

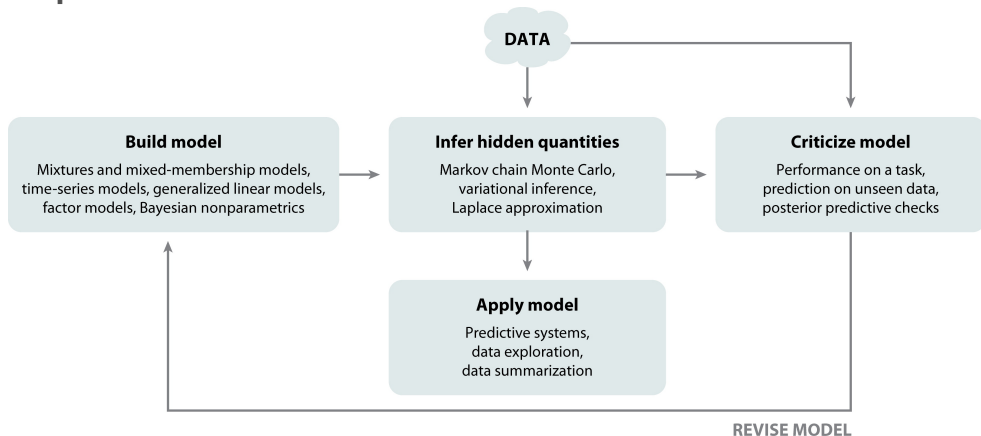
# **STATS271/371: Applied Bayesian Statistics**

## **Factor Analysis, Variational Autoencoders, and Variational EM**

Scott Linderman

May 5, 2021

# Box's Loop



Blei DM. 2014.

Annu. Rev. Stat. Appl. 1:203–32

Blei, *Ann. Rev. Stat. App.* 2014.

## Lap 6: Factor Analysis, Variational Autoencoders, and Variational EM

Let's walk before we run...

- ▶ **Model:** Factor Analysis
- ▶ **Algorithm:** Expectation Maximization

Next time

- ▶ **Model:** Variational Autoencoders
- ▶ **Algorithm:** Variational EM

# The road map

# Factor Analysis

# Notation

Let,

- ▶  $\mathbf{x}_n \in \mathbb{R}^D$  denote the  $n$ -th observation,
- ▶  $\mathbf{z}_n \in \mathbb{R}^P$  denote the corresponding latent variables,
- ▶  $\mathbf{w}_d \in \mathbb{R}^P$  for  $d = 1, \dots, D$  denote the weights each dimension,
- ▶  $\mathbf{W} \in \mathbb{R}^{D \times P}$  denote the matrix with rows  $\mathbf{w}_d$ .
- ▶  $\sigma_d^2 \in \mathbb{R}_+$  for  $d = 1, \dots, D$  denote the variance for each dimension,
- ▶  $\Sigma = \text{diag}([\sigma_1^2, \dots, \sigma_D^2])$  denote the full covariance matrix, and
- ▶  $\Theta = (\mathbf{W}, \Sigma) = \{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D$  denote the set of all parameters.

# The generative process

**Factor analysis** is a “linear Gaussian” latent variable model with the following generative process,

To generate  $N$  observations  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , each in  $\mathbb{R}^D$ ,

1. Sample model parameters  $\{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D$  from their prior distribution.
2. For each observation  $n = 1, \dots, N$ ,
  - a. Sample a **latent variable** from a multivariate Gaussian prior,  $\mathbf{z}_n \sim \mathcal{N}(\mathbf{0}, I)$ .
  - b. For each dimension  $d = 1, \dots, D$ 
    - i. Apply a **linear transformation** to get the mean  $\mathbf{w}_d^\top \mathbf{z}_n$
    - ii. Sample the  $d$ -th coordinate of the **observation**  $x_{n,d} \sim \mathcal{N}(\mathbf{w}_d^\top \mathbf{z}_n, \sigma_d^2)$ .

Note: The last two steps are equivalent to sampling  $\mathbf{x}_n \sim \mathcal{N}(\mathbf{W}\mathbf{z}_n, \Sigma)$  since  $\Sigma$  is diagonal.

