Rendiconti Sem. Mat. Univ. Pol. Torino
Vol. 80, 2 (2022), 83 – 97

P. Morando - S.M.C. Pagani

ERRORI DI TRASMISSIONE: UN BIT PER TROVARLI, TRE BIT PER INCATENARLI (CORREGGENDOLI)

Abstract. Nonostante la teoria dei codici coniughi geniali intuizioni matematiche con utilissime applicazioni pratiche, essa rimane praticamente sconosciuta al grande pubblico. Per tale ragione in questo articolo proponiamo una serie di attività che permettono di avvicinare gli studenti delle scuole secondarie ad alcuni temi di questa affascinante disciplina. In particolare, le idee di codice rilevatore e di codice correttore proposte da Hamming vengono presentate mescolando elementi di gamification ad un apprendimento di tipo esperienziale.

1. Introduzione.

La teoria dei codici è un ramo della teoria dell'informazione che studia la trasmissione dei dati e che ha come obiettivo la costruzione di modelli di comunicazione nei quali gli eventuali, ed inevitabili, errori di trasmissione possano essere individuati e corretti. Pur essendo a tutti gli effetti un ramo della matematica (già dal titolo del lavoro in cui compare per la prima volta: *A mathematical theory of communication* [4]), essa viene spesso percepita come legata piú all'informatica o comunque alle applicazioni pratiche. In realtà si tratta di una disciplina ricca di interessanti aspetti teorici e di spunti matematici di grande originalità che, se opportunamente presentati, possono risultare estremamente interessanti sia per gli studenti delle scuole secondarie che per il grande pubblico.

In questo articolo presentiamo alcune attività laboratoriali che permettono di avvicinare gli studenti di scuola secondaria alla teoria dei codici, stimolando la partecipazione ed il lavoro di gruppo. Posti di fronte a problemi semplici da presentare ma assolutamente non banali da risolvere, gli studenti si metteranno in gioco per trovare soluzioni, la cui efficacia verrà poi verificata sul campo. In questo modo i ragazzi avranno modo di sperimentare un aspetto estremamente importante nell'apprendimento della matematica, ovvero quello della scoperta. Rispetto ad una lezione di tipo tradizionale in cui l'insegnante presenta le medesime idee, le attività proposte stimolano la curiosità e l'interesse degli studenti, permettendo loro di apprezzare appieno la genialità, la semplicità e l'eleganza di alcune intuizioni matematiche alla base della teoria dei codici.

Il percorso didattico si articola in tre incontri. Nel primo, dopo una breve introduzione generale sui codici binari e sul problema degli errori di trasmissione, gli studenti vengono coinvolti in un gioco a squadre che, attraverso la ricerca di una strategia efficace per l'individuazione di eventuali errori di trasmissione, permette di introdurre in maniera naturale l'idea di bit di parità. Nel secondo incontro questo concetto viene ripreso e utilizzato per affrontare, lavorando a gruppi, il problema della correzione di eventuali errori. Attraverso le proposte dei vari gruppi, si arriva quindi a introdurre il

codice di Hamming (7,4), la cui efficacia viene sperimentata attraverso la proposta di un'attività ludico-didattica. Infine, il terzo incontro mette a frutto quanto già imparato nei due incontri precedenti per cercare di automatizzare il processo di identificazione e di correzione degli errori tramite il codice di Hamming. I risultati ottenuti permettono di estendere i risultati ottenuti per stringhe di sette caratteri a stringhe di caratteri binari di lunghezza qualsiasi, e di far toccare con mano agli studenti e la genialità dell'idea di Hamming anche in termini di economicità ed efficienza.

L'articolo è strutturato come segue: nella sezione 2 si richiama la teoria soggiacente alla trattazione; nella sezione 3 le attività didattiche vengono presentate in modo dettagliato; la sezione 4 conclude il lavoro.

2. Background teorico

2.1. Un modello di comunicazione

La teoria dei codici si occupa della realizzazione di sistemi che consentano di riprodurre, in un certo punto dello spazio, un messaggio inviato da un altro punto. Se il canale di comunicazione, cioè il mezzo utilizzato per trasmettere il messaggio, è soggetto a rumore, il contenuto della comunicazione potrebbe risultare alterato. Chi riceve il messaggio deve essere quindi in grado di individuare la presenza di errori e di correggerli. La teoria dei codici si inserisce nella piú ampia teoria della comunicazione, nata ufficialmente nel 1948 con la pubblicazione di [4] da parte di Claude Shannon. Oltre alle naturali esigenze pratiche, la ricerca in tale disciplina ha portato a felici collaborazioni con aree della matematica pura, tra le quali la geometria combinatoria (si veda ad esempio [1]).

