CS 159: Lecture 2

Maximum likelihood inference
& latent variable models

Presentation guidelines

- Detailed guidelines on course website.
- In short: groups will present papers, being sure to:
  - Explain the approach, giving a concrete toy example
  - Place the work in context—how does it relate to other papers?
  - Describe and critically evaluate the claims of the papers
  - (Optional) walk the class through a simple implementation of the method.

Structure of the lecture

- Reminder of the maximum likelihood framework
- The simplest example: how PCA fits into this framework
- Adding more structure: latent variable models
- Gaussian mixture models and the EM algorithm
- Joe: the big picture, and more modeling assumptions

Probabilistic modeling

- Given a dataset $X = \{x_1, x_2, \cdots, x_n\}$
- Wish to fit a probabilistic model $p_\theta(x)$
- E.g. for a Gaussian model, $\theta = \{\mu, \sigma^2\}$

Maximum likelihood

- A standard way to do this is to fit the parameters θ via maximizing the log likelihood of the observed data under the model:

$$\theta^* = \max_{\theta} \sum_{i=1}^{n} \log p_{\theta}(x_i)$$

- Contrast this to a full Bayesian approach which would attempt to assign a probability to every possible θ

Reminder

- As discussed last lecture, maximizing likelihood is related to minimizing the KL divergence between the model and the data distribution, since

$$
\begin{aligned}
\min_{\theta} \mathrm{KL}(p\,||\,p_\theta) &= \min_{\theta} \sum_{x} (p \log p - p \log p_\theta) \\
&\approx \max_{\theta} \sum_{i=1}^{n} \log p_\theta(x_i) \\
&= \max_{\theta} \log \prod_{i=1}^{n} p_\theta(x_i)
\end{aligned}
$$

- Where the product reflects that the generating process is iid

- Note that we can define a probabilistic model that just samples uniformly from the dataset

- The “parameters” θ of this “model” are just memorizing the data $\theta = X$

- This will lead trivially to the max possible log likelihood

- But the “model” will be useless

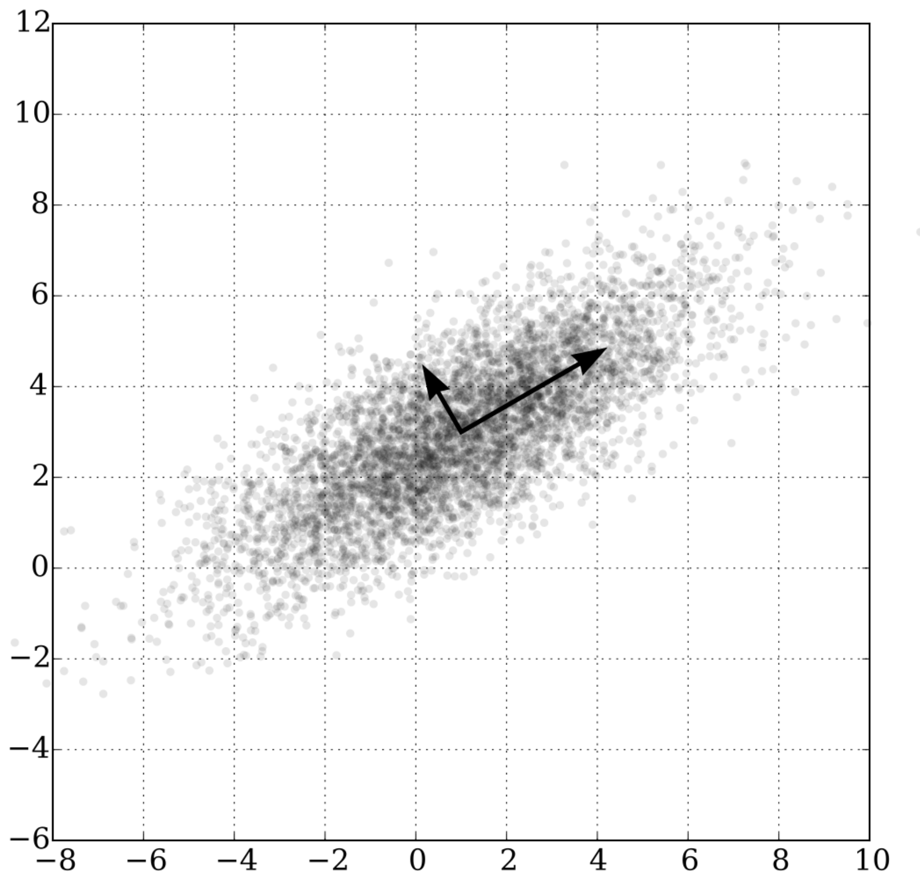
- The previous example demonstrates that the essence of probabilistic modeling via maximum likelihood is to choose a model class of the right complexity