# The joint distribution

The joint probability is,

$$p(\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N, \Theta) = p(\Theta) \prod_{n=1}^N p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) \quad (1)$$

$$= p(\Theta) \prod_{n=1}^N \mathcal{N}(\mathbf{x}_n | \mathbf{W}\mathbf{z}_n, \Sigma) \mathcal{N}(\mathbf{z}_n | \mathbf{0}, I) \quad (2)$$

$$= p(\Theta) \prod_{n=1}^N \prod_{d=1}^D \left[ \mathcal{N}(x_n | \mathbf{w}_d^\top \mathbf{z}_n, \sigma_d^2) \right] \mathcal{N}(\mathbf{z}_n | \mathbf{0}, I) \quad (3)$$

We'll place an uninformative prior over the parameters,

$$p(\Theta) = p(\{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D) \propto \prod_{d=1}^D \sigma_d^{-2} \quad (4)$$

(Recall from Lap 1 that this is the uninformative limit of a normal inverse chi-squared distribution,  $\text{Inv-}\chi^2(\sigma_d^2 | \nu, \tau^2) \mathcal{N}(\mathbf{w}_d | \boldsymbol{\mu}, \sigma_d^2 \boldsymbol{\Lambda}^{-1})$ , with  $\nu \rightarrow 0$  and  $\boldsymbol{\Lambda} \rightarrow \mathbf{0}$ .)



## Warm up: Derive the Gibbs updates

Suppose we want to sample the posterior distribution,  $p(\{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D, \{\mathbf{z}_n\}_{n=1}^N \mid \{\mathbf{x}_n\}_{n=1}^N)$ .

The posterior does not have a simple closed form, but as we learned, we can use MCMC algorithms to produce such samples.

The Gibbs sampler is one type of MCMC algorithm. It iteratively samples one (or more) variables from their conditional distribution, holding the rest fixed.

For the factor analysis model, this amounts to repeating the following steps *ad infinitum*:

1. for  $n = 1, \dots, N$ , sample  $\mathbf{z}_n \sim p(\mathbf{z}_n \mid \mathbf{x}_n, \{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D)$
2. for  $d = 1, \dots, D$ , sample  $\mathbf{w}_d, \sigma_d^2 \sim p(\mathbf{w}_d, \sigma_d^2 \mid \{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N)$

(Note: we used the fact that  $\mathbf{z}_n$  is conditionally independent of  $\{\mathbf{x}_{n'}, \mathbf{z}_{n'}\}_{n' \neq n}$  given the parameters and  $\mathbf{w}_d, \sigma_d^2$  are conditionally independent of  $\{\mathbf{w}_{d'}, \sigma_{d'}^2\}_{d'=1}^D$  given the latent variables.)

## Warm up: Derive the Gibbs updates II

**Exercise:** Derive the conditional distribution  $\mathbf{z}_n \sim p(\mathbf{z}_n \mid \mathbf{x}_n, \{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D)$ .

*Hint: it's easier to start with Eq. 2 and work with the matrices  $\mathbf{W}$  and  $\Sigma$ .*

## Warm up: Derive the Gibbs updates III

**Exercise:** Derive the conditional distribution  $p(\mathbf{w}_d, \sigma_d^2 \mid \{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N)$ .

*Hint: Recall the posterior distribution for Bayesian linear regression model from Lap 1.*

## Should we parameters and latent variables the same?

Gibbs samplings aims for a full posterior distribution over both latent variables and parameters. This is great but perhaps overkill...

In probabilistic machine learning, we often talk about *learning* the parameters and *inferring* the parameters.

The distinction makes sense when there is more uncertainty about the latent variables  $\mathbf{z}_n$ , which are informed by a single data point, than about the parameters  $\Theta$ , which depend on all the data.

In that case, we might only need a *point estimate* of the parameters and a *posterior* of the latent variables given the data and parameters.

## Two goals

The **learning goal** is to find the parameters that **maximize the marginal probability of the data**,

$$\Theta^* = \arg \max_{\Theta} p(\{\mathbf{x}_n\}_{n=1}^N, \Theta) \quad (5)$$

$$= \arg \max_{\Theta} p(\Theta) \prod_{n=1}^N \int p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) d\mathbf{z}_n \quad (6)$$

The **inference goal** is to find the **posterior distribution of latent variables**,

$$p(\mathbf{z}_n | \mathbf{x}_n, \Theta) = \frac{p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) p(\Theta)}{p(\mathbf{x}_n, \Theta)} \quad (7)$$

$$= \frac{p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta)}{\int p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) d\mathbf{z}_n} \quad (8)$$

Both goals require an integral over  $\mathbf{z}_n$ , and this could potentially be hard!

