Thomas D. Schneider

FONDAMENTI

DI

TEORIA DELL'INFORMAZIONE

Traduzione Italiana non letterale
interamente svolta da:

Simone Baldi
Laureando in Scienze dell'Informazione presso
l'Università Statale degli Studi, Milano (Italy)

versione 1.01 tratta dal file:
ftp://ftp.ncifcrf.gov/pub/delila/primer.ps.Z
terminata il 19 Febbraio 1996

FONDAMENTI
DI
TEORIA DELL'INFORMAZIONE

Sorgente : ftp://ftp.ncifcrf.gov/pub/delila/primer.ps.Z

Thomas D. Schneider (*)

versione = 2.33 di primer.tex 27 Luglio 1995

Traduzione italiana non letterale eseguita da

Simone Baldi

versione 1.01 tradotta dal file:
ftp://ftp.ncifcrf.gov/pub/delila/primer.ps.Z
terminata il 12 Marzo 1996

Questo documento e' stato scritto per quei biologi molecolari che non hanno molta famigliarita' con la Teoria dell'Informazione.
Il suo scopo, pertanto, e' presentare queste idee in modo che tutti possano capire come applicarle ai siti di filamenti (di DNA)[1,2,3,4,5,6,7,8 9].
La maggior parte di questo materiale si trova facilmente nei testi introduttivi alla Teoria dell'Informazione.
Benche' il lavoro originale di Shannon sulla Teoria dell'Informazione [10] a volte puo' risultare di difficile lettura, in altri contesti viene molto decantato.
Saltando pero' le parti piu' ostiche si puo' trovarlo interessante e piacevole.
Pierce ha gia' pubblicato un libro a portata di tutti [11] che e' una buona introduzione alla Teoria dell'Informazione.
Altre introduzioni sono elencate nei riferimenti [1].
Un manuale di grande utilita' e' in riferimento [12].
Il lavoro completo di Shannon e' stato referenziato in [13]. Informazioni su come ordinare questi libri sono fornite nel file di testo:

ftp://ftp.ncifcrf.gov/pub/delila/bionet.info-theory.faq

Reperibile via ftp anonimo.

Altri opuscoli e documentazioni in merito si possono trovare alla pagina del World Wide Web:

http://www-lmmb.ncifcrf/ ~toms/

Nota: Se si incontrano problemi nella lettura di uno o piu' passi di questo documento, per cortesia inviatemi una e-mail indicando esattamente il(i) punto(i) problematico(i). Se lo riterro' opportuno, modifichero' il testo per renderne piu' agevole la lettura. I miei personali ringraziamenti vanno a tutti coloro che hanno puntualizzato su alcune questioni portando il documento sino a questa versione.

(*)

National Cancer Institute, Frederick Cancer Research and Development Center, Laboratory of Mathematical Biology, P. O. Box B, Frederick, MD 21702. email address: toms@ncifcrf.gov. This text originated as chapter II of my PhD thesis: "The Information Content of Binding Sites on Nucleotide Sequences", University of Colorado, 1984. As a U. S. government work, this document cannot be copyrighted.

Informazione ed Incertezza

Informazione e Incertezza sono termini tecnici usati per descrivere misurazioni coinvolte in qualunque processo che debba selezionare uno, o piu' oggetti, in un ben determinato insieme di oggetti.
Non vogliamo occuparci del significato o delle implicazioni dell'Informazione dal momento che nessuno sa come fare cio' utilizzando un metodo matematico.
Supponiamo di avere un dispositivo che sia in grado di generare (produrre) 3 simboli, A, B, e C.
Quando siamo in attesa di ricevere il prossimo simbolo, siamo Incerti su quale simbolo verra' generato (prodotto).
Un simbolo ci giunge, e noi lo osserviamo; la nostra Incertezza diminuisce, e noi notiamo di aver ricevuto una certa quantita' d'Informazione.
Dunque, l'Informazione non e' altro che una diminuzione di Incertezza. Come possiamo misurare questa Incertezza?
Il modo piu' semplice puo' essere quello di dire che,in questo particolare caso, abbiamo una "Incertezza di 3 simboli".
Questo ragionamento comincia ad esser chiaro dal momento in cui iniziamo ad osservare un secondo dispositivo, il quale, supponiamo, generi (produca) i simboli 1 e 2.
Questo secondo dispositivo 'genera in noi' una "Incertezza" di 2 simboli.
Se combiniamo i due dispositivi in un unico dispositivo, osserviamo che ci sono 6 possibiilita', A1, A2, B1, B2, C1, C2.
Quest'ultimo dispositivo ha una "Incertezza" di 6 simboli. Questo non e' certo il modo in cui siamo abituati a pensare l'Informazione; se riceviamo due libri, preferiamo pensare che otteniamo due volte l'Informazione che avremmo ottenuto ricevendo un solo libro.
In questa linea di idee ci piacerebbe imporre che la nostra unita' di misura abbia una struttura additiva.

