

PARTE QUARTA

ELEMENTI

DI

TEORIA DEI CODICI LINEARI

Codificare e decodificare messaggi per permettere una loro rapida trasmissione é un'antica esigenza dell'umanitá. I metodi usati nel passato sono stati i piú svariati; si pensi ai *tam-tam* delle popolazioni indigene africane o ai *segnali di fumo* degli indiani d'America. Oggi, l'era della trasmissione dell'informazione in *tempo reale*, i mezzi che abbiamo a disposizione per inviare e ricevere messaggi hanno raggiunto un grado di sofisticazione molto elevato e il loro funzionamento si basa sull'alta tecnologia e su precise teorie matematiche, fisiche ed informatiche.

Una teoria matematica che permette, in alcuni casi, di correggere automaticamente gli errori che possono verificarsi durante la trasmissione di informazioni é la *teoria dei codici lineari*, della quale intendiamo esporre i primi elementi. Questa si presenta come una serie di naturali applicazioni degli argomenti trattati nei precedenti capitoli e, cosa molto interessante, i suoi metodi sono alla base di risultati profondi in teoria dei disegni, come ad esempio la non esistenza di un piano proiettivo d'ordine 10.

Capitolo 13

Generalit  sui codici

13.1 Prime definizioni ed esempi

Un insieme finito F con q elementi, nel linguaggio della teoria dei codici, prende il nome di *alfabeto finito con q lettere*, le lettere essendo gli elementi di F . Una *parola su F di lunghezza n*   una successione finita $a_1a_2\dots a_n$ di lettere di F .

Nel seguito, per comodit  di scrittura, identificheremo una parola $a_1a_2\dots a_n$ con l' n -pla corrispondente $\mathbf{a} = (a_1, a_2, \dots, a_n)$; cos  ogni parola di lunghezza n potr  essere considerata come un elemento di F^n .

DEFINIZIONE 13.1.1. Un *codice* C su un alfabeto F   un qualsiasi sottoinsieme finito e non vuoto di parole su F . Esso prende il nome di *codice a blocchi* se le sue parole hanno tutte la stessa lunghezza; nel caso contrario si dice *a lunghezza variabile*. La comune lunghezza delle parole di un codice a blocchi si chiama *lunghezza del codice*. \square

Riportiamo alcuni esempi di codici molto comuni i cui nomi almeno sono sicuramente noti al lettore.

ESEMPIO 13.1.2. Il *Codice Fiscale Italiano*   un codice su un alfabeto di 36 lettere (le 26 dell'alfabeto inglese e le cifre decimali da 0 a 9). Le sue parole servono a codificare qualunque persona o ente abbia rapporti con il sistema fiscale italiano. Nel caso di una persona fisica la parola corrispondente   composta da 16 lettere: le prime 6 si riferiscono a cognome e nome, il secondo gruppo di 5 individua la data di nascita e il sesso, il successivo gruppo di 4 individua la localit  italiana o lo stato estero di nascita e l'ultima, che   di controllo, si calcola mediante un opportuno algoritmo sulle prime 15. \square

ESEMPIO 13.1.3. Il *codice ISBN (International Standard Book Number)*   un codice a blocchi C di lunghezza 10 sull'alfabeto di undici lettere costituite dalle cifre decimali da 0 a 9 e dalla lettera X ed   usato per codificare i libri in commercio. Quasi ogni libro, infatti, che non sia troppo vecchio, presenta stampata sul retro della copertina una successione di dieci cifre, che   il suo *ISBN* e cio  la parola di C ad esso associata. Lo schema di codifica, ad esempio, di un

libro scritto in inglese é il seguente: la prima lettera a_1 di una parola \mathbf{a} é zero e corrisponde alla lingua inglese; le due lettere successive a_2a_3 individuano la casa editrice; le sei lettere successive $a_4a_5a_6a_7a_8a_9$ indicano un numero assegnato al libro dalla casa editrice; l'ultima lettera a_{10} é di controllo ed é uguale al resto r della divisione per 11 dell'intero

$$a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 + 6a_6 + 7a_7 + 8a_8 + 9a_9$$

se $0 \leq r \leq 9$, é invece uguale ad X se risulta $r = 10$. Come si vede, l'ultima lettera di una parola del codice *ISBN* é l'unica che può assumere il valore X. Per esempio, il libro di *E.F.Assmus* e *J.D.Key* dal titolo *Designs and Their Codes*, pubblicato dalla *Cambridge University Press*, ha *ISBN* uguale a 0-521-41361-3. Ciò significa che la casa editrice *Cambridge University Press* é codificata con "52". Da notare che i trattini che separano alcuni gruppi di cifre dell'*ISBN* non hanno alcun significato al fine della codifica. Per maggiori informazioni sul codice *ISBN* si rimanda al sito WEB: <http://www.alice.it/bookshop/law.bks/codiinte.htm> . \square

ESEMPIO 13.1.4. Il *codice Morse* é un codice a lunghezza variabile sull'alfabeto di tre lettere

$$F = \{\bullet, -, \text{spazio}\}.$$

Esso serve a codificare le lettere dell'alfabeto inglese ed é stato molto usato nel passato soprattutto per trasmettere messaggi con il telegrafo senza fili. Il codice, riportato in tabella, é stato costruito in modo che una sua parola é tanto piú lunga quanto la lettera corrispondente é meno frequente nella lingua inglese.

