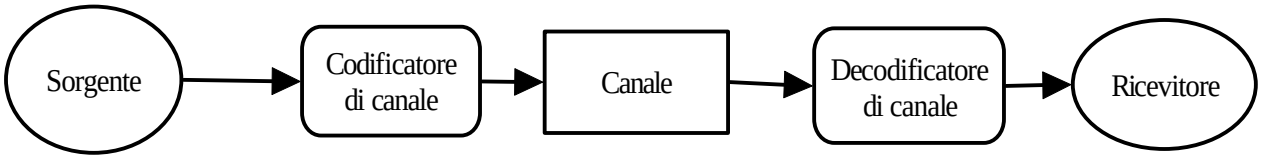
Appunti di Teoria dei Segnali

Capitolo 12 - Codifica di sorgenti discrete

Sorgenti senza memoria

INTRODUZIONE

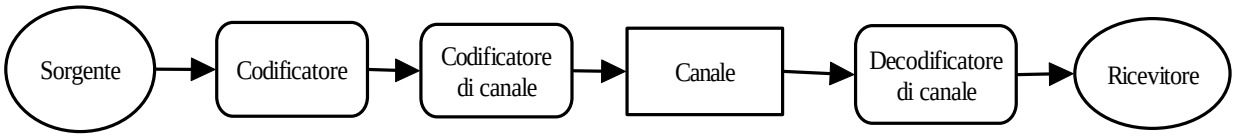
Quando abbiamo cominciato a parlare di *probabilità*, abbiamo avuto modo di dire qualche cosa a proposito del generico schema usato per la **trasmissione** di un segnale tra una **sorgente** ed un **ricevitore**:



Abbiamo descritto sinteticamente questo schema nel modo seguente: c'è intanto una "*sorgente*" che genera un segnale (binario) che deve arrivare al ricevitore (cioè all'utente); tale segnale non viene però trasmesso così com'è, ma viene in qualche modo "elaborato" da un opportuno dispositivo, che prende il nome di "*codificatore di canale*"; il segnale emesso dal codificatore viene inviato al "*canale binario*", il quale rappresenta tutti quei dispositivi necessari per la trasmissione del segnale stesso; in particolare, compito del canale è quello di far arrivare il segnale ad un altro dispositivo, il "*decodificatore di canale*", il quale, comportandosi in modo inverso al codificatore, ricostruisce il segnale così come è stato emesso dalla sorgente e lo invia al "*ricevitore*".

La necessità di **codificare** il segnale emesso dalla sorgente deriva da vari fattori, primo tra i quali i problemi legati al funzionamento del canale di trasmissione: infatti, trattandosi di un insieme di cavi e dispositivi fisici, è possibile che esso commetta degli "*errori*" nella trasmissione del segnale. Allora, il **codificatore di canale** viene progettato in modo che la sequenza di bit che corrisponde al segnale inviato dalla sorgente sia tale da permettere al decodificatore di canale di recuperare eventuali **errori** verificatisi durante la trasmissione.

Adesso, possiamo perfezionare quello schema, che come si vedrà anche in seguito, è estremamente approssimato, aggiungendo un ulteriore componente: si tratta del cosiddetto **codificatore**, ossia di un dispositivo che, dato il segnale emesso dalla sorgente, vi associa una corrispondente sequenza di bit (che costituisce appunto la **codifica** del segnale stesso) necessaria per la trasmissione del segnale stesso in forma digitale.

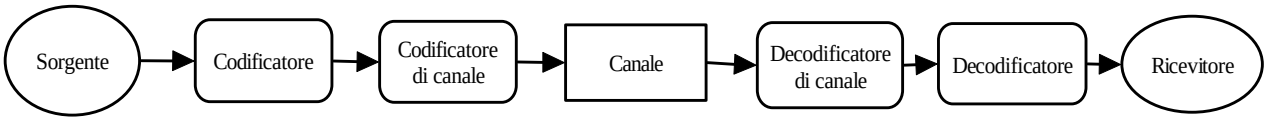


E' opportuno sottolineare le differenze di comportamento tra il "codificatore" ed il "codificatore di canale": il primo riceve in ingresso il segnale emesso dalla sorgente e genera in uscita la sequenza di bit che serve per la trasmissione di tale segnale (potremo perciò parlare di *convertitore analogico-digitale*, ma questa terminologia, in questo contesto, potrebbe essere fuorviante); il secondo, invece, riceve in ingresso la sequenza di bit emessa dal codificatore e opera su di essa quelle **manipolazioni** che permetteranno al decodificatore di canale di recuperare eventuali **errori** dovuti alla trasmissione non ideale tramite il canale.

Ovviamente, così come il decodificatore di canale si comporta in modo pressoché inverso al codificatore di canale, ci dovrà essere un ulteriore dispositivo che si comporta in modo inverso al

codificatore: tale dispositivo riceve in ingresso la sequenza di bit "ripulita", tramite il decodificatore di canale, degli errori dovuti alla trasmissione e, da questa sequenza, ricostruisce il segnale emesso dalla sorgente inviandolo al ricevitore.

Possiamo perciò concludere che lo schema più o meno completo di un sistema di trasmissione è il seguente:



DEFINIZIONE DI "SORGENTE DISCRETA"

Mentre in precedenza ci siamo occupati, molto velocemente, degli errori cui è soggetto il segnale codificato durante la trasmissione, quello di cui ci occupiamo adesso è il funzionamento del "*codificatore*": vogliamo cioè studiare quali sono i principali metodi di codifica del segnale. In particolare, ci limitiamo a considerare una situazione particolare, individuata dalle seguenti due caratteristiche fondamentali:

- in primo luogo, supponiamo che la codifica del segnale sia "**binaria**": ciò significa che al segnale emesso dalla sorgente, quale che esso sia, verrà sempre associata, da parte appunto del codificatore, una sequenza di bit;
- in secondo luogo, ci interessa il tipo di **sorgente**: innanzitutto, supponiamo che questa sorgente non emetta in modo continuo nel tempo il proprio segnale, ma solo in istanti successivi discreti; in secondo luogo, supponiamo anche che il segnale emesso dalla sorgente sia costituito da una successione di "simboli" discreti appartenenti al cosiddetto "*alfabeto*" della sorgente stessa. Per capirci meglio, se supponiamo che l'alfabeto della sorgente sia genericamente $X = \{x_1, x_2, \ldots, x_N\}$, l'ipotesi che stiamo facendo, che prende il nome di ipotesi di **sorgente discreta**, è che la sorgente emetta, in istanti di tempo discreti e uno alla volta, i simboli appartenenti ad X (in un ordine che dipende dal tipo di informazioni che si intende trasmettere).

Un'altra ipotesi semplificativa con la quale lavoriamo è che i simboli emessi dalla sorgente siano indipendenti tra loro: ciò significa che ogni simbolo emesso è indipendente dai simboli emessi in precedenza. In termini quantitativi, possiamo esprimerci dicendo che la probabilità di emettere un simbolo ad un certo istante è sempre la stessa, a prescindere da quali simboli siano stati emessi in precedenza. Una sorgente che gode di questa proprietà prende il nome di **sorgente senza memoria**.

Indichiamo allora con $P(x_i)$ la probabilità che la sorgente emetta il generico simbolo $\mathbf{x_i}$: se supponiamo che l'alfabeto della sorgente comprenda N simboli (con N finito o infinito numerabile), avremo allora N **probabilità di trasmissione**

$$P(x_1), P(x_2), .., P(x_N)$$

E' ovvio che queste probabilità di trasmissione siano legate dalla relazione

$$\sum_{i=1}^{N} P(x_i) = 1$$

E' chiaro infatti che la sorgente può emettere o il simbolo x_1 o il simbolo x_2 e così via, per cui la probabilità di emettere almeno uno dei simboli dell'alfabeto (ossia il termine a primo membro di quella relazione) è pari ad 1.

CONCETTO DI "INFORMAZIONE"

Ogni simbolo emesso dalla sorgente costituisce chiaramente una "*informazione*" che la sorgente emette perché raggiunga il ricevitore. Il tipo di informazione è assolutamente generico, nel senso che la sorgente può trasmettere informazioni di qualsiasi tipo. Dovendo però studiare il funzionamento della sorgente e del codificatore, è utile rappresentare in qualche modo tale informazione dal punto di vista matematico. Dato allora il generico simbolo x_i, possiamo definire, in termini matematici, l'**informazione** ad essa associata nel modo seguente:

$$\boxed{x_i \longrightarrow I(x_i) = \log_2 \frac{1}{P(x_i)}}$$

Si tratta adesso di giustificare questa formula, ossia di vedere perché l'informazione è stata espressa in questo modo. Le giustificazioni che possiamo dare sono tutte di natura essenzialmente intuitiva.

Per esempio, supponiamo che la sorgente in esame sia una sorgente che ha un solo simbolo e quindi trasmette solo quello: è evidente, allora, che questa sorgente, di fatto, non trasmette alcuna informazione, proprio perché emette sempre la stessa cosa. In termini matematici, questo significa che l'informazione associata all'unico simbolo deve valere 0 e questo effettivamente accade in base alla definizione di $I(x_i)$: infatti, se la sorgente possiede un solo simbolo, è ovvio che $P(x_i) = 1$ (in quanto c'è la certezza che emetta quell'unico simbolo); andando allora a sostituire nell'espressione di $I(x_i)$, si trova che quest'ultima è proprio pari a 0.

