Latent Variable Models

Rafael BENATTI

December 7, 2023

I Introduction

These class notes focus on latent variables for generative problems, although they may appear in any other applications.
Latent variable models offer a potent framework for uncovering concealed patterns and intrinsic structures within observed data. Within this paradigm, a latent variable, denoted as z_i, encapsulates information not directly discernible in a given example x_i. Visualize z_i as a hidden feature or characteristic associated with each data point, representing aspects of the data that may elude explicit measurement or recording. These latent variables serve as a form of missing information, contributing to a more nuanced representation of the complexities inherent in real-world datasets.

Example: Clustering Setting) For instance, in a clustering algorithm, x_i could represent a feature vector describing a data point, and z_i might indicate the cluster to which the data point belongs. The latent variable z_i captures the inherent, unobservable structure that governs the cluster assignment of x_i. It could encapsulate information about the data point's inherent properties, preferences, or characteristics that lead it to belong to a particular cluster. As seen in the image the cluster can be understood as a latent variable that is missing in the data points.

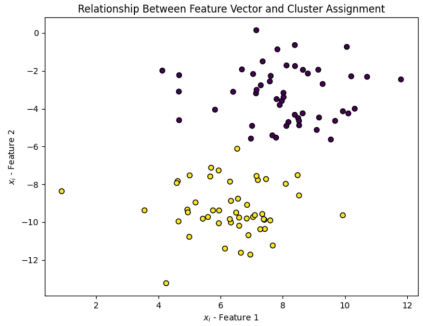


Figure 1: Example of clustering as latent variable

Example: Dimensionality reduction) Latent variables can also be used in the context of dimensionality reduction, capturing the essential information embedded within high-dimensional data and representing it in a lower-dimensional space. Consider a scenario where we have a dataset with numerous features, each contributing to the overall complexity of the data. Latent variables, denoted as z_i, act as unobservable factors that summarize the inherent patterns, dependencies, and variations within the original data points x_i.

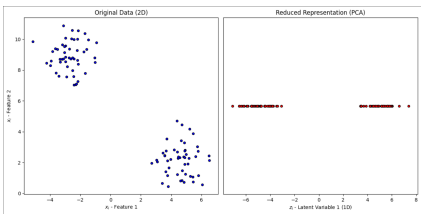


Figure 2: Use of latent variables for dimentsionality reduction

Usually, the best learning algorithm to estimate latent variables is the ExpectationMaximisation (EM).

I.1 Probabilistic Machine Learning

Probabilistic Machine Learning (ML) is the most appropriate framework for handling latent variables in data analysis. In this approach, it is assumed that the observed data, or both the observed data and latent variables, are generated according to a probabilistic generative model with certain unknown parameters. Unlike traditional machine learning methods that may treat data as deterministic and fixed, probabilistic ML acknowledges the inherent uncertainty in real-world phenomena. By incorporating probabilistic models, practitioners can capture the inherent variability and hidden structures within the data, often attributed to latent variables. This framework allows for a more nuanced understanding of complex relationships, offering a probabilistic interpretation of observed phenomena and enabling principled methods for uncertainty quantification. Learning a model in this framework is done by Maximum Likelihood Estimation (MLE) or Maximum A Posteriori estimation (MAP).
In the context of Probabilistic Machine Learning, consider a generative model for data points on a curve described by $f : [0, 1] \to \mathbb{R}^2$. Each data point X_i is generated using a latent variable Z_i sampled from a uniform distribution on $[0, 1]$, and a noise term E_i sampled from a normal distribution with mean 0 and standard deviation 1. The observed data point is then given by $X_i = f(Z_i) + E_i$. The conditional probability $P(X|Z)$ is modeled as a normal distribution with mean $f(Z)$ and identity covariance matrix. The marginal probability $P(X)$ is obtained by integrating the product of the conditional probability and the prior probability $P(Z)$ over all possible values of Z. The posterior probability $P(Z|X)$ is calculated using Bayes' theorem. This probabilistic framework allows for uncertainty modeling in the generative process.

$$X_i = f(Z_i) + E_i \tag{1}$$

$$P(X|Z) = \mathcal{N}(f(Z), I) \tag{2}$$

$$P(X) = \int P(X|Z)P(Z)\, dZ \tag{3}$$

$$P(Z|X) = \frac{P(X|Z)P(Z)}{P(X)} \tag{4}$$

The generative model may include unobserved variables. These are called *Latent Variables*.

