Gli errori sui singoli simboli sono più difficili da studiare. Infatti un errore di parola può corrispondere a uno o più errori di simbolo. Inoltre gli errori in diverse posizioni della parola sono in generale dipendenti. Nei casi pratici si valuta una frequenza media di errore di simbolo per parola. Ci limiteremo in queste note in un paragrafo successivo a considerare solo il caso binario ($d = 1$, *bit-error probability*).

Bit-error e Word Error probability

$$S^K \in \mathcal{A}^K = \{a_1^K, a_2^K, \ldots, a_{2^K}^K\}, \quad \hat{S} \in \mathcal{A}^K. \tag{7.102}$$

Il canale equivalente complessivo ha matrice di canale

$$\mathbf{P}_t = \left[Pr\{\hat{S}^K = a_i^K | S = a_j^K\}\right]_{i,j=1,\ldots,2^K} = [p_{ij}]_{i,j=1,\ldots,2^K}. \tag{7.103}$$

La probabilità di errore a livello di parola si esprime come

$$P(e_w) = 1 - \sum_{j=1}^{2^K} p_{jj}^K \Pi_j^K. \tag{7.106}$$

$$= 1 - (\operatorname{diag} \mathbf{P}_t)^T \boldsymbol{\Pi}_S^K$$

Se le prob. a priori sono uguali

$$Pr\{S^K = s_i^K\} = \frac{1}{2^K}$$

$$P(e_w) = 1 - (\operatorname{diag} \mathbf{P}_t)^T \underline{e} \cdot 2^{-K} \qquad \text{con } \underline{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

La $P(e_w)$ solo parzialmente rappresenta le prestazioni del sistema in quanto le parole binarie potrebbero differire per uno o più bit e in varie posizioni. Pertanto se le prestazioni devono essere valutate in termini di numero di bit errati, bisogna valutare il numero medio di bit in errore. Chiaramente le prestazioni dipendono anche dal tipo di codifica binaria adottata. Più in particolare abbiamo $A = \{$

Definiamo inoltre la variabile

$$d_H^K(s_i^K, s_j^K) = N_{ij}^K = \#\text{bit diversi tra } s_i^K \text{ e } s_j^K, \quad i,j = 1,...,2^K. \tag{7.104}$$

Pertanto la probabilità di errore per bit (*bit-error probability*) si può definire come

$$P_b = \frac{1}{K} E[\#\text{ di bit in errore}] = \frac{1}{K} \sum_{i,j=1}^{2^K} N_{ij}^K p_{ij}^K p_j^K, \tag{7.105}$$

dove p_j^K sono le probabilità dell'alfabeto esteso di ordine K alla sorgente.

$$= \frac{1}{K} \sum_{i=1}^{2^K} \sum_{j=1}^{2^K} d_H^K(s_i^K, s_j^K) P_r\{\hat{S}^K = s_i^K, S^K = s_j^K\}$$

$$= \frac{1}{K} \sum_{i=1}^{2^K} \sum_{j=1}^{2^K} d_H^K(s_i^K, s_j^K) P_r\{\hat{S}^K = s_i^K \mid S^K = s_j^K\} P_r\{S^K = s_i^K\}$$

$$= \frac{2^{-K}}{K} \underline{e}^T \left(D_H^K \odot P_t\right) \underline{e}$$

Esempio

Si consideri $K=3$. L'alfabeto esteso è

$$\mathcal{S}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

La matrice delle distanze di Hamming è

$D_H^3 =$

	000	001	010	011	100	101	110	111
000	0	1	1	2	1	2	2	3
001	1	0	2	1	2	1	3	2
010	1	2	0	1	2	3	1	2
011	2	1	1	0	3	2	2	1
100	1	2	2	3	0	1	1	2
101	2	1	3	2	1	0	2	1
110	2	3	1	2	1	2	0	1
111	3	2	2	1	2	1	1	0

Immaginiamo ora un canale che stabilisce le stringhe secondo la seguente matrice P_t (presa a caso)

$$P_t = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & \frac{4}{5} & 0 & 0 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{3} & 0 & \frac{1}{6} & 0 & \frac{1}{6} \\ 0 & \frac{1}{4} & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{5} & 0 & \frac{1}{5} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Assumendo prob. a priori uguali

$$P(e_\omega) = 1 - (\text{diag } P_t)^T \underline{e} = 0.3729$$

(Usando MATLAB)

$$P_b = \frac{1}{8 \cdot 3} \cdot \underline{e}^T (D_H \circledast P_t) \underline{e} = 0.1896$$

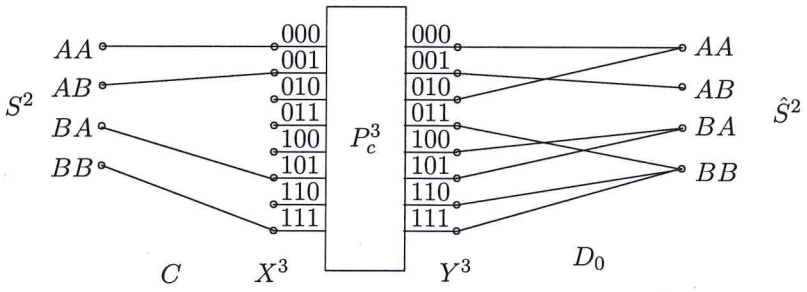
Se ad ogni errore di parola corrispondesse un nol bit errato avremmo $P_b = \frac{P(e_\omega)}{3} = 0.1214$

La situazione è un p' peggiore poiché mediamente

$KP_b = 3 \cdot 0.1896 = 0.5688$. Circa il 57% delle bit in ogni parola è in errore.

Il ricevitore **D**, va progettato, noto il canale e il codificatore, come quello che minimizza la probabilità di errore a livello di parola. Se le probabilità della sorgente sono note e non uniformi, si adotta un ricevitore MAP, altrimenti, più comunemente, un ricevitore MV. Non è sempre semplice gestire le matrici quando la lunghezza della parole codice cresce. Anche il progetto di codificatori **C** che consentano di utilizzare il canale vicino alla capacità è un problema ancora non completamente risolto. La complessità computazionale del progetto del codificatore, e dell'implementazione di codificatore e decodificatore vengono spesso gestiti mediante lgoritmi sub-ottimi. Torneremo su questo argomento dopo aver esaminato alcuni esempi.

