Introduction to Probabilistic Machine Learning

Ralf Herbrich

Information Theory

Overview

1. Basics of Information Theory
2. Arithmetic Coding
3. Distance Measures for Probabilities

Overview

1. **Basics of Information Theory**
2. Arithmetic Coding
3. Distance Measures for Probabilities

Motivating Example: Information and Coin Tosses

- **Scenario 1**:
  - A coin toss with uncertain outcome modelled via $X \sim \mathrm{Ber}(p)$
  - $h(x; p)$ is the information/surprise received when you observe the value of x
  - **Question**:
    - How much is $h(1; 1)$ when the success probability was 100%?
    - What's the relation between $h(1; p = 99\%)$ and $h(1; q = 1\%)$?
  - **Conclusion**: $h(x)$ is monotonically decreasing in $p(x)$
- **Scenario 2**:
  - Two independent coins are tossed modelled via $p(x, y) = p(x) \cdot p(y)$
  - **Question**: In what relation does $h(x, y)$ stand to $h(x)$ and $h(y)$?
  - **Conclusion**: If $p(x, y) = p(x) \cdot p(y)$ then $h(x, y) = h(x) + h(y)$

$$h(x) = -\log_b\big(p(x)\big)$$

$\boldsymbol{h(x,y)}$
$\boldsymbol{= h(x) + h(y)}$

$\boldsymbol{h(x,y)}$
$\boldsymbol{> h(x) + h(y)}$

$\boldsymbol{h(x,y)}$
$\boldsymbol{< h(x) + h(y)}$

Measure of Information: Entropy

- **Entropy**. *The* entropy *of a random variable* X *is the* average level of information *inherent to the variables outcomes and is defined by* $(b > 1)$

$$H_b[X] := -\sum_x P(X=x)\cdot\log_b\big(P(X=x)\big)$$
$$= E_{x\sim P}\big[-\log_b\big(p(x)\big)\big]$$

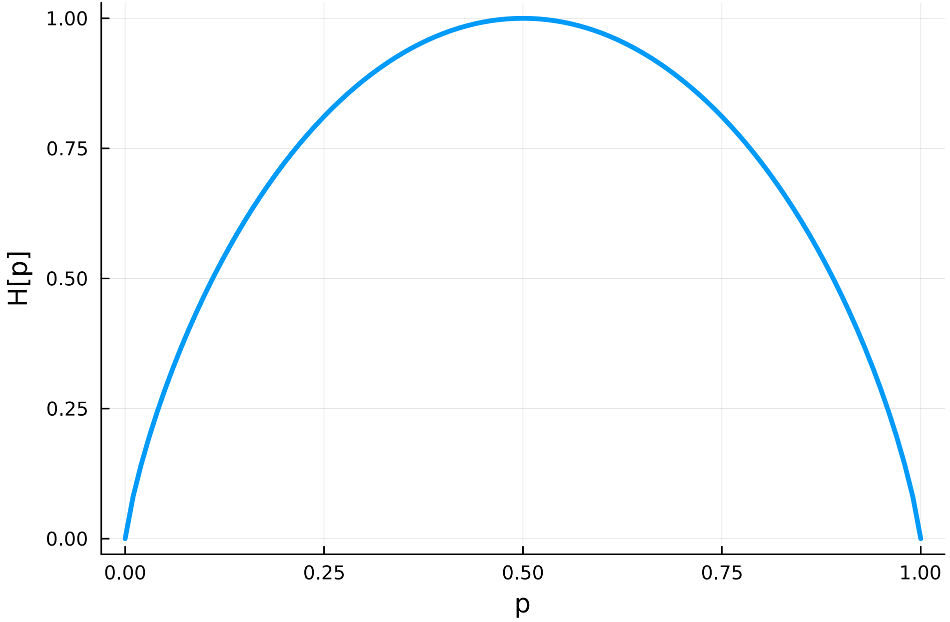
- **Khinchin (1957)**. *Entropy* $H[X]$ *as a* measure of information *of a random variable* X *follows from the following four axioms:*
  1. $H[X]$ *depends only on the probability distribution of* X.
  2. $H[X]$ *is maximal for the uniform distribution* $P(X)$.
  3. $H[Y] = H[X]$ *if* X *and* Y *have the same non-zero probabilities.*
  4. *For any random variables* X *and* Y,

$$H[X,Y] = H[X] + \underbrace{\sum_x P(X=x)\cdot\ H[Y\,|X=x]}_{H[Y|X]}$$

Aleksandr Khinchin (1894 – 1959)

