Leggere, scrivere e ... comprimere

Alberto Policriti

Università di Udine

Quali tecniche per comprimere un testo?

Leggere, scrivere e comprimere: algoritmi e dizionari

Cosa resta da fare?

Serve proprio decomprimere?

Quali tecniche per comprimere un testo?

Codificare
codificare v. tr. [dal fr. codifier, der. di code «codice»] (io codìfico, tu codìfichi, ecc.).

1. *Ridurre in codice, dare cioè un ordine sistematico a un complesso di norme giuridiche relative a una determinata materia*

Codificare
codificare v. tr. [dal fr. codifier, der. di code «codice»] (io codìfico, tu codìfichi, ecc.).

2. *Esprimere informazioni e messaggi mediante le regole e i simboli di un sistema convenzionale (il codice) stabilito concordemente dall'emettitore e dal ricevitore dei messaggi allo scopo di trasmettere o elaborare automaticamente le informazioni o, talora, di mantenerle segrete: c. un ordine, un messaggio, un'istruzione. Riferito a calcolatori elettronici, convertire istruzioni del programma e dati nel codice di macchina.*

Codificare
codificare v. tr. [dal fr. codifier, der. di code «codice»] (io codìfico, tu codìfichi, ecc.).

3. *In biologia, inserire gli aminoacidi portati dall'RNA di trasferimento, per corrispondenza tra la tripletta di questo e le triplette dell'RNA messaggero, durante la sintesi proteica.*

Codifichiamo una stringa (di DNA)

A C T T T A C C T T G T

Table 1: codice `ASCII`

A	00
T	01
C	10
G	11

Codifichiamo una stringa (di DNA)

A C T T T A C C T T G T

Table 1: codice `ASCII`

A	00
T	01
C	10
G	11

Table 2: codice di Huffman

A	001
T	1
C	01
G	000

Codifichiamo una stringa (di DNA)

A C T T T A C C T T G T

Table 1: codice `ASCII`

A	00
T	01
C	10
G	11

Table 2: codice di Huffman

A	001
T	1
C	01
G	000

24 bits vs. 21 bits

Domande:

1. in quali casi funziona?
2. in quali casi *non* funziona?
3. funziona nel caso del genoma umano?
4. ci sono alternative? ci possiamo inventare qualcos'altro?

Entropia di una stringa di caratteri

- Quanto è probabile che il carattere sia (ad esempio) una A?

Entropia di una stringa di caratteri

- Quanto è probabile che il carattere sia (ad esempio) una A?
- Quanto è frequente il carattere A?

Entropia di una stringa di caratteri

- Quanto è probabile che il carattere sia (ad esempio) una A?
- Quanto è frequente il carattere A?
- Quanti bit mi servono per il carattere A?

Entropia di una stringa di caratteri

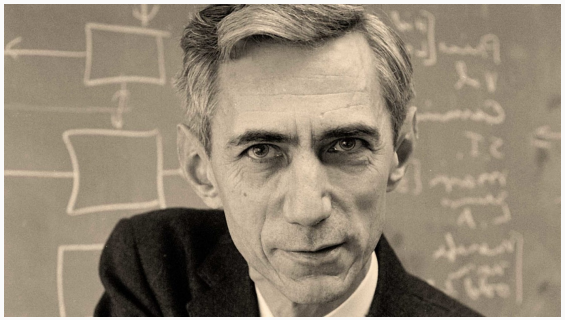
- Quanto è probabile che il carattere sia (ad esempio) una A?
- Quanto è frequente il carattere A?
- Quanti bit mi servono per il carattere A?

$$\uparrow \text{frequenza} \Leftrightarrow \downarrow \text{sorpresa} \Leftrightarrow \downarrow \text{informazione} \Leftrightarrow \uparrow \text{probabilità}$$

Entropia di una stringa di caratteri

- Quanto è probabile che il carattere sia (ad esempio) una A?
- Quanto è frequente il carattere A?
- Quanti bit mi servono per il carattere A?

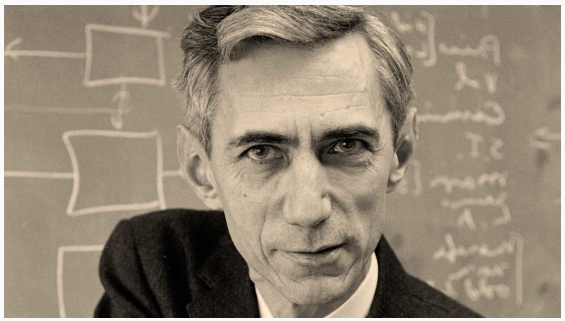
$\uparrow$ frequenza $\Leftrightarrow$ $\downarrow$ sorpresa $\Leftrightarrow$ $\downarrow$ informazione $\Leftrightarrow$ $\uparrow$ probabilità



Entropia di una stringa di caratteri

- Quanto è probabile che il carattere sia (ad esempio) una A?
- Quanto è frequente il carattere A?
- Quanti bit mi servono per il carattere A?

