Data Compression With and Without Deep Probabilistic Models

Lecture 3 (5 May 2022); lecturer: Robert Bamler
more course materials online at https://robamler.github.io/teaching/compress22/

Recap From Last Lecture:

- Entropy: fundamental lower bound for expected code word length L_C of any symbol code C:

$$H[p] \leq L_C \qquad \forall \text{ uniquely decodable } C$$

- Shannon code: reaches lower bound within less than 1 bit of overhead (per symbol)

$$H[p] \leq L_C < L_{C_{\text{Shannon}}} + 1$$

- bonus: a Shanon code satisfies the above guarantee not only in expectation, but even individually for each symbol

→ information content: $-\log_2 p(x)$

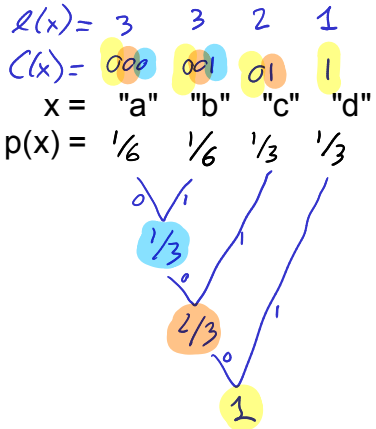
Recap From Last Problem Set:

Huffman Coding:
- conceptually simple algorighm (takes probability distribution as input and returns a code book as output)

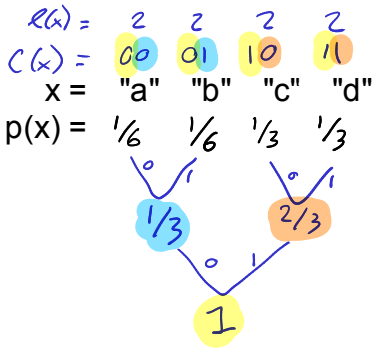
- claim: Huffman Coding builds an optimal code book (i.e., it minimizes the expected code word length)

→ Proof: today

- While the code words and even their individual lengths may not be uniquely defined (due to ties during execution of the algorithm), the expected code word length is independent of how one breaks a tie:



alternatively:



expected code word length:

$$L = \sum_{x \in \mathcal{X}} p(x)\ell(x) = \frac{1}{6}\times 3 + \frac{1}{6}\times 3 + \frac{1}{3}\times 2 + \frac{1}{3}\times 1 = 2$$

$$L = \frac{1}{3}\times 1 + \frac{2}{3}\times 1 + 1\times 1 = 2$$

expected code word length:

$$L = 2$$

$$L = \frac{1}{3}\times 1 + \frac{2}{3}\times 1 + 1\times 1 = 2$$

Today:

- proof of optimality of Huffman coding

- theoretical groundwork for more powerful (machine learning based) probabilistic models

Optimality of Huffman Coding

Goal: find an optimal (uniquely decodable) symbol code for a given probability distribution p,
i.e., one with the lowest expected code word length L.

Reminder: - Among all optimal uniquely decodable symbol codes for a given p, there is at least one prefix-free code.
→ Why?

- We define "optimality" here as minimizing the expected code word length. This is appropriate for many applications, but there are also use cases of data compression where one should optimize different metrics.
→ Examples:

The Huffman algorithm for finite alphabets: see Problem Set 1

Theorem: The Huffman algorithm constructs an optimal symbol code.
More precisely: assume we have

Then:

Reminder: We may assume, without loss of generality, that C is a prefix-free code (due to the corollary to the Kraft-McMillan Theorem).

Lemma 1: Assume again (*), and let C be an optimal (w.r.t. p) prefix code; let's sort the symbols s.t.

We break ties by code word lengths (descendingly):

(if there are still ties after this, break them arbitrarily)

Then:

Proof of Lemma 1:

(i) by contradiction:

(ii) proof by contradiction, building on (i):

Lemma 2: Assume again (*), and let C be an optimal (w.r.t. p) prefix code. Then $\exists x, x' \in \mathcal{X}$ with $x \neq x'$ and:

(i)

(ii)

Proof of Lemma 2: Assume that such a pair does not exist. But, from Lemma 1, we know:

Claim: either C is not optimal because we can drop the last bit of C(x) without violating the properties of a prefix code, or there exists a different pair of symbols that satisfy both (i) and (ii)

Proof of the claim:

Recap:

Theorem: The Huffman algorithm constructs an optimal symbol code.
More precisely: assume we have

- finite alphabet $\mathcal{X}$ with $|\mathcal{X}| \geq 2$
- probability distribution $p: \mathcal{X} \to [0,1]$ with $p(x) > 0 \quad \forall x \in \mathcal{X}$ } (*)

Then:

$\forall$ uniq. dec. sym. codes C on $\mathcal{X}$ that minimize the expected code word length L w.r.t. p: $\exists$ a Huffman code C_H with the same code word lengths, i.e., $|C(x)| = |C_H(x)| \quad \forall x \in \mathcal{X}$.