Esempio 7.6 Si consideri l'alfabeto sorgente $\mathcal{A} = \{A, B\}$ con distribuzione dei simboli $\Pi_S = \{0.3, 0.7\}$. Si scelga $K = 2$ e $N = 3$ per canale binario con $p_e = 0.01$. Il codificatore operi la seguente associazione: $AA \rightarrow 000$, $AB \rightarrow 001$, $BA \rightarrow 110$, $BB \rightarrow 111$. Abbiamo $d = 2$, $n = m = 2$. Il rapporto di codifica è $R_c = 2/3$. La capacità per singolo uso del canale binario è $C = 1 - \mathcal{H}(p_e) = 0.9192$ bit. L'entropia della sorgente è $\mathcal{H}(S) = 0.1187$ bit. Il flusso informativo ogni due simboli di sorgente è $2\mathcal{H}(S) = 0.2374$ bit, su tre usi di canale con capacità $3C = 2.7576$ bit.



La matrice del codificatore a monte del canale è

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{7.89}$$

La cascata di codificatore e canale ha matrice equivalente

$$\mathbf{CP}_c^3 = \begin{pmatrix} 0.9703 & 0.0098 & 0.0098 & 0.0001 & 0.0098 & 0.0001 & 0.0001 & 0.0000 \\ 0.0098 & 0.9703 & 0.0001 & 0.0098 & 0.0001 & 0.0098 & 0.0000 & 0.0001 \\ 0.0001 & 0.0098 & 0.0000 & 0.0001 & 0.0098 & 0.9703 & 0.0001 & 0.0098 \\ 0.0000 & 0.0001 & 0.0001 & 0.0098 & 0.0001 & 0.0098 & 0.0098 & 0.9703 \end{pmatrix} \tag{7.90}$$

Le probabilità della sorgente estesa di ordine 2 sono $\boldsymbol{\pi}_S^2 = (0.0900, 0.2100, 0.2100, 0.4900)^T$. La matrice delle probabilità a posteriori dopo codificatore e canale è

$$\mathbf{P}_p = \begin{pmatrix} 0.9767 & 0.0043 & 0.9270 & 0.0013 & 0.2931 & 0.0000 & 0.0018 & 0.0000 \\ 0.0230 & 0.9855 & 0.0218 & 0.2987 & 0.0069 & 0.0098 & 0.0000 & 0.0000 \\ 0.0002 & 0.0100 & 0.0002 & 0.0030 & 0.6839 & 0.9674 & 0.0043 & 0.0043 \\ 0.0000 & 0.0002 & 0.0510 & 0.6970 & 0.0161 & 0.0228 & 0.9938 & 0.9956 \end{pmatrix} \tag{7.91}$$

Il ricevitore MAP, guardando alle colonne di $\mathbf{P}_p$ è

$$\mathbf{C}_0^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{7.92}$$

Anche il ricevitore MV guardando alle colonne di $\mathbf{CP}_c^3$ è lo stesso. In effetti ci sono alcune colonne con massimi uguali ma abbiamo mantenuto le scelte del ricevitore MAP. Il canale equivalente è descritto dalla matrice 4×4

$$\mathbf{P}_t = \mathbf{CP}_c^3\mathbf{D}_0 = \begin{pmatrix} 0.9801 & 0.0098 & 0.0099 & 0.0002 \\ 0.0099 & 0.9703 & 0.0099 & 0.0099 \\ 0.0001 & 0.0098 & 0.9801 & 0.0100 \\ 0.0001 & 0.0001 & 0.0099 & 0.9899 \end{pmatrix} \tag{7.93}$$

La probabilità di errore è

$$P(e_w) = 1 - diag(\mathbf{P}_t)^T \boldsymbol{\pi}_S^2 = 0.0172. \tag{7.94}$$

7.8 Il teorema inverso della codifica di canale

Ricordiamo come nel capitolo precedente (30)
siano state messe in relazione l'entropia
del canale e la probabilità di errore mediante
il teorema di Fano. Gli stessi risultati possono
essere applicati ora ad una catena più complessa
che include anche codificatore e decodificatore.
Ricordiamo che la capacità del canale a blocchi
è definita come

$$C_{can} = \max_{\pi_{X^N}} I(X^N; Y^N).$$

Nel caso tipico di $N=M$, abbiamo $C_{can} = C^N = \max_{\pi_{X^N}} I(X^N; Y^N)$
che più specificamente nel caso di N usi indipendenti
del canale diventa $C_{can} = NC$, dove $C = \max_{\pi_X} I(X;Y)$.
Il teorema inverso della codifica è il seguente.

Teorema 7.4 *Nel sistema di comunicazione di figura 7.9, vale la seguente disuguaglianza*

$$\mathcal{H}(S^K) \leq NC + \mathcal{H}(e_w) + P(e_w)\log_2(d^K - 1), \qquad (7.95)$$

dove $\mathcal{H}(e_w)$ è l'entropia binaria corrispondente alla probabilità di errore $P(e_w)$.

La figura 7.12 mostra la regione delle possibili coppie $\big(P(e_w), \mathcal{H}(S^K)\big)$. La condizione di $P(e_w) = 0$ è compatibile solo con la condizione $\mathcal{H}(S^K) < C_{can}$, ovvero il flusso informativo per simbolo $\frac{\mathcal{H}(S^K)}{K}$ deve essere inferiore alla capacità di canale. Ovviamente trasmissione con $\mathcal{H}(S^K)$ al di sopra della capacità di canale è possibile, ma è incompatibile con una $P(e_w) = 0$.

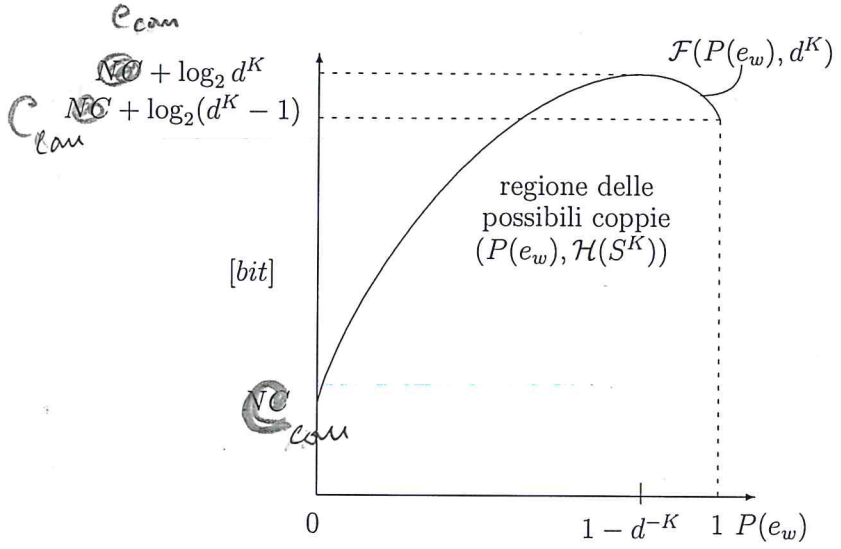


Figura 7.12: La regione delle possibili coppie $\left(P(e_w), \mathcal{H}(S^K)\right)$ del teorema inverso della codifica di canale

