Factor Analysis, Probabilistic Principal Component Analysis, Variational Inference, and Variational Autoencoder: Tutorial and Survey

Benyamin Ghojogh BGHOJOGH@UWATERLOO.CA
Department of Electrical and Computer Engineering,
Machine Learning Laboratory, University of Waterloo, Waterloo, ON, Canada

Ali Ghodsi ALI.GHODSI@UWATERLOO.CA
Department of Statistics and Actuarial Science & David R. Cheriton School of Computer Science,
Data Analytics Laboratory, University of Waterloo, Waterloo, ON, Canada

Fakhri Karray KARRAY@UWATERLOO.CA
Department of Electrical and Computer Engineering,
Centre for Pattern Analysis and Machine Intelligence, University of Waterloo, Waterloo, ON, Canada

Mark Crowley MCROWLEY@UWATERLOO.CA
Department of Electrical and Computer Engineering,
Machine Learning Laboratory, University of Waterloo, Waterloo, ON, Canada

Abstract

This is a tutorial and survey paper on factor analysis, probabilistic Principal Component Analysis (PCA), variational inference, and Variational Autoencoder (VAE). These methods, which are tightly related, are dimensionality reduction and generative models. They asssume that every data point is generated from or caused by a low-dimensional latent factor. By learning the parameters of distribution of latent space, the corresponding low-dimensional factors are found for the sake of dimensionality reduction. For their stochastic and generative behaviour, these models can also be used for generation of new data points in the data space. In this paper, we first start with variational inference where we derive the Evidence Lower Bound (ELBO) and Expectation Maximization (EM) for learning the parameters. Then, we introduce factor analysis, derive its joint and marginal distributions, and work out its EM steps. Probabilistic PCA is then explained, as a special case of factor analysis, and its closed-form solutions are derived. Finally, VAE is explained where the encoder, decoder and sampling from the latent space are introduced. Training VAE using both EM and backpropagation are explained.

1. Introduction

Learning models can be divided into discriminative and generative models (Ng & Jordan, 2002; Bouchard & Triggs, 2004). Discriminative models discriminate the classes of data for better separation of classes while the generative models learn a latent space which generates the data points from a latent space. The methods introduced in this paper are generative models.

Variational inference is a technique which finds a lower bound on the log-likelihood of data and maximizes the lower bound rather than the log-likelihood in the Maximum Likelihood Estimation (MLE). This lower bound is usually referred to as the Evidence Lower Bound (ELBO). Learning the parameters of latent space can be done using Expectation Maximization (EM) (Bishop, 2006). Variational Autoencoder (VAE) (Kingma & Welling, 2014) implements the variational inference in an autoencoder neural network setup where the encoder and decoder model the E-step and M-step of EM, respectively. Although, learning it using backprogatation is usually used in practice (Rezende et al., 2014; Hou et al., 2017). Variational inference and VAE have had many applications in Bayesian analysis; for example, see the application of variational inference in 3D human motion analysis (Sminchisescu & Jepson, 2004) and the application of VAE in forecasting (Walker et al., 2016).

Factor analysis assumes that every data point is generated

from a latent factor/variable where some noise may have been added to data in the data space. Using the EM introduced in variational inference, the ELBO is maximized and the parameters of the latent space are learned iteratively. Probabilistic PCA (PPCA), as a special case of factor analysis, restricts the noise of dimensions to be uncorrelated and assumes the variance of noise to be equal in all dimensions. This restriction makes the solution of PPCA closed-form and simpler.

In this paper, we explain the theory and details of factor analysis, PPCA, variational inference, and VAE. The remainder of this paper is organized as follows. Section 2 introduces variational inference. We explain factor analysis and PPCA in Sections 3 and 4, respectively. VAE is explained in Section 5. Finally, Section 6 concludes the paper.

Required Background for the Reader

This paper assumes that the reader has general knowledge of calculus, probability, linear algebra, and basics of optimization.

2. Variational Inference

Consider a dataset $\{\boldsymbol{x}_i\}_{i=1}^n$. Assume that every data point $\boldsymbol{x}_i \in \mathbb{R}^d$ is generated from a latent variable $\boldsymbol{z}_i \in \mathbb{R}^p$. This latent variable has a prior distribution $\mathbb{P}(\boldsymbol{z}_i)$. According to Bayes' rule, we have:

$$\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i) = \frac{\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i)\, \mathbb{P}(\boldsymbol{z}_i)}{\mathbb{P}(\boldsymbol{x}_i)}. \tag{1}$$

Let $\mathbb{P}(\boldsymbol{z}_i)$ be an arbitrary distribution denoted by $q(\boldsymbol{z}_i)$. Suppose the parameter of conditional distribution of $\boldsymbol{z}_i$ on $\boldsymbol{x}_i$ is denoted by $\boldsymbol{\theta}$; hence, $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i) = \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})$. Therefore, we can say:

$$\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}) = \frac{\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta})\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\theta})}{\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})}. \tag{2}$$

2.1. Evidence Lower Bound (ELBO)

Consider the Kullback-Leibler (KL) divergence (Kullback & Leibler, 1951) between the prior probability of the latent variable and the posterior of the latent variable:

$$\begin{aligned}
&\mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})\big) \\
&\overset{(a)}{=} \int q(\boldsymbol{z}_i) \log\big(\frac{q(\boldsymbol{z}_i)}{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})}\big) d\boldsymbol{z}_i \\
&= \int q(\boldsymbol{z}_i)\big(\log(q(\boldsymbol{z}_i)) - \log(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}))\big) d\boldsymbol{z}_i \\
&\overset{(2)}{=} \int q(\boldsymbol{z}_i)\big(\log(q(\boldsymbol{z}_i)) - \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta})) \\
&\qquad - \log(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\theta})) + \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}))\big)\, d\boldsymbol{z}_i \\
&\overset{(b)}{=} \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) + \int q(\boldsymbol{z}_i)\big(\log(q(\boldsymbol{z}_i)) \\
&\qquad - \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta})) - \log(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\theta}))\big)\, d\boldsymbol{z}_i \\
&= \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) \\
&\qquad + \int q(\boldsymbol{z}_i) \log(\frac{q(\boldsymbol{z}_i)}{\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta})\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\theta})})\, d\boldsymbol{z}_i \\
&= \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) + \int q(\boldsymbol{z}_i) \log(\frac{q(\boldsymbol{z}_i)}{\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})})\, d\boldsymbol{z}_i \\
&= \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) + \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big).
\end{aligned}$$

where (a) is for definition of KL divergence and (b) is because $\log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}))$ is independent of $\boldsymbol{z}_i$ and comes out of integral and $\int d\boldsymbol{z}_i = 1$. Hence:

$$\begin{aligned}
\log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) =& \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})\big) \\
&- \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big).
\end{aligned} \tag{3}$$

We define the *Evidence Lower Bound (ELBO)* as:

$$\mathcal{L}(q, \boldsymbol{\theta}) := -\mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big). \tag{4}$$

So:

$$\log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) = \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})\big) + \mathcal{L}(q, \boldsymbol{\theta}).$$

Therefore:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) - \underbrace{\mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})\big)}_{\geq 0}. \tag{5}$$

As the second term is negative with its minus, the ELBO is a lower bound on the log likelihood of data:

$$\mathcal{L}(q, \boldsymbol{\theta}) \leq \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})). \tag{6}$$

The likelihood $\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})$ is also referred to as the *evidence*. Note that this lower bound gets tight when:

$$\begin{aligned}
&\mathcal{L}(q, \boldsymbol{\theta}) \approx \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta})) \\
&\quad \implies 0 \leq \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta})\big) \overset{\text{set}}{=} 0 \\
&\quad \implies q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}).
\end{aligned} \tag{7}$$

This lower bound is depicted in Fig. 1.

2.2. Expectation Maximization

2.2.1. Background on Expectation Maximization

This part is taken from our previous tutorial paper (Ghojogh et al., 2019a). Sometimes, the data are not fully observable. For example, the data are known to be whether zero or greater than zero. In this case, Maximum Likelihood Expectation (MLE) cannot be directly applied as we do not have access to complete information and some data

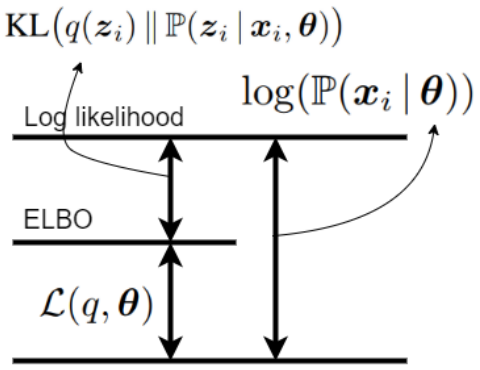

Figure 1. Depiction of ELBO as the lower bound on log likelihood. The image is inspired by (Bishop, 2006).

are missing. In this case, Expectation Maximization (EM) is useful. The main idea of EM can be summarized in this short friendly conversation:

– What shall we do? Some data are missing! The log-likelihood is not known completely so MLE cannot be used.
– Mmm, probably we can replace the missing data with something...
– Aha! Let us replace it with its mean.
– You are right! We can take the mean of log-likelihood over the possible values of the missing data. Then everything in the log-likelihood will be known, and then...
– And then we can do MLE!

EM consists of two steps which are the E-step and the M-step. In the E-step, the log-likelihood is taken expectation with respect to the missing data in order to have a mean estimation of it. In the M-step, the MLE approach is used where the log-likelihood is replaced with its expectation. These two steps are iteratively repeated until convergence of the estimated parameters.

2.2.2. Expectation Maximization in Variational Inference

According to MLE, we want to maximize the log-likelihood of data. According to Eq. (6), maximizing the ELBO will also maximize the log-likelihood. The Eq. (6) holds for any prior distribution q. We want to find the best distribution to maximize the lower bound. Hence, EM for variational inference is performed iteratively as:

$$\text{E-step:} \quad q^{(t)} := \arg\max_q \quad \mathcal{L}(q, \boldsymbol{\theta}^{(t-1)}), \tag{8}$$

$$\text{M-step:} \quad \boldsymbol{\theta}^{(t)} := \arg\max_{\boldsymbol{\theta}} \quad \mathcal{L}(q^{(t)}, \boldsymbol{\theta}), \tag{9}$$

where t denotes the iteration index.

E-step in EM for Variational Inference: The E-step is:

$$\begin{aligned}
\max_q \mathcal{L}(q, \boldsymbol{\theta}^{(t-1)}) &\overset{(5)}{=} \max_q \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}^{(t-1)})) \\
&\quad + \max_q \big(-\mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)})\big)\big) \\
&= \max_q \log(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}^{(t-1)})) \\
&\quad + \min_q \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)})\big).
\end{aligned}$$

The second term is always non-negative; hence, its minimum is zero:

$$\begin{aligned}
&\mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)})\big) \overset{\text{set}}{=} 0 \\
&\implies q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)}),
\end{aligned}$$

which was already found in Eq. (7). Thus, the E-step assigns:

$$q^{(t)}(\boldsymbol{z}_i) \leftarrow \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)}). \tag{10}$$

In other words, in Fig. 1, it pushes the middle line toward the above line by maximizing the ELBO.

