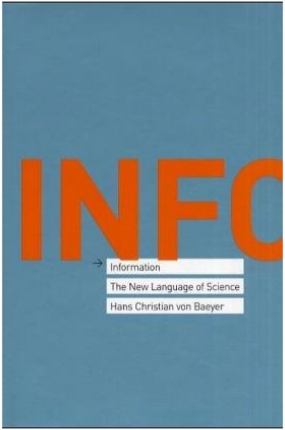
Information-Theoretic Modeling

Teemu Roos

Department of Computer Science, University of Helsinki

Fall 2009



“Whether on the internet, encoded in radio waves or coursing through wires, information is all around us. Our senses record it, our brains process it and our genes pass it on. But what exactly is information? Can it be analysed and measured? In this extraordinary book, Hans Christian von Baeyer illuminates a concept that could soon become as central to science as space, time mass or energy.”

1. Administrative issues
    - Course details
    - Prerequisites
    - What do I need to do?
    - Grading

1. Administrative issues
   - Course details
   - Prerequisites
   - What do I need to do?
   - Grading
2. Overview of Contents
   - What is Information?
   - Why Information?
   - Information vs. Complexity
   - Information Theory

1. Administrative issues
   - Course details
   - Prerequisites
   - What do I need to do?
   - Grading
2. Overview of Contents
   - What is Information?
   - Why Information?
   - Information vs. Complexity
   - Information Theory
3. Compression
   - Dots and Dashes
   - Codes as Mappings
   - Data Compression
   - Information vs. Complexity (contd.)

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.
- 4 credit units.

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.
- 4 credit units.
- Lectures: 8.9.–16.10. Tue & Fri 10–12 in C222.

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.
- 4 credit units.
- Lectures: 8.9.–16.10. Tue & Fri 10–12 in C222.
- Exercises: 15.9.–13.10. Tue (or Fri?) 12–14 in B119.

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.
- 4 credit units.
- Lectures: 8.9.–16.10. Tue & Fri 10–12 in C222.
- Exercises: 15.9.–13.10. Tue (or Fri?) 12–14 in B119.
- Instructor: **Teemu Roos**, A322,
  `teemu.roos at cs.helsinki.fi`
  (Pls. make an appointment).

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.
- 4 credit units.
- Lectures: 8.9.–16.10. Tue & Fri 10–12 in C222.
- Exercises: 15.9.–13.10. Tue (or Fri?) 12–14 in B119.
- Instructor: **Teemu Roos**, A322,
  `teemu.roos at cs.helsinki.fi`
  (Pls. make an appointment).
- Course assistant: **Anupam Arohi**.
  `anupam.arohi at cs.helsinki.fi`

582650, Information-Theoretic Modeling

- An advanced studies (laudatur) course.
- *Algorithms and Machine Learning* (was *Intelligent Systems*) sub-programme, optional.
- 4 credit units.
- Lectures: 8.9.–16.10. Tue & Fri 10–12 in C222.
- Exercises: 15.9.–13.10. Tue (or Fri?) 12–14 in B119.
- Instructor: **Teemu Roos**, A322,
  teemu.roos at cs.helsinki.fi
  (Pls. make an appointment).
- Course assistant: **Anupam Arohi**.
  anupam.arohi at cs.helsinki.fi
- www.cs.helsinki.fi/group/cosco/Teaching/Information/2009/

Resources

There is no required textbook on the course, but the following are recommended.

- Highly recommended: Cover & Thomas, *Elements of Information Theory*,
- MacKay, *Information Theory, Inference and Learning Algorithms*,
- Grünwald, *The Minimum Description Length Principle*,
- Solomon, *Data Compression: The Complete Reference*.

Resources

There is no required textbook on the course, but the following are recommended.

- Highly recommended: Cover & Thomas, *Elements of Information Theory*,
- MacKay, *Information Theory, Inference and Learning Algorithms*,
- Grünwald, *The Minimum Description Length Principle*,
- Solomon, *Data Compression: The Complete Reference*.

Copies of required material will be made available at the lectures, and afterwards in the *course folder* in room C127.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.
- This course is a prerequisite.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.
- This course is a prerequisite.
- Together they replace the old *Three Concepts: Information* course — both old and new version cannot be included in your degree.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.
- This course is a prerequisite.
- Together they replace the old *Three Concepts: Information* course — both old and new version cannot be included in your degree.
- Groups of 2–3 persons.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.
- This course is a prerequisite.
- Together they replace the old *Three Concepts: Information* course — both old and new version cannot be included in your degree.
- Groups of 2–3 persons.
- Task: compress data.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.
- This course is a prerequisite.
- Together they replace the old *Three Concepts: Information* course — both old and new version cannot be included in your degree.
- Groups of 2–3 persons.
- Task: compress data.
- Best compressor wins! Intermediate results annonced periodically.

