9 Latent Variable Models and Variational Approaches

K-means and PCA are two popular unsupervised learning methods. Their goals are for summarising and visualising data. A wider class of methods for unsupervised learning are based on **latent variable models**, where the goal is to model the data generating process using latent variables, with the latent variables representing a summary of the data. For example, in mixture models, the latent variables correspond to assignments of data items to clusters, while in variational auto-encoders, the latent variables are lower dimensional representations of data. Latent variable models are based on a probabilistic generative modeling approach, which provides a rich toolbox of techniques, and a coherent principle to reason about, and model, complex data and to derive new algorithms.

9.1 Latent Variable Models

A **latent variable model** (**LVM**) is a probabilistic generative model where each datapoint x_i has a corresponding **latent variable** z_i. For data $\mathbf{X} = \{x_i\}_{i=1}^n$ and latents $\mathbf{Z} = \{z_i\}_{i=1}^n$, an LVM is given by

$$p_\theta(\mathbf{X}, \mathbf{Z}) = p_\theta(\mathbf{Z}) \prod_{i=1}^n p_\theta(x_i|z_i) \tag{9.1}$$

where θ are global variables that are usually treated deterministically. Occasionally we take a Bayesian approach for θ instead, giving

$$p(\theta, \mathbf{X}, \mathbf{Z}) = p(\theta)p(\mathbf{Z}|\theta) \prod_{i=1}^n p(x_i|z_i, \theta). \tag{9.2}$$

In both cases, the underlying assumption is that each datapoint x_i can be explained by its latent variable z_i in conjunction with the global parameters θ. Each z_i is often lower dimensional, simpler, and/or more interpretable than the corresponding x_i, thereby providing a useful **representation** of the datapoint.

If we assume that our data is i.i.d. given θ, this implies that all the z_i should also be conditionally independent of each other given θ. This is typically a reasonable assumption when there is no natural ordering to the data. Moreover, it can be an extremely powerful assumption to make as it needs a number of important factorizations that we will exploit throughout the chapter. The resulting class of models are known as **factorized** LVMs

and have joint distributions

$$p_\theta(\mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} p_\theta(z_i) p_\theta(x_i|z_i)$$

We will focus mostly on such factorized LVMs.

9.1.1 Clustering and Mixture Modelling

K-means and hierarchical clustering are non-probabilistic algorithms, based on intuitive notions of clustering "similar" instances together and "dissimilar" instances apart. Their goal is not to model the probability of the observed data items. In contrast, a **mixture model** is a probabilistic counterpart which described a full generative process of data, and can produce "soft" assignment of data items to clusters. Mixture models assume that our dataset $\mathbf{X}$ was created by sampling iid from K distinct populations (called **mixture components**). In other words, datapoints come from a mixture of several sources and the model for the data can be viewed as a convex combination of several distinct probability distributions, often modelled with a given parametric family.

Samples in population k can be modelled using a distribution F_k with density $f_k(x; \lambda_k)$, where λ_k is the *model parameter* for the k-th component. For a concrete example, consider a p-dimensional multivariate normal density with unknown mean μ_k and *known diagonal* covariance $\sigma^2 I$ (such that $\lambda_k = \mu_k$),

$$f_k(x; \lambda_k) = |2\pi\sigma^2|^{-\frac{p}{2}} \exp\left(-\frac{1}{2\sigma^2}\|x - \mu_k\|_2^2\right). \tag{9.3}$$

More generally, a mixture model corresponds to the following generative model, whereby for each data item $i = 1, 2, \ldots, n$, we

(i) first sample the **assignment variable** (independently for each data item i):

$$Z_i \overset{iid}{\sim} \text{Discrete}(\pi_1, \ldots, \pi_K) \qquad \text{i.e., } \mathbb{P}(Z_i = k) = \pi_k$$

where for $k = 1, \ldots, K$, $\pi_k \geq 0$, such that $\sum_{k=1}^{K} \pi_k = 1$, are the **mixture weights** (also known as the **mixing proportions**), which are additional model parameters to be inferred;

(ii) then, given the assignment $Z_i = k$ of the mixture component, $X_i = (X_i^{(1)}, \ldots, X_i^{(p)})^\top$ is sampled (independently) from the corresponding k-th component:

$$X_i|(Z_i = k) \sim f_k(x; \lambda_k).$$

We observe $X_i = x_i$ for each i but do not observe its assignment Z_i (latent variables), and would like to learn the parameters $\theta = (\lambda_1, \ldots, \lambda_K, \pi_1, \ldots, \pi_K)$ as well as infer the latent variables.

Here the log joint density of the model is given by

$$p_\theta(\mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} \pi_{z_i} f_k(x_i; \lambda_{z_i}) \tag{9.4}$$

from which we can marginalize over the latent variables to give the **marginal distribution**

$$p_\theta(\mathbf{X}) = \sum_{z_1=1}^{K} \cdots \sum_{z_n=1}^{K} \prod_{i=1}^{n} \pi_{z_i} f_k(x_i; \lambda_{z_i}) = \prod_{i=1}^{n} \left(\sum_{k=1}^{K} \pi_k f_k(x_i; \lambda_k) \right). \tag{9.5}$$

Once $\mathbf{X}$ is observed, this forms the marginal likelihood of the model, or equivalently the model evidence. The log of this marginal likelihood, sometimes called the **marginal log likelihood**, is quantity that will be of particular interest:

$$\ell(\theta) := \log p_\theta(\mathbf{X}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f_k(x_i; \lambda_k).$$

Another important quantity is the **posterior distribution**,

$$p_\theta(\mathbf{Z}|\mathbf{X}) = \frac{p_\theta(\mathbf{X}, \mathbf{Z})}{p_\theta(\mathbf{X})} = \prod_{i=1}^{n} p_\theta(z_i|x_i) \tag{9.6}$$

which factorises into posterior distributions of individual latent variables z_i given the corresponding observations x_i in case of K-means, as our data is assumed iid.

In the context of latent variable models, there are two important processes: **inference**, which is the computation or approximation of the posterior distribution over latent variables, and learning, which is the estimation of the model parameters θ. Typically the learning algorithm is done via maximising the marginal log likelihood (or some approximation thereof), which generally quires some form of inference to be conducted alongside or as a subroutine. However, direct maximisation is often not feasible and the marginal log likelihood will often have many local optima. Fortunately, there is a general approach for both learning and inference based on the formulation of an objective function called the *evidence lower bound* (ELBO), which is a lower bound on the marginal log likelihood. We will find that the ELBO permits both model learning and inference, either in alternating fashion through the *Expectation Maximisation (EM) algorithm*, or simultaneously, via the *variational auto-encoder* framework.

9.2 KL Divergence and Gibbs' Inequality

Before we describe the variational formulation, we will review the notion of **Kullback-Leibler (KL) divergence** or **relative entropy** between probability distributions P and Q.

KL divergence.

- Let P and Q be two absolutely continuous probability distributions on $\mathcal{X} \subseteq \mathbb{R}^d$ with densities p and q respectively. Then the KL divergence *from* P *to* Q is defined as

$$\mathbb{D}_{\mathrm{KL}}(P \parallel Q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx. \tag{9.7}$$

- Let P and Q be two discrete probability distributions with probability mass functions p and q respectively. Then the KL divergence *from* P *to* Q is defined as

$$\mathbb{D}_{\mathrm{KL}}(P \parallel Q) = \sum_i p(x_i) \log \frac{p(x_i)}{q(x_i)}. \tag{9.8}$$

In both cases, we can write

$$\mathbb{D}_{\mathrm{KL}}(P \parallel Q) = \mathbb{E}_p \left[\log \frac{p(X)}{q(X)} \right], \tag{9.9}$$

where $\mathbb{E}_p$ denotes that expectation is taken over p. By convexity of $f(x) = -\log(x)$ and Jensen's inequality (9.11), we have that

$$\mathbb{D}_{\mathrm{KL}}(P \parallel Q) = \mathbb{E}_p \left[-\log \frac{q(X)}{p(X)} \right] \geq -\log \mathbb{E}_p \frac{q(X)}{p(X)} = 0, \tag{9.10}$$

where in the last step we used that $\int_{\mathcal{X}} q(x)dx = 1$ in continuous case and $\sum_i q(x_i) = 1$ in discrete case.

