Expectation-Maximization, Variational Auto-Encoders, and Variational Inference

Guillaume Wang

guiwang@student.ethz.ch

September 3, 2020*

These are personal notes, feel free to use or edit them (the source `tex` *file should be available wherever you found the* `pdf`*) but they come with no guarantee of exactitude. If you spot any mistake, please do let me know!*

1 Overview

I have heard mentioned in several courses that expectation-maximization (EM), variational auto-encoders (VAE) and variational inference (VI) are more or less the same thing. Indeed the derivations look similar (lower-bound on log-likelihood involving the ELBO).

Here I try to clarify the links between the three, and in what sense they are the same. To do this I start by introducing them in a principled way. Suppose the following setting.

- Given data $\boldsymbol{x}$, typically consisting of several observations $\boldsymbol{x} = (x_1, ..., x_n)$
- Assume the generative model with latent variable z

$$z \xrightarrow{p_\theta(x|z)} x, \qquad z \sim p_\theta(z) \tag{1}$$

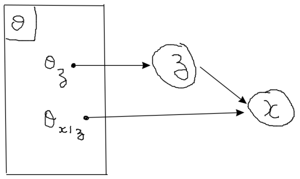

Figure 1: Probabilistic graphic model (Bayesian network) of the generative model

*v1: May 2020. v2: August 2020 (fix typos, rework section on VI, add paragraphs on mean-field approximation). v3: Septembre 2020 (add some remarks, rework section on VAE).

EM EM is when we want to find θ that maximizes the log-likelihood $\log p_\theta(\boldsymbol{x})$, where $\boldsymbol{x}$ is the data. In other words we want the maximum-likelihood estimator (MLE) of the parameter θ. In practice the flow of the discussion is often like this:

1. We have data $\boldsymbol{x}$ (fixed design) and a complicated generative model $x \sim p_\theta(x)$

2. We notice that by using a well-chosen auxiliary random variable z that acts as missing data, the complicated generative model can be cast into the form (1)

3. We use the EM algorithm based on that (see Section 3)

VAE One way to motivate the use of VAE is to suppose the following setting.

- We have a certain source of data $p^*(x)$ (e.g pictures of cats, randomly drawn from the manifold consisting of pictures of cats), which we want to model using (1) as generative model.

- We want the latent variables z to be easy to sample from. In fact we impose a certain $p(z)$, typically standard Gaussian. [1] In the terms of Figure 1, we don't consider learning θ_z.

- We also want $p_\theta(x|z)$ to be simple, typically deterministic $x = F_\theta(z)$.
  This way we get a generative algorithm to simulate the source of data: draw $z \sim p(z)$ and then $x \sim p_\theta(x|z)$. Of course the quality of this generative algorithm depends on the choice of θ.

- We consider θ as a continuous variable. Hence we can look for the best θ using techniques like gradient descent.

In summary, VAE consists in finding a $p_\theta(x|z)$ such that $p^*(x) \approx p_\theta(x|z)p(z)$, based on observations of x only.

Clearly this setting can be seen as a special case of EM where the prior $p(z)$ is fixed, since the goal is again to find parameters θ for the generative model based on data $\boldsymbol{x}$. But there is an alternative interpretation as an auto-encoder with a "prior on the codewords", as we will see in Section 4.

VI Variational inference is when we want to estimate a complicated distribution $\Phi(X)$ using a simpler one, $Q(X)$:

$$Q = \underset{Q \in \mathcal{Q}}{\arg\min}\, D\left[Q \| \Phi\right] \tag{2}$$

At first sight this problem is quite different from (1). However the ELBO pops up in the analysis, which points to a link with that model. This is discussed in Section 5.

Remark 1 (I.i.d samples). As mentioned earlier, typically the data consists of n observations $\boldsymbol{x} = (x_1, ..., x_n)$, and it is natural to assume that they are i.i.d samples of some data source. In other words the generative model (1) is

$$z_i \xrightarrow{p_\theta(x|z)} x_i, \qquad z_i \sim p_\theta(z) \tag{3}$$

for all $i \in [n]$, where the parameter θ and the distributions $p_\theta(x|z), p_\theta(z)$ do not depend on i.

In this case, there are some simplifications possible in the implementation of EM and VAE and VI. But the theory doesn't depend on it being the case. So, in this document, we simply denote by x and z the values of observation and latent variable respectively, and/or the associated random variables.

[1] Or can use Gaussian with parameters θ_z but I don't think it changes the theory as it can be absorbed in $p_\theta(x|z)$.

