Information Theory - Lecture 1,2,3

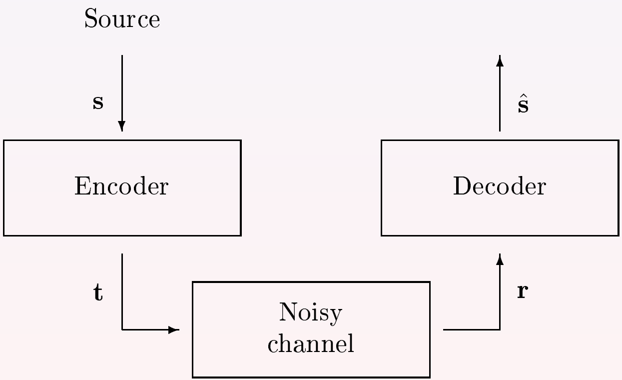
cosmin.bonchis@e-uvt.ro

October 20, 2021

Information Theory

What do you understand by Information Theory?

Communication System



Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
    - the theorem
    - state-of-the art error-correcting codes
- Source coding (Data commpression)
    - key ideas
    - optimal symbol codes
    - arithmetic coding

Bayes' theorem

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]} = \frac{P[A \cap B]}{P[B]}$$
$$= \frac{P[B|A]P[A]}{P[B|A]P[A] + P[B|\overline{A}]P[\overline{A}]}$$

- 3 cards example
- The Monty Hall Paradox

The Monty Hall Paradox

There is one BIG PRIZE behind one of doors 1, 2, 3. You chosed first the door 1. The host open door 2. Do you switch?

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]}$$

- Who is A?
- Who is B?
- $P[\text{prize behind 1}|\text{open 2}] = \frac{P[\text{open 2}|\text{prize behind 1}]P[\text{prize behind 1}]}{P[\text{open 2}]} = \frac{\frac{1}{2}\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$
- $P[\text{prize behind 2}|\text{open 2}] = \frac{P[\text{open 2}|\text{prize behind 2}]P[\text{prize behind 2}]}{P[\text{open 2}]} = \frac{0\cdot\frac{1}{3}}{\frac{1}{2}} = 0$
- $P[\text{prize behind 3}|\text{open 2}] = \frac{P[\text{open 2}|\text{prize behind 3}]P[\text{prize behind 3}]}{P[\text{open 2}]} = \frac{1\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$

The Monty Hall Paradox

There is one BIG PRIZE behind one of doors 1, 2, 3. You chosed first the door 1. The host open door 2. Do you switch?

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]}$$

- Who is A?
- Who is B?
- $P[\text{prize behind } 1|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 1]P[\text{prize behind } 1]}{P[\text{open } 2]} = \frac{\frac{1}{2}\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$
- $P[\text{prize behind } 2|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 2]P[\text{prize behind } 2]}{P[\text{open } 2]} = \frac{0\cdot\frac{1}{3}}{\frac{1}{2}} = 0$
- $P[\text{prize behind } 3|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 3]P[\text{prize behind } 3]}{P[\text{open } 2]} = \frac{1\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$

The Monty Hall Paradox

There is one BIG PRIZE behind one of doors 1, 2, 3. You chosed first the door 1. The host open door 2. Do you switch?

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]}$$

- Who is A?
- Who is B?
- $P[\text{prize behind 1}|\text{open 2}] = \frac{P[\text{open 2}|\text{prize behind 1}]P[\text{prize behind 1}]}{P[\text{open 2}]} = \frac{\frac{1}{2}\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$
- $P[\text{prize behind 2}|\text{open 2}] = \frac{P[\text{open 2}|\text{prize behind 2}]P[\text{prize behind 2}]}{P[\text{open 2}]} = \frac{0\cdot\frac{1}{3}}{\frac{1}{2}} = 0$
- $P[\text{prize behind 3}|\text{open 2}] = \frac{P[\text{open 2}|\text{prize behind 3}]P[\text{prize behind 3}]}{P[\text{open 2}]} = \frac{1\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$

The Monty Hall Paradox

There is one BIG PRIZE behind one of doors 1, 2, 3. You chosed first the door 1. The host open door 2. Do you switch?

