Problem Set 9

published: 6 July 2022
discussion: 8 July 2022

Data Compression With And Without Deep Probabilistic Models
Prof. Robert Bamler, University of Tuebingen

Course materials available at https://robamler.github.io/teaching/compress22/

Problem 9.1: Simple Variational Autoencoder (VAE)

The accompanying jupyter notebook guides you through the implementation of a simple (toy) variational autoencoder. Follow the instructions in the notebook to complete the implementation.

Problem 9.2: Random Sampling by Decoding Random Bit Strings

In this problem, we show that decoding some message from a uniformly distributed random bit string with an entropy coder that is optimal for some probabilistic model is equivalent to drawing a random sample from the same probabilistic model.

As a reminder, this issue came up in the lecture on June 23 when we derived the connection between the (modified) bits-back coding algorithm and variational inference. The encoding process of our modified bits-back coding algorithm started with decoding z from some existing bit string, where the entropy model for the coder was the variational distribution $Q_\phi(Z \,|\, \mathbf{X} = \mathbf{x})$. Since the existing bit string was not under our control, we wanted to average over all possible bit strings, and we claimed that this averaging was equivalent to taking the expectation under $z \sim Q_\phi(Z \,|\, \mathbf{X} = \mathbf{x})$. You will provide the proof for this claim in this exercise.

Since the equivalence between sampling and decoding from a uniformly distributed random bit string holds in general and not just for bits-back coding, we won't use the letters z and Q here and we will instead follow our usual naming conventions and consider the case of decoding a message $\mathbf{x} \in \mathfrak{X}^k$ that is a sequence of k symbols $x_i \in \mathfrak{X}$ from some discrete alphabet $\mathfrak{X}$ using a model $P(\mathbf{X})$. For simplicity, we'll assume that $P(\mathbf{X})$ models the symbols as i.i.d., i.e., $P(\mathbf{X}) = \prod_{i=1}^{k} P(X_i)$ where $P(X_i)$ is the same probability distribution for all $i \in \{1, \ldots, k\}$.

(a) Let's first convince ourselves that the claim holds for the concrete case of decoding with Asymmetric Numeral Systems (ANS). Assume you have a string of statistically independent and uniformly distributed random bits, and you decode the first symbol x_1 from it using ANS with the model $P(X_1)$.

Recall how decoding with ANS works and argue why x_1 will be distributed (almost) as $x_1 \sim P(X_1)$ (with the only deviation coming from the fact that ANS approximates $P(X_1)$ in fixed point arithmetic). You may assume that the random

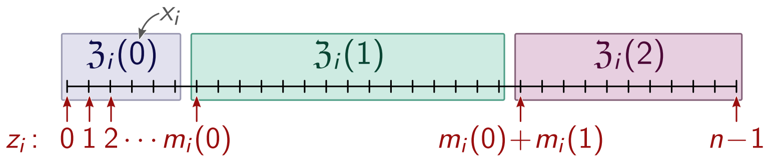

Figure 1: Reminder of how ANS works (for Problem 9.2 (a)).

bit string is at least `precision` bits long so that ANS doesn't run out of bits. You may want to refer to Figure 1 for your argument.

The equivalence between decoding from a random bit string and sampling from the employed entropy model is actually not just a special property of ANS but holds for all optimal entropy coders. Roughly speaking, the argument for this is that *encoding* symbols that are distributed according to the employed entropy model must result in a bit string of maximum entropy (i.e., independent and uniformly distributed bits) because otherwise the bit string could be further compressed and thus the coder is not optimal. Therefore, *decoding* from independent and uniformly distributed random bits must reverse the process and result in samples from the model. However, formalizing this argument is a bit more subtle because the length of the resulting bit string depends on the encoded symbols.

Let C be an encoder for symbols $x_i \in \mathfrak{X}$ that can append to some existing bit string. For concreteness, we'll assume that encoding and decoding operate with "stack" semantics (i.e., "last in first out", as in ANS). Thus, $C : (\{0,1\}^*, \mathfrak{X}) \to \{0,1\}^*$ is an injective function that maps some existing bit string $s \in \{0,1\}^*$ and a symbol $x_i \in \mathfrak{X}$ to a new bit string $C(s, x_i) \in \{0,1\}^*$. The decoding operation C^{-1} inverts this process and recovers both the encoded symbol x_i as well as the original bit string s, i.e., $C^{-1}(C(s, x_i)) = (s, x_i)$.

We further introduce the shorthands $\ell_{\text{min}} := \min_{x_i \in \mathfrak{X}}[-\log_2 P(X_i = x_i)]$ and $\ell_{\text{max}} := \max_{x_i \in \mathfrak{X}}[-\log_2 P(X_i = x_i)]$ for the minimum and maximum information content per symbol and we assume, for simplicity, that our model P has $\ell_{\text{min}} > 0$ and $\ell_{\text{max}} < 0$.