Possiamo modellizzare la trasmissione secondo lo schema rappresentato in Figura 1. Il messaggio m che il mittente vuole inviare viene codificato in un vettore $\mathbf{c}$, detto parola di codice, che è ciò che viene effettivamente trasmesso. Le componenti del vettore sono simboli di un alfabeto predefinito; nella nostra trattazione useremo l'alfabeto binario. Ad esempio, per trasmettere il messaggio 'sí', come prima cosa è necessario tradurlo in una parola di codice (come '1' o '1011'). Tale parola viene poi inviata al ricevitore attraverso un canale, che supponiamo rumoroso. In questo processo alcune componenti della parola possono essere alterate, per cui la parola ricevuta $\mathbf{c}'$ può non coincidere con $\mathbf{c}$ ed è quindi importante capire quando il messaggio ricevuto non corrisponde al messaggio originale, in modo da poterlo scartare o correggere. Per far questo, una possibile strategia è quella di codificare il messaggio in una parola piú lunga di quanto strettamente necessario, aggiungendo un certo numero di caratteri utili per rilevare gli eventuali errori di trasmissione. In questo modo il ricevitore decodifica la parola ricevuta, identificando e correggendo eventuali errori, ed ottiene (sperabilmente) il messaggio originale.

2.2. Concetti basilari della teoria dei codici

In questo articolo ci occuperemo di codici *a blocchi*, nei quali i dati da trasmettere vengono codificati in stringhe aventi tutte lo stesso numero di caratteri. Tali stringhe

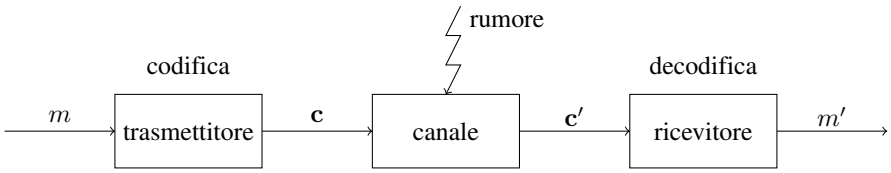

Figura 1: Un modello di comunicazione tramite un canale rumoroso.

saranno, nella nostra trattazione, vettori di n componenti, detti *parole di codice*. Ogni componente, detta bit, contiene un simbolo preso da un alfabeto finito; nel nostro caso si tratterà dell'alfabeto binario, contenente solo i simboli 0 ed 1. In informatica i codici binari sono utilizzati dalla quasi totalità degli elaboratori elettronici, in quanto le caratteristiche fisiche dei circuiti digitali rendono molto conveniente la gestione di due soli valori, rappresentati fisicamente da due diversi livelli di tensione elettrica. Un *codice* è un insieme di parole di codice della stessa lunghezza e definite sul medesimo alfabeto.

La *distanza di Hamming* tra due parole di codice $\mathbf{x}$ ed $\mathbf{y}$ è il numero di posizioni in cui esse differiscono:

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

La distanza di Hamming di un codice C è il minimo delle possibili distanze tra due parole distinte di C:

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

Un codice si dice *t-correttore* se è in grado di correggere al piú t errori occorsi durante la trasmissione.

TEOREMA 1. *Sia C un codice t-correttore con distanza minima d. Allora*

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Inoltre, C può rilevare almeno $d-1$ errori in ogni parola.

Per una dimostrazione si rimanda a [2].

Come detto in precedenza, una possibile idea per rilevare, ed eventualmente correggere, un errore all'interno di una parola di codice è quella di codificare il messaggio in una parola piú lunga di quanto strettamente necessario, aggiungendo un certo numero di bit utili per rilevare gli eventuali errori di trasmissione. Questi bit aggiuntivi prendono il nome di *bit di ridondanza*. I bit di ridondanza si suddividono in bit rilevatori (in grado di rilevare la presenza o meno di errori, ma non di individuare la loro posizione) e bit correttori (in grado di rilevare una o piú posizioni errate e quindi di correggerle per semplice inversione del valore del bit).

Un esempio molto semplice di utilizzo di bit correttori è il seguente: ogni bit da trasmettere viene triplicato, quindi se la parola originale è '101', la parola trasmessa

diventa ‘111000111’. In questo modo il ricevente può facilmente identificare eventuali errori (che si verificano se i tre bit non sono identici) e correggerli in base a un criterio di maggioranza. La correzione dell’errore funziona come segue: alla ricezione di 000, 001, 010 o 100 il bit di dati decodificato diventa 0, mentre alla ricezione di 111, 110, 101 o 011 il bit di dati decodificato diventa 1.

Nel seguito ci occuperemo di come rilevare, ed eventualmente correggere, un singolo errore all’interno di una parola di codice. Nonostante si tratti di una drastica semplificazione rispetto alla complessità del problema reale, questo modello fornisce materiale abbondantemente sufficiente per introdurre alcune idee alla base della teoria dei codici.

2.3. Il bit di parità

Lavorando sull’alfabeto binario, una possibile strategia per verificare la presenza di eventuali errori di trasmissione è quella di contare il numero di ‘1’ presenti nella parola di codice. Aggiungendo una componente a ciascun vettore si ha a disposizione un bit ridondante, cioè che non trasmette parte del messaggio, il cui compito sarà quello di rendere pari il numero di ‘1’ presenti: assumerà quindi valore ‘0’ se il numero di ‘1’ è già pari e valore ‘1’ altrimenti. Questo bit viene chiamato bit di parità di Hamming.

In concreto, se la parola di codice ricevuta, contenente il bit di parità, ha un numero dispari di ‘1’, si può essere certi che ci sia stato (almeno) un errore di trasmissione. Si noti che il bit di parità permette di scoprire la presenza di un numero dispari di errori (e in particolare di un singolo errore), mentre non fornisce informazioni utili nel caso di un numero pari di errori. Infine, il bit di parità non permette di identificare la posizione dell’errore. Abbiamo quindi a che fare con un bit rilevatore, ma non correttore.