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]}$$

- Who is A?
- Who is B?
- $P[\text{prize behind } 1|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 1]P[\text{prize behind } 1]}{P[\text{open } 2]} = \frac{\frac{1}{2}\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$
- $P[\text{prize behind } 2|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 2]P[\text{prize behind } 2]}{P[\text{open } 2]} = \frac{0\cdot\frac{1}{3}}{\frac{1}{2}} = 0$
- $P[\text{prize behind } 3|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 3]P[\text{prize behind } 3]}{P[\text{open } 2]} = \frac{1\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$

The Monty Hall Paradox

There is one BIG PRIZE behind one of doors 1, 2, 3. You chosed first the door 1. The host open door 2. Do you switch?

$$P[A|B] = \frac{P[B|A]P[A]}{P[B]}$$

- Who is A?
- Who is B?
- $P[\text{prize behind } 1|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 1]P[\text{prize behind } 1]}{P[\text{open } 2]} = \frac{\frac{1}{2}\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$
- $P[\text{prize behind } 2|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 2]P[\text{prize behind } 2]}{P[\text{open } 2]} = \frac{0\cdot\frac{1}{3}}{\frac{1}{2}} = 0$
- $P[\text{prize behind } 3|\text{open } 2] = \frac{P[\text{open } 2|\text{prize behind } 3]P[\text{prize behind } 3]}{P[\text{open } 2]} = \frac{1\cdot\frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$

References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

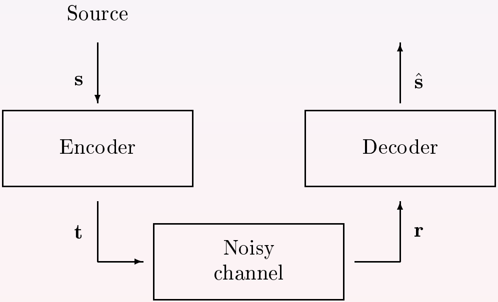
References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

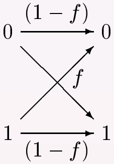
References

- C. E. Shannon: A mathematical theory of communication. Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948
- Data Compression, complete reference, David SOLOMON
- Introduction to Data Compression, Guy E. Blelloch (short)
- The Data Compression Book, Mark Nelson and Jean-loup Gailly
- Principles of Digital Communication, ROBERT G. GALLAGER (coding)
- Elements of Information Theory, Thomas M. Cover and Joy A. thomas
- Information Theory, Inference, and Learning Algorithms, David J.C. MacKay
- ...

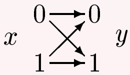
Error correction codes (Channel coding)



BSC - Binary symmetric channel

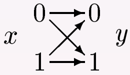


eg: $f = 0.1$



$$P(y=0\,|\,x=0) = 1-f; \quad P(y=0\,|\,x=1) = f;$$
$$P(y=1\,|\,x=0) = f; \quad P(y=1\,|\,x=1) = 1-f.$$

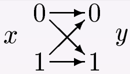
System solutions for BSC



$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

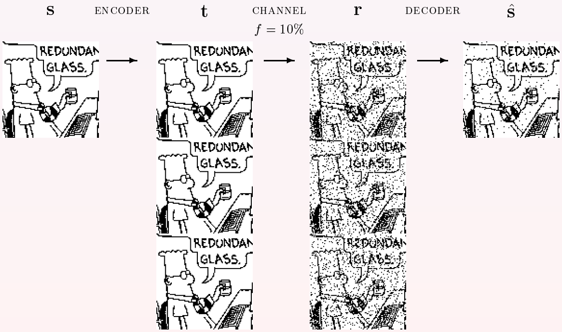
- Repetition codes: R_3, R_4, ... R_N
- (7,4) Hamming code

