{0}, \mathbf{I})$$

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

$$W = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \Psi = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{cov}(X) = WW^T + \Psi = \begin{bmatrix} 101 & 1 \\ 1 & 2 \end{bmatrix}$$

FA vs. PCA: An Example



$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\mathbf{0}, \mathbf{I})$$

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

$$W = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \Psi = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{cov}(X) = WW^T + \Psi = \begin{bmatrix} 101 & 1 \\ 1 & 2 \end{bmatrix}$$

- PCA would give a top eigenvector primarily lying along the first dimension

Factor Analysis: Inferring the latents

$$p(z \mid x) \propto p(x \mid z)p(z)$$

$$= \mathcal{N}(x \mid Wz, \Psi) \cdot \mathcal{N}(z \mid 0, I)$$

Factor Analysis: Inferring the latents

$$p(z \mid x) \propto p(x \mid z)p(z)$$

$$= \mathcal{N}(x \mid Wz, \Psi) \cdot \mathcal{N}(z \mid 0, I)$$

$$\vdots$$

$$= \mathcal{N}(\Lambda W^T \Psi^{-1} x, \Lambda) \quad \text{where} \quad \Lambda = \left(W^T \Psi^{-1} W + I\right)^{-1}$$

Factor Analysis: Inferring the latents

$$p(z \mid x) \propto p(x \mid z)p(z)$$

$$= \mathcal{N}(x \mid Wz, \Psi) \cdot \mathcal{N}(z \mid 0, I)$$

$$\vdots$$

$$= \mathcal{N}(\Lambda W^T \Psi^{-1} x, \Lambda) \quad \text{where} \quad \Lambda = \left(W^T \Psi^{-1} W + I\right)^{-1}$$

- When inferring the latent, the components of x are downweighted in proportion to their amount of independent noise (value in Ψ).

EM for Factor Analysis

- E step: Estimate the posterior, $p(z|x)$, given set parameters
- M step: Estimate the parameters, $[W, \Psi]$, given the expectations of the latents

