Intro to coding and information theories and the noisy-channel coding theorem

Jakub (Kuba) Perlin

Churchill College

2017-10-11

Introduction to coding theory

The big picture

$$\begin{array}{c}\text{messages}\\ ms \in \{1..M\}^*\end{array} \xrightarrow{\text{encode}} w \in \Sigma_{in}^* \xrightarrow[\text{channel}]{\text{add errors}} w' \in \Sigma_{out}^* \xrightarrow{\text{decode}} \begin{array}{c}\text{est. messages}\\ ms' \in \{1..M\}^*\end{array}$$

$$\textbf{Code} = \{\text{codewords}\}$$

Discrete, memoryless, noisy channel model

- Input alphabet Σ_{in},
- output alphabet Σ_{out},
- transition probabilities $Pr(out = o_j | in = i_k)$.

Often $\Sigma_{in} = \Sigma_{out}$ but not necessarily.

Example: an additional "error" output symbol.

Example:
$\Sigma_{in} = \Sigma_{out} = \{0, 1\}$,
channel flips every bit with a probability p.

Coding theory branches

$$\underset{ms \in \{1..M\}^*}{\text{messages}} \xrightarrow{\text{encode}} w \in \Sigma_{in}^* \xrightarrow[\text{channel}]{\text{add errors}} w' \in \Sigma_{out}^* \xrightarrow{\text{decode}} \underset{ms' \in \{1..M\}^*}{\text{est. messages}}$$

- Source coding - compression
- **Channel coding - error correction**

Goals

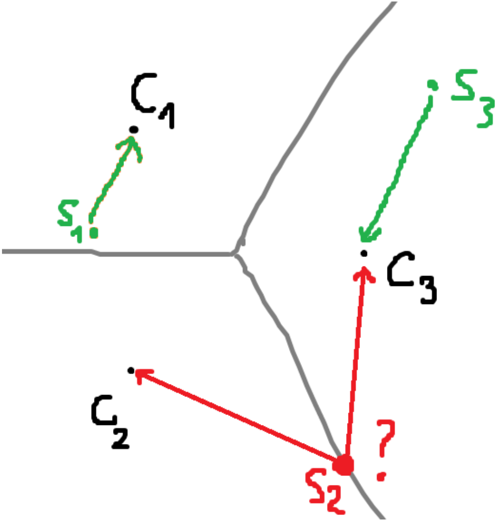
$$\underset{ms \in \{1..M\}^*}{\text{messages}} \xrightarrow{\text{encode}} w \in \Sigma_{in}^* \xrightarrow[\text{channel}]{\text{add errors}} w' \in \Sigma_{out}^* \xrightarrow{\text{decode}} \underset{ms' \in \{1..M\}^*}{\text{est. messages}}$$

- Error detection
- Error correction

Decoding schemes

- Maximum-likelihood decoding.
  Requires knowledge of transition probabilities.
- Minimum-distance decoding.

Min-dist decoding



Definition of distance

Hamming distance of two n-words $x, y \quad = \quad \#$places where they differ:

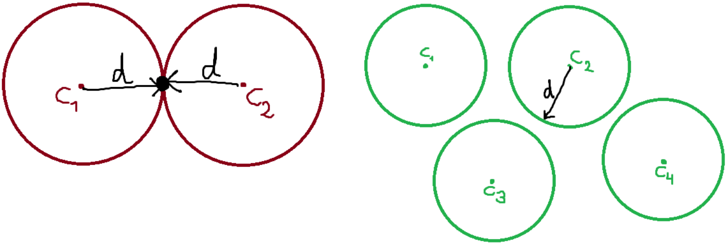
$$d(x, y) = |\{i : x[i] \neq y[i]\}|$$

Example: $d(91111, 94321) = 3$.

Min-dist error correction

Code's **minimal distance** = smallest distance between any two codewords.

- A code with minimal distance $2d+1$ **corrects up to d errors**.



The trade off: error correction vs. efficiency

- Coding is about adding **redundancy**.

