Codifiche in presenza di rumore /06-05

Un segnale attraverso un canale è soggetto a rumore
→ nei dati in memoria ha spesso poca probabilità
→ nella rete può alterare la comunicazione

Il problema non è comprimere il segnale ma trovare codifiche che resistono al rumore aumentando la ridondanza

Distanza di Hamming

Date due sequenze $u = u_1, u_2 \dots u_n$

$V = v_1, v_2 \dots v_n$

$$d(u,v) = \sum_{i=1}^{N} u_i \overset{\text{XOR}}{\oplus} v_i$$

conta i bit diversi

minimo zero, max N

(stringhe uguali) (stringhe complementari)

Determina proprietà:

1. $d(u,u) \geq 0$ $= 0$ su $u=v$ non distiamo dagli stessi
2. $d(u,v) = d(v,u)$ simmetria
3. $d(u,v) + d(v,z) \geq d(u,z)$

È una distanza? Sì perché positiva, se uguale $=0$, vale la simmetria.

Per la ③: se $u \neq z$ ma allora z andrà "d'accordo" con uno dei due v o u a sinistra, quindi $\geq$

Con n bit posso costruire 2^n file (messaggi)

$k=3$

000
001
010
011
100
101
110
111

trasmetto → 101

ho ricevuto esattamente 101?
Ho rumore, potevo forse intendere
un bit di errore 001, 100 ...
due bit di errore 011, 000 ...
...

conde molto rumoroso ←

Aumentando la distanza posso diversificare meglio i messaggi
→ ciò necessita di più bit

k bit → 2^k messaggi
$k+M$ bit → 2^k messaggi ma in uno spazio più grande

Spedisco bit. Come sono sicuro del messaggio ricevuto?

11 → 11 99/100
10 3/100
01 3/100
00 1/100

più sicurezza
ma ancora problemi

con un bit è difficile:
1 → 1 o 0 cosa intendevo prima della trasmissione?

111 → 110
101
011
:
} non sono neanche problematici, perché ho capito di avere 1 (due bit su tre hanno bit 1)

Al posto di 1 bit ne mando 3
ma ho più sicurezza

Codifica convoluzionale

bit di parità: bit in più che permette di controllare se il n° di bit 1 è pari o dispari

l'idea non è spedire il messaggio ma spedire i bit di parità associati

equazione di parità

$$y_1 =_2 x_1 + x_2 + x_3$$
$$y_2 =_2 x_2 + x_3$$
$$y_3 =_2 x_1 + x_3$$

↑ modulo 2

bit da voglio codificare

bit precedenti ↘

1 1 0 1 0 0

$y_1\ y_2\ y_3$

→ 0 1 0 | 1 1 0

invece di spedire un bit ne mando 3 e non spedisco il bit da voglio codificare (_ ∅ _)

Sequenza: $x[1]\ x[2] \dots x[N]$ scelgo $k=3$ considero triplette di bit

con $x[-1] = x[0] = \emptyset$

e ho P equazioni di parità scelgo $P=3$

$$y_1 =_2 x[n-2] + x[n-1] + x[n]$$
$$y_2 =_2 x[n-1] + x[n]$$
$$y_3 =_2 x[n-2] + x[n]$$

↑ mess da codificare ↑ mess da codificare

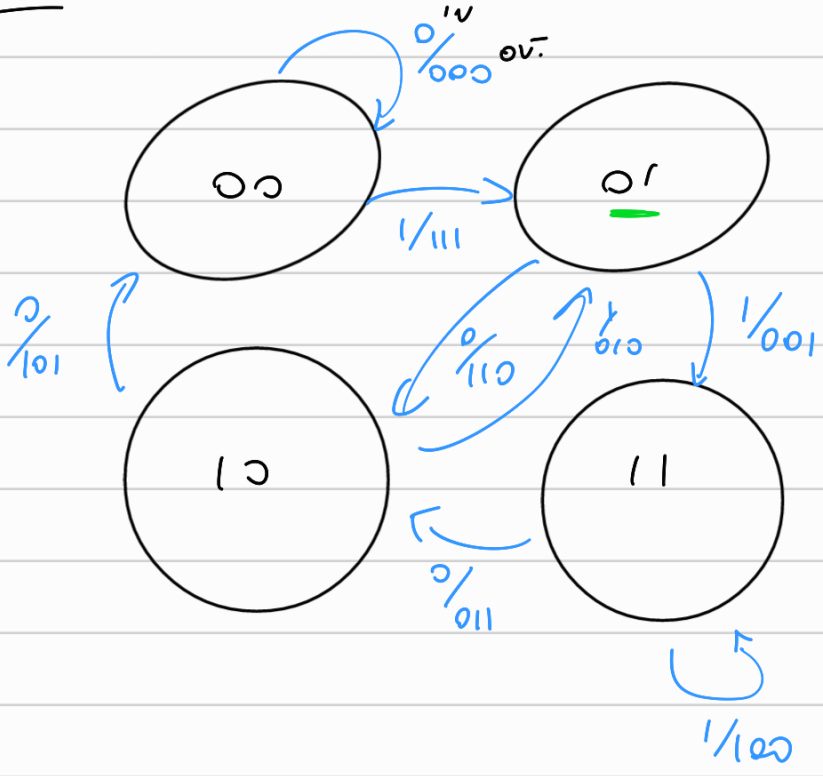
Rate $\frac{1}{P}$: ogni volta che spedisco un bit ne invio P

Codifica

FSM = macchine a stati finiti

a coppie di bit (stati)

00
01
10
11
} stati



sulla base degli input (bit) produce un output (tripletta) per poi transitare in un altro stato (altre coppie di bit)

es

$y_1 =_2 x[n-2] + x[n-1] + x[n]$

$y_2 =_2 x[n-1] + x[n]$

$y_3 =_2 x[n-2] + x[n]$

coppie ottenute

INPUT 0 1

il nuovo stato si forma dall'ultimo bit dello stato concatenato con l'input

È una macchina che si basa su transizioni da uno stato all'altro, prendendo input e restituendo output.