Example: Binary Entropy

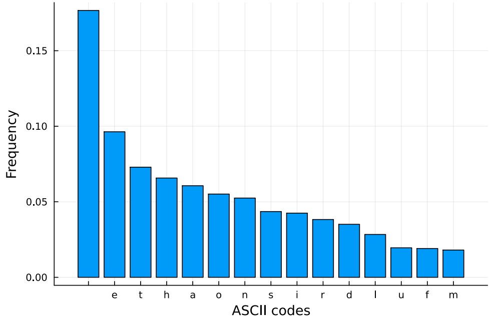
$$H_2[p] = p \cdot \log_2(p) + (1 - p) \cdot \log_2(1 - p)$$

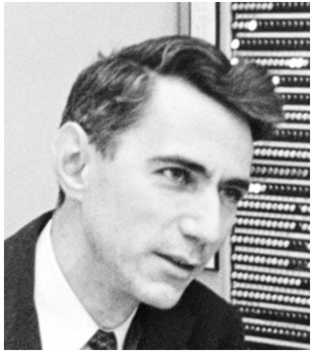


Entropy and the Noiseless Coding Theorem

- **(Shannon 1948).** *N independent and identically distributed random variables each with entropy $H[X]$ can be compressed into more than $N \cdot H[X]$ bits with negligible risk of information loss, as $N \to \infty$; but if they are compressed into fewer than $N \cdot H[X]$ bits it is virtually certain that information will be lost.*
- **Application** in data compression when modelling the value X of a byte modelled as a random variable over $n = 256$ values
  - **Random bytes**: $H[X] = -\sum_{i=1}^{256} \frac{1}{256} \log_2\left(\frac{1}{256}\right) = -\log_2\left(\frac{1}{256}\right) = 8$
  - **Random letters from the English alphabet**:

$$H = 4.48917$$





Claude Shannon
(1913 – 2001)

Introduction to Probabilistic Machine Learning

Unit 11 – Information Theory

Noiseless Coding Theorem: An Example

- **Scenario**: We have 8 class labels with probabilities $\left\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\right\}$
- **Naïve Encoding**: We use a uniform distribution with 3 bits per symbol
$$H\left[\left\{\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right\}\right] = 3$$
- **However**, the entropy is 2 bits!
$$H[X] = 2$$
- **Prefix Code**: Unique binary prefix of consecutive 1's for each unique probability
- **Decode**: $\underbrace{\boxed{1\ 1\ 0}}_{C_3}\underbrace{\boxed{0}}_{C_1}\underbrace{\boxed{1\ 1\ 1\ 0}}_{C_4}$

Class	Code	$P(C)$	Length	$E[\text{Length}]$
1	0	1/2	1	16/32
2	10	1/4	2	16/32
3	110	1/8	3	12/32
4	1110	1/16	4	8/32
5	111100	1/64	6	3/32
6	111101	1/64	6	3/32
7	111110	1/64	6	3/32
8	111111	1/64	6	3/32