The trade off: error correction vs. efficiency

- Coding is about adding **redundancy**.

$$\text{Rate} = \frac{\#\text{meaningful bits}}{\#\text{all transmitted bits}}$$

5-repetition code 'RC5'

$M = \{0, 1\}$ - the messages
$\Sigma_{in} = \Sigma_{out} = \{0, 1\}$ - the alphabet
$C = \{00000, 11111\}$ - the codewords
$0 \mapsto 00000$
$1 \mapsto 11111$

- Minimal distance $= 5$.
- Rate $= 0.2$.

Example:

- Receive 00000 11101 01010.
- Min-dist codewords are 00000, 11111, 00000.
- Decode to 010.

Linear codes

Fields

$\mathbb{Z}_q$ (q prime) is a **field**.

Can add, subtract, multiply, divide.

Structure like the real numbers.

Vector spaces

$\mathbb{Z}_q^n$ is a **vector space** with field of scalars $\mathbb{Z}_q$.

Can add vectors and multiply by scalars.

Structure like Euclidean space $\mathbb{R}^3$.

Definition of linear codes

A linear code $= \{\textit{codewords}\}$.

Definition of linear codes

A linear code $= \{codewords\}$.

A **linear code** is a subspace of the vector space $\mathbb{Z}_q^n$. (q prime)

A **subspace** of $V =$
a subset of V that's a vector space w.r.t. inherited operations.

$C \subseteq \mathbb{Z}_q^n$ is a linear code $\iff c_i + c_j \in C$ and $\forall a \in \mathbb{Z}_q \, . \, ac \in C$.

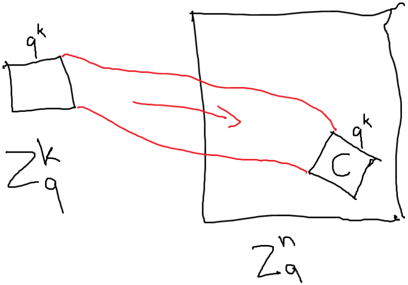
Linear code examples

- RC5 is a (binary) linear code,
- $\{0000, 1011, 0101, 1110\}$ is a (binary) linear code,
- $\{000, 001, 100, 101\}$ is a (binary) linear code,
- $\{00, 01, 10\}$ is not (missing $01 + 10$).

Encoding

For an $[n, k]$ linear code C:

- Code is a k-dimensional subspace of $\mathbb{Z}_q^n$.
- Identify up to q^k messages with elements of $\mathbb{Z}_q^k$.
- Encoding maps $\mathbb{Z}_q^k$ injectively into $\mathbb{Z}_q^n$.



Code generator matrix

We can pick a basis for the code, consisting of k vectors.
Consider a $k \times n$ **generator matrix** G whose k rows are basis vectors for the linear code C.

$$\begin{pmatrix} \text{---} \; \vec{b}_1 \; \text{---} \\ \text{---} \; \vec{b}_2 \; \text{---} \\ \vdots \\ \text{---} \; \vec{b}_k \; \text{---} \end{pmatrix}$$

- $m \mapsto mG$ maps a k-word to its encoding (codeword of C)

Linear-code encoding is just a matrix multiplication.

Parity check matrix

Every linear code has an $n \times (n-k)$ **parity check matrix** H such that:

$$cH = 0 \iff c \text{ is a codeword.}$$

Syndromes, a quick min-dist decoding

1. Receive an n-word r. Compute $s(r) = rH$ (**syndrome** of r).
2. $s(r) = rH = (c + e)H = 0 + eH = s(e)$
3. Pick the least-weight (most zero-components) vector e' satisfying $s(e') = s(r)$.
4. Decode as $r - e'$.

What we need is a precomputed mapping from syndromes to vectors e. There are q^{n-k} syndromes.

- Storage space: $O(nq^{n-k})$.
- Lookup time: $O(n - k)$.

This is a **min-dist** decoding.

Properties of linear codes

An $[n, k]$ linear code has a rate of k/n.

The minimum distance of a linear code is equal to the weight of the lowest-weight nonzero codeword.

(The Singleton bound)
The minimum distance of an $[n, k]$ linear code is $\leq n - k + 1$.