C(1101)

00 1101 input

stato di partenza

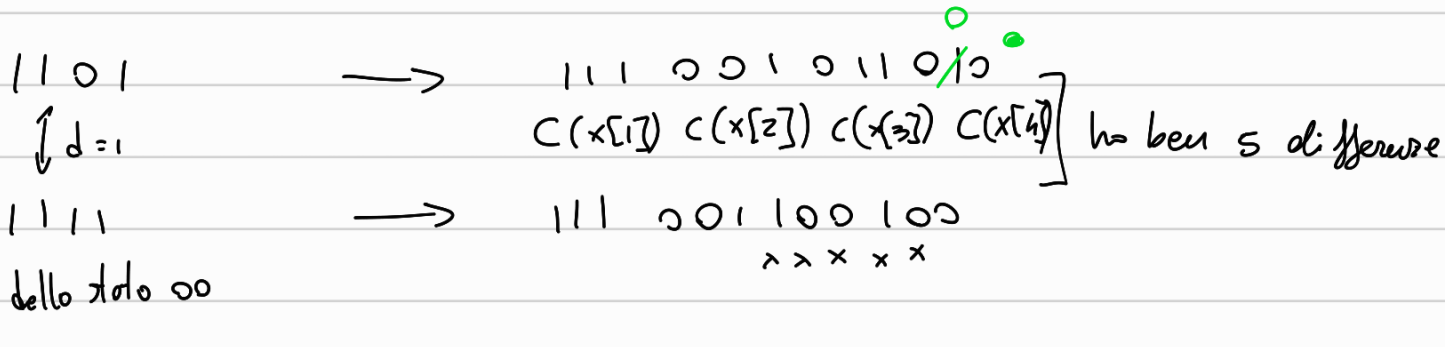
posso mandare 16 stringhe

→ 111 001 011 010

ne posso ricevere 4096

(senza errori ne prendo 16)

L'idea è prendere stringhe distanti 1 in Hamming



parte dello stato 00

se fosse arrivato qualche bit sbagliato di •
ha comunque più' possibilità' che quella stringa sia generata da 1101

Decodifica

Ricevo $110\ 001\ 111\ 010 = c^{ric}$ dove $C(x) = 111\ 001\ 011\ 010$ inviato (non nodo a pezzi)

$$L = d(c(x), c^{ric}) = \sum_{n=1}^{4 \to \text{triplette}} d(C(x_n), c^{ric}(n)) = 2$$

d tra ogni triplette

cerco la codifica con distanza minima

se le stringhe sono tante e non 4 ho parecchie codifiche e una ricerca lunga

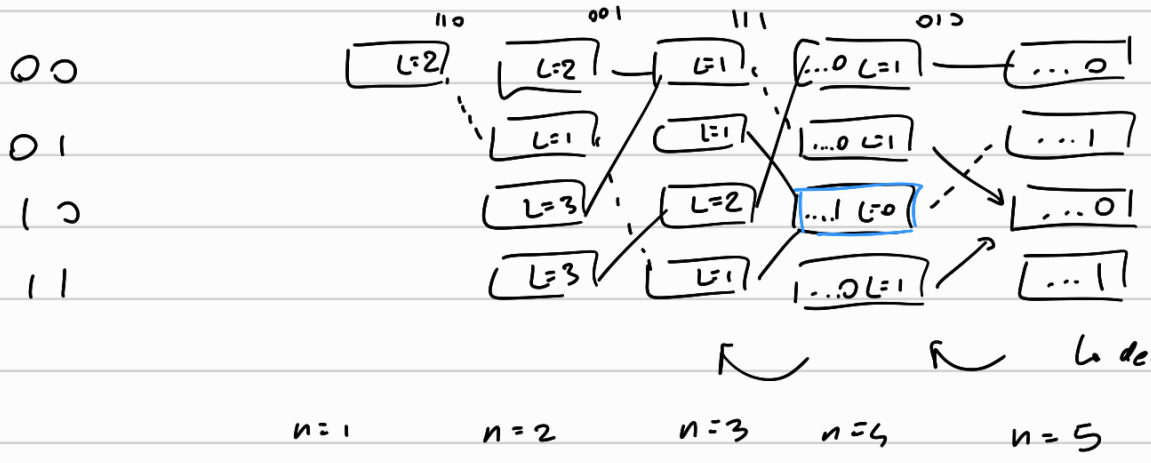
Osservando la coda della soluzione come suggerito dall'algoritmo di Viterbi posso trovare la decodifica ottimale

/12-05

1101 → (111) (001) (011) (010) codificato

(110) (001) (111) (010) ricevuto

dove L = errori



La decodifica va all'indietro (l'errore L viene sommato)

- l'alternativa sarebbe proseguire su 101 partendo da 10 con bit = 0, avrebbe distanza di Hamming > di quella con bit = 1

La programmazione dinamica permette, evitando a ritroso, di escludere strade che sicuramente non sono la soluzione