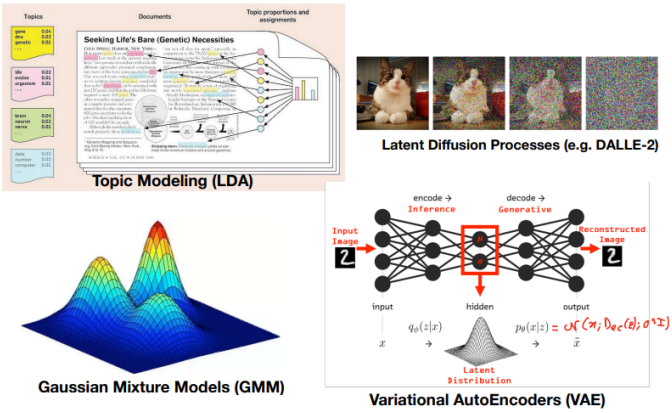


Figure 3: Examples of generative models with *Latent Variables*

II Gaussian mixtures and the EM algorithm

II.1 Univariate Gaussian

The univariate Gaussian (or Single Variate Gaussian) is a probability distribution that characterizes a single random variable. Also known as the normal distribution, it is fully determined by two parameters: the mean (μ), representing the central location of the distribution, and the standard deviation (σ), indicating the spread or dispersion of the data. The probability density function (PDF) of the univariate Gaussian is given by the formula $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. This distribution is symmetric and bell-shaped, with the majority of values concentrated around the mean. The Single Variate Gaussian is widely used in various fields due to

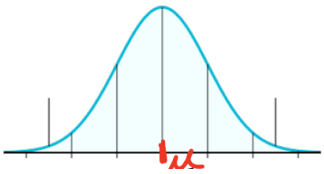


Figure 4: Univariate Gaussian

its mathematical convenience and its prevalence in natural phenomena, making it a foundational concept in probability and statistics.

II.2 Multivariate Normal Distribution

In the realm of probability and statistics, the Multivariate Gaussian distribution extends the concept of the Single Variate Gaussian to multiple dimensions. In its general form, the probability density function (PDF) for the Multivariate Gaussian is given by:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

where $\mathbf{x}$ is a vector of random variables, $\boldsymbol{\mu}$ is the mean vector, Σ is the covariance matrix, and k is the dimensionality of the distribution.
When the covariance matrix is chosen to be the identity matrix ($\Sigma = I$), the Multivariate Gaussian simplifies, and the PDF becomes:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{k/2}} e^{-\frac{1}{2}\|\mathbf{x}-\boldsymbol{\mu}\|^2}$$

This special case corresponds to uncorrelated variables with unit variance. The covariance matrix being the identity implies that the variables are independent, and the shape of the distribution is spherical.

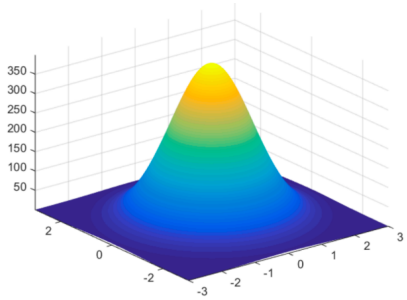
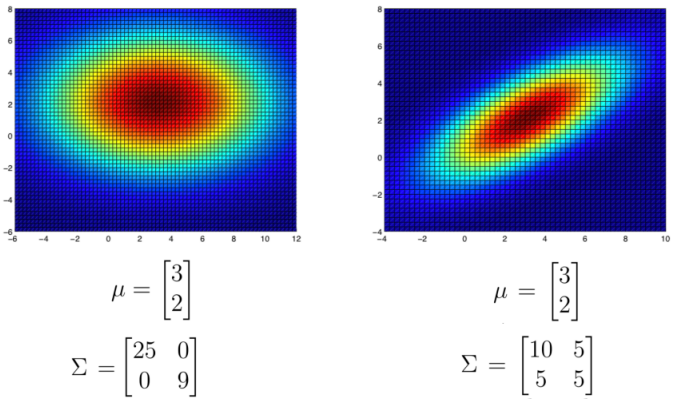