A comparison

Repetition code RC5:

- Rate $= 0.2$,
- Corrects up to 2 errors.

There exists a [5, 4] linear code with:

- Rate $= 0.8$,
- Corrects up to 2 errors.

Reed-Solomon codes

Invented in the 60s.
Family of codes still used in real life applications.
Let you scratch and touch your CDs.

- Cyclic codes.
- More algebra!

Introduction to information theory

Model

$$w \in \Sigma_{in}^* \xrightarrow[\text{channel}]{\text{add errors}} w' \in \Sigma_{out}^*$$

- Model the information source as a random variable X.

Surprisal

Surprisal is a property of a single outcome of a random variable.

- How much information we get when we learn $X = x_i$.

$$-\log_2 Pr(X = x_i) \in [0, \infty)$$

Log is the only differentiable function of $Pr(X = x_i)$*,*
that is additive for independent events.

Example: If $Pr(X = 0) = 1$ and we 'learn' that $X = 0$, we are not surprised **at all** - the surprisal is 0.

Information entropy

Information entropy is a property of a random variable.

- Expected surprisal.

$$H(X) = E[-\log_2 Pr(X = x_i)] = -\sum_i Pr(X = x_i) \log_2 Pr(X = x_i)$$

Entropy examples

- A Mobius strip coin has 0 entropy.
- A fair coin has $\frac{1}{2}\left(-\log_2 \frac{1}{2}\right) + \frac{1}{2}\left(-\log_2 \frac{1}{2}\right) = 1$ bit of entropy.
- A fair die roll will have $\log_2 6$ bits of entropy.

Properties of entropy

1. Entropy is additive for independent r.v.s.: $H(U, V) = H(U) + H(V)$.

Example: Entropy of n coin tosses is n times that of a single toss.

2. Entropy of a r.v. with n possible outcomes is $\leq \log_2 n$.

Source coding theorem

For a source with H bits of entropy,
lossless compression at less than H bits per average message is **impossible**.

Model

$$w \in \Sigma_{in}^* \xrightarrow[\text{channel}]{\text{add errors}} w' \in \Sigma_{out}^*$$

- Model the information source as a random variable X.
- The channel output Y is a random variable dependent on X.

Conditional entropy

$$H(X|Y) = E_Y[H(X|y)]$$

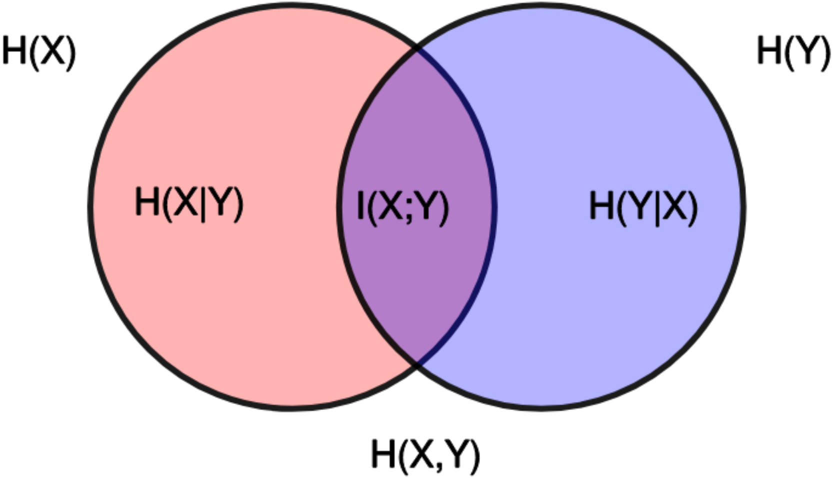
Mutual information

Mutual information $I(X, Y)$ is a property of a pair of r.v.s.

$$I(X, Y) = H(X) - H(X|Y)$$

The information shared between X and Y.

Mutual information



Channel capacity

$$w \in \Sigma_{in}^* \xrightarrow[\text{channel}]{\text{add errors}} w' \in \Sigma_{out}^*$$

$$C = \max_{P_X} I(X, Y)$$

Channel capacity $=$ mutual information between input and output maximized over all input symbols probability distributions.