Jensen's inequality. Let f be a convex function and X be a random variable. Then

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}X). \tag{9.11}$$

If f is strictly convex, then equality holds if and only if X is almost surely a constant.

Thus, we conclude that KL-divergence is always non-negative. This consequence of Jensen's inequality is called **Gibbs' inequality**. Moreover, since $f(x) = -\log(x)$ is strictly convex on $x > 0$, the equality holds if and only if $p(x) = q(x)$ almost everywhere, i.e. $P = Q$. Note that in general KL-divergence is *not symmetric*: $\mathbb{D}_{\mathrm{KL}}(P \parallel Q) \neq \mathbb{D}_{\mathrm{KL}}(Q \parallel P)$.

9.3 Variational Free Energy and the EM Algorithm

We now break from the structuring of the lectures/slides—where we introduce ELBOs through the eyes of variational inference—to provide an alternative viewpoint through

the **expectation maximization algorithm**, more commonly known as simply the **EM algorithm**. If you have not already done so, it is recommend that you go through the lectures/slides before this section, as the viewpoint there will be the main one we consider and the one that is most prevalent in modern research and applications; with the following given as an alternative (somewhat more "old–school") perspective.

The EM algorithm is a general purpose iterative strategy for local maximisation of the marginal likelihood under missing data/hidden variables. The method has been proposed many times for specific models—it was given its name and studied as a general framework by [16].

Let $\mathbf{X}$ be a set of observed variables and let $\mathbf{Z}$ be a set of latent variables. Our probabilistic model is given by $p_\theta(\mathbf{X}, \mathbf{Z})$, where $\mathbf{Z}$ is a set of unknown variable we need to infer through Bayesian inference. We would like to maximise the marginal log-likelihood $\ell(\theta) = \log p_\theta(\mathbf{X}) = \log \int p_\theta(\mathbf{X}, \mathbf{Z}) d\mathbf{Z}$ with respect to θ. However, marginalisation of latent variables typically results in an intractable optimization problem and we need to resort to approximations or find some other way around this intractability.

Now, assume for a moment that we have access to another objective function $\mathcal{L}(\theta, q)$, where $q(\mathbf{Z})$ is a certain distribution on $\mathbf{Z}$, which we are free to choose and will call **variational distribution**. Moreover, assume that $\mathcal{L}$ satisfies

$$\mathcal{L}(\theta, q) \leq \ell(\theta) \text{ for all } \theta,\, q, \tag{9.12}$$

$$\max_q \mathcal{L}(\theta, q) = \ell(\theta), \tag{9.13}$$

i.e. $\mathcal{L}(\theta, q)$ is a *lower bound on the log-likelihood* for any variational distribution q (9.12), which also *matches the log-likelihood* at a particular choice of q (9.13).

Given these two properities, we can construct an alternating maximisation: *coordinate ascent* algorithm as follows:

Coordinate ascent on the lower bound. For $t = 1, 2 \ldots$ until convergence:

$$q^{(t)} := \operatorname{argmax}_q \mathcal{L}(\theta^{(t-1)}, q)$$
$$\theta^{(t)} := \operatorname{argmax}_\theta \mathcal{L}(\theta, q^{(t)}).$$

Theorem 16. *Assuming (9.12) and (9.13), coordinate ascent on the lower bound $\mathcal{L}(\theta, q)$ does not decrease the log likelihood $\ell(\theta)$.*

Proof. $\ell(\theta^{(t-1)}) = \mathcal{L}(\theta^{(t-1)}, q^{(t)}) \leq \mathcal{L}(\theta^{(t)}, q^{(t)}) \leq \mathcal{L}(\theta^{(t)}, q^{(t+1)}) = \ell(\theta^{(t)})$. □

But how can we find such lower bound $\mathcal{L}$? It is given by the so called *variational free energy*, which we define next.

Definition 17. Variational free energy in a latent variable model $p_\theta(\mathbf{X}, \mathbf{Z})$ is defined as

$$\mathcal{L}(\theta, q) = \mathbb{E}_{\mathbf{Z}\sim q}[\log p_\theta(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z})], \tag{9.14}$$

where q is any probability density/mass function over the latent variables $\mathbf{Z}$.

The variational free energy is equivalent to the concept of the **ELBO** as covered in lectures and to which we will return to later in the notes.

Consider the KL divergence between $q(\mathbf{Z})$ and the true conditional based on our model $p_\theta(\mathbf{Z}|\mathbf{X}) = p_\theta(\mathbf{X}, \mathbf{Z})/p_\theta(\mathbf{X})$ for the observations $\mathbf{X}$ and a fixed parameter vector θ. Since KL is non-negative,

$$\begin{aligned} 0 \leq \mathbb{D}_{\mathrm{KL}}\left[q(\mathbf{Z}) \parallel p_\theta(\mathbf{Z}|\mathbf{X})\right] &= \mathbb{E}_{\mathbf{Z}\sim q}\left[\log \frac{q(\mathbf{Z})}{p_\theta(\mathbf{Z}|\mathbf{X})}\right] \\ &= \log p_\theta(\mathbf{X}) + \mathbb{E}_{\mathbf{Z}\sim q}\left[\log \frac{q(\mathbf{Z})}{p_\theta(\mathbf{X}, \mathbf{Z})}\right]. \end{aligned}$$

Thus, we have obtained a lower bound on the marginal log-likelihood which holds true for *any parameter value* θ and *any choice of the variational distribution* q:

$$\ell(\theta) = \log p_\theta(\mathbf{X}) \geq \mathbb{E}_{\mathbf{Z}\sim q}\left[\log \frac{p_\theta(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})}\right] = \underbrace{\mathbb{E}_{\mathbf{Z}\sim q}\left[\log p_\theta(\mathbf{X}, \mathbf{Z})\right]}_{\text{energy}} \overbrace{-\mathbb{E}_{\mathbf{Z}\sim q}\left[\log q(\mathbf{Z})\right]}^{\text{entropy}}. \tag{9.15}$$

The right hand side in (9.15) is precisely the variational free energy; we see it decomposes in two terms. The first term is sometimes referred to as the *energy*. If we observed $\mathbf{Z}$, we would just maximise the complete data log-likelihood $\log p_\theta(\mathbf{X}, \mathbf{Z})$, but since $\mathbf{Z}$ is not observed we need to integrate it out, recalling that we are free to choose the distribution this done with respect to q. The second term is the Shannon entropy $H(q) = -\mathbb{E}_q \log q(\mathbf{Z})$ of the variational distribution $q(\mathbf{Z})$, and does not depend on θ (it can be thought of as the complexity penalty on q).

The terms "energy" and "entropy" come from statistical physics. The energy term here should in fact be the *negative* of the average energy of a system in thermal equilibrium, and "free energy" means energy that is accessible or useful (free as opposed to being tied up in entropy so is not accessible). Again the free energy term here should in fact be the *negative* free energy. Physical systems prefer low energy states, while here we are trying to maximise the variational free energy. We keep this inverted usage as it is common in machine learning.

The inequality becomes an equality when KL divergence is zero, i.e. when $q(\mathbf{Z}) = p_\theta(\mathbf{Z}|\mathbf{X})$, which means that the optimal choice of variational distribution q for fixed parameter value θ is *the true conditional of the latent variables given the observations and that* θ.

