

Introduzione

La teoria dell'informazione risponde principalmente alle seguenti domande:

1. data una sorgente che emette messaggi che devono essere inviati ad uno o più destinatari, quale è il modo meno costoso per rappresentare con una data qualità l'informazione da trasmettere o da memorizzare su un qualche supporto fisico?
2. dato un canale di trasmissione rumoroso (non ideale), la cui uscita sia una replica distorta del segnale di ingresso, come è possibile trasmettere o memorizzare l'informazione in modo il più possibile affidabile?

Per entrambe le domande si hanno due aspetti da considerare, uno teorico e l'altro pratico. L'aspetto pratico mette in evidenza le tecniche che possono essere utilizzabili per la codifica di una sorgente e per la trasmissione dell'informazione. L'aspetto teorico, invece, si presta alla scelta delle prestazioni migliori che possono essere ottenute con i vincoli prefissati sulla complessità dell'elaborazione. Il caso più semplice ma anche più utilizzato è senza vincoli di complessità.

Il nostro compito è quello di concentrarci sull'aspetto teorico relativo alle trasmissioni con disturbi e alle prestazioni della codifica di sorgente.

I problemi più semplici e più noti della codifica di sorgente riguardano la codifica senza perdita di informazione e, quindi, invertibile senza alcuna degradazione. ad esempio se vogliamo comprimere e trasmettere un qualsiasi file testo è ovvio che lo vogliamo riottenere in ricezione senza alcuna perdita di informazione. Invece, se consideriamo una immagine o un segnale vocale vogliamo riottenerli con una perdita minima di informazione, cioè si tollera una certa degradazione se essa è nota a priori. La disponibilità di avere una certa imprecisione nella ricostruzione è sicuramente condizionata dal risparmio ottenibile dalla codifica di sorgente.

Analogamente, quando si trasmette informazione se codificata opportunamente si pretende una probabilità di errore molto piccola, addirittura nulla. In altri casi più complessi è, invece, accettabile una codifica con probabilità di errore prefissata e non necessariamente piccola.

Codifica di sorgente

Il primo problema da porsi è definire e, quindi, misurare la quantità di informazione emessa da sorgente, chiamata entropia della sorgente di messaggi. Questa grandezza deve essere strettamente legata al costo minimo per la rappresentazione della sequenza di messaggi emessi dalla sorgente.

Se, per esempio, ci è richiesto il risultato di alcuni lanci consecutivi di una moneta non truccata (testa e croce), ciò può essere visto come una successione binaria di uni e zero. Anche se in questo caso, l'informazione corrispondente non ha molto valore per il destinatario, si vedrà che fra le sorgenti binarie questa ha il massimo di informazione: se davvero occorre far conoscere il risultato di un lancio di una moneta, è inevitabile trasmettere un bit, essendo ogni lancio indipendente da tutti gli altri e non avendo la sorgente alcuna preferenza per le teste e per le croci, cioè per gli uni e gli zeri. L'informazione, corrispondente ad un bit per lancio, non misura l'utilità del messaggio ma il costo della rappresentazione con cifre binarie. In definitiva, il progettista e/o il gestore dei sistemi di trasmissione non hanno il compito di sindacare su cosa è richiesto da trasmettere.

Le sorgenti di messaggi possono essere classificate come discrete e continue.

Noi considereremo solo sorgenti discrete che emettono una successione di cifre che possono essere binarie e non. Le sorgenti continue nel tempo possono essere discretizzate mediante l'uso di un campionamento con una contenuta degradazione se la frequenza di campionamento è scelta in modo opportuno. Le sorgenti continue in ampiezza, invece, vengono rese discrete mediante l'uso della quantizzazione. In generale, per noi le sorgenti saranno già discrete sia nel tempo sia nell'ampiezza.

Una ulteriore suddivisione delle sorgenti, le classifica come sorgenti con memoria e sorgenti senza memoria.

Le proprietà statistiche della sorgente sono importanti per la possibilità di codifica economica: occorre, infatti, sfruttare sia l'eventuale non equiprobabilità dei messaggi sia la eventuale non indipendenza dei simboli successivamente emessi dalla sorgente (memoria della sorgente), assegnando stringhe codificate più lunghe e quindi più costose ai messaggi o sequenze di messaggi meno frequenti.

Le sorgenti senza memoria sono più semplici da trattare e più facili da codificare. Peccato che sono molto rare nella pratica. Per una descrizione completa di una sorgente senza nome è sufficiente fornire le singole probabilità di emissione dei messaggi m_i tratto da un insieme M detto alfabeto.

Le sorgenti con memoria sono caratterizzate anche dalle probabilità congiunte, dei vari ordini, di messaggi successivi. Cioè è necessario ricordare i messaggi precedentemente usciti ed in funzione

Teoria dell'Informazione

di essi si ha la probabilità di un dato nuovo messaggio. Sfortunatamente tale informazione spesso non è nota a priori. Ad esempio, se vogliamo analizzare un testo scritto per annotare le frequenze¹ dei singoli caratteri, delle coppie di caratteri, delle terne e così via, ... il compito diventa sempre più difficile perché servono parti sempre più lunghi di testo.

Spesso non sono note a priori neanche le probabilità dei singoli messaggi. Tuttavia, anche in questo caso esistono tecniche di codifica delle sorgenti che danno delle ottime prestazioni vicine ai limiti teorici, qualunque essi siano.

Il concetto di Entropia

Preso una sorgente discreta che emette messaggi scelti fra un numero finito di M possibili (m_i , $i = 1, \dots, M$, alfabeto della sorgente) e supposto che tale sorgente sia senza memoria, cioè i messaggi sono emessi indipendentemente uno dall'altro, si definisce l'informazione associata a ciascun messaggio come:

$$I_i = -\log_2 p(m_i) = \log_2 \left(\frac{1}{p(m_i)} \right)$$

dove le $p(m_i)$ rappresentano le probabilità di emissione dei vari possibili messaggi. Questa definizione è congruente col fatto che tanto più il messaggio è improbabile tanta più grande sarà l'informazione ad esso associata.