Ora supponiamo di considerare due generici simboli x_i e x_j tra quelli appartenenti all'alfabeto della sorgente; supponiamo anche che il primo abbia meno probabilità del secondo di essere trasmesso, ossia supponiamo che $P(x_i) < P(x_j)$. Questo significa che, dato un qualsiasi messaggio emesso dalla sorgente, il simbolo x_i compare in media MENO volte rispetto al simbolo x_j; se compare meno volte, è lecito aspettarsi che l'informazione associata a tale simbolo sia PIU' importante di quella associata al simbolo x_j, ossia è lecito aspettarsi che sia $I(x_i) > I(x_j)$. Effettivamente questo accade: infatti, poiché $P(x_i) < P(x_j)$, andando a sostituire nella definizione di $I(x_i)$, noi troviamo che

$$I(x_i) = \log_2 \frac{1}{P(x_i)} > I(x_j) = \log_2 \frac{1}{P(x_j)}$$

Un'ultima giustificazione della definizione di $I(x_i)$ potrebbe essere la seguente: supponiamo che la sorgente emetta i due simboli x_i e x_j in modo del tutto indipendente dall'altro (come noi stiamo supponendo). Allora, noi siamo portati a credere che l'informazione complessiva associata alla coppia (x_i, x_j) corrisponda alla somma delle informazioni. Anche in questo caso, questo si verifica: infatti, indicata con

$$I(x_i, x_j) = \log_2 \frac{1}{P(x_i \cap x_j)}$$

l'informazione associata alla coppia, si ha che

$$I(x_i, x_j) = \log_2 \frac{1}{P(x_i)P(x_j)} = \log_2 \frac{1}{P(x_i)} \frac{1}{P(x_j)} = \log_2 \frac{1}{P(x_i)} + \log_2 \frac{1}{P(x_j)} = I(x_i) + I(x_j)$$

ENTROPIA DELLA SORGENTE

Consideriamo adesso la nostra sorgente discreta X e adottiamo sempre l'ipotesi per cui ciascun simbolo emesso sia indipendente dai simboli emessi precedentemente. Si definisce allora **entropia** della sorgente la quantità

$$\boxed{H(X) = \sum_{i=1}^{N} P(x_i) I(x_i)}$$

dove ovviamente N è il numero di simboli che la sorgente è in grado di emettere, $P(x_i)$ è la probabilità di emettere il generico simbolo x_i e $I(x_i)$ è l'informazione associata a tale simbolo.

Possiamo trovare una espressione più precisa dell'entropia, considerando la definizione data dell'informazione: infatti, usando la relazione $I(x_i) = \log_2 \frac{1}{P(x_i)}$, abbiamo che

$$H(X) = \sum_{i=1}^{N} P(x_i) \log_2 \frac{1}{P(x_i)}$$

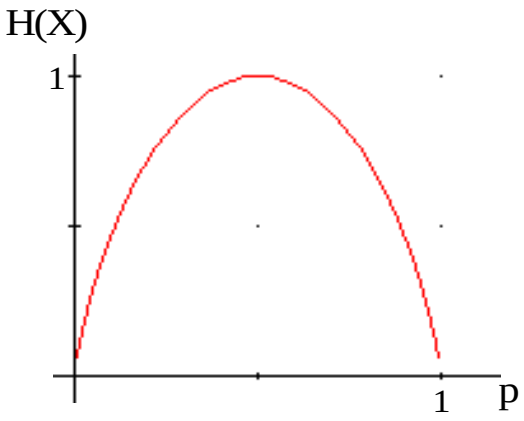
Al fine di semplificare un po' le nostre notazioni, possiamo porre $p_i = P(x_i)$, per cui l'entropia diventa

$$\boxed{H(X) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}}$$

Il valore numerico dell'entropia della sorgente dipende dunque dalle probabilità di emissione della sorgente stessa e dal numero di simboli di cui la sorgente dispone; per esempio, supponiamo che la nostra sorgente abbia solo due simboli, per cui N=2; dato che la somma delle probabilità di emissione deve essere pari ad 1, è chiaro che, se p è la probabilità di emettere uno dei due simboli, 1-p sarà la probabilità di emettere l'altro simbolo. Allora, il valore dell'entropia risulta essere il seguente:

$$H(X) = \sum_{i=1}^{2} p_i \log_2 \frac{1}{p_i} = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

Evidentemente, al variare di p avremo valori diversi di entropia. Possiamo perciò provare a diagrammare H(X) in funzione di p (dove p varia ovviamente tra 0 ed 1 essendo una probabilità): ciò che si ottiene è



Si nota quindi che l'entropia assume il suo massimo valore quando p=0.5, ossia quando i due simboli di cui dispone la sorgente sono **equamente probabili**, mentre assume decrescenti, in modo simmetrico, quando p aumenta o cresce.

Entropia e numero di simboli dell'alfabeto sorgente

L'entropia è una grandezza importante per una serie di motivi che illustreremo: il primo di questi è che essa è legata al numero N di simboli di cui dispone la sorgente dalla relazione

$$\boxed{H(X) \leq \log_2 N}$$

Vogliamo dimostrare la veridicità di questa formula e, in particolare, vogliamo far vedere che il simbolo di uguaglianza vale SOLO quando tutti i simboli sono **equiprobabili**.

Dimostrare quella relazione equivale ovviamente a dimostrare che

$$H(X) - \log_2 N \leq 0$$

Per prima cosa, possiamo sostituire ad H(X) l'espressione trovata prima:

$$H(X) - \log_2 N = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} - \log_2 N$$

Ora, ricordandoci della proprietà secondo cui $\sum_{i=1}^{N} p_i = 1$, possiamo riscrivere quella relazione nella forma

$$H(X) - \log_2 N = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} - \log_2 N \sum_{i=1}^{N} p_i$$

Il termine $\log_2 N$ non dipende dall'indice i della sommatoria, per cui possiamo portarlo dentro di essa:

$$H(X) - \log_2 N = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} - \sum_{i=1}^{N} p_i \log_2 N$$

Adesso, il secondo membro può essere posto tutto sotto un'unica sommatoria:

$$H(X) - \log_2 N = \sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{p_i} - \log_2 N \right)$$

Inoltre, usando una nota proprietà dei logaritmi, possiamo anche scrivere che

$$H(X) - \log_2 N = \sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{Np_i} \right)$$

A questo punto, se supponiamo che tutti i simboli della sorgente sono uguali, ossia supponiamo che $p_i=p$, quella relazione diventa

$$H(X) - \log_2 N = \sum_{i=1}^{N} \left(p \log_2 \frac{1}{Np} \right) = p \log_2 \frac{1}{Np} \sum_{i=1}^{N} 1 = Np \log_2 \frac{1}{Np}$$

Tuttavia, dire che gli N simboli sono equiprobabili significa dire che p=1/N, per cui

$$H(X) - \log_2 N = N \frac{1}{N} \log_2 \frac{N}{N} = 0$$

Abbiamo dunque dimostrato che, quando i simboli della sorgente sono tutti equiprobabili, vale la relazione

$$H(X) = \log_2 N$$

Resta perciò da far vedere che, nel caso generale di simboli con probabilità di emissione diverse, vale la relazione

$$H(X) < \log_2 N$$

ossia, sostituendo l'espressione della H(X) e riprendendo i calcoli fatti nei passaggi precedenti, la relazione

$$\sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{Np_i} \right) < 0$$

Per dimostrare questa relazione possiamo ricorrere a due proprietà dei logaritmi:

- la prima si ottiene usando lo sviluppo in serie di Taylor e dice che

$$\ln y \le y - 1$$

- la seconda riguarda invece il cambiamento di base e dice che

$$\log_2 y = (\ln e) \ln y$$

Autore: Sandro Petrizzelli

Mettendole insieme, troviamo che

$$\log_2 y = (\ln e)\ln y \le (\ln e)(y-1)$$

Applicando questa proprietà, possiamo scrivere che

$$\log_2 \frac{1}{Np_i} \le (\ln e)\left(\frac{1}{Np_i} - 1\right)$$

Posto per comodità $\alpha = (\ln e)$, questa diventa

$$\log_2 \frac{1}{Np_i} \le \alpha\left(\frac{1}{Np_i} - 1\right)$$

per cui possiamo scrivere che

$$\sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{Np_i}\right) \le \sum_{i=1}^{N} \alpha p_i \left(\frac{1}{Np_i} - 1\right) = \alpha \sum_{i=1}^{N} p_i \left(\frac{1}{Np_i} - 1\right)$$

Sdoppiando in due diverse sommatorie, abbiamo che

$$\sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{Np_i}\right) \le \alpha \sum_{i=1}^{N} \frac{1}{N} - \alpha \sum_{i=1}^{N} p_i$$

A questo punto, sappiamo che $\sum_{i=1}^{N} p_i = 1$, per cui

$$\sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{Np_i}\right) \le \alpha \sum_{i=1}^{N} \frac{1}{N} - \alpha$$