Figure 5: Multivariate Gaussian



$$\mu = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \qquad \mu = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 25 & 0 \\ 0 & 9 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 10 & 5 \\ 5 & 5 \end{bmatrix}$$

Figure 6: Different Multivariate Normal Distributions for different parameters

II.3 Gaussian Mixture Models

One of the simplest latent variable model is the Gaussian Mixture Model (GMM). It is the same as Linear Discriminant Analysis (LDA) without class information. The LDA for 2 classes can be understood as a

generative model with two gaussians $\mathcal{N}(\mu_a, E_a)$, $\mathcal{N}(\mu_b, E_b)$ and $\pi \in [0, 1]$. We have that for each i = 1...N:

$$
\begin{aligned}
Z_i &\sim \text{Ber}(\pi) + 1 \\
X_i &\sim \begin{cases} \text{if } y_i = 1 \text{ then } X_i & \sim \mathcal{N}(\mu_a, E_a) \\ \text{if } y_i = 2 \text{ then } X_i & \sim \mathcal{N}(\mu_b, E_b) \end{cases} \\
\text{Observed data: } & X_1, X_2, \ldots, X_N
\end{aligned}
$$

In the context of Gaussian Mixture Models (GMM), a generative model with two classes is defined using parameters π, μ_a, E_a, μ_b, and E_b. For each data point i ($i = 1$ to N), the class membership is determined by a Bernoulli distribution: $Z_i \sim \text{Ber}(\pi) + 1$. If $Z_i = 1$, the data point is generated from a Gaussian distribution with mean μ_a and covariance matrix E_a; otherwise, if $Z_i = 2$, the data point is generated from a Gaussian distribution with mean μ_b and covariance matrix E_b. The observed data consists of N data points denoted as $X_1, X_2, \ldots, X_N$. The latent variables $Z_1, Z_2, \ldots, Z_N$ representing the true class memberships are unknown. The mathematical formulation is given by:

$$
\begin{aligned}
Z_i &\sim \text{Ber}(\pi) + 1 \\
X_i &\sim \begin{cases} \mathcal{N}(\mu_a, E_a), & \text{if } Z_i = 1 \\ \mathcal{N}(\mu_b, E_b), & \text{if } Z_i = 2 \end{cases} \\
\text{Observed data: } & X_1, X_2, \ldots, X_N
\end{aligned}
$$

Exercise) Consider a scenario with two Gaussian distributions ($K = 2$). Let the parameters for the first Gaussian be $u_1 = 0$ and $\sigma_1 = 1$. The probability of belonging to class 1 is given by $\pi = P(Z_i = 1) = 80\%$. For the second Gaussian, let $u_2 = 1$ and $\sigma_2 = 1$, and accordingly, $P(Z_i = 2) = 1 - \pi$.

1. Draw informally $P(X|Z = 1)$ and $P(X|Z = 2)$ on a graph. These are functions of X.
2. Draw informally $P(X, Z = 1)$ and $P(X, Z = 2)$ on a graph. These are functions of X
3. Draw $P(Z = 1|X)$
4. Draw points "by hand" from this process
5. Compute and simplify $P(Z = 1|X)$

The solution should look like:

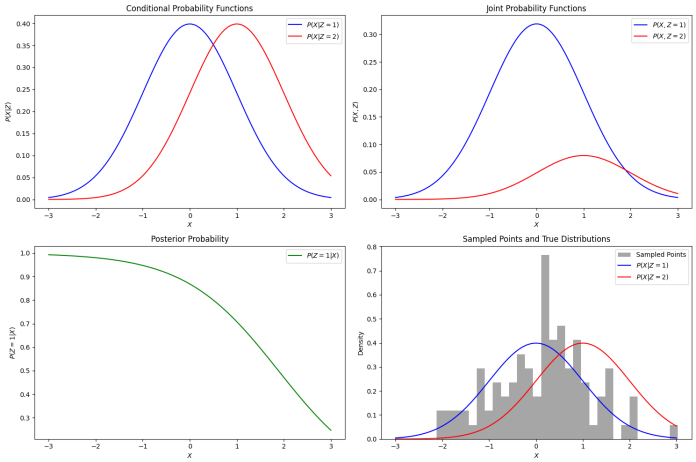


Figure 7: Solution for the given problem

For the last topic we have:

$$
P(Z = 1|X) = \frac{P(X|Z = 1) \cdot \pi}{\pi \cdot P(X|Z = 1) + (1 - \pi) \cdot P(X|Z = 2)}
$$

$$= \frac{\pi \exp\left(-\frac{1}{2}(x-\mu_1)^2\right)}{\pi \exp\left(-\frac{1}{2}(x-\mu_1)^2\right) + (1-\pi)\exp\left(-\frac{1}{2}(x-\mu_2)^2\right)}$$

Then we multiply everything by: $\exp\left(\frac{1}{2}(x-\mu_1)^2\right) \times \frac{1}{\pi}$ Which yields:

$$\frac{1}{1 + \frac{1-\pi}{\pi}\exp\left\{-\frac{1}{2}(x-\mu_2)^2 + \frac{1}{2}(x-\mu_1)^2\right\}}$$

$$= \frac{1}{1 + \exp\left\{x \times (\mu_2 - \mu_1) + \frac{\mu_2^2 - \mu_1^2}{2} + \ln\frac{1-\pi}{\pi}\right\}}$$

Taking $a = (\mu_2 - \mu_1)$ and $b = \frac{\mu_2^2 - \mu_1^2}{2} + \ln\frac{1-\pi}{\pi}$, we finally have:

$$P(Z = 1|X) = \frac{1}{1 + \exp(aX + b)}$$

II.4 Computing the log likelihoof of a GMM model

Taking K gaussians $\mathcal{N}(\mu_1, \Sigma_1) \dots d(\mu_k, \Sigma_k)$, with priors

$$p(z_i = k) = \pi_k \text{ for } k \in \{1 \dots k\}$$
$$\theta = (\mu_1, \Sigma_1, \dots \mu_k, \Sigma_k, \pi_1 \dots \pi_k)$$

$P_\theta(x)$ is the distribution induced by parameters θ. We have a dataset $x_1 \dots x_N \in \mathbb{R}^N$ the log likelihood is

$$\begin{aligned}
\mathcal{LL}(\theta) = \log P_\theta(x_1 \dots x_N) &= \sum_{k=1} \log P_\theta(x_i) \\
&= \sum_{i=1}^{N} \log \sum_{k=1}^{K} P_\theta(x_i \mid z_i = k) P_\theta(z_i = k) \\
&= \sum_{i=1}^{N} \log \sum_{k=1}^{K} N(x_i \mid \mu_k, \Sigma_k) \cdot \pi(k)
\end{aligned}$$

The max likelihood estimate is $\hat{\theta} = \text{argmax}_\theta \mathcal{LL}(\theta)$
Note: for $K = 1$, computing $\hat{\theta}$ is easy because we know how to compute it. for $k > 1$: non convex, often "hard" to compute.
We could learn θ by gradient descent, but much better strategy.

II.5 GMM with K > 1. The Complete Log Likelihood

For $(K > 1)$, maximising the likelihood is hard. To address this challenge, the proposed strategy is to optimize a surrogate known as the Complete Log Likelihood (CLL). The CLL is introduced as an alternative approach, leveraging the assumption that we possess knowledge of the latent variables $z_1 \dots z_N$. However, even with this information, direct computation of the CLL remains a non-trivial task. The CLL is defined as the logarithm of the joint probability $P_\theta(x_1 \dots x_N, z_1 - z_N)$. This shift towards optimizing the CLL aims to tackle the difficulties associated with maximizing the likelihood directly, providing a more feasible and effective route for model optimization.

$$\text{CLL}(\theta, z_1 \dots z_N) = \log P_\theta(x_1 \dots x_N, z_1 - z_N)$$

II.6 GMM with K > 1. The Expected Log Likelihood

Instead of directly optimizing the challenging likelihood function, the ECCL is introduced as a surrogate for optimization. This approach assumes knowledge of latent variables $z_1 \dots z_N$ but acknowledges the inherent computational challenges. To address this, the concept of estimating the distribution $p\left(z_i = k \mid x_i; \hat{\theta}\right)$ for each i is introduced, denoted as $q_i(k)$. The Expected CLL is then computed using these estimates, denoted as $\mathcal{L}(q_i, \theta, i)$. The ECCL ($\alpha(q, \theta)$) is formulated as the expectation of the CLL across all latent variable estimates. The resulting expression involves the sum of logarithmic terms, indicating the likelihood of data points given specific latent variable assignments $\alpha(q, \theta)$ in $\mu_1 \dots \mu_k$ is convex.