Figura 13.1: Codice Morse

A	●—	B	—●●●	C	—●—●	D	—●●●
E	●	F	●●—●	G	— — ●	H	●●●●
I	●●	J	●— — —	K	— ● —	L	● — ●●
M	— —	N	— ●	O	— — —	P	● — — ●
Q	— — ● —	R	● — ●	S	●●●	T	—
U	●● —	V	●●● —	W	● — —	X	— ●● —
Y	— ● — —	Z	— — ●●				

Si noti, per esempio, che nella lingua inglese la lettera E é piú frequente della Z e quindi la parola del codice "●" corrispondente ad E é relativamente piú corta di quella " — — ●●" corrispondente a Z. Questo semplice accorgimento si presta chiaramente a rendere piú veloce la codifica, la trasmissione e la decodifica dei messaggi. Lo *spazio* che figura fra le lettere di F non é mai utilizzato per la codifica di una singola lettera ma é essenziale per *dividere* tra loro le parole del codice presenti in un messaggio codificato. Piú precisamente, quando si codifica una frase, bisogna inserire esattamente uno *spazio* tra ogni due lettere dell'alfabeto codificate ed almeno due spazi fra ogni due parole. Per esempio, se usiamo il simbolo @ per indicare uno *spazio*, l'espressione

CODICE MORSE

si codifica con

— • — • @ — — — @ — • • • @ • • • @ — • — • • @ • @ @ — — @ — — — @ • — • @ • • • @ •

Osserviamo infine che il codice Morse non distingue le lettere minuscole dalle maiuscole. □

ESEMPIO 13.1.5. L'American Standard Code for Information Interchange, noto come *Codice ASCII*, é il codice a blocchi sull'alfabeto $F = \{0, 1\}$ formato da tutte le parole di lunghezza sette e, quindi, contiene esattamente $2^7 = 128$ parole. Esso é stato costruito per codificare le lettere dell'alfabeto inglese maiuscole e minuscole, le cifre decimali da 0 a 9 e una serie di altri simboli e istruzioni allo scopo di permettere all'architettura interna di un computer di operare solo con i simboli 0 e 1. Se ad ogni parola di questo codice si aggiunge 0 o 1, a seconda che contenga un numero pari o dispari di 1 rispettivamente, si ottiene un codice a blocchi di lunghezza otto, detto *codice ASCII esteso*. Nella tabella che segue sono riportate le parole del codice *ASCII esteso* che corrispondono alle lettere maiuscole dell'alfabeto.

Figura 13.2: Codice ASCII esteso

<i>A</i>	10000010	<i>B</i>	10000100	<i>C</i>	10000111	<i>D</i>	10001000
<i>E</i>	10001011	<i>F</i>	10001101	<i>G</i>	10001110	<i>H</i>	10010000
<i>I</i>	10010011	<i>J</i>	10010101	<i>K</i>	10010110	<i>L</i>	10011001
<i>M</i>	10011010	<i>N</i>	10011100	<i>O</i>	10011111	<i>P</i>	10100000
<i>Q</i>	10100011	<i>R</i>	10100101	<i>S</i>	10100110	<i>T</i>	10101001
<i>U</i>	10101010	<i>V</i>	10101100	<i>W</i>	10101111	<i>X</i>	10110001
<i>Y</i>	10110010	<i>Z</i>	10110100				

□

Nel seguito prenderemo in considerazione soltanto codici a blocchi. Useremo, pertanto, il termine *codice* come sinonimo di *codice a blocchi*.

DEFINIZIONE 13.1.6. Un codice su un alfabeto con q lettere che contenga esattamente M parole di lunghezza n prende il nome di (n, M) -codice q -ario, o semplicemente di (n, M) -codice, se q risulta chiaro dal contesto. Nei casi $q = 2, 3$ il codice si dice rispettivamente *binario* e *ternario*. □

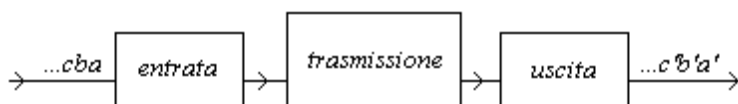
13.2 Canali di trasmissione

Per *canale di trasmissione*, o *di comunicazione*, intendiamo un sistema fisico in grado di accettare in una *entrata* le lettere di un alfabeto $F = \{a_1, a_2, \dots, a_q\}$ (*alfabeto di input*) e, in corrispondenza di ciascuna lettera accettata, emettere in una *uscita* lettere di un alfabeto $F' = \{b_1, b_2, \dots, b_t\}$ (*alfabeto di output*). Questo significa che in entrata il sistema possiede q possibili stati fisici

in corrispondenza biunivoca con le lettere dell'alfabeto F e una situazione analoga si riproduce in uscita con l'alfabeto F' . Osserviamo esplicitamente che escludiamo la possibilità che all'immissione di una lettera in input non corrisponda l'emissione di una lettera in output.

Noi, per semplicità, tratteremo il caso in cui gli alfabeti di input e di output coincidano e quindi, d'ora in avanti, supporremo sempre $F = F'$; ciò, come è facile convincersi, non lede di molto le generalità del problema. Inoltre, quando nel canale di trasmissione Σ si immettono successivamente le lettere $a_{i_1}, a_{i_2}, \dots, a_{i_n}$ e in uscita si trovano nell'ordine le lettere $a_{j_1}, a_{j_2}, \dots, a_{j_n}$ diremo che è stata trasmessa la parola $\mathbf{a}_i = (a_{i_1}, a_{i_2}, \dots, a_{i_n})$ e che è stata ricevuta la parola $\mathbf{a}_j = (a_{j_1}, a_{j_2}, \dots, a_{j_n})$; in queste ipotesi, il numero di componenti omologhe distinte tra \mathbf{a}_i e \mathbf{a}_j prende il nome di *numero di errori* commesso nella trasmissione della parola \mathbf{a}_i .