Inoltre, la definizione precedente ha una importante proprietà additiva, legata alla indipendenza con cui i messaggi sono generati. Infatti, l'informazione associata alla coppia di messaggi m_i, m_k è:

$$I_{i,k} = -\log_2 p(m_i, m_k) = -\log_2 (p(m_i) \cdot p(m_k)) = -\log_2 p(m_i) - \log_2 p(m_k) = I_i + I_k.$$

Se si media questa quantità di informazione su tutti i possibili messaggi della sorgente, si ottiene quella che è denominata l'entropia della sorgente $H(M)$:

$$H(M) = \sum_{i=1}^M p(m_i) \cdot I_i = \sum_{i=1}^M p(m_i) \cdot \log_2 \left(\frac{1}{p(m_i)} \right) = -\sum_{i=1}^M p(m_i) \cdot \log_2 (p(m_i)).$$

¹ La frequenza dei caratteri è fondamentale per la crittografia di un messaggio segreto.

Teoria dell'Informazione

Quando la sorgente emette messaggi equiprobabili, l'entropia della sorgente diventa massima e vale:

$$H(M)_{\max} = \log_2 \frac{1}{p(m)} = \log_2 M.$$

L'unità di misura dell'entropia è il bit. Corrisponde all'informazione associata a ciascun messaggio emesso da una sorgente binaria con $p(m_1) = p(m_2) = 0.5$.

L'importanza della nozione di entropia sta nel suo significato operativo. Rappresenta, infatti, la minima lunghezza media dei codici di rappresentazione dei messaggi della sorgente.

Consideriamo una sequenza di N messaggi. La sequenza dei messaggi costituisce un campione statistico in cui la frequenza relativa dei singoli messaggi, al tendere di N all'infinito, tende a coincidere² con la probabilità dei messaggi stessi. In altre parole, in un messaggio composto da N messaggi, al tendere di N all'infinito, si troveranno $N \cdot p(m_1)$ messaggi m_1 , $N \cdot p(m_2)$ messaggi m_2 , e così via. In queste condizioni tutti gli altri messaggi composti possono essere trascurati perché hanno probabilità nulla di emissione. Allora, i messaggi considerati sono tutti equiprobabili ed hanno probabilità:

$$p = [p(m_1)]^{N \cdot p(m_1)} \cdot [p(m_2)]^{N \cdot p(m_2)} \cdots [p(m_M)]^{N \cdot p(m_M)} = \prod_{i=1}^M [p(m_i)]^{N \cdot p(m_i)},$$

avendo tenuto conto del fatto che i messaggi sono indipendenti. Il numero di messaggi composti considerato è dato da $1/p$. Il numero di cifre binarie necessarie per codificare ciascun messaggio

composito è dato da $\log_2 \left(\frac{1}{p} \right)$. Perciò, mediamente, per codificare ciascun messaggio elementare

saranno utilizzati:

$$\frac{1}{N} \cdot \log_2 \left(\frac{1}{p} \right) = \frac{1}{N} \cdot \log_2 \left[\prod_{i=1}^M \left(\frac{1}{p(m_i)} \right)^{N \cdot p(m_i)} \right] = \sum_{i=1}^M p(m_i) \cdot \log_2 \frac{1}{p(m_i)} \quad [\text{bit}].$$

Tale quantità coincide proprio con l'entropia della sorgente.

² La probabilità con cui tendono a coincidere è 1.

Codifica a lunghezza variabile

Si è visto come sia possibile rappresentare i messaggi con un numero medio di cifre binarie pari all'entropia della sorgente codificando messaggi composti da sequenze molto lunghe di messaggi elementari.

Esistono procedure approssimate che consentono in molti casi una codifica efficiente senza dover ricorrere a lunghi blocchi di messaggi. Il principio sfruttato è quello della codifica a lunghezza variabile. Cioè codici più corti vengono assegnati ai messaggi che hanno una elevata probabilità e codici più lunghi ai messaggi meno probabili (fig.1).

m_i	$p(m_i)$	Cod. lung. fissa	Cod. lung. variabile
m_1	0.6	00	0
m_2	0.2	01	10
m_3	0.1	10	110
m_4	0.1	11	111

Fig.1-Schema di codifica a lunghezza variabile.

La lunghezza media di un codice si determina effettuando la somma dei prodotti della probabilità per il numero di bit corrispondenti:

$$L = \sum_{i=1}^M (p(m_i) \cdot n_i) .$$

Nel caso della fig.1, abbiamo che per il codice a lunghezza fissa il numero di bit per messaggio è sempre pari a due e, quindi, la lunghezza media sarà proprio pari a 2 bit. L'entropia con un semplice calcolo logaritmico vale circa 1,57 bit.

Per il codice a lunghezza variabile la lunghezza media vale: $1 \cdot 0,6 + 2 \cdot 0,2 + 3 \cdot 0,1 + 3 \cdot 0,1 = 1,6$ bit, mentre l'entropia non cambia.

Una caratteristica fondamentale di un codice è la sua efficienza che è pari al rapporto tra l'entropia e la lunghezza media:

$$\eta = \frac{H(M)}{L} .$$

E' ovvio che la codifica è ottimale se tale efficienza è prossima al valore unitario.

Teoria dell'Informazione

Nei due casi in questione si ha che per il codice a lunghezza fissa l'efficienza è pari a 0,785 ossia al 78,5%, mentre per il codice a lunghezza variabile è pari a 98,12%. Si nota immediatamente il miglior utilizzo del codice a lunghezza variabile rispetto a quello a lunghezza fissa.

In ogni caso deve essere garantita la univocità delle sequenze dei codici e la decodificabilità dei codici generati.

Generare i codici a lunghezza variabile diventa complicato se l'alfabeto della sorgente è grande. Vedremo due procedure di generazione dei codici: l'algoritmo di Huffman e l'algoritmo di Shannon-fano.

Per l'algoritmo di Huffman (fig.2), inizialmente i messaggi vengono ordinati secondo probabilità decrescenti, producendo un primo ordinamento. Poi, i due messaggi di minore probabilità vengono distinti da una prima cifra binaria. I due messaggi vengono, quindi, considerati come un unico gruppo, avente probabilità somma delle loro singole probabilità, il quale viene inserito al posto appropriato in un secondo ordinamento. La procedura viene ripetuta fino alla codificazione di tutti i messaggi. Alla fine, il codice relativo di un messaggio viene letto a partire da destra (fig.2).