$$q_i(k) \approx p_\theta(z_i = k \mid x_i)$$

$$\mathcal{L}(q_i, \theta_i) = \mathbb{E}_{z_{i \sim q_i}} \log P_\theta(x_i, z_i) = \mathbb{Z}_{z_i q_i} \log\left(P_\theta(x_i \mid z_i) \times \pi_i\right)$$

$$= \sum_{k=1}^{k} q_i(k) \log\left(P_\theta(x_i \mid z_i = k) \cdot \pi_k\right)$$

$$ECLL = \delta(q, \theta) = \sum_{i=1}^{N} \alpha(q, \theta, \lambda) = \mathbb{E}_{z_1, \sim q_1, \dots, z_N \sim q_N}\left[CLL(\theta, z_1, \dots z_N)\right]$$

$$= \sum_{i=1}^{N} \sum_{k=1}^{K} q_i(k) \log[P_0(x_i \mid z_i = \ell)\, \pi_k)]$$

II.7 The EM algorithm

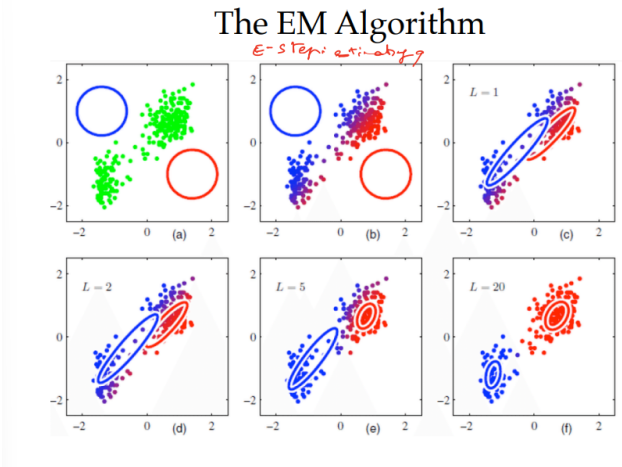
The Expectation-Maximization (EM) algorithm used in scenarios where data involves latent variables or unobservable quantities. The algorithm aims to maximize the likelihood function with respect to the model parameters, even when some variables are hidden or unknown.

1. Initialization: The algorithm begins by initializing the model parameters $\hat{\theta}$ arbitrarily. The choice of initial parameters can influence the convergence and results of the algorithm.

2. E-Step (Expectation Step): In this step, for each data point i and each latent variable k, the algorithm computes the posterior probability $q_i(k)$, representing the probability that z_i takes the value k given the observed data x_i and the current estimate of the parameters $\hat{\theta}$. Depending on the number of classes (K), the calculation involves either the sigmoid function (for $K = 2$) or the softmax function (for $K > 2$).

3. M-Step (Maximization Step): The M-step involves updating the model parameters $\hat{\theta}$ to maximize the expected log-likelihood $\mathcal{L}(q, \theta)$. This maximization step often requires solving optimization problems that vary depending on the specific form of the likelihood function. In the case of Gaussian Mixture Models (GMMs), it typically involves updating the means and variances of the Gaussian distributions.

Details (In 1D): For one-dimensional data, the algorithm's efficacy can be demonstrated by examining the Expected Complete Log Likelihood (ECLL), a convex set that serves as a key concept in the EM algorithm. The update for the mean $\hat{\mu}_k$ is calculated for a fixed value of $\sigma_k = \sigma$, demonstrating the iterative nature of the algorithm in refining parameter estimates.



Variational Analysis of EM: The ELBO (Evidence Lower Bound) - Part 1

In the context of Expectation-Maximization (EM), our primary goal is to optimize the likelihood function $p_\theta(x_1 \dots x_N)$. However, as a practical strategy, we often optimize a surrogate function known as the expected Complete Log Likelihood (ECLL), denoted by $L(q, \theta)$. Understanding the relationship between the likelihood and this surrogate is crucial.