Figura 13.3: Canale di trasmissione



Per ogni $a_i, a_j \in F$, denotiamo con $P(a_j, a_i)$ la probabilità che immettendo nel canale Σ la lettera a_i si trovi corrispondentemente in uscita la lettera a_j e supponiamo che tale probabilità dipenda soltanto dalla coppia (a_i, a_j) . In queste ipotesi, il canale di trasmissione si dice *senza memoria* e, posto

$$p_{ij} := P(a_j, a_i), \quad (13.1)$$

la matrice

$$P := (p_{ij})$$

si chiama *matrice del canale rispetto all'alfabeto F* . Quando l'alfabeto è chiaro dal contesto si parla semplicemente di *matrice di Σ* . Naturalmente, poiché ogni riga di P contiene le probabilità di tutte le lettere che in uscita possono corrispondere all'immissione della lettera relativa alla riga scelta, la somma degli elementi su ogni riga di P è uguale ad 1, cioè

$$0 \leq p_{ij} \leq 1 \quad e \quad \sum_{j=1}^q p_{ij} = 1;$$

proprietà che si esprimono anche dicendo che P è una *matrice stocastica*.

ESEMPIO 13.2.1. Nell'ambito dei canali di comunicazione senza memoria sono particolarmente importanti i *canali simmetrici*. Un canale di questo tipo è definito dalla proprietà che la probabilità p che una lettera a_i in input sia trasformata in output in una lettera diversa a_j non dipende da a_i e a_j , ma è la stessa per tutte le coppie di lettere distinte; in altre parole, nella matrice del canale P risulta $p_{ij} = p$, per ogni coppia (i, j) , con $i \neq j$. Il numero p si chiama *probabilità d'errore* del canale. La matrice di un canale simmetrico rispetto ad un alfabeto con

m lettere é, dunque, del tipo

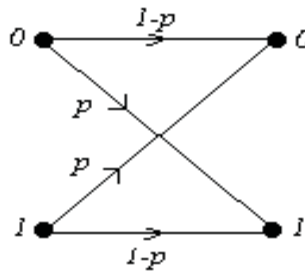
$$P = \begin{bmatrix} 1 - (m - 1)p & p & \dots & p \\ p & 1 - (m - 1)p & \dots & p \\ \vdots & \vdots & \ddots & \vdots \\ p & p & \dots & 1 - (m - 1)p \end{bmatrix} .$$

In particolare, sono molto usati i *canali simmetrici binari*. Questi sono caratterizzati dall'operare con un alfabeto binario, per esempio $\{0, 1\}$, e ciascuna lettera in input ha la stessa probabilità p di essere trasformata nell'altra in output e, quindi, la matrice del canale é data da

$$P = \begin{bmatrix} 1 - p & p \\ p & 1 - p \end{bmatrix} .$$

A volte, per descrivere un canale simmetrico binario con probabilità d'errore p , si usa anche lo schema riportato nella figura che segue. □

Figura 13.4: Canale binario simmetrico



ESERCIZIO 13.2.2. *Si consideri un canale simmetrico binario con probabilità d'errore p . Provare che:*

(a) *la probabilità che nella trasmissione di una parola di lunghezza n si verifichino k errori é*

$$\binom{n}{k} p^k (1 - p)^{n-k} ;$$

(b) *il numero di errori atteso nella trasmissione di una parola di lunghezza n é np .*

Siano, ora, $\mathbf{a}_i = (a_{i_1}, a_{i_2}, \dots, a_{i_n})$ e $\mathbf{a}_j = (a_{j_1}, a_{j_2}, \dots, a_{j_n})$ due parole di lunghezza n sull'alfabeto F e denotiamo con $P(\mathbf{a}_j, \mathbf{a}_i)$ la probabilità che, immettendo nel canale Σ la parola \mathbf{a}_i , si trovi corrispondentemente in uscita la parola \mathbf{a}_j ; osserviamo che, in forza della (13.1), risulta

$$P(\mathbf{a}_j, \mathbf{a}_i) = p(a_{j_1}, a_{i_1})p(a_{j_2}, a_{i_2}) \cdots p(a_{j_n}, a_{i_n}) . \tag{13.2}$$

Ordiniamo linearmente l'insieme delle parole su F di lunghezza n ,

$$F^n = \{\mathbf{a}_1, \mathbf{a}_2, \dots\},$$

e posto

$$p_{ij}^{(n)} := P(\mathbf{a}_j, \mathbf{a}_i),$$

la matrice stocastica

$$P^{(n)} := \left(p_{ij}^{(n)} \right) \quad (13.3)$$

prende il nome di *n-esima matrice del canale* Σ (rispetto all'alfabeto F). Non dovrebbe essere difficile convincersi che $P^{(n)}$ può riguardarsi come la matrice di Σ rispetto ad F^n , considerato come alfabeto.

ESEMPIO 13.2.3. Sia Σ un canale binario simmetrico con probabilità d'errore p (cfr. esempio 13.2.1) Allora, se

$$\{(0, 0), (0, 1), (1, 0), (1, 1)\}$$

è l'insieme linearmente ordinato delle parole binarie di lunghezza 2, la seconda matrice di Σ è data da

$$P^{(2)} = \begin{bmatrix} (1-p)^2 & p(1-p) & p(1-p) & p^2 \\ p(1-p) & (1-p)^2 & p^2 & p(1-p) \\ p(1-p) & p^2 & (1-p)^2 & p(1-p) \\ p^2 & p(1-p) & p(1-p) & (1-p)^2 \end{bmatrix} = \begin{bmatrix} (1-p)P & pP \\ pP & (1-p)P \end{bmatrix} = P \otimes P.$$

□

ESERCIZIO 13.2.4. Sia $P^{(n)}$ la n -esima matrice di un canale simmetrico binario. Provare che ogni riga (risp. colonna) di $P^{(n)}$ è una permutazione della prima.