Prova: Poiché $\mathcal{H}(S^K|\hat{S}^K) = \mathcal{H}(S^K) - \mathcal{I}(S^K; \hat{S}^K)$, e dal teorema del trattamento dati $\mathcal{I}(S^K; \hat{S}^K) \leq \mathcal{I}(X^N; Y^N)$, abbiamo

$$\mathcal{H}(S^K|\hat{S}^K) \geq \mathcal{H}(S^K) - \mathcal{I}(X^N; Y^N). \tag{7.96}$$

Poiché gli usi del canale sono indipendenti $\mathcal{I}(X^N; Y^N) \leq NC$, pertanto

$$\mathcal{H}(S^K) - NC \leq \mathcal{H}(S^K|\hat{S}^K). \tag{7.97}$$

Ma $\mathcal{H}(S^K|\hat{S}^K)$ può essere maggiorato usando la disuguaglianza di Fano

$$\mathcal{H}(S^K|\hat{S}^K) \leq \mathcal{H}(e_w) + P(e_w)\log_2(d^K - 1), \tag{7.98}$$

da cui il risultato. □

Il teorema è un risultato inverso poiché ci dice che la condizione di trasporto affidabile con $P(e_w) = 0$ è compatibile solo con un flusso che è al di sotto della capacità, ma non ci dice se tale condizione sia ottenibile e quale debba essere il codificatore. La risposta alla prima domanda sarà fornita dal teorema della codifica (II teorema di Shannon), il progetto del codificatore resta ancora in parte un problema aperto.

Il teorema inverso può essere riscritto come

$$\frac{H(S^K)}{K} \leq \frac{C_{com}}{K} + \frac{H(e_w)}{K} + \frac{P(e_w)}{K}\log_2(d^K - 1)$$

dove a sinistra della disuguaglianza compare il tasso entropico della sorgente. $\frac{C_{com}}{K}$ può essere interpretato come la capacità del canale mediante utilizzato da ogni simbolo di sorgente.

Le

Assumendo che i simboli alla sorgente siano emessi in maniera indipendente $\mathcal{H}(S^K) = K\mathcal{H}(S)$.

Per inoltre $M=N$ e gli ingressi del canale sono indipendenti, abbiamo $C_{com} = NC$ e

, il risultato del teorema inverso si può scrivere

$$\mathcal{H}(S) \leq \frac{N}{K}C + \frac{\mathcal{H}(e_w)}{K} + \frac{P(e_w)}{K}\log_2(d^K - 1) \qquad (7.99)$$

(Attenzione che anche se i simboli di S^K sono indipendenti non lo sono quelli delle parole codice X^N!! Analogamente i simboli di ogni parola Y^M non sono indipendenti anche se gli usi del canale sono indipendenti!!

La situazione con $P(e_w) = 0$ è compatibile solo con la condizione sul rapporto di codifica $R_c = K/N$,

$$R_c \leq \frac{C}{\mathcal{H}(S)}. \qquad (7.100)$$

Si noti come nel caso binario a simboli equiprobabili ($\mathcal{H}(S) = 1$), la condizione è $R_c < C$.

7.9 Caso binario con canale BSC.

(39)

Esaminiamo qui con maggiore dettaglio lo scenario di sorgente e canale che operano escusivamente su alfabeti binari ($d = n = m = 2$). Il codificatore associa ad ogni sequenza di K bit una sequanza di N bit. Dalla sequanza di N bit all'uscita del canale, il decodificatore deve recostruire la parola binaria di K bit emessa dalla sorgente. La probabilità di errore a livello di parola (word-error), è

$$P(e_w) = Pr\{S^K \neq \hat{S}^K\}. \tag{7.101}$$

dove S^K e $\hat{S}^K$ sono stringhe binarie di lunghezza K.

Dall'esempio 7.6 si può immediatamente evincere che la trattazione generale del caso binario con il calcolo delle probabilità a posteriori può diventare impraticabile già per K e N dell'ordine delle decine. Infatti l'alfabeto delle stringhe binarie estratte dalla sorgente è

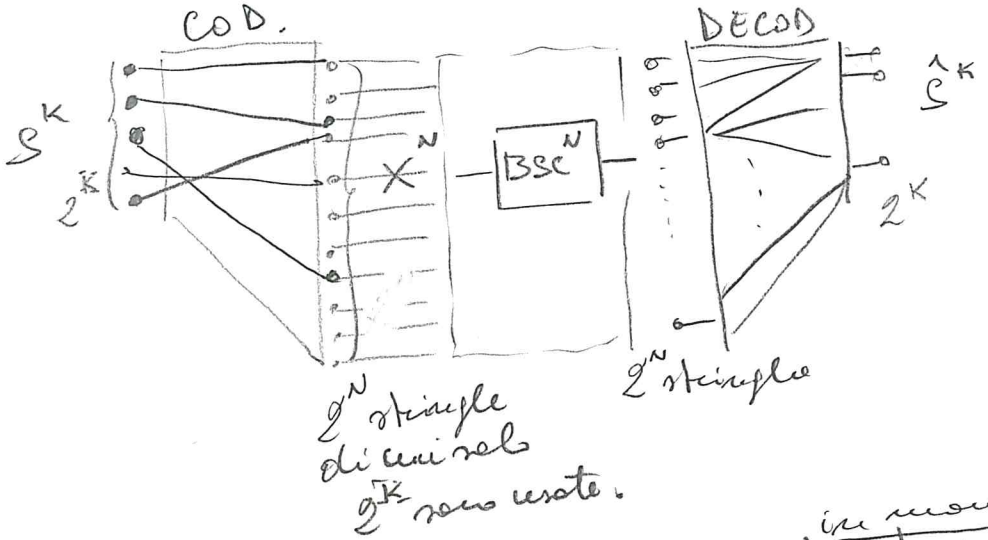
$$\mathcal{S}^K = \{s_1^K, s_2^K, \dots, s_{2^K}^K\} = \underbrace{\mathcal{S} \times \mathcal{S} \times \dots \times \mathcal{S}}_{K}.$$

Lo spazio delle stringhe X^N è un sottoinsieme di

$$\mathcal{X}^N = \{x_1^N, x_2^N, \dots x_{2^N}^N\} = \mathcal{X} \times \mathcal{X} \times \dots \times \mathcal{X},$$

dove sia $\mathcal{S}$ che $\mathcal{X}$ sono $\{0,1\}$.