A 'noisy typewriter' example

$Pr('b'|'a') = Pr('a'|'a') = 1/2, \ \ldots, \ Pr('a'|'z') = Pr('z'|'z') = 1/2.$

Typewriter's capacity $= \log_2 13$ bits:

$$
\begin{aligned}
C &= \max I(X, Y) = \max H(Y) - H(Y|X) = \max H(Y) - 1 \\
&= \log_2 26 - 1 = \log_2 13
\end{aligned}
$$

Noisy-channel coding theorem

Code's information rate

A code with M codewords of length n has **information rate** of

$$R = \frac{\log_2 M}{n} \text{ bits per transmission.}$$

The noisy-channel coding theorem

For any $\varepsilon > 0$

and a channel with capacity C

and a number $\delta \in (0, C)$,

there is a code with information rate $R \geq C - \delta$

that allows data transmission with error probability $< \varepsilon$.

No such code exists with information rate $R > C$.

A measure of success

$\lambda_i =$ prob. of incorrect decoding, given codeword x_i was sent.

Maximal probability of error:

$$\lambda_{max} = \max_i \lambda_i$$

Proof strategy

0. Fix $\varepsilon > 0$ and a rational $R \in (0, C)$.

Proof strategy

0. Fix $\varepsilon > 0$ and a rational $R \in (0, C)$.

1. Look at random codes with 2^{nR} codewords of length n.
Information rate $= \frac{1}{n} \log_2(2^{nR}) = R$.

Proof strategy

0. Fix $\varepsilon > 0$ and a rational $R \in (0, C)$.

1. Look at random codes with 2^{nR} codewords of length n.
Information rate $= \frac{1}{n} \log_2(2^{nR}) = R$.

2. Let X be the probability distribution (of input symbols), that achieves the channel capacity.

Proof strategy

0. Fix $\varepsilon > 0$ and a rational $R \in (0, C)$.

1. Look at random codes with 2^{nR} codewords of length n.
Information rate $= \frac{1}{n} \log_2(2^{nR}) = R$.

2. Let X be the probability distribution (of input symbols), that achieves the channel capacity.

3. Generate 2^{nR} codewords according to X.
(One symbol at a time, independently.)

Proof strategy

0. Fix $\varepsilon > 0$ and a rational $R \in (0, C)$.

1. Look at random codes with 2^{nR} codewords of length n.
Information rate $= \frac{1}{n} \log_2(2^{nR}) = R$.

2. Let X be the probability distribution (of input symbols),
that achieves the channel capacity.

3. Generate 2^{nR} codewords according to X.
(One symbol at a time, independently.)

4. Prove that, for each codeword c_i,
the error probability averaged over all codes is $< 2\varepsilon$.

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

$$\Longrightarrow$$

The error probability averaged over all codes and all codewords is $< 2\varepsilon$.

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

$$\Longrightarrow$$

The error probability averaged over all codes and all codewords is $< 2\varepsilon$.

$$\Longrightarrow$$

There is a code with error prob. averaged over all its codewords $< 2\varepsilon$.

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

$$\Longrightarrow$$

The error probability averaged over all codes and all codewords is $< 2\varepsilon$.

$$\Longrightarrow$$

There is a code with error prob. averaged over all its codewords $< 2\varepsilon$.

- At least half of its codewords have error probability $< \varepsilon$.

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

$$\Longrightarrow$$

The error probability averaged over all codes and all codewords is $< 2\varepsilon$.

$$\Longrightarrow$$

There is a code with error prob. averaged over all its codewords $< 2\varepsilon$.

- At least half of its codewords have error probability $< \varepsilon$.
- Remove the others from the code!

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

$$\Longrightarrow$$

The error probability averaged over all codes and all codewords is $< 2\varepsilon$.

$$\Longrightarrow$$

There is a code with error prob. averaged over all its codewords $< 2\varepsilon$.