13.3 La trasmissione dell'informazione

La crescente esigenza dell'uomo di rendere sempre più veloce la trasmissione delle informazioni è alla base del grande sviluppo che la teoria dei codici ha avuto negli ultimi cinquanta anni. Naturalmente, una delle caratteristiche essenziali richieste ad un sistema di comunicazione è l'affidabilità: *un messaggio trasmesso deve poter arrivare senza alterazioni al destinatario*. Purtroppo, non esistendo sistemi di comunicazione perfetti, la probabilità che nel corso di una trasmissione si verifichino degli errori non può mai ridursi a zero. Gli effetti negativi dovuti a questa situazione possono essere ridotti sostanzialmente in due modi:

1. *Intervenire direttamente sui canali di trasmissione mediante l'utilizzo di nuove tecnologie, modificando in parte quelli in uso o costruendone dei nuovi, allo scopo di ridurre la possibilità*

che si verificano errori nei canali stessi (si pensi per esempio all'alta affidabilità dei sistemi che fanno uso di fibre ottiche o di compact disc).

2. Adottare codici che, tenendo conto del grado di affidabilità del sistema di comunicazione in uso, permettano di scoprire e correggere automaticamente eventuali errori.

Il primo approccio, vicino all'ingegneria e alla fisica, necessita ovviamente di disponibilità finanziarie non indifferenti. Già a livello di ricerca i progetti hanno costi molto alti e non è detto che eventuali risultati positivi siano convenienti per immediate applicazioni. Basti pensare che le scelte legate a modifiche, anche parziali, delle architetture dei sistemi di comunicazione a larga diffusione, al di là dei problemi economici e tecnologici, possono avere notevoli implicazioni anche di carattere politico.

Il secondo approccio, invece, di tipo matematico-informatico, pur avendo bisogno nella pratica di supporti tecnologici, non presenta gli inconvenienti del primo ed è molto meno costoso. In altre parole, *correggere gli errori è più conveniente che prevenirli* intervenendo direttamente sui sistemi di comunicazione. Il primo a rendersi conto di ciò è stato il matematico americano *Richard Hamming*, che nel 1948 fondò la *teoria della correzione degli errori* con la scoperta di una classe di codici correttori binari che ora portano il suo nome. Tali codici, che introdurremo più avanti, si rivelarono particolarmente adatti ai canali simmetrici binari (*cf.* 13.2.1) e furono subito molto utilizzati, specialmente nel caso di canali che facevano uso di onde elettromagnetiche nell'etere e di impulsi elettrici nei fili. Successivamente essi hanno dato origine alla *teoria dei codici lineari* che, al di là dell'importanza per le applicazioni, ha ormai assunto un ruolo di rilievo nell'ambito delle teorie combinatorie e in particolare delle geometrie finite.

Per rendere più chiaro il problema della correzione automatica degli errori, descriviamo sinteticamente le trasformazioni cui viene sottoposto un messaggio immesso in un sistema di comunicazione che, molto schematicamente, supporremo composto da:

- una *stazione emittente* \mathbf{E} ;
- un *canale di trasmissione senza memoria* Σ , con *codice* C e *matrice* P ;
- una *stazione ricevente* \mathbf{R} .

La stazione \mathbf{E} può inviare ad \mathbf{R} messaggi scelti in un insieme prefissato \mathcal{M} (*insieme dei messaggi*); i messaggi in \mathcal{M} possono essere identificati con parole del codice C mediante una funzione iniettiva γ tra \mathcal{M} e C (*funzione di codifica*).

Supponiamo che l'emittente \mathbf{E} debba inviare un messaggio M , scelto in \mathcal{M} , alla stazione ricevente \mathbf{R} . Prima di essere trasmesso, M deve essere messo in una forma accettabile per il canale Σ ; deve essere cioè trasformato in una parola \mathbf{x} del codice C . Questa operazione, detta *codifica*, si realizza mediante un algoritmo che calcola automaticamente il valore su M della funzione di codifica γ ; nel nostro caso $\gamma(M) = \mathbf{x}$. A questo punto, la parola \mathbf{x} viene immessa nel canale Σ e viene ricevuta in uscita una parola \mathbf{y} che, a causa di disturbi del canale, può essere diversa da \mathbf{x} .

La parola \mathbf{y} viene finalmente tradotta (*decodificata*) in un messaggio $M' \in \mathcal{M}$ mediante un *algoritmo di decodifica* che opera nel seguente modo:

- se $\mathbf{y} \in C$, pone $M' = \gamma^{-1}(\mathbf{y})$;
- se $\mathbf{y} \notin C$, individua la parola \mathbf{z} di C in qualche modo piú simile a \mathbf{y} e pone $M' = \gamma^{-1}(\mathbf{z})$.

Per una buona trasmissione, dunque, nell'ipotesi $\mathbf{y} \neq \mathbf{x}$, non deve accadere che $\mathbf{y} \in C$, altrimenti non vi é alcun modo di capire che vi sono stati degli errori e il destinatario riceve inevitabilmente un messaggio sbagliato. Inoltre, alla fine del processo di trasmissione deve ottenersi $\mathbf{z} = \mathbf{x}$, cioè \mathbf{x} deve potersi riconoscere a partire da \mathbf{y} . Queste condizioni, che riterremo le *condizioni di affidabilità* del sistema, nell'ipotesi che C abbia lunghezza n , sono evidentemente soddisfatte se sono verificate le seguenti proprietà:

- $P(\mathbf{a}, \mathbf{a}) \neq 0$, per ogni parola $\mathbf{a} \in C$;
- $P(\mathbf{b}, \mathbf{a}) = 0$ per ogni \mathbf{a}, \mathbf{b} parole distinte di C ;
- se $\mathbf{a} \in C$ e $P(\mathbf{z}, \mathbf{a}) \neq 0$, allora $P(\mathbf{z}, \mathbf{b}) = 0$ per ogni parola \mathbf{b} di C diversa da \mathbf{a} .