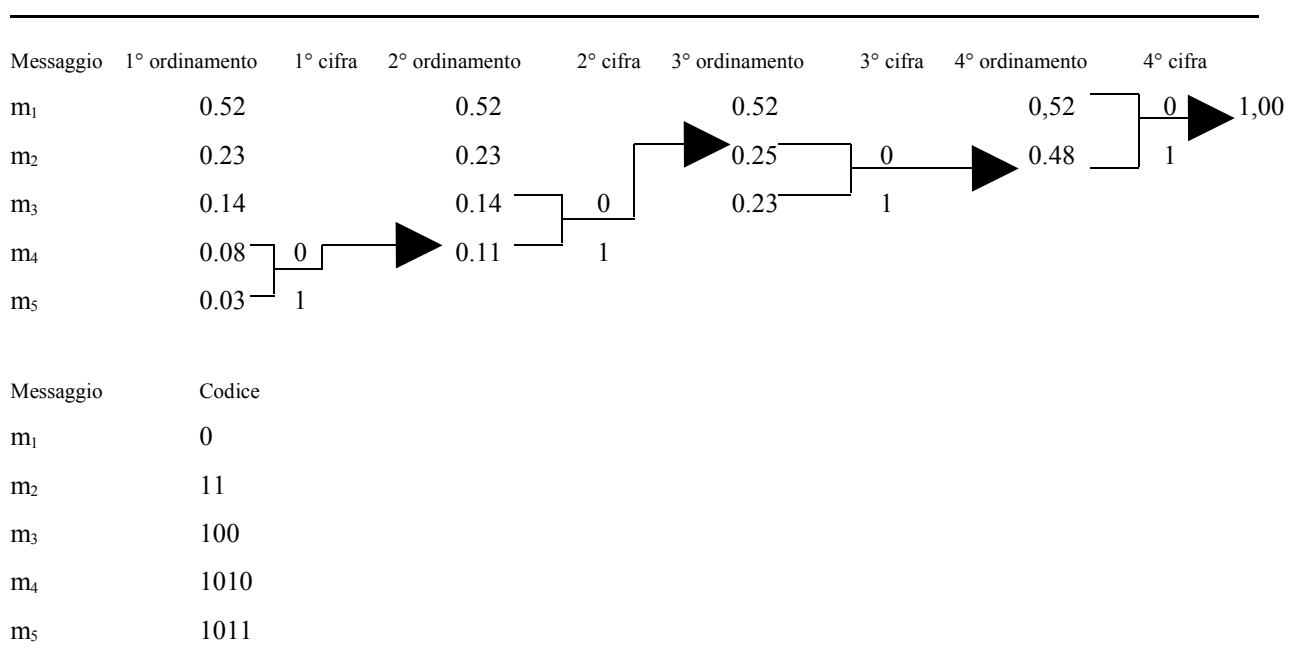


fig.2 – Codificazione di Huffman.

Teoria dell'Informazione

Per l'algoritmo di Shannon-Fano (fig.3) si costruisce una tabella dove vengono ordinati in ordine decrescente di probabilità i simboli dell'alfabeto nella prima colonna. Poi, nella prima fase si uniscono i messaggi in modo da dare circa il 50% di probabilità. Così si forniscono zeri e uni per ogni singolo gruppo. Poi, si ripete l'operazione in varie fasi fino al completamento del codice.

Utilizziamo lo stesso esempio mostrato per l'algoritmo di Huffman.

Probabilità	I fase	II Fase	III Fase	IVFase	Codice
0.52	<u>0</u>				0
0.23	1	<u>0</u>			10
0.14	1	1	<u>0</u>		110
0.08	1	1	1	<u>0</u>	1110
0.03	1	1	1	1	1111

Fig.3 – Codifica di Shannon-fano

Per i due codici visti si ha lo stesso numero di bit per messaggio, pertanto, si ha la stessa lunghezza media, la stessa entropia e la stessa efficienza anche se i codici sono diversi.

Le prestazioni ottenibili dal codificatore di Huffman o di Shannon-Fano tendono a quelle indicate dall'entropia quando i messaggi composti sono di sufficiente lunghezza. Nel caso si debba codificare una stringa di bit è quindi necessario raggruppare a blocchi i simboli elementari (bit) ottenendo simboli composti che verranno codificati con codici a lunghezza variabile.

Alcune conseguenze non trascurabili di una efficace codifica di sorgente sono le seguenti:

1. dalla sequenza binaria codificata, ed inviata al sistema di trasmissione, è stata rimossa quasi ogni ridondanza (ripetizione) e non è possibile comprimere ulteriormente i messaggi; quindi, la sequenza codificata contiene zeri ed uni pressoché equiprobabili;
2. errori nella trasmissione dei bit codificati sono più gravi di errori commessi trasmettendo una sorgente fortemente ridondante (come gli errori di stampa in un testo scritto, fastidiosi ma spesso innocui); se si è ricercata la massima compressione della sorgente il sistema di trasmissione deve essere molto affidabile.

In ogni caso, per tutte le codifiche possibili, uno dei problemi più importanti che sorgono nelle applicazioni pratiche è quello legato alla conoscenza a priori della probabilità di emissione dei simboli, che spesso non è nota ma può e deve essere stimata dai dati a disposizione.

Teoria dell'Informazione

L'approccio maggiorante utilizzato è quello dei codificatori universali, la cui struttura è descritta in fig.4. I messaggi in ingresso al codificatore sono inviati anche ad un "modellatore statistico" che a partire dalle frequenze relative dei vari messaggi, stima le probabilità di emissione dei simboli stessi e quindi i codici a lunghezza variabile da utilizzare. Inizialmente il codificatore opera con una tabella di probabilità standard. Ad intervalli regolari, le probabilità stimate dal modellatore statistico sono confrontate con quelle utilizzate in quel momento dal codificatore; se le due tabelle sono molto diverse viene aggiornata la tabella e quindi vengono calcolati nuovamente i codici a lunghezza variabile utilizzati.