2.4. Il codice di Hamming (7,4)

Un solo bit di parità, come si è visto, non consente di individuare la posizione degli errori di trasmissione, e quindi non consente la correzione di tali errori. Tuttavia, sfruttando piú volte l’idea del controllo di parità si riesce a costruire un codice 1-correttore, cioè un codice in grado di correggere al piú un errore di trasmissione. Ne è un esempio il codice di Hamming (7,4), che descriviamo brevemente in questa sezione e che riprenderemo nella descrizione del secondo incontro, quando avremo a disposizione una rappresentazione visiva.

Nel codice di Hamming (7,4) le parole di codice sono vettori di sette componenti, quattro delle quali dedicate al messaggio da trasmettere, mentre le tre rimanenti sono bit di controllo. Convenzionalmente, i tre bit di controllo sono messi nelle posizioni che sono potenze di 2: prima (2^0), seconda e quarta.

Ciascuno dei tre bit di parità controlla la presenza di errori su tre componenti dedicate al messaggio:

- il bit in posizione 1 controlla le componenti dispari: la terza, la quinta e la set-

tima, oltre ovviamente alla prima;

- il bit in posizione 2 controlla la seconda, la terza, la sesta e la settima componente;
- il bit in posizione 4 controlla le componenti dalla quarta alla settima.

L'assegnazione delle componenti da controllare è legata alla rappresentazione binaria dei numeri. Il primo bit controlla infatti quelle posizioni che rappresentate in binario terminano con 1: le posizioni dispari. Il secondo bit controlla le posizioni che in binario hanno un 1 come penultima cifra: la posizione 2 (10 in binario), la 3 (11), la 6 (110) e la 7 (111). Infine, il terzo bit di controllo, in posizione 4, si occupa delle posizioni che in binario hanno la cifra uno nella terza posizione da destra: la 4 (100), la 5 (101), la 6 e la 7. La numerazione binaria verrà approfondita durante il terzo incontro.

Ciascuno dei tre bit di parità assume valore '0' o '1' a seconda che la somma delle posizioni da esso controllate sia pari o dispari, rispettivamente. Se viene rilevato un errore, cioè se almeno una delle tre posizioni di controllo assume un valore che non si accorda con la parità delle posizioni controllate, il bit errato viene rilevato utilizzando contemporaneamente le informazioni. Ciò è possibile in quanto ogni posizione è controllata da un insieme di bit diversi da ogni altra posizione.

Ad esempio, se i bit in posizione 1 e 4 danno un valore di parità sbagliato, mentre quello in posizione 2 è corretto, significa che il bit errato si trova nella posizione controllata dai bit in posizione 1 e 4, e non da quello in posizione 2: si tratta della quinta posizione.

Con i vincoli precedenti, si dimostra facilmente che le parole di codice ammissibili (cioè quelle in cui tutte le parità sono corrette) sono le sedici seguenti:

0000000,	0001111,	0010110,	0011001,
0100101,	0101010,	0110011,	0111100,
1000011,	1001100,	1010101,	1011010,
1100110,	1101001,	1110000,	1111111.

Il codice di Hamming (7,4) è un codice 1-correttore e si verifica facilmente che la distanza minima è proprio 3.

3. Descrizione delle attività

In questa sezione descriviamo un percorso didattico articolato in tre incontri che permette di presentare agli studenti, attraverso attività ludico didattiche e laboratoriali, alcuni concetti di teoria dei codici.

3.1. Primo incontro

Parte 1: introduzione

L'obiettivo di questo incontro è quello di coinvolgere attivamente gli studenti nella soluzione di un problema, stimolando la loro creatività e la loro fantasia, in modo da

portarli ad apprezzare la genialità, la semplicità e l'eleganza dell'introduzione del bit di parità. Prima di proporre l'attività vera e propria, il docente introduce brevemente i concetti riportati nella sezione 2.2, spiegando cosa si intende con con parola di codice binario e con bit di ridondanza e presentando l'esempio della triplicazione dei bit.

Parte 2: esperienza ludico-didattica

Si dividono gli studenti in squadre, formate da 4-6 ragazzi ciascuna, e si chiede ad ogni squadra di concordare al proprio interno una regola con cui inserire, in una parola di codice formata da 4 bit, un bit supplementare che funzioni da bit rilevatore, cioè che permetta di individuare un eventuale errore occorso durante la trasmissione della parola di codice. Le squadre avranno circa 10 minuti di tempo per decidere la strategia con cui inserire tale bit.

Allo scadere dei 10 minuti ogni squadra viene divisa in due gruppi e ad ognuno dei due gruppi vengono assegnate cinque parole di codice formate da 4 bit ciascuna. Ogni gruppo deve trascrivere le cinque parole di codice ricevute inserendo il bit rilevatore come concordato con i compagni di squadra.