# Learning via gradient ascent

Let's start with the learning goal. We could try gradient ascent...

$$\nabla \log p(\{\mathbf{x}_n\}_{n=1}^N, \Theta) = \nabla \log p(\Theta) + \sum_{n=1}^N \nabla \log \int p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) d\mathbf{z}_n \quad (9)$$

$$= \nabla \log p(\Theta) + \sum_{n=1}^N \frac{\int \nabla p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) d\mathbf{z}_n}{\int p(\mathbf{x}_n | \mathbf{z}_n, \Theta) p(\mathbf{z}_n | \Theta) d\mathbf{z}_n} \quad (10)$$

For conjugate exponential family models, we can get a nice closed form gradient expression, but in general this is tricky.

Moreover, gradient ascent requires choices of a step size, and while that's not so difficult, it is a consideration.

## Return of the ELBO

**Idea:** Use the ELBO to get a bound on the marginal probability and maximize that instead.

$$\log p(\{\mathbf{x}_n\}_{n=1}^N, \Theta) = \log p(\{\mathbf{x}_n\}_{n=1}^N \mid \Theta) + \log p(\Theta) \quad (11)$$

$$\geq \mathbb{E}_q \left[ \log p(\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N \mid \Theta) - \log q(\{\mathbf{z}_n\}_{n=1}^N) \right] + \log p(\Theta) \quad (12)$$

$$\triangleq \mathcal{L}(q, \Theta) \quad (13)$$

Here, I've explicitly written the ELBO as a function of both the distribution  $q(\{\mathbf{z}_n\}_{n=1}^N)$  and the parameters  $\Theta$ .

## When is the ELBO tight?

*Show that the ELBO is maximized when  $q(\{\mathbf{z}_n\}_{n=1}^N) = p(\{\mathbf{z}_n\}_{n=1}^N \mid \{\mathbf{x}_n\}_{n=1}^N, \Theta)$ .*



# The Expectation-Maximization (EM) algorithm

The **expectation-maximization (EM) algorithm** maximizes this lower bound by iteratively optimizing wrt  $q$  and  $\Theta$ .

1. Initialize parameters  $\Theta$
2. Repeat until  $\Theta$  (or the ELBO) converges:

**a. E step:** Set

$$q(\{\mathbf{z}_n\}_{n=1}^N) = \arg \max_q \mathcal{L}(q, \Theta) = p(\{\mathbf{z}_n\}_{n=1}^N \mid \{\mathbf{x}_n\}_{n=1}^N, \Theta). \quad (14)$$

At this point, the ELBO is tight and  $\mathcal{L}(q, \Theta) = p(\{\mathbf{x}_n\}_{n=1}^N, \Theta)$ . Track this quantity!

**b. M step:** Set

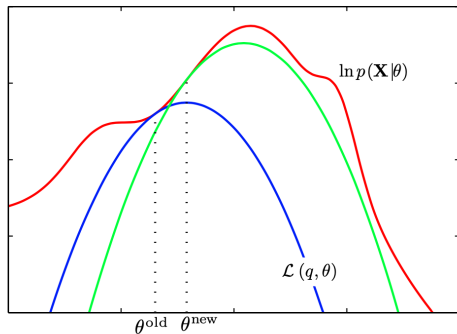
$$\Theta = \arg \max_{\Theta} \mathcal{L}(q, \Theta) \quad (15)$$

$$= \arg \max_{\Theta} \mathbb{E}_q[\log p(\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N \mid \Theta)] + \log p(\Theta). \quad (16)$$

The optimization in the second step can often be done in closed form.

# The EM algorithm in pictures

**Figure 9.14** The EM algorithm involves alternately computing a lower bound on the log likelihood for the current parameter values and then maximizing this bound to obtain the new parameter values. See the text for a full discussion.



From Bishop [2006].

## E step for factor analysis

We've already derived the posterior,

$$p(\{\mathbf{z}_n\}_{n=1}^N \mid \{\mathbf{x}_n\}_{n=1}^N, \Theta) = \prod_{n=1}^N p(\mathbf{z}_n \mid \mathbf{x}_n, \Theta) \quad (17)$$

$$= \prod_{n=1}^N \mathcal{N}(\mathbf{z}_n \mid \boldsymbol{\mu}'_n, \boldsymbol{\Sigma}'_n) \quad (18)$$

where

$$\mathbf{J}'_n = \mathbf{I} + \mathbf{W}^\top \boldsymbol{\Sigma}^{-1} \mathbf{W} \quad \mathbf{h}'_n = \mathbf{W}^\top \boldsymbol{\Sigma}^{-1} \mathbf{x}_n \quad (19)$$

$$\boldsymbol{\Sigma}'_n = \mathbf{J}'_n{}^{-1} \quad \boldsymbol{\mu}'_n = \mathbf{J}'_n{}^{-1} \mathbf{h}'_n. \quad (20)$$