582651, Project in Information-Theoretic Modeling

There is also a related project:

- 2 credit units.
- Period II.
- This course is a prerequisite.
- Together they replace the old *Three Concepts: Information* course — both old and new version cannot be included in your degree.
- Groups of 2–3 persons.
- Task: compress data.
- Best compressor wins! Intermediate results annonced periodically.
- Programming + report.

Prerequisites

No formal prerequisites **but** you will need

Prerequisites

No formal prerequisites **but** you will need

- Calculus: integrals, derivatives, convergence, ...

Prerequisites

No formal prerequisites **but** you will need

- Calculus: integrals, derivatives, convergence, ...
- Probability theory: joint & conditional distributions, expectations, law of large numbers, ...

Prerequisites

No formal prerequisites **but** you will need

- Calculus: integrals, derivatives, convergence, ...
- Probability theory: joint & conditional distributions, expectations, law of large numbers, ...
- Programming: language is up to you (but need to work in groups in project).

What do I need to do?

- Weekly exercises:

What do I need to do?

- Weekly exercises:
  - Mathematical problems.

What do I need to do?

- Weekly exercises:
    - Mathematical problems.
    - Programming tasks.

What do I need to do?

- Weekly exercises:
  - Mathematical problems.
  - Programming tasks.

What do I need to do?

- Weekly exercises:
  - Mathematical problems.
  - Programming tasks.
- Final exam (date TBA).

What do I need to do?

- Weekly exercises:
  - Mathematical problems.
  - Programming tasks.
- Final exam (date TBA).

You do *not* have to attend the classes, unless otherwise stated. However, we recommend that you do. (At least off-line.)

What do I need to do?

- Weekly exercises:
  - Mathematical problems.
  - Programming tasks.
- Final exam (date TBA).

You do *not* have to attend the classes, unless otherwise stated. However, we recommend that you do. (At least off-line.)

If you find that the course is not for you, please let us know *as soon as possible*. There are people willing to take your place.

Grading

The course grading is based on:

1. Exercises (40 %)

Grading

The course grading is based on:

1. Exercises (40 %)
2. Exam (60 %)

Grading

The course grading is based on:

1. Exercises (40 %)
2. Exam (60 %)

Minimum 50 % of exercises have to be solved (or at least seriously attempted).

1. Administrative issues
   - Course details
   - Prerequisites
   - What do I need to do?
   - Grading

2. Overview of Contents
   - What is Information?
   - Why Information?
   - Information vs. Complexity
   - Information Theory

3. Compression
   - Dots and Dashes
   - Codes as Mappings
   - Data Compression
   - Information vs. Complexity (contd.)

What is Information?

- Etymology: *informare* = give form, 14th century.

What is Information?

- Etymology: *informare* = give form, 14th century.
- *knowledge [...], intelligence, news, facts, data, [...], (as nucleotides in DNA or binary digits in a computer program) [...], a signal [...], a numerical quantity that measures the uncertainty in the outcome of an experiment to be performed.* (source: Merriam-Webster).

What is Information?

- Etymology: *informare* = give form, 14th century.
- *knowledge [...], intelligence, news, facts, data, [...], (as nucleotides in DNA or binary digits in a computer program) [...], a signal [...], a numerical quantity that measures the uncertainty in the outcome of an experiment to be performed.* (source: Merriam-Webster).
- Data $<$ Information $<$ Knowledge.

What is Information?

- Etymology: *informare* = give form, 14th century.
- *knowledge [...], intelligence, news, facts, data, [...], (as nucleotides in DNA or binary digits in a computer program) [...], a signal [...], a numerical quantity that measures the uncertainty in the outcome of an experiment to be performed.* (source: Merriam-Webster).
- Data < Information < Knowledge.
- Information technology.

What is Information?

- Etymology: *informare* = give form, 14th century.
- *knowledge [...], intelligence, news, facts, data, [...], (as nucleotides in DNA or binary digits in a computer program) [...], a signal [...], a numerical quantity that measures the uncertainty in the outcome of an experiment to be performed.* (source: Merriam-Webster).
- Data $<$ Information $<$ Knowledge.
- Information technology.
- Physical information.