... E, infatti, cosi' faremo.

E' semplice farlo, basta prendere il logaritmo del numero dei simboli possibili percio' possiamo sommare i logaritmi invece di moltiplicare tra loro il numero dei simboli dei dispositivi.

Nel nostro esempio, il primo dispositivo 'genera in noi' una Incertezza di log(3), il secondo di log(2) ed il dispositivo combinato di log(3)+log(2)=log(6).

Le basi che prendiamo per calcolare i logaritmi determinano le unita' di misura.
Quando usiamo la base 2 l'unita' e' il bit (la base 10 ci da' i digits e la base naturale dei logaritmi, $\mathcal{E}$, ci da' i nits).
Quindi se un dispositivo genera un solo simbolo, abbiamo una Incertezza pari a $\log_2(1)$ = 0 bits, cioe' noi NON abbiamo nessuna Incertezza su cio' che il dispositivo sta per produrre.
Se esso genera due simboli la nostra Incertezza sara' pari a $\log_2(2)$ = 1 bit. (D'ora in poi useremo sempre la base 2.)
Leggendo un mRNA, si osserva che se il ribosoma incontra una qualunque delle 4 basi ugualmente probabili, allora l'Incertezza e' pari a 2 bits.
In questo modo, la nostra formula per l'Incertezza e' $\log_2(M)$, dove M e' il numero totale di simboli possibili.
Il prossimo passo e' quello di estendere la formula in modo che si possano trattare i casi in cui i simboli NON siano equiprobabili.
Per esempio, se ci sono 3 simboli possibili ma uno di essi NON appare mai, allora l'Incertezza e' pari a 1 bit.
Se il terzo simbolo appare raramente rispetto agli altri 2, allora la nostra Incertezza potra' essere superiore a 1 bit, ma certamente non raggiungera' mai $\log_2(3)$ bits.

Procediamo dunque a rivedere la formula in questo modo:

(**1**)

$$\log_2(M) = -\log_2(M^{-1}) = -\log_2(\frac{1}{M}) = -\log_2(P)$$

Dove P = 1/M e' la probabilita' che uno qualunque dei nostri possibili simboli appaia. (Se non si ricorda questo trucchetto del 'tirare fuori il segno' ricordiamo che $\log(M^b) = b*\log(M)$ e poniamo b=-1.)

Ma ora generalizziamo al caso in cui i simboli abbiano differenti probabilità' di apparire e chiamiamo queste probabilita' P_i; sappiamo che sommando tutte queste probabilita', estendendo la somma a tutti gli M simboli possibili, si ottiene 1 (ce lo dice il calcolo delle probabilita', e la statistica):

(**2**)

$$\sum_{i=1}^{M} P_i = 1.$$

(Ricordiamo che col simbolo $\sum_{i=1}^{M}(-_i)$ si intende: 'sommare gli oggetti che stanno entro la parentesi e che sono indicizzati da i, e far partire i da 1 fermamdosi quando i arriva ad M').

La '*sorpresa*' che riceviamo quando vediamo apparire l'i-esimo tipo di simbolo, chiamato anche "surprisal" da Tribus [14], e definita per analogia con $\log_2 P$ e' pari a:

(***3***)

$$\mu_i = -\log_2(P_i)$$

Per esempio, se P_i si avvicina molto a 0, allora saremo molto sorpresi nel vedere apparire l-iesimo simbolo (dal momento che quest'ultimo non dovrebbe MAI apparire), e infatti la formula dice che μ_i si avvicina moltissimo a $+\infty$.

D'altro canto, se P_i=1, non resteremo affatto sorpresi nel vederlo apparire (in quanto puo' apparire SEMPRE) in effetti la formula ci dice che, in questo caso, μ_i=0.