## M step for factor analysis

**Intuition:** Given  $\mathbf{z}$ , this is basically a (Bayesian) linear regression problem.

As a function of  $\Theta = \{\mathbf{w}_d, \sigma_d^2\}_{d=1}^D$ ,

$$\mathcal{L}(q, \Theta) = \mathbb{E}_q[\log p(\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N, \Theta)] \quad (21)$$

$$= \mathbb{E}_q \left[ \sum_{n=1}^N \log p(\mathbf{x}_n | \mathbf{z}_n, \Theta) \right] + \log p(\Theta) + c \quad (22)$$

$$= - \sum_{n=1}^N \sum_{d=1}^D \mathbb{E}_{q(\mathbf{z}_n)} \left[ -\frac{1}{2} \log \sigma_d^2 + \frac{1}{2\sigma_d^2} (x_{n,d} - \mathbf{w}_d^\top \mathbf{z}_n)^2 \right] - \sum_{d=1}^D \log \sigma_d^2 \quad (23)$$

$$= \sum_{d=1}^D \left[ -\frac{N+2}{2} \log \sigma_d^2 - \left\langle \frac{1}{2\sigma_d^2}, \sum_{n=1}^N x_{n,d}^2 \right\rangle + \left\langle \frac{\mathbf{w}_d}{\sigma_d^2}, \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n] x_{n,d} \right\rangle - \left\langle \frac{\mathbf{w}_d \mathbf{w}_d^\top}{2\sigma_d^2}, \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n \mathbf{z}_n^\top] \right\rangle \right] \quad (24)$$

We recognize the term in the square brackets as the log of a normal inverse chi-squared distribution, the posterior in a Bayesian linear regression!

## M step for factor analysis II

Recalling Lap 1, we have

$$\mathcal{L}(q, \Theta) = \mathbb{E}_q[\log p(\{\mathbf{x}_n, \mathbf{z}_n\}_{n=1}^N, \Theta)] \sum_{d=1}^D [\log \text{Inv-}\chi^2(\sigma_d^2 \mid \nu'_d, \tau_d'^2) + \log \mathcal{N}(\mathbf{w}_d \mid \boldsymbol{\mu}'_d, \sigma_d^2 \boldsymbol{\Lambda}'_d)] \quad (25)$$

where

$$\nu'_d = N \qquad \boldsymbol{\Lambda}'_d = \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n \mathbf{z}_n^\top] \quad (26)$$

$$\boldsymbol{\mu}'_d = \boldsymbol{\Lambda}'_d{}^{-1} \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n] x_{n,d} \qquad \tau_d'^2 = \frac{1}{\nu'_d} \left( \sum_{n=1}^N x_{n,d}^2 - \boldsymbol{\mu}'_d{}^\top \boldsymbol{\Lambda}'_d \boldsymbol{\mu}'_d \right) \quad (27)$$

## M step for factor analysis III

The maximum is achieved at the mode of this distribution,

$$\mathbf{w}_d \leftarrow \boldsymbol{\mu}'_d = \left( \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n \mathbf{z}_n^\top] \right)^{-1} \left( \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n] x_{n,d} \right) \quad (28)$$

$$\sigma_d^2 \leftarrow \frac{\nu'_d \tau_d'^2}{\nu'_d + 2} = \frac{1}{N + 2} \left( \sum_{n=1}^N x_{n,d}^2 - \left( \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n] x_{n,d} \right)^\top \left( \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n \mathbf{z}_n^\top] \right)^{-1} \left( \sum_{n=1}^N \mathbb{E}_q[\mathbf{z}_n] x_{n,d} \right) \right) \quad (29)$$

It's exactly like in Lap 1, except here we have **expected sufficient statistics** of the latent variables instead of observed covariates.

# The big picture

EM for latent variable models like factor analysis basically iterates between two steps:

1. Compute the posterior of the latent variables given the current parameters,
2. Find the parameters that maximize the expected log joint probability.

For many models (factor analysis, mixture models, etc.), these two steps can be done exactly.

In practice, EM often converges very quickly, but the objective is non-convex so you might need to try many random initializations!

## Questions for next time

- ▶ What if we had  $x_{n,d} \sim \mathcal{N}(f(\mathbf{z}_n, \mathbf{w}_d), \sigma_d^2)$  for some nonlinear function  $f$ ? Where did we use the linearity assumption?
- ▶ What can we do if the posterior needed for the E step is intractable?
- ▶ What if the M step doesn't have a closed form solution?

Variational autoencoders offer one set of answers to these questions.



## References I

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.