

Elementi di teoria dell'informazione e della trasmissione

Introduzione

Efficienza nell'uso dei supporti

Quantità di informazione ed entropia

La caratterizzazione statistica dei canali

Luca Mari, Strumentazione Elettronica di Misura

La centralità dell'informazione e del suo trattamento

E' stimato che intorno all'80% dell'attuale prodotto interno lordo degli Stati Uniti sia correlato al *trattamento di informazione*

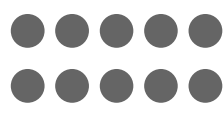
E la transizione verso una società post-industriale è segnata da un cambiamento della risorsa fondamentale:

non è più la terra, o l'energia, o il capitale, ma l'informazione

Evidentemente non si fa riferimento solo al settore di mercato dei cosiddetti *mass media*:

in quale senso si intende, in questo caso, il termine "informazione" ?

Informazione e supporto fisico

10 10 

... possono portare la stessa informazione, ma ...

10

... può portare informazioni differenti

L'informazione è "scritta" su un supporto fisico:

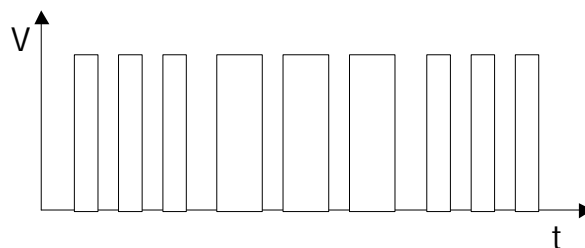
→ il supporto è necessario per il trattamento dell'informazione

→ l'informazione non coincide con il suo supporto

(informazione vs. supporto ↔ software vs. hardware)

Informazione sintattica e semantica

Un esempio:



1

2

[punto punto punto] [linea linea linea] [punto punto punto]

3

S O S

4

"correte a salvarci ..."

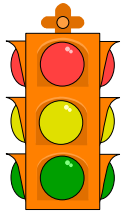
La relazione tra un livello e il successivo è definita mediante un *codice* ...

... che fornisce un' *interpretazione* per le entità del livello precedente

Il nostro interesse è per il livello sintattico ("tecnico") dell'informazione

Verso il concetto di informazione

Cosa fa sì che un supporto possa portare informazione?



- se si sa che è rotto, non occorre guardarlo
- se si conosce in anticipo lo stato, non occorre guardarlo

Dunque:

- il supporto deve poter assumere almeno due stati (condizione oggettiva: informazione come *varietà*)
- lo stato attuale non deve essere noto in anticipo (condizione soggettiva: informazione come *incremento di certezza*)

Nota: la seconda condizione sussume la prima

Come definire la quantità di informazione portata?

Due esempi

Un semaforo è il supporto fisico per le entità di informazione: stop, attenzione, avanti

Si può / come minimizzare l'uso di lampadine nei semafori? Per esempio:

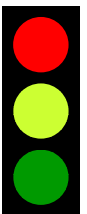
stop



avanti

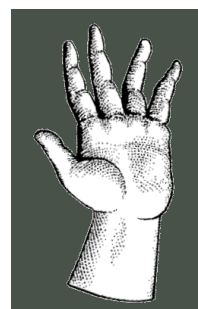


attenzione



Per identificare 3 entità di informazione è (più che) sufficiente un supporto costituito da una successione ordinata di 2 sottosupporti bistabili

Un esempio inverso: con questo supporto, pensato come costituito da cinque sottosupporti bistabili, quante entità di informazione diverse possono essere identificate?
... 2^5 , cioè 32



La prima formalizzazione

L'insieme degli stati distinguibili di un supporto fisico costituito da k sottosupporti bistabili è costituito da 2^k elementi

Cioè: per identificare un'entità di informazione scelta da un insieme di 2^k entità è sufficiente usare un sistema fisico costituito da k sottosistemi bistabili

Inversamente: se si deve identificare un'entità di informazione scelta da un insieme di k entità, il supporto fisico dovrà essere costituito da non meno di $\log_2(k)$ sottosupporti bistabili

Scegliamo come strumento per definire l'unità di misura della quantità di informazione i sistemi bistabili:

un sistema a due stati è in grado di portare 1 unità di informazione (perché $1 = \log_2(2)$)