L'**Incertezza** e' dunque la *sorpresa media* per una sequenza infinita di simboli generati dal nostro dispositivo.
Per il momento, troviamo la media per una sequenza di soli N simboli. Supponiamo che l'i-esimo simbolo appaia N_i volte cosi' che:

(***4***)

$$N = \sum_{i=1}^{M} N_i$$

Ci saranno N_i casi in cui avremo sorpresa μ_i. La sorpresa media per N simboli sara':

(***5***)

$$\frac{\sum_{i=1}^{M} N_i \mu_i}{\sum_{i=1}^{M} N_i}$$

Inserendo il denominatore della (***5***) nella somma che sta al numeratore della stessa otteniamo:

(***6***)

$$\sum_{i=1}^{M} \frac{N_i}{N} \mu_i$$

Se valutiamo questa misura su una sequenza infinita di simboli, allora la frequenza N_i/N tende a P_i, la probabilita' dell'i-esimo simbolo. Con questa

sostituzione, osserviamo che la nostra *sorpresa media* (che chiameremo H) diventa:

(7)

$$H = \sum_{i=1}^{M} P_i \mu_i$$

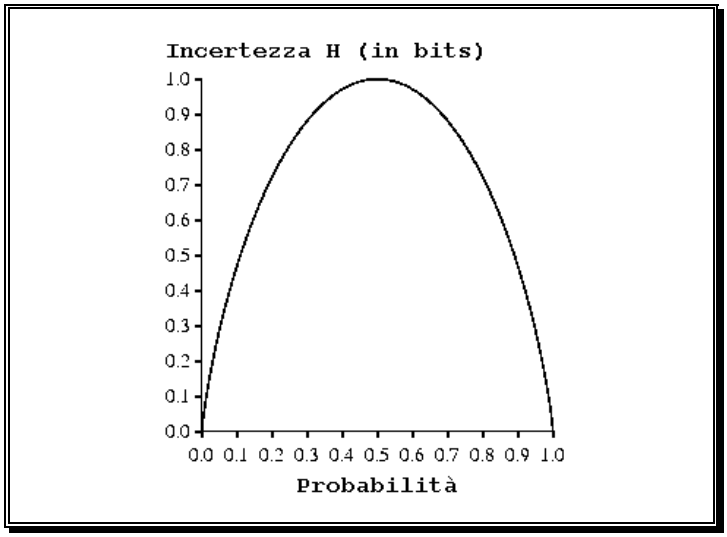
Infine, sostituendo μ_i con la sua espressione esplicita, abbiamo la famosa formula generale di Shannon per l'**Incertezza:**

(8)

Formula Generale di Shannon

$$H = -\sum_{i=1}^{M} P_i \log_2(P_i)$$

Shannon ricavo' questa formula attraverso passaggi assai piu' rigorosi rispetto a cio' che abbiamo fatto noi, selezionando moltissime auspicabili proprieta' per l'**Incertezza** e, solo successivamente, derivando la formula.
Spero che i passaggi che abbiamo finora seguito abbiano dato almeno il senso di come questa formula funziona. Per vedere come si presenta questa funzione possiamo tracciarla nel caso di due simboli.
Si presenta come segue:



Il programma usato per creare questo grafico e' reperibile via ftp anonimo dal file:

ftp://ftp.ncifcrf.gov/pub/delila/hgraph.p

Tengo a far notare che la curva e' simmetrica, raggiunge il suo massimo quando i due simboli sono equiprobabili (probabilita' = 0.5).

Decresce bruscamente sino a zero tutte le volte che uno dei simboli diviene dominante a spese degli altri simboli.
Infatti se uno dei due simboli ha probabilita' molto vicina ad 1 la nostra sorpresa nel vederlo apparire e' pressoche' nulla!

Come esercizio istruttivo, supponiamo che tutti i simboli siano equiprobabili. A cosa si riduce la formula per H (formula (8))? Prova a pensarci da solo prima di proseguire nella lettura.

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Equiprobabili significa che $P_i = 1/M$, percio' se sostituiamo nella equazione dell'**Incertezza** otteniamo:

(***9***)

$$H_{equiprobabile} = -\sum_{i=1}^{M}\left(\frac{1}{M}\log_2(\frac{1}{M})\right)$$

Pero', ragioniamo, M non e' funzione di i, possiamo quindi portarla fuori dalla somma e otteniamo:

(***10***)

$$H_{equiprobabile} = -\left(\frac{1}{M}\log_2(\frac{1}{M})\right)\sum_{i=1}^{M}1$$

$$H_{equiprobabile} = -\left(\frac{1}{M}\log_2(\frac{1}{M})\right)M$$

(***11***)

$$H_{equiprobabile} = -\log_2\left(\frac{1}{M}\right)$$

$$H_{equiprobabile} = \log_2(M)$$

Che e' la semplice equazione con la quale siamo partiti.

