Lecture 8, Part 1:

Variational Inference

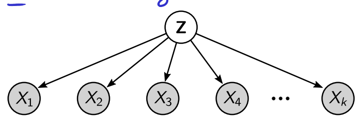
Robert Bamler · Summer Term of 2023

These slides are part of the course *"Data Compression With and Without Deep Probabilistic Models"* taught at University of Tübingen. More course materials—including video recordings, lecture notes, and problem sets with solutions—are publicly available at `https://robamler.github.io/teaching/compress23/`.

Recall: Bits-Back Coding & Latent Variable Models

assume now that $\underline{Z}$ is also high dimensional

- **Latent variable model:** $P(\mathbf{Z},\mathbf{X}) = P(\mathbf{Z})\prod_{i=1}^{k} P(X_i\,|\,\mathbf{Z})$



- **Encoding** a message $\mathbf{x}$ & side information $\mathbf{s} \in \{0,1\}^*$:
  1. $\mathbf{z} \leftarrow$ decode from $\mathbf{s}$ with ANS using posterior $P(\mathbf{Z}\,|\,\mathbf{X}{=}\mathbf{x})$. ← consumes $-\log_2 P(\underline{Z}=\underline{z}\,|\,\underline{X}=\underline{x})$ bits
  2. Encode $\mathbf{x}$ using likelihood $P(\mathbf{X}\,|\,\mathbf{Z}{=}\mathbf{z})$. ← produces $-\log_2 P(\underline{X}=\underline{x}\,|\,\underline{Z}=\underline{z})$ bits
  3. Encode $\mathbf{z}$ using prior $P(\mathbf{Z})$. ← produces $-\log_2 P(\underline{Z}=\underline{z})$ bits
- **Net bit rate:** $R^{\text{net}}(\mathbf{x}\,|\,\mathbf{s}) = -\log_2 P(\underline{X}=\underline{x}\,|\,\underline{Z}=\underline{z}) - \log_2 P(\underline{Z}=\underline{z}) + \log_2 P(\underline{X}=\underline{x}\,|\,\underline{Z}=\underline{z}) = -\log_2 P(\underline{X}=\underline{x})$
  - optimal & independent of $\mathbf{s}$ $\implies$ justifies our use of posterior $P(\mathbf{Z}\,|\,\mathbf{X}{=}\mathbf{x})$ in step 1.
- **Problem:** calculating the posterior ("Bayesian inference") is often infeasible: computational cost: $O(\exp(d))$ ← dimensionality of $\underline{Z}$
  - $P(\mathbf{Z}\,|\,\mathbf{X}{=}\mathbf{x}) = \dfrac{P(\mathbf{Z},\mathbf{X}{=}\mathbf{x})}{P(\mathbf{X}{=}\mathbf{x})}$ where $\underbrace{P(\mathbf{X}{=}\mathbf{x})}_{-\log_2 P(\underline{X}=\underline{x}) = \text{"evidence"}} = \begin{cases}\sum_{\mathbf{z}} P(\mathbf{Z}{=}\mathbf{z},\mathbf{X}{=}\mathbf{x}) & \text{if } \mathbf{Z} \text{ is discrete;}\\ \int p(\mathbf{z},\mathbf{x})\,\mathrm{d}\mathbf{z} & \text{if } \mathbf{Z} \text{ is continuous.}\end{cases}$

Variational Inference

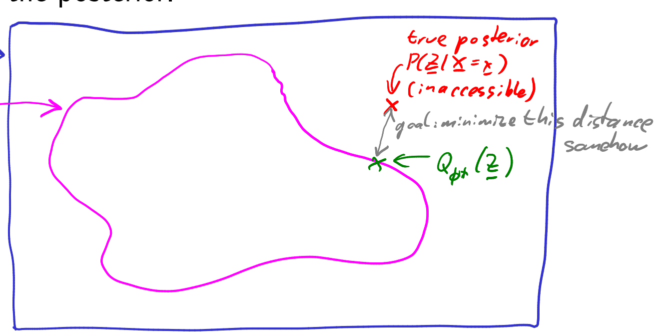
- **Problem:** calculating the posterior $P(\mathbf{Z}\,|\,\mathbf{X}{=}\mathbf{x})$ is often computationally infeasible.
- **Idea:** use a different distribution $Q_\phi(\mathbf{Z})$ instead of the posterior.
  1. Consider the space of *all* probability distributions over $\mathbf{Z}$.
  2. Choose a subspace $\mathcal{Q}$ of "simple" distributions.