M-step in EM for Variational Inference: The M-step is:

$$\begin{aligned}
&\max_{\boldsymbol{\theta}} \mathcal{L}(q^{(t)}, \boldsymbol{\theta}) \overset{(4)}{=} \max_{\boldsymbol{\theta}} \big(-\mathrm{KL}\big(q^{(t)}(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big)\big) \\
&\overset{(a)}{=} \max_{\boldsymbol{\theta}} \Big[-\int q^{(t)}(\boldsymbol{z}_i) \log\big(\frac{q^{(t)}(\boldsymbol{z}_i)}{\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})}\big)\, d\boldsymbol{z}_i\Big] \\
&= \max_{\boldsymbol{\theta}} \int q^{(t)}(\boldsymbol{z}_i) \log(\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}))\, d\boldsymbol{z}_i \\
&\qquad - \max_{\boldsymbol{\theta}} \int q^{(t)}(\boldsymbol{z}_i) \log(q^{(t)}(\boldsymbol{z}_i))\, d\boldsymbol{z}_i,
\end{aligned}$$

where (a) is for definition of KL divergence. The second term is constant w.r.t. $\boldsymbol{\theta}$. Hence:

$$\begin{aligned}
&\max_{\boldsymbol{\theta}} \mathcal{L}(q^{(t)}, \boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \int q^{(t)}(\boldsymbol{z}_i) \log(\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}))\, d\boldsymbol{z}_i \\
&\overset{(a)}{=} \max_{\boldsymbol{\theta}} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big],
\end{aligned}$$

where (a) is because of definition of expectation. Thus, the M-step assigns:

$$\boldsymbol{\theta}^{(t)} \leftarrow \arg\max_{\boldsymbol{\theta}} \ \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big]. \tag{11}$$

In other words, in Fig. 1, it pushes the above line higher. The E-step and M-step together somehow play a game where the E-step tries to reach the middle line (or the ELBO) to the log-likelihood and the M-step tries to increase the above line (or the log-likelihood). This procedure is done repeatedly so the two steps help each other improve to higher values.

To summarize, the EM in variational inference is:

$$q^{(t)}(\boldsymbol{z}_i) \leftarrow \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}^{(t-1)}), \tag{12}$$

$$\boldsymbol{\theta}^{(t)} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta})\big]. \tag{13}$$

It is noteworthy that, in variational inference, sometimes, the parameter $\boldsymbol{\theta}$ is absorbed into the latent variable $\boldsymbol{z}_i$. According to the chain rule, we have:

$$\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i, \boldsymbol{\theta}) = \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta})\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\theta})\, \mathbb{P}(\boldsymbol{\theta}).$$

Considering the term $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\theta})\, \mathbb{P}(\boldsymbol{\theta})$ as one probability, we have:

$$\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i)\, \mathbb{P}(\boldsymbol{z}_i),$$

and the parameter $\boldsymbol{\theta}$ is disappears because of absorption.

3. Factor Analysis

3.1. Background on Marginal Multivariate Gaussian Distribution

Consider two random variables $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{z}_i \in \mathbb{R}^p$ and let $\boldsymbol{y}_i := [\boldsymbol{x}_i^\top, \boldsymbol{z}_i^\top]^\top \in \mathbb{R}^{d+p}$. Assume that $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ jointly multivariate Gaussian; hence, the variable $\boldsymbol{y}_i$ has a multivariate Gaussian distribution, i.e., $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$. The mean and covariance can be decomposed as:

$$\boldsymbol{\mu}_y = [\boldsymbol{\mu}^\top, \boldsymbol{\mu}_0^\top]^\top \in \mathbb{R}^{d+p}, \tag{14}$$

$$\boldsymbol{\Sigma}_y = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \in \mathbb{R}^{(d+p)\times(d+p)}, \tag{15}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$, $\boldsymbol{\mu}_0 \in \mathbb{R}^p$, $\boldsymbol{\Sigma}_{11} \in \mathbb{R}^{d\times d}$, $\boldsymbol{\Sigma}_{22} \in \mathbb{R}^{p\times p}$, $\boldsymbol{\Sigma}_{12} \in \mathbb{R}^{d\times p}$, and $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^\top \in \mathbb{R}^{p\times d}$.

It can be shown that the marginal distributions for $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ are Gaussian distributions where $\mathbb{E}[\boldsymbol{x}_i] = \boldsymbol{\mu}$ and $\mathbb{E}[\boldsymbol{z}_i] = \boldsymbol{\mu}_0$ (Ng, 2018). The covariance matrix of the joint distribution can be simplified as (Ng, 2018):

$$\begin{aligned}
\boldsymbol{\Sigma} &= \mathbb{E}[(\boldsymbol{y}_i - \boldsymbol{\mu}_y)(\boldsymbol{y}_i - \boldsymbol{\mu}_y)^\top] \\
&= \mathbb{E}\left[\begin{bmatrix} \boldsymbol{x}_i - \boldsymbol{\mu} \\ \boldsymbol{z}_i - \boldsymbol{\mu}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_i - \boldsymbol{\mu} \\ \boldsymbol{z}_i - \boldsymbol{\mu}_0 \end{bmatrix}^\top\right] \\
&= \mathbb{E}\left[\begin{bmatrix} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top, (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{z}_i - \boldsymbol{\mu}_0)^\top \\ (\boldsymbol{z}_i - \boldsymbol{\mu}_0)(\boldsymbol{x}_i - \boldsymbol{\mu})^\top, (\boldsymbol{z}_i - \boldsymbol{\mu}_0)(\boldsymbol{z}_i - \boldsymbol{\mu}_0)^\top \end{bmatrix}\right].
\end{aligned} \tag{16}$$

This shows that the marginal distributions are:

$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_{11}), \tag{17}$$

$$\boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_{22}). \tag{18}$$

According to the definition of the multivariate Gaussian distribution, the conditional distribution is also a Gaussian distribution, i.e., $\boldsymbol{x}_i | \boldsymbol{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_{x|z}, \boldsymbol{\Sigma}_{x|z})$ where (Ng, 2018):

$$\mathbb{R}^d \ni \boldsymbol{\mu}_{x|z} := \boldsymbol{\mu} + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{z}_i - \boldsymbol{\mu}_0), \tag{19}$$

$$\mathbb{R}^{d\times d} \ni \boldsymbol{\Sigma}_{x|z} := \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}, \tag{20}$$

and likewise for $\boldsymbol{z}_i | \boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z|x}, \boldsymbol{\Sigma}_{z|x})$:

$$\mathbb{R}^p \ni \boldsymbol{\mu}_{z|x} := \boldsymbol{\mu}_0 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}), \tag{21}$$

$$\mathbb{R}^{p\times p} \ni \boldsymbol{\Sigma}_{z|x} := \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}. \tag{22}$$

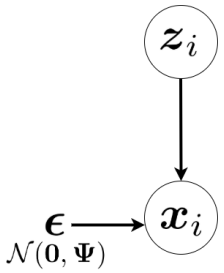

Figure 2. The graphical model for factor analysis. The image is inspired by (Ghahramani & Hinton, 1996).

3.2. Main Idea of Factor Analysis

Factor analysis (Fruchter, 1954; Cattell, 1965; Harman, 1976; Child, 1990) is one of the simplest and most fundamental generative models. Although its theoretical derivations are a little complicated but its main idea is very simple. Factor analysis assumes that every data point $\boldsymbol{x}_i \in \mathbb{R}^d$ is generated from a latent variable $\boldsymbol{z}_i \in \mathbb{R}^p$. The latent variable is also referred to as the latent factor; hence, the name of factor analysis comes from the fact that it analyzes the latent factors.

In factor analysis, we assume that the data point $\boldsymbol{x}_i$ is obtained by linear projection of the p-dimensional $\boldsymbol{z}_i$ onto a d-dimensional space by projection matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times p}$, then applying some linear translation, and finally adding a Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ with covariance matrix $\boldsymbol{\Psi} \in \mathbb{R}^{d\times d}$. Note that as the noises in different dimensions are independent, the covariance matrix $\boldsymbol{\Psi}$ is diagonal. Factor analysis can be illustrated as a graphical model (Ghahramani & Hinton, 1996). Figure 2 shows its graphical model.

3.3. The Factor Analysis Model

For simplicity, this prior distribution of the latent variable can be assumed to be a multivariate Gaussian:

$$\begin{aligned}
\mathbb{P}(\boldsymbol{z}_i) &= \mathcal{N}(\boldsymbol{z}_i \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \\
&= \frac{1}{\sqrt{(2\pi)^p |\boldsymbol{\Sigma}_0|}} \exp\Big(-\frac{(\boldsymbol{z}_i - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{z}_i - \boldsymbol{\mu}_0)}{2}\Big),
\end{aligned} \tag{23}$$

where $\boldsymbol{\mu}_0 \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{p\times p}$ are the mean and the covariance matrix of $\boldsymbol{z}_i$ and $|.|$ is the determinant of matrix. In factor analysis, we assume that the data point $\boldsymbol{x}_i$ is

obtained by linear projection of the p-dimensional $\boldsymbol{z}_i$ onto a d-dimensional space by projection matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times p}$, then applying some linear translation, and finally adding a Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ with covariance matrix $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d}$. Note that as the noises in different dimensions are independent, the covariance matrix $\boldsymbol{\Psi}$ is diagonal. Hence, the data point $\boldsymbol{x}_i$ has a conditional multivariate Gaussian distribution given the latent variable; its conditional likelihood is:

$$\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Psi}) = \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}), \tag{24}$$

where $\boldsymbol{\mu}$, which is the translation vector, is the mean of data $\{\boldsymbol{x}_i\}_{i=1}^n$:

$$\mathbb{R}^d \ni \boldsymbol{\mu} := \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i. \tag{25}$$

The marginal distribution of $\boldsymbol{x}_i$ is:

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x}_i) = \int \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i)\, \mathbb{P}(\boldsymbol{z}_i)\, d\boldsymbol{z}_i \implies \\
&\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Psi}) \\
&\quad = \int \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Psi})\, \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\, d\boldsymbol{z}_i \\
&\quad \overset{(a)}{=} \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{\Lambda}\boldsymbol{\Sigma}_0\boldsymbol{\Lambda}^\top), \qquad (26) \\
&\quad = \mathcal{N}(\widehat{\boldsymbol{\mu}}, \boldsymbol{\Psi} + \widehat{\boldsymbol{\Lambda}}\widehat{\boldsymbol{\Lambda}}^\top), \qquad (27)
\end{aligned}$$

where $\mathbb{R}^d \ni \widehat{\boldsymbol{\mu}} := \boldsymbol{\Lambda}\boldsymbol{\mu}_0 + \boldsymbol{\mu}$, $\mathbb{R}^{d \times d} \ni \widehat{\boldsymbol{\Lambda}} := \boldsymbol{\Lambda}\boldsymbol{\Sigma}_0^{(1/2)}$, and (a) is because mean is linear and variance is quadratic so the mean and variance of projection are applied linearly and quadratically, respectively.

As the mean $\widehat{\boldsymbol{\mu}}$ and covariance $\widehat{\boldsymbol{\Lambda}}$ are needed to be learned, we can absorb $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ into $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$ and assume that $\boldsymbol{\mu}_0 = \boldsymbol{0}$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$.

