Compressione senza perdita di informazione

Sommario

- Eliminazione della correlazione statistica
- Codifica entropica

Premessa

- Quanto si dirà in questi lucidi si applica a sorgenti di informazioni che producono **serie di dati discreti a precisione finita**:
  - dati digitali su file (es., eseguibili, database)
  - sorgenti multimediali dopo campionamento e quantizzazione
- Le tecniche che verranno presentate **non** provocano una perdita irreversibile di informazione.

Definizioni

- Una sorgente emette una **serie ordinata di dati**
- Tale serie di dati si può rappresentare come una **sequenza di simboli** dove ciascun simbolo appartiene ad certo **alfabeto** dipendente dal tipo di applicazione
  - Es: campioni sonori quantizzati su 16 bit con segno
    --> l'alfabeto è costituito dai numeri interi nell'intervallo [-32768, 32767]
  - Es: immagine in bianco e nero con pixel su 256 livelli di grigio
    --> l'alfabeto è costituito dai numeri interi nell'intervallo [0, 255]

Cosa vuol dire compressione ?

- Premesso che ogni simbolo deve essere codificato con dei bit per essere elaborato o trasmesso
- Definizione teorica di compressione

Cambio reversibile di alfabeto tale che la rappresentazione binaria della sequenza di simboli, tradotta nei simboli del nuovo alfabeto, richiede meno bit.

Eliminazione della correlazione statistica

Correlazione statistica

- Sia data una sequenza ordinata di simboli s_0, s_1, s_2, ..., s_n di un certo alfabeto emessa da una sorgente.
- C'è correlazione quando la probabilità che s_i assuma un certo valore dipende dal valore assunto dagli elementi precedenti della sequenza.
- Si dice che l'informazione contenuta nella sequenza è **ridondante** e si può cambiare alfabeto in modo da usare meno bit.

Esempio di correlazione nei testi

• 13% di lettere eliminate
• quante parole si riesce a ricostruire ?

L sp zie s no al eate pre iose nella p epa azi ne di suc ulenti pia ti. Es e s no ben con sci te fin dall'an ichi à; at or o ad es e vi er no ve e e pro rie vie com erci li e per esse si c mba terono du e bat agl e.

Esempio di correlazione nei testi

- 6% di lettere eliminate
- quante parole si riesce a ricostruire ?

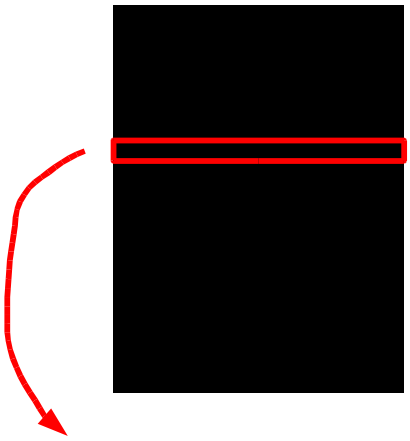
L sp zie sono al eate pre iose nella prepa azione di suc ulenti pia ti. Es e sono ben conosciute fin dall'antichità; at orno ad es e vi erano vere e proprie vie com erciali e per esse si comba terono dure bat aglie.

Esempio di correlazione nei testi

Le spezie sono alleate preziose nella preparazione di succulenti piatti. Esse sono ben conosciute fin dall'antichità; attorno ad esse vi erano vere e proprie vie commerciali e per esse si combatterono dure battaglie.

Correlazione nelle immagini

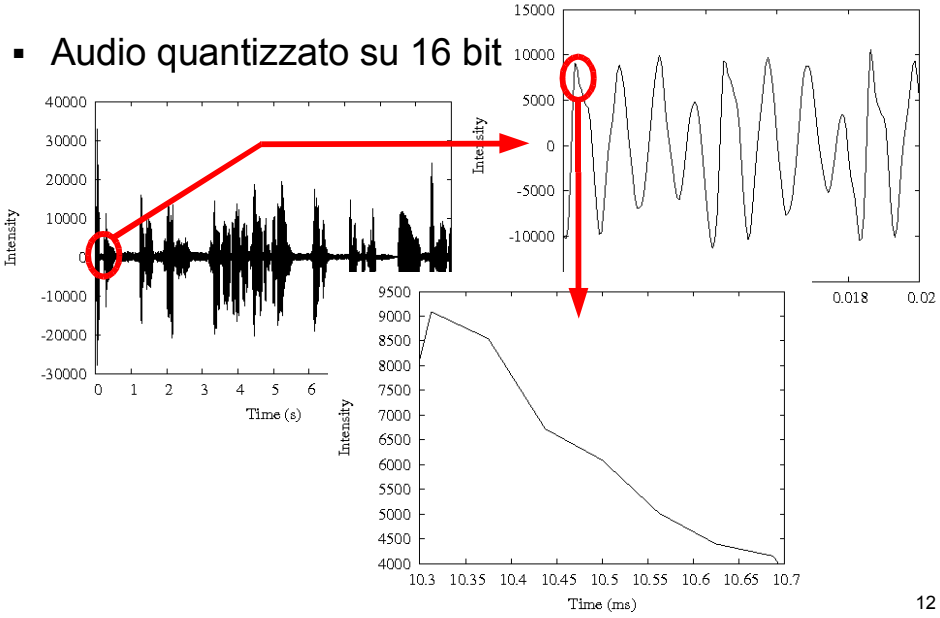
- Immagine b/n su due livelli di grigio (fax)



Sequenza: 1,1,1,1,0,0,0,0,0,0,1,1,1,1,1

Correlazione nella voce

- Audio quantizzato su 16 bit



Eliminazione della correlazione

- **Si cambia alfabeto di rappresentazione**
- Codifica run-length
  - Es del fax

