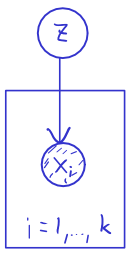
Variational Inference

Lecture 8 (23 June 2022); lecturer: Robert Bamler
more course materials online at https://robamler.github.io/teaching/compress22/

Recap from last lecture: Bits-Back Coding With Latent Variable Model

Model:



"latent variables"   message

$$P(\underline{z}, \underline{x}) = \underbrace{P(\underline{z})}_{\text{prior}} \underbrace{P(\underline{x} \mid \underline{z})}_{\text{likelihood}}$$

$\left(\text{here: } P(\underline{x} \mid \underline{z}) = \prod_{i=1}^{k} P(x_i \mid \underline{z})\right)$

Note: we want to compress $\underline{x}$ with marginal model $P(\underline{x}) = \sum_{\underline{z}} P(\underline{x}, \underline{z})$

Bits-Back Algorithm (encoder):

1) z ← decode with ANS from some existing bit string using model $\underbrace{P(\underline{z} \mid \underline{x} = \underline{x})}_{\text{posterior}}$

2) encode message $\underline{x}$ using ANS and model $P(\underline{x} \mid \underline{z} = \underline{z})$

3) encode $\underline{z}$ using ANS & prior model $P(\underline{z})$

Net Bit Rate: $R_{net}(\underline{x}) = -\log_2 P(\underline{x} = \underline{x} \mid \underline{z} = \underline{z}) - \log_2 P(\underline{z} = \underline{z}) - \left(-\log_2 P(\underline{z} = \underline{z} \mid \underline{x} = \underline{x})\right)$

$$= -\log_2 \frac{\cancel{P(\underline{z} = \underline{z}, \underline{x} = \underline{x})} \; \cancel{P(\underline{z} = \underline{z})} \; P(\underline{x} = \underline{x})}{\cancel{P(\underline{z} = \underline{z})} \; \cancel{P(\underline{z} = \underline{z}, \underline{x} = \underline{x})}} = -\log_2 P(\underline{x} = \underline{x})$$

⇒ optimal

⇒ justifies our use of the posterior in step 1

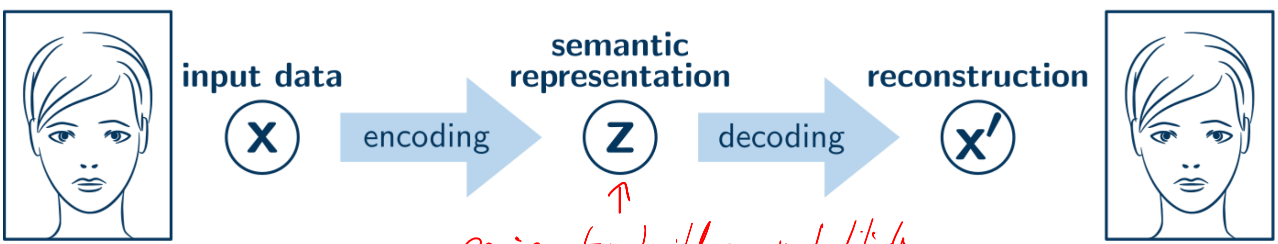
Today: Variational Inference

→ comes from a completely different field of research, unrelated to data compression;

but:
→ crucial method in modern machine-learning based data compression;
→ the precise formalism of VI can be motivated most naturally by minimizing the net bit rate of bits-back coding.

Spoiler: Variational Autoencoders

→ popular class of so-called "deep generative models" (use deep neural networks to generate data)
→ idea: rather than building a probabilistic model over a complicated message space (e.g., the space of all HD images), design a mapping between the message space and a more abstract semantic representation space and build a probabilistic model over the semantic representation space.



easier to build a probabilistic model for z than for x

Back to bits-back coding:

$$R_{net}(\underline{x}) = -\log_2 P(\underline{X}=\underline{x} \mid Z=z) - \log_2 P(Z=z) - \left(-\log_2 P(Z=z \mid \underline{X}=\underline{x})\right)$$

$$= -\log_2 \frac{\cancel{P(Z=z,\underline{X}=\underline{x})}}{\cancel{P(Z=z)}} \, \frac{\cancel{P(Z=z)}\; P(\underline{X}=\underline{x})}{\cancel{P(Z=z,\underline{X}=\underline{x})}} = -\log_2 P(\underline{X}=\underline{x})$$