In generale, un sistema a k stati è in grado di portare $\log_2(k)$ bit di informazione

Efficienza nell'uso dei supporti

Semaforo: pur essendo costituito da 3 sottosupporti bistabili, ognuno dei suoi stati porta meno di 3 bit di informazione

Se chiamiamo "bit di memoria", bit_m , un sistema bistabile e "bit di informazione", bit_i , l'unità di misura dell'informazione, ne segue che un bit_m può portare anche meno di un bit_i

Per esempio, queste due figure sono entrambe memorizzate su file (=sistemi fisici) di $100 \times 100 \times 8$ bit_m , ma la quantità di informazione che portano è certamente diversa:



... e quindi dovrebbe essere possibile ridurre la quantità di supporto senza perdere informazione: *la compressione*

Un supporto usato in modo non efficiente, e quindi comprimibile, si dice *ridondante*

Limiti alla compressione

Un supporto a k stati porta al più $\log_2(k)$ bit_i di informazione
 ... mentre la quantità minima di informazione che un supporto può portare è 0 bit_i,
 ovviamente!

Se disponiamo di informazione per $x < \log_2(k)$ bit_i su un supporto a k stati (che quindi è
 ridondante), possiamo (se vogliamo eliminare la ridondanza) ridurre il numero degli stati
 (cioè comprimere il supporto) fino a un valore k' tale che $x = \log_2(k')$ bit_i

Dunque il limite alla possibilità di compressione di un supporto è dato (naturalmente!) dalla
 quantità di informazione che esso deve portare

Ma come stabilire quanta informazione è portata effettivamente da un certo supporto?

E quindi:

qual è il limite alla possibilità di comprimere un supporto
 senza perdere l'informazione che esso porta?

Un problema

Con una classe di 100 studenti, si deve comunicare un voto: A, B, C o D ...
 ... e per la comunicazione si possono usare solo dispositivi bistabili, cioè bit_m. Per esempio:




In questo modo, per comunicare i 100 voti si usano 200 bit_m: possiamo comprimere?

Supponiamo che la distribuzione dei voti sia non uniforme, ma:

A: 1/2 B: 1/4 C: 1/8 D: 1/8

e supponiamo di ri-codificare i voti così:



Per cui, per esempio: 
 B A D C A

(la regola è corretta, nel senso che consente di ricostruire univocamente i voti)

L'ipotesi fondamentale

Con questa distribuzione e questa codifica quanti bit_m occorrono per comunicare i 100 voti?

Ogni voto richiede in media:

$$1 \cdot 1/2 + 2 \cdot 1/4 + 3 \cdot 1/8 + 3 \cdot 1/8 = 1,75 \text{ bit}_m$$

e dato che i voti sono 100 ... siamo passati da 200 a 175 bit_m ...

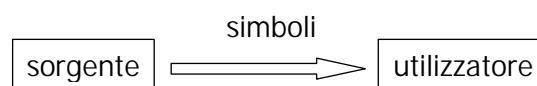
La ricodifica corrisponde a una compressione del supporto!

(avremmo potuto fare meglio, cioè ottenere una compressione ancora maggiore?)

Il risultato (qualitativamente) fondamentale è:

quanto meno è frequente / probabile un'entità di informazione,
tanto maggiore è la quantità di informazione che l'osservazione di tale entità porta

Un modello discreto



... dove la sorgente potrebbe essere, per esempio:

- un semaforo
- un calcolatore che crea un file
- un telefono che trasmette

Sia $S = \{s_i\}$ l'insieme dei possibili stati / simboli emessi dalla sorgente

Dunque:

all'aumentare di #S, numero di simboli in S, aumenta la "quantità di non-cerchezza" di ogni simbolo e corrispondentemente aumenta $Q_{\text{Inf}}(s)$, la quantità di informazione portata da ogni simbolo s

Più in generale: se si riduce $P(s)$, la probabilità di s, aumenta $Q_{\text{Inf}}(s)$

Quantità di informazione

Dato lo schema probabilistico:

$$S = \begin{bmatrix} s_1 & \dots & s_n \\ P(s_1) & \dots & P(s_n) \end{bmatrix}$$