Immaginiamo ora di essere in presenza di un canale binario simmetrico (BSC) con prob. di errore P_e usato N volte in maniera indipendente su ogni bit delle parole codice.
Le parole codice contaminate dal canale saranno ancora binarie di lunghezza N appartenenti allo spazio $\mathcal{Y}^N = \{y_1^N, y_2^N, \dots, y_{2^N}^N\}$ con $\mathcal{Y} = \{0,1\}$.

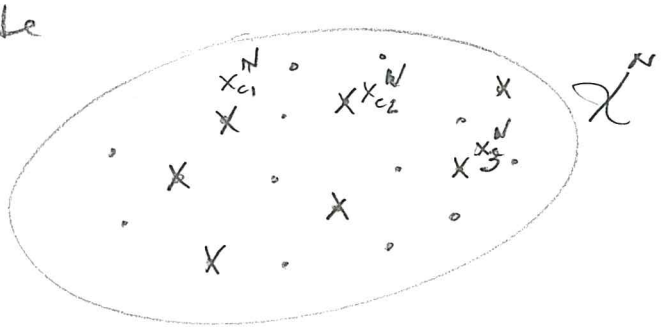


Quindi il codificatore associa in maniera univoca ogni stringa di lunghezza K una stringa di lunghezza N con N>K. Pertanto alcune stringhe dello spazio X^N non vengono utilizzate. Il canale binario inverte in maniera indipendente i vari bit restituendo una stringa Y^N di lunghezza N che in teoria può essere una qualunque delle stringhe di Y^N. Il decodificatore, dalla conoscenza della struttura del codificatore e del canale deve elaborare una strategia per associare ad ogni stringa di Y^N una stringa di Y^K in modo da minimizzare la probabilità di errore P(ew).

Definiamo con X_c^N l'insieme delle 2^K parole codice

$$X_c^N = \{ x_{c1}^N, x_{c2}^N, \dots, x_{c2^K}^N \} \subset X^N = \{ x_1^N, \dots x_{2^N}^N \}$$

i cui elementi sono rappresentati indicato in figura con delle croci



la generica parola codice è indicata come $x_c^N \in X_c^N$

Il canale farà aleatoriamente "saltare" alcune parole codice dalla loro posizione in un'altra posizione dello stesso spazio (Ricordare che $Y^N = X^N$).

Dalla figura si evince che se è definita una regione di distanza una immediata interpretazione geometrica suggerisce che:

(1) le 2^K parole codice devono essere più separate possibile nello spazio X^N.

(2) Se il canale BSC è tale da "mostrare" di poco le parole codice dalla loro posizione, il decodificatore può operare decidendo per la parola codice X_C^N più vicina e quindi per la stringa S^K corrispondente.

Vogliamo dimostrare che per un canale binario simmetrico (con $p_e < \frac{1}{2}$) e sorgente indipendente uniforme binaria, la regola a minima probabilità di errore è la regola a minima distanza.

$$\hat{S}_o^K = C^{-1}(\hat{X}_{Co}^N) \quad \text{con} \quad \hat{X}_{Co}^N = \underset{X_C^N \in \mathcal{X}_C^N}{\arg\min} \, d_H(Y^N, X_C^N)$$

con $d_H(\alpha,\beta)$ definita come la distanza di Hamming tra due stringhe α e β della stessa lunghezza. (# dei bit discordi). La prova di questo risultato è semplice e segue il seguente ragionamento.

Ricordiamo che la minima probabilità di errore $P(e_W)$ corrisponde alla regola MAP (massima probabilità a posteriori) ovvero, osservata una stringa $Y^N = y^N$, decidi per la parola codice che corrisponde alla massima probabilità a posteriori

$$\hat{X}_{Co}^N = \underset{X_{Ci}^N \in \mathcal{X}_N^C}{\arg\max} \, P_r\{X^N = X_C^N \mid Y^N = y^N\}$$

Poiché le parole codice sono equiprobabili in quanto la sorgente è uniforme, la regola è equivalente alla regola MV

$$\hat{X}_{CD}^N = \underset{X_C^N \in \mathcal{X}_C^N}{\operatorname{argmax}} \underbrace{P_R\{Y^N = y^N \mid X^N = X_C^N\}}_{\text{verosimiglianza}}$$

La verosimiglianza si può scrivere

$$P_R(Y^N = y^N \mid X^N = X_C^N) = p_e^{d_H(X_C^N, y^N)} (1-p_e)^{N - d_H(X_C^N, y^N)}$$

poiché d_H bit vengono "flippati" in maniera indipendente con probabilità p_e dal canale binario.

Per concludere la prova basta dimostrare che per $p_e < \frac{1}{2}$ $P_R\{Y^N = y^N \mid X^N = x_C\}$ è una funzione decrescente di d_H.

Prova: Per $d_1 < d_2$ vogliamo dimostrare che

$$p^{d_1}(1-p)^{N-d_1} > p^{d_2}(1-p)^{N-d_2}$$

Riscrivendo

$$p^{d_1 - d_2} > (1-p)^{N - d_2 - N + d_1}$$

$$\left(\frac{p}{1-p}\right)^{d_1 - d_2} > 1 \quad ; \quad \left(\frac{1-p}{p}\right)^{d_2 - d_1} > 1$$

Poiché $d_2 - d_1 > 0$ basta dimostrare che

$$\frac{1-p}{p} > 1 \Rightarrow 1 - 2p > 0 \Rightarrow p < \frac{1}{2} \quad \square$$

La condizione sul canale che $p_e < \frac{1}{2}$ non è restrittiva poiché: (a) in genere la probabilità è piccola e quindi è certamente $< \frac{1}{2}$; (b) se non lo fosse avremmo un canale che "inverte molto" e che è equivalente ad un canale dove i simboli sono ridefiniti in maniera invertita.

INTRODUZIONE INTUITIVA AL TEOREMA DELLA CODIFICA

Intro.....

Cominciamo con un esempio. Consideriamo il seguente canale che rappresenta un ipotetico "Dattilografo imperfetto" (Noisy typewriter, esempio famoso)

Ci sono 26 simboli all'ingresso $\{A, B, C, \ldots\}$ (lettere dell'alfabeto inglese) e gli stessi simboli in uscita che però possono risultare da una confusione dattilografica con il simbolo precedente. E' come se il dattilografo sbagli nel pigiare il tasto e con probabilità $\frac{1}{2}$ pigia quello successivo. Lo schema è riportato in figura (a) con la lettera Z "arcolorosamente" confusa con la A.