2 Calculations

All three methods are typically introduced by the same classic calculation, which we present here.

The starting point is a *missing-data trick*, where we use Jensen with a fictional distribution to get a lower-bound on the log-likelihood. We can choose any distribution on z to apply this trick, and we can even make it depend on x. Namely, let $q(z|x)$ an arbitrary distribution,

$$\begin{aligned}\log p_\theta(x) &= \log \sum_z p_\theta(x,z) && (4)\\ &= \log \sum_z q(z|x)\frac{p_\theta(x,z)}{q(z|x)} && (5)\\ &\geq \sum_z q(z|x)\log\frac{p_\theta(x,z)}{q(z|x)} && (6)\\ &=: ELBO && (7)\end{aligned}$$

This motivates the definition of the ELBO ("Evidence Lower-BOund"). [2] The ELBO has other interesting expressions, as will be discussed momentarily.

It turns out that we can actually calculate the looseness of the lower-bound (6). Indeed, the difference between (4) and (6) is equal to

$$\begin{aligned}\log p_\theta(x) - ELBO &= \log p_\theta(x) - \sum_z q(z|x)\log\frac{p_\theta(x,z)}{q(z|x)} && (8)\\ &= \sum_z q(z|x)\left(\log p_\theta(x) - \log\frac{p_\theta(x,z)}{q(z|x)}\right) && (9)\\ &= \sum_z q(z|x)\log\frac{p_\theta(x)\ q(z|x)}{p_\theta(x,z)} && (10)\\ &= \sum_z q(z|x)\log\frac{q(z|x)}{p_\theta(z|x)} && (11)\\ &= D\left[q(\cdot|x)\|p_\theta(\cdot|x)\right] && (12)\end{aligned}$$

The difference is a Kullback-Leibler divergence, so ≥ 0, as we already knew.

In summary, we actually have that

$$\begin{aligned}\log p_\theta(x) &= ELBO(\theta,q) + D\left[q(\cdot|x)\|p_\theta(\cdot|x)\right] && (13)\\ \text{where } ELBO(\theta,q) &= \sum_z q(z|x)\log\frac{p_\theta(x,z)}{q(z|x)} && (14)\end{aligned}$$

Of course it is possible to start from $\log p_\theta(x)$ and write equalities in order to obtain (13) directly.

Remark 2. Note that we derived (13) in full generality, in particular we didn't assume that θ was the true parameter. It holds for any choice of θ and q.

[2] The quantity $\log p_\theta(x)$ is commonly called log-likelihood when seen as a function of θ, and log-evidence when seen as a function of x. I don't know why people chose to take the latter view when giving its name to the ELBO, since x is fixed observations while θ is model parameters to be fitted.

The ELBO can be rewritten in several interesting ways.

1. The following expression will be helpful for EM:

$$
\begin{aligned}
ELBO(\theta, q) &= \sum_z q(z|x) \log \frac{p_\theta(x,z)}{q(z|x)} && (15)\\
&= \mathbb{E}_{z\sim q(\cdot|x)} \left[\log p_\theta(x,z)\right] + H[q(\cdot|x)] && (16)
\end{aligned}
$$

   where the second term is the entropy of $q(\cdot|x)$ *where x is fixed* (not a conditional entropy), and *it does not depend on θ.*

2. The following expression will be helpful in the context of VAE:

$$
\begin{aligned}
ELBO(\theta, q) &= \sum_z q(z|x) \log \frac{p_\theta(x|z)p_\theta(z)}{q(z|x)} && (17)\\
&= \mathbb{E}_{z\sim q(\cdot|x)} \left[\log p_\theta(x|z)\right] + \mathbb{E}_{q(z|x)} \left[\log \frac{p_\theta(z)}{q(z|x)}\right] && (18)\\
&= \mathbb{E}_{z\sim q(\cdot|x)} \left[\log p_\theta(x|z)\right] - D\left[q(\cdot|x)\|p_\theta(\cdot)\right] && (19)
\end{aligned}
$$

   where the second term penalizes choices of $q(z|x)$ that are "far" from the prior $p_\theta(z)$. VAE makes the assumption that θ_z is fixed i.e $p_\theta(z) = p(z)$, so that the second term will not depend on θ.

3. The following expression will be helpful when discussing VI:

$$
ELBO(\theta, q) = \log p_\theta(x) - D\left[q(\cdot|x)\|p_\theta(\cdot|x)\right] \qquad (20)
$$

   where *the first term does not depend on the choice of* $q(z|x)$. This is just equation (13) rewritten. (Or you can rederive it, totally equivalently, by noticing that $\log \frac{p_\theta(x,z)}{q(z|x)} = \log \frac{p_\theta(z|x)p_\theta(x)}{q(z|x)}$ and doing the same calculations as for (19).)