- It should model meaningful structure and not just memorize spurious structure

- It should be tractable to optimize

- Neural nets fit this description; in this lecture we’ll go into greater depth on some more classical methods

PCA



- Principal component analysis
- Dimensionality reduction technique
- Operates on a dataset $X = \{x_1, x_2, \cdots, x_n\}$

The PCA algorithm

- Stack data into a matrix
- Center the data (subtract mean)

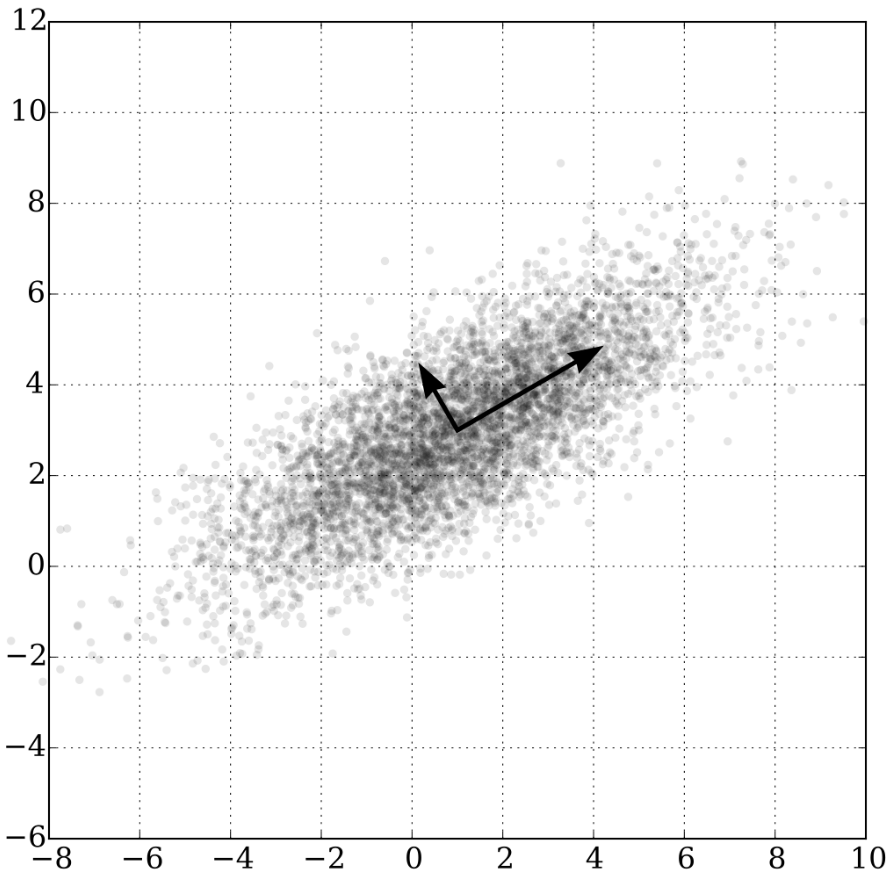
$$\tilde{X} = \begin{bmatrix} x_1 - \bar{x} \rightarrow \\ x_2 - \bar{x} \rightarrow \\ \vdots \\ x_n - \bar{x} \rightarrow \end{bmatrix}$$

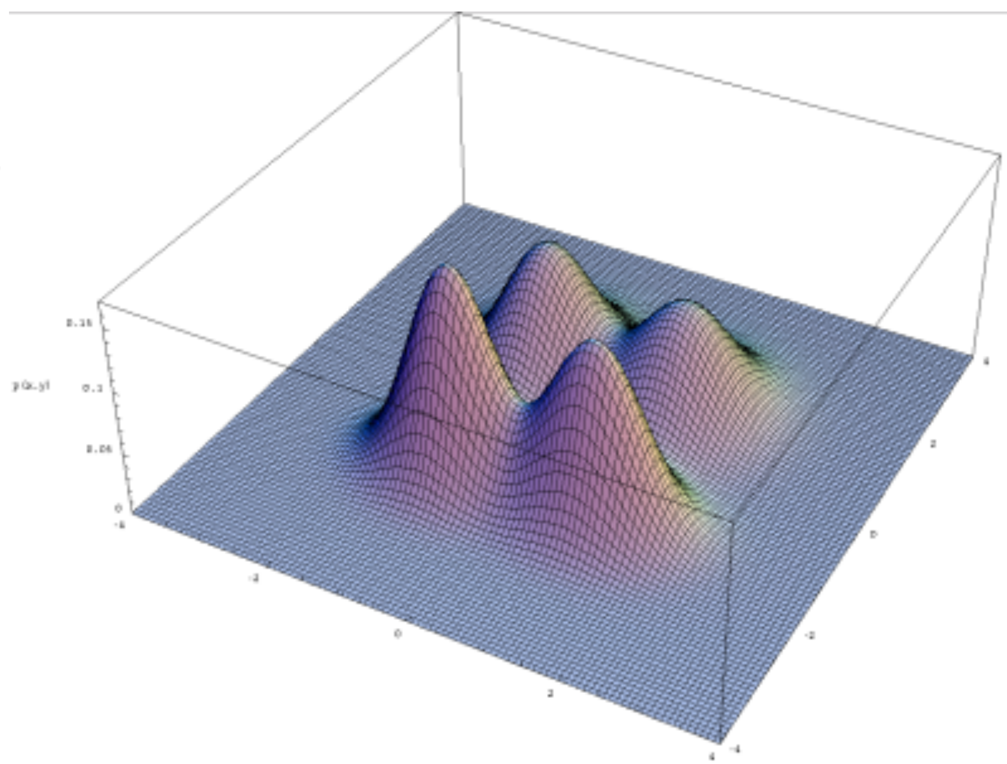
- Diagonalise $\tilde{X}\tilde{X}^T$ into orthonormal system VDV^T
- Project the data on to the top k eigenvectors of V