Dal Cover & Thomas.

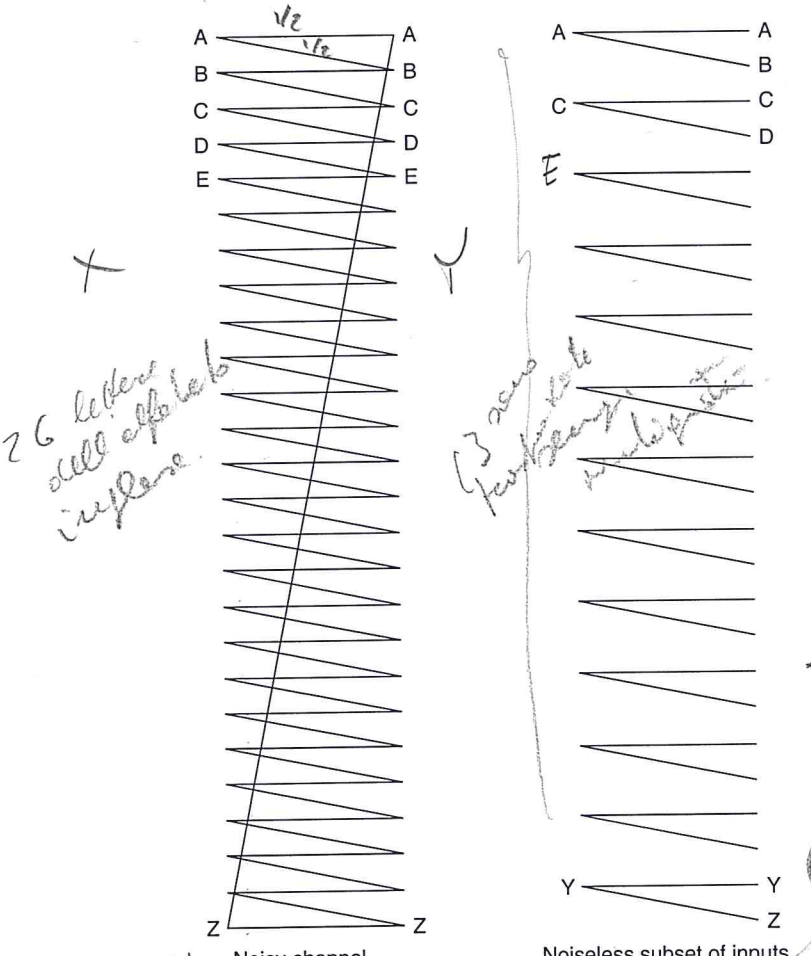


FIGURE 7.4. Noisy Typewriter. C = log 13 bits.

→ $\max I(X;Y) = \log_2 13$ è la capacità

→ si tratta di un canale senza rumore.

L'osservazione cruciale che possiamo fare è che se invece di usare tutte e 26 le lettere ne usassimo solo dello stesso insieme di 13 {A, C, E, ...} il canale non introdurrebbe alcuna ambiguità in quanto dall'osservazione della lettera in uscita si può risalire perfettamente al simbolo di ingresso ($p(e)=0$!!) lo schema equivalente è mostrato in Figura (b). È come se il canale numerico fosse stato "ridotto" a un canale senza rumore.

Interessante è anche il calcolo della capacità.

$$C = \max_{\pi_x} I(X;Y) = \max_{\pi_x} \left(H(Y) - H(Y|X)\right)$$

Ma $H(Y|X) = 1$ bit, quindi $C = \max_{\pi_x} (H(Y) - 1)$

Il massimo di $H(Y)$ si ottiene per π_x uniforme ed è $\log_2 26$. Quindi la capacità del canale è

$$C = \log_2 26 - 1 = \log_2 26 - \log_2 2 = \log_2 13$$

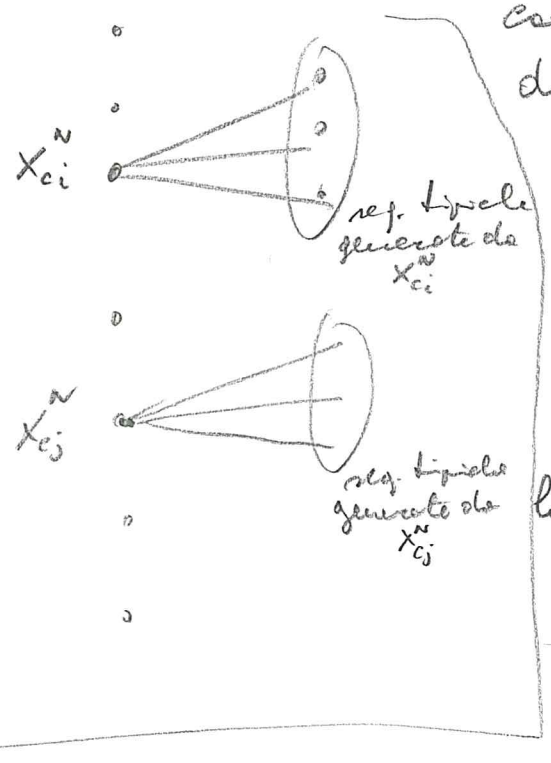
Esattamente 13 simboli sono trasportabili perfettamente dal canale.

L'esempio del dattiloscritto imperfetto è particolarmente illuminante perché ci può dimostrare che tutti i canali di rumore di N si comportano come dei dattiloscritti imperfetti.

Infatti se osserviamo lo schema di Fano, ogni parola codice genera in uscita un numero di stringhe pari circa a

$$2^{H(Y^N|X_{ci}^N)}$$

e queste equiprobabili (AEP).