- At least half of its codewords have error probability $< \varepsilon$.
- Remove the others from the code!
- Get a code with maximum error probability $< \varepsilon$.

Proof strategy (continued)

For each codeword c_i, the error probability averaged over all codes is $< 2\varepsilon$.

$$\Longrightarrow$$

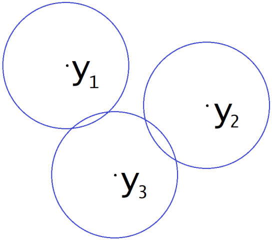
The error probability averaged over all codes and all codewords is $< 2\varepsilon$.

$$\Longrightarrow$$

There is a code with error prob. averaged over all its codewords $< 2\varepsilon$.

- At least half of its codewords have error probability $< \varepsilon$.
- Remove the others from the code!
- Get a code with maximum error probability $< \varepsilon$.
- The rate of the code drops from $\log(C)/n$ to $\log(C/2)/n$.
- A decrease by only $1/n$ which is negligible as $n \to \infty$.

Decoding scheme



We will **decode by joint typicality**, i.e.
decode an output $\vec{y}$ to a codeword $\vec{x}$ if and only if:

- $\vec{x}$ is the **unique** (only) codeword ε_2-jointly-typical with $\vec{y}$

ε_2-typicality

A sequence $\vec{x}$ of symbols from Σ is ε_2-typical (in the context of a r.v. X) if:

$$\sum_i \log Pr(x_i) \text{ is } \varepsilon_2\text{-close to its expected value.}$$

ε_2-typicality

A sequence $\vec{x}$ of symbols from Σ is ε_2-typical (in the context of a r.v. X) if:

$$\sum_i \log Pr(x_i) \text{ is } \varepsilon_2\text{-close to its expected value.}$$

$$\text{i.e. } Pr(\vec{x}) \in (2^{-nH(X)-n\varepsilon_2},\ 2^{-nH(X)+n\varepsilon_2})$$

ε_2-joint-typicality

Two sequences $\vec{x}, \vec{y}$ (same length n) of symbols from Σ_x, Σ_y are ε_2-jointly-typical (in the context of r.v.s X, Y) if:

$$\sum_i \log Pr(x_i, y_i) \text{ is } \varepsilon_2\text{-close to its expected value.}$$

$$\text{i.e. } Pr(\vec{x}, \vec{y}) \in (2^{-nH(X,Y)-n\varepsilon_2},\ 2^{-nH(X,Y)+n\varepsilon_2})$$

and both sequences are ε_2-typical on their own.

ε_2-joint-typicality - corollary

Two sequences $\vec{x}, \vec{y}$ (same length n) of symbols from Σ_x, Σ_y are ε_2-jointly-typical (in the context of r.v.s X, Y) iff:

$$Pr(\vec{x}) \in (2^{-nH(X)-n\varepsilon_2},\ 2^{-nH(X)+n\varepsilon_2}) \text{ and}$$
$$Pr(\vec{y}) \in (2^{-nH(Y)-n\varepsilon_2},\ 2^{-nH(Y)+n\varepsilon_2}) \text{ and}$$
$$Pr(\vec{x}|\vec{y}) \in (2^{-nH(X|Y)-n2\varepsilon_2},\ 2^{-nH(X|Y)+n2\varepsilon_2})$$

ε_2-joint-typicality - corollary

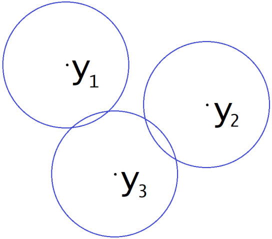
Two sequences $\vec{x}, \vec{y}$ (same length n) of symbols from Σ_x, Σ_y are ε_2-jointly-typical (in the context of r.v.s X, Y) iff:

$$Pr(\vec{x}) \in (2^{-nH(X)-n\varepsilon_2},\ 2^{-nH(X)+n\varepsilon_2}) \text{ and}$$
$$Pr(\vec{y}) \in (2^{-nH(Y)-n\varepsilon_2},\ 2^{-nH(Y)+n\varepsilon_2}) \text{ and}$$
$$Pr(\vec{x}|\vec{y}) \in (2^{-nH(X|Y)-n2\varepsilon_2},\ 2^{-nH(X|Y)+n2\varepsilon_2})$$