Overview

1. Basics of Information Theory
2. **Arithmetic Coding**
3. Distance Measures for Probabilities

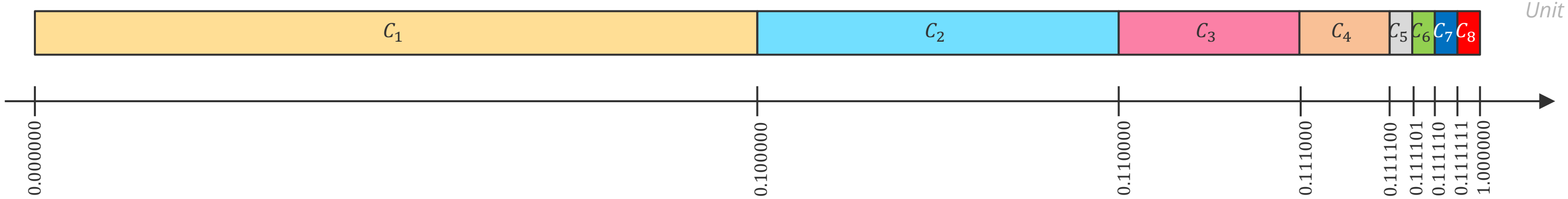
Arithmetic Coding

- **Binary Encoding of Probability**: A binary string $\boldsymbol{b} \in \{0,1\}^n$ encodes the number in $[0,1)$ via

$$N(\boldsymbol{b}) = b_1 \cdot \frac{1}{2} + b_2 \cdot \frac{1}{4} + \cdots + b_n \cdot \frac{1}{2^n} = \sum_{i=1}^{n} b_i \cdot 2^{-i}$$

- **Observations**: Given a binary string $\boldsymbol{b} \in \{0,1\}^n$ of length n, it encodes all values of probabilities $\left[N(\boldsymbol{b}), N(\boldsymbol{b}) + \frac{1}{2^n}\right)$
  - These are all the strings $\boldsymbol{bc}$ where $\boldsymbol{c} \in \{0,1\}^k, k \in \mathbb{N}$
- **From Probabilities to Data**. If we map each symbol $x \in \mathcal{X}$ to an integer $I(x)$ (e.g., ASCII code), then the cumulative distribution $P(I(x) \leq k)$ covers the interval $[0,1]$

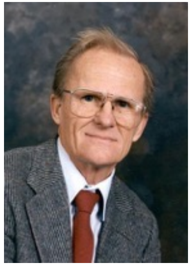
Class	Code	$P(C)$	$N(\text{Code})$
1	0	1/2	$\frac{0}{64} = 0$
2	10	1/4	$\frac{32}{64} = \frac{1}{2}$
3	110	1/8	$\frac{48}{64} = \frac{3}{4}$
4	1110	1/16	$\frac{56}{64} = \frac{7}{8}$
5	111100	1/64	$\frac{60}{64} = \frac{15}{16}$
6	111101	1/64	$\frac{61}{64} = \frac{61}{64}$
7	111110	1/64	$\frac{62}{64} = \frac{31}{32}$
8	111111	1/64	$\frac{63}{64} = \frac{63}{64}$



Arithmetic Coding: Algorithm

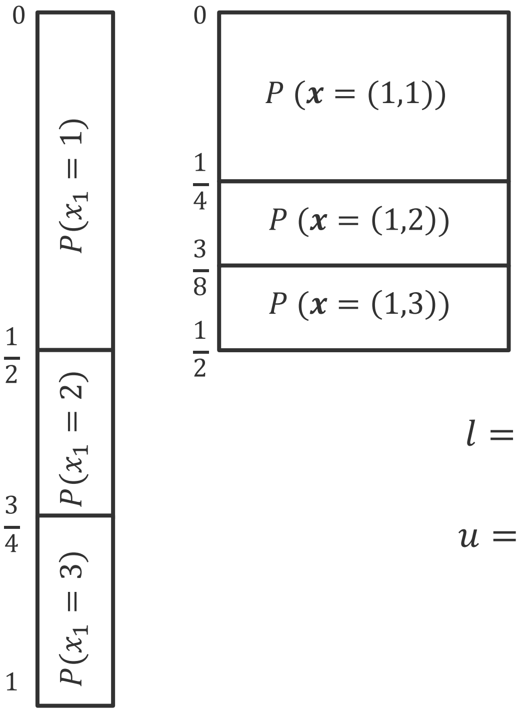
- **Given**:
  - Data as a sequence of tokens with K values: $(x_1, x_2, \dots, x_m) \in \{1, \dots, K\}^m$
  - A probability model for each token: $P(x_j | x_1, x_2, \dots, x_{j-1}) \in [0,1)$
- **Idea**: Map a data stream $\boldsymbol{x} = x_1, \dots, x_m$ to a unique interval $[N(\boldsymbol{l}), N(\boldsymbol{u})\,) \subseteq [0,1)$ that has a width of $P(x_1, x_2, \dots, x_m)$
  - **Coding of $\boldsymbol{x} \in \{\mathbf{1}, \dots, \boldsymbol{K}\}^{\boldsymbol{m}}$**: Start with $l = 0$ and $u = 1$. Successively iterate the following
    1. Compute $q_{\text{lower}} := P(X_j < x_j | x_1, x_2, \dots, x_{j-1})$ and $q_{\text{upper}} := P(X_j \leq x_j | x_1, x_2, \dots, x_{j-1})$
    2. Update $l \leftarrow l + q_{\text{lower}} \cdot (u - l)$ and $u \leftarrow l + q_{\text{upper}} \cdot (u - l)$