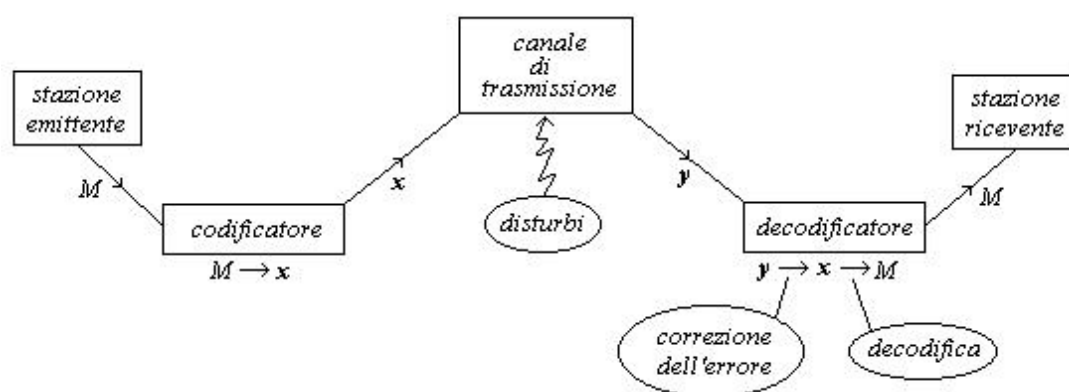
Sotto queste ipotesi, infatti, se una parola \mathbf{y} non appartiene a C esiste al piú una parola \mathbf{x} di C tale che $P(\mathbf{y}, \mathbf{x}) \neq 0$ e di conseguenza, almeno teoricamente, é possibile descrivere un algoritmo per la correzione degli errori e, quindi, per la decodifica. A tal fine, basta osservare che gli insiemi

$$B_{\mathbf{a}} = \{z \in F^n : P(z, \mathbf{a}) \neq 0\},$$

al variare di $\mathbf{a} \in C$, sono a due a due disgiunti e, quindi, se si riceve $\mathbf{y} \in B_{\mathbf{x}}$, con $\mathbf{x} \in C$, si può dedurre che \mathbf{x} é la parola di C inizialmente trasmessa.

Nel seguito diremo che un sistema di comunicazione é *affidabile* se sono verificate le precedenti tre condizioni. In tali ipotesi, il processo della trasmissione di un messaggio può rappresentarsi con la figura 13.5.

Figura 13.5: Sistema di comunicazione affidabile



13.4 Distanza di Hamming e correzione degli errori

Iniziamo ad esporre i primi elementi della teoria dei codici correttori che, come abbiamo visto, si può considerare il supporto teorico di base dei problemi descritti nel precedente paragrafo. A tale scopo introduciamo un concetto di distanza fra parole dovuto a *R. Hamming*. Ciò permetterà anche di illustrare e giustificare un principio di decodifica, detto *nearest neighbour decoding*, che è alla base della teoria che intendiamo esporre. Tale principio si presta ad essere usato con successo nei sistemi di comunicazione che utilizzano canali di trasmissione simmetrici (*cf.* 13.2.1) e codici lineari, dei quali parleremo più avanti.

DEFINIZIONE 13.4.1. Se F^n , $n > 0$, è l'insieme di tutte le parole di lunghezza n su un alfabeto F e se \mathbf{x}, \mathbf{y} sono due tali parole, si definisce *distanza di Hamming* tra \mathbf{x} e \mathbf{y} , e si denota con $d(\mathbf{x}, \mathbf{y})$, il numero di posizioni in cui $\mathbf{x} = (x_1, x_2, \dots, x_n)$ e $\mathbf{y} = (y_1, y_2, \dots, y_n)$ presentano lettere differenti, cioè

$$d(\mathbf{x}, \mathbf{y}) = |\{i : x_i \neq y_i, 1 \leq i \leq n\}|.$$

Come subito si verifica, la funzione d è una metrica su F^n , detta *metrica di Hamming*. \square

OSSERVAZIONE 13.4.2. Ricordiamo che l'essere la distanza di Hamming una metrica su F^n significa che valgono le seguenti proprietà:

- $d(\mathbf{x}, \mathbf{y}) \geq 0$,
- $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$,
- $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$,
- $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ (*disuguaglianza triangolare*),

per ogni $\mathbf{x}, \mathbf{y}, \mathbf{z} \in F^n$. \square

Nel seguito supporremo sempre che F^n sia dotato della metrica di Hamming e ogni codice C su F sarà costantemente considerato come sottospazio metrico di F^n .

Seguendo ora lo schema della figura ??, supponiamo che una parola \mathbf{x} di un fissato codice C venga trasmessa da una emittente e modificata in una parola $\mathbf{y} \neq \mathbf{x}$ dal canale di trasmissione. Se \mathbf{y} è ancora una parola di C , è chiaro che non esiste alcuna possibilità di scoprire lo scambio di parole. Se invece \mathbf{y} non appartiene al codice C , allora è evidente che c'è stato un errore e di conseguenza si pone il problema di correggerlo, cioè di risalire dalla parola ricevuta \mathbf{y} a quella \mathbf{x} effettivamente trasmessa.

Ad esempio, supponiamo di avere a disposizione un canale di trasmissione T per il quale sia molto alta la probabilità che il massimo numero di lettere di una parola che si possono modificare nel corso di una trasmissione sia più piccolo della metà della minima distanza fra due qualsiasi parole distinte del codice C . In queste ipotesi, se la parola ricevuta \mathbf{y} non appartiene a C , esiste

generalmente un'unica parola $z \in C$ a distanza minima da y^1 ; cosí il decodificatore sostituisce automaticamente y con z e la probabilit  che sia $z = x$   estremamente alta. Il principio appena esposto, secondo il quale si decodifica la parola del codice a distanza minima da quella ricevuta, prende il nome di *nearest neighbour decoding*.