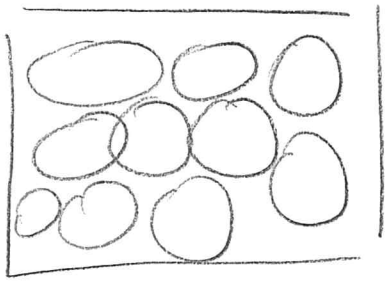


Ora se le parole codice sono equiprobabili e generano tutte quante lo stesso numero di seq. tipiche, abbiamo che mediamente

$$2^{H(Y^N|X^N)}$$

saranno le seq. tipiche generate da ogni parola codice

Il numero totale di sequenze tipiche in uscita è $2^{H(Y^N)}$. Tra queste vogliamo che le sequenze tipiche delle varie parole codice si aggiustino senza sovrapporsi come mostrato in figura



Quindi $\frac{2^{H(Y^N)}}{2^{H(Y^N|X^N)}}$ dovrebbe essere il numero massimo di parole codice da impacchettare senza sovrapposizioni (senza errori

Pertanto $\frac{2^{H(Y^N)}}{2^{H(Y^N|X^N)}} = 2^{H(Y^N)-H(Y^N|X^N)} = 2^{I(X^N;Y^N)} \simeq$ # di seq. tipiche effettivamente.

Ecco come $I(X^N; Y^N)$ come il rapporto di capacità di can

(46)

Quindi il problema del codificatore è di scegliere un set di parole codice che producano insiemi tipici in uscita al canale non sovrapposti e quindi risolvibili esattamente. E' come rendere il canale equivalente a un canale senza rumore.

La scelta del codificatore è tutt'altro che banale, ma è possibile dimostrare in generale l'esistenza di un codificatore note la capacità di canale.

IL TEOREMA DELLA CODIFICA DI CANALE PER IL BSC

Presentiamo qui il II teorema di Shannon con riferimento al canale BSC e a parole codice equiprobabili.
Le estensioni possono considerate in seguito.

TEOREMA:
Per un BSC con prob. di errore p_e, ovvero capacità $C = 1 - H(p_e)$ bit, $\forall \varepsilon > 0$ piccolo a piacere, $\exists N$ tale che è possibile selezionare $M = 2^{N(C-\varepsilon)}$ parole codice dalle 2^N all'ingresso del canale esteso tali che la probabilità d'errore di decodifica ($p(ew)$) sia arbitrariamente piccola.

Prima di intraprendere la dimostrazione della prova notiamo che il Teorema prevede l'esistenza di un codice che garantisce un trasporto assolutamente affidabile, ma non suggerisce come costruire il codificatore. Risulta intuitivamente le parole codice devono essere il più possibile spaziate tra di loro, ma questo è un po' troppo vago per essere sufficientemente per la costruzione di un codificatore.

C'è inoltre un altro problema connesso alla decodifica. Abbiamo dimostrato che la regola ottima di decodifica è quella a minima distanza di Hamming. Anche se è di facile comprensione, la regola può diventare inapplicabile quando la dimensione dell'alfabeto di codice cresce: Una applicazione immediata della regola richiede che per ogni sequenza y^N ricevuta si calcoli la distanza da tutte le parole codice di X_c^N per trovare la decisione finale. Il numero di parole codice $|X_c^N| = 2^{kR}$ potrebbe essere intrattabile già per k uguale a qualche decina.

La scelta di codificatore e decodificatore viene in genere nelle applicazioni determinata come risultato di un compromesso tra complessità della decodifica e prestazioni.

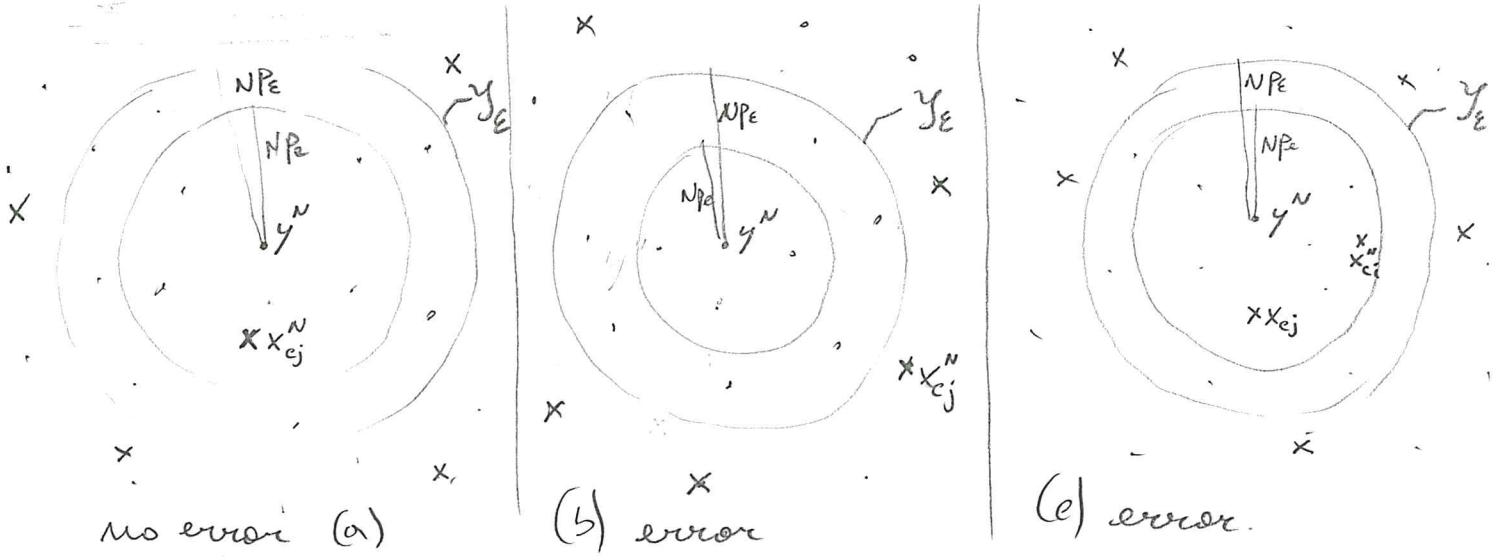
Tornando alla prova del teorema anche qui c'è un problema in quanto l'analisi del ricevitore ML è un pò complessa per la natura introduttiva di queste note. Seguendo la traccia lasciata da Shannon nei suoi primi lavori analizziamo la decodifica con un ricevitore sub-ottimo. Dimostreremo che anche se le prestazioni di questo ricevitore non sono buone come quelle del ricevitore ML, valgono ancora i risultati del teorema, ovvero la prob. di errore può essere resa arbitrariamente piccola.

La prima osservazione da fare è che per ogni sequenza trasmessa x_c^N in uscita solo in linea di principio potrà essere una qualunque delle 2^N sequenze di y^N. In realtà solo quelle condizionalmente tipiche, dato x_c^N, si potranno verificare con probabilità non trascurabile. Esse sono ovviamente a distanza media da x_c^N pari a NP_e.

Quindi decidiamo di cercare solo in un intorno della parola ricevuta y^N entro un raggio appena superiore alla media NP_E con $P_E = P_e + \varepsilon$.