Lemma 1: Assume again (*), and let C be an optimal (w.r.t. p) prefix code; let's sort the symbols s.t.

$$p(x_1) \leq p(x_2) \leq p(x_3) \leq \dots$$

We break ties by code word lengths (descendingly): $\ell(x_i) := |C(x_i)|$

if $p(x_i) = p(x_{i+1})$ then $\ell(x_i) \geq \ell(x_{i+1})$

(if there are still ties after this, break them arbitrarily)

Then: (i) $\ell(x_1) \geq \ell(x_2) \geq \ell(x_3) \geq \dots$
(ii) $\ell(x_1) = \ell(x_2)$ (↑)

Lemma 2: Assume again (*), and let C be an optimal (w.r.t. p) prefix code. Then $\exists x, x' \in \mathcal{X}$ with $x \neq x'$ and:

(i) $\ell(x) = \ell(x') \geq \ell(\tilde{x}) \quad \forall \tilde{x} \in \mathcal{X}$, and
(ii) $C(x)$ & $C(x')$ only differ in last bit

Proof of the Theorem (optimality of Huffman coding):

→ by induction over $|\mathcal{X}|$

- base case: $|\mathcal{X}| = 2$

- induction step: $|\mathcal{X}| > 2$ assuming that theorem holds for $\forall |\mathcal{X}'| = |\mathcal{X}| - 1$

↳ Claim: $\tilde{C}$ is an optimal prefix code on $\tilde{\mathcal{X}}$ (with respect to $\tilde{p}$)

↓ Proof: if it isn't an optimal prefix code then there exists a better prefix code $\tilde{\tilde{C}}$ on $\tilde{\mathcal{X}}$

Thus, $\widetilde{C}$ is indeed an optimal prefix code on $\widetilde{\mathcal{X}}$ (which has size $|\widetilde{\mathcal{X}}| = |\mathcal{X}| - 1$).

↳ Recall that x_1 and x_2 (which are "contracted" in the definition of C) are two symbols with lowest probability.

⇒ Running Huffman algorithm on $\mathcal{X}$ also contracts contracts x_1 and x_2 in the first step. The remaining steps of the algorithm then construct a prefix code with the same code word lengths as $\widetilde{C}$ on $\widetilde{\mathcal{X}}$ by induction assumption.

Remarks and Outlook:

- Huffman coding is still widely used in practice (e.g., in the "deflate" compression method used in zip/gzip and for compressed HTTP streams, in PNG, in most JPEGs, ...)

- However, Huffman coding is only an optimal symbol code. In Problem 2.4 of the current problem set (discussed tomorrow), your task is to think about the limitations of symbol codes. In the next lecture, we will start discussing so-called stream codes, which outperform Huffman coding (especially in the regime of low entropy per symbol, which is relevant for modern machine learning based data compression methods).

- On the next week's problem set (Problem Set 4), you will then use our implementation of Huffman Coding (from Problem Set 2) and you'll combine it with a machine learning model that you'll train yourself. The two components (model and entropy coding algorithm) together will result in a fully functioning (albeit ridiculously slow) deep learning based compression method for English text.

Probabilistic Models, Random Variables, and Correlations

Robert Bamler · 5 May 2022

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Quantifying Modeling Errors: The Kullback-Leibler Divergence

- **Qualitatively:** better probabilistic models ⇒ better compression performance
- **Goal:** *quantify* loss in compression performance due to imperfect probabilistic models

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Reminder: Optimal Compression Performance

Consider general lossless compression setup (i.e., no longer restricted to symbol codes)

- discrete message space $\mathcal{X}$
- some data source generates a message $\mathbf{x} \in \mathcal{X}$ with probability $p_{\text{data}}(\mathbf{x})$
- encoder C maps $\mathbf{x}$ injectively to a bit string $C(\mathbf{x}) \in \{0,1\}^*$
- Def: "bit rate" $R_C(\mathbf{x}) := |C(\mathbf{x})|$, i.e., length (in bits) of compressed representation
  ⇒ if C is the *optimal* code for p_{data} then: $R_C(\mathbf{x}) = -\log_2 p_{\text{data}}(\mathbf{x}) + \varepsilon \quad \forall \mathbf{x} \in \mathcal{X}$
  (see Problem 2.4 on Problem Set 2)

  ⇒ $\boxed{\textit{optimal expected}\text{ bit rate: } \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[R_{C_{\text{optimal for } p_{\text{data}}}}(\mathbf{x})\right] = H[p_{\text{data}}] + \varepsilon}$

Problem: In practice, we don't know p_{data}.