Thus, we have proved the following lemma:

Lemma 18. *Let $\mathcal{L}$ be the variational free energy in a latent variable model $p_\theta(\mathbf{X}, \mathbf{Z})$. Then (a) $\mathcal{L}(\theta, q) \leq \ell(\theta)$ for all q and for all θ, and (b) for any θ, $\mathcal{L}(\theta, q) = \ell(\theta)$ iff $q(\mathbf{Z}) = p_\theta(\mathbf{Z}|\mathbf{X})$.*

Thus, properties (9.12) and (9.13) are satisfied and we can recast the alternating maximisation of the variational free energy into iterative updates of q (E-step, via the plug-in full conditional of $\mathbf{Z}$ using the current estimate of θ) and the updates of θ (M-step, by maximising the 'energy' for the current estimate of q). Namely, we have the following

EM algorithm. Initialize $\theta^{(0)}$. At time $t \geq 1$:

- E-step: Set $q^{(t)}(\mathbf{Z}) = p_{\theta^{(t-1)}}(\mathbf{Z}|\mathbf{X})$
- M-step: Set $\theta^{(t)} = \arg\max_\theta \mathbb{E}_{\mathbf{Z}\sim q^{(t)}}[\log p_\theta(\mathbf{X}, \mathbf{Z})]$.

If both the E-step and M-step can be solved exactly, the EM algorithm converges to a local maximum likelihood solution. There are a number of classical models where this will indeed be the case, but as we move to more complicated approaches later in the chapter, we will consider models where neither step can be done analytically, necessitating the use of more general approaches.

9.4 EM Algorithm for Mixtures

Consider again our mixture model from Section 9.1.1 with

$$p_\theta(\mathbf{X}, \mathbf{Z}) = \prod_{i=1}^{n} \pi_{z_i} f_k(x_i; \lambda_{z_i}).$$

Recall that our latent variables $\mathbf{Z}$ are discrete (they correspond to cluster assignments) so q is a probability mass function over $\mathbf{Z} := (z_i)_{i=1}^n$. Using the expression (9.4), we can write the variational free energy as

$$\begin{aligned}
\mathcal{L}(\theta, q) =& \mathbb{E}_q[\log p_\theta(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z})] \\
=& \mathbb{E}_q\left[\left(\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}(z_i = k)\left(\log \pi_k + \log f_k(x_i; \lambda_k)\right)\right) - \log q(\mathbf{Z})\right] \\
=& \sum_{\mathbf{Z}} q(\mathbf{Z})\left[\left(\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbb{I}(z_i = k)\left(\log \pi_k + \log f_k(x_i; \lambda_k)\right)\right) - \log q(\mathbf{Z})\right] \\
=& \sum_{i=1}^{n}\sum_{k=1}^{K} q(z_i = k)\left(\log \pi_k + \log f_k(x_i; \lambda_k)\right) + H(q).
\end{aligned}$$

We will denote $Q_{ik} = q(z_i = k)$, which is called the **responsibility** of cluster k for data item i.

Now, the E-step simplifies because

$$p_\theta(\mathbf{Z}|\mathbf{X}) = \frac{p_\theta(\mathbf{X},\mathbf{Z})}{p_\theta(\mathbf{X})} = \frac{\prod_{i=1}^n \pi_{z_i} f_k(x_i;\lambda_{z_i})}{\sum_{\mathbf{Z}'}\prod_{i=1}^n \pi_{Z_i'} f_k(x_i;\lambda_{Z_i'})} = \prod_{i=1}^n \frac{\pi_{z_i} f_k(x_i;\lambda_{z_i})}{\sum_k \pi_k f_k(x_i;\lambda_k)}$$
$$= \prod_{i=1}^n p_\theta(z_i|x_i).$$

Thus, for a fixed $\theta^{(t-1)} = (\lambda_1^{(t-1)},\ldots,\lambda_K^{(t-1)},\pi_1^{(t-1)},\ldots,\pi_K^{(t-1)})$ we can set

$$Q_{ik}^{(t)} = p_{\theta^{(t-1)}}(z_i = k|x_i) = \frac{\pi_k^{(t-1)} f_k(x_i;\lambda_k^{(t-1)})}{\sum_{j=1}^K \pi_j^{(t-1)} f_k(x_i;\lambda_j^{(t-1)})}. \tag{9.16}$$

Now, consider the M-step. For mixing proportions we have a constraint that $\sum_{j=1}^K \pi_j = 1$, so we introduce the Lagrange multiplier and obtain

$$\nabla_{\pi_k}\left(\mathcal{L}(\theta,q) - \lambda(\textstyle\sum_{j=1}^K \pi_j - 1)\right) = \sum_{i=1}^n \frac{Q_{ik}}{\pi_k} - \lambda = 0 \quad \Rightarrow \quad \pi_k \propto \sum_{i=1}^n Q_{ik}.$$

Since

$$\sum_{k=1}^K\sum_{i=1}^n Q_{ik} = \sum_{i=1}^n \underbrace{\sum_{k=1}^K Q_{ik}}_{=1} = n,$$

the M-step update for mixing proportions is

$$\pi_k^{(t)} = \frac{\sum_{i=1}^n Q_{ik}^{(t)}}{n}, \tag{9.17}$$

i.e., they are simply given by the total responsibility of each cluster. Note that this update holds regardless of the form of $f_k(\cdot;\lambda_k)$ used for mixture components.

Setting the derivative with respect to λ_k to 0, we obtain

$$\nabla_{\lambda_k}\mathcal{L}(\theta,q) = \sum_{i=1}^n Q_{ik}\nabla_{\lambda_k}\log f_k(x_i;\lambda_k) = 0. \tag{9.18}$$

This equation can be solved quite easily for mixture of normals in (9.3), giving the M-step update

$$\mu_k^{(t)} = \frac{\sum_{i=1}^n Q_{ik}^{(t)} x_i}{\sum_{i=1}^n Q_{ik}^{(t)}}, \tag{9.19}$$

which implies that the k-th cluster mean estimate is simply a weighted average of all the data items, where the weights correspond to the responsibilities of cluster k for these points.

Put together, the EM for normal mixture model with known (fixed) covariance is very similar to the K-means algorithm where cluster assignments are soft, i.e. rather than assigning each data item x_i to a single cluster at each iteration, we carry forward a responsibility vector $(Q_{i1},\ldots,Q_{iK})$ giving probabilities of x_i belonging to each cluster.

EM for Normal Mixtures (known covariance) – "Soft K-means"

1. Initialize K cluster means $\mu_1, \ldots, \mu_K$ and mixing proportions $\pi_1, \ldots, \pi_K$.
2. *Update responsibilites (E-step):* For each $i = 1, \ldots, n$, $k = 1, \ldots, K$:

$$Q_{ik} = \frac{\pi_k \exp\left(-\frac{1}{2\sigma^2}\|x_i - \mu_k\|_2^2\right)}{\sum_{j=1}^K \pi_j \exp\left(-\frac{1}{2\sigma^2}\|x_i - \mu_j\|_2^2\right)} \tag{9.20}$$

3. *Update parameters (M-step):* Set $\mu_1, \ldots, \mu_K$ and $\pi_1, \ldots, \pi_K$ and based on the new cluster responsibilities:

$$\pi_k = \frac{\sum_{i=1}^n Q_{ik}}{n}, \qquad \mu_k = \frac{\sum_{i=1}^n Q_{ik} x_i}{\sum_{i=1}^n Q_{ik}}. \tag{9.21}$$

4. Repeat steps 2-3 until convergence.
5. Return the responsibilites $\{Q_{ik}\}$ and parameters $\mu_1, \ldots, \mu_K, \pi_1, \ldots, \pi_K$.