In summary, factor analysis assumes every data point $\boldsymbol{x}_i \in \mathbb{R}^d$ is obtained by projecting a latent variable $\boldsymbol{z}_i \in \mathbb{R}^p$ onto a d-dimensional space by projection matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times p}$ and translating it by $\boldsymbol{\mu} \in \mathbb{R}^d$ and finally adding some Gaussian noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ (whose dimensions are independent) as:

$$\boldsymbol{x}_i := \boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{28}$$
$$\mathbb{P}(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \tag{29}$$
$$\mathbb{P}(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi}). \tag{30}$$

3.4. The Joint and Marginal Distributions in Factor Analysis

The joint distribution of $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ is:

$$\boldsymbol{y}_i := \begin{bmatrix} \boldsymbol{z}_i \\ \boldsymbol{x}_i \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y). \tag{31}$$

The expectation of $\boldsymbol{x}_i$ is:

$$\mathbb{E}[\boldsymbol{x}_i] \overset{(28)}{=} \mathbb{E}[\boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\Lambda}\mathbb{E}[\boldsymbol{z}_i] + \boldsymbol{\mu} + \mathbb{E}[\boldsymbol{\epsilon}] \overset{(a)}{=} \boldsymbol{\mu}, \tag{32}$$

where (a) is because of Eqs. (29) and (30). Hence:

$$\boldsymbol{\mu}_y := \begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_z \end{bmatrix} \overset{(a)}{=} \begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{0} \end{bmatrix}, \tag{33}$$

where (a) is because of Eqs. (29) and (32). Consider Eq. (15). According to Eq. (29), we have $\boldsymbol{\Sigma}_{22} = \boldsymbol{\Sigma}_z = \boldsymbol{I}$. According to Eq. (28), we have:

$$\begin{aligned}
&\boldsymbol{\Sigma}_{11} = \boldsymbol{\Sigma}_x = \mathbb{E}[(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top] \\
&= \mathbb{E}[(\boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon} - \boldsymbol{\mu})(\boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon} - \boldsymbol{\mu})^\top] \\
&= \mathbb{E}[\boldsymbol{\Lambda}\boldsymbol{z}_i\boldsymbol{z}_i^\top\boldsymbol{\Lambda}^\top + \boldsymbol{\epsilon}\boldsymbol{z}_i^\top\boldsymbol{\Lambda}_i^\top + \boldsymbol{\Lambda}\boldsymbol{z}_i\boldsymbol{\epsilon}^\top + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] \\
&= \boldsymbol{\Lambda}\mathbb{E}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]\boldsymbol{\Lambda}^\top + \boldsymbol{\epsilon}\mathbb{E}[\boldsymbol{z}_i]^\top\boldsymbol{\Lambda}_i^\top + \boldsymbol{\Lambda}\mathbb{E}[\boldsymbol{z}_i]\boldsymbol{\epsilon}^\top + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top] \\
&\overset{(a)}{=} \boldsymbol{\Lambda}\boldsymbol{I}\boldsymbol{\Lambda}^\top + \boldsymbol{0} + \boldsymbol{0} + \boldsymbol{\Psi} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}, \qquad (34)
\end{aligned}$$

where (a) is because of Eqs. (29) and (30). Moreover, we have:

$$\begin{aligned}
&\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{xz} = \mathbb{E}[(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{z}_i - \boldsymbol{\mu}_0)^\top] \\
&\overset{(a)}{=} \mathbb{E}[(\boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu} + \boldsymbol{\epsilon} - \boldsymbol{\mu})(\boldsymbol{z}_i - \boldsymbol{0})^\top] \\
&\overset{(b)}{=} \boldsymbol{\Lambda}\mathbb{E}[\boldsymbol{z}_i\boldsymbol{z}_i^\top] + \mathbb{E}[\boldsymbol{\epsilon}]\mathbb{E}[\boldsymbol{z}_i^\top] = \boldsymbol{\Lambda}\boldsymbol{I} + (\boldsymbol{0}\boldsymbol{0}^\top) = \boldsymbol{\Lambda}, \qquad (35)
\end{aligned}$$

where (a) is because of Eqs. (28) and (29) and (b) is because $\boldsymbol{z}_i$ and $\boldsymbol{\epsilon}$ are independent. We also have $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^\top = \boldsymbol{\Lambda}^\top$. Therefore:

$$\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{z}_i \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi} & \boldsymbol{\Lambda} \\ \boldsymbol{\Lambda}^\top & \boldsymbol{I} \end{bmatrix} \right). \tag{36}$$

Hence, the marginal distribution of data point $\boldsymbol{x}_i$ is:

$$\mathbb{P}(\boldsymbol{x}_i) = \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Psi}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi}). \tag{37}$$

According to Eqs. (21) and (22), the posterior or the conditional distribution of latent variable given data is:

$$\begin{aligned}
q(\boldsymbol{z}_i) \overset{(12)}{=} \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i) &= \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\Lambda}, \boldsymbol{\mu}, \boldsymbol{\Psi}) \\
&= \mathcal{N}(\boldsymbol{\mu}_{z|x}, \boldsymbol{\Sigma}_{z|x}),
\end{aligned} \tag{38}$$

where:

$$\mathbb{R}^p \ni \boldsymbol{\mu}_{z|x} := \boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi})^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}), \tag{39}$$
$$\mathbb{R}^{p \times p} \ni \boldsymbol{\Sigma}_{z|x} := \boldsymbol{I} - \boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi})^{-1}\boldsymbol{\Lambda}. \tag{40}$$

Recall that the conditional distribution of data given the latent variable, i.e. $\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i)$, was introduced in Eq. (24).

If data $\{\boldsymbol{x}_i\}_{i=1}^n$ are centered, i.e. $\boldsymbol{\mu} = \boldsymbol{0}$, the marginal of data, Eq. (37), and the likelihood of data, Eq. (24), become:

$$\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi} + \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top), \tag{41}$$
$$\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{z}_i, \boldsymbol{\Psi}), \tag{42}$$

respectively. In some works, people center the data as a pre-processing to factor analysis.

3.5. Expectation Maximization in Factor Analysis

3.5.1. Maximization of Joint Likelihood

In factor analysis, the parameter $\boldsymbol{\theta}$ of variational inference is the two parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$. As we have in Eq. (13), consider the maximization of joint likelihood, which reduces to the likelihood of data, over all n data points:

$$\begin{aligned}
&\max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\Lambda}, \boldsymbol{\Psi})\big] \\
&\overset{(a)}{=} \max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \sum_{i=1}^{n} \Big(\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\Lambda}, \boldsymbol{\Psi})\big] \\
&\qquad\qquad + \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{z}_i)\big]\Big), \\
&\overset{(b)}{=} \max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\Lambda}, \boldsymbol{\Psi})\big] \\
&\overset{(24)}{=} \max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\log \mathcal{N}(\boldsymbol{\Lambda}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\big] \\
&= \max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\Big[\log \big(\frac{1}{(2\pi)^{p/2}|\boldsymbol{\Psi}|^{1/2}} \\
&\quad \exp\Big(-\frac{(\boldsymbol{z}_i - \boldsymbol{\Lambda}\boldsymbol{z}_i - \boldsymbol{\mu})^\top \boldsymbol{\Psi}^{-1}(\boldsymbol{z}_i - \boldsymbol{\Lambda}\boldsymbol{z}_i - \boldsymbol{\mu})}{2}\Big)\big)\Big] \\
&= \max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \Big(\underbrace{-\frac{p\,n}{2}\log(2\pi)}_{\text{constant}} - \frac{n}{2}\log|\boldsymbol{\Psi}| \\
&\quad - \sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\Lambda}\boldsymbol{z}_i - \boldsymbol{\mu})^\top \boldsymbol{\Psi}^{-1} \\
&\qquad\qquad (\boldsymbol{x}_i - \boldsymbol{\Lambda}\boldsymbol{z}_i - \boldsymbol{\mu})\big]\Big) \qquad (43)
\end{aligned}$$

where (a) is because of the chain rule $\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) = \mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\Lambda}, \boldsymbol{\Psi})\,\mathbb{P}(\boldsymbol{z}_i)$, and (b) is because the second term is zero because of zero mean of prior of $\boldsymbol{z}_i$ (see Eq. (29)).

3.5.2. The E-Step in EM for Factor Analysis

As we will see later in the M-step of EM, we will have two expectation terms which need to be computed in the E-step. These expectations, which are over the $q(\boldsymbol{z}_i)$ distribution, are $\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]$ and $\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]$. According to Eq. (12), we have $q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i)$. Therefore, according to Eqs. (38), (39), and (40), we have:

$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i] = \boldsymbol{\mu}_{z|x} := \boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi})^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}), \tag{44}$$

$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top] = \boldsymbol{\Sigma}_{z|x} := \boldsymbol{I} - \boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \boldsymbol{\Psi})^{-1}\boldsymbol{\Lambda}. \tag{45}$$

3.5.3. The M-Step in EM for Factor Analysis

Finding parameter $\boldsymbol{\Lambda}$: We have two variables $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ so we solve the maximization w.r.t. these variables.

$$\begin{aligned}
&\mathbb{R}^{d\times p} \ni \frac{\partial \text{Eq. (43)}}{\partial \boldsymbol{\Lambda}} \\
&= -\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\Lambda}} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\Big[\frac{1}{2}\textbf{tr}(\boldsymbol{z}_i^\top \boldsymbol{\Lambda}^\top \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\boldsymbol{z}_i) \\
&\qquad\qquad - \textbf{tr}(\boldsymbol{z}_i^\top \boldsymbol{\Lambda}^\top \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}))\Big] \\
&\overset{(a)}{=} -\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\Lambda}} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\Big[\frac{1}{2}\textbf{tr}(\boldsymbol{\Lambda}^\top \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\boldsymbol{z}_i\boldsymbol{z}_i^\top) \\
&\qquad\qquad - \textbf{tr}(\boldsymbol{\Lambda}^\top \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\boldsymbol{z}_i^\top)\Big] \\
&= -\sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\Big[\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\boldsymbol{z}_i\boldsymbol{z}_i^\top - \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\boldsymbol{z}_i^\top\Big] \\
&= -\sum_{i=1}^{n} \Big[\boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top] \\
&\qquad\qquad - \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]^\top\Big],
\end{aligned}$$

where (a) is because of the cyclic property of trace. Setting this derivative to zero gives us the optimum $\boldsymbol{\Lambda}$:

$$\begin{aligned}
&\sum_{i=1}^{n} \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top] \\
&\qquad = \sum_{i=1}^{n} \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]^\top \\
&\implies \boldsymbol{\Lambda} = \Big(\sum_{i=1}^{n} \boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]^\top\Big) \\
&\qquad \Big(\sum_{i=1}^{n} \boldsymbol{\Psi}^{-1}\boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]\Big)^{-1}. \qquad (46)
\end{aligned}$$