System solutions for BSC

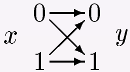


$$
\begin{aligned}
P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\
P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f.
\end{aligned}
$$

- Repetition codes: R_3



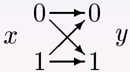
Repetition Codes



$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) = ?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

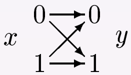
Repetition Codes



$$P(y=0\,|\,x=0) = 1-f; \quad P(y=0\,|\,x=1) = f;$$
$$P(y=1\,|\,x=0) = f; \quad P(y=1\,|\,x=1) = 1-f.$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) =?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

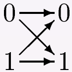
Repetition Codes



$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) = ?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

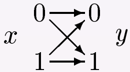
Repetition Codes

x  y

$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) = ?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

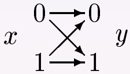
Repetition Codes



$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) = ?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

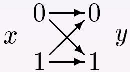
Repetition Codes



$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) = ?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

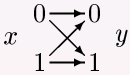
Repetition Codes



$$\begin{aligned} P(y=0\,|\,x=0) &= 1-f; & P(y=0\,|\,x=1) &= f; \\ P(y=1\,|\,x=0) &= f; & P(y=1\,|\,x=1) &= 1-f. \end{aligned}$$

- R_3
- What is a decoder for R_3?
- What is the probability of fail if we use R_3 and majority vote decoder in BSC: $P_b(\hat{s} \neq s) = ?$
- $P_b(\hat{s} \neq s) = C_3^2 \cdot f^2(1-f)$
- $P_b(\hat{s} \neq s) = \sum_{k=N/2}^{N} C_N^k f^k (1-f)^{N-k}$
- Assuming $f = 0.1$, find how many repetitions are required to get the probability of error down to 10^{-15}

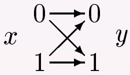
Hamming code



$$
\begin{aligned}
P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\
P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f.
\end{aligned}
$$

- (7,4) Hamming code
  - What is the probability of bit error? $P_b(\hat{s} \neq s) = ?$

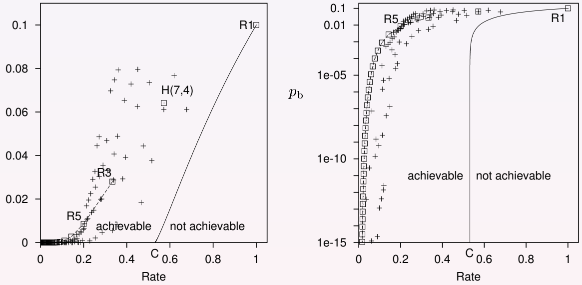
Hamming code



$$\begin{aligned} P(y=0 \mid x=0) &= 1-f; & P(y=0 \mid x=1) &= f; \\ P(y=1 \mid x=0) &= f; & P(y=1 \mid x=1) &= 1-f. \end{aligned}$$

- (7,4) Hamming code
  - What is the probability of bit error? $P_b(\hat{s} \neq s) = ?$

Shannon's theorem



$$\textit{communication rate} = \frac{\#\ \textit{of bits sended}}{\#\ \textit{of bits transmited}}$$

For any channel:

Reliable communication is posible at rates up to C.

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
  - the theorem
  - state-of-the art error-correcting codes
- Source coding (Data commpression)
  - key ideas
  - optimal symbol codes
  - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
    - the theorem
    - state-of-the art error-correcting codes
- Source coding (Data commpression)
    - key ideas
    - optimal symbol codes
    - arithmetic coding

Information Theory

- Noisy-channel coding (Channel commpression)
    - the theorem
    - state-of-the art error-correcting codes
- Source coding (Data commpression)
    - key ideas
    - optimal symbol codes
    - arithmetic coding

Read this !