In some cases, depending on the form of $f_k(\cdot; \lambda_k)$, the M-step for π_k and E-step can both be done exactly, but the M-step update for model parameters of the components cannot. In these cases, we can use *gradient ascent* algorithm *inside the M-step*:

$$\lambda_k^{(r+1)} = \lambda_k^{(r)} + \alpha \sum_{i=1}^n Q_{ik} \nabla_{\lambda_k} \log f_k(x_i; \lambda_k^{(r)}).$$

This leads to a **generalized EM algorithm** for mixture models.

Mixture models can be applied to supervised learning settings as well. In a **mixture of experts**, we assume that a dataset of input/output pairs $\{(x_i, y_i)\}_{i=1}^n$ can be modelled using K experts. Expert k predicts y given x using some supervised model (e.g. a neural network) which outputs a distribution $f_{\theta_k}(y|x)$ with parameters θ_k. Given x, an expert is stochastically chosen using a gating predictor $g_\eta(x)$ which produces a vector of probabilities over experts. The marginal probability of y given x is then:

$$p(y|x) = \sum_{k=1}^K \{g_\eta(x)\}_k f_{\theta_k}(y|x).$$

The parameters of the mixture of experts can be learnt to maximise the log marginal likelihood given the dataset using the generalised EM algorithm.

9.5 Variational Inference

One of the workhorses of Bayesian machine learning are *variational approximations*, which turn posterior inference in intractable Bayesian models into optimization. We

have seen that Bayesian model selection proceeds by optimizing (maximizing) the model evidence. While model evidence is almost always intractable, using the same principles as in the EM algorithm (Gibbs inequality), lower bounds may be available which can be optimized instead. The variational approach to Bayesian machine learning is often referred to as **variational Bayes**.

9.5.1 The ELBO

To given contrast to the lectures, assume that we wish to take a Bayesian view on both the latents $\mathbf{Z}$ and the model parameters θ, such that our joint is given by $p(\mathbf{X}, \mathbf{Z}, \theta)$. Now, because we are using a Bayesian model, our treatment of latent variables and model parameters is exactly the same. We can now consider some joint distribution $q(\mathbf{Z}, \theta)$ of latent variables and parameters, called a **variational distribution** (similarly to EM, but note that EM was not allowed to place a distribution over θ). We claim that the quantity

$$\mathcal{L}(q) = \mathbb{E}_q \left[\log p(\mathbf{X}, \mathbf{Z}, \theta)\right] + H(q) \tag{9.22}$$

is a lower bound on log-evidence $\log p(\mathbf{X})$. Namely, we can write

$$\begin{aligned} \mathcal{L}(q) &= \mathbb{E}_q \left[\log p(\mathbf{X}, \mathbf{Z}, \theta)\right] - \mathbb{E}_q[\log q(\mathbf{Z}, \theta)] \\ &= \log p(\mathbf{X}) + \mathbb{E}_q \left[\log p(\mathbf{Z}, \theta|\mathbf{X})\right] - \mathbb{E}_q[\log q(\mathbf{Z}, \theta)] \\ &= \log p(\mathbf{X}) - \mathrm{KL}\left(q(\mathbf{Z}, \theta)||p(\mathbf{Z}, \theta|\mathbf{X})\right), \end{aligned}$$

which is by Gibbs inequality maximised (and equal to log-evidence) when the KL is zero, i.e. when $q(\mathbf{Z}, \theta) = p(\mathbf{Z}, \theta|\mathbf{X})$. Thus, for any variational distribution q, $\mathcal{L}(q) \leq \log p(x)$. Expression 9.22 is called the **evidence lower bound** (ELBO).

To reason about all the unknowns in the model, we would simply need to compute the joint posterior $p(\mathbf{Z}, \theta|\mathbf{X})$, but this is almost always intractable. Hence, variational Bayesian inference *approximates* the posterior by starting with a family $\mathcal{Q}$ of tractable variational distributions $q(\mathbf{Z}, \theta)$ (e.g. $q_\phi(\mathbf{Z}, \theta)$ where ϕ are the **variational parameters**), and aims to minimize the divergence $\mathrm{KL}\left(q(\mathbf{Z}, \theta)||p(\mathbf{Z}, \theta|\mathbf{X})\right)$ over $\mathcal{Q}$ or, equivalently, maximise the ELBO, i.e. find the tightest lower bound on the log-evidence.

In a nutshell, variational Bayes projects the (intractable) posterior $p(\mathbf{Z}, \theta|\mathbf{X})$ onto a tractable family $\mathcal{Q}$ with respect to the KL divergence $\mathrm{KL}\left(q(\mathbf{Z}, \theta)||p(\mathbf{Z}, \theta|\mathbf{X})\right)$. Alternative divergences are possible in this context. In particular, since KL is not symmetric, minimization of the "reverse" divergence $\mathrm{KL}\left(p(\mathbf{Z}, \theta|\mathbf{X})||q(\mathbf{Z}, \theta)\right)$ results in a different family of approximate Bayesian methods, known as **expectation propagation** (EP).

9.5.2 Variational EM and Mean-Field Variational Family

Finding a variational approximation requires specifying the variational family $\mathcal{Q}$. The complexity of $\mathcal{Q}$ determines the difficulty of the optimization; it is more difficult to optimize over a large family $\mathcal{Q}$ than over a simpler, smaller one. Consider a family $\mathcal{Q}$ of variational distributions which factorize across the latents and the parameters:

$q(\mathbf{Z}, \theta) = q_{\mathbf{Z}}(\mathbf{Z})\, q_\theta(\theta)$. For a fixed q_θ, we can solve for $q_{\mathbf{Z}}$ which maximises ELBO (*exercise):*

$$q_{\mathbf{Z}}(\mathbf{Z}) \propto \exp\left(\int \log p(\mathbf{X}, \mathbf{Z}, \theta) q_\theta(\theta)\, d\theta\right),$$

and by symmetry, for a fixed $q_{\mathbf{Z}}$, we can solve for q_θ which maximises ELBO:

$$q_\theta(\theta) \propto \exp\left(\int \log p(\mathbf{X}, \mathbf{Z}, \theta) q_{\mathbf{Z}}(\mathbf{Z})\, d\mathbf{Z}\right).$$

Now, one can formulate an algorithm similar to EM, which alternates between optimising $q_{\mathbf{Z}}$ and q_θ, such that each iteration increases ELBO and thus decreases the KL divergence from the posterior. Hence such an algorithm is sometimes called a **variational Bayesian EM** algorithm, VBEM, or sometimes simply **variational EM**.

We notice the symmetry between $\mathbf{Z}$ and θ. Indeed, the distinction between parameters and latent variables disappears in Bayesian modelling, as all unobserved quantities in the model are treated in the same way and our goal is to approximate their posterior distribution. In the rest of this section, we will drop θ from the notation and treat them as a part of the set of all unobserved quantities $\mathbf{Z}$.

A more general simplification we often make in variational Bayes is to focus on a **mean-field approximation** where the variational distribution fully factorizes

$$q(\mathbf{Z}) = \prod_{j=1}^{m} q_j(z_j),$$

i.e. all latent variables are mutually independent and each latent z_j is governed by its own variational factor q_j. Note that there could be a mix between categorical and continuous latents, each having the appropriate factor q_j. Also, z_j itself need not be a univariate latent—see an example with LDA below. Using the mean-field family implies that we will not be able to capture any posterior correlations between the latent variables z_j and $z_{j'}$ for $j \neq j'$ and that the best we can hope for is a rich representations of the posterior marginals.

An iterative procedure similar to Bayesian EM can now be applied to each individual factor, giving rise to an algorithm called **coordinate ascent variational inference** (CAVI) shown in Algorithm 9.1. CAVI is closely related to Gibbs sampling (a popular class of MCMC algorithms), and is based on iteratively approximating the full conditionals $p(z_j|\mathbf{Z}_{-j}, \mathbf{X}) \propto p(\mathbf{X}, \mathbf{Z})$, where $\mathbf{Z}_{-j} = [z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_n]$.