(b) Assume we are given some initial random bit string string S_0 with some fixed length $|S_0|$, where the bits are independent and uniformly distributed. We now use the coder C to decode some number k of symbols X_k from S. Since the bit string S is random, we have to treat the decoded symbols X_k also as random variables, and we denote the probability distribution that is induced by decoding from S as $P_{\text{dec}}(X_1, \ldots, X_k)$ to distinguish it from our model P.

In detail, we decode one symbol after the other:

$$\begin{aligned}
(S_1, X_1) &:= C^{-1}(S_0); \\
(S_2, X_2) &:= C^{-1}(S_1); \\
(S_3, X_3) &:= C^{-1}(S_2); \\
&\;\;\vdots \\
(S_k, X_k) &:= C^{-1}(S_{k-1}).
\end{aligned} \tag{1}$$

We assume that the coder C is an optimal stream code for the model $P(\mathbf{X}) = \prod_{i=1} P(X_i)$ in the sense that decoding some specific message $\mathbf{x} \in \mathfrak{X}^k$ consumes $-\log_2 P(\mathbf{X}{=}\mathbf{x}) + \varepsilon$ bits, where $\varepsilon \in [-\gamma, \gamma]$ with some constant γ takes into account that the stream code amortizes fractional information contents over multiple bits (e.g., in ANS, we have $\gamma = \texttt{precision}$). In particular, this means that

$$|S_0| - |S_k| > -\log_2 P(\mathbf{X}{=}\mathbf{x}) - \gamma \qquad \forall \mathbf{x} \in \mathfrak{X}^k \quad \text{with} \quad k \leq (|S_0| - \gamma)/\ell_{\max}. \quad (2)$$

where $|\cdot|$ denotes the length of a bit string and we assumed that the original bit string S_0 is long enough so that we don't run out of bits, i.e., $|S_0| \geq k\,\ell_{\max} + \gamma$.

Use Eq. 2 to show that[1]

$$P_{\text{dec}}(\mathbf{X}{=}\mathbf{x}) < 2^{\gamma+1} P(\mathbf{X}{=}\mathbf{x}) \qquad \forall \mathbf{x} \in \mathfrak{X}^k \quad \text{with} \quad k \leq (|S_0| - \gamma)/\ell_{\max}. \quad (3)$$

Hint: How many initial bit strings S_0 are there at most that decode to a given message $\mathbf{x}$ given that C^{-1} is injective, and how many total bit strings of the (fixed) length $|S_0|$ are there?

(c) Use Eq. 3 to derive an upper bound on the Kullback-Leibler (KL) divergence

$$\Delta_k := D_{\text{KL}}\big(P_{\text{dec}}(X_1, \ldots, X_k) \,\big|\big|\, P(X_1, \ldots, X_k)\big) \leq \text{const} \quad (4)$$

from the model $P(X_1, \ldots, X_k) = P(X_1)P(X_2)\cdots P(X_k)$ to the probability distribution $P_{\text{dec}}(X_1, \ldots, X_k)$ that is induced by decoding from the random bit string S_0.

Eq. 4 holds for all $k \leq (|S_0| - \gamma)/\ell_{\max}$ and the bound that you should find is a constant (independent of k). Therefore, you might already be convinced that the KL-divergence must actually be zero since, if it wasn't it should grow for growing k. The remaining parts of this problem formalize this argument.

(d) Show that

$$\Delta_k - \Delta_{k-1} = H\big(P_{\text{dec}}(X_k), P(X_k)\big) - H_{P_{\text{dec}}}(X_n \,|\, X_1, X_2, \ldots, X_{k-1}) \quad (5)$$

where, following our usual notation, the first term on the right-hand side denotes the cross entropy between the marginal distribution of X_k under P_{dec} and the marginal model distribution $P(X_k)$, and the second term on the right-hand side is the conditional entropy as defined in Problem 4.3 (c) on Problem Set 4.

Then show that

$$\Delta_k - \Delta_{k-1} \geq D_{\text{KL}}\big(P_{\text{dec}}(X_k) \,\big|\big|\, P(X_k)\big). \quad (6)$$

Hint: Use the fact that the conditional entropy is smaller or equal to the entropy (recall: the difference between the two is the mutual information, which is nonnegative).

[1] An earlier version of this problem set wrongfully stated the bound as $2^\gamma P(\mathbf{X}{=}\mathbf{x})$, without the "+1" in the exponent. This error had no impact on any arguments that build on Eq. 3 since the "+1" can be absorbed in a redefinition of γ.

(e) Finally, use the telescopic sum $\Delta_k = \Delta_1 + \sum_{i=2}^{k}(\Delta_i - \Delta_{i-1})$ and Eqs. 4 and 6 to show that

$$\sum_{i=1}^{k} D_{\mathrm{KL}}\big(P_{\mathrm{dec}}(X_i)\,\big|\big|\,P(X_i)\big) \leq \Delta_k < \mathrm{const} \tag{7}$$

and thus, that the average KL-divergence from $P(X_i)$ to $P_{\mathrm{dec}}(X_i)$ *per symbol* X_i can be bounded by an arbitrarily small constant $\propto 1/k$ by considering increasingly long initial random bit strings S_0.