Information th*ory is a bran*h of a*plied ma*hematics *nd electri*al*engineer*ng involvi*g the quant*fication of inform*tion. Informat*on theory w*s deve*oped by Clau*e E. Shan*on to find *und*men*al limi*s on s*gnal*proces*ing oper*tio*s su*h a* comp*es**ng d*ta an* *n re*i*bly storing a*d co***nicating d*t*. Sin*e i*s in*ep*ion *t **s broa*e*ed to f**d ap**icati*ns ** ma*y o**er a***s, incl*d*ng s*at**t*cal i*fer*nce, n*tu*al lan**age*proc*ss*ng, c**pt*gr*p*y ge**ra*ly, ne*w**ks ***er t*an co**u*ic*tio* n*t**rks a* *n*ne***b*ol*gy, th**evol*tion *nd **nc*ion *f**olec*lar c*d*s, *o**l se*e*t*on i* *c*l*gy, t***m*l p***i*s, q**nt*m c****i*g, pl****i*m d*t*****n *******r f***s ***d*** a*****is.

Read this !

Information th*ory is a bran*h of a*plied ma*hematics *nd electri*al*engineer*ng involvi*g the quant*fication of inform*tion. Informat*on theory w*s deve*oped by Clau*e E. Shan*on to find *und*men*al limi*s on s*gnal*proces*ing oper*tio*s su*h a* comp*es**ng d*ta an* *n re*i*bly storing a*d co***nicating d*t*. Sin*e i*s in*ep*ion *t **s broa*e*ed to f**d ap**icati*ns ** ma*y o**er a***s, incl*d*ng s*at**t*cal i*fer*nce, n*tu*al lan**age*proc*ss*ng, c**pt*gr*p*y ge**ra*ly, ne*w**ks ***er t*an co**u*ic*tio* n*t**rks a* *n*ne***b*ol*gy, th**evol*tion *nd **nc*ion *f**olec*lar c*d*s, *o**l se*e*t*on i* *c*l*gy, t***m*l p***i*s, q**nt*m c****i*g, pl****i*m d*t*****n *******r f***s ***d*** a*****is.

Information theory is a branch of applied mathematics and electrical engineering involving the quantification of information. Information theory was developed by Claude E. Shannon to find fundamental limits on signal processing operations such as compressing data and on reliably storing and communicating data. Since its inception it has broadened to find applications in many other areas, including statistical inference, natural language processing, cryptography generally, networks other than communication networks — as in neurobiology, the evolution and function of molecular codes, model selection in ecology, thermal physics, quantum computing, plagiarism detection and other forms of data analysis.

ideas

- real sources has redundancy.
- describe an information source !
- formal definition = *random variable*
- ideal sources

A simple redundant source - a bent coin

01000100000001000010000100101000000001001010000000
00010101010000000001010101001000000010101001001000
00000000010100101010100000001000100000001010101000
00000010100100000000000000000000000000110000000000
00000000010101000001111000000000000000000000110010
00001000000000000000100100000000000100100100000000
00101001000000000001110000000100010100000001011000
00000101001000000011010000000000010111110000000001
11000001110000000000000001010100100000000001000110
00000000101010110100000000000000010010101000000010

$$p_1 = 0.1$$

How to compress a redundant file ?

01000100000000100001000001001010000000001001010000000000
000101010100000000000101010100100000000010101001001000
000000000101001010101000000000100010000000001010101000
00000010100100000000000000000000000000000011000000000000
000000000101010000000111100000000000000000000000000110010

$N = 1000$ tosses of a bent coin with $p_1 = 0.1$

- How to measure the information content?
- How much compression should we expect is possible?

How to compress a redundant file ?

0100010000000100001000001001010000000010010100000000
0001010101000000000010101010010000000010101001001000
0000000001010010101010000000010001000000001010101000
0000001010010000000000000000000000000000110000000000
0000000001010100000011110000000000000000000000110010

$N = 1000$ tosses of a bent coin with $p_1 = 0.1$

- How to measure the information content?
- How much compression should we expect is possible?

How to compress a redundant file ?