EM for ~~Factor Analysis~~ PPCA

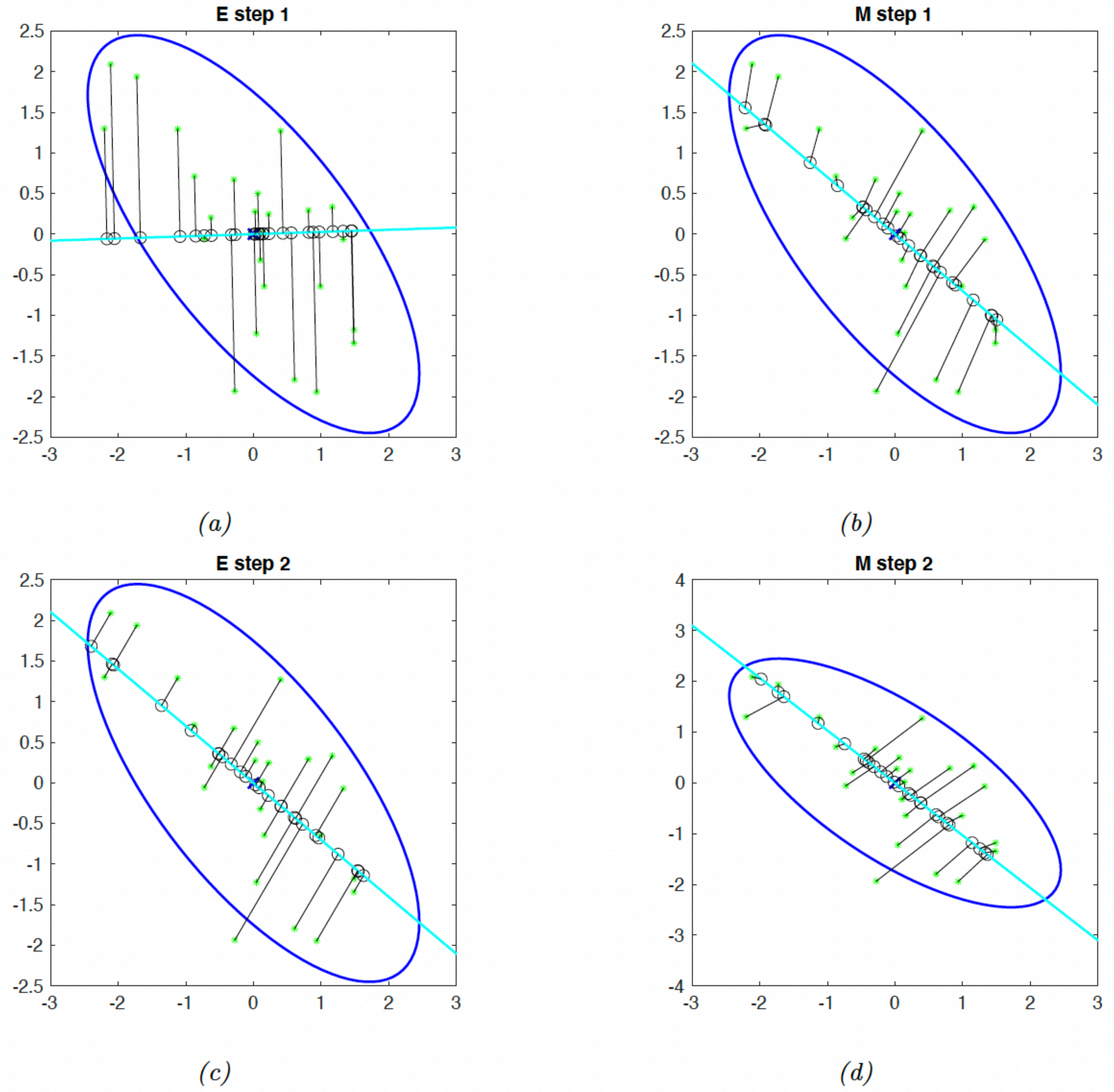


Figure 20.10: Illustration of EM for PCA when $D = 2$ and $L = 1$. Green stars are the original data points, black circles are their reconstructions. The weight vector $\mathbf{w}$ is represented by blue line. (a) We start with a random initial guess of $\mathbf{w}$. The E step is represented by the orthogonal projections. (b) We update the rod $\mathbf{w}$ in the M step, keeping the projections onto the rod (black circles) fixed. (c) Another E step. The black circles can 'slide' along the rod, but the rod stays fixed. (d) Another M step. Adapted from Figure 12.12 of [Bis06].

Why probabilistic models, versus PCA?

- Allows having more sophisticated, and more accurate models
  - Different noise models (FA vs PPCA), mixture of factor analyzers, etc…
- Principled
- Better for missing data, or streaming data

Gaussian Processes

Gaussian Processes

Now consider a function $f : \mathcal{X} \to \mathbb{R}$ evaluated at a set of inputs, $\mathbf{X} = \{\boldsymbol{x}_n \in \mathcal{X}\}_{n=1}^{N}$. Let $\boldsymbol{f}_X = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ be the set of unknown function values at these points.

Gaussian Processes

Now consider a function $f : \mathcal{X} \to \mathbb{R}$ evaluated at a set of inputs, $\mathbf{X} = \{\boldsymbol{x}_n \in \mathcal{X}\}_{n=1}^{N}$. Let $\boldsymbol{f}_X = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ be the set of unknown function values at these points. If $\boldsymbol{f}_X$ is jointly Gaussian for any set of $N \geq 1$ points, then we say that $f : \mathcal{X} \to \mathbb{R}$ is a **Gaussian process**.

Gaussian Processes

Now consider a function $f : \mathcal{X} \to \mathbb{R}$ evaluated at a set of inputs, $\mathbf{X} = \{\boldsymbol{x}_n \in \mathcal{X}\}_{n=1}^{N}$. Let $\boldsymbol{f}_X = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ be the set of unknown function values at these points. If $\boldsymbol{f}_X$ is jointly Gaussian for any set of $N \geq 1$ points, then we say that $f : \mathcal{X} \to \mathbb{R}$ is a **Gaussian process**. Such a process is defined by its **mean function** $m(\boldsymbol{x}) \in \mathbb{R}$ and a **covariance function**, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') \geq 0$, which is any positive definite **Mercer kernel**