     → e.g., if $\mathbf{Z} \in \mathbb{R}^d$: choose $\mathcal{Q} := \{Q_{\boldsymbol{\mu},\boldsymbol{\sigma}}\}_{\boldsymbol{\mu},\boldsymbol{\sigma}\in\mathbb{R}^d}$

     with PDF $q_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\mathbf{z}) = \prod_{i=1}^{d} \mathcal{N}(z_i;\mu_i,\sigma_i^2)$

     → general notation: $Q_\phi(Z)$

     "variational distribution" "variational parameters"
  3. Find *optimal* variational parameters $\boxed{\phi^*}$ such that $Q_{\phi^*}(Z) \approx P(Z\,|\,\mathbf{X}{=}\mathbf{x})$ for a given message $\mathbf{x}$.



Bits-Back Coding With an *Approximate* Posterior

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- **Encoding** a message $\mathbf{x}$ & side information $\mathbf{s} \in \{0,1\}^*$ (copied from first slide):
  1. $\mathbf{z} \leftarrow$ decode from $\mathbf{s}$ with ANS using posterior $\cancel{P(\mathbf{Z}\,|\,\mathbf{X}=\mathbf{x})}$. $Q_\phi(\underline{z})$ ← consumes $-\log_2 Q_\phi(\underline{z}=\underline{z})$ bits
  2. Encode $\mathbf{x}$ using likelihood $P(\mathbf{X}\,|\,\mathbf{Z}=\mathbf{z})$.
  3. Encode $\mathbf{z}$ using prior $P(\mathbf{Z})$.

  } produces $-\log_2 P(\underline{Z}=\underline{z}, \underline{X}=\underline{x})$ bits

- **Net bit rate:** $\boxed{R_\phi^{\text{net}}(\mathbf{x}\,|\,\mathbf{s}) = -\log_2 P(\mathbf{Z}=\mathbf{z}, \mathbf{X}=\mathbf{x}) + \log_2 Q_\phi(\mathbf{Z}=\mathbf{z})}$
  - **Naive idea:** find $\phi^* := \arg\min_\phi R_\phi^{\text{net}}(\mathbf{x}\,|\,\mathbf{s})$; then run above encoder using model $Q_{\phi^*}(\mathbf{Z})$ in step 1.

    ϕ^* depends on both $\underline{x}$ and $\underline{s}$ here...

- **Decoding:**
  1. $\mathbf{z} \leftarrow$ decode using prior $P(\mathbf{Z})$.
  2. $\mathbf{x} \leftarrow$ decode using likelihood $P(\mathbf{X}\,|\,\mathbf{Z}=\mathbf{z})$.
  3. Encode $\mathbf{z}$ using approximate posterior $Q_{\phi^*}(\mathbf{Z})$ to reconstruct side information $\mathbf{s}$.

  Problem: ϕ^* depends on $\mathbf{s}$.
  $\Longrightarrow$ cyclic dependency

  decoder has to use same ϕ^* that encoder used, but that one depended on $\underline{s}$, which the decoder only gets once it knows ϕ^*.

Expected Net Bit Rate

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- **Idea:** minimize the *expected* net bit rate for random $\mathbf{s}$ (but still for a fixed message $\mathbf{x}$):

$$\phi^* := \arg\min_\phi \Big(\mathbb{E}_{\mathbf{s}}\big[R_\phi^{\text{net}}(\mathbf{x}\,|\,\mathbf{s})\big]\Big) = \arg\min_\phi \Big(\mathbb{E}_{\mathbf{s}}\big[-\log_2 P(\mathbf{Z}=\mathbf{z}, \mathbf{X}=\mathbf{x}) + \log_2 Q_\phi(\mathbf{Z}=\mathbf{z})\big]\Big)$$

- What is the distribution of the bit string $\mathbf{s}$?
  - Generic argument: we have no idea $\Longrightarrow$ uniform distribution

    If we regard probabilistic models as (subjective) expressions of our lack of knowledge.
  - Example: assume $\mathbf{s}$ is the compressed representation of some previously encoded data.