Le nuove parole di codice, formate da 5 bit ciascuna e ottenute a partire dalle parole originali aggiungendo il bit rilevatore, vengono poi consegnate all'insegnante, il quale simula un errore casuale di trasmissione. Piú precisamente, l'insegnante copia le parole di codice ricevute modificando a caso al massimo un bit per ogni parola e poi le consegna all'altro gruppo della stessa squadra. A questo punto ogni gruppo deve identificare, tra le parole di codice ricevute, quelle che risultano corrette, e trascrivere le parole di codice originarie eliminando il bit rilevatore.

Quando tutti i gruppi hanno terminato la verifica delle parole ricevute dall'insegnante, si procede ad assegnare il punteggio ad ogni squadra nel modo seguente: $+1$ per ogni parola errata identificata come tale e per ogni parola corretta riconosciuta e decodificata correttamente eliminando il bit rilevatore; -1 per ogni parola corretta segnalata come errata o decodificata in maniera errata e per ogni parola errata e segnalata come corretta. Vince la squadra che totalizza il punteggio maggiore.

Al fine di rendere l'attività piú efficace, evitando i tempi morti e riducendo inutili attese, è possibile utilizzare post-it di colori diversi per le diverse squadre e scrivere in anticipo le dieci parole di codice da assegnare ad ogni squadra, identificando ogni parola con un numero. In questo modo i giocatori, ogni volta che riceveranno una parola di codice e dovranno operare su di essa (aggiungendo un bit rilevatore o decodificando una parola ricevuta), dovranno utilizzare un nuovo post-it riportando però sempre il numero corrispondente. Può anche essere utile preparare una tabella con l'elenco delle parole di codice numerate (tabella 5.1) ed utilizzarla per tenere traccia degli errori di trasmissione inseriti. In questo modo sarà piú rapido assegnare il punteggio alle diverse squadre. Notiamo che l'insegnante può usare tutte le parole in $\{0,1\}^4$, per cui nulla vieta che parole di codice distino 1. Ciò implica che il codice corregga al piú $\frac{1-1}{2} = 0$ errori, per cui a questo livello si può parlare solo di bit rilevatori e non correttori.

Al termine del gioco, ogni squadra dovrà spiegare la regola utilizzata ai com-

Numero	Parola originale	Errore di trasmissione
1	0011	no
2	1001	sí
3	0111	no
4	0101	no

Table 5.1: Esempio di riepilogo delle parole di codice consegnate.

pagni, i quali a loro volta dovranno cercare di individuare eventuali punti di debolezza fornendo, se possibile, esempi di possibili errori di trasmissione in cui la regola non risulta efficace. Sarà interessante vedere le proposte dei ragazzi (spesso davvero ingegnose), ma anche l'entusiasmo e la tenacia con cui gli studenti cercheranno di individuare le eventuali lacune nelle proposte dalle squadre avversarie!

Parte 3: il bit di parità

A questo punto viene introdotta (nel caso in cui non sia già emersa dalla discussione precedente) l'idea del bit di parità che non mancherà di stupire i ragazzi per la sua semplicità e al tempo stesso per la sua efficacia. Dopo aver spiegato l'idea, può essere interessante far osservare ai ragazzi che la posizione nella quale si decide di inserire il bit di parità all'interno del codice è assolutamente irrilevante ma, affinchè il ricevente possa ricostruire la parola originale, è necessario che sia stata preventivamente concordata tra chi trasmette e chi riceve.

3.2. Secondo incontro

Parte 1: introduzione

L'obiettivo di questo secondo incontro è quello di utilizzare quanto imparato sul bit di parità nell'incontro precedente per introdurre il codice di Hamming (7,4). A differenza di quanto fatto del primo incontro, questa volta non si tratta solo di stabilire se si è verificato errore di trasmissione, ma anche di identificare quale sia il bit errato, e quindi correggerlo. Per far questo gli studenti lavoreranno a gruppi con oggetti tangibili, utilizzando degli anelli di corda per rappresentare gli insiemi e dei cartoncini per rappresentare i diversi tipi di bit (bit di dati e bit di ridondanza). Per l'attività è necessario procurarsi, per ogni gruppo, degli anelli di corda di tre colori diversi, tre cartoncini rotondi degli stessi colori degli anelli con la scritta p_1, p_2 e p_3 (che rappresentano i bit di ridondanza) e quattro cartoncini quadrati con le scritte d_1, d_2, d_3, d_4 che rappresentano i quattro bit che formano la parola di codice che si vuole trasmettere.

Parte 2: Il gusto della scoperta

Dopo aver diviso gli studenti in gruppi, l'insegnante dà ad ogni gruppo tre anelli di corda e chiede di disporre i quattro bit di dati $\{d_1, d_2, d_3, d_4\}$ ed i tre bit di ridon-

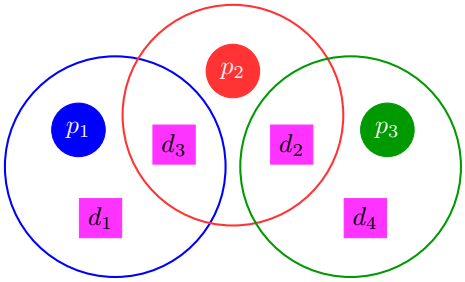

Figura 2: Una distribuzione (errata) dei bit all'interno degli anelli.

danza $\{p_1, p_2, p_3\}$ all'interno dei tre anelli, in modo tale che $\{p_1, p_2, p_3\}$ agiscano come bit di parità per i bit che si trovano nel loro anello di appartenenza e che la configurazione trovata permetta di identificare in maniera univoca un eventuale (unico) errore di trasmissione nell'insieme dei sette bit $\{p_1, p_2, p_3, d_1, d_2, d_3, d_4\}$. L'obiettivo dell'attività è trovare una configurazione che permetta di identificare la presenza e l'eventuale posizione di un generico bit errato, sia esso un bit di dati o un bit di ridondanza.