9.5.3 Complete conditionals in the exponential family

When the complete conditionals $p(z_j|\mathbf{Z}_{-j}, \mathbf{X})$ belong to an exponential family of distributions, i.e. they are given by

$$p(z_j|\mathbf{Z}_{-j}, \mathbf{X}) = h(z_j) \exp\left[z_j^\top \eta_j(\mathbf{Z}_{-j}, \mathbf{X}) - A(\eta_j(\mathbf{Z}_{-j}, \mathbf{X}))\right],$$

a particularly convenient form of CAVI is available as the updates are available in closed form. Here we assume z_j is already transformed to its appropriate sufficient statistic, $h(\cdot)$

Algorithm 9.1 Coordinate Ascent Variational Inference (CAVI)

Input: model $p(\mathbf{X}, \mathbf{Z})$, dataset $\mathbf{X}$
Output: a variational posterior $q(\mathbf{Z})$

while the ELBO has not converged **do**
  for $j = 1, \ldots, m$
    $q_j(z_j) \propto \exp\left(\mathbb{E}_{\mathbf{Z}_{-j} \sim q}\left[\log p\left(z_j | \mathbf{Z}_{-j}, \mathbf{X}\right)\right]\right)$
return $q\left(\mathbf{Z}\right) = \prod_{j=1}^{m} q_j\left(z_j\right)$

is a base measure, $A\left(\cdot\right)$ is the log-normalizer and $\eta_j\left(\mathbf{Z}_{-j}, \mathbf{X}\right)$ are the natural parameters (which depend on the conditioning set). The CAVI update is now

$$\begin{aligned} q_j(z_j) &\propto \exp\left(\mathbb{E}_{\mathbf{Z}_{-j}}\left[\log p\left(z_j | \mathbf{Z}_{-j}, \mathbf{X}\right)\right]\right) \\ &= \exp\left(\log h\left(z_j\right) + z_j^\top \mathbb{E}_{\mathbf{Z}_{-j}}\left[\eta_j\left(\mathbf{Z}_{-j}, \mathbf{X}\right)\right] - \mathbb{E}_{\mathbf{Z}_{-j}}\left[A\left(\eta_j\left(\mathbf{Z}_{-j}, \mathbf{X}\right)\right)\right]\right) \\ &\propto h\left(z_j\right) \exp\left(z_j^\top \mathbb{E}_{\mathbf{Z}_{-j}}\left[\eta_j\left(\mathbf{Z}_{-j}, \mathbf{X}\right)\right]\right) \end{aligned}$$

and thus, the variational factors are in the same exponential family as the complete conditionals with natural parameter being the expected natural parameter of the complete conditional

$$\phi_j = \mathbb{E}_{\mathbf{Z}_{-j}}\left[\eta_j\left(\mathbf{Z}_{-j}, \mathbf{X}\right)\right].$$

This setup describes many models, including Bayesian Gaussian mixtures, Dirichlet process mixtures, matrix factorization, multilevel regression, and latent Dirichlet allocation, giving thus one classical overarching CAVI algorithm with closed-form updates for many classical instances of Variational Bayes. However, such occurrences are less common in modern large–scale variational systems that tend to be based on the (stochastic) gradient based approach discussed in the lectures.

While the distinction between parameters and latent variables disappears in Bayesian modelling, there is still a relevant distinction in terms of where in the model hierarchy these unobserved quantities appear. The impact of such hierarchy is that not all updates in Algorithm 9.1 need to be performed sequentially. Multiple levels of hierarchy are also possible. In particular, for large datasets, we often perform **stochastic variational inference**, wherein updates for variational approximations of global parameters θ are made more regularly than for local latents. This is achieved by using stochastic gradient updates, analogous to those discussed in the lectures where we were looking to optimize θ rather than perform inference over it.

9.5.4 Example: Topic Modelling [Not Examinable]

Topic models[1] are a class of probabilistic models of text that lead to parsimonious representations of hidden thematic structure of a collection of documents. A popular approach

[1] While the LDA model covered here is not directly examinable, many of the discussed ideas will be helpful for more completely understanding variational inference.

to topic modelling is Latent Dirichlet Allocation (LDA) [7].[2]

Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) captures the intuition that a text document typically exhibits multiple topics and blends them in a particular way. In LDA, each topic is modelled as a probability distribution over words and each document as a mixture of corpus-wide topics (i.e. it can be identified with a distribution over topics). Each observed word in a document is then treated as a draw from the mixture, i.e. it belongs to one of the topics (mixture components). Mixture proportions are thus unique for each document, i.e. they are local latents, but mixture components are shared across the whole collection—they are global parameters.

This setting is also called a **mixed membership model**. Another important example of mixed membership model is the STRUCTURE model in population genetics. There, the DNA of each individual in a population is modelled as a mixture over ancestral populations, with each individual having different proportions of their DNA coming from each ancestral population. This is most clear for individuals of mixed ethnic ancestry, but such approaches are also applicable to, e.g. "native" British, where the the inferred population structure has been used to understand the history and migration patterns of the peoples of the British isles before modern times [37].

Let K be the number of topics and V the size of the vocabulary. LDA posits the following conditionally conjugate model.

1. For each topic in $k = 1, \ldots, K$,
   a) Draw a distribution over V words $\beta_k \sim \mathrm{Dir}_V(\eta)$
2. For each document in $d = 1, \ldots, D$,
   a) Draw a vector of topic proportions[3] $\theta_d \sim \mathrm{Dir}_K(\alpha)$
   b) For each word in $n = 1, \ldots, N_d$,
      i. Draw a topic assignment $z_{dn} \sim \mathrm{Mult}(\theta_d)$, i.e. $p(z_{dn} = k|\theta_d) = \theta_{dk}$
      ii. Draw a word $w_{dn} \sim \mathrm{Mult}(\beta_{z_{dn}})$, i.e. $p(w_{dn} = v|\beta, z) = \beta_{z_{dn}v}$

Given we observe the words in the documents, the goal of LDA is to find the posterior

$$p\left(\beta_{1:K}, \theta_{1:D}, \{z_{d,1:N_d}\}_{d=1}^{D} \,|\, \{w_{d,1:N_d}\}_{d=1}^{D}\right) = p(\text{topics, proportions, assignments} \,|\, \text{words})$$

Note that a corpus of text to be analyzed may consist of millions of documents, thus

[2] Be careful not to confuse Latent Dirichlet Allocation with Linear Discriminant Analysis which shares the acronym—these two models are not related.

[3] Note our usage of "θ" here is as a specific set of parameters rather than all the global parameters of the model. This notation is used for consistency with the standard names used for LDA in the literature.

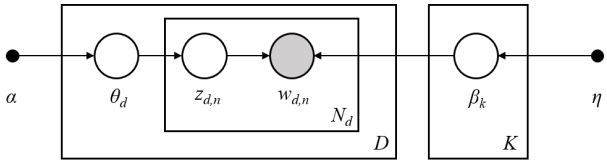

Figure 9.1: Graphical model representation of LDA. Plates represent replication, for example there are D documents each having a topic proportion vector θ_d

having possibly billions of latent variables. We can write the joint distribution as

$$\begin{aligned} p(\beta, \theta, z, w) &= \prod_{k=1}^{K} p(\beta_k; \eta) \prod_{d=1}^{D} \left\{ p(\theta_d; \alpha) \prod_{n=1}^{N_d} p(z_{dn}|\theta_d)\, p(w_{dn}|\beta, z) \right\} \\ &= \frac{1}{B(\eta)^K B(\alpha)^D} \prod_{k=1}^{K} \prod_{v=1}^{V} \beta_{kv}^{\eta_v - 1} \prod_{d=1}^{D} \left\{ \prod_{k=1}^{K} \theta_{dk}^{\alpha_k - 1} \prod_{n=1}^{N_d} \theta_{d,z_{dn}} \beta_{z_{dn},w_{dn}} \right\}. \end{aligned} \tag{9.23}$$

The model has the following latents: β (topics), θ (proportions), and z (assignments). Note that there are also hyperparameter vectors $\eta \in \mathbb{R}_+^V$ and $\alpha \in \mathbb{R}_+^K$ in the Dirichlet priors—these are assumed fixed. While β and θ are "global" variables, we will still be performing variational inference over them as well.