These generic calculations were for arbitrary distribution $q(z|x)$ and any θ. In EM, VAE and VI, $q(z|x)$ is instantiated to something smart.

3 The EM algorithm

EM is when we want to find θ that maximizes the log-likelihood $\log p_\theta(x)$, where x is the data. In other words we want the MLE estimator of the parameter θ.

3.1 Proof for vanilla EM

To maximize $\log p_\theta(x)$, EM starts from an initial guess θ_0, and finds θ_1 that provably improves upon θ_0. Then we repeat this N times, so that the guesses $\theta_0, \theta_1, ..., \theta_N$ each improve the previous one:

$$\log p_{\theta_N}(x) \geq ... \geq \log p_{\theta_1}(x) \geq \log p_{\theta_0}(x) \tag{21}$$

We have no guarantee of convergence to even a local maximum of $\theta \mapsto \log p_\theta(x)$, but at least we do increasingly better, and EM works well as a heuristic.

For convenience, we rewrite here the conclusion of the calculations (Section 2): for arbitrary distribution $q(z|x)$, and any parameter value θ_i, we have

$$\log p_{\theta_i}(x) = ELBO(\theta_i, q) + D\left[q(\cdot|x) \| p_{\theta_i}(\cdot|x)\right] \tag{22}$$

$$\text{where } ELBO(\theta_i, q) = \sum_z q(z|x) \log \frac{p_{\theta_i}(x,z)}{q(z|x)} \tag{23}$$

Given an initial guess θ_0, EM finds θ_1 in two steps:

1. **E-step** Consider (22) applied to $\theta_i = \theta_0$. Choose $q(z|x)$ that sets the D term to 0. Obviously that means we choose $q(z|x) = p_{\theta_0}(z|x)$. This corresponds to equality in the ELBO lower-bound.

$$\log p_{\theta_0}(x) = ELBO(\theta_0, p_{\theta_0}) + \underbrace{D\left[p_{\theta_0}(\cdot|x) \| p_{\theta_0}(\cdot|x)\right]}_{=0} \tag{24}$$

2. **M-step** Consider (22) applied to $q(z|x) = p_{\theta_0}(z|x)$ and $\theta_i = \theta_1$ (to be chosen). Then the lower-bound is

$$\log p_{\theta_1}(x) = ELBO(\theta_1, p_{\theta_0}) + \underbrace{D\left[p_{\theta_0}(\cdot|x) \| p_{\theta_1}(\cdot|x)\right]}_{\geq 0} \tag{25}$$

Choose θ_1 that maximizes this: $\theta_1 = \arg\max_{\theta_1} ELBO(\theta_1, p_{\theta_0})$.
According to (16), this is equivalent to $\theta_1 = \arg\max_{\theta_1} \mathbb{E}_{z\sim p_{\theta_0}(\cdot|x)}\left[\log p_{\theta_1}(x,z)\right]$.

Then by our choice of θ_1, **we have provably improved the log-likelihood**:

$$\log p_{\theta_1}(x) \geq ELBO(\theta_1, p_{\theta_0}) \geq ELBO(\theta_0, p_{\theta_0}) = \log p_{\theta_0}(x) \tag{26}$$

where we used successively (25), the definition of θ_1, and (24).

To recap EM in practice: given a previous estimate $\hat{\theta}^{\text{old}}$,

1. E-step: compute $p_{\hat{\theta}^{\text{old}}}(z|x)$
   This is the posterior of the latent random variable in the generative model when using $\hat{\theta}^{\text{old}}$ as the parameter
2. M-step: choose as new estimate $\hat{\theta} = \arg\max_{\theta'} \mathbb{E}_{z\sim p_{\hat{\theta}^{\text{old}}}(\cdot|x)} \left[\log p_{\theta'}(x,z)\right]$
   This is the (expected) log-likelihood of the (stochastic) completed dataset, where missing data z were completed using the distribution of the E-step [3]

Algorithm 1: Generic EM algorithm

```
Initialize $\hat{\theta}_0$
$t := 0$
while $t < t_f$ and $-\Delta \log p_{\hat{\theta}}(x) > \varepsilon$ do
    $t := t + 1$
    (E-step) compute $p_{\hat{\theta}_{t-1}}(z|x)$
    (M-step) $\hat{\theta}_t = \arg\max_{\theta'} \mathbb{E}_{z\sim p_{\hat{\theta}_{t-1}}(\cdot|x)} [\log p_{\theta'}(x,z)]$
end
return $\hat{\theta}_t$
```

I should highlight that this is probably not the most insightful way to motivate the EM algorithm; the goal of this section was primarily to show how it relates to the ELBO trick (6).