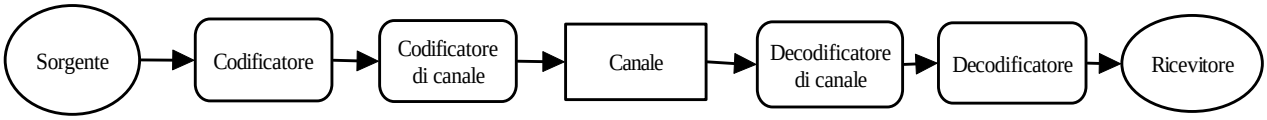
Inoltre, la sommatoria rimasta è pari anch’essa proprio ad 1 (in quanto il termine da sommare è 1/N e non dipende dall’indice di sommatoria e quindi la sommatoria stessa vale proprio N), dal che si deduce che

$$\sum_{i=1}^{N} p_i \left(\log_2 \frac{1}{Np_i}\right) \le 0$$

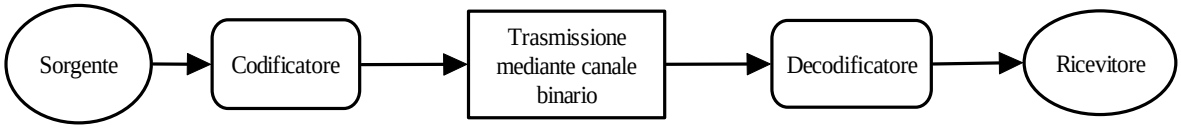
La fase di codifica del segnale

INTRODUZIONE

Riprendiamo lo schema del sistema di trasmissione introdotto all'inizio del capitolo:



Dato che ci siamo occupati già in precedenza della fase inerente la trasmissione del segnale, quindi dei passaggi che vanno dalla codifica di canale alla decodifica di canale, possiamo semplificare questo schema nel modo seguente:



Dobbiamo adesso esaminare l'aspetto riguardante la codifica e la decodifica del segnale emesso dalla sorgente.

Abbiamo detto che *il codificatore ha il compito di associare, alla sequenza di simboli emessi dalla sorgente, una opportuna sequenza di bit.* E' ovvio che questa sequenza di bit deve essere tale che, una volta arrivata al decodificatore, quest'ultimo sia in grado di ricostruire esattamente, a partire da essa, il segnale emesso dalla sorgente. Questo è possibile solo se il codice binario utilizzato gode della proprietà di essere **univocamente decodificabile**: deve cioè essere tale che la decodifica possa essere una ed una sola, in modo da avere la garanzia che il ricevitore riceva ciò che la sorgente ha emesso.

E' ovvio che i codici ai quali si può pensare sono infiniti. In linea di massima, i requisiti che un codice deve avere, oltre appunto alla "univoca decodificabilità", sono i seguenti:

- deve prevedere il più basso numero di bit possibile, in modo da ridurre i tempi di trasmissione;
- deve inoltre consentire di "risparmiare in banda".

A requisiti di questo genere rispondono sia i codici "a lunghezza fissa" sia soprattutto quelli "a lunghezza variabile". Che cosa si intende con queste espressioni? Un **codice a lunghezza fissa** è un codice in cui ad ogni simbolo viene associato sempre lo stesso numero di bit: un esempio tipico è la codifica ASCII dei caratteri. Un codice **a lunghezza variabile** è invece evidentemente un codice in cui il numero di bit associati a ciascun simbolo è variabile.

Esempio di codice a lunghezza variabile

Facciamo subito un semplice esempio di codice a lunghezza variabile: supponiamo che l'alfabeto della nostra sorgente consti di N=4 soli simboli e in particolare sia $\{A, B, C, D\}$; un possibile codice a lunghezza variabile per questo alfabeto potrebbe essere il seguente:

$$A \longrightarrow 0$$
$$B \longrightarrow 01$$
$$C \longrightarrow 10$$
$$D \longrightarrow 100$$

Allora, se per esempio il messaggio emesso dalla sorgente fosse

A C B A

la codifica di sorgente sarebbe

0 10 01 0

Quindi, il decodificatore riceve in ingresso questa sequenza di bit e deve essere in grado di ricostruire da essa la sequenza A C B A. Si intuisce allora subito quale sia il problema di fondo dei codici a lunghezza variabile: dato che i bit arrivano uno dopo l’altro intervallati della stessa quantità e senza alcun “separatore” tra di essi, ossia senza che vengano indicati l’inizio e la fine di ogni blocco, come fa il decodificatore a stabilire se un certo bit va incluso nel gruppo di bit precedenti, oppure va preso da solo oppure va incluso nel gruppo di bit successivi?

Per capirci meglio, vediamo cosa succede quando il decodificatore riceve la sequenza 0 10 01 0: il primo bit è 0, per cui il decodificatore deve stabilire se questo 0 va considerato da solo, per cui corrisponde alla codifica di A, oppure va considerato insieme all’ 1 che viene dopo, per cui costituisce parte della codifica di B. Con un codice fatto in questo modo, il decodificatore non ha alcun elemento per scegliere una delle due possibilità, dal che si deduce che quel codice NON è univocamente decodificabile. Il risultato della decodifica può essere la sequenza corretta A C B A, ma può anche essere la sequenza sbagliata B A B A.

METODO AD ALBERO PER I CODICI UNIVOCAMENTE DECODIFICABILI

Dall’esempio appena esaminato sorge subito a seguente domanda: come si fa ad ottenere un codice a lunghezza variabile che sia univocamente decodificabile?

Un risultato importante che si può dimostrare a questo proposito è il seguente: *condizione sufficiente affinché un codice a lunghezza variabile sia univocamente decodificabile è che ogni parola del codice NON sia il prefisso di alcuna altra parola.*

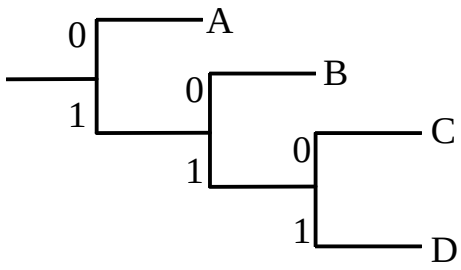
A prescindere dalla dimostrazione matematica, è intuitivo comprendere perché sussiste questo risultato: basta ad esempio considerare il motivo per cui non risultava univocamente decodificabile il codice binario esaminato nell’esempio precedente: tale codice era

$$A \longrightarrow 0$$
$$B \longrightarrow 01$$
$$C \longrightarrow 10$$
$$D \longrightarrow 100$$

per cui i possibili errori venivano dal fatto che un bit 0 può essere preso come il prefisso della parola associata ad A oppure di quella associata a B, oppure la sequenza 10 poteva essere presa come prefisso per C o come prefisso per D.

Naturalmente, trattandosi di una condizione sufficiente, è possibile che siano univocamente decodificabili codici che non abbiano quel requisito.

Viene poi da chiedersi come si possa ideare un codice binario che soddisfi a quella condizione. Il metodo migliore consiste nell'usare la tecnica ad albero: supponiamo perciò che l'alfabeto della nostra sorgente sia ancora una volta $\{A, B, C, D\}$. Tracciamo allora uno **schema ad albero** del tipo seguente:



Leggendo quest'albero da sinistra verso destra, otteniamo il seguente codice:

$$A \longrightarrow 0$$
$$B \longrightarrow 10$$
$$C \longrightarrow 110$$
$$D \longrightarrow 111$$

Questo codice, che è evidentemente a lunghezza variabile, gode della proprietà per cui nessuna è parola è prefisso di qualche altra, per cui noi siamo certi che si tratti di un codice univocamente decodificabile.

Vediamo allora come si comporta il decodificatore: supponiamo che il messaggio sia

A B D A C

La codifica binaria di questo messaggio è

0 10 111 0 110

Il primo bit ricevuto dal decodificatore è lo 0: lo 0 compare solo nella parola corrispondente ad A ed è anche l'unico termine di tale parola, per cui il decodificatore genera subito il simbolo A. Il bit successivo è 1: questo può essere il primo bit delle parole corrispondenti a B,C e D, per cui è necessario considerare il bit successivo; questo è uguale a 0: la sequenza 10 compare solo nella parola corrispondente a B, per cui il decodificatore genera immediatamente B.

Il bit ancora successivo è 1: anche qui vale lo stesso discorso di prima, per cui bisogna considerare il bit successivo, che vale ancora 1; la sequenza 11 compare come prefisso delle parole associate a C e a D, per cui bisogna considerare il bit successivo, che vale sempre 1: a questo punto, la possibilità è una sola e cioè che tali tre bit corrispondano al simbolo D che quindi viene emesso. Continuando in questo modo, viene ricostruito con esattezza il messaggio emesso dalla sorgente.

NUMERO MEDIO DI BIT

Il *metodo ad albero* appena esposto è uno dei metodi con i quali si può realizzare un codice, a lunghezza fissa o variabile, che sia certamente univocamente decodificabile. Viene allora da chiedersi, dati due codici a lunghezza variabile univocamente decodificabili, quale convenga

scegliere. Sicuramente, *il parametro più importante per la scelta è il numero di bit che il codice associa ai simboli.* In particolare, dato che si tratta di codici a lunghezza variabile, il parametro da considerare è il **numero medio di bit** associati ai simboli del codice. Questo numero medio può essere definito nel modo seguente:

$$\overline{N} = \sum_{i=1}^{N} N_i p_i$$

dove N_i è il numero di bit che il codice associa al simbolo i-simo, mentre $p_i = P(x_i)$ è la probabilità che la sorgente emetta tale simbolo (e ricordiamo a questo proposito che siamo sempre nella ipotesi che tale probabilità sia indipendente dai simboli emessi in precedenza, ossia nella ipotesi di sorgente senza memoria).