Per meglio chiarire come affrontare l'attività, può essere utile mostrare ai ragazzi un esempio. Nella situazione rappresentata in Figura 2, è possibile fare le seguenti considerazioni:

1. un valore errato del bit di parità p_2 accompagnato da valori corretti di p_1 e p_3 significa che uno dei valori che si trova all'interno dell'anello rosso è sbagliato e, poichè quelli negli anelli azzurro e verde sono corretti, l'errore corrisponde certamente al valore del bit p_2.

2. D'altra parte, un valore errato per i bit di parità p_1 e p_2, accompagnato da un valore corretto per il bit di parità p_3, significa che uno dei bit che si trova all'interno dell'anello rosso è sbagliato ed è anche sbagliato uno dei valori che si trova dentro l'anello azzurro, quindi è necessariamente sbagliato il valore di d_3 che è contenuto in entrambi gli anelli.

3. Infine, un valore errato del bit di parità p_1, accompagnato da un valore corretto di p_2 e di p_3, significa che uno dei bit che si trova nell'anello azzurro è sbagliato e, essendo d_3 corretto perchè all'interno dell'anello rosso, il bit errato è p_1 oppure d_1, ma non si hanno informazioni sufficienti a stabilire quale dei due sia quello errato.

A causa di 3., la configurazione proposta in Figura 2 non permette di rilevare e correggere un eventuale errore in posizione generica, anche se (come mostrato nei casi 1. e 2.) in alcuni casi risulta comunque efficace per identificare e correggere eventuali errori.

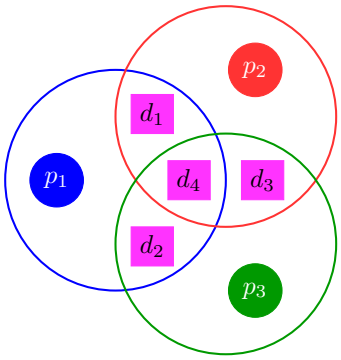

Figura 3: La corretta rappresentazione insiemistica.

Parte 3: codice di Hamming (7,4)

Dopo qualche tentativo, ed eventualmente qualche suggerimento da parte dell'insegnante, i vari gruppi arriveranno ad identificare la soluzione corretta che corrisponde al codice di Hamming (7,4) ed è riportata in Figura 3.

A questo punto è importante far osservare agli studenti che per utilizzare il codice di Hamming (7,4) per identificare e correggere una parola di codice binaria è fondamentale che coloro che trasmettono e coloro che ricevono si siano accordati sulla posizione dei bit correttori all'interno della parola di codice. Supponiamo ad esempio di aver deciso di posizionare prima i bit dei dati e poi i bit correttori e di dovere verificare se nella parola di codice 0010101 è presente un errore di trasmissione. In questo caso, ordinando sia i bit di dati che i bit correttori nel modo naturale si ha: '$d_1\,d_2\,d_3\,d_4\,p_1\,p_2\,p_3$'= '0010101'. Per verificare se durante la trasmissione si è verificato un errore, cominciamo a controllare la parità nei diversi anelli:

- anello blu: $d_1+d_2+d_4+p_1=0+0+0+1=1$, quindi uno dei bit presenti nell'anello azzurro contiene un errore;
- anello rosso: $d_1+d_3+d_4+p_2=0+1+0+0=1$, quindi uno dei bit presenti nell'anello rosso contiene un errore;
- anello verde: $d_2+d_3+d_4+p_3=0+1+0+1=2$, quindi tutti i bit presenti nell'anello verde sono corretti.

Questo significa che il bit errato si trova sia dentro l'anello azzurro che dentro l'anello rosso, ma fuori dall'anello verde, ovvero il bit errato è quello che corrisponde a d_1. Quindi, sapendo che nella parola '$d_1\,d_2\,d_3\,d_4\,p_1\,p_2\,p_3$'= '0010101' il valore che corrisponde a d_1 è sbagliato, possiamo correggerlo ottenendo la parola originale '1010101'.

Parte 4: esperienza ludico-didattica

Una volta compresa l'idea del codice di Hamming (7,4), è importante che gli studenti provino a mettere in pratica quanto appena imparato. Molto spesso infatti lo studente ha la sensazione di aver capito un concetto, ma quando prova ad applicarlo si accorge di non possederlo veramente.

Dopo aver diviso la classe in squadre di 4-6 studenti ciascuna, si assegnano ad ogni squadra 10 parole formate da sette bit ciascuna, precisando che i primi quattro sono i bit di dati, mentre gli ultimi tre sono i bit correttori p_1, p_2 e p_3 descritti prima. Ogni squadra deve, nel minor tempo possibile, decodificare le parole assegnate, correggendo gli eventuali errori di trasmissione. Per farlo può ovviamente utilizzare gli anelli di corda e i bit di cartoncino distribuiti durante l'attività precedente. Quando una squadra dichiara di aver terminato, anche le altre squadre si fermano e consegnano le parole decodificate all'insegnante. Il punteggio viene assegnato nel modo seguente: $+1$ per ogni parola decodificata correttamente, -1 per ogni parola decodificata in maniera errata e 0 per ogni parola non decodificata. Vince la squadra che totalizza il punteggio maggiore.