3.2 Variational EM

In the same way that Variational Inference replaces

$$P^* = \arg\min_{P} D[P\|P^*] \tag{27}$$

by

$$Q = \arg\min_{Q\in\mathcal{Q}} D[Q\|P^*] \tag{28}$$

similarly Variational EM consists in replacing the E-step

$$q(z|x) := p_{\theta_0}(z|x) = \arg\min_{q} D[q(z|x)\|p_{\theta_0}(z|x)] \tag{29}$$

by

$$q(z|x) := \arg\min_{q\in\mathcal{Q}} D[q(z|x)\|p_{\theta_0}(z|x)] \tag{30}$$

for a "simple" class of distributions $\mathcal{Q}$ (typically, using a less-expressive but simpler parametrization).

Then in the M-step, we do the same thing as before, except instead of using $p_{\theta_0}(z|x)$ to infer the completed dataset we use $q(z|x)$:

$$\theta_1 := \arg\max_{\theta_1} \; ELBO(\theta_1, q) \tag{31}$$

$$= \arg\max_{\theta_1} \; \mathbb{E}_{z\sim q(\cdot|x)} \left[\log p_{\theta_1}(x,z)\right] \tag{32}$$

[3] Note that I wrote "(expected) log-likelihood", not just "expected likelihood". As is easily seen from examples (think K-means), this is what makes all the difference compared to directly maximizing $\log p_\theta(x)$.

Variational EM is useful when the generative model's parametrization makes it too complicated, or even untractable, to compute the posterior $p_{\theta_0}(z|x)$ explicitly.

Note that contrary to vanilla EM, the log-likelihood does not always improve at each iteration in general, as the derivation of the previous subsection does not apply anymore. I don't know whether any guarantee can be derived in the general case.

Remark 3. Disclaimer: I don't know whether calling this method "variational EM" is standard terminology, as a quick search on google only gives a handful of resources that do so. But there doesn't seem to be a terminology conflict with another notion, and I think the name fits. Beware however that it should not be confused with "Variational Bayes EM".

3.3 EM as alternating maximization of the ELBO

It's not difficult to check that (variational) EM effectively performs alternating maximization of $ELBO(\theta, q)$:

1. E-step: keep θ fixed and choose q that maximizes (by equality (20))

$$q = \arg\max_{q' \in \mathcal{Q}} ELBO(\theta, q') = \underbrace{\log p_\theta(x)}_{\text{cst w.r.t } q'} - D[q'(\cdot|x) \| p_\theta(\cdot|x)] \tag{33}$$

2. M-step: keep q fixed and choose θ that maximizes (by equality (16))

$$\theta = \arg\max_{\theta'} ELBO(\theta', q) = \mathbb{E}_{z \sim q(\cdot|x)} [\log p_{\theta'}(x, z)] + \underbrace{H[q(\cdot|x)]}_{\text{cst w.r.t } \theta'} \tag{34}$$

Since vanilla EM is just variational EM with $\mathcal{Q}$ consisting of all possible distributions, the above applies to it too.

Obviously the ELBO can only increase at each iteration, when running EM. This gives a possible stopping criterion: monitor the (relative) increase of $ELBO(\theta_t, q_t)$. This also gives a "simpler" reason why the vanilla EM improves the log-likelihood at each iteration: the ELBO always improves, and the E-step chooses q such that $\log p_\theta(x) = ELBO$, so clearly the log-likelihood also always improves.

4 Variational Auto-Encoders

To summarize the motivation presented in Section 1, VAE consists in finding parameter value θ such that $p^*(x) \approx p_\theta(x|z)p(z)$, based on observations of x only. This way we can generate samples mimicking the source of data $p^*(x)$ by sampling $z \sim p(z)$ (a prespecified distribution easy to sample from), then sampling $x \sim p_\theta(x|z)$ (assumed also easy to sample from, by choice of parametrization).

4.1 VAE as variant of (variational) EM

Hence, we are in the same situation as for EM: given observations x we want $\arg\max_\theta \log p_\theta(x) = \log \sum_z p(z)p_\theta(x|z)$. So we can do the exact same kind of reasoning: introduce fictional distributions $q(z|x)$ to infer completed-data, and get theoretical guarantees using the missing-data trick (6) that led to the ELBO.