Finding parameter $\boldsymbol{\Psi}$: Now, consider maximization w.r.t $\boldsymbol{\Psi}$. We restate Eq. (43) as (Paola Garcia, 2018):

$$\max_{\boldsymbol{\Lambda},\boldsymbol{\Psi}} \Big(\underbrace{-\frac{pn}{2}\log(2\pi)}_{\text{constant}} - \frac{n}{2}\log|\boldsymbol{\Psi}| - \frac{n}{2}\textbf{tr}(\boldsymbol{\Psi}^{-1}\boldsymbol{S})\Big), \tag{47}$$

where $\boldsymbol{S} \in \mathbb{R}^{d\times d}$ is a sample covariance matrix defined as:

$$\boldsymbol{S} := \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}\big[(\boldsymbol{x}_i - \boldsymbol{\Lambda}\boldsymbol{z}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\Lambda}\boldsymbol{z}_i - \boldsymbol{\mu})^\top\big] \tag{48}$$

$$\begin{aligned}
&= \frac{1}{n}\sum_{i=1}^{n} \Big((\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \\
&- 2\boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i](\boldsymbol{x}_i - \boldsymbol{\mu})^\top + \boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}\boldsymbol{z}^\top]\boldsymbol{\Lambda}\Big).
\end{aligned} \tag{49}$$

The maximization is (Paola Garcia, 2018):

$$\mathbb{R}^{d\times d} \ni \frac{\partial \text{Eq. (47)}}{\partial \boldsymbol{\Psi}^{-1}} = \frac{n}{2}\boldsymbol{\Psi} - \frac{n}{2}\boldsymbol{S} \overset{\text{set}}{=} \mathbf{0} \implies \boldsymbol{\Psi} = \boldsymbol{S}.$$

Note that as the dimensions of noise $\boldsymbol{\epsilon} \in \mathbb{R}^d$ are independent, the covariance matrix of noise, $\boldsymbol{\Psi}$, is a diagonal matrix. Hence:

$$\boldsymbol{\Psi} = \textbf{diag}(\boldsymbol{S}) \overset{(49)}{=} \frac{1}{n}\textbf{diag}\Big(\sum_{i=1}^{n}\Big[(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^\top - 2\boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i](\boldsymbol{x}_i-\boldsymbol{\mu})^\top + \boldsymbol{\Lambda}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}\boldsymbol{z}^\top]\boldsymbol{\Lambda}\Big]\Big). \tag{50}$$

3.5.4. Summary of Factor Analysis Algorithm

According to the derived equations, the EM algorithm in factor analysis is summarized as follows. The mean of data, $\boldsymbol{\mu}$, is computed. Then, for every data point $\boldsymbol{x}_i$, we iteratively solve as:

$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i] \leftarrow \boldsymbol{\Lambda}^{(t)\top}(\boldsymbol{\Lambda}^{(t)}\boldsymbol{\Lambda}^{(t)\top} + \boldsymbol{\Psi}^{(t)})^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu}),$$
$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top] \leftarrow \boldsymbol{I} - \boldsymbol{\Lambda}^{(t)\top}(\boldsymbol{\Lambda}^{(t)}\boldsymbol{\Lambda}^{(t)\top} + \boldsymbol{\Psi}^{(t)})^{-1}\boldsymbol{\Lambda}^{(t)},$$
$$\boldsymbol{\Lambda}^{(t+1)} \leftarrow \Big(\sum_{i=1}^{n}(\boldsymbol{\Psi}^{(t)})^{-1}(\boldsymbol{x}_i-\boldsymbol{\mu})\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]^\top\Big)\Big(\sum_{i=1}^{n}(\boldsymbol{\Psi}^{(t)})^{-1}\boldsymbol{\Lambda}^{(t)}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]\Big)^{-1}.$$
$$\boldsymbol{\Psi}^{(t+1)} \leftarrow \frac{1}{n}\textbf{diag}\Big(\sum_{i=1}^{n}\Big[(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^\top - 2\boldsymbol{\Lambda}^{(t+1)}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i](\boldsymbol{x}_i-\boldsymbol{\mu})^\top + \boldsymbol{\Lambda}^{(t+1)}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}\boldsymbol{z}^\top]\boldsymbol{\Lambda}^{(t+1)}\Big]\Big).$$

Note that if data are centered as a pre-processing to factor analysis, i.e. $\boldsymbol{\mu} = \mathbf{0}$, the algorithm of factor analysis is simplified as:

$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i] \leftarrow \boldsymbol{\Lambda}^{(t)\top}(\boldsymbol{\Lambda}^{(t)}\boldsymbol{\Lambda}^{(t)\top} + \boldsymbol{\Psi}^{(t)})^{-1}\boldsymbol{x}_i,$$
$$\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top] \leftarrow \boldsymbol{I} - \boldsymbol{\Lambda}^{(t)\top}(\boldsymbol{\Lambda}^{(t)}\boldsymbol{\Lambda}^{(t)\top} + \boldsymbol{\Psi}^{(t)})^{-1}\boldsymbol{\Lambda}^{(t)},$$
$$\boldsymbol{\Lambda}^{(t+1)} \leftarrow \Big(\sum_{i=1}^{n}(\boldsymbol{\Psi}^{(t)})^{-1}\boldsymbol{x}_i\,\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]^\top\Big)\Big(\sum_{i=1}^{n}(\boldsymbol{\Psi}^{(t)})^{-1}\boldsymbol{\Lambda}^{(t)}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]\Big)^{-1}.$$
$$\boldsymbol{\Psi}^{(t+1)} \leftarrow \frac{1}{n}\textbf{diag}\Big(\sum_{i=1}^{n}\Big[\boldsymbol{x}_i\boldsymbol{x}_i^\top - 2\boldsymbol{\Lambda}^{(t+1)}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}_i]\,\boldsymbol{x}_i^\top + \boldsymbol{\Lambda}^{(t+1)}\mathbb{E}_{\sim q^{(t)}(\boldsymbol{z}_i)}[\boldsymbol{z}\boldsymbol{z}^\top]\boldsymbol{\Lambda}^{(t+1)}\Big]\Big).$$

As it can be seen, factor analysis does not have a closed-form solution and its solution, which are the projection matrix $\boldsymbol{\Lambda}$ and the noise covariance matrix $\boldsymbol{\Psi}$, are found iteratively until convergence.

It is noteworthy that mixture of factor analysis (Ghahramani & Hinton, 1996) also exists in the literature which considers a mixture distribution for the factor analysis and trains the parameters of mixture using EM (Ghojogh et al., 2019a).

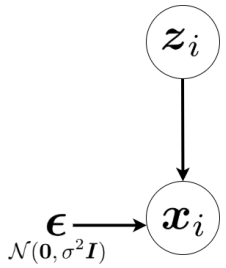

Figure 3. The graphical model for PPCA.

4. Probabilistic Principal Component Analysis

4.1. Main Idea of Probabilistic PCA

Probabilistic PCA (PPCA) (Roweis, 1997; Tipping & Bishop, 1999b) is a special case of factor analysis where the variance of noise is equal in all dimensions of data space with covariance between dimensions, i.e.:

$$\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}. \tag{51}$$

In other words, PPCA considers an isotropic noise model. Therefore, Eq. (30) is simplified to:

$$\mathbb{P}(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I}). \tag{52}$$

Because of having zero covariance of noise between different dimensions, PPCA assumes that the data points are independent of each other given latent variables. PPCA can be illustrated as a graphical model. Figure 3 shows its graphical model.

4.2. MLE for Probabilistic PCA

As PPCA is a special case of factor analysis, it also is solved using EM. Similar to factor analysis, it can be solved iteratively using EM (Roweis, 1997). However, one can also find a closed-form solution to its EM approach (Tipping & Bishop, 1999b). Hence, by restricting the noise covariance to be isotropic, its solution becomes simpler and closed-form. The iterative approach is as we had in factor analysis. Here, we derive the closed-form solution.

Consider the likelihood or the marginal distribution of data points $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ which is Eq. (37). The log-likelihood

of data is:

$$\sum_{i=1}^{n} \log \mathbb{P}(\boldsymbol{x}_i) \overset{(37)}{=} \sum_{i=1}^{n} \log \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})$$
$$= \sum_{i=1}^{n} \Big[\log \big(\frac{1}{(2\pi)^{d/2} |\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I}|^{1/2}} \times \exp\Big(-\frac{(\boldsymbol{x}_i - \boldsymbol{\mu})^\top (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})}{2} \Big) \big) \Big]$$

$$= \underbrace{-\frac{d\,n}{2} \log(2\pi)}_{\text{constant}} - \frac{n}{2} \log |\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I}| - \sum_{i=1}^{n} \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu})^\top (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})$$
$$= \underbrace{-\frac{d\,n}{2} \log(2\pi)}_{\text{constant}} - \frac{n}{2} \log |\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I}| - \frac{n}{2} \textbf{tr}\big((\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{S}_x \big),$$

where $\boldsymbol{S}_x \in \mathbb{R}^{d \times d}$ is the sample covariance matrix of data:

$$\boldsymbol{S}_x := \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top. \tag{53}$$

We use MLE where the variables of maximization optimization are the projection matrix $\boldsymbol{\Lambda}$ and the noise variance σ:

$$\max_{\boldsymbol{\Lambda}, \sigma} \Big(\underbrace{-\frac{d\,n}{2} \log(2\pi)}_{\text{constant}} - \frac{n}{2} \log |\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I}| - \frac{n}{2} \textbf{tr}\big((\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{S}_x \big) \Big). \tag{54}$$

It is noteworthy that literature usually defines:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{C} := (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I}_{d \times d}). \tag{55}$$
$$\mathbb{R}^{p \times p} \ni \boldsymbol{M} := (\boldsymbol{\Lambda}^\top \boldsymbol{\Lambda} + \sigma^2 \boldsymbol{I}_{p \times p}). \tag{56}$$

According to the matrix inversion lemma, we have:

$$\boldsymbol{C}^{-1} = \sigma^{-1} \boldsymbol{I}_{d \times d} - \sigma^{-2} \boldsymbol{\Lambda} \boldsymbol{M}^{-1} \boldsymbol{\Lambda}^\top. \tag{57}$$

This inversion is interesting because the inverse of a $(d \times d)$ matrix $\boldsymbol{C}$ is reduced to inversion of a $(p \times p)$ matrix $\boldsymbol{M}$ which is much simpler because we usually have $p \ll d$.

4.2.1. MLE FOR DETERMINING $\boldsymbol{\Lambda}$

Taking the derivative of Eq. (54) w.r.t. $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times p}$ and setting it to zero is:

$$\frac{\partial\, \text{Eq.}(54)}{\partial \boldsymbol{\Lambda}} = -n\big((\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{S}_x (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{\Lambda} - (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{\Lambda} \big) \overset{\text{set}}{=} \boldsymbol{0}$$
$$\implies (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{S}_x (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{\Lambda} = (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{\Lambda}$$
$$\implies \boldsymbol{S}_x (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{\Lambda} = \boldsymbol{\Lambda},$$

whose trivial solutions are $\boldsymbol{\Lambda} = \boldsymbol{0}$ and $\boldsymbol{S}_x = (\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2 \boldsymbol{I})$ which are not true and valid. For the non-trivial solution, consider the Singular Value Decomposition (SVD) $\mathbb{R}^{d \times p} \ni \boldsymbol{\Lambda} = \boldsymbol{U}\boldsymbol{L}\boldsymbol{V}^\top$ where $\boldsymbol{U} \in \mathbb{R}^{d \times p}$ and $\boldsymbol{V} \in \mathbb{R}^{p \times p}$ contain the left and right singular vectors, respectively, and $\boldsymbol{L} \in \mathbb{R}^{p \times p}$ is the diagonal matrix containing the singular values denoted by $\{l_j\}_{j=1}^{p}$. Moreover, note that $\textbf{tr}(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top) = \textbf{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{V}^\top \boldsymbol{V}\boldsymbol{L}\boldsymbol{U}^\top) = \textbf{tr}(\boldsymbol{U}\boldsymbol{L}\boldsymbol{L}\boldsymbol{U}^\top) = \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{L}^2) = \textbf{tr}(\boldsymbol{L}^2)$ because $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices.