(ovviamente con $\sum_i P(s_i) = 1$)

cerchiamo una funzione QInf: insieme di successioni di simboli $\rightarrow [0, \infty)$

tale che: $QInf(s) = f(P(s))$, con f decrescente al crescere di $P(s)$

e inoltre: $QInf(s_1 \circ s_2) = QInf(s_1) + QInf(s_2)$ per simboli statisticamente indipendenti

La definizione di **quantità di informazione**:

$$QInf(s) = \log_2\left(\frac{1}{P(s)}\right) = -\log_2(P(s)) \text{ bit}_i$$

(la base 2 ha il solo scopo di qualificare il bit come unità di misura)

(ancora sulle ragioni di \log_2 : in un albero binario di scelte, con n domande si seleziona 1 alternativa da 2^n possibili; quindi 1 alternativa da n porta informazione sulle risposte a $\log_2(n)$ domande)

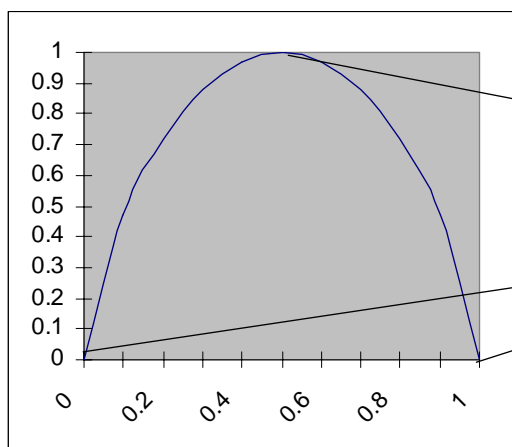
Entropia

Mediando QInf sull'insieme dei simboli, si ottiene:

$$H(S) = \langle QInf(s_i) \rangle = \sum_i P(s_i) QInf(s_i) = -\sum_i P(s_i) \log_2(P(s_i)) \text{ bit/simbolo}$$

detta funzione **entropia** (dell'insieme dei simboli / di sorgente)

Per esempio, dato un insieme di 2 simboli (dunque $P(s_2) = 1 - P(s_1)$), l'andamento dell'entropia in funzione di $P(s_1)$:



massima incertezza / informazione media

incertezza / informazione nulla

Bit come unita' di misura: bit_i

Bit come unita' di memoria: bit_m

Che relazioni ci sono tra bit_i e bit_m ?

(nota che i bit_m sono simboli solo se il numero di simboli possibili è pari a 2)

Per esempio: $S = \{s_1, s_2, s_3, s_4\}$ codificati in forma di bit_m come $\{00, 01, 10, 11\}$

Ogni simbolo s_i , che richiede dunque 2 bit_m di memoria, porta 2 bit_i di informazione?

Risposta: se i simboli sono equiprobabili, sì! (la verifica è banale ...)

Ma supponiamo, invece, che: $P(s_1)=1/2, P(s_2)=1/4, P(s_3)=1/8, P(s_4)=1/8$

Con la codifica precedente, per esempio, s_1 richiede ancora 2 bit_m ma porta 1 bit_i !

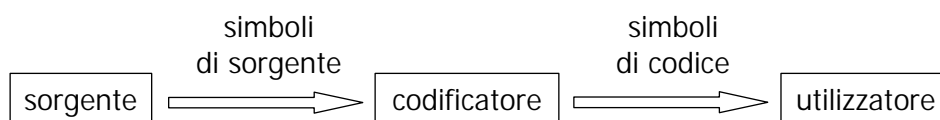
... così che un file di 100 simboli richiederebbe 200 bit_m ma porterebbe, in media:

$$\log_2(2)/2 + \log_2(4)/4 + \log_2(8)/8 + \log_2(8)/8 = 1,75 \text{ bit}_i/\text{simbolo}$$

che è dunque l'entropia per lo schema probabilistico indicato

La codifica

Tra sorgente e utilizzatore si introduce un codificatore, con il compito di convertire i simboli "di sorgente" (scelti dallo schema S) in (successioni di) simboli "di codice", per esempio bit_m



Nell'esempio precedente, dunque:

$\{s_1, s_2, s_3, s_4\}$ è l'*alfabeto di sorgente*

$\{0, 1\}$ l'*alfabeto di codice*

e il codificatore cod realizza la funzione:

$$\text{cod}(s_1)=00$$

$$\text{cod}(s_2)=01$$

$$\text{cod}(s_3)=10$$

$$\text{cod}(s_4)=11$$

... e se i simboli *non* sono statisticamente indipendenti ?