    $\Rightarrow \underline{s}$ can't be further compressed $\Rightarrow \underline{s}$ has information content $|\underline{s}|$, i.e., all bits s_i are independent and have information content $-\log_2 P(S_i=s_i) = 1$ bit $\Rightarrow P(S_i=s_i) = \frac{1}{2}$ $\forall i$, $s_i \in \{0,1\}$
- What distribution does this induce for $\mathbf{z}$?
  - Assume decoding with model $Q_\phi(\mathbf{Z})$ results in a value $\mathbf{z}$. $\Rightarrow$ consumes $-\log_2 Q_\phi(\underline{Z}=\underline{z})$ bits from $\underline{s}$
  - For an optimal coder, only 1 bit string corresponds to $\mathbf{z}$.
    $\Rightarrow$ probability that each one of the $-\log_2 Q_\phi(\mathbf{Z}=\mathbf{z})$ consumed bits matches: $\left(\frac{1}{2}\right)^{-\log_2 Q_\phi(\underline{Z}=\underline{z})} = Q_\phi(\underline{Z}=\underline{z})$

Decoding from a uniform random bit string with a code that is optimal for some model probabilistic model	$\Longleftrightarrow$	sampling from the same probabilistic model

Evidence Lower Bound (ELBO)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Minimize the *expected net bit rate* for encoding a given message $\mathbf{x}$:

$$\phi^* := \arg\min_\phi \underbrace{\mathbb{E}_{\mathbf{s}}\big[R_\phi^{\text{net}}(\mathbf{x}\,|\,\mathbf{s})\big]}_{\text{see last slide}} = \arg\min_\phi \boxed{\mathbb{E}_{Q_\phi(\mathbf{Z})}\big[-\log_2 P(\mathbf{Z}, \mathbf{X}=\mathbf{x}) + \log_2 Q_\phi(\mathbf{Z})\big]}$$

same except for global sign

- In the (non-compression) literature, one typically *maximizes* the *negative* net bit rate:

$$\boxed{\phi^* = \arg\max_\phi \text{ELBO}(\phi, \mathbf{x}) \quad \text{where} \quad \text{ELBO}(\phi, \mathbf{x}) := \mathbb{E}_{Q_\phi(\mathbf{Z})}\big[\log P(\mathbf{Z}, \mathbf{X}=\mathbf{x}) - \log Q_\phi(\mathbf{Z})\big]}$$

- **Problem 8.1:** derive and interpret three equivalent expressions for the ELBO:
  - maximum a-posteriori (MAP) + entropy: $\boxed{\text{ELBO}(\phi, \mathbf{x}) = \mathbb{E}_{Q_\phi(\mathbf{Z})}\big[\log P(\mathbf{Z}, \mathbf{X}=\mathbf{x})\big] + H_{Q_\phi}[\mathbf{Z}]}$
  - regularized maximum likelihood: $\boxed{\text{ELBO}(\phi, \mathbf{x}) = \mathbb{E}_{Q_\phi(\mathbf{Z})}\big[\log P(\mathbf{X}=\mathbf{x}\,|\,\mathbf{Z})\big] - D_{\text{KL}}\big(Q_\phi(\mathbf{Z})\,\|\,P(\mathbf{Z})\big)}$
  - evidence lower bound: $\boxed{\text{ELBO}(\phi, \mathbf{x}) = \log P(\mathbf{X}=\mathbf{x}) - D_{\text{KL}}\big(Q_\phi(\mathbf{Z})\,\|\,P(\mathbf{Z}\,|\,\mathbf{X}=\mathbf{x})\big) \leq \log P(\mathbf{X}=\mathbf{x})}$

    $\Rightarrow Q_{\phi^*}(\mathbf{Z})$ minimizes KL-divergence from true posterior $\Rightarrow$ "variational inference" "evidence" "lower bound"

How Can We Maximize the ELBO?

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

$$\text{ELBO}(\phi, \mathbf{x}) := \mathbb{E}_{\mathbf{z}\sim Q_\phi(\mathbf{z})}\big[\ell(\phi, \mathbf{x}, \mathbf{z})\big] \quad \text{where} \quad \ell(\phi, \mathbf{x}, \mathbf{z}) = \log P(\mathbf{Z}=\mathbf{z}, \mathbf{X}=\mathbf{x}) - \log Q_\phi(\mathbf{Z}=\mathbf{z})$$

- **Goal:** find $\phi^* := \arg\max_\phi \text{ELBO}(\phi, \mathbf{x})$ (or at least an approximate maximum).
- **Gradient Descent:** (technically here: gradient *ascent*)
  1. initialize $\phi \leftarrow$ random
  2. repeat: ∘ calculate gradient $g = \nabla_\phi \text{ELBO}(\phi, \mathbf{x})$
     ∘ update $\phi \leftarrow \phi + \rho\, g$ with some "learning rate" $\rho > 0$

Problem: $\mathbb{E}_{Q_\phi(\mathbf{z})}[\ell(\phi, \mathbf{x}, \mathbf{z})]$ typically can't be calculated analytically (especially for deep probabilistic models, i.e., models parameterized by deep neural networks).