- E.g., consider the probability distribution p_{data} for videos that you might take with your phone's camera.
  - huge message space $\mathcal{X}$ (all possible HD videos);
  - it's inconceivable to know p_{data} exactly.
- We might, however, have some empirical samples ("training set") from p_{data}.
  $\Rightarrow$ Use these samples to fit a model p_{model} that approximates the (unknown) p_{data}.
- Then, consider a compression code C that is optimal for p_{model}:
  - bit rate of a given message $\mathbf{x}$: $R_C(\mathbf{x}) =$ $\forall \mathbf{x} \in \mathcal{X}$
  - *expected* bit rate: $\mathbb{E}_{\mathbf{x}\sim p}\big[R_C(\mathbf{x})\big] =$
- Idea: fit optimal p_{model} by minimizing $H(p_{\text{model}}, p_{\text{data}})$

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Entropy, Cross Entropy, and Kullback-Leibler Divergence

- Compare:
  - true entropy of the data source: $H[p_{\text{data}}] = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[-\log_2 p_{\text{data}}(\mathbf{x})]$
  - entropy of the model: $H[p_{\text{model}}] = \mathbb{E}_{\mathbf{x}\sim p_{\text{model}}}[-\log_2 p_{\text{model}}(\mathbf{x})]$
  - Cross entropy between data source and model: $H(p_{\text{data}}, p_{\text{model}}) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[-\log_2 p_{\text{model}}(\mathbf{x})]$
- **Def. "Kullback Leibler" divergence** $:=$ bit rate overhead due to modeling errors

$$\boxed{D_{\text{KL}}(p_{\text{data}} \,||\, p_{\text{model}}) := H(p_{\text{data}}, p_{\text{model}}) - H[p_{\text{data}}] \geq 0}$$ (see Problem 3.2)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Needed: Expressive Probabilistic Models

So far: $\mathbf{x} \in \mathfrak{X}^*$ and $p_{\text{model}}(\mathbf{x}) = (p_{\text{length}}(k)) \prod_{i=1}^{k} p(x_i)$.

I.e., symbols were assumed to be "i.i.d." ("*independent* and *identically distributed*")

- *"identically distributed:"* p is the same probability distribution for all $i \in \{1, \ldots, k\}$
  - We can easily overcome this limitation: $p_{\text{model}}(\mathbf{x}) = (p_{\text{length}}(k)) \prod_{i=1}^{k} p_i(x_i)$
  - Construct an individual code book C_i (optimized for p_i) for each $i \in \{1, \ldots, k\}$.
  - Easy to see: if all C_i are prefix codes then the concatenation of code words $C_1(x_1) \,||\, C_2(x_2) \,||\, \ldots \,||\, C_k(x_k)$ is still uniquely decodable.
- *"independent:"* the probability distribution p_i does not change if we change the value of some symbol x_j with $j \neq i$.
  - simplistic assumption: e.g., in English text, p_i('u') increases considerably if x_{i-1} = 'q'.
  - This limitation is more difficult to overcome. $\rightarrow$ *correlations*

Interlude: Probability Theory & Random Variables

What makes up a "probabilistic model":

- *sample space* Ω*:* (abstract) space of "all states of the world"
  - *event* $E \subset \Omega$*:* "event E occurs" := "the world is in some state $\omega \in E$".
- *probability measure:* a function $P : \Sigma \to [0, 1]$ where
  - Σ is a so-called σ-algebra on Ω (a set of subsets of Ω)
  - $P(\Omega) = 1$
  - $P(\emptyset) = 0$
  - countable additivity: $P\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} P(E_i)$ if all E_i are pairwise disjoint
  - therefore: $P\left(\bigcup_{i=1}^{k} E_i\right) = \sum_{i=1}^{k} P(E_i)$ if all E_i are pairwise disjoint (proof: set $E_i = \emptyset \; \forall i > k$)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Examples of Probability Measures