From the previous calculations, we have (Hauskrecht, 2007):

$$\boldsymbol{S}_x \boldsymbol{U}\boldsymbol{L}(\boldsymbol{L}^2 + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{V}^\top = \boldsymbol{U}\boldsymbol{L}\boldsymbol{V}^\top$$
$$\implies \boldsymbol{S}_x \boldsymbol{U}\boldsymbol{L}(\boldsymbol{L}^2 + \sigma^2 \boldsymbol{I})^{-1} = \boldsymbol{U}\boldsymbol{L}$$
$$\implies \boldsymbol{S}_x \boldsymbol{U}\boldsymbol{L} = \boldsymbol{U}(\boldsymbol{L}^2 + \sigma^2 \boldsymbol{I})\boldsymbol{L}$$
$$\implies \boldsymbol{S}_x \boldsymbol{U} = \boldsymbol{U}(\boldsymbol{L}^2 + \sigma^2 \boldsymbol{I}), \tag{58}$$

which is an eigenvalue problem (Ghojogh et al., 2019b) for the covariance matrix $\boldsymbol{S}_x$ where the columns of $\boldsymbol{U}$ are the eigenvectors of $\boldsymbol{S}_x$ and the eigenvalues are $\sigma^2 + l_j^2$. Recall that σ is the variance of noise in different dimensions and l_j is the j-th singular value of $\boldsymbol{\Lambda}$ (sorted from largest to smallest). We denote the j-th eigenvalue of the covariance matrix $\boldsymbol{S}_x$ by:

$$\delta_j := \sigma^2 + l_j^2 \implies l_j = (\delta_j - \sigma^2)^{(1/2)}. \tag{59}$$

We consider only the top p singular values l_j and the top p eigenvalues δ_j; substituting the singular values in the SVD of the projection matrix $\boldsymbol{\Lambda}$ results in:

$$\boldsymbol{\Lambda} = \boldsymbol{U}\boldsymbol{L}\boldsymbol{V}^\top = \boldsymbol{U}(\boldsymbol{\Delta}_p - \sigma^2 \boldsymbol{I})^{(1/2)} \boldsymbol{V}^\top,$$

where $\boldsymbol{\Delta}_p := \textbf{diag}(\delta_1, \dots, \delta_p)$. However, as Eq. (58) does not include any $\boldsymbol{V}$, we can replace $\boldsymbol{V}^\top$ with any arbitrary orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{p \times p}$ (Tipping & Bishop, 1999b):

$$\boldsymbol{\Lambda} = \boldsymbol{U}(\boldsymbol{\Delta}_p - \sigma^2 \boldsymbol{I})^{(1/2)} \boldsymbol{R}. \tag{60}$$

The arbitrary orthogonal matrix $\boldsymbol{R}$ is a rotation matrix which rotates data in projection. It is arbitrary because rotation is not important in manifold learning. A simple choice for this rotation matrix is $\boldsymbol{R} = \boldsymbol{I}$ which results in:

$$\boldsymbol{\Lambda} = \boldsymbol{U}(\boldsymbol{\Delta}_p - \sigma^2 \boldsymbol{I})^{(1/2)}. \tag{61}$$

4.2.2. MLE FOR DETERMINING σ

If we substitute Eq. (60) in the log-likelihood, Eq. (54), the log-likelihood becomes (Hauskrecht, 2007):

$$\max_{\sigma} \; -\frac{n}{2}\Big(\underbrace{d\log(2\pi)}_{\text{constant}} + \sum_{j=1}^{p}\log(\delta_j) + \sum_{j=p+1}^{d}\delta_j + (d-p)\log((\sigma^2)^{-1}) + p\Big). \tag{62}$$

Note that we have d eigenvalues $\{\delta_j\}_{j=1}^{d}$ because the covariance matrix $\boldsymbol{S}_x$ is a $(d\times d)$ matrix. However, as we have only p singular values $\{l_j\}_{j=1}^{p}$, the eigenvalues $\{\delta_j\}_{j=p+1}^{d}$ are very small.

Taking the derivative of Eq. (62) w.r.t. σ^2 and setting it to zero is (Tipping & Bishop, 1999b):

$$\frac{\partial \text{Eq.}(62)}{\partial \sigma^2} = -\frac{n}{2}\big(0 + \sum_{j=p+1}^{d}\delta_j + (d-p)\,\sigma^2 + 0\big) \overset{\text{set}}{=} 0$$

$$\implies \sigma^2 = \frac{1}{d-p}\sum_{j=p+1}^{d}\delta_j. \tag{63}$$

4.2.3. SUMMARY OF MLE FORMULAS

In summary, the MLE estimations for the variables of PPCA are:

$$\sigma^2 = \frac{1}{d-p}\sum_{j=p+1}^{d}\delta_j, \tag{64}$$

$$\boldsymbol{\Lambda} = \boldsymbol{U}(\boldsymbol{\Delta}_p - \sigma^2\boldsymbol{I})^{(1/2)}\boldsymbol{R} = \boldsymbol{U}(\boldsymbol{\Delta}_p - \sigma^2\boldsymbol{I})^{(1/2)}. \tag{65}$$

Note that Eq. (64) is a measure of the variance lost in the projection by the projection matrix $\boldsymbol{\Lambda}$. The Eq. (65) is the projection or mapping matrix from the latent space to the data space.

4.3. Zero Noise Limit: PCA Is a Special Case of Probabilistic PCA

Recall the posterior which is Eq. (38). According to Eqs. (39), (40), and (51), the posterior in PPCA is:

$$q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i) = \mathcal{N}\Big(\boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}), \; \boldsymbol{I} - \boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Lambda}\Big). \tag{66}$$

Consider zero noise limit where the variance of noise goes to zero:

$$\lim_{\sigma^2\to 0}\mathbb{P}(\boldsymbol{\epsilon}) = \lim_{\sigma^2\to 0}\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I}) = \mathcal{N}(\boldsymbol{0}, \lim_{\sigma^2\to 0}(\sigma^2)\boldsymbol{I}). \tag{67}$$

In this case, the uncertainty of PPCA almost disappears. In the following, we show that in zero noise limit, PPCA is reduced to PCA (Ghojogh & Crowley, 2019; Jolliffe & Cadima, 2016) and this explains why the PPCA method is a probabilistic approach to PCA.

In the zero noise limit, the posterior becomes:

$$\begin{aligned}\lim_{\sigma^2\to 0} q(\boldsymbol{z}_i) &= \lim_{\sigma^2\to 0}\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i) \\ &\overset{(66)}{=} \mathcal{N}\Big(\boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top)^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}), \boldsymbol{I} - \boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top)^{-1}\boldsymbol{\Lambda}\Big) \\ &\overset{(a)}{=} \mathcal{N}\Big((\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^\top(\boldsymbol{x}_i - \boldsymbol{\mu}), \boldsymbol{I} - (\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda}\Big).\end{aligned} \tag{68}$$

where (a) is because according to (Roweis, 1997, footnote 4), we have:

$$\boldsymbol{\Lambda}^\top(\boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top)^{-1} = (\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^\top. \tag{69}$$

On the other hand, according to (Tipping & Bishop, 1999b, Appendix C), PCA minimizes the reconstruction error as:

$$\underset{\boldsymbol{\Lambda}}{\text{minimize}} \quad ||(\boldsymbol{X} - \boldsymbol{\mu}) - \boldsymbol{\Lambda}\boldsymbol{\Lambda}^\top(\boldsymbol{X} - \boldsymbol{\mu})||_F^2, \tag{70}$$

where $\|\|_F$ denotes the Frobenius norm of matrix. See (Ghojogh & Crowley, 2019) for more details on minimization of reconstruction error by PCA.

Instead of minimizing the reconstruction error, one may minimize the reconstruction error for the mean of posterior (Tipping & Bishop, 1999b, Appendix C):

$$\underset{\boldsymbol{\Lambda}}{\text{minimize}} \quad ||(\boldsymbol{X} - \boldsymbol{\mu}) - \boldsymbol{\Lambda}(\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^\top(\boldsymbol{X} - \boldsymbol{\mu})||_F^2. \tag{71}$$

This is the minimization of reconstruction error after projection by $(\boldsymbol{\Lambda}^\top\boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}^\top$. According to the posterior in the zero noise limit (see Eq. (68)), this is equivalent to PPCA in the zero noise limit model. Hence, PCA is a deterministic special case of PPCA where the variance of noise goes to zero.

4.4. Other Variants of Probabilistic PCA

There exist some other variants of PPCA. We briefly mention them here. PPCA has a hyperparameter which is the dimensionality of latent space p. Bayesian PCA (Bishop, 1999) models this hyperparameter as another latent random variable which is learned during the EM training.

According to Eqs. (28), (29), and (52), note that PPCA uses Gaussian distributions. PPCA with Student-t distribution (Zhao & Jiang, 2006) has been proposed which uses t distributions. This change may improve the embedding of PPCA because of the heavier tails of Student-t distribution compared to Gaussian distribution. This avoids the crowding problem which has motivated the proposal of t-SNE (Ghojogh et al., 2020a).

Sparse PPCA (Guan & Dy, 2009; Mattei et al., 2016) has inserted sparsity to PPCA. Supervised PPCA (Yu et al., 2006) makes use of class labels in the formulation of

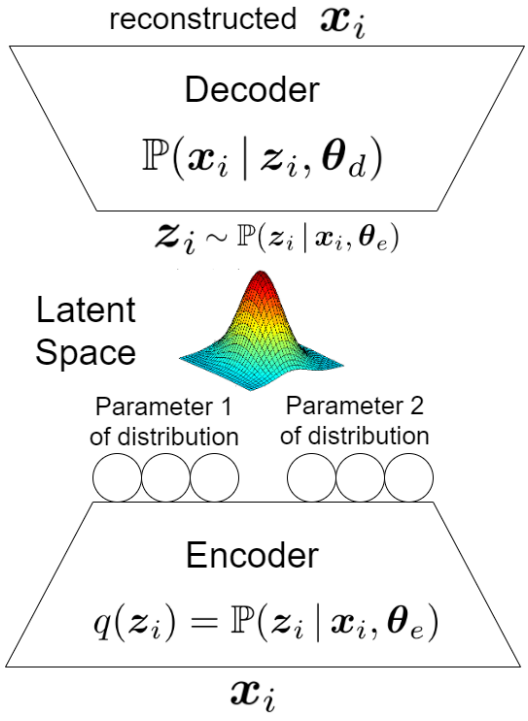

Figure 4. The structure of a variational autoencoder.