... cioè se $P(s_2 | s_1) \neq P(s_2)$

Evidentemente: $Q\text{Inf}(s_1 \circ s_2) < Q\text{Inf}(s_1) + Q\text{Inf}(s_2)$

fino all'estremo che: $Q\text{Inf}(s_1 \circ s_2) = Q\text{Inf}(s_1)$

quando la presenza di s_1 è sufficiente per assicurare la successiva presenza di s_2

(l'esempio delle lingue naturali: in italiano, dopo una "q", la probabilità di avere una lettera diversa dalla "u" è molto bassa!)

Anche in questi casi si è in presenza di ridondanza, che può essere eliminata introducendo un opportuno codificatore:



... che dunque attua una compressione del supporto

Risultato fondamentale

Il limite inferiore alla comprimibilità "lossless" è dato dall'entropia della sorgente

... e la compressione si attua mediante la tecnica della *codifica a lunghezza variabile*, cioè mediante un codificatore che associa ai simboli di sorgente successioni di simboli di codice di lunghezza non costante

Questo risultato è noto, in forma generale, come *primo teorema di Shannon*:

se num_i è il numero di simboli di codice di cui è costituita la successione $\text{cod}(s_i)$

e $\text{num} = \sum_i P(s_i) \text{num}_i$ è il valor medio dei num_i

e $\#C$ è il numero di simboli di codice diversi utilizzabili (2 nel caso dei bit_m dunque)

allora:

$$\text{num} \times \log_2(\#C) \geq H(S)$$

Codifica a lunghezza variabile

Per esempio:

| | | | |
|-------|-----------|-----------|-----|
| s_1 | $P = 1/2$ | codifica: | 0 |
| s_2 | $1/4$ | | 10 |
| s_3 | $1/8$ | | 110 |
| s_4 | $1/8$ | | 111 |

(è evidente la regola di lettura)

Un file di 100 caratteri richiederebbe, in media, 175 bit_m, tanti quanti sono i bit_i !

Un altro esempio: #S = 8; $P(s_1) = 25/32$, $P(s_i) = 1/32$, $i = 2, \dots, 8$

Con codifica a lunghezza fissa, ogni simbolo richiede 3 bit_m e dunque un file di 100 simboli richiede 300 bit_m, mentre $H(S) = 1,37$ bit_i/simbolo

Una codifica alternativa:

| | | |
|-------|-----------|------|
| s_1 | codifica: | 0 |
| s_2 | | 1000 |
| s_3 | | 1001 |
| ... | | ... |
| s_8 | | 1110 |

(è evidente la regola di lettura)

... così che un file di 100 simboli richiederebbe in media 166 bit_m

Problema

Un altro esempio:

$$S_P = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ 0.3 & 0.3 & 0.1 & 0.1 & 0.05 & 0.05 & 0.05 & 0.05 \end{bmatrix}$$

Calcolare l'entropia e identificare una codifica efficiente (cioè "vicina all'entropia")

Entropia = 2,57 bit_i/simbolo

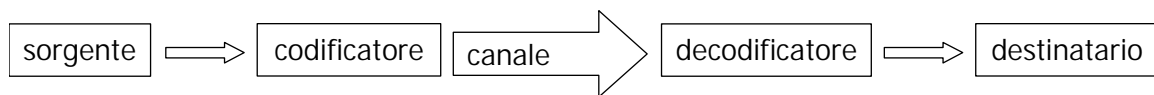
Una codifica possibile:

| | | | |
|--------------|--------------|--------------|--------------|
| s_1 : 00 | s_2 : 01 | s_3 : 100 | s_4 : 101 |
| s_5 : 1100 | s_6 : 1101 | s_7 : 1110 | s_8 : 1111 |

(è evidente la regola di lettura)

corrispondente a 2,6 bit_m/simbolo

Un modello di sistema di comunicazione



(comunicazione unidirezionale e point-to-point)

- La sorgente emette "simboli di sorgente",
- che un codificatore trasforma in "simboli di canale" e invia a un canale,
- in cui i simboli possono venire modificati a causa della presenza di rumore;
- in uscita dal canale un decodificatore ritrasforma i "simboli di canale"
- e li invia al destinatario

I problemi generali posti in questo contesto sono:

→ date le caratteristiche statistiche della sorgente, come realizzare la codifica in modo da ottimizzare (in termini di numero di simboli) il segnale inviato al canale?