Gaussian Processes

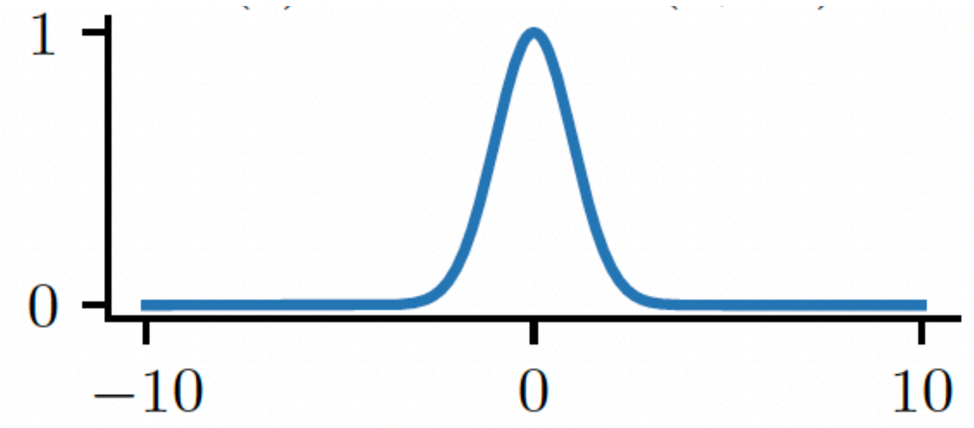
Now consider a function $f : \mathcal{X} \to \mathbb{R}$ evaluated at a set of inputs, $\mathbf{X} = \{\boldsymbol{x}_n \in \mathcal{X}\}_{n=1}^N$. Let $\boldsymbol{f}_X = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ be the set of unknown function values at these points. If $\boldsymbol{f}_X$ is jointly Gaussian for any set of $N \geq 1$ points, then we say that $f : \mathcal{X} \to \mathbb{R}$ is a **Gaussian process**. Such a process is defined by its **mean function** $m(\boldsymbol{x}) \in \mathbb{R}$ and a **covariance function**, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') \geq 0$, which is any positive definite **Mercer kernel**

- Example Kernel ("Radial Basis Function"): $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}'; \ell) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right)$

Gaussian Processes

Now consider a function $f : \mathcal{X} \to \mathbb{R}$ evaluated at a set of inputs, $\mathbf{X} = \{\boldsymbol{x}_n \in \mathcal{X}\}_{n=1}^{N}$. Let $\boldsymbol{f}_X = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ be the set of unknown function values at these points. If $\boldsymbol{f}_X$ is jointly Gaussian for any set of $N \geq 1$ points, then we say that $f : \mathcal{X} \to \mathbb{R}$ is a **Gaussian process**. Such a process is defined by its **mean function** $m(\boldsymbol{x}) \in \mathbb{R}$ and a **covariance function**, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') \geq 0$, which is any positive definite **Mercer kernel**

- Example Kernel (“Radial Basis Function”): $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}'; \ell) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right)$

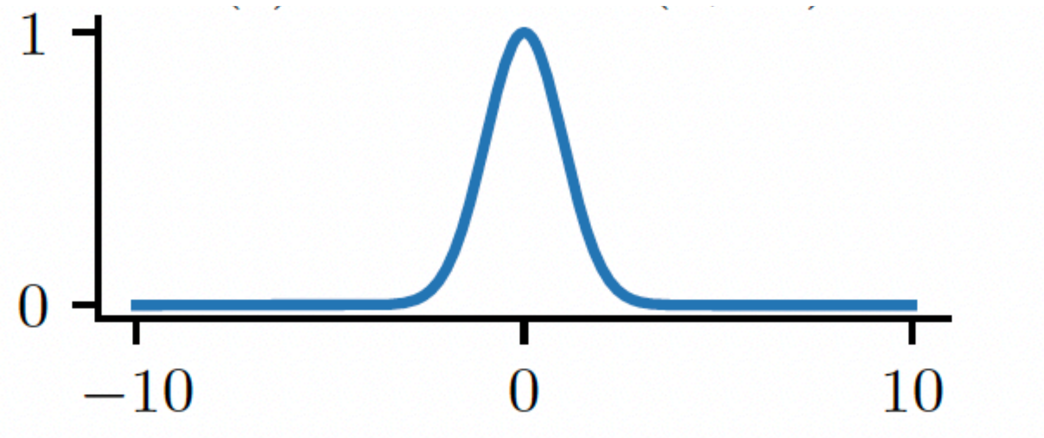


(f) RBF k(x,0.0)