Recall that $L(q, \theta) = \sum_{i=1}^{N} L(q, \theta, i)$, where $L(q, \theta, i)$ represents the expected likelihood for a single data point x_i when latent variables are drawn from the distribution q_i.

Consider a single point x, and let's compute its likelihood $\log P_\theta(x)$. We can express this as the logarithm of the sum over latent variables k, where $q_x(k)$ approximates the probability $p(z = k \mid x = x)$:

$$\log P_\theta(x) = \ln \sum_{k=1}^{k} q_x(k) \times \frac{P_\theta(x, z = k)}{q_x(k)}$$

Here, we invoke Jensen's inequality, stating that if f is a concave function, then $\mathbb{E}(f(x)) \leqslant f(\mathbb{E}(x))$. Applying this to our expression, we obtain:

$$\log P_\theta(x) \geqslant \mathbb{E}_{k \sim q_x} \ln \left[\frac{P_\theta(x, z = k)}{q_x(k)} \right]$$

$$\log p_\theta(x) \geqslant \underbrace{\sum_{k=1}^{k} q_x(k) \ln p_\theta(x, z = k)}_{\alpha(q_x, \theta, x) = ECLL} \underbrace{- \sum_{k=1}^{k} q_x(k) \ln q_x(k)}_{\text{Entropy}}$$

This establishes the Evidence Lower Bound (ELBO), which is a crucial concept in variational inference. The ELBO acts as a lower bound for the log-likelihood and forms the basis for variational EM algorithms.

In the context of Expectation-Maximization (EM), we aim to optimize the likelihood $p(x_1 \dots x_N \mid \theta$, but instead, we optimize a surrogate: the expected Complete Log Likelihood (ECLL) $L(q, \theta)$. Understanding the connection between the likelihood and the ECLL is crucial.

Recall that $L(q, \theta) = \sum^{N} L(q, \theta, i)$, where $L(q, \theta, i)$ represents the expected likelihood for a single data point x_i when latent variables are drawn from the distribution q_i. What is the relationship between the likelihood and the ELBO?

$$\begin{aligned}
&\ln P_\theta(x) - \text{ELBO}\,(x, \theta, q_x) = \ln P_\theta(x) - \mathbb{E}_{k \sim q_x} \left[\ln \left(\frac{f(x, z = k)}{q_x(k)} \right) \right] \\
&= \mathbb{E}_{k \sim q_x} \left[\ln \frac{p_\theta(x) q_x(k)}{p_\theta(x, z = k)} \right] = \mathbb{E} \left[\ln \frac{q_x(k)}{p_\theta^{(z-k|x)}} \right] \\
&= KL\left(q_x(z) \| p_\theta(z \mid x)\right)
\end{aligned}$$

This equation establishes the connection between the likelihood and the Evidence Lower Bound (ELBO). The ELBO acts as a measure of how well our variational distribution q_x approximates the true posterior distribution $p_\theta(z \mid x)$.

The Kullback-Leibler (KL) Divergence, denoted as $KL\left(q \| q'\right)$, possesses several important properties:

$$\begin{aligned}
&KL\left(q \| q'\right) = \mathop{\mathbb{E}}_{k \sim q} \left[\ln \left[\frac{q(x)}{q'(x)} \right] \right] \\
&KL\left(q \| q'\right) \geqslant 0 \quad \forall q, q' \\
&KL\left(q''\left(q'\right) = 0\right) \text{ iff } q = q' \\
&KL\left(q'\left(q'\right) \neq KL\left(q' \| q\right)\right)
\end{aligned}$$

From these properties we can derive:

$$\ln P_\theta(x) = ELBO(x, \theta, q_x) \text{ iff } q_x(z) = P_0(z|x)$$

$$\Rightarrow \arg\max_{q_x(\cdot)} ELBO(x, \theta, q_x) = P_\theta(z \mid x)$$

$$\operatorname{argmax}_\theta \, ELBO(x, \theta, p_\theta(z \mid x))) = \operatorname*{argmax}_\theta \ln P_\theta(x)$$

$$\arg\max_{\theta, q_x} ELBO(x, \theta, q) = \operatorname{argmax} \ln P_\theta(x)$$