La regola sarà (regola "sphere decoding" o "decodifica tipica"):

Ricevuta una parola y^N se c'è una parola codice nell'intorno, di raggio NP_E quella sarà la nostra decodifica. Dichiareremo errore se non ce n'è nessuna o se ce ne sono più di una.



La Figura mostra graficamente le situazioni per parola codice come x_{cj}^N: (a) x_{cj}^N cade nella sfera di ricerca ed è l'unica;
(b) Nessuna parola codice cade nella sfera; (c) x_{cj}^N cade nella sfera ma non è l'unica.

La regola adottata è un po' strana ed è decisamente sub-ottima. Infatti nelle situazioni in cui si dichiara errore si potrebbe cercare meglio, ovvero prendere la parola codice più vicina cercando anche al di fuori della sfera in (b), o tra quelle contenute nella sfera in (c).
Dimostreremo comunque che tale regola può ottenere una prob. di errore arbitrariamente piccola.

Fratteggiando che sia stata trasmessa la parola x_{cj}^N, consideriamo prob. di errore condizionata

$$P(e_\omega | X^N = x_{cj}^N) \triangleq P_j(e_\omega)$$

Seguendo la regola di decodifica adottata, abbiamo

$$P_j(e_\omega) = P_r\{ x_{cj}^N \notin Y_\varepsilon \} + P_r\{ x_{cj}^N \in Y_\varepsilon \} \cdot P_r\{ \text{almeno un'altra parola codice appartiene a } Y_\varepsilon \}$$

Poiché $P_r\{ x_{cj}^N \in Y_\varepsilon \} < 1$, possiamo maggiorare

$$P_j(e_\omega) < P_r\{ x_{cj}^N \notin Y_\varepsilon \} + P_r\{ \text{almeno un'altra parola codice appartiene a } Y_\varepsilon \}$$

Inoltre usando il bound dell'unione abbiamo che

$$P_r\{ \text{almeno un'altra parola codice appartiene a } Y_\varepsilon \} \leq \sum_{\substack{i=1 \\ i \neq j}}^{M} P_r\{ x_{ci}^N \in Y_\varepsilon \}$$

Quindi la maggiorazione diventa

$$P_j(e_\omega) \leq \underbrace{P_r\{ x_{cj}^N \notin Y_\varepsilon \}}_{(I)} + \sum_{\substack{i=1 \\ i \neq j}}^{M} P_r\{ x_{ci}^N \in Y_\varepsilon \}$$

Concentriamoci per ora sul termine (I).

$$P_r\{ x_{cj}^N \notin Y_\varepsilon \} = P_r\{ \text{più di } N(p_e+\varepsilon) \text{ bit nella parola codice sono stati invertiti} \}$$

Poiché Np_e è il numero medio di bit invertiti, fissato ε, quando N cresce ci sarà un numero sempre più piccolo di parole che eccede $N\varepsilon$ oltre la media. Più precisamente per la legge debole dei grandi numeri, $\forall \varepsilon, \delta > 0 \ \exists N_0$: per $N > N_0$ la prob. che il numero di errori ecceda la media di più di $N\varepsilon$ è $< \delta$.

Quindi per N grande $P_r\{ x_{cj}^N \notin Y_\varepsilon \} < \delta$ con δ piccolo a piacere

Quindi $$P_j(e_\omega) < \delta + \sum_{\substack{i=1 \\ i \neq j}}^{M} P_r\{ x_{ci}^N \in Y_\varepsilon \}$$

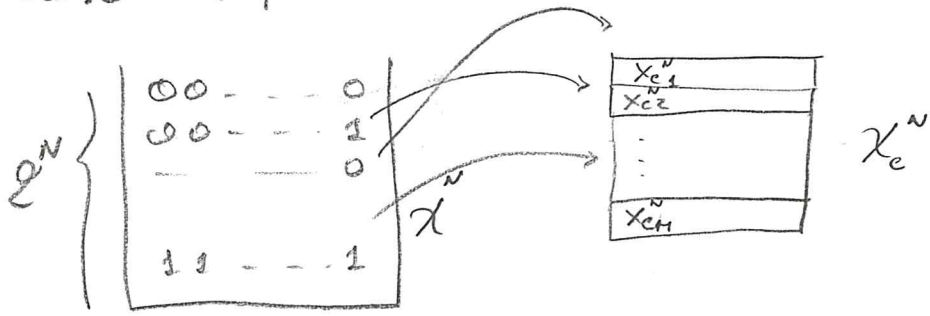
Per prima cosa notiamo che δ non dipende dalla particolare scelta delle parole codice. Viceversa l'altro termine dipende da come le parole sono state distribuite in $\mathcal{X}^N$.

Quindi come fare per svincolarci dalla particolare scelta di $\mathcal{X}_c^N$?

L'idea di Shannon fu quella di considerare delle prestazioni mediate nell'insieme dei possibili codici per dimostrare che mediamente la $P(e_w)$ tende a zero. È evidente che se la media di $P(e_w)$ va a zero, deve esistere qualche codice per cui ciò avviene puntualmente.

Questo tipo di ragionamento è quello che conduce al risultato che è anche noto come teorema della codifica canale

Supponiamo quindi di selezionare un codice prendendo a caso M parole codice dall'una delle 2^N stringhe di $\mathcal{X}^N$.



Supponiamo che la estrazione sia con rimpiazzo permettendo eventualmente che possa essere estratto un codice selezionato con qualche parola codice ripetuta. Avere considerato una estrazione con rimpiazzo semplifica l'analisi e comunque l'emergenza di un codice selezionato è un evento a probabilità molto piccola per $M \ll 2^N$.

Tenendo, il numero di codici selezionabili è

$$\left(2^N\right)^M = 2^{NM}$$

(# di modi di selezionare con rimpiazzo M elementi da un'urna di 2^N elementi)

Poiché tutto avviene in maniera indipendente e uniforme, la probabilità di ogni codice è 2^{-NH}.

Studiamo ora la espressione per la prob. di errore su tutti i possibili codici

$$\overline{P_j(e_w)} \leq \delta + (M-1)\, P_r\{x_{cj}^N \in \mathcal{Y}_\varepsilon\}$$

← media sui codici.

Semplifichiamo ulteriormente

$$\overline{P_j(e_w)} \leq \delta + M\, \overline{P_r\{x_{cj}^N \in \mathcal{Y}_\varepsilon\}}$$

Ma la prob. media che una sequenza sia contenuta in una sfera di raggio NP_ε è uguale al rapporto

$$\overline{P_r\{x_{ci}^N \in \mathcal{Y}_\varepsilon\}} = \frac{N(NP_\varepsilon)}{2^N}$$

← # di parole contenute in una sfera di raggio NP_ε centrata su y^N.