What is Information?

- Etymology: *informare* = give form, 14th century.
- *knowledge [...], intelligence, news, facts, data, [...], (as nucleotides in DNA or binary digits in a computer program) [...], a signal [...], a numerical quantity that measures the uncertainty in the outcome of an experiment to be performed.* (source: Merriam-Webster).
- Data < Information < Knowledge.
- Information technology.
- Physical information.
- This course: measuring *the amount* of information in data, and using such measures for automatically buiding *models*.

Why Information?

- The amount of information around us is exploding – internet!

Why Information?

- The amount of information around us is exploding – internet!
- Need to *store, transmit, and process* information efficiently.

Why Information?

- The amount of information around us is exploding – internet!
- Need to *store, transmit, and process* information efficiently.
- Wish to *understand* more and more complex phenomena.

Why Information?

- The amount of information around us is exploding – internet!
- Need to *store, transmit, and process* information efficiently.
- Wish to *understand* more and more complex phenomena.
- Computer science: make things automatic (intelligent).

Information vs. Complexity

Is complexity the same as information?

Information vs. Complexity

Is complexity the same as information?

Is there a lot of *information* in a random string? **No.**

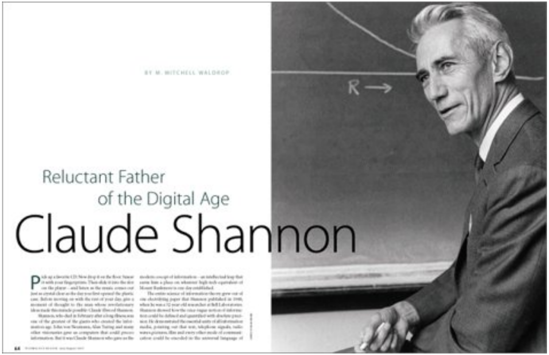
Information vs. Complexity

Is complexity the same as information?

Is there a lot of *information* in a random string? **No.**

$$\begin{aligned} \textit{Complexity} &= \textit{Information} + \textit{Noise} \\ &= \textit{Regularity} + \textit{Randomness} \\ &= \textit{Algorithm} + \textit{Compressed file} \end{aligned}$$

Information Theory



BY M. MITCHELL WALDROP

Reluctant Father
of the Digital Age

Claude Shannon

”The real birth of modern information theory can be traced to the publication in 1948 of Claude Shannon's *“The Mathematical Theory of Communication”* in the Bell System Technical Journal. ” (Encyclopædia Britannica)

Course Topics

Information Theory:

- entropy and information, bits,

Course Topics

Information Theory:

- entropy and information, bits,
- compression,

Course Topics

Information Theory:

- entropy and information, bits,
- compression,
- error correction.

Course Topics

Information Theory:

- entropy and information, bits,
- compression,
- error correction.

Fundamental limits (mathematical and statistical) and practice (computer science).

Course Topics

Information Theory:

- entropy and information, bits,
- compression,
- error correction.

Fundamental limits (mathematical and statistical) and practice (computer science).

Modeling:

Course Topics

Information Theory:

- entropy and information, bits,
- compression,
- error correction.

Fundamental limits (mathematical and statistical) and practice (computer science).

Modeling:

- statistical models,

Course Topics

Information Theory:

- entropy and information, bits,
- compression,
- error correction.

Fundamental limits (mathematical and statistical) and practice (computer science).

Modeling:

- statistical models,
- complexity (in data and models),

Course Topics

Information Theory:

- entropy and information, bits,
- compression,
- error correction.

Fundamental limits (mathematical and statistical) and practice (computer science).

Modeling:

- statistical models,
- complexity (in data and models),
- over-fitting, Occam's Razor, and MDL Principle.

1. Administrative issues
   - Course details
   - Prerequisites
   - What do I need to do?
   - Grading
2. Overview of Contents
   - What is Information?
   - Why Information?
   - Information vs. Complexity
   - Information Theory
3. Compression
   - Dots and Dashes
   - Codes as Mappings
   - Data Compression
   - Information vs. Complexity (contd.)

Coding Game

Form groups of 3–4 persons. Each group constructs a *code* for the letters A–Z by using as *code-words* unique sequences of dots • and dashes (—) like “•”, “— •”, “ — • — —”, etc.