Data are the observed words $\{w_{dn}\}$. There will be a slight abuse of notation here: we denote by w_{dn} both the appropriate draw from vocabulary $\{1, \ldots, V\}$—to be used for indexing—and its "one-hot" encoding, i.e. a binary V-vector with $w_{dn}[v] = 1$ if $w_{dn} = v$ and zero otherwise. We will write $w_{dn}[\cdot]$ in the case of the latter. Similarly, we denote by z_{dn} both the appropriate topic assignment from $\{1, \ldots, K\}$ and its "one-hot" encoding i.e. a binary K-vector with $z_{dn}[k] = 1$ if $z_{dn} = k$ and zero otherwise. We will write $z_{dn}[\cdot]$ in the case of the latter.

We will use a mean-field family of the form

$$q(\beta, \theta, z) \quad = \quad \prod_{k=1}^{K} q(\beta_k; \zeta_k) \prod_{d=1}^{D} \left\{ q(\theta_d; \gamma_d) \prod_{n=1}^{N_d} q(z_{dn}; \phi_{dn}) \right\}.$$

The complete conditionals are proportional to the joint distribution in (9.23):

1. Complete conditional on the topic assignment is a multinomial with

$$p(z_{dn} = k|\theta_d, \beta, w_d) \propto \theta_{dk} \beta_{k,w_{dn}} = \exp(\log \theta_{dk} + \log \beta_{k,w_{dn}}). \tag{9.24}$$

   Thus, for the variational approximation we also use a multinomial but with a "free parameter" ϕ_{dn}, where we denote $\phi_{dn}[k] = q(z_{dn} = k)$, i.e. ϕ_{dn} is simply a probability mass function over K topics.

2. Complete conditional on the topic proportions depends only on the assignments and is given by

$$p(\theta_d|z_d) = \mathrm{Dir}_K \left(\theta_d; \alpha + \sum_{n=1}^{N_d} z_{dn}[\cdot] \right). \tag{9.25}$$

For the variational approximation we also use Dirichlet, with parameter vector $\gamma_d \in \mathbb{R}_+^K$.

3. Complete conditional on the topics is

$$p\left(\beta_k | z, w\right) = \mathrm{Dir}_V\left(\beta_k; \eta + \sum_{d=1}^{D}\sum_{n=1}^{N_d} z_{dn}\left[k\right] w_{dn}\left[\cdot\right]\right). \tag{9.26}$$

For the variational approximation we also use Dirichlet, with parameter vector $\zeta_k \in \mathbb{R}_+^V$.

With these full conditionals we can derive the CAVI updates in the LDA model. We will need the following standard result about the Dirichlet distribution given here without proof.

Proposition 19. *If $\pi \sim Dir_L\left(\alpha\right)$, then*

$$\mathbb{E}\left[\log \pi_j\right] = \psi\left(\alpha_j\right) - \psi\left(\sum_{\ell=1}^{L}\alpha_\ell\right),$$

where $\psi\left(u\right) = \frac{\Gamma'(u)}{\Gamma(u)} = \int_0^\infty \left(\frac{e^{-t}}{t} - \frac{e^{-ut}}{1-e^{-t}}\right) dt$ is the digamma function.

Now we can obtain the closed-form updates for each set of the latents.

Proposition 20. *CAVI updates in the LDA model are given by*

1. $\phi_{dn}[k] \propto \exp\left(\psi\left(\gamma_{dk}\right) + \psi\left(\zeta_{k,w_{dn}}\right) - \psi\left(\sum_{v=1}^{V}\zeta_{k,v}\right)\right)$,
2. $\gamma_d = \alpha + \sum_{n=1}^{N_d}\phi_{dn}$,
3. $\zeta_k = \eta + \sum_{d=1}^{D}\sum_{n=1}^{N_d}\phi_{dn}\left[k\right] w_{dn}\left[\cdot\right]$,

where ψ is the digamma function.

Proof. Steps (2) and (3) directly follow from the exponential family properties of Dirichlet distribution. Namely for (2), we note that

$$\log p\left(\theta_d | z_d\right) = \mathrm{const} + \sum_{k=1}^{K}\left(\alpha_k + \sum_{n=1}^{N_d} z_{dn}\left[k\right] - 1\right)\log\theta_{dk},$$

so that

$$\exp\left(\mathbb{E}_{z_d\sim q}\left[\log p\left(\theta_d | z_d\right)\right]\right) \propto \prod_{k=1}^{K}\theta_{dk}^{\alpha_k + \sum_{n=1}^{N_d}\phi_{dn}[k]-1},$$

which is proportional to the Dirichlet distribution with parameter vector $\gamma_d = \alpha + \sum_{n=1}^{N_d}\phi_{dn}$. The same logic applies for (3).

For (1), we make use of Proposition 19 and write

$$\begin{aligned}
\phi_{dn}[k] &\propto \exp\left(\mathbb{E}_{\theta_d,\beta\sim q}\left[\log p\left(z_{dn}=k|\theta_d,\beta,w_d\right)\right]\right) \\
&\propto \exp\left(\mathbb{E}_{\theta_d\sim q}\left[\log\theta_{dk}\right]+\mathbb{E}_{\beta_k\sim q}\left[\log\beta_{k,w_{dn}}\right]\right) \\
&\propto \exp\left(\psi\left(\gamma_{dk}\right)-\psi\left(\sum_{\ell=1}^{K}\gamma_{d\ell}\right)+\psi\left(\zeta_{k,w_{dn}}\right)-\psi\left(\sum_{v=1}^{V}\zeta_{k,v}\right)\right) \\
&\propto \exp\left(\psi\left(\gamma_{dk}\right)+\psi\left(\zeta_{k,w_{dn}}\right)-\psi\left(\sum_{v=1}^{V}\zeta_{k,v}\right)\right),
\end{aligned}$$

as required. □

9.6 Variational Auto–Encoders

So far we have mostly implicitly been thinking in terms of manually constructed generative models where the global parameters θ are small to moderate dimensional, with the model making fairly strong assumptions about the generative process. However, in many scenarios one needs to deal with complex or high–dimensional data, where such manual construction is not feasible and we would rather specify a very general model class and then train the model from the data.

Deep generative models (**DGMs**) are a class of, typically factorized, LVMs that allow us to deal with such settings by using deep neural networks to parameterize the generative process in high flexible manner and then learning the parameters of this network in an end–to–end manner using stochastic gradient techniques.

DGMs provide a mechanism to train generative models in a data–driven manner that imposes far weaker assumptions than traditional approaches like graphical models. This allows them to learn extremely powerful models and operate in settings that would be far beyond the reach of traditional approaches, particularly those with high–dimensional data like images. This leads to a wide range of applications such as synthetic data generation, feature/representation learning, nonlinear dimensionality reduction (compared with, e.g., the linear dimensionality reduction of PCA), manipulation of individual datapoints, construction of robust machine learning systems, data cleaning (i.e. dealing with things like incomplete data, outliers, non–numeric data, and multiple data modalities), interpretable machine learning, metric/manifold learning, and simply data (pre–)processing in general.

There are a wide range of DGMs used in practice, far more than we can realistically cover; we will focus on one particular popular example: **variational auto–encoders** (VAEs) [29, 48].