This time, let us give more details on the intuition behind that reasoning, as it will be helpful to motivate the auto-encoder view. (By the way, this intuition is also helpful for understanding the EM algorithm.)

The task would be easy if we had samples of the joint (x, z): just pick θ that maximizes the joint's log-likelihood, $\theta = \arg\max_\theta \log p_\theta(x|z)p(z)$. But we only have samples from x. So instead, we use the following high-level scheme:

1. For each sample x_i, "guess" the corresponding z_i based on a current estimate of θ. I.e, "invert" the generative model $z \xrightarrow{p_\theta(x|z)} x, \quad z \sim p(z)$ into

$$x \xrightarrow{q_\phi(z|x)} z \tag{35}$$

   (If $p_\theta(x|z)$ is deterministic with $x = F_\theta(z)$, this means find F_θ^{-1}.)

   Ideally we would want to use $q_\phi(z|x) = p_\theta(z|x) := \frac{p(z)p_\theta(x|z)}{\sum_z p(z)p_\theta(x|z)}$, but that may not be possible; for example the sum over z in the denominator may be untractable.

   How to do this "inversion" is not trivial, and will be discussed momentarily.

2. For a given q_ϕ, which gives a (stochastic) dataset $(x_i, z_i)_{i\in[n]}$, find θ that maximizes the (expected) log-likelihood

$$\theta = \arg\max_{\theta'} \mathbb{E}_{z\sim q_\phi(\cdot|x)} \left[\log p_{\theta'}(x|z)p(z)\right] \tag{36}$$

   And repeat those two steps, using the updated estimates of θ and ϕ each time.

It's not difficult to realize that this is essentially what the (variational) EM algorithm does: the E-step corresponds to step 1, i.e find a $q_\phi(z|x)$ which is ideally just the posterior $p_{\theta^{\text{old}}}(z|x)$; and the M-step corresponds to step 2, i.e use q_ϕ to infer a (stochastic) dataset of the joint, and update θ by maximizing the (expected) log-likelihood.

In the variational EM, the "inversion" was done by choosing $\phi = \arg\min_{\phi'} D[q_{\phi'}(z|x)\|p_\theta(z|x)]$. However, this might not be tractable. Moreover the EM algorithm requires solving for θ in (36), which might also not be tractable. What VAE typically does instead, is choose parametrizations such that θ and ϕ are continuous and use gradient descent; see Section 4.3.

4.2 VAE as stochastic auto-encoders with prior on codewords

The high-level scheme described above can be interpreted as:

1. encode the data x using latent variables z ("codewords"), with a stochastic encoder $q_\phi(z|x)$
2. using the (stochastic) completed dataset, "fit" a stochastic decoder $p_\theta(x|z)$ [4]

Based on this idea, let us give a different way of introducing VAE, one which justifies the name "variational auto-encoder".

Schematically, VAE does the following.

- Consider a stochastic encoding policy $q_\phi(z|x)$, and a stochastic decoding policy $p_\theta(x|z)$. In other words, the codeword for the input x is chosen by sampling $z \sim q_\phi(\cdot|x)$, and the decoded estimate is chosen by sampling $\tilde{x} \sim p_\theta(\cdot|z)$.
- On a high level, the goal is to choose the encoder q_ϕ and decoder p_θ such that: for any input x, encoding it into z and decoding it into $\tilde{x}$ yields $\tilde{x} \approx x$, i.e $\tilde{x}$ as close as possible to the original x.

Still on a high level, this idea can be refined as follows. First off, instead of "for any input" we should only be interested in inputs x that come from the data source $x \sim p^*(x)$; this is easy to take into account, since the data we have, $\boldsymbol{x}$, already comes from p^*. In addition,

- Instead of decoding z into $\tilde{x}$ and thereby adding even more stochasticity, we can be content with sampling only z, and use an alternative way of judging the goodness of our encoder-decoder pair.
- Namely, if the encoder-decoder pair is good, then the probability that $\tilde{x} = x$ should be high. So we should maximize the quantity

  $$\mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x|z)\right] \tag{37}$$

  i.e the log-probability that $\tilde{x} = x$, averaged over the stochasticity of the codeword z.
- However as an additional requirement, we also want our encoder q_ϕ to be close to some pre-chosen prior $p(z)$ (explanation just below). In other words, we want to penalize encoding-policies that are "far" from the prior; so we add a penalization term

  $$-D[q_\phi(z|x)\|p(z)] \tag{38}$$

- Thus our criterion is just the ELBO, as can be seen from equality (19)

  $$\arg\max_{\theta,\phi}\ \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x|z)\right] - D[q_\phi(z|x)\|p(z)] \quad = ELBO(\theta,\phi) \tag{39}$$