Interpreting PCA

- PCA can be seen through the lens of maximum likelihood estimation
- We model the data matrix X as n samples from a multivariate Gaussian
- Check: the maximum likelihood estimator of the covariance is $\tilde{X}\tilde{X}^T$
- After “learning” this model, we can use it to perform dimensionality reduction.

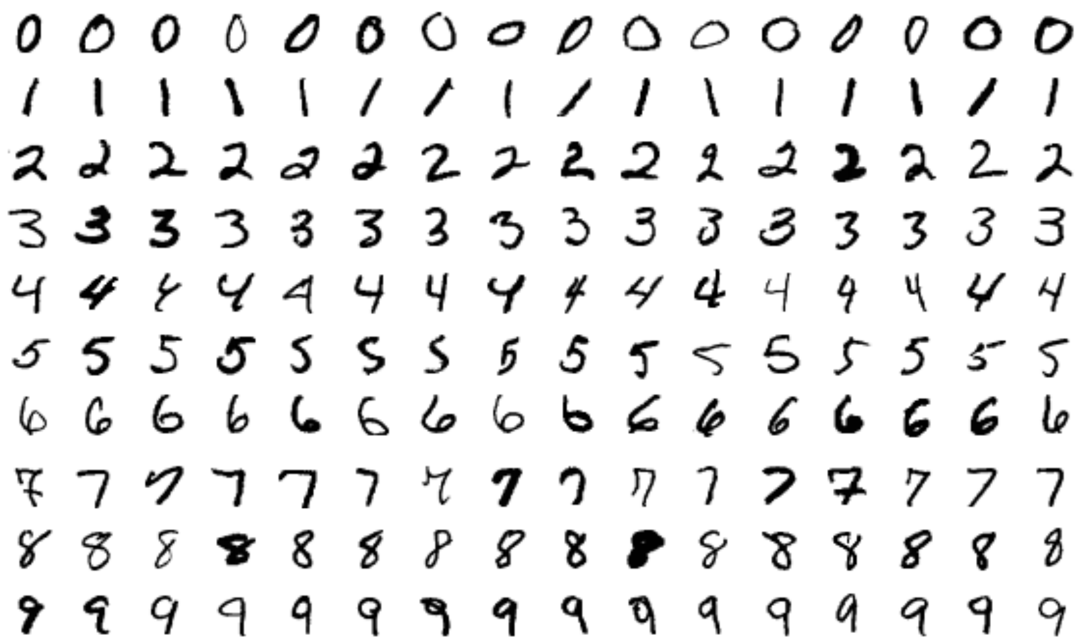
When will PCA fail?