0100010000000100001000001001010000000001001010000000
0001010101000000000001010101001000000000101010010010000
0000000001010010101010000000001000100000000101010101000
0000001010010000000000000000000000000000000110000000000
000000000101010000000111100000000000000000000000000110010

$N = 1000$ tosses of a bent coin with $p_1 = 0.1$

- How to measure the information content?
- How much compression should we expect is possible?

How to measure information content?

- Let $X = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix}$ be a random variable.
- The Shannon information content of an outcome

$$h(x \quad = \quad a_i) = log_2(\frac{1}{P(x = a_i)})$$

is a sensible measure of information content.

- The Entropy

$$H(X) \quad = \quad \sum_x P(x) log_2(\frac{1}{P(x)})$$

is a sensible measure of expected information content.

The weighing problem

You are given 12 balls and a two-pan balance to use:

- all equal in weight except for one that is either heavier or lighter.
- in each use of the balance you may put any number balls on the balance
- there are three possible outcomes:
  - either the weights are equal,
  - or the balls on the left are heavier,
  - or the balls on the left are lighter.

Design a strategy to determine which is the odd ball and whether it is heavier or lighter than the others in as few uses of the balance as possible.

The Entropy

1. $H(X) \geq 0$
2. $H(X) \leq \log_2 n = H(X|_{p_i=\frac{1}{n}})$
3. $H_b(X) = \log_b 2 \cdot H_2(X)$
4. Joint entropy: $H(X,Y) = H(X) + H(Y)$ iff independents
5. Conditional entropy: $H(X|Y) = \sum_{x \in X} p(x) \cdot H(Y|X = x)$
6. Relative entropy: $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ very important!

The Entropy

1. $H(X) \geq 0$
2. $H(X) \leq \log_2 n = H(X|_{p_i=\frac{1}{n}})$
3. $H_b(X) = \log_b 2 \cdot H_2(X)$
4. Joint entropy: $H(X,Y) = H(X) + H(Y)$ iff independents
5. Conditional entropy: $H(X|Y) = \sum_{x \in X} p(x) \cdot H(Y|X=x)$
6. Relative entropy: $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ very important!

The Entropy

1. $H(X) \geq 0$
2. $H(X) \leq \log_2 n = H(X|_{p_i=\frac{1}{n}})$
3. $H_b(X) = \log_b 2 \cdot H_2(X)$
4. Joint entropy: $H(X,Y) = H(X) + H(Y)$ iff independents
5. Conditional entropy: $H(X|Y) = \sum_{x \in X} p(x) \cdot H(Y|X=x)$
6. Relative entropy: $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ very important!

The Entropy

1. $H(X) \geq 0$
2. $H(X) \leq \log_2 n = H(X|_{p_i = \frac{1}{n}})$
3. $H_b(X) = \log_b 2 \cdot H_2(X)$
4. Joint entropy: $H(X, Y) = H(X) + H(Y)$ iff independents
5. Conditional entropy: $H(X|Y) = \sum_{x \in X} p(x) \cdot H(Y|X = x)$
6. Relative entropy: $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ very important!

The Entropy

1. $H(X) \geq 0$
2. $H(X) \leq \log_2 n = H(X|_{p_i=\frac{1}{n}})$
3. $H_b(X) = \log_b 2 \cdot H_2(X)$
4. Joint entropy: $H(X, Y) = H(X) + H(Y)$ iff independents
5. Conditional entropy: $H(X|Y) = \sum_{x \in X} p(x) \cdot H(Y|X = x)$
6. Relative entropy: $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ very important!

The Entropy

1. $H(X) \geq 0$
2. $H(X) \leq \log_2 n = H(X|_{p_i=\frac{1}{n}})$
3. $H_b(X) = \log_b 2 \cdot H_2(X)$
4. Joint entropy: $H(X, Y) = H(X) + H(Y)$ iff independents
5. Conditional entropy: $H(X|Y) = \sum_{x \in X} p(x) \cdot H(Y|X = x)$
6. Relative entropy: $D(p \parallel q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ very important!