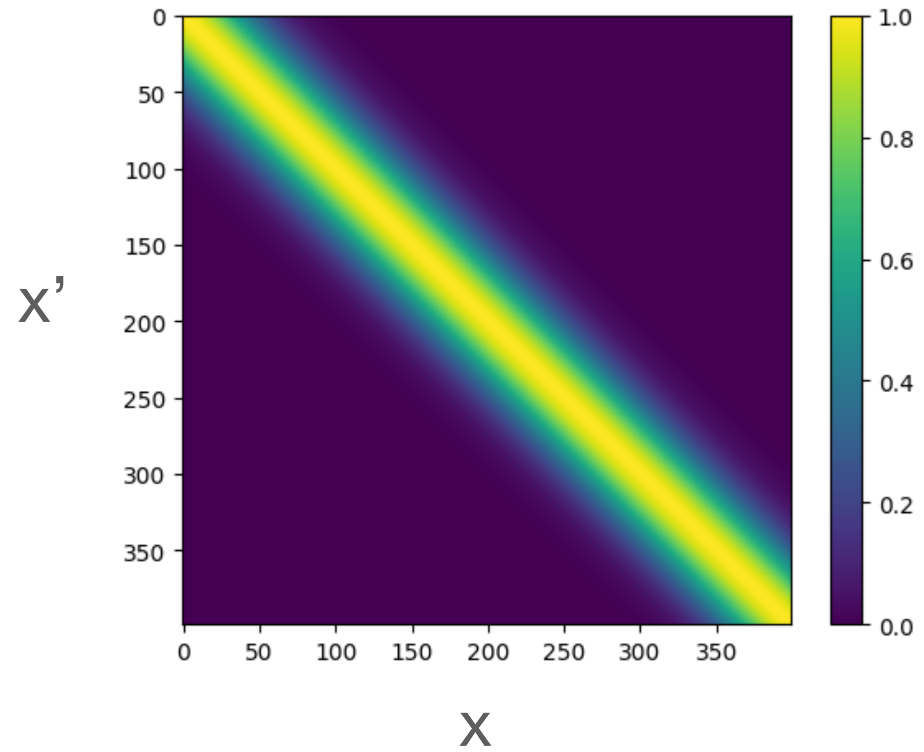
Gaussian Processes

Now consider a function $f : \mathcal{X} \to \mathbb{R}$ evaluated at a set of inputs, $\mathbf{X} = \{\boldsymbol{x}_n \in \mathcal{X}\}_{n=1}^{N}$. Let $\boldsymbol{f}_X = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]$ be the set of unknown function values at these points. If $\boldsymbol{f}_X$ is jointly Gaussian for any set of $N \geq 1$ points, then we say that $f : \mathcal{X} \to \mathbb{R}$ is a **Gaussian process**. Such a process is defined by its **mean function** $m(\boldsymbol{x}) \in \mathbb{R}$ and a **covariance function**, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') \geq 0$, which is any positive definite **Mercer kernel**

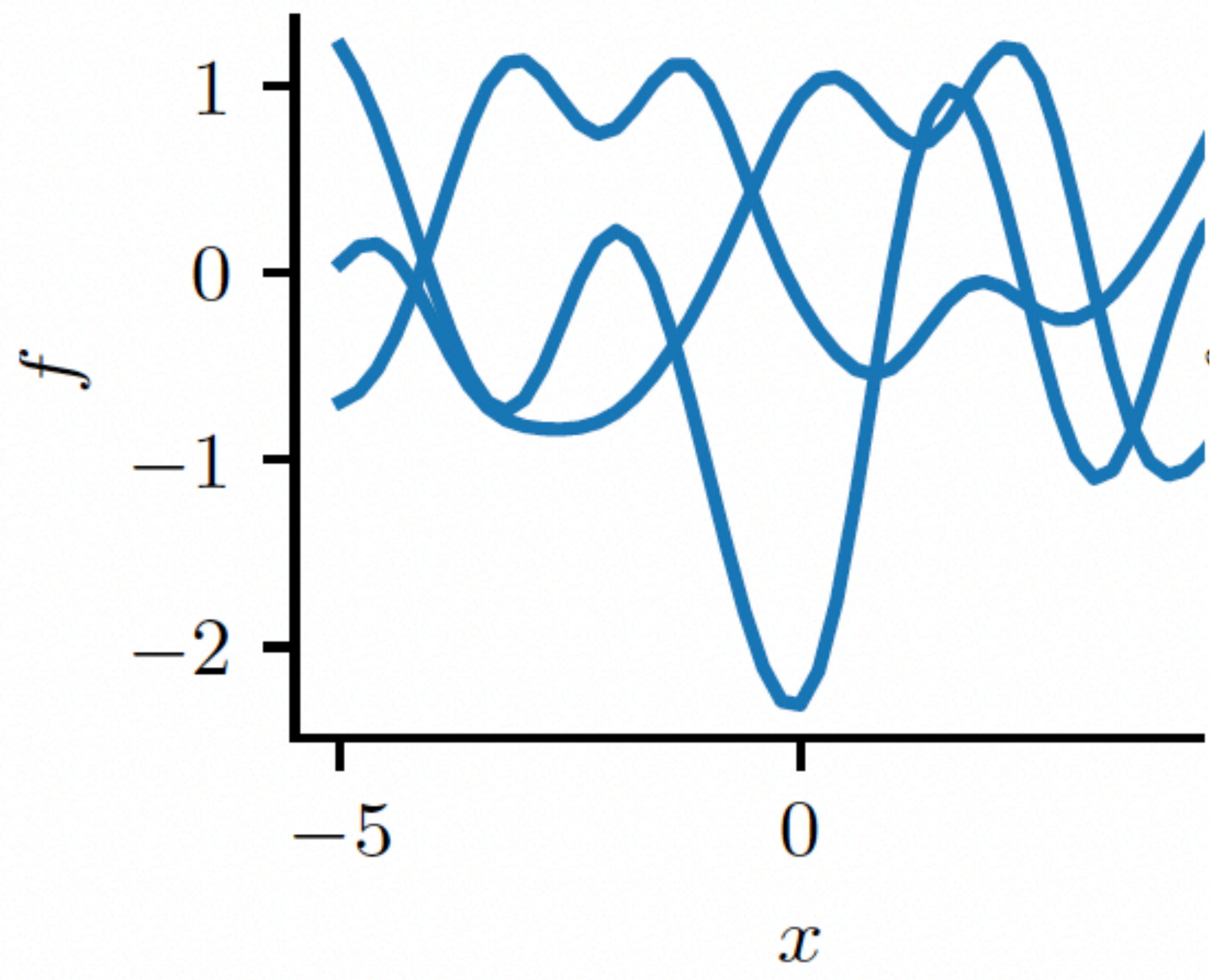
- Example Kernel ("Radial Basis Function"): $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}'; \ell) = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{2\ell^2}\right)$