Poich  nella realt  non esistono canali di trasmissione immuni da disturbi, quanto finora detto suggerisce di non scegliere mai il codice C uguale ad F^n . In altre parole, un buon codice deve essere un sottoinsieme proprio di F^n con la propriet  che ogni sua parola, al fine di una buona decodifica, oltre a contenere il minimo numero di lettere necessarie per la codifica del messaggio associato, contenga delle ulteriori lettere, dette di *controllo*. Queste permettono di ricostruire la parola trasmessa nel caso non si siano verificati troppi errori durante la trasmissione.

Chiariamo subito il discorso con un esempio.

ESEMPIO 13.4.3. Supponiamo di disporre di un ipotetico canale di trasmissione con la propriet  di modificare al pi  una componente per ogni parola binaria di lunghezza non superiore a cinque; per esempio un canale simmetrico con probabilit  d'errore $p = \frac{1}{5}$ (*cfr.* 13.2.2). Supponiamo, inoltre, di avere necessit  di trasmettere dei messaggi scelti fra i seguenti quattro: NORD, SUD, EST, OVEST. In questo caso il modo pi  naturale e veloce per trasmettere, ma anche il pi  ingenuo,   quello di codificarli usando il codice binario

$$C_1 = \{0, 1\}^2 = \{(0, 0), (0, 1), (1, 0), (1, 1)\};$$

per esempio ponendo

$$\text{NORD} \equiv (0, 0), \text{SUD} \equiv (0, 1), \text{EST} \equiv (1, 0), \text{OVEST} \equiv (1, 1).$$

In questo modo, se una parola viene modificata, per esempio (1, 0) in (1, 1), il destinatario riceve il messaggio OVEST invece di EST, non avendo il decodificatore alcun elemento per scoprire che la parola ricevuta   diversa da quella trasmessa.

Proviamo ora ad usare di un codice C_2 di lunghezza tre, per esempio

$$C_2 = \{(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)\}$$

e poniamo

$$\text{NORD} \equiv (0, 0, 0), \text{SUD} \equiv (0, 1, 1), \text{EST} \equiv (1, 0, 1), \text{OVEST} \equiv (1, 1, 0).$$

In questo caso, se su una parola si commette un solo errore, questo pu  essere scoperto ma non corretto. Per esempio, se (0, 0, 0) si trasforma in (1, 0, 0), il decodificatore si trova di fronte ad una parola non appartenente al codice C_2 e riconosce l'errore ma, contenendo C_2 pi  di una parola che pu  trasformarsi in (1, 0, 0) col cambio di una sola componente, non   in grado di risalire alla parola effettivamente trasmessa.

A questo punto   facile rendersi conto che, se vogliamo avere la possibilit  di scoprire e correggere almeno un errore, abbiamo bisogno di un codice di lunghezza almeno cinque, per esempio

$$C_3 = \{(0, 0, 0, 0, 0) \equiv \text{NORD}, (0, 1, 1, 0, 1) \equiv \text{SUD},$$

¹Si osservi che questa propriet    falsa se il numero di lettere di x modificate nel corso della trasmissione   maggiore della met  della minima distanza fra due qualsiasi parole distinte del codice C .

$$(1, 0, 1, 1, 0) \equiv \text{EST}, \quad (1, 1, 0, 1, 1) \equiv \text{OVEST}\}.$$

E' infatti immediato provare che, se una parola $\mathbf{x} \in C_3$ si trasforma in una \mathbf{y} mediante lo scambio di una sola componente, allora \mathbf{x} é l'unica parola del codice a distanza 1 da \mathbf{y} , mentre tutte le altre hanno distanza almeno due.

Osserviamo che nei codici C_2 e C_3 le prime due lettere delle parole individuano completamente il messaggio mentre le rimanenti sono le cosiddette *lettere di controllo*. Inoltre, nel passaggio dal codice C_1 al codice C_3 si arriva ad un sistema di comunicazione affidabile. Il prezzo pagato per l'affidabilità é chiaramente la minore velocità della trasmissione; infatti, essendo le parole di C_3 piú lunghe di quelle di C_1 , la trasmissione stessa sará necessariamente piú lenta.

□

Nel seguito, tranne avviso contrario, riterremo fissato un (n, M) -codice C su un alfabeto F con q lettere. Inoltre, spesso identificheremo C con la matrice su F le cui righe sono le parole di C , preventivamente ordinate. Una tale matrice ha M righe ed n colonne e si dice *associata* a C . Due matrici associate ad uno stesso codice differiscono, quindi, per una permutazione delle righe.

DEFINIZIONE 13.4.4. Diciamo che due (n, M) - codici sullo stesso alfabeto sono *equivalenti* se due matrici ad essi rispettivamente associate possono ottenersi l'una dall'altra mediante una successione finita di operazioni dei seguenti tipi:

(A) *scambio di due colonne (questa operazione equivale a scambiare tra loro in ogni parola del codice le lettere che si trovano in due posizioni fissate);*

(B) *applicazione di una permutazione dell'alfabeto F alle lettere che si trovano in una fissata colonna.* □

ESEMPIO 13.4.5. Si consideri il codice ternario

$$C_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 & 2 \\ 2 & 1 & 0 & 2 & 0 \end{bmatrix}$$

sull'alfabeto $F = \{0, 1, 2\}$. Il codice

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 2 & 0 \\ 1 & 0 & 0 & 2 & 2 \\ 1 & 2 & 2 & 0 & 0 \end{bmatrix}$$

é equivalente a C_1 perché si ottiene da questo scambiando la prima colonna con la seconda e la terza con la quarta. Ancora, il codice

$$\begin{bmatrix} 1 & 0 & 2 & 0 & 1 \\ 1 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 2 \\ 2 & 1 & 1 & 2 & 0 \end{bmatrix}$$

é equivalente a C_1 perché si ottiene da questo applicando la permutazione ciclica $(0, 1, 2)$ alle lettere della terza colonna. \square

ESERCIZIO 13.4.6. *Provare che i seguenti $(5, 4)$ -codici sull'alfabeto $F = \{0, 1, 2\}$*

$$C_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 0 & 2 \\ 2 & 1 & 0 & 2 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 2 & 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 & 2 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 1 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 2 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 2 & 1 & 0 & 2 \end{bmatrix}$$

sono equivalenti.