Il criterio di scelta è allora il seguente: *quanto più piccolo è $\overline{N}$, tanto migliore è il codice.*

Infatti, quanto più piccolo è $\overline{N}$, tanto minore è il numero di bit da trasmettere e quindi tanto minore è il tempo richiesto per la trasmissione del messaggio completo. Per esempio, se supponiamo che la nostra sorgente emetta un messaggio composto da k simboli, il numero medio di bit da trasmettere sarà $k\overline{N}$.

Facciamo osservare, come sarà chiaro anche dagli esempi, che anche piccole variazioni di $\overline{N}$ sono comunque importanti: ad esempio, supponiamo che il messaggio da trasmettere sia composto da k=10000 simboli; allora il numero medio di simboli da trasmettere è $10000\overline{N}$; se $\overline{N} = 2.5$, tale numero medio vale 25000, mentre, se $\overline{N} = 2.8$, esso vale 28000. Abbiamo perciò una differenza di 3000 simboli che non è da niente, specialmente poi se la velocità del canale binario non è elevatissima.

LA CODIFICA DI HUFFMAN

Da quanto detto è intuitivo accorgersi quanto sia opportuno trovare un modo per ideare un codice a lunghezza variabile che abbia il minimo valore possibile di $\overline{N}$. Questo problema è stato studiato e ottimizzato mediante una tecnica di codifica che prende il nome di **codifica di Huffman**. Questa tecnica usa uno *schema ad albero* del tipo visto prima, ma in modo più particolareggiato e questo al fine proprio di ottenere il minimo valore possibile per $\overline{N}$.

Supponiamo ancora una volta che l'alfabeto della nostra sorgente sia composto da N=4 simboli e precisamente $\{A, B, C, D\}$. Supponiamo anche di conoscere le probabilità p_i di trasmissione di tali simboli: ad esempio, prendiamo

$$p_A = 0.1$$
$$p_B = 0.3$$
$$p_C = 0.4$$
$$p_D = 0.2$$

Per costruire l'albero, cominciamo a disporre i 4 simboli dell'alfabeto, con le rispettive probabilità, in ordine di probabilità decrescente, ossia partendo dal più probabile e andando verso il meno probabile: nel nostro caso, la scala risulta essere

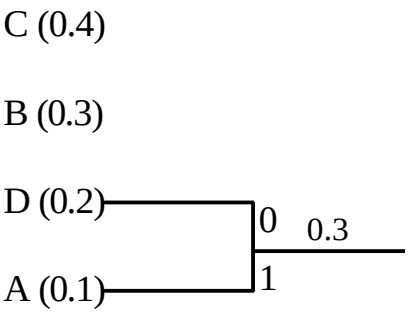
C (0.4)

B (0.3)

D (0.2)

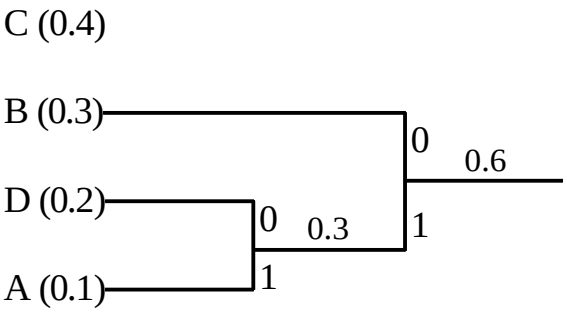
A (0.1)

Adesso, consideriamo i due simboli con probabilità minore, che si troveranno evidentemente al fondo della scala, e cominciamo a costruire l'albero associando il bit 0 a quello più probabile ed il bit 1 a quello meno probabile: abbiamo dunque

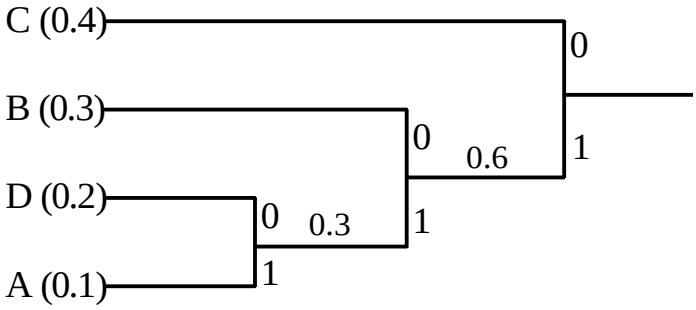


A questo punto, consideriamo questi due simboli come un unico simbolo avente probabilità di trasmissione pari alla somma delle probabilità, che in questo caso 0.3.

Il passo successivo consiste nel ripetere il ragionamento considerando i simboli non ancora esaminati e il simbolo corrispondente agli altri due (con relativa possibilità). Considerando sempre i due simboli con probabilità minore e associando 0 a quello più probabile ed 1 all'altro, abbiamo che



A questo punto rimangono solo due simboli e quindi l'albero può essere completato:



A partire da quest'albero, siamo adesso in grado di ottenere il **codice di Huffman** della sorgente considerata leggendo l'albero stesso a partire dalla *radice*, ossia da destra verso sinistra. Il codice che si ottiene è il seguente:

$$C \longrightarrow 0$$
$$B \longrightarrow 10$$
$$D \longrightarrow 110$$
$$A \longrightarrow 111$$

Una osservazione importante che possiamo fare è che, data una coppia di simboli, non è assolutamente obbligatorio associare 0 a quello più probabile e 1 all'altro. E' possibile anche invertire, a patto però di farlo sempre. Il codice che risulta alla fine è chiaramente diverso, ma gode delle stesse caratteristiche.

Andiamo a calcolare il valore di $\overline{N}$ mediante la semplice definizione:

$$\overline{N} = \sum_{i=1}^{N} N_i p_i = \underbrace{3*0.1}_{A} + \underbrace{2*0.3}_{B} + \underbrace{1*0.4}_{C} + \underbrace{3*0.2}_{D} = 1.9$$

Confrontiamo adesso questo valore con quello che si ottiene se, mantenendo le stesse probabilità di emissione, venisse usato il codice ricavato in precedenza: si trattava del codice

$$A \longrightarrow 0$$
$$B \longrightarrow 10$$
$$C \longrightarrow 110$$
$$D \longrightarrow 111$$

(anch'esso univocamente decodificabile) e, facendo i calcoli, si trova un valore del numero medio di bit pari a 2.5.

Abbiamo dunque trovato una conferma del seguente principio fondamentale: *codificando una sorgente discreta senza memoria con il codice di Huffman, si ottiene il minimo valore possibile del numero medio di bit associati a ciascun simbolo.*

Questo comporta un altro importante risultato: supponiamo di avere un certo alfabeto e di ideare, con metodo qualsiasi, un codice per tale alfabeto che risulti univocamente decodificabile; ci andiamo poi a calcolare il valore di $\overline{N}$ relativo a tale codice: se troviamo, secondo criteri che saranno esposti tra poco, che questo valore è quello minimo possibile, allora potremo star certi che il codice ideato corrisponde a quello di Huffman, ossia a quello ottenibile con il metodo prima descritto.

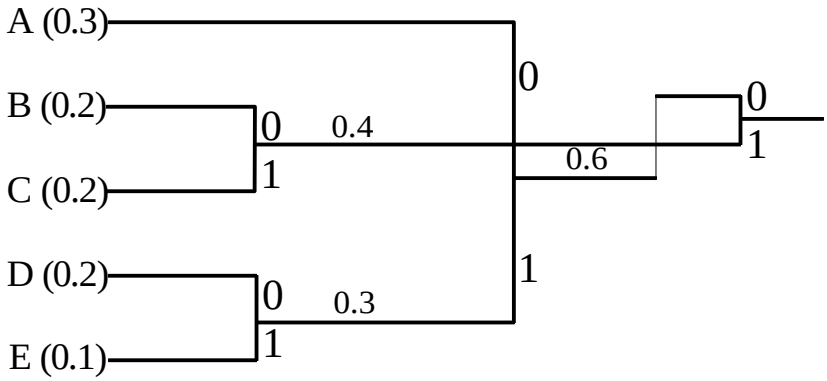
Esempio di codifica di Huffman

Supponiamo che l'alfabeto della nostra sorgente sia composto da N=5 simboli e precisamente $\{A, B, C, D, E\}$. Supponiamo poi che le probabilità p_i di trasmissione di tali simboli siano le seguenti:

$$p_A = 0.3$$
$$p_B = 0.2$$
$$p_C = 0.2$$
$$p_D = 0.2$$
$$p_E = 0.1$$

(si noti che la somma di quelle probabilità deve sempre essere =1 e che i simboli sono già stati ordinati per probabilità di emissione decrescente).

Troviamo la codifica di Huffman di quell'alfabeto: il primo passo è la costruzione dell'albero, che è



Leggendo l'albero a partire dalla radice ricaviamo il codice:

$$A \longrightarrow 00$$
$$B \longrightarrow 10$$
$$C \longrightarrow 11$$
$$D \longrightarrow 010$$
$$E \longrightarrow 011$$

Andando ora a calcolare quanto vale $\overline{N}$, troviamo 3.5.