Cerchiamo ora di ottenere un bound per $N(NP_\varepsilon)$

Il numero di sequenze di lunghezza N ad una distanza di esattamente K bit da y^N è semplicemente il numero di possibili modi in cui K su N bit possono essere invertiti, ovvero $\binom{N}{K}$. Sommando fino a NP_ε

$$N(NP_\varepsilon) = 1 + \binom{N}{1} + \binom{N}{2} + \cdots + \binom{N}{NP_\varepsilon},$$

ma poiché in generale NP_ε non è intero

$$N(NP_\varepsilon) \leq \underbrace{\sum_{K=0}^{\lceil NP_\varepsilon \rceil} \binom{N}{K}}_{\text{Dis. nota.}} \leq 2^{NH(P_\varepsilon)} \qquad P_\varepsilon < \frac{1}{2}$$

Quindi

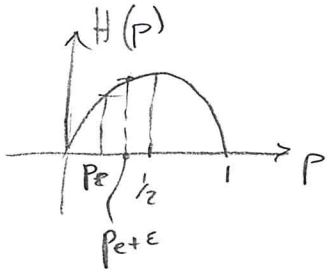
$$\overline{P_j(e_\omega)} \le \delta + M\, e^{-N(1-H(p_\varepsilon))}$$

(M: cresce con N; $e^{-N(\cdots)}$: decresce con N)

Affinché il bound vada a zero
M deve crescere meno di quanto decresce l'altro
termine. Questo si verifica se

$$\log_2 M < N(1+H(p_\varepsilon))$$

Ma $1+H(p_\varepsilon) \le (1-H(p_e))$ $\Longleftarrow$



Quindi $\log_2 M < N(1-H(p_e))$

$$M < 2^{N\underbrace{(1-H(p_e))}_{\text{capacità del canale binario}}} = 2^{NC} \qquad \boxed{C}$$

Quindi si può ottenere una prob. di errore effettivamente piccola (in media) se $M < 2^{NC}$. Quindi deve esistere qualche codice per cui questo è vero puntualmente.

Ricordando che nello shannon generale $M = 2^K$,

$$\frac{K}{N} < C \quad \text{ovvero il tasso di codifica } R_c = \frac{K}{N}$$

deve essere inferiore alla capacità del canale.

CONSIDERAZIONI GENERALI SUL TEOREMA DELLA CODIFICA

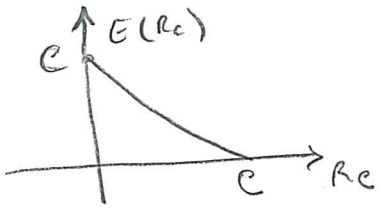
Notare che $P(e_w) \leq M\, 2^{-NC}$ visto che $M = 2^K$

si può scrivere $P(e_w) < 2^{K-NC} = 2^{-N(C-\frac{K}{N})}$ ovvero R_c

$$P(e_w) < 2^{-NE(R_c)}$$

$E(R_c)$ prende il nome di esponente della codifica

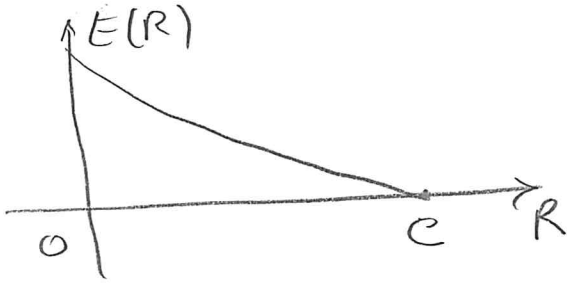
Nel nostro caso $E(R_c) = C - R_c$



Ma in generale si può dimostrare che la prob di errore risulta sempre un bound del tipo

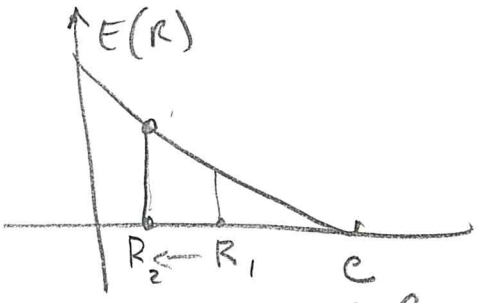
$$P(e_w) \leq 2^{-NE(R)}$$

dove $R = R_c H(S)$ e $E(R)$ è una funzione convessa U decrescente non negativa di R con $0 < R \leq C$

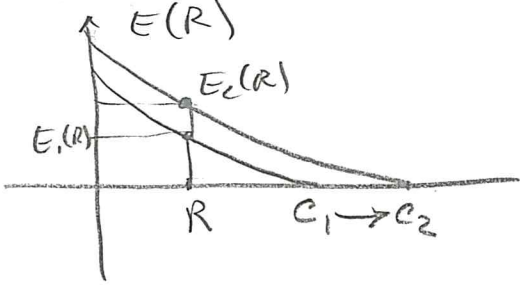


Quindi per un sistema di comunicazione per ottenere migliori prestazioni si possono adottare 3 strategie / guardo all'esame

(1) Riduci R riducendo $R_c = \frac{K}{N}$, ovvero aumento la ridondanza, ovvero per una data codifica uso il canale più spesso, ovvero uso un canale a banda più larga, oppure riduci il tasso informativo da trasmettere. Sul grafico questo corrisponde a spostare l'esercizio a cui si opera



(2) Migliora il canale / migliori aumentando la potenza (o la banda) — quindi la capacità



(3) Aumenta N mantenendo invariato $R_c = \frac{K}{N}$
Quindi purché si sia sotto capacità basta aumentare N per ottenere prestazioni sempre migliori!! Questo ovviamente comporta un aumento della complessità computazionale nella decodifica, ma è un risultato fondamentale della teoria dell'informazione.
Il miglioramento dipende solo dalla nostra capacità computazionale!!

le difficoltà sono tanto più grandi quanto più ci si avvicina al limite della capacità dove l'esponente è più piccolo. (55)
In tal caso è necessario usare N molto grande per ottenere prestazioni soddisfacenti.
Molta della teoria dei codici si concentra in questi anni nella ricerca di codici "semplici" e quindi con buone prestazioni, ma che presentino complessità di decodifica trattabili (codici TURBO, codici a bassa densità LDPC, etc.).