The weighing problem

- How many weighings are enough?
- How many weighings if you have 13 balls?
- $N = \frac{3^w - 3}{2}$

The weighing problem

- How many weighings are enough?
- How many weighings if you have 13 balls?
- $N = \frac{3^w - 3}{2}$

The weighing problem

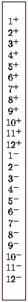


Figure: Optimal solution

The weighing problem

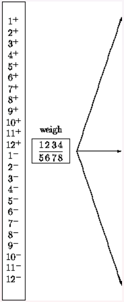


Figure: Optimal solution

The weighing problem

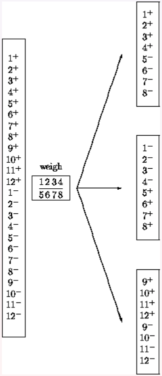


Figure: Optimal solution

The weighing problem

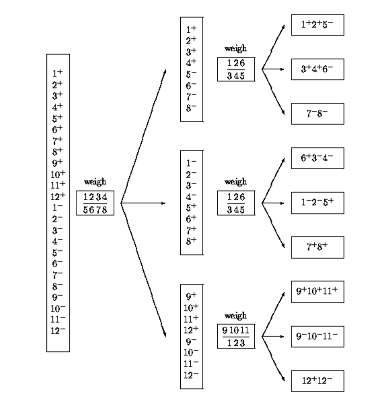


Figure: Optimal solution

The weighing problem

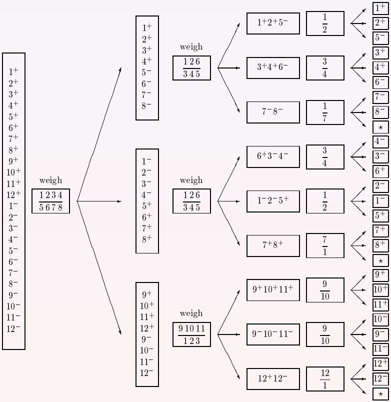


Figure: Optimal solution

How to measure information content?

- The Shannon information content of an outcome

$$h(X = x_i) = log_2(\frac{1}{P(X = x_i)})$$

is a sensible measure of information content.

- The Entropy

$$H(X) = \sum_x P(x) log_2 \left(\frac{1}{P(x)}\right) = -\sum_x P(x) log_2 P(x) = -\sum_{i=1}^{n} p_i log_2 p_i$$

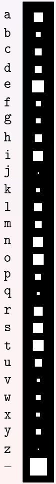
is a sensible measure of expected information content.

Uniform codes

<table>
<tr><td rowspan="8">0</td><td rowspan="4">00</td><td rowspan="2">000</td><td>0000</td></tr>
<tr><td>0001</td></tr>
<tr><td rowspan="2">001</td><td>0010</td></tr>
<tr><td>0011</td></tr>
<tr><td rowspan="4">01</td><td rowspan="2">010</td><td>0100</td></tr>
<tr><td>0101</td></tr>
<tr><td rowspan="2">011</td><td>0110</td></tr>
<tr><td>0111</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">10</td><td rowspan="2">100</td><td>1000</td></tr>
<tr><td>1001</td></tr>
<tr><td rowspan="2">101</td><td>1010</td></tr>
<tr><td>1011</td></tr>
<tr><td rowspan="4">11</td><td rowspan="2">110</td><td>1100</td></tr>
<tr><td>1101</td></tr>
<tr><td rowspan="2">111</td><td>1110</td></tr>
<tr><td>1111</td></tr>
</table>

Symbol codes

i	a_i	p_i
1	a	0.0575
2	b	0.0128
3	c	0.0263
4	d	0.0285
5	e	0.0913
6	f	0.0173
7	g	0.0133
8	h	0.0313
9	i	0.0599
10	j	0.0006
11	k	0.0084
12	l	0.0335
13	m	0.0235
14	n	0.0596
15	o	0.0689
16	p	0.0192
17	q	0.0008
18	r	0.0508
19	s	0.0567
20	t	0.0706
21	u	0.0334
22	v	0.0069
23	w	0.0119
24	x	0.0073
25	y	0.0164
26	z	0.0007
27	–	0.1928



some definitions

Definition

A *code (source code)* C for a random variable X is a mapping from $\{x_1, x_2, \ldots, x_n\}$ to $\mathcal{D}*$the set of finite-length strings of symbols from an alphabet of length D.