A VAE is a factorized LVM[4] with two key components: a **generative network**, or **decoder**, $p_\theta(x|z)$, and an **inference network**, or **encoder**, $q_\phi(z|x)$. The parameters of these networks, θ and ϕ, are trained by simultaneously optimizing the ELBO with respect to both.

[4] There are also some extensions that do not make fully factorized assumptions, e.g. [35].

In the standard approach, which we will focus on, the prior, $p(z)$, is fixed to an isotropic Gaussian $p(z) = \mathcal{N}(z; 0, I)$, though there are various extensions that instead employ hierarchical latent spaces [54] or learn the prior itself during training [55]. Note that each of the prior, encoder, and decoder are defined on for a single arbitrary datapoint: because of the factorization assumption the generative model is the same for all datapoints, while there are no *stochastic* global parameters. Thus there are no probabilistic interactions between different datapoints x_i and x_j, or their corresponding latents z_i and z_j. We, therefore, define our generative model as $p(z)p_\theta(x|z)$ and use $\mathbf{X}$ as a set of example datapoints for training θ and ϕ.

While the role of the decoder is self–apparent through its inclusion in the generative model itself, the encoder can be viewed in a number of ways. The viewpoint most in line with the concepts we have introduce thus far is that it is a functional approximation of the posterior $p_\theta(z|x)$.[5] That is, rather than individually learning a variational approximation for each datapoint, we learn a *mapping* from datapoints to a variational approximation for that datapoint. In other words, we learn a conditional distribution on z given x.

This process is known as **amortised inference**. The word amortised comes, via the probabilistic programming literature, from the compilers literature, where amortisation refers to pre-computing functions at compile time so that function evaluations can be performed efficiently at run time. In the fixed model setting, amortised inference uses a function approximator which is learnt at 'training time,' so that inference for new data items at 'test time' (i.e. when dealing with a particular datapoint or dataset depending on context) can be performed efficiently using the function approximator. This is as opposed to iterative inference procedures, e.g. Markov chain Monte Carlo or coordinate ascent variational inference, which can be much more expensive at test time. In the context of VAEs, this amortised approximation is learned simultaneously to the generative model, and allows us to share information across different datapoints, rather than requiring separate inference schemes/variational parameters for each.

Both the decoder and the encoder are former by using a deep neural network to map from their respective inputs to distribution parameters in their corresponding output spaces. For example, a common choice for both is a Gaussian with diagonal covariance for which we have

$$p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \sigma^2_\theta(z)) \tag{9.27}$$

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x)) \tag{9.28}$$

[5] It is beyond the scope of this course to properly explore the plethora of subtleties surrounding VAE encoders, and indeed VAEs more generally, with these still being a highly active area of research. However, it is important to be aware of the fact that this viewpoint of the encoder as simply an approximation of the decoder posterior is actually somewhat limited: the way VAEs are trained means that the encoder influences the learning of the decoder just as much as the decoder influences the learning of the encoder. As such, the two are always intertwined and in many applications, such as representation learning, the encoder is actually the primary entity of interest, rather than something introduced simply to aid with the training of the generative model. Moreover, inductive biases in the encoder design, such as the standard choice of a diagonal covariance, can heavily influence the models learned.

where $\mu_\theta(z), \sigma^2_\theta(z)$ are outputs of the decoder deep neural network with parameters θ (where $\sigma^2_\theta(z)$ is a diagonal covariance matrix with zeros off the main diagonal) and analogously $\mu_\phi(x), \sigma^2_\phi(x)$ are the outputs of the encoder network with parameters ϕ. The distributional form of the decoder naturally changes with the type of data used, with, for example, Bernoulli distributions also common. However, it is generally important that the *form* of this distribution remains simple (but with the parameter mappings themselves complex) to ensure that the information about the generations is stored in the latents, rather than the randomness in the generation of x given $\mathbb{E}_{X\sim p_\theta(x|z=z_i)}[X]$. In short, the variability in x for a given z should effectively correspond to noise regardless of the form of the decoder (see e.g. `http://ruishu.io/2017/01/14/one-bit/`).

More complicated forms for the encoder distribution can also be helpful. For example, some approaches augment VAE posterior approximations by transforming drawn samples through mappings (so called *flows*) with additional trainable parameters to achieve richer variational families [42].

9.6.1 Revisiting the ELBO

Training in VAEs is done by simultaneously optimizing the ELBO with respect to both θ and ϕ using a stochastic gradient approach. This is based on exploiting the factorization assumption over datapoints to employ mini–batching. Namely, we have

$$\mathcal{L}(\mathbf{X}, \theta, \phi) := \sum_{i=1}^{n} \mathcal{L}(x_i, \theta, \phi) = \sum_{i=1}^{n} \mathbb{E}_{q_\phi(z_i|x_i)}\left[\log \frac{p_\theta(x_i, z_i)}{q_\phi(z_i|x_i)}\right] \leq \log p_\theta(\mathbf{X}) \tag{9.29}$$

where each $\mathcal{L}(x_i, \theta, \phi) \leq \log p_\theta(x_i)$ is the ELBO for a single datapoint and $p_\theta(x) = \mathbb{E}_{p(z)}[p_\theta(x|z)]$. Here we can approximate the sum using a mini-batch B in standard way, that is

$$\nabla_{\theta,\phi}\mathcal{L}(\mathbf{X}, \theta, \phi) \approx \frac{n}{|B|}\sum_{i\in B} \nabla_{\theta,\phi}\mathbb{E}_{q_\phi(z_i|x_i)}\left[\log \frac{p_\theta(x_i, z_i)}{q_\phi(z_i|x_i)}\right].$$

Dealing with the expectation can be dealt with using (typically single sample) Monte Carlo estimates, but requires some care for the gradients of ϕ, as discussed later, due to its presence in the distribution the expectation is taken with respect to. Note that, as opposed to the EM algorithm, which has two separate steps for inference and learning, we will calculate unbiased derivative estimates with respect to θ and ϕ at the same time with the same mini-batch / latent samples, and then update both in the same step.

Consider now different ways in which we can write the ELBO for a single x:

$$\begin{aligned}
\mathcal{L}(x, \theta, \phi) &= \log p_\theta(x) - \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x)) \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x, z)}{q_\phi(z|x)}\right] \qquad (9.30) \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(z)}{q_\phi(z|x)}\right] \\
&= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathbb{D}_{\mathrm{KL}}\left(q_\phi(z|x)\|p(z)\right). \qquad (9.31)
\end{aligned}$$

The final expression is particularly convenient as for a standard choices of a diagonal Gaussian encoder (as per (9.28)) and a standard normal prior $p(z)$, the KL divergence is available in closed form:

$$\begin{aligned}\mathbb{D}_{\mathrm{KL}}\left(q_\phi(z|x)||p(z)\right) &= \frac{1}{2}\left[\mu_\phi(x)^\top\mu_\phi(x) + \mathrm{tr}\left(\Sigma_\phi(x)\right) - \log\det\left(\Sigma_\phi(x)\right) - k\right] \\ &= \frac{1}{2}\sum_{j=1}^{k}\left[\mu_{\phi,j}^2(x) + \sigma_{\phi,j}^2(x) - \log\left(\sigma_{\phi,j}^2(x)\right) - 1\right]. \end{aligned} \tag{9.32}$$

When this assumption does not hold, we can simply use the form of the ELBO in (9.30) instead, noting though that this tends to be slightly higher variance.