(f) RBF k(x,0.0)



Gaussian Processes - sampling from the prior



(a)

Figure 18.7: Left: some functions sampled from a GP prior with RBF kernel. Middle: some samples from a GP posterior, after conditioning on 5 noise-free observations. Right: some samples from a GP posterior, after conditioning on 5 noisy observations. The shaded area represents $\mathbb{E}[f(\boldsymbol{x})] \pm 2\sqrt{\mathbb{V}[f(\boldsymbol{x})]}$. *Adapted from Figure 2.2 of [RW06]. Generated by gpr_demo_noise_free.ipynb.*

Gaussian Processes - Example kernels

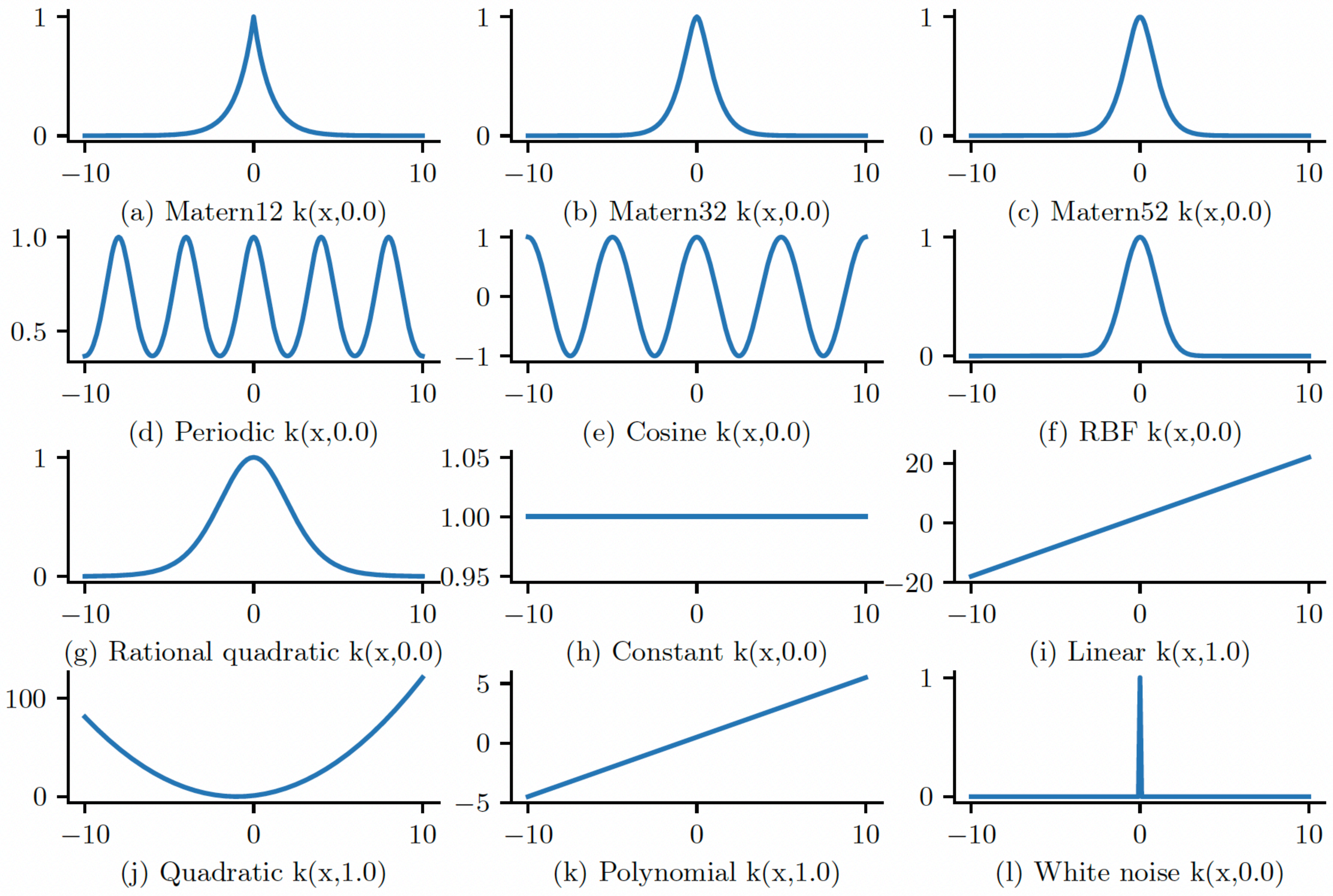
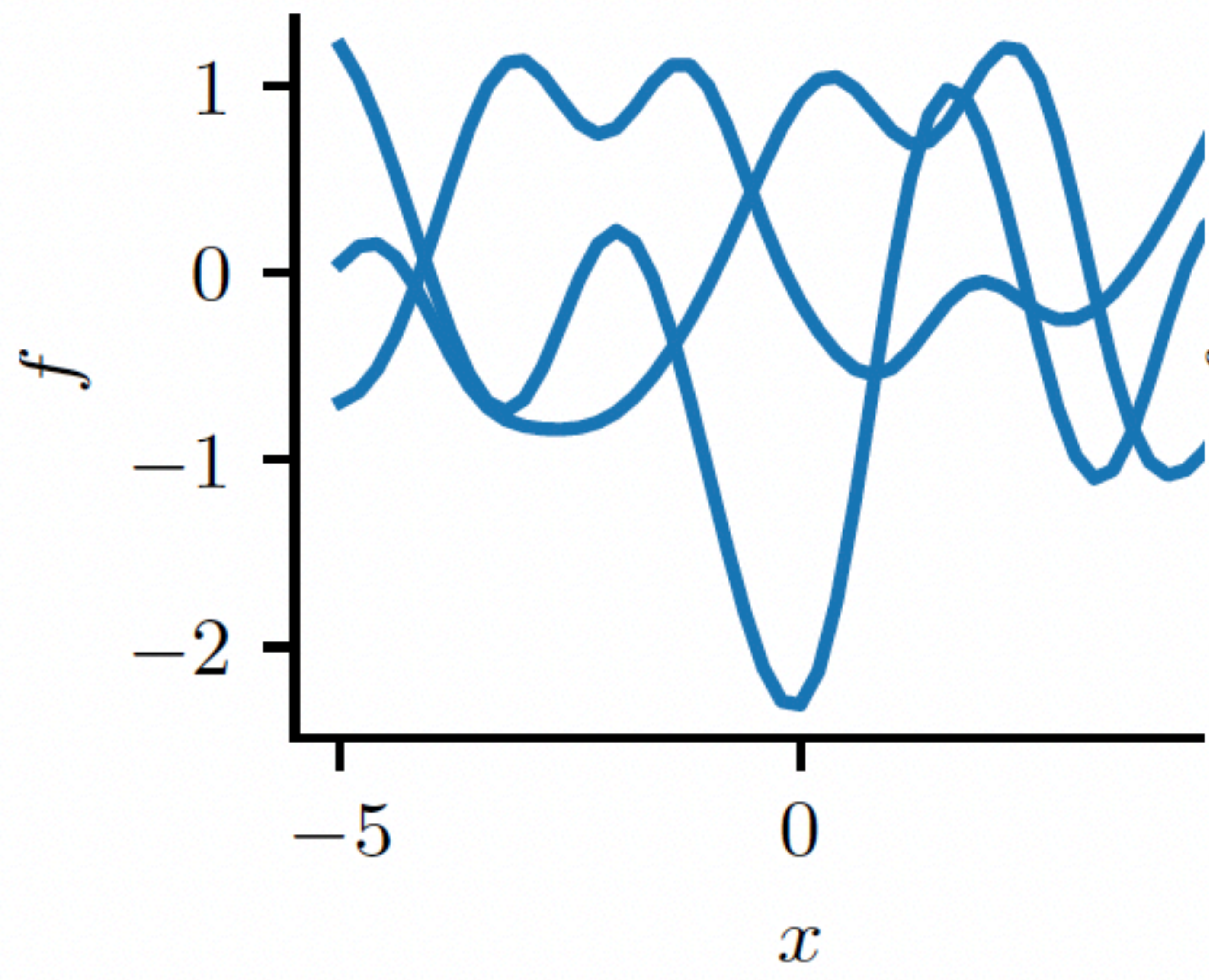