A ________
B ________
C ________
D ________
E ________
F ________
G ________
H ________
I ________
J ________
K ________
L ________
M ________
N ________
O ________
P ________
Q ________
R ________
S ________
T ________
U ________
V ________
W ________
X ________
Y ________
Z ________

Coding Game

Use your code to *encode* the message
"`WHAT DOES THIS HAVE TO DO WITH INFORMATION`".

Coding Game

Use your code to *encode* the message
"`WHAT DOES THIS HAVE TO DO WITH INFORMATION`".

Now count how long the encoded message is using the rule:

- A dot •: 1 units.
- A dash —: 2 units.
- A space between words: 2 units.

Coding Game

Use your code to *encode* the message
"`WHAT DOES THIS HAVE TO DO WITH INFORMATION`".

Now count how long the encoded message is using the rule:

- A dot •: 1 units.
- A dash —: 2 units.
- A space between words: 2 units.

• • • — — — • • •: $1 + 1 + 1 + 2 + 2 + 2 + 1 + 1 + 1 = 12$.

Coding Game

Use your code to *encode* the message
“`WHAT DOES THIS HAVE TO DO WITH INFORMATION`”.

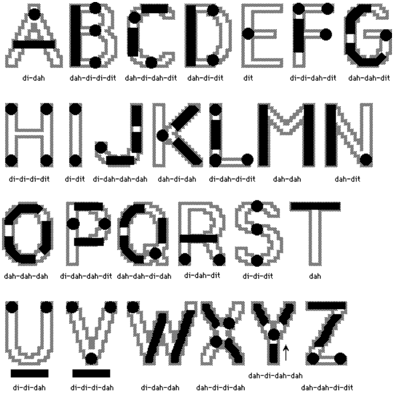
Now count how long the encoded message is using the rule:

- A dot •: 1 units.
- A dash —: 2 units.
- A space between words: 2 units.

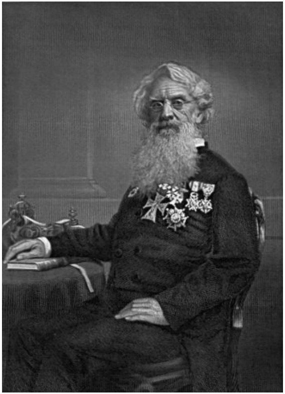
• • • — — — • • •: $1 + 1 + 1 + 2 + 2 + 2 + 1 + 1 + 1 = 12$.

The *coding rate* of your code is the length of the encoded message divided by the length of the original message, including spaces (42).

Coding Game



© 1989 A.G. Reinhold.



Samuel F.M. Morse (1791–1872)

Coding Game

```
WHAT DOES THIS HAVE TO DO WITH INFORMATION
```

Coding Game

```
WHAT DOES THIS HAVE TO DO WITH INFORMATION

.-- .... .- -   -.. --- . ...   - .... .. ...
.... .- ...- .   - ---   -.. ---   .-- .. - ....
.. -. ..-. --- .-. -- .- - .. --- -.
```

Coding Game

```
WHAT DOES THIS HAVE TO DO WITH INFORMATION
```

```
.-- .... .- -   -.. --- . ...   - .... .. ...
.... .- ...- .   - ---   -.. ---   .-- .. - ....
.. -. ..-. --- .-. -- .- - .. --- -.
```

51 dots, 36 dashes, 7 spaces: $51 + 72 + 14 = 137$ units.

Coding Game

```
WHAT DOES THIS HAVE TO DO WITH INFORMATION
```

```
.-- .... .- -   -.. --- . ...   - .... .. ...
.... .- ...- .   - ---   -.. ---   .-- .. - ....
.. -. ..-. --- .-. -- .- - .. --- -.
```

51 dots, 36 dashes, 7 spaces: $51 + 72 + 14 = 137$ units.

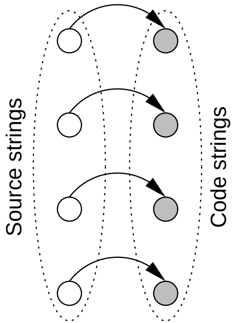
Morse code

Coding rate: $\frac{137}{42} \approx 3.26$

Did you do better or worse? Why?