- There are about $2^{nH(X)}$ typical $\vec{x}$s.
- For a given $\vec{y}$ there are about $2^{nH(X|Y)}$ jointly typical $\vec{x}$s.
- Probability of an $\vec{x}$ being jointly typical to a given $\vec{y}$:

$$p \leq \frac{2^{nH(X|Y)+n2\varepsilon_2}}{2^{nH(X)-n\varepsilon_2}} = 2^{n[H(X|Y)-H(X)+3\varepsilon_2]} = 2^{-n[I(X,Y)+3\varepsilon_2]}$$

Decoding scheme



We will **decode by joint typicality**, i.e.
decode an output $\vec{y}$ to a codeword $\vec{x}$ if and only if:

- $\vec{x}$ is the **unique** (only) codeword ε_2-jointly-typical with $\vec{y}$

Possible errors

Say we transmitted $\vec{x}$. An error will occur if one of the following happens:

- $\vec{x}$ is not jointly typical with $\vec{y}$
  - Probability vanishes (for long n) below some ε_1.
- there's a different $\vec{x'}$ that's jointly typical with $\vec{y}$
  - We know the probability that a single $\vec{x'}$ is jointly typical with $\vec{y}$.

Error probability estimation

(Averaged over codes)
Probability of a codeword c_1 being incorrectly decoded:

$$
\begin{aligned}
Pr(\frown) \leq\ & Pr(c_1 \text{ not jointly typical with } \vec{y}) \\
& + Pr(c_2 \text{ being jointly typical with } \vec{y}) \\
& + Pr(c_3 \text{ being jointly typical with } \vec{y}) \\
& + \dots \\
& + Pr(c_{2^{nR}} \text{ being jointly typical with } \vec{y})
\end{aligned}
$$

Error probability estimation

(Averaged over codes)
Probability of a codeword c_1 being incorrectly decoded:

$$\begin{aligned} Pr(☹) &\leq \varepsilon_1 + (2^{nR} - 1)2^{-nI(X,Y)+n3\varepsilon_2} \\ &= \varepsilon_1 + (2^{nR} - 1)2^{-nC+n3\varepsilon_2} \\ &\leq \varepsilon_1 + (2^{nR})2^{-nC+n3\varepsilon_2} \\ &= \varepsilon_1 + 2^{-n(C-R+3\varepsilon_2)} \end{aligned}$$

Error probability estimation

(Averaged over codes)
Probability of a codeword c_1 being incorrectly decoded:

$$\begin{aligned} Pr(☹) &\leq \varepsilon_1 + (2^{nR} - 1)2^{-nI(X,Y)+n3\varepsilon_2} \\ &= \varepsilon_1 + (2^{nR} - 1)2^{-nC+n3\varepsilon_2} \\ &\leq \varepsilon_1 + (2^{nR})2^{-nC+n3\varepsilon_2} \\ &= \varepsilon_1 + 2^{-n(C-R+3\varepsilon_2)} \end{aligned}$$

And because we do have $R < C$ and $\varepsilon_1 \xrightarrow{n\to\infty} 0$:

$$Pr(☹) \xrightarrow{n\to\infty} 0$$

What we've shown

For any $\varepsilon > 0$
and a channel with capacity C
and a number $\delta \in (0, C)$,
there is a code with information rate $R \geq C - \delta$
that allows data transmission with error probability $< \varepsilon$.

Sources

- Sarah Spence Adams (Cornell University):
  Introduction to algebraic coding theory, 2008.
- Mai Vu (Tufts University):
  EE194 – Network Information Theory, Lecture 2.
- Amon Elders (Universiteit van Amsterdam):
  Shannon's Noisy-Channel Theorem, 2016.
- J.H. Van Lint:
  Introduction to coding theory, 1992.
- Wikipedia.
- codetables.de

Questions?

Thanks for the attention.

Thanks to Jasper Lee.