1. Simplified Game of Monopoly: (throw two fair three-sided dice)
   - sample space: $\Omega = \{1, 2, 3\}^2 = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 2), (3, 3)\}$
   - sigma algebra: $\Sigma = 2^{\Omega} := \{\text{all subsets of } \Omega \text{ (including } \emptyset \text{ and } \Omega)\}$
   - probability measure P: for all $E \in \Sigma$, let $P(E) := |E|/|\Omega| = |E|/9$
2. Departure times of the next three buses from "Sternwarte":
   - sample space (in a simple model): $\Omega = \mathbb{R}^3$
   - sigma algebra: all "measurable subsets" of $\mathbb{R}^3$
     (= all cuboids $[a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ and their countable unions)
   - probability measure P: a complicated function, but we know it satisfies certain properties, e.g., P("next bus departs before 1.15 pm") = P("next bus departs before 1.10 pm") + P("next bus departs between 1.10 pm and 1.15 pm").
   - **Question:** what is the probability that the next bus departs *exactly* at 1.15 pm?
     I.e., what is $P(\{1.15 \text{ pm}\} \times \mathbb{R}^2)$?

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Random Variables

- Often, we we're not interested in the *full* description of the state $\omega \in \Omega$, but only in certain properties of it.
- *Def. random variable:* function $X : \Omega \to \mathbb{R}$

Examples:

1. Simplified Game of Monopoly; $\Omega = \{(a, b) \quad \text{where} \quad a, b \in \{1, 2, 3\}\}$
   - total value: $X_{\text{sum}}((a, b)) = a + b \in \{2, 3, 4, 5, 6\}$
   - value of the red die: $X_{\text{red}}((a, b)) = a$
   - value of the blue die: $X_{\text{blue}}((a, b)) = b$
2. In our bus schedule model from before; $\Omega = \mathbb{R}^3$
   - Time between the next bus and the one after it: $X_{\text{gap}}((x_1, x_2, x_3)) = x_2 - x_1$

Properties of Individual Random Variables

- "Probability that a random variable X has some given value x":
  $P(X = x) := P\big(X^{-1}(x)\big) = P\big(\{\omega \in \Omega : X(\omega) = x\}\big)$
  - Example 1 (Simplified Game of Monopoly): $P(X_{\text{sum}} = 3) =$
  - Example 2 (bus schedule): $P(X_{\text{gap}} = 20 \text{ minutes}) =$
- When we write just $P(X)$, then we mean the *function* that maps $x \mapsto P(X = x)$.
- Expectation value of a random variable X under a model P
  - discrete case: $\mathbb{E}_P[X] := \sum_{\omega \in \Omega} P(\{\omega\})\, X(\omega) = \sum_{x \in X(\Omega)} P(X{=}x)\, x$

    examples: $\mathbb{E}_P[X_{\text{red}}] =$; $\mathbb{E}_P[X_{\text{blue}}] =$; $\mathbb{E}_P[X_{\text{sum}}] =$
  - continuous case: $\mathbb{E}_P[X] := \int X(\omega)\, dP(\omega)$ (see next slide)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Properties of Individual Random Variables (cont'd)

- *Cumulative Density Function (CDF):* $P(X \le x) := P\big(\{\omega \in \Omega : X(\omega) \le x\}\big)$
  - Example 1 (Simplified Game of Monopoly): $P(X_{\text{sum}} \le 3) =$
  - Example 2 (bus schedule): $P(X_{\text{gap}} \le 20 \text{ minutes}) \in [0, 1]$ (can be nonzero)
- Analogously definitions for: $P(X < x)$, $P(X \ge x)$, $P(X > x)$, $P(X \in \text{some set})$, ...
- *Probability Density Function (PDF)* of a real-valued random variable X:
  $p(x) := \frac{d}{dx} P(X \le x)$ (if derivative exists)
  $\rightarrow$ expectation value: $\mathbb{E}_P[X] = \int X(\omega)\, dP(\omega) = \int_{-\infty}^{\infty} x\, p(x)\, dx$
  (if a density $p(x)$ exists)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Properties of *Collections* of Rand. Variables: Statistical Independence

- Def. *joint probability distribution* of two random variables X and Y:
  $P(X{=}x, Y{=}y) = P\big(\omega \in \Omega : X(\omega) = x \ \wedge \ Y(\omega) = y\big)$
  (notation: "$P(X, Y)$": function that maps $(x, y) \mapsto P(X{=}x, Y{=}x)$)
- Def.: X and Y are *(statistically) independent* if and only if: $P(X, Y) = P(X)\, P(Y)$
  (i.e., if $P(X{=}x, Y{=}y) = P(X{=}x)\, P(Y{=}y) \ \ \forall x, y$)

  Examples (Simplified Game of Monopoly):
  - X_{red} and X_{blue} are statistically independent.
  - X_{red} and X_{sum} are *not* statistically independent. (proof: Problem 3.1)
- Def.: *conditional* independence of X and Y given Z: see Lecture 6