    1,1,1,1,0,0,0,0,0,0,1,1,1,1,1 --> (1,4),(0,6),(1,5)
- Codifica differenziale
  - Es. dell'audio (vedi la figura più dettagliata)

    $s_0\ s_1\ s_2 \ldots s_n$ --> $(s_1 - s_0), (s_2 - s_1), \ldots, (s_n - s_{n-1})$
- Trasformata
  - Es. dell'audio (vedi la figura intermedia)

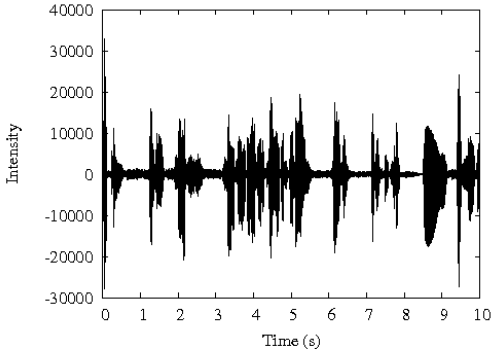
    Sviluppo in serie di Fourier: ampiezza delle componenti sinusoidali

    $s_0\ s_1\ s_2 \ldots s_n$ --> $a_0, a_1, \ldots, a_n$

Codifica entropica

Frequenza dei simboli: esempio

- Audio su 16 bit: alfabeto = interi su [-32768, 32767]



- I simboli hanno tutti la stessa occorrenza ?
- Ricordate il caso dell'alfabeto Morse ?

Quanti bit/simbolo ?

- E' meglio rappresentare con meno bit i simboli più frequenti e con più bit quelli meno frequenti.
- Numero minimo teorico di bit/simbolo per l'alfabeto *s*:

$$H(s) = \sum_k p_k \log_2 \frac{1}{p_k}$$

  - p_k è la frequenza di apparizione del simbolo *k*-esimo
  - **entropia della sorgente di simboli** (*Shannon*)
  - tale formula fornisce un valore medio minimo ma non dice come assegnare i bit ai simboli per ottenerlo !

Esempio di calcolo dell'entropia

- Una sorgente emette simboli appartenenti ad un alfabeto di 4 simboli *a, b, c ,d*
- Frequenze di apparizione dei 4 simboli sono:
  - 1/2, 1/4, 1/8, 1/8
  - somma delle frequenze uguale a 1
- Entropia della sorgente:

$$H(s)=\frac{1}{2}\log_2(2)+\frac{1}{4}\log_2(4)+\frac{1}{8}\log_2(8)+\frac{1}{8}\log_2(8)$$
$$.=\frac{1}{2}+\frac{1}{2}+\frac{3}{8}+\frac{3}{8}$$
$$.=1.75\,bit/simbolo$$

Codifica entropica: problema 1

- Simboli rappresentati su un numero variabile di bit

- come fa il decoder a sapere quando inizia/finisce un simbolo ?

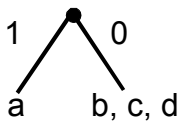
Algoritmo di Huffman

- Metodo di assegnamento bit ai simboli che garantisce la corretta decodifica
- Si costruisce un albero binario dove le foglie rappresentano i simboli
- Ad ogni biforcazione la somma delle frequenze dei simboli del sotto-albero di destra deve essere il più possibile vicina alla somma delle frequenze dei simboli del sotto-albero di sinistra.
- Ciascun arco di una biforcazione è etichettato con 0 e 1 rispettivamente

Algoritmo di Huffman: esempio

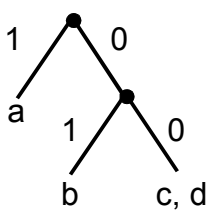
- Alfabeto di 4 simboli *a, b, c ,d* con frequenze rispettivamente di 1/2, 1/4, 1/8, 1/8
- Primo passo: *a,* (1/2) *b, c ,d* (1/2)

a --> 1



- Secondo passo: *b,* (1/4) *c,d* (1/4)

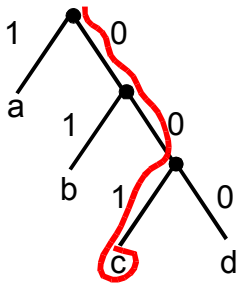
a --> 1
b --> 01



Algoritmo di Huffman: esempio (2)

- Terzo passo: *c* *d*
  1/8 1/8

a --> 1
b --> 01
c --> 001
d --> 000



- In ricezione per decodificare i simboli si visita l'albero binario in base ai bit in arrivo

  Es: 001 ---> c

Huffman: considerazioni

- L'algoritmo di Huffman è ottimo (rispetto all'entropia) se le frequenze sono potenze di 2.
- Per distribuzioni di frequenze più complicate esistono algoritmi più sofisticati
  - Codifica aritmetica
  - Codici di Golomb

Codifica entropica: altri problemi

- Stima delle frequenze p_k
  - migliore è la stima e maggiore è la compressione
  - ho bisogno di fare ipotesi vincolanti sulla sorgente oppure considerare un elevato numero di esempi
- Errori sul bit durante la trasmissione rendono la decodifica impossibile
  - punti di re-sincronizzazione periodici nel flusso di bit
  - diminuzione dell'efficienza di compressione

Considerazioni sulla complessità

In generale se si vuole comprimere di più una sequenza di simboli si devono usare **algoritmi più complessi** che impiegano più tempo oppure necessitano di una **CPU più potente** e comunque sempre portano ad un **maggiore consumo di energia**.