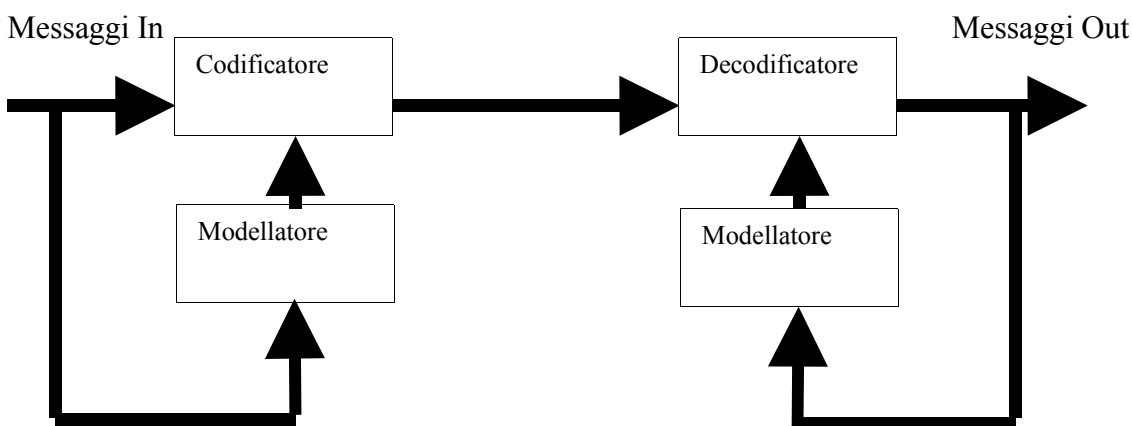


Fig.4 – Schema a blocchi di un codificatore e di un decodificatore universale.

Anche al ricevitore esiste un "modellatore statistico che opera sui messaggi già decodificati. Le stime ottenute da questo modellatore saranno coerenti con quelle ottenute al lato codificatore (fig.4). La tabella delle probabilità e i codici corrispondenti verranno aggiornati al lato decodificatore con le stesse modalità utilizzate per la codifica. In questo modo sono garantite sia la congruenza tra le modalità operative del codificatore e del decodificatore sia l'adattabilità del sistema di co-decodifica a sorgenti con statistica diversa.

Codifica di messaggi non indipendenti (con memoria)

Quando non è possibile considerare i messaggi emessi indipendenti tra loro, la definizione di entropia diventa più complessa. Non possiamo considerare le semplici probabilità di emissione $p(m_i)$, ma dobbiamo considerare le probabilità condizionate $p(m_i/S_j)$. Come al solito, m_i

Teoria dell'Informazione

rappresentano gli M messaggi elementari possibili, mentre S_j è uno dei possibili stati della sorgente descritto dagli h messaggi precedenti. Ovviamente, il numero di messaggi in memoria h può essere maggiore di M . Assumiamo, inoltre, che la memoria della sorgente si estende appunto su un numero di messaggi pari ad h . Chiaramente si hanno M^h possibili stati. L'entropia della sorgente quando essa è nello stato S_j vale:

$$H_j = \sum_{i=1}^M p\left(\frac{m_i}{S_j}\right) \cdot \log_2 \frac{1}{p\left(\frac{m_i}{S_j}\right)};$$

e mediando su tutti i possibili stati (M^h), si ottiene:

$$H = \sum_{j=1}^{M^h} p(S_j) \cdot H_j = \sum_{j=1}^{M^h} \sum_{i=1}^M p(S_j) \cdot p\left(\frac{m_i}{S_j}\right) \cdot \log_2 \frac{1}{p\left(\frac{m_i}{S_j}\right)} = \sum_{j=1}^{M^h} \sum_{i=1}^M p(m_i, S_j) \cdot \log_2 \frac{1}{p\left(\frac{m_i}{S_j}\right)}.$$

Nella formula precedente si è introdotta la probabilità congiunta fra il messaggio m_i e lo stato S_j .

Dalla formula si vede come il problema diventa rapidamente sempre più complicato all'allungarsi della memoria della sorgente.

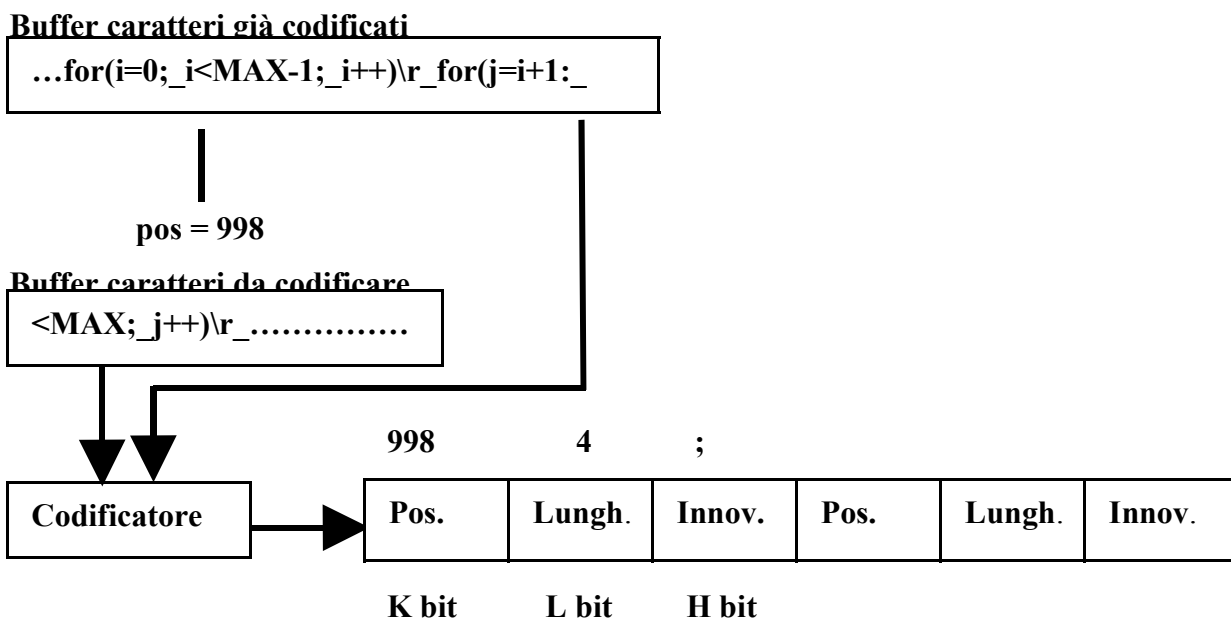
Qualora si vogliano utilizzare tecniche di codifica alla Huffman, a ciascun messaggio m_i non verrà più associato un solo codice, ma M^h ottenuti in base alla probabilità $p(m_i/S_j)$. Se le probabilità condizionate di emissione dei simboli vanno stimate direttamente dai dati da codificare la complessità del sistema può diventare insostenibile e le stime delle probabilità di emissione potrebbero essere inaffidabili per la scarsità, nel senso di quantità, del campione statistico.