Dal punto di vista didattico, la scelta fatta per l'assegnazione dei punteggi mira a far riflettere gli studenti sul fatto che talvolta è meglio procedere con calma e lasciare non svolto qualche esercizio piuttosto che consegnare in gran fretta, ma essere poi fortemente penalizzati per aver fatto un gran numero di errori.

D'altra parte, durante questa attività gli studenti toccheranno con mano il fatto che il codice di Hamming (7,4) è "scomodo" da utilizzare. Tanto era semplice verificare l'eventuale presenza di un errore attraverso il singolo bit di parità, tanto appare farraginosa l'applicazione della versione con tre bit di parità. Ed è proprio su questa riflessione che si innestano le attività del terzo incontro.

3.3. Terzo incontro

Parte 1: numerazione binaria

Come già osservato alla fine del secondo incontro, l'utilizzo pratico del codice di Hamming (7,4) per individuare e correggere eventuali errori in una parola binaria di 7 caratteri presenta alcune criticità di tipo operativo, dovute principalmente alla necessità di organizzare i caratteri in opportuni sottoinsiemi che non sono immediatamente identificabili a partire dalla posizione dei caratteri stessi nella parola di codice considerata. Al fine di rendere piú facile l'identificazione dei diversi sottoinsiemi di bit può risultare utile posizionare in maniera strategica i tre bit correttori. Per far questo il primo passo consiste nell'introdurre il sistema di numerazione binario. In tale sistema di numerazione ci sono solo due cifre (0 e 1) ed ogni cifra in posizione n (contando le posizioni da destra verso sinistra e iniziando da 0) si considera moltiplicata per 2^n, anzichè per 10^n, come avviene nella numerazione decimale. Ad esempio il numero 1101_2 corrisponde al numero decimale 13, poichè

$$1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 1 = 13.$$

Data una parola di codice formata da quattro caratteri binari, inseriamo i tre bit di ridondanza nelle posizioni che corrispondono alle potenze di due, ovvero $2^0 = 1$, $2^1 = 2$, $2^2 = 4$ e posizioniamo i quattro bit di dati nelle altre posizioni seguendo l'ordine naturale:

1	2	3	4	5	6	7
p_1	p_2	d_1	p_3	d_2	d_3	d_4

In questo modo sarà facile riconoscere i sottoinsiemi di bit considerati nell'incontro precedente. Infatti il bit correttore p_1 controlla la parità dei caratteri che si trovano nelle posizioni che corrispondono a numeri binari che hanno 1 come ultima cifra:

$$001_2 = 1, \quad 011_2 = 3 \quad 101_2 = 5 \quad 111_2 = 7.$$

Visivamente,

1	2	3	4	5	6	7
p_1	p_2	d_1	p_3	d_2	d_3	d_4

ll bit correttore p_2 controlla la parità dei caratteri che si trovano nelle posizioni che corrispondono a numeri binari che hanno 1 come penultima cifra:

$$010_2 = 2, \quad 011_2 = 3, \quad 110_2 = 6, \quad 111_2 = 7,$$

ovvero

1	2	3	4	5	6	7
p_1	p_2	d_1	p_3	d_2	d_3	d_4

Infine, il bit correttore p_3 controlla la parità dei caratteri che si trovano nelle posizioni che corrispondono a numeri binari che hanno 1 come terz'ultima cifra:

$$100_2 = 4, \quad 101_2 = 5, \quad 110_2 = 6, \quad 111_2 = 7,$$

cioè

1	2	3	4	5	6	7
p_1	p_2	d_1	p_3	d_2	d_3	d_4

Con questa scelta per le posizioni dei diversi bit, l'utilizzo del codice di Hamming (7,4) diventa molto piú agevole. Ad esempio, se si riceve la parola di codice '0011010' e si deve identificare un eventuale errore di trasmissione, come prima cosa bisogna calcolare la somma degli elementi che corrispondono a p_1:

0	0	1	1	0	1	0

Poichè $0+1+0+0 = 1$, si deduce che si è verificato un errore di parità in questo insieme di bit. Calcolando la somma degli elementi che corrispondono a p_2, il controllo di parità dà esito positivo poichè $0+1+1+0 = 2$:

0	0	1	1	0	1	0

Infine, osservando che anche la somma degli elementi che corrispondono a p_3 è pari ($1+0+1+0=2$),

0	0	1	1	0	1	0

sappiamo che c'è stato un errore di trasmissione (perchè uno dei bit di parità è errato) e che l'errore si trova certamente in una posizione dispari. Inoltre, poichè i bit nelle posizioni 3 e 7 sono corretti (p_2 non dà errore), cosí come quello nella posizione 5 (anche p_3 non dà errore), possiamo concludere che il bit errato è quello in prima posizione. È quindi possibile correggere la parola di codice ricevuta semplicemente invertendo il valore del primo bit: da '0011010' a '1011010'.