Lo studio dei codici verrà fatto a meno di equivalenze. Fra le proprietà comuni a due codici equivalenti sono particolarmente utili quelle espresse dalle seguenti due proposizioni, le cui semplici dimostrazioni vengono lasciate per esercizio al Lettore.

PROPOSIZIONE 13.4.7. *Siano C_1 e C_2 due codici equivalenti. Allora, per ogni intero positivo t , il numero di coppie di parole di C_1 a distanza t é uguale al corrispondente numero in C_2 .*

PROPOSIZIONE 13.4.8. *Siano C un codice di lunghezza n su un alfabeto F e x una qualsiasi parola su F di lunghezza n . Allora esiste un codice su F equivalente a C e contenente la parola x .*

DEFINIZIONE 13.4.9. Si chiama *distanza minima* di un (n, M) -codice C l'intero $d(C)$ dato dalla piú piccola distanza fra due parole distinte di C , cioè

$$d(C) := \min\{d(x, y) : x, y \in C, x \neq y\}.$$

Se non vi é possibilità di equivoci scriveremo d in luogo di $d(C)$. Un (n, M) -codice di distanza minima d si dice anche un (n, M, d) -codice e gli interi n, M, d si dicono *parametri* del codice. \square

ESERCIZIO 13.4.10. *Provare che codici equivalenti hanno gli stessi parametri.*

DEFINIZIONE 13.4.11. Diciamo che il codice C é k -*sistematico* o semplicemente *sistematico*, se in una delle sue matrici associate esistono k colonne di posto i_1, i_2, \dots, i_k tali che, per ogni k -pla $(\alpha_1, \alpha_2, \dots, \alpha_k)$ di lettere di F , esiste un'unica parola (a_1, a_2, \dots, a_n) di C per cui risulta

$$a_{i_1} = \alpha_1, a_{i_2} = \alpha_2, \dots, a_{i_k} = \alpha_k.$$

Gli interi i_1, i_2, \dots, i_k si chiamano anche *posti di informazione*. In queste ipotesi l'intero $n - k$ prende il nome di *ridondanza* di C e si dicono *ridondanti* o *di controllo* le lettere delle parole di C che occupano posizioni diverse da i_1, i_2, \dots, i_k . \square

ESERCIZIO 13.4.12. *Provare che il codice C_2 dell'esempio 13.4.3 é 2-sistematico e che il codice ASCII esteso é 7-sistematico.*

Per ogni parola $x \in F^n$ e per ogni intero positivo r , consideriamo la *sfera di centro x e raggio r* (*sfera di Hamming*), cioè l'insieme

$$S(x, r) := \{y \in F^n : d(x, y) \leq r\}.$$

L'insieme

$$\bar{S}(\mathbf{x}, r) := \{\mathbf{y} \in F^n : d(\mathbf{x}, \mathbf{y}) = r\}$$

prende il nome di *superficie sferica di centro \mathbf{x} e raggio r* e risulta

$$\begin{cases} \bar{S}(\mathbf{x}, r) \cap \bar{S}(\mathbf{x}, r') = \emptyset, & r \neq r', \quad r, r' \leq n \\ S(\mathbf{x}, r) = \bigcup_{0 \leq s \leq r} \bar{S}(\mathbf{x}, s) \end{cases} \quad (13.4)$$

PROPOSIZIONE 13.4.13. *Una sfera di raggio r , $0 \leq r \leq n$, in F^n contiene esattamente*

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{r}(q-1)^r$$

parole.

DIMOSTRAZIONE. Le parole a distanza s da una fissata parola \mathbf{x} si ottengono modificando arbitrariamente s lettere di \mathbf{x} e quindi sono esattamente

$$\binom{n}{s}(q-1)^s.$$

Dalle (13.4) segue allora l'asserto. □

DEFINIZIONE 13.4.14. Si dice che un (n, M) -codice C *scopre k errori*, ove k é un intero positivo, se la sfera $S(\mathbf{x}, k)$ ha in comune con C la sola parola \mathbf{x} , per ogni $\mathbf{x} \in C$. Si dice poi che C *corregge k errori*, se scopre k errori e due qualsiasi sfere di raggio k con centri in parole distinte di C sono ad intersezione vuota. □

Queste definizioni sono del tutto naturali se si pensa alle osservazioni fatte precedentemente, specialmente nell'esempio 13.4.3, e la proposizione che segue é una loro immediata conseguenza.

PROPOSIZIONE 13.4.15. *Un (n, M) -codice C scopre k errori se, e soltanto se, risulta*

$$d \geq k + 1$$

e corregge h errori se, e soltanto se, risulta

$$d \geq 2h + 1.$$

Di solito il massimo numero di errori che un codice puó correggere viene denotato con e e, in questo caso, il codice si dice *e -correttore*. La prop.13.4.15 assicura che, se d é la minima distanza di C , allora é

$$d = 2e + 1 \quad \text{o} \quad d = 2e + 2, \quad (13.5)$$

a seconda che d sia dispari o pari, rispettivamente.