Per la codifica dei messaggi emessi da sorgenti con memoria ci riferiamo a tecniche chiamate di "codificatori basati su vocabolario". Le tecniche più note sono state sviluppate da Lempel e Ziv, e sono alla base delle tecniche di compressione dei file utilizzate nei modem e nei programmi di compattazione file.

L'approccio a vocabolario si basa su un concetto molto semplice. Supponiamo, ad esempio, di dover codificare dei messaggi che altro non sono che caratteri alfanumerici che costituiscono un testo scritto. Piuttosto che codificare un carattere con un byte (codice ASCII), è più comodo e conveniente codificare ciascuna parola o gruppi di lettere con l'indice che la individua in un vocabolario noto sia al codificatore sia al decodificatore. Questo perché solo un numero molto limitato di sequenze di N lettere hanno un significato, in una determinata lingua, ed appariranno nel testo. Le sequenze senza significato o non appariranno mai o appariranno con una probabilità molto bassa.

Teoria dell'Informazione

L'approccio più semplice alla codifica basata su vocabolario è mostrata in fig.5. Nell'esempio i simboli da codificare sono caratteri alfanumerici caratterizzati da un byte ciascuno. I simboli da codificare entrano in un buffer di ingresso (Buffer "Caratteri da codificare") di dimensioni predefinite (nel nostro esempio tale dimensione è pari a $M = 2^L = 16$). Invece, i caratteri che man mano vengono codificati alimentano il Buffer dei "Caratteri già codificati", con dimensione, nel nostro esempio, di $N = 2^K - 1 = 1023$. E' proprio questo buffer che rappresenta il vocabolario del sistema.



Per esempio: K = 10, L = 4, H = 8

Pos. = 1...1023; 0 nel caso di nessuna stringa in comune.

Fig.5 – Schema di funzionamento di un semplice codificatore basato su vocabolario.

Il codificatore ricerca, partendo dall'inizio del buffer caratteri da codificare, la stringa più lunga presente anche nel buffer dei caratteri già codificati. Nella situazione presentata in fig.5, la stringa in comune ai due buffer è `<MAX`. Il codice generato è costituito da tre parti:

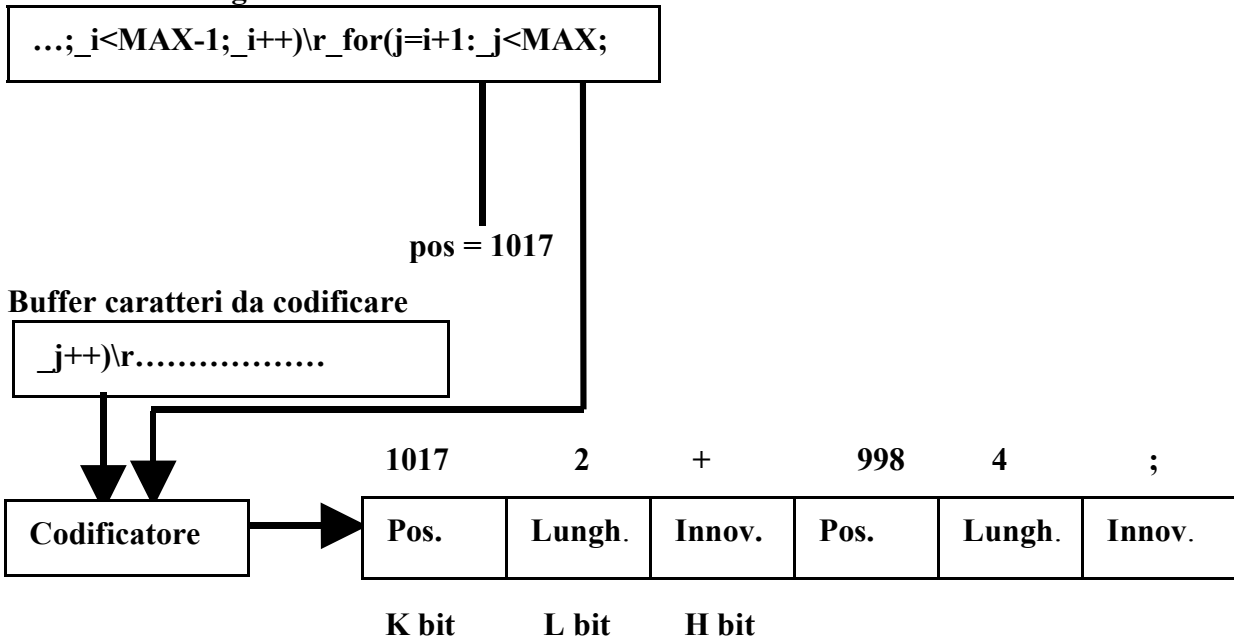
1. La posizione di inizio della stringa nel buffer dei caratteri già codificati (da 1 a 1023), rappresentato nell'esempio con $K = 10$ bit (ricordiamoci che $2^{10} = 1024$). Se non si individua alcuna stringa in comune tra i due buffer viene selezionato il codice 0.
2. Lunghezza della stringa selezionata. Nel nostro caso si utilizzano $L = 4$ bit.

Teoria dell'Informazione

3. Il carattere da codificare che segue la stringa selezionata. Viene chiamato Innovazione e rappresenta, come detto, il carattere successivo. Nel nostro esempio si utilizzano $H = 8$ bit per rappresentarlo.