→ date le caratteristiche statistiche del rumore sul canale, come realizzare la codifica in modo da ottenere un'accettabile probabilità che i simboli che giungono al destinatario siano identici a quelli emessi dalla sorgente?

La descrizione statistica del canale

Per esempio:

| Codificatore | Decodificatore | | $P(d_j s_i)$ | d_1 | d_2 | d_3 | d_4 |
|--------------|---|--|--------------|-------|-------|-------|-------|
| s_1 | $\begin{array}{l} \xrightarrow{3/4} d_1 \\ \xrightarrow{1/4} d_2 \end{array}$ | | s_1 | $3/4$ | $1/4$ | 0 | 0 |
| s_2 | $\xrightarrow{1} d_2$ | | s_2 | 0 | 1 | 0 | 0 |
| s_3 | $\xrightarrow{1} d_3$ | | s_3 | 0 | 0 | 1 | 0 |
| s_4 | $\xrightarrow{1} d_4$ | | s_4 | 0 | 0 | 0 | 1 |

... ricordando che $P(s_i|d_j) = P(d_j|s_i) * P(s_j) / P(d_i)$ (Bayes), per cui se, per ogni i , $P(s_i) = 1/4$:

| $P(s_i d_j)$ | d_1 | d_2 | d_3 | d_4 |
|--------------|-------|-------|-------|-------|
| s_1 | 1 | 1/5 | 0 | 0 |
| s_2 | 0 | 4/5 | 0 | 0 |
| s_3 | 0 | 0 | 1 | 0 |
| s_4 | 0 | 0 | 0 | 1 |

Equivocazione

La funzione:

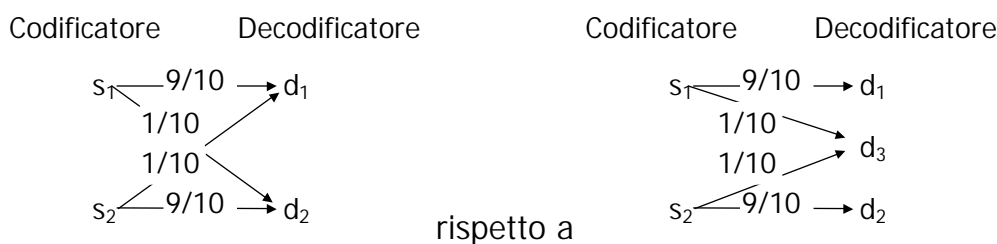
$$H(S|D) = \langle \langle Q \ln f(s_i | d_j) \rangle_j \rangle_i = - \sum_j P(d_j) \sum_i P(s_i | d_j) \log_2(P(s_i | d_j))$$

è detta *equivocazione* e rappresenta la quantità media di informazione che si otterrebbe osservando i simboli in ingresso al canale dopo che si sono osservati i simboli in uscita

→ se $H(S|D)=0$ una volta osservato in uscita un simbolo d_j non c'è più alcun dubbio su quale fosse il simbolo in ingresso: *il canale non introduce rumore*

→ se $H(S|D)=H(S)$ i simboli in uscita dal canale sono statisticamente indipendenti dai simboli in ingresso e il canale si comporta come se fosse una nuova sorgente: l'osservazione di un certo simbolo in uscita non porta alcuna informazione su quale fosse il simbolo in ingresso, *il canale è massimamente rumoroso e quindi inutile*

Un esempio



e ipotizzando $P(s_1) = P(s_2) = 0,5$ in entrambi i casi

Quale dei due canali si comporta meglio (= ha un'equivocazione minore)?

$$P(d_1)=P(d_2)=0,5$$

$$P(d_1)=P(d_2)=9/20, P(d_3)=2/20$$

$$P(s_1|d_1)=0,9; P(s_2|d_1)=0,1$$

$$P(s_1|d_1)=1; P(s_2|d_1)=0; P(s_3|d_1)=0$$

$$P(s_1|d_3)=0,5; P(s_2|d_3)=0,5$$

Se facciamo i conti su $H(S|D)$:

$$0,47 \text{ bit}_i/\text{simbolo}$$

$$0,1 \text{ bit}_i/\text{simbolo}$$

Quando non si è sicuri e si stima in 1/10 il margine di errore, è meglio rispondere "non lo so" piuttosto che tirare a caso!