PROPOSIZIONE 13.4.16. (*Disuguaglianza di Hamming*) Per ogni (n, M) -codice C che sia e -correttore, risulta

$$M \left[\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{e}(q-1)^e \right] \leq q^n. \quad (13.6)$$

DIMOSTRAZIONE. La prop.13.4.15 assicura che sfere di centro due parole distinte di C e raggio e sono fra loro disgiunte. Dalla prop.13.4.13 segue allora l'asserto. \square

DEFINIZIONE 13.4.17. I codici i cui parametri verificano l'uguaglianza nella (13.6) si dicono *perfetti*. \square

ESEMPIO 13.4.18. Il codice $C = F^n$, i codici contenenti una sola parola e i codici di ripetizione binari di lunghezza dispari, definiti nel seguito dalla (13.8), sono perfetti; essi vengono detti *codici perfetti banali*. \square

ESERCIZIO 13.4.19. Verificare che un (n, M, d) -codice C che sia e -correttore è perfetto se, e soltanto se, le sfere di raggio e e centro le parole di C formano una partizione di F^n . Dedurne che, se C è perfetto, allora d deve essere dispari. \square

OSSERVAZIONE 13.4.20. Notiamo esplicitamente che la proprietà di un codice di essere perfetto dipende esclusivamente dai suoi parametri. Questo significa che, se un (n, M, d) -codice C è perfetto, ogni codice con gli stessi parametri di C è anch'esso perfetto. \square

Terminiamo il paragrafo con un esempio di codice perfetto non banale.

ESEMPIO 13.4.21. Sia

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \\ \mathbf{a}_7 \end{matrix}$$

una matrice d'incidenza del piano di Fano $PG(2, 2)$ e indichiamo con $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7$ le sue righe. Aggiungiamo ad A le righe $\mathbf{0} = (0, 0, 0, 0, 0, 0, 0)$, $\mathbf{1} = (1, 1, 1, 1, 1, 1, 1)$ e le righe \mathbf{b}_i

che si ottengono dalle \mathbf{a}_i scambiando tra loro 1 e 0. Otteniamo cosí la matrice

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \begin{matrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \mathbf{a}_3 \\ \mathbf{a}_4 \\ \mathbf{a}_5 \\ \mathbf{a}_6 \\ \mathbf{a}_7 \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \\ \mathbf{b}_6 \\ \mathbf{b}_7 \end{matrix}$$

E' facile verificare che le righe di H costituiscono le parole di un $(7, 16, 3)$ -codice. Piú precisamente, per $i \neq j$, si ha: $d(\mathbf{0}, \mathbf{a}_i) = 3$, $d(\mathbf{0}, \mathbf{b}_i) = 4$, $d(\mathbf{0}, \mathbf{1}) = 7$, $d(\mathbf{a}_i, \mathbf{1}) = 4$, $d(\mathbf{b}_i, \mathbf{1}) = 3$, $d(\mathbf{a}_i, \mathbf{a}_j) = 4$, $d(\mathbf{a}_i, \mathbf{b}_i) = 7$, $d(\mathbf{a}_i, \mathbf{b}_j) = 3$, $d(\mathbf{b}_i, \mathbf{b}_j) = 4$. Il codice appena definito si chiama *codice binario associato al piano di Fano* ed é un semplice esercizio verificare che é perfetto. E' questo un esempio di codice costruito a partire da un disegno.

Notiamo che se 0 e 1 si pensano come gli elementi del campo di Galois F_2 d'ordine 2, allora le parole del codice formano un sottospazio vettoriale di dimensione 4 dello spazio vettoriale F_2^7 .

□

13.5 Il problema fondamentale della teoria dei codici

Quanto esposto nei precedenti paragrafi giustifica il fatto che ad un buon codice si richiede che abbia:

- lunghezza n *abbastanza piccola* per permettere una trasmissione veloce delle sue parole;
- un numero M di parole *abbastanza grande* per codificare una buona quantitá di messaggi;
- distanza minima *abbastanza grande* per correggere il maggior numero possibile di errori (*cf*.prop.13.4.15).

Chiaramente queste richieste sono tra loro contrastanti e di conseguenza non é possibile ottimizzare uno dei parametri senza avere preventivamente fissato gli altri due. Quest'ultimo tipo di problema é quello che comunemente va sotto il nome di *problema fondamentale della teoria dei codici*. Per esempio, fissati n e d , se denotiamo con $A_q(n, d)$ il piú grande valore di M per cui

esiste un (n, M, d) -codice q -ario, abbiamo

$$\begin{cases} A_q(n, 1) = |F^n| = q^n \\ A_q(n, n) = |F| = q \end{cases} . \quad (13.7)$$

La condizione $d = 1$, infatti, impone soltanto che le parole di C siano tutte distinte fra loro e quindi il massimo valore di M si ottiene per $C = F^n$, cioè la prima delle (13.7). Se invece abbiamo $d = n$, allora le lettere che figurano in una fissata posizione nelle parole di C devono essere a due a due distinte e quindi $A_q(n, n) \leq q$. D'altra parte il codice

$$C(F, n) = \{(a, a, \dots, a) : a \in F\} \quad (13.8)$$

contiene esattamente q parole e gode della proprietà richiesta; abbiamo così la seconda delle (13.7). Il codice $C(F, n)$ si chiama *codice di ripetizione q -ario di lunghezza n* o anche (n, q, n) -*codice di ripetizione su F* .

ESERCIZIO 13.5.1. *Posto $d = 2e + 1$ o $d = 2e + 2$, provare che*

$$A_q(n, d) \left[\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \dots + \binom{n}{e}(q-1)^e \right] \leq q^n. \quad (13.9)$$

Provare, inoltre, che l'uguaglianza può aversi solo nel caso d dispari.

ESERCIZIO 13.5.2. *(Disuguaglianza di Gilbert-Varshamov) Provare che*

$$A_q(n, d) \left[\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \dots + \binom{n}{d-1}(q-1)^{d-1} \right] \geq q^n. \quad (13.10)$$