PPCA. Mixture of PPCA (Tipping & Bishop, 1999a) uses mixture of distributions in the formulation PPCA. It trains the parameters of a mixture distribution using EM (Ghojogh et al., 2019a). Generalized PPCA for correlated data is another recent variant of PPCA (Gu & Shen, 2020).

5. Variational Autoencoder

Variational Autoencoder (VAE) (Kingma & Welling, 2014) applies variational inference, i.e., maximizes the ELBO, but in an autoencoder setup and makes it differentiable for the backpropagation training (Rumelhart et al., 1986). As Fig. 4 shows, VAE includes an encoder and a decoder each of each can have several network layers. A latent space is learned between the encoder and decoder. The latent variable $\boldsymbol{z}_i$ is sampled from the latent space. In the following, we explain the encoder and decoder parts. The input of encoder in VAE is the data point $\boldsymbol{x}_i$ and the output of decoder in VAE is its reconstruction $\boldsymbol{x}_i$.

5.1. Parts of Variational Autoencoder

5.1.1. Encoder of Variational Autoencoder

The encoder of VAE models the distribution $q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ where the parameters of distribution $\boldsymbol{\theta}_e$ are the weights of encoder layers in VAE. The input and output of encoder are $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{z}_i \in \mathbb{R}^p$, respectively. As Fig. 4 depicts, the output neurons of encoder are supposed to determine the parameters of the conditional distribution $\mathbb{P}(\boldsymbol{z}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}_e)$. If this conditional distribution has m number of parameters, we have m sets of output neurons from the encoder, denoted by $\{\boldsymbol{e}_j\}_{j=1}^m$. The dimensionality of these sets may defer depending the size of every parameters.

For example, let the latent space be p-dimensional, i.e., $\boldsymbol{z}_i \in \mathbb{R}^p$. If the distribution $\mathbb{P}(\boldsymbol{z}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ is a multivariate Gaussian distribution, we have two sets of output neurons for encoder where one set has p neurons for the mean of this distribution $\boldsymbol{\mu}_{z|x} = \boldsymbol{e}_1 \in \mathbb{R}^d$ and the other set has $(p \times p)$ neurons for the covariance of this distribution $\boldsymbol{\Sigma}_{z|x} =$ matrix form of $\boldsymbol{e}_2 \in \mathbb{R}^{p \times p}$. If the covariance matrix is diagonal, the second set has p neurons rather than $(p \times p)$ neurons. In this case, we have $\boldsymbol{\Sigma}_{z|x} = \textbf{diag}(\boldsymbol{e}_2) \in \mathbb{R}^{d \times d}$. Any distribution with any number of parameters can be chosen for $\mathbb{P}(\boldsymbol{z}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ but the multivariate Gaussian with diagonal covariance is very well-used:

$$q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}_e) = \mathcal{N}(\boldsymbol{z}_i \mid \boldsymbol{\mu}_{z|x}, \boldsymbol{\Sigma}_{z|x}). \tag{72}$$

Let the network weights for the output sets of encoder, $\{\boldsymbol{e}_j\}_{j=1}^m$, be denoted by $\{\boldsymbol{\theta}_{e,j}\}_{j=1}^m$. As the input of encoder is $\boldsymbol{x}_i$, the j-th output set of encoder can be written as $\boldsymbol{e}_j(\boldsymbol{x}_i, \boldsymbol{\theta}_{e,j})$. In the case of multivariate Gaussian distribution for the latent space, the parameters are $\boldsymbol{\mu}_{z|x} = \boldsymbol{e}_1(\boldsymbol{x}_i, \boldsymbol{\theta}_{e,1})$ and $\boldsymbol{\Sigma}_{z|x} = \textbf{diag}(\boldsymbol{e}_2(\boldsymbol{x}_i, \boldsymbol{\theta}_{e,2}))$.

5.1.2. Sampling the Latent Variable

When the data point $\boldsymbol{x}_i$ is fed as input to the encoder, the parameters of the conditional distribution $q(\boldsymbol{z}_i)$ are obtained; hence, the distribution of latent space, which is $q(\boldsymbol{z}_i)$, is determined corresponding to the data point $\boldsymbol{x}_i$. Now, in the latent space, we sample the corresponding latent variable from the distribution of latent space:

$$\boldsymbol{z}_i \sim q(\boldsymbol{z}_i) = \mathbb{P}(\boldsymbol{z}_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}_e). \tag{73}$$

This latent variable is fed as input to the decoder which is explained in the following.

5.1.3. Decoder of Variational Autoencoder

As Fig. 4 shows, the decoder of VAE models the conditional distribution $\mathbb{P}(\boldsymbol{x}_i \mid \boldsymbol{z}_i, \boldsymbol{\theta}_d)$ where $\boldsymbol{\theta}_d$ are the weights of decoder layers in VAE. The input and output of decoder are $\boldsymbol{z}_i \in \mathbb{R}^p$ and $\boldsymbol{x}_i \in \mathbb{R}^d$, respectively. The output neurons of decoder are supposed to either generate the reconstructed data point or determine the parameters of the conditional distribution $\mathbb{P}(\boldsymbol{x}_i \mid \boldsymbol{z}_i, \boldsymbol{\theta}_d)$; the former is more common. In the latter case, if this conditional distribution has l number of parameters, we have l sets of output neurons from the decoder, denoted by $\{\boldsymbol{d}_j\}_{j=1}^l$. The dimensionality of these sets may defer depending the size of every parameters. The example of multivariate Gaussian distribution also can be mentioned for the decoder. Let the network

weights for the output sets of decoder, $\{\boldsymbol{d}_j\}_{j=1}^{l}$, be denoted by $\{\boldsymbol{\theta}_{d,j}\}_{j=1}^{l}$. As the input of decoder is $\boldsymbol{z}_i$, the j-th output set of decoder can be written as $\boldsymbol{d}_j(\boldsymbol{z}_i, \boldsymbol{\theta}_{d,j})$.

5.2. Training Variational Autoencoder with Expectation Maximization

We use EM for training the VAE. Recall Eqs. (8) and (9) for EM in variational inference. Inspired by that, VAE uses EM for training where the ELBO is a function of encoder weights $\boldsymbol{\theta}_e$, decoder weights $\boldsymbol{\theta}_d$, and data point $\boldsymbol{x}_i$:

$$\text{E-step:} \quad \boldsymbol{\theta}_e^{(t)} := \arg\max_{q} \quad \mathcal{L}(\boldsymbol{\theta}_e, \boldsymbol{\theta}_d^{(t-1)}, \boldsymbol{x}_i), \tag{74}$$

$$\text{M-step:} \quad \boldsymbol{\theta}_d^{(t)} := \arg\max_{q} \quad \mathcal{L}(\boldsymbol{\theta}_e^{(t)}, \boldsymbol{\theta}_d, \boldsymbol{x}_i). \tag{75}$$

We can simplify this iterative optimization algorithm by alternating optimization (Jain & Kar, 2017) where we take a step of gradient ascent optimization in every iteration. We consider mini-batch stochastic gradient ascent and take training data in batches where b denotes the mini-batch size. Hence, the optimization is:

$$\text{E-step:} \quad \boldsymbol{\theta}_e^{(t)} := \boldsymbol{\theta}_e^{(t-1)} + \eta_e \frac{\partial \sum_{i=1}^{b} \mathcal{L}(\boldsymbol{\theta}_e, \boldsymbol{\theta}_d^{(t-1)}, \boldsymbol{x}_i)}{\partial \boldsymbol{\theta}_e}, \tag{76}$$

$$\text{M-step:} \quad \boldsymbol{\theta}_d^{(t)} := \boldsymbol{\theta}_d^{(t-1)} + \eta_d \frac{\partial \sum_{i=1}^{b} \mathcal{L}(\boldsymbol{\theta}_e^{(t)}, \boldsymbol{\theta}_d, \boldsymbol{x}_i)}{\partial \boldsymbol{\theta}_d}, \tag{77}$$

where η_e and η_d are the learning rates for $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_d$, respectively.

The ELBO is simplified as:

$$\begin{aligned}\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) &\overset{(4)}{=} -\sum_{i=1}^{b} \mathrm{KL}\big(q(\boldsymbol{z}_i) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)\big) \\ &\overset{(12)}{=} -\sum_{i=1}^{b} \mathrm{KL}\big(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)\big).\end{aligned} \tag{78}$$

Note that the parameter of $\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)$ is $\boldsymbol{\theta}_d$ because $\boldsymbol{z}_i$ is generated after the encoder and before the decoder.

There are different ways for approximating the KL divergence in Eq. (78) (Hershey & Olsen, 2007; Duchi, 2007). We can simplify the ELBO in at least two different ways which are explained in the following.

5.2.1. Simplification Type 1

We continue the simplification of ELBO:

$$\begin{aligned}\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) &= -\sum_{i=1}^{b} \mathrm{KL}\big(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)\big) \\ &= -\sum_{i=1}^{b} \mathbb{E}_{\sim q^{(t-1)}(\boldsymbol{z}_i)}\Big[\log\big(\frac{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)}\big)\Big] \\ &= -\sum_{i=1}^{b} \mathbb{E}_{\sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}\Big[\log\big(\frac{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)}\big)\Big].\end{aligned} \tag{79}$$

This expectation can be approximated using Monte Carlo approximation (Ghojogh et al., 2020b) where we draw ℓ samples $\{\boldsymbol{z}_{i,j}\}_{j=1}^{\ell}$, corresponding to the i-th data point, from the conditional distribution distribution as:

$$\boldsymbol{z}_{i,j} \sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e), \quad \forall j \in \{1, \dots, \ell\}. \tag{80}$$

Monte Carlo approximation (Ghojogh et al., 2020b), in general, approximates expectation as:

$$\mathbb{E}_{\sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}\big[f(\boldsymbol{z}_i)\big] \approx \frac{1}{\ell} \sum_{j=1}^{\ell} f(\boldsymbol{z}_{i,j}), \tag{81}$$

where $f(\boldsymbol{z}_i)$ is a function of $\boldsymbol{z}_i$. Here, the approximation is:

$$\begin{aligned}\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) &\approx \sum_{i=1}^{b} \widetilde{\mathcal{L}}(q, \boldsymbol{\theta}) \\ &= -\sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \log\big(\frac{\mathbb{P}(\boldsymbol{z}_{i,j} \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_{i,j} \,|\, \boldsymbol{\theta}_d)}\big) \\ &= \sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \Big[\log\big(\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_{i,j} \,|\, \boldsymbol{\theta}_d)\big) \\ &\qquad - \log\big(\mathbb{P}(\boldsymbol{z}_{i,j} \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)\big)\Big].\end{aligned} \tag{82}$$