Let us motivate why, in this approach, we want to choose an encoder that yields distributions $q_\phi(z|x)$ that are close to the prior $p(z)$. Recall that the goal is to use the decoder $p_\theta(x|z)$ to generate samples mimicking the data source, by sampling $z \sim p(z)$ and then $x \sim p_\theta(x|z)$. So the distribution of the codewords fed into the decoder will be $p(z)$.

[4] Here the terms "encoder" and "decoder" are used rather liberally, and not in the sense of source coding theory, since we don't even consider deterministic mappings. But this terminology seems to be pretty common in the context of VAE.

Now it is generally a good idea, when training or fitting any machine learning model, to ensure that the distribution of the inputs is the same during training and during evaluation. Otherwise the model may perform well on training data but poorly in real applications.

In our case, the decoder $p_\theta(x|z)$ can be seen as a model whose inputs are the codewords z (and outputs are the decoded $\tilde{x}$). During training, its inputs are samples $z \sim q(z|x)$. Thus, we should strive to ensure that the distributions $q_\phi(z|x)$ and $p(z)$ are "close".

Remark 4. As a bonus: using $D[q_\phi(z|x) \| p(z)]$ (reverse KL-divergence) is a good criterion in our case, since it heavily penalizes choices of q_ϕ for which $\text{support}(q_\phi(\cdot|x)) \cap \text{support}(p) \neq \emptyset$, i.e for which some inputs that are likely in real applications never show up during training.

(Where $\text{support}(P) = \{z; P(z) > \varepsilon\}$ for some $\varepsilon > 0$.)

(This is pretty informal but hopefully you get the idea.)

Remark 5. Based on the interpretation presented here, a natural move would be to use a slightly different criterion: introduce an explicit tradeoff between the two terms, i.e add a hyperparameter $\lambda > 0$ and use

$$\arg\max_{\theta,\phi} \; \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x|z)\right] - \lambda \; D[q_\phi(z|x) \| p(z)] \tag{40}$$

I don't know whether it's a good idea and whether people do so in practice. In any case, the ELBO interpretation breaks down if we do so.

4.3 Training the VAE

In summary, VAE updates θ and ϕ alternatingly, in order to maximize $ELBO(\theta, \phi)$ by gradient ascent.

"Generative model updates": θ, aka the decoder Keep ϕ fixed and perform the update step

$$\theta_t = \theta_{t-1} + \eta_\theta \nabla_\theta ELBO(\theta_{t-1}, \phi) \tag{41}$$

for some learning rate η_θ.

All that remains is to compute $\nabla_\theta ELBO(\theta, \phi)$. This is not difficult: using ELBO's expression (19), we directly have

$$ELBO(\theta, \phi) = \mathbb{E}_{z\sim q(\cdot|x)}\left[\log p_\theta(x|z)\right] - D[q_\phi(z|x) \| p(z)] \tag{42}$$

$$\nabla_\theta ELBO(\theta, \phi) = \mathbb{E}_{z\sim q(\cdot|x)}\left[\nabla_\theta \log p_\theta(x|z)\right] \tag{43}$$

which can be evaluated e.g simply by Monte-Carlo approximation (draw i.i.d samples $z \sim q_\phi(z|x)$, compute the gradient-log-probability [5] and take the average).

"Inference model updates": ϕ, aka the encoder Keep θ fixed and perform the update step

$$\phi_t = \phi_{t-1} + \eta_\phi \nabla_\phi ELBO(\theta, \phi_{t-1}) \tag{44}$$

for some learning rate η_ϕ.

Now we need to compute $\nabla_\phi ELBO(\theta, \phi)$. This turns out to be the difficult part. I won't discuss it here.

[5] Indeed computing the gradient-log-probability is "often" easy, e.g if the generative model uses Gaussian $p_\theta(x|z)$.

5 Variational Inference

Variational inference (VI) is when we want to estimate a complicated distribution $\Phi(X)$ using a simpler one, $Q_\nu(X)$:

$$Q_\nu = \operatorname*{arg\,min}_{Q_\nu \in \mathcal{Q}_N} D\left[Q_\nu \| \Phi\right] \tag{45}$$

At first sight this problem is quite different from (1). However the ELBO pops up in the analysis, which points to a link with that model.