The formulation of (9.31) is also insightful from the perspective of the VAE itself. Namely, it shows how we can view a VAE as a stochastic **auto–encoder** as discussed in the lectures. In short, $\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$ forms a **reconstruction loss** that measures how effectively the original input is preserved when passed through the encoder and then back through the decoder, while $\mathbb{D}_{\mathrm{KL}}\left(q_\phi(z|x)\|p(z)\right)$ acts a regulariser that increases the entropy of the encoding process to enforce smoothness in the space of z. This alternative viewpoint of VAEs is extremely important as it shows how VAEs can used for **representation learning**, wherein z can be viewed as a compressed feature representation of x. This can then be used in downstream prediction tasks, or as mechanism for interpreting or manipulating individual datapoints. See the lectures for more in depth discussion.

9.6.2 The Reparametrization Trick

In order to optimize the ELBO objective over θ and ϕ, it remains to unbiasedly estimate the $\nabla_{\theta,\phi}\mathcal{L}(x_i,\theta,\phi)$ terms.

For θ, this can be straightforwardly achieved by drawing Monte Carlo samples from $q_\phi\left(z|x_i\right)$. Typically only a single such sample is drawn in practice for each datapoint, on the basis that it is almost always better to increase the size of the mini–batch to reduce the variance, rather than drawing additional samples of z for each x_i. We can see this by noting that such an approach is essentially equivalent to drawing $|B|$ samples from the joint $p_{\mathrm{data}}(x)q_\phi(z|x)$ (where $p_{\mathrm{data}}(x)$ is the empirical data distribution).

For ϕ, such an approach would result in a biased estimator and thus cannot be used directly. This is because ϕ affects the distribution the expectation is taken with respect to itself. Thankfully, there is a simple solution to this known as the **reparametrization trick**, or sometimes as the **pathwise estimator**. This involves expressing the distribution $q_\phi(z|x)$ it the form of a base random draw $\epsilon \sim q(\epsilon)$ (typically just $\mathcal{N}(0,I)$) and a deterministic mapping $z = g(\epsilon,x,\phi)$. This then allows us to write $\mathcal{L}(x_i,\theta,\phi)$ as an expectation over ϵ, which is itself independent of ϕ, taking care to note that z itself now depends on ϕ. Namely we have

$$\nabla_\phi\mathbb{E}_{q_\phi(z_i|x_i)}\left[\log\left(\frac{p_\theta(x_i,z_i)}{q_\phi(z_i|x_i)}\right)\right] = \mathbb{E}_{q(\epsilon)}\left[\nabla_\phi\log\left(\frac{p_\theta(x_i,g(\epsilon_i,x_i,\phi))}{q_\phi(g(\epsilon_i,x_i,\phi)|x_i)}\right)\right] \tag{9.33}$$

where we can sample $\epsilon_i \sim \mathcal{N}(0, I)$ and then calculate the required derivatives using automatic differentiation. Critically, the resulting stochastic gradients will be unbiased estimates of the true gradients.

To give a concrete example, in the case of a diagonal Gaussian encoder, a draw $z_i \sim \mathcal{N}\left(z; \mu_\phi(x), \sigma^2_\phi(x)\right)$ can be written as $z_i = \mu_\phi(x) + \sigma_\phi(x)\epsilon_i$, with $\epsilon_i \sim \mathcal{N}(0, I)$, and the multiplication with $\sigma_\phi(x)$ being element–wise.

9.6.3 Score Function Estimator

Unfortunately, the reparameterization trick can only be applied if $q_\phi(z|x)$ is continuous with respect to z as otherwise applying the chain rule to (9.33) requires us to back-propagate our derivatives through z. This essentially means that we cannot use the reparametrization trick whenever the z are discrete.

On the plus side, there is an alternative approach to obtain unbiased estimates of gradients with respect to variational parameters, termed the **score function gradient** or **REINFORCE gradient**, that can still operate in such settings. On the downside, this estimator tends to have much higher variance than the reparametrized gradient and should thus be avoided if possible. Variance reduction techniques, like control variates, can be helpful in cases where the approach is unavoidable.

To derive this estimator, first note the following result, known as the **score identity**,

$$\mathbb{E}_{q_\phi(z)}\left[\nabla_\phi \log q_\phi(z)\right] = \int q_\phi(z) \nabla_\phi \log q_\phi(z)\, dz = \int \nabla_\phi q_\phi(z)\, dz = \nabla_\phi \int q_\phi(z)\, dz = 0,$$

which critically still also holds when dz is a counting measure. The score function gradient can now be derived as follows

$$\begin{aligned}
\nabla_\phi \mathcal{L} &= \nabla_\phi \int q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz \\
&= \int (\nabla_\phi q_\phi(z|x)) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz + \underbrace{\int q_\phi(z|x) \left(\nabla_\phi \log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right) dz}_{=0}
\end{aligned}$$

where the second term being zero follows by the score identity and lack of dependence of $p_\theta(x, z)$ on ϕ, and so also noting $\nabla x = x \nabla \log x$, we have

$$= \mathbb{E}_{q_\phi(z|x)}\left[(\nabla_\phi \log q_\phi(z|x)) \log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right].$$

This is now something where we can directly construct Monte Carlo estimates, calculating the required derivatives through automatic differentiation (note that here z is not a function of ϕ as it was for the reparametrized estimator).

9.6.4 Alternative Bounds

Assume that we have access to some strictly *positive* and *unbiased* estimator $\hat{p}_\theta(x)$ of $p_\theta(x)$, that is $\mathbb{E}[\hat{p}_\theta(x)] = p_\theta(x)$ and $\hat{p}_\theta(x) > 0$. Jensen's inequality then tells us that

$$\mathbb{E}[\log \hat{p}_\theta(x)] \leq \log \mathbb{E}[\hat{p}_\theta(x)] = \log p_\theta(x)$$

such that the expectation of the logarithm of our estimator is a lower bound on the evidence. The VAE framework we have described now simply correspond to the special case of $\hat{p}_\theta(x) = p_\theta(x,z)/q_\phi(z|x)$, which is the importance weight if $q_\phi(z|x)$ is the proposal distribution and the target is the posterior $p_\theta(z|x)$.

One can thus naturally consider other types of estimators. Essentially, any inference method that produces an unbiased marginal likelihood estimator will produce a valid lower bound. Examples of suitable such estimators are importance sampling, annealed importance sampling, and sequential Monte Carlo. For example, the former of these is constructing by independently sampling from the encoder K time, $z_k \overset{iid}{\sim} q_\phi(\cdot|x)$, and then constructing the bound

$$\mathcal{L}_K(x,\theta,\phi) = \mathbb{E}_{\prod_{k=1}^K q_\phi(z_k|x)}\left[\log\left(\frac{1}{K}\sum_{k=1}^K \frac{p_\theta(x,z_k)}{q_\phi(z_k|x)}\right)\right] \tag{9.34}$$

In general, the use of alternative marginal likelihood estimators like this allow one to achieve *tighter* variational bounds than the standard ELBO, which thus more accurately represent the true model evidence.

Another important variation in the standard ELBO is given by the so-called β–VAE [1, 25]. This tries to directly control the level of regularization in the ELBO by introducing a scaling factor β as follows

$$\mathcal{L}_\beta(x,\theta,\phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \beta\,\mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)\|p(z)).$$

Higher values of β correspond to larger degrees of regularization and tend to produce smoother latent spaces (i.e. models were small changes in z lead to small changes in $p_\theta(x|z)$), at the expense of a drop in reconstruction quality and typical generation fidelity. It is still a valid lower bound on the evidence for $\beta \geq 1$, though $0 < \beta < 1$ is also sometimes useful in practice if the amount of regularization in the original ELBO is deemed too large. However, the bound no longer becomes tight if $q_\phi(z|x) = p_\theta(z|x)$ for any β other than $\beta = 1$. Numerous interesting results have been shown for the β–VAE, such as the fact that the value of β is intricately linked to the smoothness of the latent space which is learned [38] and also the amount of information lost in the encoding–decoding process, with such an *information bottleneck* allowing one the encoding of only important information [1].