Interlude: latent variables

- One way to go beyond Gaussian models and PCA is the idea of latent variables

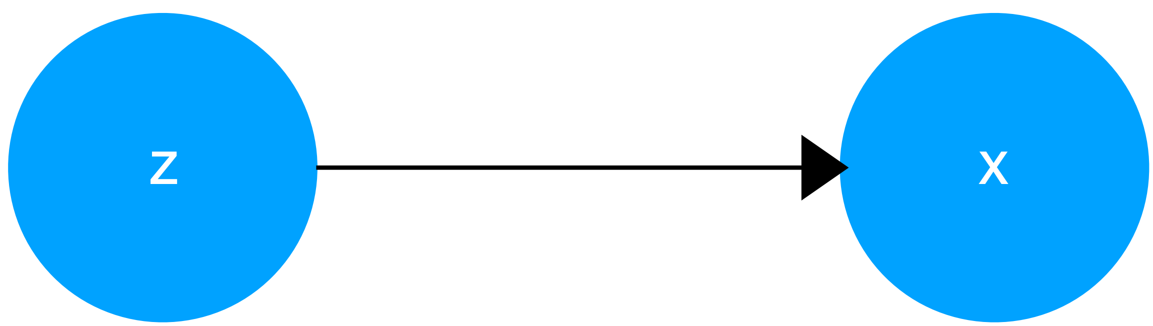


- Some unobserved variable that adds structure to the data generating process

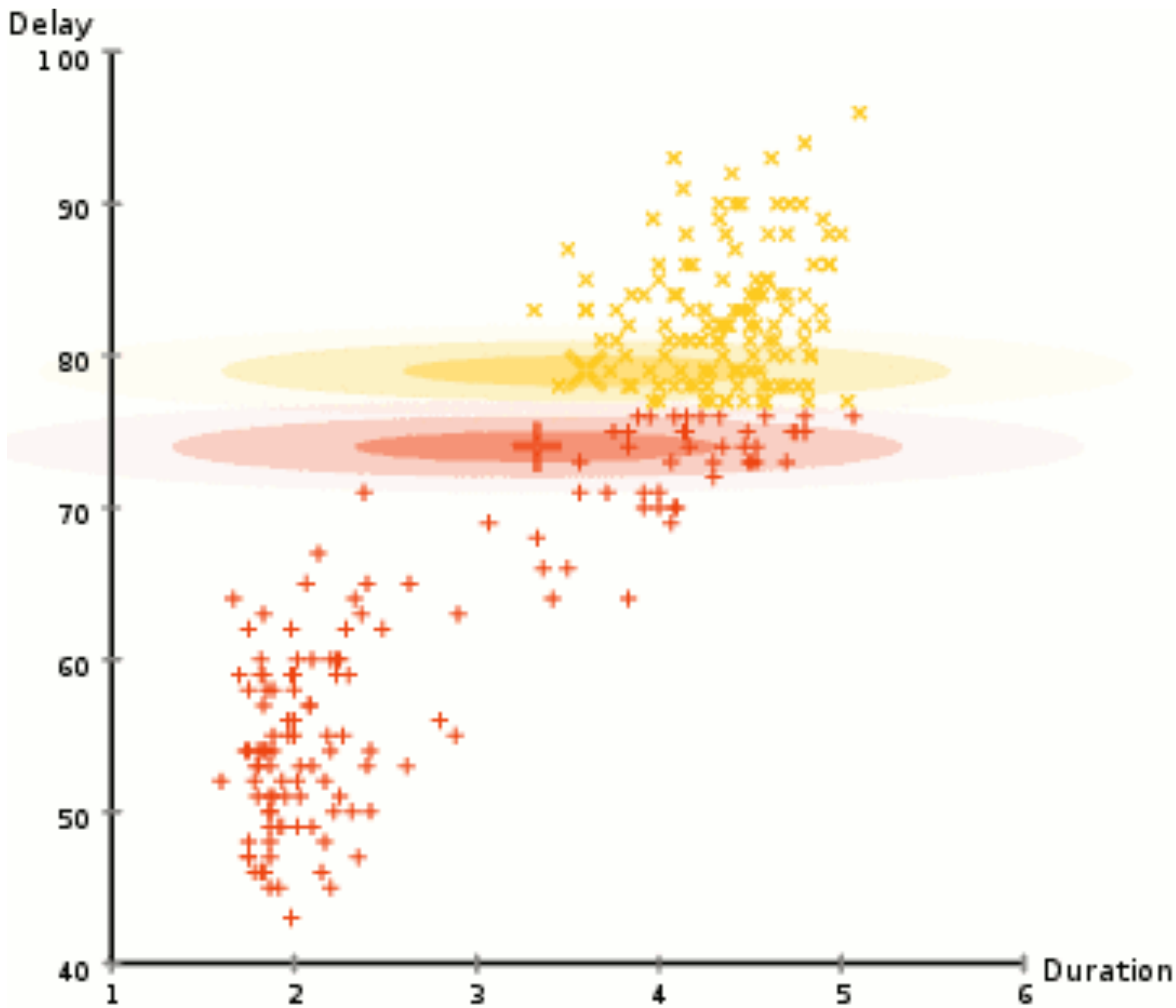
Taking advantage of latent variables

- Model the data generating process as

$$p(x \mid \theta) = \sum_{z} p(x \mid \theta, z) p(z)$$



Mixture of Gaussians



linear factor models

linear factor models

GENERAL FORM:

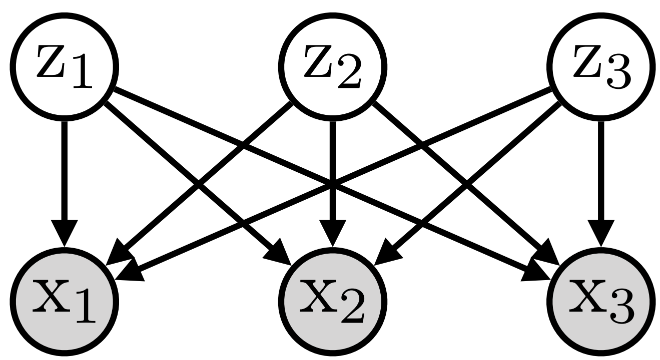
probabilistic model: $p_\theta(\mathbf{x}, \mathbf{z})$

latent variables: $\mathbf{z} \sim p(\mathbf{z})$ *where* $p(\mathbf{z}) = \prod_i p(\mathrm{z}_i)$

marginalize: $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \left[p_\theta(\mathbf{x}|\mathbf{z}) \right]$

linear transformation: $\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \mathrm{noise}$

where noise *is typically Gaussian and diagonal.*

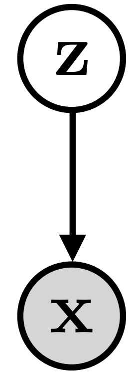


Goodfellow *et al.*, 2016, Chapter 13

LINEAR GAUSSIAN SYSTEM:

prior: $$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z})$$

conditional likelihood: $$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mathbf{b}, \boldsymbol{\Sigma}_\mathbf{x})$$



Bayes' Rule:

posterior: $$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{x}})$$

$$\boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{x}}^{-1} = \boldsymbol{\Sigma}_\mathbf{z}^{-1} + \mathbf{W}^\intercal \boldsymbol{\Sigma}_\mathbf{x}^{-1} \mathbf{W}$$

$$\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}} = \boldsymbol{\Sigma}_{\mathbf{z}|\mathbf{x}} \left[\mathbf{W}^\intercal \boldsymbol{\Sigma}_\mathbf{x}^{-1} (\mathbf{x} - \mathbf{b}) + \boldsymbol{\Sigma}_\mathbf{z}^{-1} \boldsymbol{\mu}_\mathbf{z} \right]$$

Marginalization:

marginal likelihood: $$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\boldsymbol{\mu}_\mathbf{z} + \mathbf{b}, \boldsymbol{\Sigma}_\mathbf{x} + \mathbf{W}\boldsymbol{\Sigma}_\mathbf{z}\mathbf{W}^\intercal)$$

Murphy, 2012, Chapter 4

linear factor models

general form: $\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \text{noise}$

FACTOR ANALYSIS:

standard Gaussian prior: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

x_i are assumed conditionally independent given $\mathbf{z}$

$$\text{noise} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\psi})$$

where $\boldsymbol{\psi} = \text{diag}(\boldsymbol{\sigma}^2)$ *with* $\boldsymbol{\sigma}^2 = \left[\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2\right]^\intercal$

in this case,

$$p_\theta(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{W}\mathbf{W}^\intercal + \boldsymbol{\psi})$$

Goodfellow *et al.*, 2016, Chapter 13

linear factor models

general form: $\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \text{noise}$

PROBABILISTIC PCA:

standard Gaussian prior: $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

x_i are assumed conditionally independent given $\mathbf{z}$

$$\text{noise} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\psi})$$

where $\boldsymbol{\psi} = \text{diag}(\boldsymbol{\sigma}^2)$ *with* $\boldsymbol{\sigma}^2 = \left[\sigma^2, \sigma^2, \ldots, \sigma^2\right]^\intercal$

in this case,

$$p_\theta(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{b}, \mathbf{W}\mathbf{W}^\intercal + \sigma^2\mathbf{I})$$

as $\sigma \to 0$ *we recover PCA*

Goodfellow *et al.,* 2016, Chapter 13

Tipping & Bishop, 1999

linear factor models

general form: $\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \text{noise}$

INDEPENDENT COMPONENTS ANALYSIS (ICA):

independent prior

$\mathbf{z}$ deterministically generates $\mathbf{x}$

$$\text{noise} = 0$$

$$\mathbf{x} = \mathbf{W}\mathbf{z}$$

*train using the **change of variables formula** (see lecture 1)*

Goodfellow *et al.*, 2016, Chapter 13

linear factor models

general form: $\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \text{noise}$