5.1 General case and Hidden Markov Model (HMM)

Consider data generated by a true generative process

$$z \xrightarrow{p^*(x|z)} x, \qquad z \sim p^*(z) \tag{46}$$

and suppose we're interested in the *posterior* distribution of the latent variables:

$$p^*(z|x) \tag{47}$$

More precisely, we want to find a parametric approximation $q_\nu(z|x) \approx p^*(z|x)$, where q_ν can be chosen among a class of simple distributions $\mathcal{Q}_N$.

Remark 6. This is typically the case in the context of a Hidden Markov Model: [6]

$$X \xrightarrow{P^*(Y|X)} Y, \qquad X \sim P^*(X) \tag{48}$$

Typically, we have some observations of Y and we want to infer (the distribution of) the corresponding hidden variables X. However when $P^*(X|Y)$ is too complicated, we would prefer working with a tractable approximation $Q_\nu(X|Y) \approx P^*(X|Y)$.

Temporarily suppose that p^* comes from some parametric family $\mathcal{P}_\Theta$ and that $p^* = p_{\theta^*}$. Recall equation (13), which we derived in full generality. It holds for any fictional distribution $q(z|x)$ and any value of θ, including θ^*. Even if we don't suppose that p^* comes from a parametric family, we can still do the same derivation using p^* directly, so we can always write

$$\log p^*(x) = ELBO^*(q) + D\left[q(\cdot|x) \| p^*(\cdot|x)\right] \tag{49}$$

$$\text{where } ELBO^*(q) = \sum_z q(z|x) \log \frac{p^*(x,z)}{q(z|x)} \tag{50}$$

Now, (49) implies that the ELBO can be written in the form:

$$ELBO^*(q) = \log p^*(x) - D\left[q(\cdot|x) \| p^*(\cdot|x)\right] \tag{51}$$

where the first term *does not depend on the choice of* $q(z|x)$. (We had already noticed this fact in (20).)

[6] I will use uppercase letters (X, Y, $P(X,Y)$...) in the context of Hidden Markov Models, in contrast to the lowercase letters used so far (x, z, $p(x,z)$...). This is because the notations for HMM are in conflict with the previous generative model (1).

Thus, if the $\Phi(X)$ to estimate in the original VI problem (45) is $\Phi(z) = p^*(z|x)$, the problem is equivalently restated in terms of the ELBO: (conditionally on x)

$$\arg\min_{q_\nu} D\left[q_\nu(z)\|\Phi(z)\right] \quad \equiv \quad \arg\max_{q_\nu} ELBO^*(q_\nu) \tag{52}$$

Thus VI is yet another problem that consists in maximizing a lower-bound on the evidence.

Remark 7. This is *not* an explanation of why VI uses the (reverse) KL divergence as a metric for the goodness of approximation $Q_\nu(X) \approx \Phi(X)$, though. Indeed, *a priori* nothing indicates that maximizing the ELBO is a sensible thing to do. One could argue that "we want to maximize the evidence $\log p^*(x)$ so we maximize the ELBO as a surrogate", but it doesn't really make sense since the evidence is a constant, in this setting.

5.2 No-observation case and Mean Field Approximation (MFA)

In other instances of VI, for example in the case of mean-field approximation, we only have one category of random variable X (or z) and we want to approximate $\Phi(X)$ by a parametric distribution $Q_\nu(X) \in \mathcal{Q}_N$. How does this relate to the previous discussion?

My choice of notation already gives the answer: we can still reduce this to the case of (46), except there is no observed variable

$$z \longrightarrow \varnothing, \qquad z \sim p^*(z) \tag{53}$$

Then the reasoning of the previous subsection applies without modification, except that there is no conditioning on observations: $p^*(z|x) = p^*(z|\varnothing) = p^*(z)$ and $q^*(z|x) = q^*(z)$.

Moreover, introducing the ELBO in this setting is particularly weird, since the evidence is not just constant, it's not really defined. If we write things out replacing x by $\varnothing$ formally, we get

$$\log p^*(x = \varnothing) = ELBO^*(q) + D\left[q(\cdot|\varnothing)\|p^*(\cdot|\varnothing)\right] \tag{54}$$

$$\text{where } ELBO^*(q) = \sum_z q(z|\varnothing) \log \frac{p^*(\varnothing, z)}{q(z|\varnothing)} \tag{55}$$

By simplifying, we get $ELBO^*(q) = \sum_z q(z) \log \frac{p^*(z)}{q(z)} = -D[q\|p^*]$ and $\log p^*(x = \varnothing) = \log 1 = 0$. So equation (54) still kind of makes sense but it is pointless.