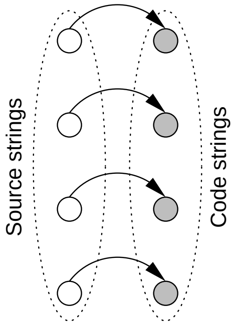
Codes as Mappings

Lossless compression:
injective mapping

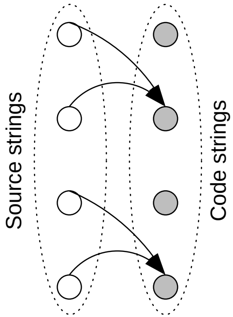


Codes as Mappings

Lossless compression:
injective mapping

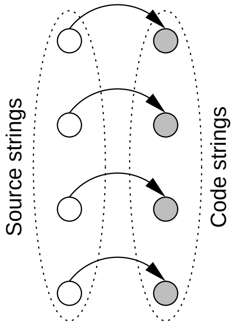


Lossy compression:
non-injective mapping

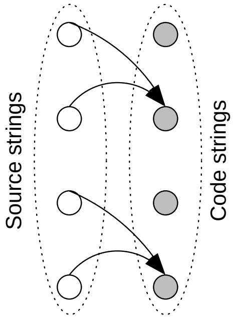


Codes as Mappings

Lossless compression:
injective mapping



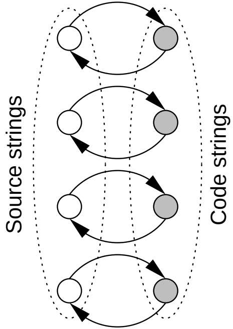
Lossy compression:
non-injective mapping



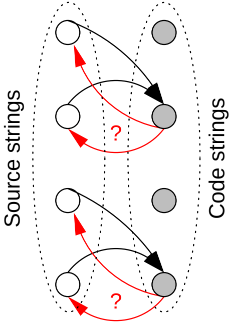
Only *lossless* codes are *uniquely decodable*.

Codes as Mappings

Lossless compression:
injective mapping



Lossy compression:
non-injective mapping



Only *lossless* codes are *uniquely decodable*.

Examples

general purpose **`gzip`** **`bzip`**

Examples

general purpose **`gzip`**

`bzip`

image **`png`**

`jpeg`

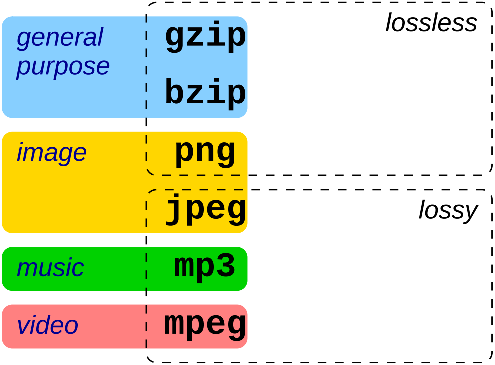
Examples

general purpose	`gzip` `bzip`
image	`png` `jpeg`
music	`mp3`

Examples

general purpose	**`gzip`** **`bzip`**
image	**`png`** **`jpeg`**
music	**`mp3`**
video	**`mpeg`**

Examples



lossless

general purpose: gzip, bzip

image: png

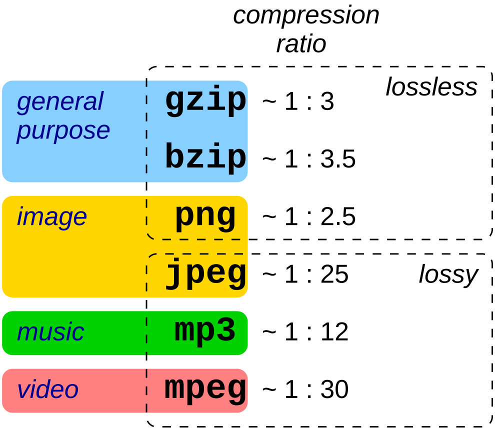
image: jpeg

lossy

music: mp3

video: mpeg

Examples



		compression ratio	
general purpose	**gzip**	~ 1 : 3	*lossless*
	bzip	~ 1 : 3.5	
image	**png**	~ 1 : 2.5	
	jpeg	~ 1 : 25	*lossy*
music	**mp3**	~ 1 : 12	
video	**mpeg**	~ 1 : 30	

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.The number of binary code strings of length less than $n - k$ is $2^0 + 2^1 + 2^2 + \ldots + 2^{n-k-1}$

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.The number of binary code strings of length less than $n - k$ is $2^0 + 2^1 + 2^2 + \ldots + 2^{n-k-1} = 2^{n-k} - 1$.