Dopo aver generato in uscita l'insieme dei bit precedentemente indicato, la stringa selezionata viene spostata dal buffer dei caratteri da codificare al buffer dei caratteri già codificati. La necessità del carattere di innovazione è legata al fatto che senza questo elemento, se non vi fosse una stringa comune tra i due buffer di sistema, si creerebbe una situazione di stallo che bloccherebbe il sistema stesso.

In fig.6 viene presentata una nuova operazione di codifica. Si è shiftato verso destra.

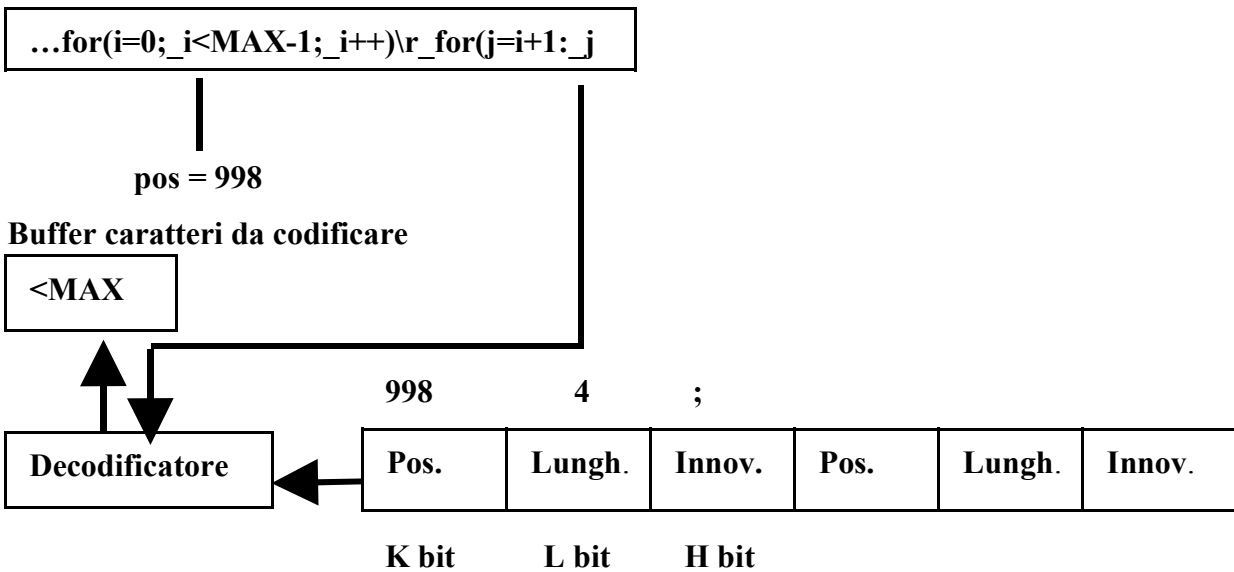
Buffer caratteri già codificati

Per esempio: $K = 10, L = 4, H = 8$

Pos. = 1...1023; 0 nel caso di nessuna stringa in comune.

Fig.6 – Una nuova stringa viene codificata utilizzando lo schema proposto in fig.5.

Il principio di funzionamento del sistema di decodifica viene presentato schematizzato in fig.7. Dalla sequenza di bit che rappresenta la stringa già decodificata si ritorna alla sequenza di caratteri originali. Ovviamente, per un buon funzionamento i due buffer dei caratteri già codificati e decodificati presenti nei due sistemi devono essere inizializzati allo stesso modo.

*Teoria dell'Informazione***Buffer caratteri già decodificati**

Per esempio: K = 10, L = 4, H = 8

Pos. = 1...1023; 0 nel caso di nessuna stringa in comune.

Fig.7 – Principio di funzionamento del sistema di decodifica associato al codificatore presentato nelle figure precedenti.

Se le stringhe, di volta in volta codificate, sono sufficientemente lunghe, l'efficienza del sistema può essere elevata. Infatti, nel caso di una stringa di 10 caratteri, al posto di $10 \cdot 8 = 80$ bit si utilizzano $10 + 4 + 8 = 22$ bit. Si vede subito che l'utilizzo del simbolo di innovazione seppur necessario è molto oneroso. Un possibile modo per superare il problema è quello di utilizzare un bit per identificare il fatto che sia presente o meno una stringa comune fra i due buffer. Avremo, pertanto, queste due possibili situazioni:

1. Esiste una stringa comune. Essa viene codificata come
 - Esiste la stringa campo da un bit) = 1;
 - Posizione della stringa (campo da K = 10 bit nel nostro caso);
 - Lunghezza della stringa (campo da L = 4 bit nel nostro esempio);
2. Non esiste la stringa comune. Viene generato:
 - Esiste la stringa campo da un bit) = 0;
 - Simbolo di innovazione (campo da H = 8 bit per noi).

In questo modo trasmettiamo il simbolo di innovazione solo quando strettamente necessario e, quindi, si aumenta l'efficienza del sistema. L'efficienza è sicuramente influenzata dalla scelta dei

Teoria dell'Informazione

valori di K e L, anche se tali parametri non potranno mai superare dei valori massimi legati alla velocità alla quale si vuole operare per individuare le stringhe comuni tra i due buffer.

Con la tecnica presentata si tende a rimuovere la correlazione presente fra simboli che si ripetono a breve distanza (devono essere presenti nel buffer dei caratteri codificati, altrimenti non né ho memoria). Supponiamo di voler codificare un elenco telefonico. Ovviamente cognomi uguali si susseguono a breve distanza, e la tecnica studiata è efficiente in quanto elimina la ridondanza del testo. Però, se si considerano i nomi delle vie presenti negli indirizzi degli utenti telefonici le cose cambiano. In una data via possono abitare svariate famiglie con cognomi diversi e che non si susseguono a breve distanza nell'elenco. In questo caso la ridondanza rimane. La domanda che ci dobbiamo porre è se è possibile ridurre tale ridondanza e se si come.

E' possibile risolvere il problema utilizzando un vocabolario più sofisticato rispetto a quello visto finora. Facciamo, ancora, un esempio di codifica di un testo alfanumerico con ogni simbolo rappresentato da un byte.