NUMERO MEDIO DI BIT ED ENTROPIA DELLA SORGENTE

Abbiamo detto che è sempre buona norma ideare un codice, sia esso a lunghezza variabile o a lunghezza fissa, che abbia il più piccolo valore possibile per $\overline{N}$. Abbiamo anche detto, senza dimostrarlo, che dato un certo alfabeto, la codifica secondo Huffman è quella che presenta sempre il minimo valore di $\overline{N}$. Sorge allora questa domanda: supponiamo di avere un certo alfabeto, di idearne un codice e di calcolarne il valore di $\overline{N}$; come facciamo a stabilire se questo valore sia o meno il più piccolo possibile? Ossia, in altre parole, dato un generico codice, come si fa a calcolare il valore minimo di $\overline{N}$?

Avendo detto che la codifica di Huffman presenta sempre il valore minimo di $\overline{N}$, un modo potrebbe anche quello di trovare in ogni caso tale codifica e di andarsi a calcolare $\overline{N}$. Tuttavia, è

chiaro che si tratta di un metodo tutt'altro che agevole, specialmente considerando che il numero N di simboli di cui è costituito l'alfabeto di una sorgente è generalmente molto alto.

Allora, per risolvere il problema, viene in aiuto il seguente risultato:

$$\boxed{H(X) \le \overline{N} \le H(X) + 1}$$

Esso dice quindi che *il numero medio di bit che il codice associa a ciascun simbolo dell'alfabeto non può essere mai inferiore all'entropia della sorgente,* ossia a quella quantità definita come

$$H(X) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}$$

In altre parole, H(X) è l'estremo inferiore dell'insieme dei possibili valori di $\overline{N}$. Da sottolineare il concetto di "estremo inferiore": solo in alcuni casi molto particolari, $\overline{N}$ può raggiungere il valore di H(X), mentre in generale è sempre leggermente maggiore.

Di conseguenza, quando dobbiamo codificare una certa sorgente, della quale siano note le probabilità di emissione (sempre nell'ipotesi di indipendenza tra i simboli), ci possiamo calcolare l'entropia della sorgente, in modo da sapere il valore minimo di $\overline{N}$ al quale noi dobbiamo tendere nell'ideare il codice.

Esempio

Vediamo subito un esempio in cui il numero medio $\overline{N}$ coincide proprio con l'entropia della sorgente.

Supponiamo che l'alfabeto della sorgente sia $\{A, B, C, D\}$ e che le probabilità di emissione sia tutte uguali e pari quindi ad 1/4.

Calcoliamo subito l'entropia della sorgente:

$$H(X) = \sum_{i=1}^{4} p_i \log_2 \frac{1}{p_i} = \sum_{i=1}^{4} \frac{1}{4} \log_2 4 = \sum_{i=1}^{4} \frac{1}{4} 2 = 2$$

Se l'entropia è pari a 2, 2 sarà anche il valore minimo che noi potremo trovare per $\overline{N}$.

Andiamo allora a definire il codice da utilizzare: avendo 4 diversi simboli da codificare, la codifica che viene in mente per prima è certamente quella binaria, ossia

$$A \longrightarrow 00$$
$$B \longrightarrow 01$$
$$C \longrightarrow 10$$
$$D \longrightarrow 11$$

Evidentemente, essa ha $\overline{N} = 2$, ossia ha un valore di $\overline{N}$ coincidente con il valore della entropia. Ciò significa che questa codifica è proprio quella di Huffman. La verifica di questo può essere fatta applicando il metodo standard per la determinazione del codice di Huffman.

La codifica a blocchi

INTRODUZIONE

Abbiamo più volte detto, negli ultimi paragrafi, che, *nell'ideare un codice con cui codificare l'alfabeto di una certa sorgente, l'obbiettivo primario da raggiungere, oltre la univoca decodificabilità del codice, è quello di ottenere il valore più piccolo possibile per* $\overline{N}$. Abbiamo anche visto che il valore minimo cui si può aspirare (anche se difficilmente può essere raggiunto) corrisponde al valore dell'entropia H(X) della sorgente, valore che si raggiunge con la codifica di Huffman.

Ci poniamo allora il problema seguente: vogliamo ideare un codice, che non sia quello di Huffman, il cui valore di $\overline{N}$, pur essendo maggiore del valore minimo H(X), sia comunque il più vicino possibile ad H(X) stesso.

Supponiamo, per comodità di ragionamento, che l'alfabeto sorgente sia $\{A, B, C\}$. Di conseguenza, ogni messaggio emesso dalla sorgente sarà una successione di tali simboli. Un esempio potrebbe essere il seguente:

B A C C B A A A C B

Una sequenza di questo tipo può essere pensata in due modi differenti:

- il primo è appunto quello di vederla come una sequenza di simboli singoli emessi dalla sorgente X;
- un altro modo è invece quello di vederla come una sequenza di **COPPIE di simboli** emessi da una certa sorgente Y.

Possiamo cioè immaginare quel messaggio come prodotto dalla sorgente Y il cui alfabeto sia il seguente:

$$\{(A,A),(A,B),(A,C),(B,A),(B,B),(B,C),(C,A),(C,B),(C,C)\}$$

Si tratta cioè di un alfabeto in cui ogni simbolo corrisponde ad una delle possibili coppie di simboli che si possono formare con l'alfabeto della sorgente X. Chiaramente, dato che X ha 3 simboli, la sorgente Y avrà 3*3=9 simboli.

Naturalmente, dato che noi conosciamo i simboli della sorgente X, conosciamo anche i simboli della sorgente Y, per cui possiamo fare su di essa tutti i discorsi relativi ad una generica sorgente discreta. In particolare, possiamo pensare di effettuare una codifica di Huffman di tale sorgente: indicato con $\overline{N}_Y$ il numero medio di bit che tale codifica associa a ciascun simbolo di Y e indicata con H(Y) l'entropia della sorgente Y, varrà allora la relazione

$$H(Y) \le \overline{N}_Y \le H(Y) + 1$$

Vediamo allora quanto vale H(Y): si tratta dell'entropia della sorgente Y , per cui è definita come

$$H(Y) = \sum_{i=1}^{3}\sum_{j=1}^{3} P(x_i, x_j) \log_2 \frac{1}{P(x_i, x_j)}$$

dove $P(x_i, x_j)$ è la probabilità che venga emessa la coppia (x_i, x_j), ossia che la sorgente X emetta prima il simbolo x_i e poi il simbolo x_j.

Poiché siamo nelle ipotesi di indipendenza dei simboli emessi dalla sorgente Y, possiamo scrivere che

$$P(x_i, x_j) = P(x_i)P(x_j)$$

per cui

$$H(Y) = \sum_{i=1}^{3}\sum_{j=1}^{3} P(x_i)P(x_j)\log_2 \frac{1}{P(x_i)P(x_j)} = \sum_{i=1}^{3}\sum_{j=1}^{3} P(x_i)P(x_j)\left[\log_2 \frac{1}{P(x_i)} + \log_2 \frac{1}{P(x_j)}\right] =$$

$$= \sum_{i=1}^{3}\sum_{j=1}^{3} P(x_i)P(x_j)\log_2 \frac{1}{P(x_i)} + \sum_{i=1}^{3}\sum_{j=1}^{3} P(x_i)P(x_j)\log_2 \frac{1}{P(x_j)} =$$

$$= \sum_{i=1}^{3} P(x_i)\log_2 \frac{1}{P(x_i)} \sum_{j=1}^{3} P(x_j) + \sum_{i=1}^{3} P(x_i) \sum_{j=1}^{3} P(x_j)\log_2 \frac{1}{P(x_j)} =$$

$$= \sum_{i=1}^{3} P(x_i)\log_2 \frac{1}{P(x_i)} + \sum_{j=1}^{3} P(x_j)\log_2 \frac{1}{P(x_j)} = H(X) + H(X) = 2H(X)$$

Abbiamo dunque trovato che

$$\boxed{H(Y) = 2H(X)}$$

per cui possiamo scrivere che

$$2H(X) \le \overline{N}_Y \le 2H(X) + 1$$

o anche che

$$H(X) \le \frac{\overline{N}_Y}{2} \le H(X) + \frac{1}{2}$$

Adesso, se $\overline{N}_Y$ è il numero medio di bit che il codice di Huffman associa a ciascun simbolo di Y, $\overline{N}_Y / 2$ non è altro che $\overline{N}_X$, in quanto ogni simbolo di Y corrisponde a 2 simboli di X.

Abbiamo perciò ottenuto che

$$\boxed{H(X) \le \overline{N}_X \le H(X) + \frac{1}{2}}$$

il che significa che abbiamo senz'altro ottenuto un avvicinamento di $\overline{N}_X$ al suo valore minimo rispetto a quello che avremmo ottenuto se avessimo codificato secondo Huffman direttamente la sorgente X.

In definitiva, quindi, abbiamo ottenuto che il valore di $\overline{N}_X$ migliora (ossia si avvicina a H(X)) se andiamo a codificare secondo Huffman la sorgente Y: questo metodo prende il nome di **codifica a 2 blocchi** e la sorgente Y prende il nome di **sorgente a 2 blocchi**.