SPARSE CODING:

sparse prior, e.g. Laplace, Cauchy, etc.:

$$p(\mathrm{z}_i) = \text{Laplace}(\mathrm{z}_i; 0, \frac{2}{\lambda}) = \frac{\lambda}{4} e^{-\frac{1}{2}\lambda|\mathrm{z}_i|}$$

x_i are assumed conditionally independent given $\mathbf{z}$

$$\text{noise} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\psi})$$

where $\boldsymbol{\psi} = \text{diag}(\boldsymbol{\sigma}^2)$ *with* $\boldsymbol{\sigma}^2 = \left[\sigma^2, \sigma^2, \ldots, \sigma^2\right]^\intercal$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mathbf{b}, \sigma\mathbf{I})$$

train using approximate inference techniques

Goodfellow *et al.,* 2016, Chapter 13

Olshausen & Field, 1996

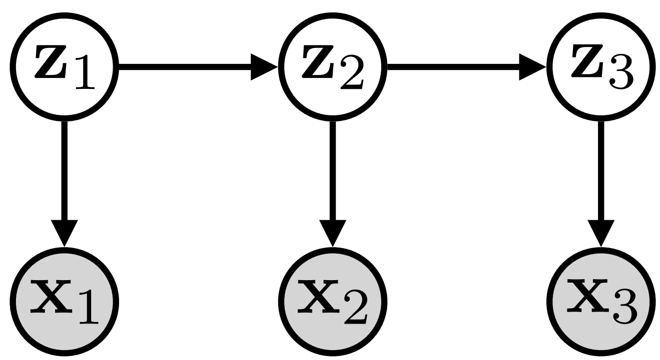
dynamical linear factor models

extend linear Gaussian models to the dynamical setting

LINEAR GAUSSIAN STATE SPACE MODEL (LG-SSM):

transition model: $\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{b}_t + \boldsymbol{\epsilon}_t$ *where* $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$

observation model: $\mathbf{x}_t = \mathbf{C}_t \mathbf{z}_t + \mathbf{d}_t + \boldsymbol{\delta}_t$ *where* $\boldsymbol{\delta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$



Murphy, 2012, Chapter 18

Kalman filtering

LG-SSM:

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{b}_t + \boldsymbol{\epsilon}_t \qquad \textit{where} \;\; \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$$

$$\mathbf{x}_t = \mathbf{C}_t \mathbf{z}_t + \mathbf{d}_t + \boldsymbol{\delta}_t \qquad \textit{where} \;\; \boldsymbol{\delta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$$

performing exact filtering inference: $p(\mathbf{z}_t | \mathbf{x}_{1:t})$

prediction

assume we know $p(\mathbf{z}_t - 1 | \mathbf{x}_{1:t-1}) = \mathcal{N}(\mathbf{z}_{t-1}; \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$

$$\begin{aligned}
p(\mathbf{z}_t | \mathbf{x}_{1:t-1}) &= \int p(\mathbf{z}_t | \mathbf{z}_{t-1}) p(\mathbf{z}_{t-1} | \mathbf{x}_{1:t-1}) d\mathbf{z}_{t-1} \\
&= \int \mathcal{N}(\mathbf{z}_t; \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{b}_t, \mathbf{Q}_1) \mathcal{N}(\mathbf{z}_{t-1}; \boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}) d\mathbf{z}_{t-1} \\
&= \mathcal{N}(\mathbf{z}_t; \mathbf{A}_t \boldsymbol{\mu}_{t-1} + \mathbf{b}_t, \mathbf{A}_t \boldsymbol{\Sigma}_{t-1} \mathbf{A}_t^\intercal + \mathbf{Q}_t) \\
&= \mathcal{N}(\mathbf{z}_t; \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})
\end{aligned}$$

Murphy, 2012, Chapter 18

Kalman filtering

LG-SSM:

$$\mathbf{z}_t = \mathbf{A}_t \mathbf{z}_{t-1} + \mathbf{b}_t + \boldsymbol{\epsilon}_t \qquad \textit{where } \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$$

$$\mathbf{x}_t = \mathbf{C}_t \mathbf{z}_t + \mathbf{d}_t + \boldsymbol{\delta}_t \qquad \textit{where } \boldsymbol{\delta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$$

performing exact filtering inference: $p(\mathbf{z}_t | \mathbf{x}_{1:t})$

update

Bayes' rule: $$p(\mathbf{z}_t | \mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_t | \mathbf{z}_t) p(\mathbf{z}_t | \mathbf{x}_{1:t-1})}{p(\mathbf{x}_t | \mathbf{x}_{1:t-1})}$$ ← *prediction*

⋮

$$p(\mathbf{z}_t | \mathbf{x}_{1:t}) = \mathcal{N}(\mathbf{z}_t; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$$

where $\boldsymbol{\mu}_t = \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t \mathbf{r}_t$, $\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \boldsymbol{\Sigma}_{t|t-1}$

residual: $$\mathbf{r}_t \equiv \mathbf{x}_t - \hat{\mathbf{x}}_t = \mathbf{x}_t - (\mathbf{C}_t \boldsymbol{\mu}_{t|t-1} + \mathbf{d}_t)$$