Figure 18.3: GP kernels evaluated at $k(x, 0)$ as a function of x. Generated by gpKernelPlot.ipynb.

Gaussian Processes - estimating a posterior



(a)

Figure 18.7: Left: some functions sampled from a GP prior with RBF kernel. Middle: some samples from a GP posterior, after conditioning on 5 noise-free observations. Right: some samples from a GP posterior, after conditioning on 5 noisy observations. The shaded area represents $\mathbb{E}[f(\boldsymbol{x})] \pm 2\sqrt{\mathbb{V}[f(\boldsymbol{x})]}$. *Adapted from Figure 2.2 of [RW06]. Generated by gpr_demo_noise_free.ipynb.*

Gaussian Processes - estimating a posterior

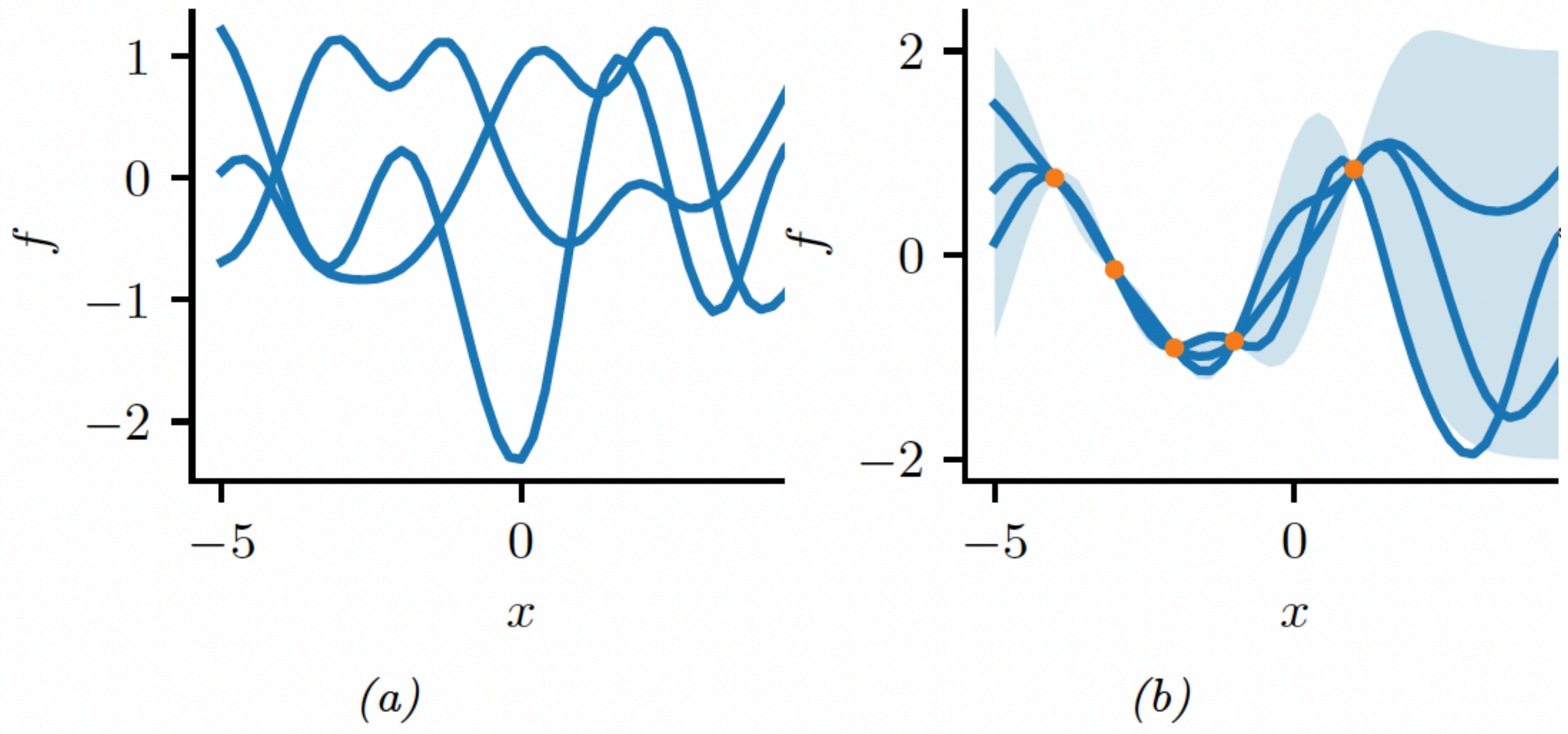


Figure 18.7: Left: some functions sampled from a GP prior with RBF kernel. Middle: some samples from a GP posterior, after conditioning on 