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.The number of binary code strings of length less than $n-k$ is $2^0 + 2^1 + 2^2 + \ldots + 2^{n-k-1} = 2^{n-k} - 1$. Thus the ratio is

$$\frac{2^{n-k}-1}{2^n} < \frac{2^{n-k}}{2^n} = 2^{-k}.$$

□

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.The number of binary code strings of length less than $n - k$ is $2^0 + 2^1 + 2^2 + \ldots + 2^{n-k-1} = 2^{n-k} - 1$. Thus the ratio is

$$\frac{2^{n-k} - 1}{2^n} < \frac{2^{n-k}}{2^n} = 2^{-k}.$$

□

Less than 50 % of files are compressible by more than one bit.

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.The number of binary code strings of length less than $n - k$ is $2^0 + 2^1 + 2^2 + \ldots + 2^{n-k-1} = 2^{n-k} - 1$. Thus the ratio is

$$\frac{2^{n-k} - 1}{2^n} < \frac{2^{n-k}}{2^n} = 2^{-k}.$$

□

Less than 1 % of files are compressible by more than 7 bits.

Compression

Is it always possible to compress data?

Theorem

The proportion of binary strings compressible by more than k bits is less than 2^{-k}.

Proof. For all $n \geq 1$, the number of binary strings of length n is 2^n.The number of binary code strings of length less than $n-k$ is $2^0 + 2^1 + 2^2 + \ldots + 2^{n-k-1} = 2^{n-k} - 1$. Thus the ratio is

$$\frac{2^{n-k}-1}{2^n} < \frac{2^{n-k}}{2^n} = 2^{-k}.$$

□

Less than 0.000000000000000000000000000001 % of files are compressible by 100 bits.

How is it possible?

Why was the compression ratio greater than one in all the examples we saw?

What are those rare files that are compressible?

Why are the files we use in practice so often compressible?

Compression

```
echo <x> | gzip - | wc -c        # multiply by 8 for bits
```

Source string, x		$\ell(C(x))$	**ratio**
$aaa \ldots a$	$(10000 \times a)$	368	27.2 : 1.

Compression

`echo <x> | gzip - | wc -c` # multiply by 8 for bits

Source string, x		$\ell(C(x))$	**ratio**
$aaa\ldots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb\ldots$	(10000 random letters)	13456	0.74 : 1

Compression

`echo <x> | gzip - | wc -c` # multiply by 8 for bits

Source string, x		$\ell(C(x))$	**ratio**
$aaa\ldots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb\ldots$	(10000 random letters)	13456	0.74 : 1
$abababab\ldots ab$	$(5000 \times ab)$	368	27.2 : 1

Compression

```
echo <x> | gzip - | wc -c        # multiply by 8 for bits
```

Source string, x		$\ell(C(x))$	**ratio**
$aaa \dots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb \dots$	(10000 random letters)	13456	0.74 : 1
$abababab \dots ab$	$(5000 \times ab)$	368	27.2 : 1
$aaa \dots abbb \dots b$	$(5000 \times a, 5000 \times b)$	376	26.6 : 1

Compression

```
echo <x> | gzip - | wc -c        # multiply by 8 for bits
```

Source string, x		$\ell(C(x))$	**ratio**
$aaa\ldots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb\ldots$	(10000 random letters)	13456	0.74 : 1
$abababab\ldots ab$	$(5000 \times ab)$	368	27.2 : 1
$aaa\ldots abbb\ldots b$	$(5000 \times a, 5000 \times b)$	376	26.6 : 1
$abbaababba\ldots$	$(1000 \times abbaababba)$	488	20.5 : 1

Compression

```
echo <x> | gzip - | wc -c        # multiply by 8 for bits
```

Source string, x		$\ell(C(x))$	**ratio**
$aaa\ldots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb\ldots$	(10000 random letters)	13456	0.74 : 1
$abababab\ldots ab$	$(5000 \times ab)$	368	27.2 : 1
$aaa\ldots abbb\ldots b$	$(5000 \times a, 5000 \times b)$	376	26.6 : 1
$abbaababba\ldots$	$(1000 \times abbaababba)$	488	20.5 : 1

Strings following a rule are compressible?