    – After all data have been incorporated, compute $\boldsymbol{l} = N^{-1}\left(l + \frac{1}{4}(u - l)\right)$ and $\boldsymbol{u} = N^{-1}\left(l + \frac{3}{4}(u - l)\right)$ and return the first n bits of $\boldsymbol{l}$ abd $\boldsymbol{u}$ that are the same
  - **Decoding of $\boldsymbol{b} \in \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$**: Successively iterate the following procedure
    1. Compute $q_k := P(X_j \leq k | x_1, x_2, \dots, x_{j-1})$ for $k = 1, \dots, K$ and $q_0 = 0$
    2. Pick $x_j = k^*$ if $N(\boldsymbol{b}) \in [q_{k-1^*}, q_{k^*}\,)$



Jorma Rissanen
(1932 – 2020)

Arithmetic Coding: An Example



$$\boldsymbol{x} = (1,2)$$

$$\boldsymbol{b} = (0,1,0)$$

$$l = \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{8} = \frac{9}{32} = 0.01001_2$$

$$u = \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{8} = \frac{11}{32} = 0.01011_2$$

$x = k$	$P(x = k)$	$-\log_2 P(x = k)$
1	1/2	1
2	1/4	2
3	1/4	2

Arithmetic Coding: Optimal Code Length

- **Question**: How large will n be for a given data stream $\boldsymbol{x}$?

$$\boxed{-\log_2 P(\boldsymbol{x}) \leq n \leq -\log_2 P(\boldsymbol{x}) + 1}$$

- **Proof**. Assume that $\boldsymbol{l}$ and $\boldsymbol{u}$ differ first at bit $n+1$. Because $N(\boldsymbol{u}) - N(\boldsymbol{l}) = \frac{1}{2}P(\boldsymbol{x})$

$$\frac{1}{2}P(\boldsymbol{x}) = \underbrace{\sum_{i=1}^{n}(u_i - l_i)2^{-i}}_{=0} + \sum_{i=n+1}(u_i - l_i) \cdot 2^{-i}$$

$$2^{-(n+1)} \leq 2^{-1}P(\boldsymbol{x}) \leq 2^{-n}$$

$$-(n+1) \leq \log_2 P(\boldsymbol{x}) - 1 \leq -n$$

$$n + 1 \geq -\log_2 P(\boldsymbol{x}) + 1 \geq n$$

- **Intuitively**: Between 0 and 1 there are exactly $\frac{2}{P(\boldsymbol{x})}$ many intervals of length $\frac{1}{2}P(\boldsymbol{x})$ and one needs $\log_2 \frac{2}{P(\boldsymbol{x})} = -\log_2 P(\boldsymbol{x}) + 1$ many bits to index them with a number.