Capacità di canale

La differenza tra informazione media all'ingresso del canale (*entropia di sorgente*) e informazione media persa lungo il canale (*equivocazione di canale*) è un indice della *capacità del canale di trasferire informazione*

(nota che $H(S) - H(S|D) = H(D) - H(D|S)$ (cf. Bayes ...), e quindi si tratta di una grandezza che formalizza il comportamento del canale come identico bidirezionalmente)

Massimizzando tra le possibili sorgenti:

$$\text{Capacità di canale } K(C) = \max_S [H(S) - H(S|D)] \text{ bit}_i/\text{s}$$

... e dal confronto tra l'entropia di sorgente e la capacità di canale si verifica se sorgente e canale sono correttamente accoppiati

(valori confrontabili in un contesto statico [bit_i/simbolo] oppure, introducendo le velocità di emissione e di trasmissione di simboli, in un contesto dinamico [bit_i/s])

Il problema della comunicazione

Dato un canale, esiste un (qual è il) limite superiore alla quantità di informazione trasmissibile in un intervallo di tempo?

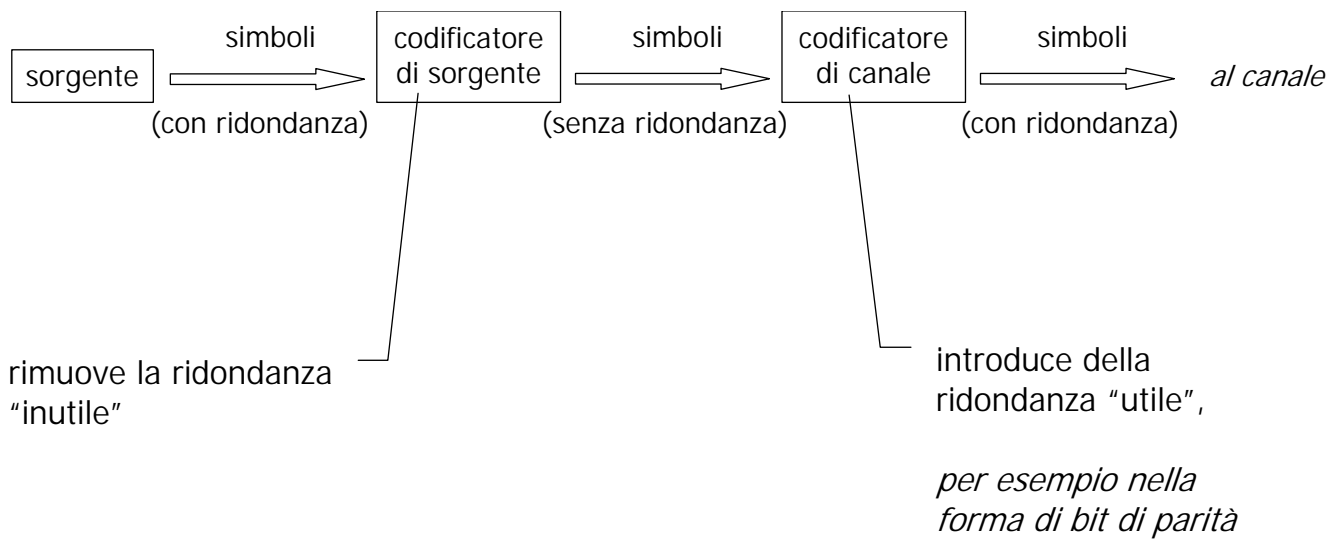
Data una velocità di emissione di informazione da parte della sorgente, è possibile, per un dato canale, ridurre la probabilità di errore a un qualsiasi valore desiderato?

Il secondo teorema di Shannon:

la probabilità di errore può essere ridotta arbitrariamente con un'appropriata codifica dell'informazione in partenza, purché la capacità di canale sia superiore all'entropia di sorgente

(nota: *ridotta arbitrariamente*, non *annullata* !)

La codifica di canale



In base al tipo di ridondanza introdotta in codifica di canale, può essere possibile riconoscere la presenza di errori oppure anche correggere gli errori riconosciuti