$\uparrow$ frequenza $\Leftrightarrow$ $\downarrow$ sorpresa $\Leftrightarrow$ $\downarrow$ informazione $\Leftrightarrow$ $\uparrow$ probabilità



Let's get real

Probabilità ed entropia di una *random variable distribution*

Let's get real

Probabilità ed entropia di una *random variable distribution*

Definition

Sia $p(x)$ probabilità che la r.v. X sia $x \in \{a, c, g, t\}$.

Let's get real

Probabilità ed entropia di una *random variable distribution*

Definition

Sia $p(x)$ probabilità che la r.v. X sia $x \in \{a, c, g, t\}$.

$$\mathcal{H}(X) = \sum_{x \in \{a,c,g,t\}} p(x) \log(1/p(x))$$

$$= \mathbb{E}[\log(1/p(x)] \text{ ... BITs (SHANNONs)}$$

Let's get real

Probabilità ed entropia di una *random variable distribution*

Definition

Sia $p(x)$ probabilità che la r.v. X sia $x \in \{a, c, g, t\}$.

$$\mathcal{H}(X) = \sum_{x \in \{a,c,g,t\}} p(x) \log(1/p(x))$$

$$= \mathbb{E}[\log(1/p(x)] \text{ ... BITs (SHANNONs)}$$

Non possiamo codificare una stringa S con meno di

$$H(S) = - \sum_{x \in \{a,c,g,t\}} P(x) \cdot \log_2(P(x))$$

bits/simbolo.

Torniamo alle nostre domande:

1. in quali casi funziona?
2. in quali casi *non* funziona?
3. funziona nel caso del genoma umano?
4. ci sono alternative? ci possiamo inventare qualcos'altro?

Comprimere due genomi

Per un solo genoma: compressori basati su entropia

codifica (ottimale)

Comprimere due genomi

Per un solo genoma: compressori basati su entropia

codifica (ottimale)

Per due (o più) genomi?

Comprimere due genomi

Per un solo genoma: compressori basati su entropia

codifica (ottimale)

Per due (o più) genomi?

Le frequenze non cambiano!

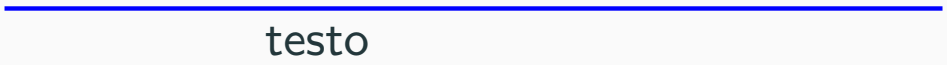
Idea!

Idea!

Alternativa

non rifare quello che hai già fatto

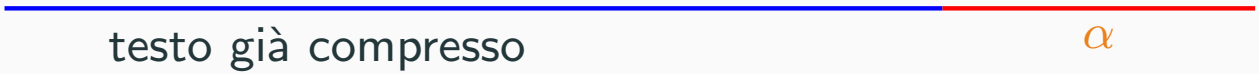
Idea!



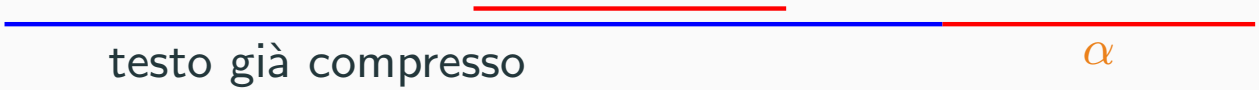
Idea!

testo già compresso

Idea!



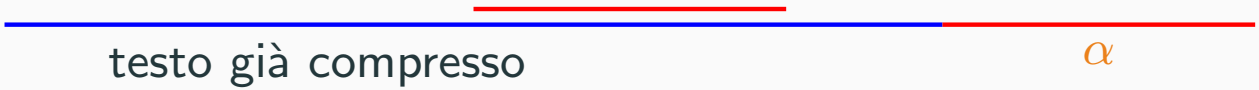
Idea!



Idea!

tutto ciò che mi serve è sapere: dove e quanto è lunga α

Idea!



tutto ciò che mi serve è sapere: dove e quanto è lunga α

due numeri interi

Idea!

tutto ciò che mi serve è sapere: dove e quanto è lunga α

due numeri interi

... codifica?

Leggere, scrivere e comprimere: algoritmi e dizionari

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

$$T = T^1 T^2 \ldots T^p$$

in p *frasi*.

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

$$T = T^1 T^2 \cdots T^p$$

in p *frasi*.