E' intuitivo allora comprendere come ulteriori miglioramenti si otterrebbero considerando sorgenti a 3, 4 o più blocchi: si trova infatti che

$$\boxed{H(X) \le \overline{N}_X \le H(X) + \frac{1}{n}}$$

Il problema, però, viene dal fatto che, con questo metodo, gli apparati di codifica e decodifica diventano sempre più complessi e quindi sempre più costosi.

Esempio

Vediamo un esempio in cui la codifica a blocchi risulta particolarmente conveniente.

Sia $\{A, B\}$ l'alfabeto della nostra sorgente e siano p_A=0.1 e p_B=0.9 le rispettive probabilità di emissione. Calcoliamo l'entropia della sorgente: abbiamo già avuto modo di vedere che la formula generale, per una sorgente con 2 soli simboli, è

$$H(X) = \sum_{i=1}^{2} p_i \log_2 \frac{1}{p_i} = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$$

Ponendo allora p=0.9. essa diventa

$$H(X) = 0.9 \log_2 \frac{1}{0.9} + (0.1) \log_2 \frac{1}{0.1}$$

Il valore esatto di H(X) risulta essere 0.2

Se codifichiamo secondo Huffman questa sorgente, otteniamo $\overline{N}_X = 1$: anche se si tratta del minimo valore che possiamo ottenere con i normali metodi di codifica di X, è evidente che si tratta di un valore molto distante dal valore dell'entropia 0.2, per cui questo è un tipico caso in cui è opportuna una codifica a blocchi.

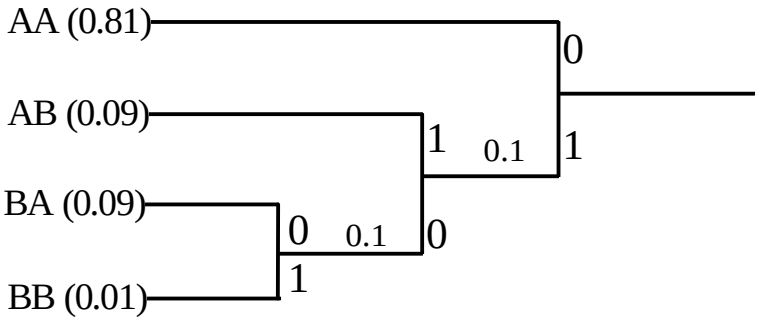
Vediamo perciò se e come migliorano le cose usando una codifica a 2 blocchi: intanto, la sorgente a 2 blocchi sarà evidentemente

$$Y = \{(A, A), (A, B), (B, A), (B, B)\}$$

Le probabilità di emissione sono inoltre le seguenti:

$$p_{AA} = 0.9 * 0.9 = 0.81$$
$$p_{AB} = 0.9 * 0.1 = 0.09$$
$$p_{BA} = 0.09$$
$$p_{BB} = 0.01$$

Andiamo allora a codificare secondo Huffman questa sorgente Y: l'albero risulta essere



per cui il codice è

$$AA \longrightarrow 0$$
$$AB \longrightarrow 11$$
$$BA \longrightarrow 100$$
$$BB \longrightarrow 101$$

Andando allora a calcolare il numero medio di bit che questo codice associa a ciascun simbolo della sorgente Y si trova che $\overline{N}_Y = 1.29$, da cui

$$\overline{N}_X = \frac{\overline{N}_Y}{2} = 0.645$$

Evidentemente, questo valore è già molto minore del valore 1 che avevamo all'inizio, per cui effettivamente la situazione è migliorata.

Sorgenti con memoria

INTRODUZIONE

Fino ad ora noi abbiamo sempre fatto in nostri ragionamenti riferendoci a due ipotesi fondamentali di base:

- la prima è che la sorgente sia "*discreta*", il che significa che essa emette, ad intervalli di tempo regolare, simboli facenti parte di un alfabeto finito, del tipo $\{s_1, s_2, \ldots, s_N\}$
- la seconda è che la sorgente sia "*senza memoria*", il che significa che l'emissione di ciascun simbolo è del tutto indipendente dai simboli emessi in precedenza:

E' ovvio che, mentre la prima ipotesi è assolutamente realistica, non lo è certamente la seconda, in quanto, nella realtà, la maggior parte delle sorgenti sono **sorgenti con memoria**, ossia sorgenti in cui *l'emissione di ciascun simbolo è comunque condizionata dai simboli emessi in precedenza.* Vogliamo allora studiare questo tipo di sorgenti: come si vedrà, valgono grossomodo gli stessi concetti visti fino ad ora, con la differenza principale di una maggiore complicazione matematica, derivante essenzialmente dalla presenza delle **probabilità condizionate** in luogo di quelle assolute viste fino ad ora.

ENTROPIA CONDIZIONALE

Supponiamo dunque che l'alfabeto della nostra sorgente con memoria sia $\{s_1, s_2, \ldots, s_M\}$: questa sorgente, usando i simboli di questo alfabeto, emette dei messaggi che dobbiamo codificare. Ciascuno di questi messaggi, dato che dobbiamo metterci nel caso più generale possibile, può essere sicuramente visto come una possibile realizzazione di un processo stocastico, il che ci consente di studiare il problema sfruttando ciò che sappiamo a proposito dei **processi stocastici**. Supponiamo allora di avere i seguenti messaggi generici:

$$\text{messaggio } 1 \longrightarrow s_3\ s_4\ s_5\ s_1\ s_2\ s_M\ \ldots.$$
$$\text{messaggio } 2 \longrightarrow s_1\ s_2\ s_5\ s_2\ s_M\ s_{M-1}\ \ldots.$$
$$\ldots\ldots$$
$$\text{messaggio } k \longrightarrow s_M\ s_1\ s_4\ s_2\ s_2\ s_{M-2}\ \ldots.$$

Queste **realizzazioni** sono evidentemente delle funzioni discrete nel tempo (in quanto l'emissione dei simboli avviene non in modo continuo bensì in istanti di tempo discreti) a valori discreti (in quanto l'alfabeto sorgente possiede un numero finito di simboli). Siamo cioè in presenza di un **processo tempo-discreto a valori discreti**.

Inoltre, possiamo pensare di estrarre dal processo una o più variabili aleatorie: la **variabile aleatoria X_n** estratta al generico istante **t=nT** indica perciò il valore assunto dal processo all'istante nT o, ciò che è lo stesso, il valore assunto dal processo al passo n, che corrisponde all'intervallo di tempo $[(n-1)T, nT]$. Indichiamo in particolare con X_0 la variabile aleatoria estratta al passo 0, cioè all'istante t=0, e con X_1 quella estratta al passo 1, ossia all'istante t=T.

Prende allora il nome di **entropia condizionale** della sorgente la seguente quantità:

$$H(X_1|X_0) = \sum_{i=1}^{M}\sum_{j=1}^{M} P(X_0 = s_i \cap X_1 = s_j) \log_2 \frac{1}{P(X_1 = s_j | X_0 = s_i)}$$

Vediamo in primo luogo di descrivere cosa compare in questa definizione:

- M è il numero (finito o infinito numerabile) di simboli di cui è costituito l'alfabeto sorgente;
- s_i ed s_j sono due generici simboli appartenenti a tale alfabeto;
- $P(X_1 = s_j | X_0 = s_i)$ è la probabilità che il processo al passo 1 assuma lo stato s_j, dopo che al passo 0 ha assunto lo stato s_i (o, in termini di sorgente, la probabilità che la sorgente emetta al passo 1 il simbolo s_j dopo che al passo 0 ha emesso il simbolo s_i);
- $P(X_0 = s_i \cap X_1 = s_j)$ è la probabilità che il processo al passo 0 assuma lo stato s_i e al passo 1 lo stato s_j (o, in termini di sorgente, la probabilità che la sorgente emetta al passo 0 il simbolo s_i e al passo 1 il simbolo s_j).

Facciamo anche osservare l'analogia esistente tra questa definizione e quella di "entropia" data per le sorgenti senza memoria, che era

$$H(X) = \sum_{i=1}^{N} P(x_i) \log_2 \frac{1}{P(x_i)}$$

Facciamo infine osservare che la definizione dell'entropia condizionale $H(X_1|X_0)$ non è altro che la definizione di valor medio di una variabile aleatoria ottenuta come funzione della variabile aleatoria $(X_1|X_0)$, dove la funzione non è altro che

$$\log_2 \frac{1}{P(X_1 = s_j | X_0 = s_i)}$$

L'entropia condizionale può essere definita sia per le variabili aleatorie estratte al passo 0 ed al passo 1 sia anche per quelli estratte al passo 0, al passo 1 e così via fino al generico **passo n-simo**: in questo caso, la definizione diventa evidentemente

$$H\left(X_n|X_{n-1},...,X_1,X_0\right)=\sum_{i_0=1}^{M}\sum_{i_1=1}^{M}...\sum_{i_n=1}^{M}P\left(X_n=s_{i_n}\cap...\cap X_1=s_{i_1}\cap X_0=s_{i_0}\right)\log_2\frac{1}{P\left(X_n=s_{i_n}\left|X_{n-1}=s_{i_{n-1}}\cap..\cap X_0=s_{i_0}\right.\right)}$$

Proprietà

Consideriamo l'entropia condizionale al passo 1, ossia $H\left(X_1|X_0\right)$: dimostriamo che vale la relazione

$$\boxed{H\left(X_1|X_0\right)\le H\left(X_1\right)}$$

dove X_1 è in pratica vista come una sorgente senza memoria avente come alfabeto quello stesso della sorgente X che stiamo considerando.