Una volta effettuata la correzione, è possibile ricostruire il messaggio originale cancellando i caratteri che si trovano nelle posizioni che corrispondono alle potenze di due: ~~1~~ ~~0~~ 1 ~~1~~ 0 1 0, cioè 1010.

Parte 2: esperienza ludico-didattica

Si dividono gli studenti in squadre da 4-6 studenti l'una ed ogni squadra a sua volta in due gruppi; si assegnano poi ad ogni gruppo cinque parole di codice binario formate da quattro caratteri ciascuna. Gli studenti devono aggiungere ad ogni parola di codice ricevuta i tre bit correttori secondo la logica precedentemente illustrata. Le nuove parole di codice, di sette caratteri ciascuna, devono essere consegnate all'insegnante, il quale, trascrivendole, simula un errore casuale di trasmissione modificando a caso al piú un carattere per ogni parola, e provvede allo scambio tra i due gruppi della stessa squadra. A questo punto ogni gruppo trascrive, identificando e correggendo eventuali errori, le parole originarie (formate da quattro caratteri). I punteggi sono assegnati nel modo seguente: $+1$ per ogni parola decodificata in maniera corretta e -1 per ogni parola decodificata in maniera errata. Vince la squadra che totalizza il punteggio maggiore.

Come già discusso nell'ambito delle attività proposte nel secondo incontro, l'obiettivo di questo gioco è che gli studenti non si limitino ad una fruizione passiva dell'informazione, ma partecipino attivamente al processo di apprendimento attraverso un'esperienza diretta. In questo modo i ragazzi saranno in grado di apprezzare come il fatto di aver inserito i bit correttori nelle posizioni che corrispondono a potenze di 2 permetta di sveltire le operazioni necessarie per identificare la presenza e l'eventuale posizione del bit errato.

D'altra parte, nonostante questo presenti un indubbio miglioramento rispetto al metodo insiemistico proposto nel secondo incontro, si può fare un ulteriore passo avanti, sfruttando meglio il posizionamento strategico dei bit rilevatori nelle posizioni che corrispondono alle potenze di 2. Consideriamo nuovamente l'esempio precedente, e supponiamo di ricevere sempre la parola di codice 0011010 e di voler stabilire se si sia verificato un errore di trasmissione.

Poichè il bit di parità p_1 rileva la presenza di un errore, sappiamo che la posizione del bit errato, scritta in base due, avrà un 1 come ultimo carattere a destra:

$$??1_2.$$

Poichè il bit di parità p_2 non rileva la presenza di errori, sappiamo che la posizione del bit errato, scritta in base due, non avrà un 1 come penultimo carattere, quindi dovrà avere necessariamente uno 0:

$$?01_2.$$

Poichè il bit di parità p_3 non rileva la presenza di errori, come nel caso precedente anche il terz'ultimo carattere dovrà essere necessariamente uno 0:

$$001_2.$$

Quindi abbiamo determinato la posizione del bit errato, che scritta come numero binario è $001_2 = 1$, ovvero come visto precedentemente il bit da correggere in questo caso è il primo.

In generale, una volta individuati quali siano i bit di parità che segnalano la presenza di un errore, per determinare la posizione del bit errato è sufficiente scrivere il numero (in base due) che ha 1 in corrispondenza dei bit correttori che rilevano errori e 0 in corrispondenza dei bit correttori che non ne rilevano.

Riprendendo l'esempio proposto nella sezione 2.4, se dai vari controlli di parità risultassero degli errori associati ai bit correttori p_1 e p_3 mentre la parità relativa al sottoinsieme di bit controllato da p_2 risultasse corretta, il bit da correggere sarebbe quello in posizione $101_2 = 5$, ovvero d_2.

Parte 3: aumentando il numero di bit

Il lavoro fatto fino ad ora ha permesso di mettere in luce come, nella trasmissione di una parola costituita da 7 bit, sia possibile individuare e correggere un singolo errore utilizzando 3 dei 7 bit totali come bit di ridondanza. In questo caso solo i 4/7 del messaggio corrispondono all'informazione originale, mentre i 3/7 servono per individuare e correggere un eventuale errore di trasmissione. Rispetto all'esempio di codice correttore mostrato nel primo incontro (quello in cui ogni singolo bit veniva triplicato) si tratta certamente di un miglioramento (in quel caso solo 1/3 dei dati trasmessi corrispondeva a informazione, mentre 2/3 erano dedicati alla ridondanza), ma è interessante capire cosa succede al numero di bit correttori di Hamming all'aumentare della lunghezza del messaggio trasmesso. Prima di procedere, proponete il seguente sondaggio tra i vostri studenti:

> Quanti bit di ridondanza saranno necessari per individuare e correggere un eventuale singolo errore nella trasmissione di una parola di codice formata da 255 bit?
>
> (a) Meno di 10. (b) Tra 10 e 20. (c) Tra 20 e 30. (d) Piú di 30.

In generale la risposta piú votata sarà la (d), dal momento che nell'esperienza fatta i bit di ridondanza sono 3 su 7 bit totali.