Indico con L_i la posizione di una *occorrenza* di T^i alla sua sinistra,

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

$$T = T^1 T^2 \cdots T^p$$

in p *frasi*.

Indico con L_i la posizione di una *occorrenza* di T^i alla sua sinistra,

comprimo T sostituendola con

$$(|T^1|, L_1)(|T^2|, L_2)\cdots(|T^p|, L_p)$$

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

$$T = T^1 T^2 \cdots T^p$$

in p *frasi*.

Indico con L_i la posizione di una *occorrenza* di T^i alla sua sinistra,

comprimo T sostituendola con

$$(|T^1|, L_1)(|T^2|, L_2)\cdots(|T^p|, L_p)$$

... leggere, scrivere e comprimere.

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

$$T = T^1 T^2 \cdots T^p$$

in p *frasi*.

Indico con L_i la posizione di una *occorrenza* di T^i alla sua sinistra,

comprimo T sostituendola con

$$(|T^1|, L_1)(|T^2|, L_2)\cdots(|T^p|, L_p)$$

Provate con la stringa usata per Huffman. Pensate al caso di due (o più) genomi da comprimere.

Lempel-Ziv parsing

$$T = t_1 t_2 \cdots t_n$$

vogliamo ottenere il *parsing* di T:

$$T = T^1 T^2 \cdots T^p$$

in p *frasi*.

Indico con L_i la posizione di una *occorrenza* di T^i alla sua sinistra,

comprimo T sostituendola con

$$(|T^1|, L_1)(|T^2|, L_2)\cdots(|T^p|, L_p)$$

https://www.ncbi.nlm.nih.gov

Cosa resta da fare?

Problemi da risolvere

- come trovo le occorrenze di α?

Problemi da risolvere

- come trovo le occorrenze di α?
- quanto tempo ci metto?

Problemi da risolvere

- come trovo le occorrenze di α?
- quanto tempo ci metto?
- quanto spazio occupo?

Problemi da risolvere

- come trovo le occorrenze di α?
- quanto tempo ci metto?
- quanto spazio occupo?
- da dove mi conviene partire?

Problemi da risolvere

- come trovo le occorrenze di α?
- quanto tempo ci metto?
- quanto spazio occupo?
- da dove mi conviene partire?
- ...

Lempel-Ziv-Welch

È possibile combinare LZ e codifica?

Codifichiamo *frasi*.

Lempel-Ziv-Welch

The initial patents for LZW were held by the Sperry Corporation, which later became Unisys Corporation. The first patent in the LZ family was filed in 1981. US patent 4,558,302 was issued to Terry Welch and Sperry Corporation on June 20,1983 (US 4558302 A).

The LZW method has been referenced in over one hundred additional patents, including ones as recent as 2013. Companies developing patented technologies that employ LZW include Unisys, Hewlett Packard, Apple, Google, Nielsen, Cisco, & Microsoft (US 4558302 A).

Lempel-Ziv-Welch

EDUCATION CENTER CALL US: (212) 634-2830 Search...

GOLDSTEIN PATENT LAW

CONTACT US ABOUT US PATENTS TRADEMARKS PODCASTS ABOUT THE PATENT PROCESS HOW TO GET STARTED

CAN YOU PATENT AN ALGORITHM?

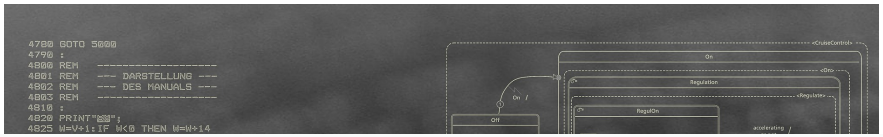
Have you developed a new software code and are wondering if you can patent an algorithm? This is a common question with developers, as they often want to know if their algorithm can be protected under intellectual property law.

CAN YOU PATENT AN ALGORITHM?

Unfortunately, Algorithms on their own cannot be patented because they are considered an “abstract idea.” However, you can patent the software *process* underlying your algorithm.

ALGORITHMS IN TODAY’S INFORMATION AGE

Artificial intelligence and machine learning developments have skyrocketed in recent years. As we continue deeper into the information age, software developments have become increasingly important as our culture becomes more and more dependent on computer devices and software.



SCHEDULE A FREE IDEA PROTECTION PLANNING SESSION

First Name *

Last Name

Email *

Phone *

By clicking Schedule Now, you agree to our Privacy Policy, including our Cookie Use.

SCHEDULE NOW

Serve proprio decomprimere?

Compact Data Structures

A Practical Approach

Gonzalo Navarro

Department of Computer Science,
University of Chile