5.2.2. Simplification Type 2

We can simplify the ELBO using another approach:

$$\begin{aligned}\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) &= -\sum_{i=1}^{b} \mathrm{KL}\big(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \,\|\, \mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)\big) \\ &= -\sum_{i=1}^{b} \int \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \log\big(\frac{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{x}_i, \boldsymbol{z}_i \,|\, \boldsymbol{\theta}_d)}\big)\, d\boldsymbol{z}_i \\ &= -\sum_{i=1}^{b} \int \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \log\big(\frac{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta}_d)\, \mathbb{P}(\boldsymbol{z}_i)}\big)\, d\boldsymbol{z}_i\end{aligned}$$

$$= -\sum_{i=1}^{b} \int \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \log\big(\frac{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{z}_i)}\big)\, d\boldsymbol{z}_i + \sum_{i=1}^{b} \int \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \log\big(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta}_d)\big)\, d\boldsymbol{z}_i$$
$$= -\sum_{i=1}^{b} \mathrm{KL}\big(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \,\|\, \mathbb{P}(\boldsymbol{z}_i)\big) + \sum_{i=1}^{b} \mathbb{E}_{\sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}\Big[\log\big(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_i, \boldsymbol{\theta}_d)\big)\Big]. \quad (83)$$

The second term in the above equation can be estimated using Monte Carlo approximation (Ghojogh et al., 2020b) where we draw ℓ samples $\{\boldsymbol{z}_{i,j}\}_{j=1}^{\ell}$ from $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$:

$$\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) \approx \sum_{i=1}^{b} \widetilde{\mathcal{L}}(q, \boldsymbol{\theta}) = -\sum_{i=1}^{b} \mathrm{KL}\big(\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \,\|\, \mathbb{P}(\boldsymbol{z}_i)\big) + \sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \log\big(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_{i,j}, \boldsymbol{\theta}_d)\big). \quad (84)$$

The first term in the above equation can be converted to expectation and then computed using Monte Monte Carlo approximation (Ghojogh et al., 2020b) again, where we draw ℓ samples $\{\boldsymbol{z}_{i,j}\}_{j=1}^{\ell}$ from $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$:

$$\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) \approx \sum_{i=1}^{b} \widetilde{\mathcal{L}}(q, \boldsymbol{\theta})$$
$$= -\sum_{i=1}^{b} \mathbb{E}_{\sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}\Big[\log\big(\frac{\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}{\mathbb{P}(\boldsymbol{z}_i)}\big)\Big] + \sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \log\big(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_{i,j}, \boldsymbol{\theta}_d)\big)$$
$$\approx -\sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \log\big(\mathbb{P}(\boldsymbol{z}_{i,j} \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)\big) - \log\big(\mathbb{P}(\boldsymbol{z}_{i,j})\big) + \sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \log\big(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_{i,j}, \boldsymbol{\theta}_d)\big). \quad (85)$$

In case we have some families of distributions, such as Gaussian distributions, for $\mathbb{P}(\boldsymbol{z}_{i,j} \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ and $\mathbb{P}(\boldsymbol{z}_{i,j})$, the first term in Eq. (84) can be computed analytically. In the following, we simply Eq. (84) further for Gaussian distributions.

5.2.3. Simplification Type 2 for Special Case of Gaussian Distributions

We can compute the KL divergence in the first term of Eq. (84) analytically for univariate or multivariate Gaussian distributions. For this, we need two following lemmas.

Lemma 1. *The KL divergence between two univariate Gaussian distributions $p_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $p_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ is:*

$$KL(p_1 \| p_2) = \log(\frac{\sigma_2}{\sigma_1}) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}. \quad (86)$$

Proof. See Appendix A for proof. □

Lemma 2. *The KL divergence between two multivariate Gaussian distributions $p_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $p_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ with dimensionality p is:*

$$KL(p_1 \| p_2) = \frac{1}{2}\Big(\log(\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}) - p + \boldsymbol{tr}(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\Big). \quad (87)$$

Proof. See (Duchi, 2007, Section 9) for proof. □

Consider the case in which we have:

$$\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e) \sim \mathcal{N}(\boldsymbol{\mu}_{z|x}, \boldsymbol{\Sigma}_{z|x}), \quad (88)$$
$$\mathbb{P}(\boldsymbol{z}_i) \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z), \quad (89)$$

where $\boldsymbol{z}_i \in \mathbb{R}^p$. Note that the parameters $\boldsymbol{\mu}_{z|x}$ and $\boldsymbol{\Sigma}_{z|x}$ are trained in neural network while the parameters $\mathbb{P}(\boldsymbol{z}_{i,j})$ can be set to $\boldsymbol{\mu}_z = \boldsymbol{0}$ and $\boldsymbol{\Sigma}_z = \boldsymbol{I}$ inspired by Eq. (29) in factor analysis. According to Lemma 2, the approximation of ELBO, i.e. Eq. (84), can be simplified to:

$$\sum_{i=1}^{b} \mathcal{L}(q, \boldsymbol{\theta}) \approx \sum_{i=1}^{b} \widetilde{\mathcal{L}}(q, \boldsymbol{\theta}) = -\sum_{i=1}^{b} \frac{1}{2}\Big(\log(\frac{|\boldsymbol{\Sigma}_z|}{|\boldsymbol{\Sigma}_{z|x}|}) - p + \mathbf{tr}(\boldsymbol{\Sigma}_z^{-1}\boldsymbol{\Sigma}_{z|x}) + (\boldsymbol{\mu}_z - \boldsymbol{\mu}_{z|x})^\top \boldsymbol{\Sigma}_z^{-1}(\boldsymbol{\mu}_z - \boldsymbol{\mu}_{z|x})\Big) + \sum_{i=1}^{b} \frac{1}{\ell} \sum_{j=1}^{\ell} \log\big(\mathbb{P}(\boldsymbol{x}_i \,|\, \boldsymbol{z}_{i,j}, \boldsymbol{\theta}_d)\big). \quad (90)$$

5.2.4. Training Variational Autoencoder with Approximations

We can train VAE with EM, where Monte Carlo approximations are applied to ELBO. The Eqs. (76) and (77) are replaced by the following equations:

$$\text{E-step:} \quad \boldsymbol{\theta}_e^{(t)} := \boldsymbol{\theta}_e^{(t-1)} + \eta_e \frac{\partial \sum_{i=1}^{b} \widetilde{\mathcal{L}}(\boldsymbol{\theta}_e, \boldsymbol{\theta}_d^{(t-1)}, \boldsymbol{x}_i)}{\partial \boldsymbol{\theta}_e}, \quad (91)$$

$$\text{M-step:} \quad \boldsymbol{\theta}_d^{(t)} := \boldsymbol{\theta}_d^{(t-1)} + \eta_d \frac{\partial \sum_{i=1}^{b} \widetilde{\mathcal{L}}(\boldsymbol{\theta}_e^{(t)}, \boldsymbol{\theta}_d, \boldsymbol{x}_i)}{\partial \boldsymbol{\theta}_d}, \quad (92)$$

where the approximated ELBO was introduced in previous sections.

5.2.5. Prior Regularization

Some works regularize the ELBO, Eq. (78), with a penalty on the prior distribution $\mathbb{P}(\boldsymbol{z}_i)$. Using this, we guide the learned distribution of latent space $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ to have a specific prior distribution $\mathbb{P}(\boldsymbol{z}_i)$. Some examples for prior regularization in VAE are geodesic priors (Hadjeres et al., 2017) and optimal priors (Takahashi et al., 2019). Note that this regularization can inject domain knowledge to the latent space. It can also be useful for making the latent space more interpretable.

5.3. The Reparametrization Trick

Sampling the ℓ samples for the latent variables, i.e. Eq. (73), blocks the gradient flow because computing the derivatives through $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ by chain rule gives a high variance estimate of gradient. In order to overcome this problem, we use the reparameterization technique (Kingma & Welling, 2014; Rezende et al., 2014; Titsias & Lázaro-Gredilla, 2014). In this technique, instead of sampling $\boldsymbol{z}_i \sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$, we assume $\boldsymbol{z}_i$ is a random variable but is a deterministic function of another random variable $\boldsymbol{\epsilon}_i$ as follows:

$$\boldsymbol{z}_i = g(\boldsymbol{\epsilon}_i, \boldsymbol{x}_i, \boldsymbol{\theta}_e), \tag{93}$$

where $\boldsymbol{\epsilon}_i$ is a stochastic variable sampled from a distribution as:

$$\boldsymbol{\epsilon}_i \sim \mathbb{P}(\boldsymbol{\epsilon}). \tag{94}$$

The Eqs. (79) and 83 both contain an expectation of a function $f(\boldsymbol{z}_i)$. Using this technique, this expectation is replaced as:

$$\mathbb{E}_{\sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}[f(\boldsymbol{z}_i)] \rightarrow \mathbb{E}_{\sim \mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)}[f(g(\boldsymbol{\epsilon}_i, \boldsymbol{x}_i, \boldsymbol{\theta}_e))]. \tag{95}$$

Using the reparameterization technique, the encoder, which implemented $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$, is replaced by $g(\boldsymbol{\epsilon}_i, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$ where in the latent space between encoder and decoder, we have $\boldsymbol{\epsilon}_i \sim \mathbb{P}(\boldsymbol{\epsilon})$ and $\boldsymbol{z}_i = g(\boldsymbol{\epsilon}_i, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$.

A simple example for the reparameterization technique is when z_i and ϵ_i are univariate Gaussian variables:

$$\begin{aligned} &z_i \sim \mathcal{N}(\mu, \sigma^2), \\ &\epsilon_i \sim \mathcal{N}(0, 1), \\ &z_i = g(\epsilon_i) = \mu + \sigma \epsilon_i. \end{aligned}$$

For some more advanced reparameterization techniques, the reader can refer to (Figurnov et al., 2018).

5.4. Training Variational Autoencoder with Backpropagation

In practice, VAE is trained by backpropagation (Rezende et al., 2014) where the backpropagation algorithm (Rumelhart et al., 1986) is used for training the weights of network. Recall that in training VAE with EM, the encoder and decoder are trained separately using the E-step and the M-step of EM, respectively. However, in training VAE with backpropagation, the whole network is trained together and not in separate steps. Suppose the whole weights if VAE are denoted by $\boldsymbol{\theta} := \{\boldsymbol{\theta}_e, \boldsymbol{\theta}_d\}$. Backpropagation trains VAE using the mini-batch stochastic gradient descent with the negative ELBO, $\sum_{i=1}^{b} -\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{x}_i)$, as the loss function:

$$\boldsymbol{\theta}^{(t)} := \boldsymbol{\theta}^{(t-1)} - \eta \frac{\partial \sum_{i=1}^{b} -\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{x}_i)}{\partial \boldsymbol{\theta}}, \tag{96}$$

where η is the learning rate. Note that we are minimizing here because neural networks usually minimize the loss function.

5.5. The Test Phase in Variational Autoencoder

In the test phase, we feed the test data point $\boldsymbol{x}_i$ to the encoder to determine the parameters of the conditional distribution of latent space, i.e., $\mathbb{P}(\boldsymbol{z}_i \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_e)$. Then, from this distribution, we sample the latent variable $\boldsymbol{z}_i$ from the latent space and generate the corresponding reconstructed data point $\boldsymbol{x}_i$ by the decoder. As you see, VAE is a generative model which generates data points (Ng & Jordan, 2002).

5.6. Other Notes and Other Variants of Variational Autoencoder

There exist many improvements on VAE. Here, we briefly review some of these works. One of the problems of VAE is generating blurry images when data points are images. This blurry artifact may be because of several following reasons:

- sampling for the Monte Carlo approximations
- lower bound approximation by ELBO
- restrictions on the family of distributions where usually simple Gaussin distributions are used.