Problem: obtaining the true posterior is computationally impossible in all but very special models:

$$P(Z \mid \underline{X}=\underline{x}) = \frac{P(Z,\underline{X}=\underline{x})}{P(\underline{X}=\underline{x})} = \begin{cases} \dfrac{P(Z,\underline{X}=\underline{x})}{\sum_z P(Z=z,\underline{X}=\underline{x})} \\[2em] \dfrac{P(Z,\underline{X}=\underline{x})}{\int P(Z=z,\underline{X}=\underline{x})\,dz} \end{cases}$$

high-dimensional integral
(exponentially expensive (without tricks))

Idea 1: what if we simply don't use the posterior P(Z | X=x), but instead some other distribution Q(Z | X=x)?

$$\tilde{R}^{(z)}_{net}(\underline{x}) = -\log_2 P(\underline{X}=\underline{x} \mid Z=z) - \log_2 P(Z=z) - \left(-\log_2 Q(Z \mid \underline{X}=\underline{x})\right)$$

$$= -\log_2 \frac{\cancel{P(Z=z,\underline{X}=\underline{x})}}{\cancel{P(Z=z)}}\,\cancel{P(Z=z)} + \log_2 Q(Z=z \mid \underline{X}=\underline{x}) \overset{\text{in general}}{\neq} -\log_2 P(\underline{X}=\underline{x})$$

→ depends on z, which is out of our control

Recall: if Q(Z | X=x) = P(Z | X=x), then the net bit rate is independent of z and optimal.

⇒ for any other $Q(Z \mid \underline{X}=\underline{x}) \neq P(Z \mid \underline{X}=\underline{x})$: expected net bit rate is larger

$$\mathbb{E}_{Q(Z \mid \underline{X}=\underline{x})}\left[\tilde{R}^{(z)}_{net}(\underline{x})\right] \geq R_{net}(\underline{x}) = -\log_2 P(\underline{X}=\underline{x})$$

→ Problem 8.1 (b): proof of

$$\mathbb{E}_{Q(Z \mid \underline{X}=\underline{x})}\left[\tilde{R}^{(z)}_{net}(\underline{x})\right] = \underbrace{-\log_2 P(\underline{X}=\underline{x})}_{\text{optimal}} + \underbrace{D_{KL}\left(Q(Z \mid \underline{X}=\underline{x}) \,\|\, P(Z \mid \underline{X}=\underline{x})\right)}_{\text{overhead} \geq 0}$$

(expectations here only over z)

Idea 2: optimize the expected net bit rate over various Q(Z | X=x)

→ parameterize $Q_\phi(Z \mid \underline{X}=\underline{x})$ by some parameters ϕ

$$\Rightarrow \boxed{\text{minimize } \mathbb{E}_{Q_\phi(Z \mid \underline{X}=\underline{x})}\left[\tilde{R}^{(z)}_{net}(\underline{x})\right] \text{ over } \phi}$$

Question: what is the distribution of z in our modified bits-back algorithm?

1) $z \leftarrow$ decode with ANS from some existing bit string using model $Q(z \mid \underline{X} = \underline{x})$ $\Rightarrow$ claim: thus z is distributed $Q(z \mid \underline{X} = \underline{x})$

$\rightarrow$ Problem Set 9: Prove that decoding a uniformly distributed random bit string with same model $Q(z \mid \underline{X} = \underline{x})$ $\Leftrightarrow$ drawing $z \sim Q(z \mid \underline{X} = \underline{x})$

For historic reasons, one typically talks about maximizing the negative expected net bit rate instead. This is called the Evidence Lower BOund (ELBO):

$$\mathrm{ELBO}(\phi) = -\mathbb{E}_{Q_\phi(z \mid \underline{X} = \underline{x})}\left[\tilde{R}^{(z)}_{net}(\underline{x})\right]$$

$$= \mathbb{E}_{Q_\phi(z \mid \underline{X} = \underline{x})}\left[\log P(z, \underline{X} = \underline{x}) - \log Q_\phi(z \mid \underline{X} = \underline{x})\right]$$