Conditional Probability Distributions

- When two random variables X and Y are *not* statistically independent, then learning about X tells us something about Y.
- Examples (Simplified Game of Monopoly): $P(E) = \frac{|E|}{9}$
  - What are the *marginal* probability distributions $P(X_{\text{red}})$ and $P(X_{\text{sum}})$ for the red die and the sum, respectively?
  - Assume the red die came up with value 1. What is the probability distribution of X_{sum} once you know that $X_{\text{red}} = 1$? We call this the *conditional* probability distribution $P(X_{\text{sum}} \mid X_{\text{red}} = 1)$.
  - Assume the sum of both dies is 4. What is the conditional probability distribution $P(X_{\text{red}} \mid X_{\text{sum}} = 4)$?
  - Assume the blue die came up with value 1. What is the conditional probability distribution $P(X_{\text{red}} \mid X_{\text{blue}} = 1)$?

Conditional Probability Distributions (cont'd)

- Def. "conditional probability of event E_2 *given* event E_1": $P(E_2 \mid E_1) := \frac{P(E_1 \cap E_2)}{P(E_1)}$
  - Thus, $P(E_2 \mid E_1)$ is a (properly normalized) probability distribution with respect to the first parameter, i.e., $P(E_2 \mid E_1) + P(\Omega \setminus E_2 \mid E_1) = \frac{P(E_2 \cap E_1) + P((\Omega \setminus E_2) \cap E_1)}{P(E_1)} = \frac{P(E_1)}{P(E_1)} = 1$.
- Def. "conditional probability distribution of random variable Y *given* random variable X": $P(Y \mid X) := \frac{P(X,Y)}{P(X)}$ i.e., $P(Y = y \mid X = x) := \frac{P(X=x, Y=y)}{P(X=x)} \quad \forall x, y$
  - Thus, if X and Y are statistically independent *(but only then!)*: $P(Y \mid X) = \frac{P(X,Y)}{P(X)} = \frac{P(X)\,P(Y)}{P(X)} = P(Y)$
  - In the general case: "chain rule" of probability theory: (follows directly from above def.) $P(X_1, X_2, X_3, \ldots) =$

Warning: Conditionality $\neq$ Causation

- We'll often specify a joint probability distribution as, e.g., $P(X, Y) = P(X)\,P(Y \mid X)$.

  But just because we write down $P(Y \mid X)$, this does not necessarily mean that X is the cause for Y.
- **Example:** (Simplified Game of Monopoly):
  - X_{red} and X_{blue} can be considered the cause for X_{sum}.
  - But, in the examples two slides ago, we were still able to calculate, e.g., $P(X_{\text{red}} \mid X_{\text{sum}})$, i.e., the *probability of the cause X_{red} given its effect X_{sum}*.

    $\rightarrow$ This is called "posterior inference".
    (more in Lectures 6 and 7)

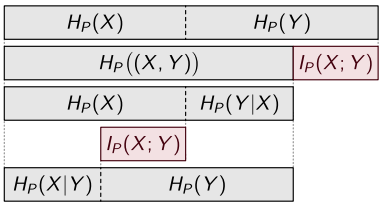
Information Content and Entropy of Random Variables

- Definitions as you'd expect:
  - information content of the statement "$X = x$":
  - entropy of a random variable X under a model P: $H_P(X) :=$
  - joint and conditional information content and entropy: see Problems 3.3 and 3.4.
- Subadditivity of entropies: For any two random variables X and Y, we have:

  $$\boxed{H_P\big((X,Y)\big) \leq H_P(X) + H_P(Y)}$$ (Proof: Problem 3.5)

  - equality holds if X and Y are statistically independent
  - Thus, wrongfully assuming independence incurs a compression overhead of $I_P(X;Y)$ bits:

    Def. *mutual information*: $\boxed{I_P(X;Y) := H_P(X) + H_P(Y) - H_P\big((X,Y)\big) \geq 0}$



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Outlook

- Next Lecture: computationally efficient models for correlated random variables
  - autoregressive models
  - latent variable models
- Going forward: how can we use such models for data compression?
  - Problem Set 4 (handed out next week): compressing English text with a learnt autoregressive model (using recurrent neural networks)
  - Lecture 4 (next week): computationally efficient lossless compression methods for autoregressive models (with better compression performance than Huffman coding)
  - Lectures 5 and 6: efficient and effective compression methods for latent variable models
  - Lectures 7 and 8: deep-learning based latent variable models