Compression

```
echo <x> | gzip - | wc -c        # multiply by 8 for bits
```

Source string, x		$\ell(C(x))$	**ratio**
$aaa \ldots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb \ldots$	(10000 random letters)	13456	0.74 : 1
$abababab \ldots ab$	$(5000 \times ab)$	368	27.2 : 1
$aaa \ldots abbb \ldots b$	$(5000 \times a, 5000 \times b)$	376	26.6 : 1
$abbaababba \ldots$	$(1000 \times abbaababba)$	488	20.5 : 1
$aaabbabbabb \ldots$	$(\pi,\ 0\text{–}4 \mapsto a,\ 5\text{–}9 \mapsto b)$	13416	0.74 : 1

π follows a rule but isn't compressed!

Compression

`echo <x> | gzip - | wc -c` # multiply by 8 for bits

Source string, x		$\ell(C(x))$	**ratio**
$aaa\ldots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb\ldots$	(10000 random letters)	13456	0.74 : 1
$abababab\ldots ab$	$(5000 \times ab)$	368	27.2 : 1
$aaa\ldots abbb\ldots b$	$(5000 \times a, 5000 \times b)$	376	26.6 : 1
$abbaababba\ldots$	$(1000 \times abbaababba)$	488	20.5 : 1
$aaabbabbabb\ldots$	$(\pi$, 0–4 $\mapsto a$, 5–9 $\mapsto b)$	13416	0.74 : 1

π follows a rule but isn't compressed!

Maybe it's just `gzip`? It would be possible to create to *special program* to compress π into a short file.

Compression

```
echo <x> | gzip - | wc -c        # multiply by 8 for bits
```

Source string, x		$\ell(C(x))$	**ratio**
$aaa\dots a$	$(10000 \times a)$	368	27.2 : 1.
$aabaabbbbabbbbb\dots$	(10000 random letters)	13456	0.74 : 1
$abababab\dots ab$	$(5000 \times ab)$	368	27.2 : 1
$aaa\dots abbb\dots b$	$(5000 \times a, 5000 \times b)$	376	26.6 : 1
$abbaababba\dots$	$(1000 \times abbaababba)$	488	20.5 : 1
$aaabbabbabb\dots$	$(\pi$, 0–4 $\mapsto a$, 5–9 $\mapsto b)$	13416	0.74 : 1

π follows a rule but isn't compressed!

Maybe it's just `gzip`? It would be possible to create to *special program* to compress π into a short file.

But what does it mean to compress an *individual* string???

Information

An individual string is “simple” (as opposed to “complex”) if it can be compressed into a small file by a *prespecified* program.

Information

An individual string is "simple" (as opposed to "complex") if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

Information

An individual string is "simple" (as opposed to "complex") if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

We can use several compressors if we prefix the code string by an index of the used program.

Information

An individual string is “simple” (as opposed to “complex”) if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

We can use several compressors if we prefix the code string by an index of the used program.

How about new compressors?

Information

An individual string is "simple" (as opposed to "complex") if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

We can use several compressors if we prefix the code string by an index of the used program.

How about new compressors? *Self-extracting files!*

Information

An individual string is "simple" (as opposed to "complex") if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

We can use several compressors if we prefix the code string by an index of the used program.

How about new compressors? *Self-extracting files!*

Can it be made automatic? Find the shortest program to print x.

Information

An individual string is “simple” (as opposed to “complex”) if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

We can use several compressors if we prefix the code string by an index of the used program.

How about new compressors? *Self-extracting files!*

Can it be made automatic? Find the shortest program to print x. **No.** *Kolmogorov complexity.*

Information

An individual string is "simple" (as opposed to "complex") if it can be compressed into a small file by a *prespecified* program.

But which program? `gzip` is not good for images (or for π).

We can use several compressors if we prefix the code string by an index of the used program.

How about new compressors? *Self-extracting files!*

Can it be made automatic? Find the shortest program to print x. **No.** *Kolmogorov complexity.*

Project

Next lecture

On Friday:

- Brief excursion to *noisy* channel coding (error correction).

Next lecture

On Friday:

- Brief excursion to *noisy* channel coding (error correction).

Next week:

- First exercises (will be posted on website tomorrow),

Next lecture

On Friday:

- Brief excursion to *noisy* channel coding (error correction).

Next week:

- First exercises (will be posted on website tomorrow),
- Due Tuesday (or Friday?)

Next lecture

On Friday:

- Brief excursion to *noisy* channel coding (error correction).

Next week:

- First exercises (will be posted on website tomorrow),
- Due Tuesday (or Friday?)
- Exercise sessions by Anupam.