Kalman gain: $\mathbf{K}_t$

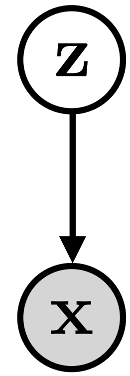
Murphy, 2012, Chapter 18

intractabilities

in simple models, exact inference and marginalization depend on linear Gaussian assumptions

prior: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z})$

conditional likelihood: $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mathbf{b}, \boldsymbol{\Sigma}_\mathbf{x})$



this allowed us to evaluate analytical forms for $p(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{x})$

however, these assumptions limit the model capacity

Goodfellow *et al.,* 2016, Chapter 13

to improve flexibility, allow $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$
to be non-Gaussian and/or have non-linear dependencies

DEEP LATENT GAUSSIAN MODEL:

prior: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

conditional likelihood: $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \boldsymbol{\Sigma}_\theta(\mathbf{z}))$

where $\boldsymbol{\mu}_\theta(\mathbf{z})$ *and* $\boldsymbol{\Sigma}_\theta(\mathbf{z})$ *are deep networks*

however, $p(\mathbf{z}|\mathbf{x})$ *and* $p(\mathbf{x})$ *no longer have tractable analytical forms due to the non-linear deep networks*

cannot tractably evaluate

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$
$$= \int \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \boldsymbol{\Sigma}_\theta(\mathbf{z}))\mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})d\mathbf{z}$$

Kingma & Welling, 2014
Rezende *et al.*, 2014

DEEP LATENT GAUSSIAN MODEL:

prior: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

conditional likelihood: $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \boldsymbol{\Sigma}_\theta(\mathbf{z}))$

where $\boldsymbol{\mu}_\theta(\mathbf{z})$ *and* $\boldsymbol{\Sigma}_\theta(\mathbf{z})$ *are deep networks*

must resort to *approximate inference* methods

Variational Inference:

introduce approximate posterior, e.g.: $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z})$

variational lower bound: $\log p(\mathbf{x}) \geq \mathcal{L}$

optimize $\mathcal{L}$ *w.r.t.* q *and* θ

Variational Autoencoder (VAE):

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\Sigma}_\phi(\mathbf{x}))$$

where $\boldsymbol{\mu}_\phi(\mathbf{x})$ *and* $\boldsymbol{\Sigma}_\phi(\mathbf{x})$ *are deep networks*

Kingma & Welling, 2014

Rezende *et al.*, 2014

how does variational inference relate to exact inference in latent Gaussian models?

for a simplified linear Gaussian model + exact inference:

prior: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

conditional likelihood: $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mathbf{b}, \mathbf{\Sigma}_{\mathbf{x}})$

$\longrightarrow$ **posterior:** $p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}, \mathbf{\Sigma}_{\mathbf{z}|\mathbf{x}})$

can write $\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}} = \mathbf{B}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}})$ *where* $\boldsymbol{\mu}_{\mathbf{x}|\mathbf{z}} = \mathbf{W}\mathbf{z} + \mathbf{b}$

for a deep Gaussian model + variational inference:

prior: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$

conditional likelihood: $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \mathbf{\Sigma}_\theta(\mathbf{z}))$

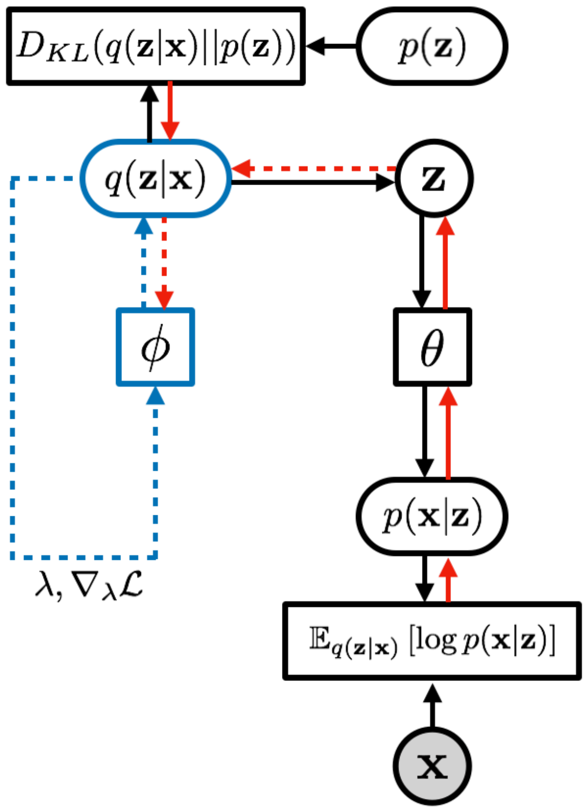
approximate posterior: $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}, \mathbf{\Sigma}_{\mathbf{z}})$

maximize $\mathcal{L}$ *w.r.t.* $\boldsymbol{\mu}_{\mathbf{z}}$

$\longrightarrow$ $\nabla_{\boldsymbol{\mu}_{\mathbf{z}}} \mathcal{L} = \mathbb{E}\left[\mathbf{D}(\mathbf{x} - \boldsymbol{\mu}_\theta(\mathbf{z})) + \mathbf{F}\right]$