5 noise-free observations. Right: some samples from a GP posterior, after conditioning on 5 noisy observations. The shaded area represents $\mathbb{E}[f(\boldsymbol{x})] \pm 2\sqrt{\mathbb{V}[f(\boldsymbol{x})]}$. *Adapted from Figure 2.2 of [RW06]. Generated by gpr_demo_noise_free.ipynb.*

Gaussian Processes - estimating a posterior

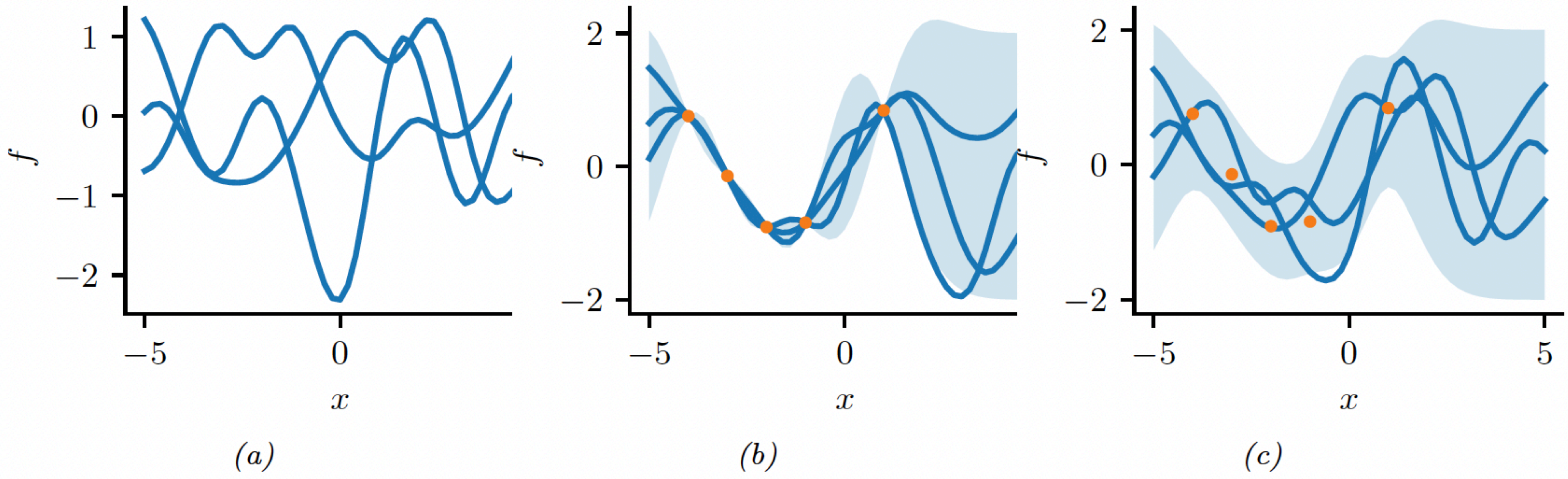


Figure 18.7: Left: some functions sampled from a GP prior with RBF kernel. Middle: some samples from a GP posterior, after conditioning on 5 noise-free observations. Right: some samples from a GP posterior, after conditioning on 5 noisy observations. The shaded area represents $\mathbb{E}[f(\boldsymbol{x})] \pm 2\sqrt{\mathbb{V}[f(\boldsymbol{x})]}$. *Adapted from Figure 2.2 of [RW06]. Generated by gpr_demo_noise_free.ipynb.*