Dimostrare quella relazione equivale evidentemente a dimostrare che

$$H\left(X_1|X_0\right)-H\left(X_1\right)\le 0$$

Cominciamo a sostituire l'espressione di $H\left(X_1|X_0\right)$ dataci dalla definizione:

$$H\left(X_1|X_0\right)-H\left(X_1\right)=\sum_{i=1}^{M}\sum_{j=1}^{M}P\left(X_0=s_i\cap X_1=s_j\right)\log_2\frac{1}{P\left(X_1=s_j|X_0=s_i\right)}-H\left(X_1\right)$$

Sostituendo inoltre la definizione di entropia di una sorgente senza memoria, abbiamo che

$$H\left(X_1|X_0\right)-H\left(X_1\right)=\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)\log_2\frac{1}{P\left(X_1=s_{i_1}|X_0=s_{i_0}\right)}-\sum_{i_1=1}^{N}P\left(X_1=x_{i_1}\right)\log_2\frac{1}{P\left(X_1=x_{i_1}\right)}$$

Al fine di ricondurci ad un unica doppia sommatoria, possiamo usare il *teorema delle probabilità totali* e scrivere che

$$P\left(X_1=x_{i_1}\right)=\sum_{i_0=1}^{M}P\left(X_1=x_{i_1}\cap X_0=x_{i_0}\right)$$

per cui, andando a sostituire, abbiamo che

$$H\left(X_1|X_0\right)-H\left(X_1\right)=\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)\log_2\frac{1}{P\left(X_1=s_{i_1}|X_0=s_{i_0}\right)}-$$
$$-\sum_{i_1=1}^{N}\sum_{i_0=1}^{M}P\left(X_1=x_{i_1}\cap X_0=x_{i_0}\right)\log_2\frac{1}{P\left(X_1=x_{i_1}\right)}$$

Ponendo tutto sotto un'unica doppia sommatoria, abbiamo

$$H\left(X_1|X_0\right)-H\left(X_1\right)=\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)\left(\log_2\frac{1}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}-\log_2\frac{1}{P\left(X_1=x_{i_1}\right)}\right)$$

Usando adesso una nota proprietà dei logaritmi, abbiamo

$$H\left(X_1|X_0\right)-H\left(X_1\right)=\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)\log_2\frac{P\left(X_1=x_{i_1}\right)}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}$$

Adesso dobbiamo usare un'altra proprietà dei logaritmi che è stata già utilizzata in precedenza: si tratta della proprietà secondo cui

$$\log_2 y=(\ln e)\ln y\le(\ln e)(y-1)$$

Nel nostro caso, abbiamo, applicando tale proprietà, che

$$\log_2\frac{P\left(X_1=x_{i_1}\right)}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}=(\ln e)\left(\frac{P\left(X_1=x_{i_1}\right)}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}-1\right)$$

per cui

$$H\left(X_1|X_0\right)-H\left(X_1\right)\le\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)(\ln e)\left(\frac{P\left(X_1=x_{i_1}\right)}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}-1\right)=$$

$$=(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)\left(\frac{P\left(X_1=x_{i_1}\right)}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}-1\right)$$

Adesso, il termine $P\left(X_0=s_{i_0}\cap X_1=s_{i_1}\right)$ equivale anche a $P\left(X_1=s_{i_1}\cap X_0=s_{i_0}\right)$ e , inoltre, questo è possibile esprimerlo le probabilità condizionate:

$$H\left(X_1|X_0\right)-H\left(X_1\right)\le(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)P\left(X_0=s_{i_0}\right)\left(\frac{P\left(X_1=x_{i_1}\right)}{P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)}-1\right)$$

Semplificando e scomponendo adesso le due doppie sommatorie, abbiamo che

$$H\left(X_1|X_0\right)-H\left(X_1\right)\le(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\right)\left(P\left(X_1=x_{i_1}\right)-P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)\right)=$$

$$=(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\right)P\left(X_1=x_{i_1}\right)-(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M}P\left(X_0=s_{i_0}\right)P\left(X_1=s_{i_1}\middle|X_0=s_{i_0}\right)$$

A questo punto, possiamo mostrare come entrambi i termini a secondo membro sono pari a $\ln e$, da cui consegue la tesi: per quanto riguarda il primo termine abbiamo che

$$(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M} P\left(X_0 = s_{i_0}\right)P\left(X_1 = x_{i_1}\right) = (\ln e)\sum_{i_1=1}^{M} P\left(X_0 = s_{i_0}\right)\sum_{i_0=1}^{M} P\left(X_1 = x_{i_1}\right) = (\ln e)1\cdot 1 = \ln e$$

Per quanto riguarda invece il secondo, abbiamo che

$$(\ln e)\sum_{i_1=1}^{M}\sum_{i_0=1}^{M} P\left(X_0 = s_{i_0}\right)P\left(X_1 = s_{i_1}\middle|X_0 = s_{i_0}\right) = (\ln e)\sum_{i_0=1}^{M}\sum_{i_1=1}^{M} P\left(X_0 = s_{i_0}\right)P\left(X_1 = s_{i_1}\middle|X_0 = s_{i_0}\right) =$$

$$= (\ln e)\sum_{i_0=1}^{M} P\left(X_0 = s_{i_0}\right)\sum_{i_1=1}^{M} P\left(X_1 = s_{i_1}\middle|X_0 = s_{i_0}\right) = (\ln e)$$

Generalizzazione

Ciò che abbiamo appena dimostrato è quindi che

$$H\left(X_1\middle|X_0\right) \le H\left(X_1\right)$$

Questa proprietà può essere anche estesa alla entropia condizionale al passo n, in quanto si dimostra che vale la relazione

$$\boxed{H\left(X_n\middle|X_{n-1},\ldots,X_1,X_0\right) \le H\left(X_n\middle|X_{n-1},\ldots,X_1\right)}$$

ENTROPIA ALL'INFINITO

Data l'entropia condizionale al passo n, ossia $H\left(X_1\middle|X_{n-1},\ldots,X_1,X_0\right)$, si definisce invece **entropia all'infinito** della sorgente considerata la seguente quantità:

$$\boxed{H_\infty(X) = \lim_{n\to\infty} H\left(X_1\middle|X_{n-1},\ldots,X_1,X_0\right)}$$

Sulla base della proprietà

$$H\left(X_n\middle|X_{n-1},\ldots,X_1,X_0\right) \le H\left(X_n\middle|X_{n-1},\ldots,X_1\right)$$

è possibile dimostrare quest'altra relazione fondamentale:

$$\boxed{H_\infty(X) \le H\left(X\right)}$$

Essa dice che *l'entropia all'infinito della sorgente con memoria X è minore o al più uguale all'entropia della stessa sorgente, calcolata però come se essa fosse senza memoria.*

Naturalmente, in base a come abbiamo definito le varie entropie, si capisce come si tratti sempre di quantità positive, per cui possiamo ulteriormente perfezionare quella proprietà scrivendo che

$$\boxed{0 \le H_\infty(X) \le H(X)}$$

Perché è importante questa relazione? Perché, così come abbiamo trovato per le sorgenti senza memoria la relazione $\overline{N} \ge H(X)$, dove $\overline{N}$ è il numero medio di bit che noi possiamo associare a ciascun simbolo dell'alfabeto sorgente quando codifichiamo la sorgente, allo stesso modo, per le sorgenti con memoria, si trova che

$$\boxed{\overline{N} \ge H_\infty(X)}$$

per cui abbiamo ancora una volta una stima del valore minimo che possiamo raggiungere per $\overline{N}$.

SORGENTI MARKOVIANE OMOGENEE

In base alla definizione data, risulta evidente quanto complicato sia, in generale, la valutazione dell'entropia all'infinito della nostra sorgente con memoria. Ci sono però dei casi in cui invece questo calcolo risulta semplificato: uno di questi è senz'altro quello delle cosiddette "*sorgenti markoviane*", ossia sorgenti con memoria, dove però tale memoria è limitata al simbolo precedente.

Per capirci meglio, una sorgente si dice "**markoviana**" (in accordo a quanto visto a proposito delle catene di Markov) quando l'emissione di ciascun simbolo dipende solo dal simbolo emesso in precedenza e non da tutti gli altri.