- ***Stochastic* Gradient Descent (SGD):** the naive way
  1. initialize $\phi \leftarrow$ random
  2. repeat: ∘ draw a random $\mathbf{z} \sim Q_\phi(\mathbf{Z})$
     ∘ calculate *gradient estimate* $\hat{g}(\mathbf{z}) = \nabla_\phi \ell(\phi, \mathbf{x}, \mathbf{z})$
     ∘ update $\phi \leftarrow \phi + \rho\, g$ with some (decaying) learning rate $\rho > 0$

Idea: replace analytic integration over $\mathbf{z}$ by sampling.

- **Problem:** SGD only works if $\hat{g}(\mathbf{z})$ is an *unbiased gradient estimate*: if $\mathbb{E}_{Q_\phi(\mathbf{Z})}\big[\hat{g}(\mathbf{Z})\big] = g$.
  - But we have: $\mathbb{E}_{Q_\phi(\mathbf{Z})}\big[\hat{g}(\mathbf{Z})\big] = \mathbb{E}_{Q_\phi(\mathbf{z})}\big[\nabla_\phi \ell(\phi, \mathbf{x}, \mathbf{z})\big]$

    and: $g = \nabla_\phi \mathbb{E}_{Q_\phi(\mathbf{z})}\big[\ell(\phi, \mathbf{x}, \mathbf{z})\big] = \nabla_\phi \sum_{\mathbf{z}} Q_\phi(\mathbf{Z}=\mathbf{z})\, \ell(\phi, \mathbf{x}, \mathbf{z}) \overset{\text{product rule}}{=} \underbrace{\sum_{\mathbf{z}} \big(\nabla_\phi Q_\phi(\mathbf{Z}=\mathbf{z})\big)\, \ell(\phi, \mathbf{x}, \mathbf{z})}_{\text{additional part that's not taken into account by our naive gradient estimates}} + \mathbb{E}_{Q_\phi(\mathbf{z})}\big[\hat{g}(\mathbf{z})\big]$

Black-Box Variational Inference (BBVI)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- **Problems 8.2 & 8.3:** derive 2 *unbiased* gradient estimators for variational inference:

  (8.2) **Reparameterization gradients:** [Kingma and Welling, 2014] e.g., scales & shifts Q_0 (see Problem 8.2.(a))

  In SGD, express $\mathbf{z} \sim Q_\phi(\mathbf{Z})$ as $\mathbf{z} = f(\epsilon, \phi)$ where $\epsilon \sim Q_0$ ← independent of ϕ

  $\Longrightarrow g = \nabla_\phi \mathbb{E}_{\mathbf{z}\sim Q_\phi(\mathbf{Z})}\big[\ell(\phi, \mathbf{x}, \mathbf{z})\big] = \nabla_\phi \mathbb{E}_{\epsilon\sim Q_0}\big[\ell(\phi, \mathbf{x}, f(\epsilon, \phi))\big] = \mathbb{E}_{\epsilon\sim Q_0}\big[\nabla_\phi \ell(\phi, \mathbf{x}, f(\epsilon, \phi))\big]$

  ☺ Typically low gradient variance $\mathbb{E}_{Q_\phi(\mathbf{Z})}\big[(\hat{g}(\mathbf{Z}) - g)^2\big] \Longrightarrow$ fast(-ish) convergence of SGD.

  ☹ Doesn't work for discrete $\mathbf{Z}$ (without an additional approximation called "Gumbel-softmax").

  (8.3) **Score function gradients** (aka "REINFORCE method"): [Ranganath et al., 2014]

  $\hat{g}(\mathbf{z}) = \big(\nabla_\phi \log Q_\phi(\mathbf{Z}=\mathbf{z})\big)\, \ell(\phi, \mathbf{x}, \mathbf{z})$ ← proof that this is an unbiased gradient estimate: Problem 8.3

  ☹ Typically higher gradient variance than reparameterization gradients.

  ☺ Works for all (explicit) variational families.

- **Remark:** for some models, $\mathbb{E}_{\mathbf{z}\sim Q_\phi(\mathbf{Z})}\big[\ell(\phi, \mathbf{x}, \mathbf{z})\big]$ can be calculated analytically.
  ⟶ *much* faster optimization with "coordinate ascent variational inference" (CAVI) [Blei et al. (2017)]

full references on problem set (also for other citations on this slide)

Outlook

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- **Next Week:**
  - How can we learn the generative model? → variational expectation maximization
  - How can we learn to do inference? → *amortized* variational inference
  - **Combined:** Variational Autoencoders (VAEs)
- **Afterwards:** lossy data compression (in theory & in practice)