Arithmetic Coding: Fixed Point Arithmetic

- **Problem**: Arbitrary precision multiplication and addition in binary numbers!
- **Idea**: Fixed point arithmetic with 32-bit registers and 64-bit addition/multiplication
  1. Instead of l and u, represent l and $r \coloneqq u - l$ as as unsigned 32-bit integer
  2. When updating l and u, multiply them by two whenever the most significant bit of l **and** u are identical (because they cannot change anymore!)
  3. Represent the probabilities as fixed points to the basis 2^d ($d = 16$ typically)
- **Update**: If $q_{\text{lower}} \in \{0, \ldots, 2^d\}$ and $q_{\text{upper}} \in \{0, \ldots, 2^d\}$ then

$$l \leftarrow l + \text{UInt32}\left(\text{UInt64}(r) \cdot \frac{q_{\text{lower}}}{2^d}\right)$$

$$u \leftarrow l + \text{UInt32}\left(\text{UInt64}(r) \cdot \frac{q_{\text{upper}}}{2^d}\right)$$

- Implement all multiplications and divisions by 2 via left-/right-shifts
- **Practical Considerations**: The decoder would never stop because even a $N(\boldsymbol{b}) \in [q_{k^*}, q_{k^*+1})$ is true for the most likely value k!
  - Add a special symbol that indicates end-of-stream and encode it to later stop the decoder.

Overview

1. Basics of Information Theory
2. Arithmetic Coding
3. **Distance Measures for Probabilities**

Relative Entropy: Kullback-Leibler Divergence

- **Approximation**. Assume we receive symbols (e.g., classes) from an unknown distribution $p(x)$ but we use the approximate distribution $q(x)$ for encoding each symbol x.
  - Each symbol x has an "information difference" of $-\log_2\left(q(x)\right) + \log_2\left(p(x)\right)$!
- **Kullback-Leibler Divergence (1951)**. *The Kullback-Leibler divergence of two distributions* p *and* q *is defined as*

$$\mathrm{KL}(p|q) \coloneqq \sum_x p(x) \cdot \left[\log_2\left(p(x)\right) - \log_2\left(q(x)\right)\right]$$
$$= \sum_x p(x) \cdot \left[\log_2\left(\frac{p(x)}{q(x)}\right)\right]$$

- **Two properties of Kullback-Leibler Divergence**.
  1. In general, $\mathrm{KL}(p|q) \neq \mathrm{KL}(q|p)$
  2. $\mathrm{KL}(p|q) = 0$ if and only if $p = q$

Solomon Kullback
(1909 – 1994)

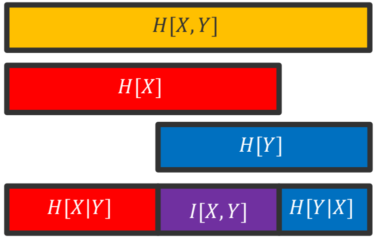
Richard Leibler
(1914 – 2003)

Mutual Information

- **Motivation**: If we have a joint distribution $p(x, y)$, how much information is conveyed by knowing a value of one of the variables about the other?
  - The minimal amount of information is achieved if $p(x, y) = p(x)p(y)$ (in fact, this is zero information!)
- **Mutual Information**. *The* mutual information *between two random variables X and Y is the Kullback-Leibler divergence of the joint distribution $p(x, y)$ to a fully factorizing distribution $p(x) \cdot p(y)$*

$$I[X, Y] \coloneqq \mathrm{KL}(p(x, y) | p(x) \cdot p(y))$$

- **Properties of Mutual Information**
  - $I[X, Y] \geq 0$ for all probabilities $p(x, y)$
  - $I[X, Y] = 0$ if and only if X and Y are independent
  - $I[X, Y] = H[X] - H[X|Y] = H[Y] - H[Y|X]$



Summary

1. **Information Theory**
   - Information theory measures the amount of uncertainty and only depends on probabilities.
   - The information of each outcome of a probability distribution is $-\log_2(p(x))$ bits and entropy is the expected information across all outcomes.
   - Entropy is the smallest size a file can be compressed to without loss (coding theory!)

2. **Arithmetic Coding**
   - Compression algorithm that maps each sequence of tokens to its probability under a given probabilistic model
   - Using fixed point arithmetic for arbitrary-precision numbers is key
   - Achieves a near-optimal encoding/compression length (up to one bit!)

Hasso Plattner Institut
Digital Engineering • Universität Potsdam

See you next week!