# Variational Inference and Variational Auto-Encoder

Jingfeng Wu

Created: March 2018
Last updated: June 27, 2020

For more details, please refer to [1, 2, 3, 4].

## 1 Variational Inference

Let $X = \{x_i\}_{i=1}^N$ be a set of observed data. In Variational Inference (VI), we want to approximate a complicated and intractable conditional distribution $P(z|X)$ with some simple and tractable distribution $Q(z;\upsilon)$ parameterized by $\upsilon$. Here we do not write the dependence of $Q(z;\upsilon)$ on $X$ explicitly, since $X$, the observed data, is fixed. $Q(z;\upsilon)$ can be replaced with a conditional distribution when one assumes $X$ is variable and drawn from some distribution.

First one can easily obtain that

$$D_{\mathrm{KL}}(Q(z;\upsilon) \mid\mid P(z|X)) = \sum_z Q(z;\upsilon)\log\frac{Q(z;\upsilon)}{P(z|X)} = \log P(X) + \sum_z Q(z;\upsilon)\log\frac{Q(z;\upsilon)}{P(z,X)}. \tag{1}$$

Note that $\log P(X)$ is fixed since $X$ is given. Suppose the desired conditional distribution $P(z|X)$ is not that complicated, and our model $Q(z;\upsilon)$ is flexible enough such that $Q(z;\upsilon^*) = P(z|X)$ for $\upsilon^* = \arg\min_\upsilon D_{\mathrm{KL}}(Q(z;\upsilon) \mid\mid P(z|X))$. Thus by taking minimization with respect to $\upsilon$ in both sides, we have

$$0 = \min_\upsilon D_{\mathrm{KL}}(Q(z;\upsilon) \mid\mid P(z|X)) = \log P(X) + \min_\upsilon \sum_z Q(z;\upsilon)\log\frac{Q(z;\upsilon)}{P(z,X)}. \tag{2}$$

Thus

$$\log P(X) = \max_\upsilon -\sum_z Q(z;\upsilon)\log\frac{Q(z;\upsilon)}{P(z,X)}. \tag{3}$$

The key ingredient in VI is to smartly model the distributions such that the right hand side of Eq. (3) is tractable.

## 2 Variational Auto-Encoder

For an example of VI, let us elaborate Variational Auto-Encoder (VAE) [3]. Suppose we have observed a dataset $X = \{x_i\}_{i=1}^N$, and we aim to learn its distribution, i.e. we want to maximize the log-likelihood over the observed data,

$$\max_\theta \mathbb{E}_{x\in X}\log P(x;\theta). \tag{4}$$

Now let us introduce $Q(z|x;\upsilon)$ to approximate $P(z|x;\theta)$. By VI (3), we have

$$\begin{aligned}\log P(x;\theta) &= \max_\upsilon -\sum_z Q(z|x;\upsilon)\log\frac{Q(z|x;\upsilon)}{P(z,x;\theta)} \\ &= \max_\upsilon \sum_z Q(z|x,\upsilon)\log P(x|z;\theta) - D_{\mathrm{KL}}(Q(z|x;\upsilon) \mid\mid P(z;\theta))\end{aligned} \tag{5}$$

Thus the maximum log-likelihood (4) becomes

$$\max_\theta \mathbb{E}_{x\in X}\log P(x;\theta) = \max_{\theta,\upsilon} \mathbb{E}_{x\in X}\sum_z Q(z|x,\upsilon)\log P(x|z;\theta) - D_{\mathrm{KL}}(Q(z|x;\upsilon) \mid\mid P(z;\theta)) \tag{6}$$

Let $A = \sum_z Q(z|x,\upsilon)\log P(x|z;\theta)$ and $B = D_{\mathrm{KL}}(Q(z|x;\upsilon) \mid\mid P(z;\theta))$. In order to optimization the VAE loss (6), it remains to show how to compute the right hand side of Eq. (6), i.e., $A$ and $B$.

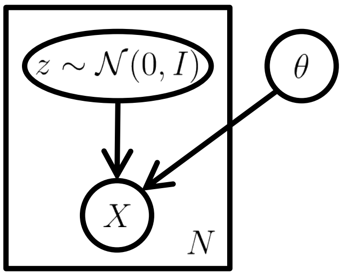


Figure 1: The standard VAE model represented as a graphical model. Note the conspicuous lack of any structure or even an "encoder" pathway: it is possible to sample from the model without any input. Here, the rectangle is "plate notation" meaning that we can sample from $z$ and $X$ $N$ times while the model parameters $\theta$ remain fixed.