Note:
- $C(x)$ the **codeword** of x
- $l(x)$ the length of codeword $C(x)$

Definition

The expectation length $L(C)$ of a source code $C(X)$ for a random variable X with probability mass function $p(x)$is given by:

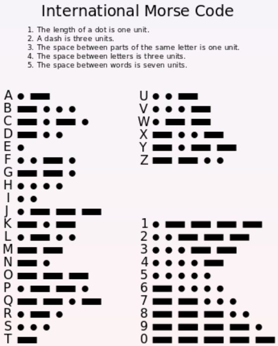
$$L(C) = \sum_{x_i} p(x_i) l(x_i) = \sum_i p_i l_i$$

source code examples

Example

ASCII codes
- 96 printable keyboard characters
- $\log(96) = ?$

Symbol code example



Definition

A code is called a *prefix code* or *instantaneous code (prefix-free code)* if no codeword is a prefix of any other codeword.

Kraft-McMillan Inequality.

Theorem

*For any **uniquely decodable code** C*

$$\sum_{x \in C} 2^{-l(x)} \leq 1,$$

where $l(x)$ is the length of the codeword. Also, for any set of lengths L Âsuch that: $\sum_{l \in L} 2^{-l} \leq 1$, there is a prefix code C of the same size such that $l(x) = l_{i, i \in [1, \ldots, |L|]}$

Prefix code application

- **UTF-8 :** Universal Character Set Transformation Format—8-bit
  - defined in 1992 as an extention for ASCII
  - is a variable-width encoding
  - default character encoding in operating systems, programming languages, APIs, and software applications
  - labelled "Unicode"
  - the most common encoding for HTML files

Prefix code application

- **UTF-8 :** Universal Character Set Transformation Format—8-bit
  - defined in 1992 as an extention for ASCII
  - is a variable-width encoding
  - default character encoding in operating systems, programming languages, APIs, and software applications
  - labelled "Unicode"
  - the most common encoding for HTML files

Prefix code application

- **UTF-8** : Universal Character Set Transformation Format—8-bit
  - defined in 1992 as an extention for ASCII
  - is a variable-width encoding
  - default character encoding in operating systems, programming languages, APIs, and software applications
  - labelled "Unicode"
  - the most common encoding for HTML files

Prefix code application

- **UTF-8** : Universal Character Set Transformation Format—8-bit
  - defined in 1992 as an extention for ASCII
  - is a variable-width encoding
  - default character encoding in operating systems, programming languages, APIs, and software applications
  - labelled "Unicode"
  - the most common encoding for HTML files

Prefix code application

- **UTF-8** : Universal Character Set Transformation Format—8-bit
  - defined in 1992 as an extention for ASCII
  - is a variable-width encoding
  - default character encoding in operating systems, programming languages, APIs, and software applications
  - labelled "Unicode"
  - the most common encoding for HTML files

Prefix code application

- **UTF-8** : Universal Character Set Transformation Format—8-bit
  - defined in 1992 as an extention for ASCII
  - is a variable-width encoding
  - default character encoding in operating systems, programming languages, APIs, and software applications
  - labelled "Unicode"
  - the most common encoding for HTML files

UTF-8, 16, 32

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	U+0000	U+007F	1	0xxxxxxx					
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx				
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx			
21	U+10000	U+1FFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
26	U+200000	U+3FFFFFF	5	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
31	U+4000000	U+7FFFFFFF	6	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

Source Coding Theorem

Theorem

There exists a variable-length encoding C of an ensemble X such that the average length of an encoded symbol $L(C,X)$ satify:

$$H(X) \leq \quad L(C,X) \quad < H(X) + 1.$$