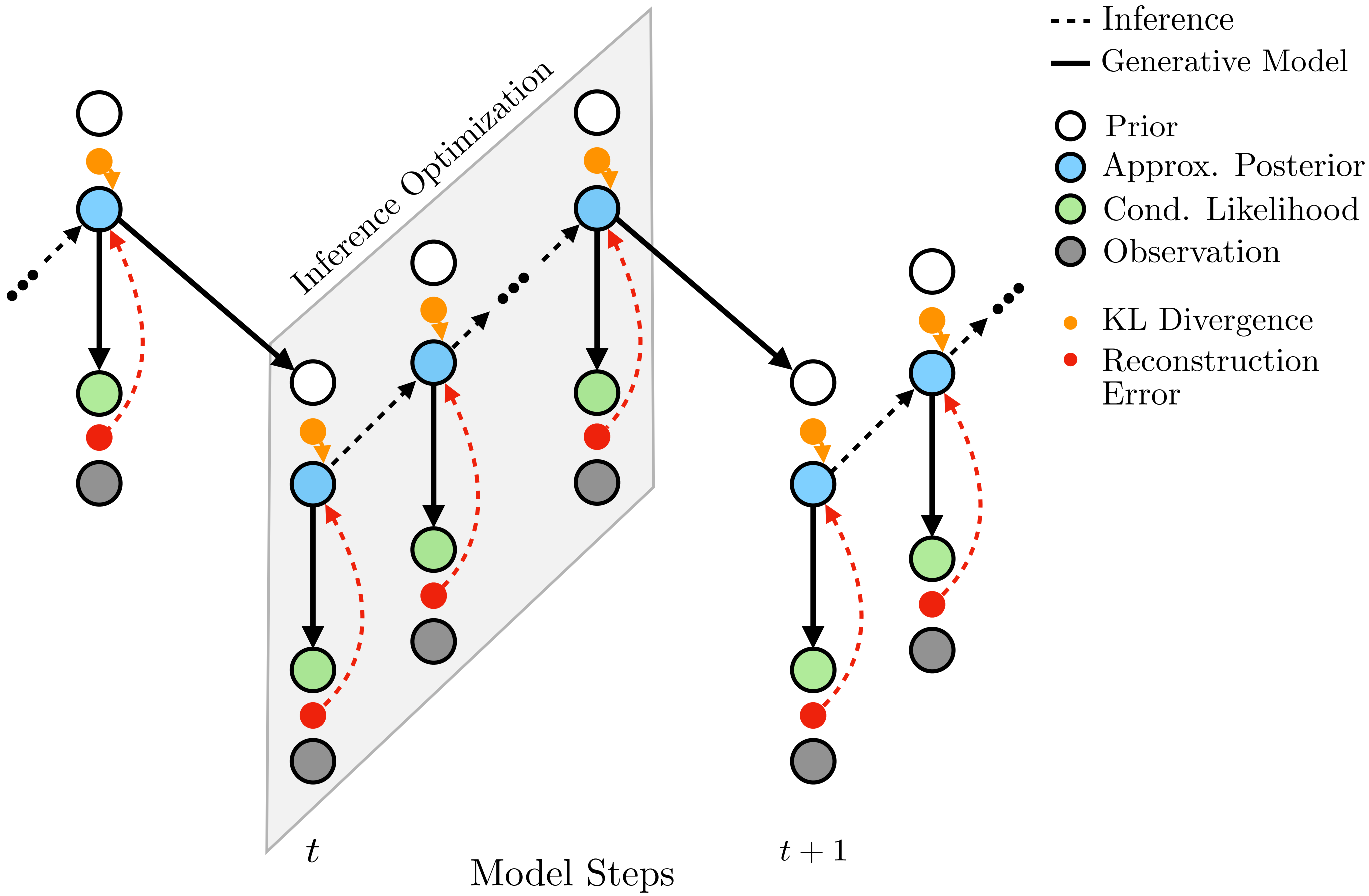
similar terms appear in both inference approaches can also use gradients in an encoder network

$$\boldsymbol{\lambda} = \{\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}}\}$$



Marino *et al.,* Iterative amortized inference, 2018

can also use gradients in an encoder network



Marino *et al.,* A general method for amortizing variational filtering, 2018 next time: deeper dive into latent variable models + variational inference