## 2.1 The reparameterization trick

Notice that $A = \mathbb{E}_{z\sim Q(z|x,\upsilon)} \log P(x|z;\theta)$ is an expectation over some hidden random variable $z$. The optimization of $\theta$ can be done with Monte-Carlo estimation and typical gradient descent (or its variants). Generally, however, it is intractable to calculate the gradient on $\upsilon$ since a random variable $z$ is not differentiable. The solution to this challenge involves an important trick called *the reparameterization trick*. See Figure 1 for some intuition.

Let us model $Q(z|x,\upsilon)$ as a Gaussian distribution:

$$Q(z|x,\upsilon) = \mathcal{N}(\mu(x;\upsilon), \Sigma(x;\upsilon)). \tag{7}$$

Thus $z$ can be reparameterized as

$$z = \mu(x;\upsilon) + \Sigma(x;\upsilon)^{\frac{1}{2}} \cdot \epsilon, \quad \epsilon \ \sim \mathcal{N}(0, I). \tag{8}$$

Then we have

$$A = \mathbb{E}_{z\sim Q(z|x,\upsilon)} \log P(x|z;\theta) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \log P(x|z = \mu(x;\upsilon) + \Sigma(x;\upsilon)^{1/2} \cdot \epsilon). \tag{9}$$

In this way we can calculate gradient with respect to $\upsilon$ as

$$\frac{\partial A}{\partial \upsilon} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \frac{\partial \log P(x|z;\theta)}{\partial z} \frac{\partial z}{\partial \upsilon} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)} \frac{\partial \log P(x|z;\theta)}{\partial z} \left( \frac{\partial \mu(x;\upsilon)}{\partial \upsilon} + \frac{\partial \Sigma(x;\upsilon)^{\frac{1}{2}}}{\partial \upsilon} \epsilon \right), \tag{10}$$

which could be approximated via Monte-Carlo estimation.

## 2.2 KL divergence between Gaussian distributions

The second term $B = D_{\text{KL}}(Q(z|x;\upsilon) \mid\mid P(z;\theta))$ can be simply handled by assuming the distributions are Gaussian.

Remember that for two $k$-dimensional Gaussian distributions, their KL divergence can be computed in closed form,

$$D_{\text{KL}}(\mathcal{N}(\mu_0, \Sigma_0) \mid\mid \mathcal{N}(\mu_1, \Sigma_1)) = \frac{1}{2} \left( \text{Tr}\left(\Sigma_1^{-1}\Sigma_0\right) + (\mu_1 - \mu_0)^\top \Sigma_1^{-1}(\mu_1 - \mu_0) - k + \log\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right). \tag{11}$$

Thus when we assume $Q(z|x;\upsilon), P(z;\theta)$ are Gaussian distributions,

$$Q(z|x,\upsilon) = \mathcal{N}(\mu(x;\upsilon), \Sigma(x;\upsilon)), \quad P(z;\theta) = \mathcal{N}(z|0, I_k), \tag{12}$$

we obtain

$$B = D_{\text{KL}}(Q(z|x,\upsilon) \mid\mid P(z;\theta)) = \frac{1}{2} \left( \text{Tr}\left(\Sigma(x;\upsilon)\right) + \mu(x;\upsilon)^T \mu(x;\upsilon) - k - \log\det\left(\Sigma(x;\upsilon)\right)\right). \tag{13}$$

For the efficiency of evaluating determinate, we further assume $\Sigma(x;\upsilon)$ is diagonal.

### 2.3 Summary

The key ideas behind VAE are 1) variational inference and 2) the reparameterization trick. Suppose the family $Q(z|x;v)$ and $P(z;\theta)$, e.g., diagonal Gaussian parameterized by neural networks, are flexible enough, VAE indeed has the ability to learn the distribution over $X$. Nonetheless, in practice, there could be much trouble with such over-simplified modeling, i.e., 1) the Gaussian prior casues blur in generated $x$, and 2) the diagonal Gaussian fails to model the comprehensive coupling between different features.

All in all, no matter how fancy VAE looks like, it is still an "auto-encoder". One can view $P(x|z;\theta)$ as the decoder, and $Q(z|x;v)$ as the encoder. Under this interpretation, the term $A$ in Eq. (6) is actually the reconstruction error as in other typical auto-encoders. The difference happens in the term $B$ in Eq. (6), which is an regularizer related to a Gaussian prior for the hidden variable $z$. It is quite surprising such a simple regularization brings auto-encoder the ability to generate meaningful, at least looks meaningful, new data.

## References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[2] Carl Doersch. Tutorial on variational autoencoders, 2016.

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[4] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders, 2019.