La prima evidente semplificazione per una sorgente markoviana è la seguente: mentre per le sorgenti con memoria generica noi consideriamo le probabilità condizionate di emissione

$$P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \cap \ldots \cap X_0 = s_{i_0}\right)$$

per le sorgenti markoviane tali probabilità si riducono evidentemente a

$$P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}}\right)$$

Tra l'altro, se la sorgente è **markoviana omogenea**, queste probabilità condizionate di emissione risultano costanti passo per passo, ossia risultano indipendenti da n, mentre risultano dipendenti solo dai simboli s_{i_n} e $s_{i_{n-1}}$

Vediamo allora perché diventa facile il calcolo dell'entropia all'infinito di una sorgente markoviana: intendiamo dimostrare che sussiste la relazione

$$\boxed{H_\infty(X) = \sum_{j=1}^{M} H(s_j)\pi_j}$$

dove π_j è la probabilità asintotica del generico simbolo s_j mentre $H(s_j)$ è l'entropia dello stato s_j, definita come

$$H(s_j) = \sum_{i=1}^{M} p_{ji} \log_2 \frac{1}{p_{ji}}$$

Dobbiamo dunque dimostrare che

$$H_\infty(X) = \sum_{j=1}^{M} \sum_{i=1}^{M} p_{ji} \log_2 \left(\frac{1}{p_{ji}} \right) \pi_j$$

Intanto, per definizione, l’entropia all’infinito è

$$H_\infty(X) = \lim_{n \to \infty} H\left(X_1 \middle| X_{n-1}, \ldots, X_1, X_0 \right)$$

Sostituendo l’espressione di $H\left(X_1 \middle| X_{n-1}, \ldots, X_1, X_0 \right)$, abbiamo che

$$H_\infty(X) = \lim_{n \to \infty} \sum_{i_0=1}^{M} \sum_{i_1=1}^{M} \ldots \sum_{i_n=1}^{M} P\left(X_n = s_{i_n} \cap \ldots \cap X_1 = s_{i_1} \cap X_0 = s_{i_0} \right) \log_2 \frac{1}{P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \cap \ldots \cap X_0 = s_{i_0} \right)}$$

In base alla proprietà delle sorgenti markoviane per cui

$$P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \cap \ldots \cap X_0 = s_{i_0} \right) = P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \right)$$

abbiamo poi che

$$H_\infty(X) = \lim_{n \to \infty} \sum_{i_0=1}^{M} \sum_{i_1=1}^{M} \ldots \sum_{i_n=1}^{M} P\left(X_n = s_{i_n} \cap \ldots \cap X_1 = s_{i_1} \cap X_0 = s_{i_0} \right) \log_2 \frac{1}{P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \right)}$$

A questo punto, invertiamo gli ordini di sommatoria, in modo da avere per primo l’indice i_n e per ultimo l’indice i_0:

$$H_\infty(X) = \lim_{n \to \infty} \sum_{i_n=1}^{M} \ldots \sum_{i_1=1}^{M} \sum_{i_0=1}^{M} P\left(X_n = s_{i_n} \cap \ldots \cap X_1 = s_{i_1} \cap X_0 = s_{i_0} \right) \log_2 \frac{1}{P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \right)}$$

Il termine logaritmico dipende solo dalle prime due sommatorie, per cui possiamo riscrivere quella relazione nella forma

$$H_\infty(X) = \lim_{n \to \infty} \sum_{i_n=1}^{M} \sum_{i_{n-1}=1}^{M} \log_2 \frac{1}{P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \right)} \sum_{i_{n-2}=1}^{M} \ldots \sum_{i_0=1}^{M} P\left(X_n = s_{i_n} \cap \ldots \cap X_1 = s_{i_1} \cap X_0 = s_{i_0} \right)$$

Ora noi possiamo fare una importante semplificazione considerando quali sono gli indici delle sommatorie che devono essere applicate al termine $P\left(X_n = s_{i_n} \cap \ldots \cap X_1 = s_{i_1} \cap X_0 = s_{i_0} \right)$: si intuisce infatti come quella relazione si riduca a

$$H_\infty(X) = \lim_{n \to \infty} \sum_{i_n=1}^{M} \sum_{i_{n-1}=1}^{M} \log_2 \frac{1}{P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}} \right)} P\left(X_n = s_{i_n} \cap X_{n-1} = s_{i_{n-1}} \right)$$

A questo punto, possiamo esprimere il termine $P\left(X_n = s_{i_n} \cap X_{n-1} = s_{i_{n-1}}\right)$ mediante le probabilità condizionate: ciò che otteniamo è che

$$P\left(X_n = s_{i_n} \cap X_{n-1} = s_{i_{n-1}}\right) = P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}}\right) P\left(X_{n-1} = s_{i_{n-1}}\right)$$

per cui, andando a sostituire, abbiamo

$$H_\infty(X) = \lim_{n\to\infty} \sum_{i_n=1}^{M} \sum_{i_{n-1}=1}^{M} \log_2 \frac{1}{P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}}\right)} P\left(X_n = s_{i_n} \middle| X_{n-1} = s_{i_{n-1}}\right) P\left(X_{n-1} = s_{i_{n-1}}\right)$$

Ora possiamo semplificare un po' l'aspetto formale di questa relazione: ponendo infatti

$$j = s_{i_{n-1}}$$
$$i = s_{i_n}$$

essa può essere riscritta nella forma

$$H_\infty(X) = \lim_{n\to\infty} \sum_i \sum_j \log_2 \frac{1}{P\left(X_n = i \middle| X_{n-1} = j\right)} P\left(X_n = i \middle| X_{n-1} = j\right) P\left(X_{n-1} = j\right)$$

Adesso, nell'ipotesi che la sorgente sia anche omogenea, abbiamo detto che il termine $P\left(X_n = i \middle| X_{n-1} = j\right)$ non dipende in alcun modo da n, per cui possiamo spostare il limite nel modo seguente:

$$H_\infty(X) = \sum_i \sum_j \log_2 \frac{1}{P\left(X_n = i \middle| X_{n-1} = j\right)} P\left(X_n = i \middle| X_{n-1} = j\right) \lim_{n\to\infty} P\left(X_{n-1} = j\right)$$

Inoltre, indicando con $p_{ji} = P\left(X_n = i \middle| X_{n-1} = j\right)$ la generica probabilità di transizione ad un passo, quella relazione diventa ancora

$$H_\infty(X) = \sum_i \sum_j p_{ji} \log_2 \frac{1}{p_{ji}} \lim_{n\to\infty} P\left(X_{n-1} = j\right)$$

A questo punto abbiamo ottenuto ciò che volevamo dimostrare, in quanto il termine $\lim_{n\to\infty} P\left(X_{n-1} = j\right)$ non è altro che la probabilità asintotica π_j, per cui

$$H_\infty(X) = \sum_i \sum_j p_{ji} \pi_j \log_2 \frac{1}{p_{ji}}$$

Cosa ci dice questa relazione? Essa ci dice che, data la nostra sorgente markoviana omogenea, possiamo calcolarci la sua entropia all'infinito, in modo abbastanza semplice, conoscendo solo le probabilità di transizione ad un passo e le probabilità asintotiche. Di

solito, le probabilità di transizione ad un passo costituiscono un dato del problema, per cui tutto si riduce a calcolare le probabilità asintotiche.

Esempio

Vediamo un esempio pratico di quanto appena detto. Supponiamo di avere una sorgente markoviana omogenea il cui alfabeto sia $\{A, B, C\}$. Supponiamo di conoscere, di questa sorgente, la matrice di transizione ad un passo:

$p_{AA} = \frac{1}{2}$	$p_{AB} = \frac{1}{4}$	$p_{AC} = \frac{1}{4}$
$p_{BA} = \frac{1}{8}$	$p_{BB} = \frac{1}{2}$	$p_{BC} = \frac{3}{8}$
$p_{CA} = \frac{1}{4}$	$p_{CB} = \frac{1}{8}$	$p_{CC} = \frac{3}{4}$

Ovviamente, il significato di quelle probabilità è quello che si ricava dalla relazione

$$p_{ji} = P\left(X_n = i \middle| X_{n-1} = j\right)$$

Ad esempio:

$$p_{AA} = P\left(X_n = A \middle| X_{n-1} = A\right)$$
$$p_{BA} = P\left(X_n = B \middle| X_{n-1} = A\right)$$
$$p_{CB} = P\left(X_n = B \middle| X_{n-1} = C\right)$$

Naturalmente, essendo la sorgente omogenea, il valore di n può essere qualsiasi tra 1 e ∞.
Per conoscere l'entropia all'infinito di questa sorgente, dobbiamo applicare la relazione

$$H_\infty(X) = \sum_i \sum_j p_{ji} \pi_j \log_2 \frac{1}{p_{ji}}$$

per cui quello che ci serve sono le probabilità asintotiche. queste si calcolano risolvendo il sistema rappresentato da

$$\begin{cases} [\pi] = [\pi][P] \\ \sum_{i=1}^{3} \pi_i = 1 \end{cases}$$

Nel nostro caso, la relazione matriciale $[\pi] = [\pi][P]$ è la seguente:

$$\begin{bmatrix} \pi_A \\ \pi_B \\ \pi_C \end{bmatrix} = \begin{bmatrix} \pi_A \\ \pi_B \\ \pi_C \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{2} & \frac{3}{8} \\ \frac{1}{4} & \frac{1}{8} & \frac{3}{4} \end{bmatrix}$$

Alle tre equazioni rappresentate da questa relazione matriciale va aggiunta una quarta equazione in quanto solo 2 di esse sono linearmente indipendenti: la quarta equazione è appunto $\pi_A + \pi_B + \pi_C = 1$.

Risolvendo perciò il sistema, si determinano le probabilità asintotiche e quindi si può calcolare l'entropia all'infinito della sorgente considerata.

Autore: **SANDRO PETRIZZELLI**
e-mail: sandry@iol.it
sito personale: http://users.iol.it/sandry
succursale: http://digilander.iol.it/sandry1