Prima di dichiarare la risposta corretta, osservate come nel caso del codice di Hamming (7,4), i tre bit di ridondanza sono stati messi nelle posizioni corrispondenti alle potenze di due e che la lunghezza delle parole di codice è $7 = 2^3 - 1$. D'altra parte, con due simboli a disposizione (in questo caso 0 e 1) e al piú tre cifre si possono ottenere 2^3 numeri: 0, 1, 10, 11, 100, 101, 110, 111. A ciascuno di questi numeri potremmo associare una posizione del codice di Hamming (7,4), se non fosse che i numeri scritti sopra sono uno in piú della lunghezza delle parole di codice. Questo perchè di solito la "posizione zero", corrispondente al primo degli otto numeri, non viene contata nella costruzione del codice tradizionale. Essa viene usata in quello che viene chiamato codice di Hamming esteso ed il suo bit rappresenta un controllo di secondo livello, grazie al quale è possibile individuare, ma non correggere, il verificarsi di due errori. Per un approfondimento si rimanda a [3].

Tolto lo zero, restano quindi sette numeri, ai quali associamo, come abbiamo già fatto nelle attività precedenti, le posizioni del codice di Hamming (7,4): avremo quindi quattro posizioni (3, 5, 6 e 7) per codificare il messaggio che vogliamo inviare e tre posizioni (1, 2 e 4) per i bit di ridondanza.

Aumentando la lunghezza delle parole di codice, ci saranno altre posizioni corrispondenti a potenze di due, che immaginiamo verranno occupate da bit di ridondanza. In particolare, la posizione 8 sarà la sede del quarto bit di ridondanza: in questo caso la lunghezza massima delle parole di codice controllabili da 4 bit di ridondanza sarà $2^4 - 1 = 15$. Di conseguenza, in una parola formata da 15 bit le posizioni che potremo dedicare alla trasmissione dei dati saranno 11.

I caratteri p_1, p_2 e p_3 verranno utilizzati come nel caso precedente, mentre il nuovo carattere p_4 (che andrà inserito in posizione $2^3 = 8$) controllerà la parità del gruppo di bit che si trovano in posizioni corrispondenti ai numeri che in binario hanno un 1 in quart'ultima posizione, ovvero i numeri da 8 a 15:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
p1	**p2**	d_1	**p3**	d_2	d_3	d_4	**p4**	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
p1	**p2**	d_1	**p3**	d_2	d_3	d_4	**p4**	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
p1	**p2**	d_1	**p3**	d_2	d_3	d_4	**p4**	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
p1	**p2**	d_1	**p3**	d_2	d_3	d_4	**p4**	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}

A questo punto la generalizzazione dovrebbe sorgere spontanea: ad esempio, per stabilire quanti bit serviranno per individuare e correggere un possibile errore nella trasmissione di 63 caratteri, è sufficiente osservare che i numeri da 1 a 63 in base due si scrivono utilizzando sei cifre binarie (infatti $111111_2 = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 63$) . Quindi, per individuare e correggere un eventuale errore in una parola di codice binario formata da 63 caratteri sono sufficienti 6 bit. Analogamente, per individuare e correggere un eventuale errore in una parola di codice binario formata da 255 caratteri sono sufficienti 8 bit, e in generale per individuare e correggere un eventuale errore in

una parola di codice binario di lunghezza $2^n - 1$ sono sufficienti n bit!

Questo risultato, piuttosto stupefacente, oltre a consentire un raffronto fra crescita lineare e logaritmica mostra ancora una volta la genialità dell'intuizione di Hamming, e mette in luce come alcune idee matematiche anche molto semplici possano avere ricadute pratiche di grandissimo impatto.

4. Conclusione

In questo lavoro abbiamo descritto una serie di attività didattiche che possono essere utilizzate per presentare agli studenti della scuola secondaria alcuni temi legati alla teoria dei codici. Attraverso momenti ludici alternati a momenti di confronto e restituzione, vengono introdotte le idee che stanno alla base del codice di Hamming, presentando i concetti di errore di trasmissione, codice rilevatore e codice correttore.

L'approccio laboratoriale permette agli studenti di cimentarsi in prima persona in alcuni semplici problemi, dando loro modo di poter apprezzare fino in fondo l'eleganza e la semplicità della soluzione trovata da Hamming.

References

[1] ETZION T. AND STORME L., *Galois geometries and coding theory*, Des. Codes Cryptogr. **78**(1) (2016), 311–350.

[2] GIUZZI L., *Codici correttori. Una introduzione*, Springer 2006.

[3] HAMMING R.W., *Error detecting and error correcting codes*, Bell System Tech J. **29**(2) (1950), 147–160.

[4] SHANNON C.E., *A mathematical theory of communication*, Bell System Tech J. **27** (1948), 379–423.

AMS Subject Classification: 97D50, 97M10, 97K20, 94B05

Paola MORANDO,
Dipartimento di Scienze Agrarie e Ambientali - Produzione, Territorio, Agroenergia, Università degli Studi di Milano
via Celoria 2, 20133 Milano, ITALY
e-mail: paola.morando@unimi.it

Silvia M.C. PAGANI,
Dipartimento di Matematica e Fisica "N. Tartaglia", Università Cattolica del Sacro Cuore
via della Garzetta 48, 25133 Brescia, ITALY
e-mail: silvia.pagani@unicatt.it

Lavoro pervenuto in redazione il 15.04.2022.