La codifica procede ricercando nel vocabolario il vocabolo più lungo che corrisponda ai caratteri iniziali della stringa da codificare, e ad esso si aggiunge ancora una volta il carattere di innovazione. La stringa complessiva, oltre ad essere codificata e scaricata dal buffer dei caratteri da codificare, diventa anche una nuova parola del vocabolario, che inizialmente contiene un solo vocabolo: la stringa vuota.

Indice	Vocabolo
0	“ “ Stringa vuota

Consideriamo il seguente testo da codificare:

DAD_DADA_DADDY_DADO.....

La codifica procede in questo modo:

Stringa codificata	Vocabolo utilizzato	Carattere Innovazione
“D”	0 (“ “)	‘D’
“A”	0 (“ “)	‘A’
“D_”	1 (“D”)	‘_’
“DA”	1 (“D”)	‘A’
“DA_”	4 (“DA”)	‘_’

Nel frattempo il vocabolario si evolve in questo modo:

Teoria dell'Informazione

Codice vocabolo	Vocabolo
0	“ ”
1	“D”
2	“A”
3	“D_”
4	“DA”
5	“DA_”

La dimensione del vocabolario aumenta sempre di più all'aumentare dei caratteri codificati. Inoltre, i vocaboli che vengono aggiunti diventano, mediamente, sempre più lunghi. Quando si raggiunge la massima dimensione del vocabolario si dovrà trovare un modo per sostituire i vecchi vocaboli o quelli meno usati con quelli nuovi. In ogni caso l'efficienza del sistema è garantita dal fatto che il numero di bit necessari a rappresentare l'indice del vocabolario è, normalmente, molto più piccolo del numero di bit necessari a rappresentare i vocaboli stessi.

Anche in questo caso il carattere di innovazione è necessario per evitare stalli, ma è molto oneroso. Per aumentare l'efficienza della codifica è possibile organizzare la codifica in modo che il carattere di innovazione sia considerato il carattere iniziale della stringa successiva. Inoltre, si può pensare che già all'inizio del processo, sia di codifica sia di decodifica, il vocabolario contenga il numero minimo di vocaboli necessari a permettere il funzionamento del sistema. Ad esempio si potrebbero inserire $2^8 = 256$ vocaboli (da 0 a 255) corrispondenti ciascuno ad uno dei possibili simboli elementari.

Se il testo da codificare è il seguente:

_WED_WE_WEE_WEB_WET.....

si ha:

Stringa codificata	Vocabolo utilizzato	Carattere innovazione (trasmesso come inizio stringa successiva)	Nuovo vocabolo inserito
“_W”	‘_’	‘W’	“_W” cod. 256
“WE”	‘W’	‘E’	“WE” cod. 257
“ED”	‘E’	‘D’	“ED” cod. 258
“D_”	‘D’	‘_’	“D_” cod. 259
“_WE”	256	‘E’	“_WE” cod. 260

Teoria dell'Informazione

Bisogna stare attenti a non usare immediatamente l'ultimo vocabolo inserito, in quanto esso è noto al codificatore ma non al decodificatore. Lo diventerà solo al ciclo successivo.

Codifica di sorgente

Lo scopo della codifica di sorgente è quello di ridurre la quantità di informazione generata dalla sorgente numerica per rendere minima la quantità di dati da inviare sul canale di trasmissione.

La codifica di sorgente può operare secondo due distinte modalità: codifica reversibile e codifica non reversibile.

Per la prima, bisogna diminuire le informazioni ridondanti presenti nei dati generati dalla sorgente. Ciò è possibile se si conosce la statistica dei messaggi emessi dalla sorgente stessa. Un tipico esempio è la codifica a lunghezza variabile dei messaggi, dove l'operazione di codifica è sempre reversibile. In questo caso si parla di riduzione della ridondanza oggettiva o di ridondanza statistica dei messaggi generati dalla sorgente. La ridondanza statistica associata ad una sorgente, sia con

memoria sia senza memoria, è definita come: $R = 1 - \frac{H}{H_{MAX}}$. H è ovviamente l'entropia della sorgente e H_{MAX} è il numero di bit utilizzati per la rappresentazione dei simboli elementari con codici a lunghezza fissa. Se i simboli sono M sarà $H_{MAX} = \log_2 M$. Per un testo scritto, ad esempio in inglese con 32 caratteri elementari: 26 lettere per l'alfabeto più spazio, virgola, due punti, punto e virgola, punto e punto interrogativo, $H_{MAX} = 5$ bit. Se supponiamo che la sorgente sia senza memoria e consideriamo solo le frequenze relative alle varie lettere, otteniamo $H \approx 4.03$ bit. Siccome, nella realtà, la sorgente avrà memoria e considerando la frequenza dei gruppi di tre lettere (trigrammi), si otterrà $H \approx 3.1$ bit. Estendendo la memoria a tutto il testo, esistono stime che assumono $H \approx 1$ bit. Pertanto, la ridondanza di un testo scritto è, a seconda delle ipotesi fatte, compresa tra $R \approx 0.194$ ed $R \approx 0.8$.

Nel caso della codifica non reversibile bisogna ridurre o eliminare del tutto le informazioni irrilevanti per l'utente finale presenti nei dati generati dalla sorgente. È facile capire come in questo modo si riducono o si eliminano le informazioni statisticamente ridondanti. Quello che possiamo ricostruire dopo la decodifica è un insieme di messaggi che per l'utente finale sia "percettivamente

Teoria dell'Informazione

indistinguibile” dall’insieme dei messaggi realmente generati dalla sorgente. Spesse volte vengono accettate delle degradazioni sulla qualità dei dati ricostruiti dopo la decodifica se a ciò corrisponde una diminuzione significativa della quantità di informazione da trasmettere. In questo caso si parlerà di ridondanza soggettiva o di irrilevanza. Sono tecniche usate normalmente per la codifica di segnali numerici ottenuti dal campionamento di segnali analogici come il segnale audio e soprattutto il segnale video, dove la riduzione notevole dei dati in gioco giustifica una codifica non reversibile. E’, anche, ovvio che le tecniche di riduzione della ridondanza soggettiva sono legate al tipo di segnale da trasmettere.