Problem 8.1 (b):

$$\underbrace{\mathrm{ELBO}(\phi)}_{\text{lower bound on the evidence}} = \underbrace{\log P(\underline{X} = \underline{x})}_{\text{"evidence"}} \underbrace{- D_{KL}\big(Q_\phi(z \mid \underline{X} = \underline{x}) \,\|\, P(z \mid \underline{X} = \underline{x})\big)}_{\leq 0}$$

Thus, the following three are equivalent:

use $Q_\phi(z \mid \underline{X} = \underline{x})$ instead of posterior

$\boxed{\text{minimizing the expected net bit rate of our modified bits-back algorithm}}$

$\Updownarrow$

$\boxed{\text{maximizing the ELBO}}$

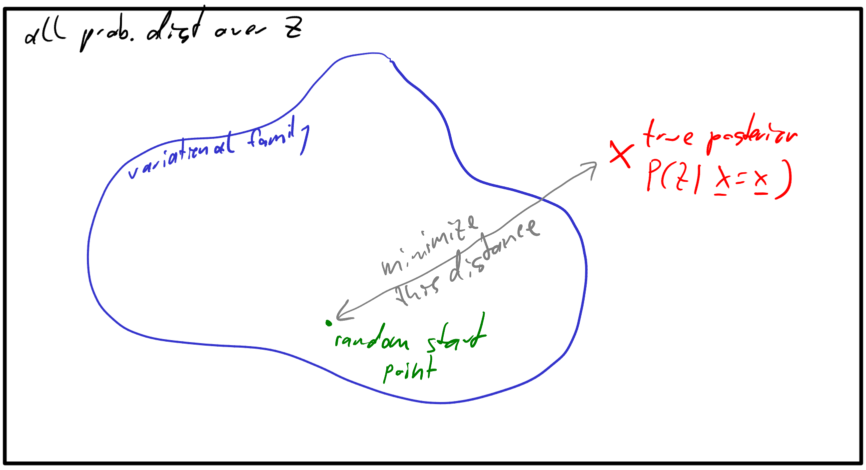
$\Updownarrow$

$\boxed{\text{minimizing } D_{KL}\big(Q_\phi(z \mid \underline{X} = \underline{x}) \,\|\, P(z \mid \underline{X} = \underline{x})\big)}$ $\Rightarrow$ result in an approximate posterior $Q_{\phi^*}(z \mid \underline{X} = \underline{x}) \approx P(z \mid \underline{X} = \underline{x})$

$\phi^* := \arg\max_\phi \mathrm{ELBO}(\phi)$

How Can We Maximize The ELBO?

1) Choosing a Variational Family $= \{Q_\phi(Z|X=x)\}_\phi$



In practice: often $Z \in \mathbb{R}^d$

$$\boxed{Q_\phi(Z|X=x) = \prod_{i=1}^{d} Q_{\phi_i}(Z_i|X=x)}$$

where $Q_{\phi_i}(Z_i|X=x)$ is, for example, a normal distribution with som mean μ_i and std. deviation σ_i ← $\phi_i \equiv (\mu_i, \sigma_i)$

This is called the "mean field approximation" due to an analogy to physics.

2) Performing the Maximization

Three methods:

→ "coordinate ascent variational inference (CAVI)": fastest optimization algorithm, but only possible in special models (mostly so-called "conditional conjugate" models; see references)

details: Problem 8.3

→ "reparameterization gradients": very simple in practice and relatively widely applicable, but not possible for all variational distributions Q (in particular, not for discrete Q)

→ "score function gradients" = "REINFORCE method": works also in some cases where reparameterization gradients don't work, but typically slower in practice unless additional tricks are used.

$$ELBO(\phi) = \mathbb{E}_{Q_\phi(Z|X=x)}\left[\log P(Z, X=x) - \log Q_\phi(Z|X=x)\right]$$

Goal: find $\phi^* := \arg\max_\phi ELBO(\phi)$

→ stochastic gradient descent

i.e., estimate $\mathbb{E}_{Z\sim Q}[\dots]$ by sampling

$g := \nabla_\phi ELBO(\phi)$ $\quad \delta > 0$

update $\phi \leftarrow \phi + \delta g$

repeat

Problem: we have to estimate the gradient $\boxed{\nabla_\phi ELBO(\phi) = \nabla_\phi \mathbb{E}_{Z\sim Q_\phi}[\dots]}$

this dependency on ϕ makes estimation by samples tricky

Solutions: Problem 8.2