There are some other interpretations for the reason of this problem; for example, see (Zhao et al., 2017). This work also proposed a generalized ELBO. Note that generative adversarial networks (Goodfellow et al., 2014) usually generate more clear images; therefore, some works have combined variational and adversarial inferences (Mescheder et al., 2017) for using the advantages of both models.

Variational discriminant analysis (Yu et al., 2020) has also been proposed for classification and discrimination of classes. Two other tutorials on VAE are (Doersch, 2016) and (Odaibo, 2019). Some more recently published papers on VAE are nearly optimal VAE (Bai et al., 2020), deep VAE (Hou et al., 2017), Hamiltonian VAE (Caterini et al., 2018), and Nouveau VAE (Vahdat & Kautz, 2020) which

is a hierarchical VAE. For image data and image and caption modeling, a fusion of VAE and convolutional neural network is also proposed (Pu et al., 2016). The influential factors in VAE are also analyzed in the paper (Liu et al., 2020).

6. Conclusion

This paper was a tutorial and survey on several dimensionality reduction and generative model which are tightly related. Factor analysis, probabilistic PCA, variational inference, and variational autoencoder are covered in this paper. All of these methods assume that every data point is generated from a latent variable or factor where some noise have also been applied on data in the data space.

Acknowledgement

The authors hugely thank the instructors of deep learning course at the Carnegie Mellon University (you can see their YouTube channel) whose lectures partly covered some materials mentioned in this tutorial paper.

A. Proof for Lemma 1

$$\begin{aligned}\mathrm{KL}(p_1\|p_2) &= \int p_1(x)\log\big(\frac{p_1(x)}{p_2(x)}\big)dx \\ &= \int p_1(x)\log(p_1(x))\,dx - \int p_1(x)\log(p_2(x))\,dx.\end{aligned}$$

According to integration by parts, we have:

$$\int p_1(x)\log(p_1(x))\,dx = -\frac{1}{2}(1+\log(2\pi\sigma_1^2)).$$

We also have:

$$\begin{aligned}&-\int p_1(x)\log(p_2(x))\,dx \\ &= -\int p_1(x)\log\Big(\frac{1}{\sqrt{2\pi\sigma_2^2}}e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}\Big)\,dx \\ &= \frac{1}{2}\log(2\pi\sigma_2^2)\underbrace{\int p_1(x)dx}_{=1} - \int p_1(x)\log\big(e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}\big)\,dx \\ &= \frac{1}{2}\log(2\pi\sigma_2^2) + \int p_1(x)\frac{(x-\mu_2)^2}{2\sigma_2^2}\,dx \\ &= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{1}{2\sigma_2^2}\Big(\int p_1(x)x^2dx \\ &\qquad - \int p_1(x)2x\mu_2dx + \int p_1(x)\mu_2^2dx\Big) \\ &= \frac{1}{2}\log(2\pi\sigma_2^2) \\ &\qquad + \frac{1}{2\sigma_2^2}\Big(\mathbb{E}_{\sim p_1(x)}[x^2] - 2\mu_2\mathbb{E}_{\sim p_1(x)}[x] + \mu_2^2\Big).\end{aligned}$$

We know that:

$$\mathbb{V}\mathrm{ar}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 \implies \mathbb{E}[x^2] = \sigma_1^2 + \mu_1^2$$

Hence:

$$\begin{aligned}&-\int p_1(x)\log(p_2(x))\,dx \\ &= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{1}{2\sigma_2^2}\Big(\sigma_1^2+\mu_1^2-2\mu_2\mu_1+\mu_2^2\Big) \\ &= \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{1}{2\sigma_2^2}\big(\sigma_1^2+(\mu_1-\mu_2)\big).\end{aligned}$$

Therefore, finally, we have:

$$\begin{aligned}\mathrm{KL}(p_1\|p_2) &= -\frac{1}{2}(1+\log(2\pi\sigma_1^2)) \\ &\quad + \frac{1}{2}\log(2\pi\sigma_2^2) + \frac{1}{2\sigma_2^2}\big(\sigma_1^2+(\mu_1-\mu_2)\big) \\ &= \log(\frac{\sigma_2}{\sigma_1}) + \frac{\sigma_1^2+(\mu_1-\mu_2)^2}{2\sigma_2^2} - \frac{1}{2}.\end{aligned}$$

Q.E.D.

References

Bai, Jincheng, Song, Qifan, and Cheng, Guang. Nearly optimal variational inference for high dimensional regression with shrinkage priors. *arXiv preprint arXiv:2010.12887*, 2020.

Bishop, Christopher M. Bayesian PCA. In *Advances in neural information processing systems*, pp. 382–388, 1999.

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

Bouchard, Guillaume and Triggs, Bill. The tradeoff between generative and discriminative classifiers. In *16th IASC International Symposium on Computational Statistics*, 2004.

Caterini, Anthony L, Doucet, Arnaud, and Sejdinovic, Dino. Hamiltonian variational auto-encoder. *Advances in Neural Information Processing Systems*, 31:8167–8177, 2018.

Cattell, Raymond B. A biometrics invited paper. factor analysis: An introduction to essentials i. the purpose and underlying models. *Biometrics*, 21(1):190–215, 1965.

Child, Dennis. *The essentials of factor analysis*. Cassell Educational, 1990.

Doersch, Carl. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

Duchi, John. Derivations for linear algebra and optimization. Technical report, Berkeley, California, 2007.

Figurnov, Mikhail, Mohamed, Shakir, and Mnih, Andriy. Implicit reparameterization gradients. *Advances in Neural Information Processing Systems*, 31:441–452, 2018.

Fruchter, Benjamin. *Introduction to factor analysis*. Van Nostrand, 1954.

Ghahramani, Zoubin and Hinton, Geoffrey E. The EM algorithm for mixtures of factor analyzers. Technical report, Technical Report CRG-TR-96-1, University of Toronto, 1996.

Ghojogh, Benyamin and Crowley, Mark. Unsupervised and supervised principal component analysis: Tutorial. *arXiv preprint arXiv:1906.03148*, 2019.

Ghojogh, Benyamin, Ghojogh, Aydin, Crowley, Mark, and Karray, Fakhri. Fitting a mixture distribution to data: tutorial. *arXiv preprint arXiv:1901.06708*, 2019a.

Ghojogh, Benyamin, Karray, Fakhri, and Crowley, Mark. Eigenvalue and generalized eigenvalue problems: Tutorial. *arXiv preprint arXiv:1903.11240*, 2019b.

Ghojogh, Benyamin, Ghodsi, Ali, Karray, Fakhri, and Crowley, Mark. Stochastic neighbor embedding with Gaussian and Student-t distributions: Tutorial and survey. *arXiv preprint arXiv:2009.10301*, 2020a.

Ghojogh, Benyamin, Nekoei, Hadi, Ghojogh, Aydin, Karray, Fakhri, and Crowley, Mark. Sampling algorithms, from survey sampling to Monte Carlo methods: Tutorial and literature review. *arXiv preprint arXiv:2011.00901*, 2020b.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gu, Mengyang and Shen, Weining. Generalized probabilistic principal component analysis of correlated data. *Journal of Machine Learning Research*, 21(13):1–41, 2020.

Guan, Yue and Dy, Jennifer. Sparse probabilistic principal component analysis. In *Artificial Intelligence and Statistics*, pp. 185–192, 2009.

Hadjeres, Gaëtan, Nielsen, Frank, and Pachet, François. GLSR-VAE: Geodesic latent space regularization for variational autoencoder architectures. In *2017 IEEE Symposium Series on Computational Intelligence*, pp. 1–7. IEEE, 2017.

Harman, Harry H. *Modern factor analysis*. University of Chicago press, 1976.

Hauskrecht, Milos. CS3750 lecture notes for probabilistic principal component analysis and the E-M algorithm. Technical report, University of Pittsburgh, 2007.

Hershey, John R and Olsen, Peder A. Approximating the Kullback Leibler divergence between Gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pp. IV–317. IEEE, 2007.

Hou, Xianxu, Shen, Linlin, Sun, Ke, and Qiu, Guoping. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision*, pp. 1133–1141. IEEE, 2017.

Jain, Prateek and Kar, Purushottam. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.

Jolliffe, Ian T and Cadima, Jorge. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065), 2016.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kullback, Solomon and Leibler, Richard A. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Liu, Shiqi, Liu, Jingxin, Zhao, Qian, Cao, Xiangyong, Li, Huibin, Meng, Deyu, Meng, Hongying, and Liu, Sheng. Discovering influential factors in variational autoencoders. *Pattern Recognition*, 100:107166, 2020.

Mattei, Pierre-Alexandre, Bouveyron, Charles, and Latouche, Pierre. Globally sparse probabilistic pca. In *Artificial Intelligence and Statistics*, pp. 976–984, 2016.

Mescheder, Lars, Nowozin, Sebastian, and Geiger, Andreas. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning*, 2017.

Ng, Andrew. CS229 lecture notes for factor analysis. Technical report, Stanford University, 2018.

Ng, Andrew Y and Jordan, Michael I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in neural information processing systems*, pp. 841–848, 2002.

Odaibo, Stephen. Tutorial: Deriving the standard variational autoencoder (VAE) loss function. *arXiv preprint arXiv:1907.08956*, 2019.

Paola Garcia, Leibny. Lecture notes for factor analysis. Technical report, Carnegie Mellon University, 2018.

Pu, Yunchen, Gan, Zhe, Henao, Ricardo, Yuan, Xin, Li, Chunyuan, Stevens, Andrew, and Carin, Lawrence. Variational autoencoder for deep learning of images, labels and captions. In *Advances in neural information processing systems*, pp. 2352–2360, 2016.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Roweis, Sam. EM algorithms for PCA and SPCA. *Advances in neural information processing systems*, 10: 626–632, 1997.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Sminchisescu, Cristian and Jepson, Allan. Generative modeling for continuous non-linearly embedded visual inference. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 96, 2004.

Takahashi, Hiroshi, Iwata, Tomoharu, Yamanaka, Yuki, Yamada, Masanori, and Yagi, Satoshi. Variational autoencoder with implicit optimal priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5066–5073, 2019.

Tipping, Michael E and Bishop, Christopher M. Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482, 1999a.

Tipping, Michael E and Bishop, Christopher M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999b.

Titsias, Michalis and Lázaro-Gredilla, Miguel. Doubly stochastic variational Bayes for non-conjugate inference. In *International conference on machine learning*, pp. 1971–1979, 2014.

Vahdat, Arash and Kautz, Jan. NVAE: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020.

Walker, Jacob, Doersch, Carl, Gupta, Abhinav, and Hebert, Martial. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pp. 835–851. Springer, 2016.

Yu, Shipeng, Yu, Kai, Tresp, Volker, Kriegel, Hans-Peter, and Wu, Mingrui. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 464–473, 2006.

Yu, Weichang, Azizi, Lamiae, and Ormerod, John T. Variational nonparametric discriminant analysis. *Computational Statistics & Data Analysis*, 142:106817, 2020.

Zhao, J and Jiang, Q. Probabilistic PCA for t distributions. *Neurocomputing*, 69(16-18):2217–2226, 2006.

Zhao, Shengjia, Song, Jiaming, and Ermon, Stefano. Towards deeper understanding of variational autoencoding models. *arXiv preprint arXiv:1702.08658*, 2017.