Canali di trasmissione

Iniziamo col dire che ogni canale di trasmissione continuo nel tempo può essere descritto con un equivalente canale discreto. Ci limitiamo, per semplicità, al caso di canali senza memoria, dove la descrizione statistica del canale stesso richiede la conoscenza di:

- un alfabeto X di ingresso, per esempio binario: $x_i = 0, 1$;
- un alfabeto Y di uscita, ad esempio un bit deciso: $y_j = 0, 1$;
- la probabilità di transizione fra ingresso e uscita; cioè la probabilità condizionata:
 $p(y_j/x_i)$.

Per valutare il comportamento del canale è utile conoscere le probabilità $P(x_i)$ e le analoghe probabilità di uscita $P(y_j)$, che derivano dalle probabilità di ingresso e dalla probabilità di transizione fra ingresso e uscita.

Alcuni semplici esempi di canali sono:

- canale binario simmetrico (BSC: Binary Symmetric Channel): alfabeto binario $\{0,1\}$ sia in ingresso sia in uscita; probabilità di errore $p(1/0) = p(0/1) = p$ indipendente dall’ingresso;
- canale gaussiano: l’uscita è la somma dell’ingresso e di una variabile casuale gaussiana con varianza σ^2 ; è il modello tipico di trasmissione in presenza di rumore additivo gaussiano bianco con ingresso binario; esula dal nostro studio;
- canale binario con cancellazione (BEC: Binary Erasure Channel): alfabeto binario in ingresso e ternario in uscita $\{0,1,E\}$ dove E indica incertezza completa; $p(E/0) = p(E/1) = p$;

Teoria dell'Informazione

$p(0/0) = p(1/1) = 1 - p$ in questo semplice modello si assume che sia gli zeri sia gli uni ricevuti siano sempre corretti.

Nel caso di canale binario generico senza alcun tipo di rumore con o senza cancellazione, le probabilità di uscita diventano:

$$\begin{cases} p(y_0) = p(y_0/x_0) \cdot p(x_0) + p(y_0/x_1) \cdot p(x_1) \\ p(y_1) = p(y_1/x_0) \cdot p(x_0) + p(y_1/x_1) \cdot p(x_1). \end{cases}$$

Per calcolare le probabilità di ingresso $p(x_0)$ e $p(x_1)$ bisogna moltiplicare il numero degli zeri e degli uni di ogni sequenza del codice per la propria probabilità e poi dividere per la lunghezza L del codice stesso. Da ricordare che a secondo del comportamento del canale nei confronti dei singoli bit, è possibile invertire i bit del codice.

Ogni canale trasmissivo non può essere ideale, ma è affetto da disturbi esterni ed interni. Infatti, il canale inserisce il rumore di fondo, il quale incide sulla qualità della trasmissione ed introduce anche attenuazione; questa attenua il segnale che deve pertanto essere rigenerato più volte. La scelta del mezzo di trasmissione deve essere legata al tipo di trasmissione ed al tipo di informazione da trasmettere.

Il rumore di cui sopra limita la capacità informativa del canale. Shannon ha individuato il valore del limite di questa capacità (C) in relazione alla larghezza di banda del canale (B), misurata in Hz, ed al grado di rumorosità del canale stesso $(S/N)^3$, affinché l'informazione possa essere trasmessa senza errori:

$$C = B \cdot \log_2 \left(1 + \frac{S}{N} \right) = 3,32 \cdot B \cdot \log_{10} \left(1 + \frac{S}{N} \right) \text{ [bit/s]}$$

E', ovvio, che un possibile miglioramento della capacità di canale si può ottenere o allargando la banda del canale o aumentando il rapporto segnale / rumore (S/N) .

I mezzi di trasmissione sono divisi in due grandi categorie: supporti fisici e ponti radio. I primi si chiamano anche mezzi di trasmissione ad onde guidate, perché vi è un collegamento fisico tra il lato trasmittente e quello ricevente. Invece, i secondi sono chiamati mezzi di trasmissione ad onde irradiate, in quanto l'informazione viene trasportata sotto la forma di onde elettromagnetiche che si propagano tra due antenne.

I supporti fisici più importanti sono i doppiini telefonici, i cavi coassiali, le guide d'onda, le fibre ottiche. I ponti radio, a loro volta, possono essere suddivisi in terrestri e spaziali (satelliti).

³ Con S si intende la potenza del segnale e con N la potenza del rumore.

Teoria dell'Informazione

Nella seguente tabella sono indicati, per i vari tipi di canali visti, la banda passante del canale, il rapporto segnale rumore e la capacità informativa di canale:

Tipi di canale	Banda passante [Hz]	S/N [dB]	capacità del canale[bit/s]
Doppino telefonico	4×10^3	30	40×10^3
Coppia coassiale (10800 canali)	60×10^6	40	800×10^6
Cavo a 100 coppie simmetriche	400×10^3	30	4×10^6
Fibre ottiche	10×10^9	80	265×10^9
Guide d'onda	2×10^9	60	40×10^9
Ponti radio (2700 canali)	12×10^6	50	200×10^6
(35000 canali) satelliti - Italsat	3×10^9	70	80×10^9

Esiste un legame tra la velocità di propagazione e la frequenza del segnale: $\lambda = v \cdot T = v/f$ [m], dove λ rappresenta la distanza percorsa da un'onda elettromagnetica in un periodo (T) e si chiama lunghezza d'onda; v è la velocità con cui tale onda si propaga nel canale ed f è la frequenza del segnale propagato. Se il mezzo è l'aria la velocità di propagazione è quella della luce.

Un simbolo non necessariamente binario viene trasmesso in un tempo generico pari a T_s . L'inverso di tale tempo mi rappresenta la velocità di trasmissione di simbolo (baud rate) misurata in baud o simboli/s: $r = 1/T_s$.

Invece il bit rate o information rate, che è la velocità di trasmissione di un singolo bit, è data da $R = r \cdot H(M)$ ed è misurato in bit/s. Il tempo di trasmissione di un singolo bit è pari all'inverso del bit rate.

E' facilmente visibile che se il simbolo è binario le due velocità coincidono.

Il rapporto segnale rumore è in generale dato in decibel (dB), pari a $10 \cdot \log_{10} \left(\frac{S}{N} \right)$