I suppose we could still say formally though, that again, if the $\Phi(X)$ to estimate in the original VI problem (45) is $\Phi(z) = p^*(z)$, then it is equivalently restated in terms of the ELBO:

$$\arg\min_{q_\nu} D\left[q_\nu(z)\|\Phi(z)\right] \quad \equiv \quad \arg\max_{q_\nu} ELBO^*(q_\nu) \tag{56}$$

6 Sources and references

Section 3 is based on the many places where I came across EM. Especially useful was exercise sheet 2a of Statistical Learning Theory, ETH FS2020: "Series 2A Feb 24, 2020 (The EM-algorithm meets Probability theory)".

Bishop 2006 Ch. 9 (pp. 423-460) gives a more classic and better-motivated view of EM; in particular Sections 9.3 and 9.4 say essentially the same thing as my Section 3, but with more words and explanations, and with some extensions.

Section 4 is based on Lecture 8 of Computational Intelligence Lab, ETH FS2020: "Generative Models", since it's the first and only time I encountered VAE.

Section 5 is based on part of tutorial 5 of Probabilistic AI, ETH HS2019. I wrote the paragraph on the no-observation case when I tried (and failed) to apply these insights to the mean-field-approximation case.

An online note by some 1st year PhD student, from 2018, gives another illustration of the link between VI and EM: `https://aodongli.github.io/files/EM%20algorithm%20and%20variational%20inference.pdf`

Appendix A is based on Bishop 2006 Section 10.1 (pp. 462-474), which claims to motivate VI starting from the ELBO. I find it unsatisfactory, so I tried to elaborate on it, but the result is also unsatisfactory, which is why I put it as an appendix.

For further reading, check out all the cool tutorials that exist online (just google "EM variational inference" for example).

A (Failed:) Variational Inference and ELBO

Remark 8. Bishop 2006 Section 10.1 (pp. 462-474) claims to motivate variational inference starting from the ELBO. I found it unsatisfactory, so in this subsection I tried to elaborate on it.

But I find the result also unsatisfactory. So, I don't think we can motivate VI in this way. Fundamentally, the ELBO trick gives a way to approximate $p^*(z|x) \approx q_\nu(z|x)$, which is what we did in 5; but it does *not* give a way to approximate $p^*(x)$ the distribution of the observed variable (the evidence).

Let us try to motivate VI starting from the ELBO. Consider the following equality holding for all $p(x, z)$ and $q(z|x)$:

$$\log p(x) = ELBO(p, q) + D\left[q(\cdot|x) \| p(\cdot|x)\right] \tag{57}$$

$$\text{where } ELBO(p, q) = \sum_z q(z) \log \frac{p(x, z)}{q(z)} \tag{58}$$

which is just equation (13) except the θ is dropped. (It's exactly the same if we suppose that $\{p_\theta, \theta \in \Theta\}$ describes all possible distributions.)

In particular, let us choose $q(z|x) = q(z)$ i.e **we consider fictional distributions that are independent of** x, and let us further **restrict to a parameterized family** $q_\nu(z)$. Then for some true distribution $p^*(z), p^*(x|z)$,

$$\log p^*(x) = ELBO^*(q_\nu) + D\left[q_\nu(\cdot) \| p^*(\cdot|x)\right] \tag{59}$$

$$\text{where } ELBO^*(q_\nu) = \sum_z q_\nu(z) \log \frac{p^*(x, z)}{q(z)} \tag{60}$$

Remark 9. This equation (59) is exactly the starting point of Bishop 2006 Section 10.1 (pp. 462-474).)

Now we want to argue that the quality of the the approximation $p^*(x) \approx \sum_z p * (x|z) q_\nu(z)$ is given by $ELBO^*(q_\nu)$. Then, clearly the best choice of $q_\nu(z)$ will be

$$\arg\max_\nu ELBO^*(q_\nu) = \arg\min_\nu D\left[q_\nu(\cdot) \| p^*(\cdot|x)\right] \tag{61}$$

↬ Nope, following this path won't allow us to relate ELBO to $\arg\min_{\tilde{q}} D\left[\tilde{q}(x) \| p^*(x)\right]$. What we would want is to arrive to

$$D_{\mathrm{KL},x}\left[p^*(x) \,\middle\|\, \sum_z p^*(x|z) q_\nu(z)\right] \tag{62}$$

which I have no idea how we could do, and I don't think I've ever seen anything that looks like this before.