Puo' essere valutata per un dato numero di simboli (ad esempio, con M fissato) allora l'**Incertezza** H ha il suo valore massimo quando i simboli sono equiprobabili.
Per esempio una moneta perfettamente bilanciata e' molto piu' difficile da trovare rispetto ad una moneta sbilanciata.
Un altro esercizio potrebbe essere:
Qual'e' l'incertezza se abbiamo 10 simboli e soltanto uno di questi appare? (Suggerimento: $\lim_{p\to 0}(P\log(P))=0$ se poniamo $P=\frac{1}{M}$ e usiamo la regola di de l'Hopital, allora $0\log_2(0)=0$.)

Cosa significa, allora, dire che un segnale ha 1.75 bits per simbolo?

Cio' significa che possiamo convertire il segnale originale in sequenze di zeri e uni (cifre binarie), e, mediamente, dover utilizzare 1.75 cifre binarie per ogni simbolo del segnale originale.
Certi simboli (i piu' rari) richiederanno piu' cifre binarie, mentre altri (i piu' comuni) ne richiederanno meno.

Ecco un esempio di quanto abbiamo appena detto:

Supponiamo di avere M=4 simboli:

(***12***)

A C G T

con relative probabilita' (P_i):

(***13***)

$$P_A=\frac{1}{2},\quad P_C=\frac{1}{4},\quad P_G=\frac{1}{8},\quad P_T\frac{1}{8}$$

Che hanno, rispettivamente, sorprese ($-\log_2(P_i)$):

(***14***)

μ_A=1 bit, μ_C=2 bits, μ_G=3 bits, μ_T=3 bits

Con tutto cio', otteniamo una **Incertezza** pari a:

(***15***)

$$H=\frac{1}{2}\cdot 1+\frac{1}{4}\cdot 2+\frac{1}{8}\cdot 3+\frac{1}{8}\cdot 3 \;=\; 1.75 \quad \text{(bits per simbolo)}$$

Ricodifichiamo tutto cio' in modo che il numero di cifre binarie eguaglino la sorpresa:

(*16*)

```
A  :=    1
C  :=   01
G  :=  000
T  :=  001
```

Per cui la sequenza di caratteri

(*17*)

ACATGAAC

che ha una frequenza di apparizione univocamente determinata dalle probabilita' precedentemente assegnate ad ogni simbolo, viene codificata come:

(*18*)

10110010001101

14 cifre binarie vengono utilizzate per codificare 8 simboli, percio' la media e' 14/8=1.75 cifre binarie per simbolo. Questo metodo di codifica viene chiamato codice di Fano.

I codici di Fano hanno la proprieta' che si possono decodificare senza bisogno di elementi separatori tra un simbolo e l'altro.

Usualmente occorre conoscere la lunghezza del "frame di lettura", ma in questo esempio, come in tutti i codici di Fano, non e' assolutamente necessario conoscerla.

In questo particolare tipo di codice, la prima cifra permette di distinguere l'insieme contenente A, (che abbiamo simboleggiato con A) dall'insieme unione di C,G,T, questi due insiemi sono equiprobabili.

La seconda cifra, che viene usata se la prima e' 0, distingue C dall'unione di G e T; anche questi due insiemi sono equiprobabili.

L'ultima cifra distingue G da T.

Essendo ogni scelta equiprobabile (per come abbiamo assegnato le probabilita' dei simboli all'inizio dell'esempio), ogni cifra binaria di questo codice porta con se' esattamente 1 bit d'informazione.

Attenzione! Questo puo' non essere sempre vero.

Una cifra binaria puo' portare con se' 1 bit se e solo se i due insiemi che essa rappresenta sono equiprobabili (come quelli che sono stati costruiti per questo esempio).

Se questi non sono equiprobabili una cifra binaria puo' benissimo portare con se' meno di un bit d'informazione.

(Ricordiamoci che H e' massima quando le probabilita' sono identiche.)

Quindi se le probabilita' fossero:

(***19***)

$$P_A=\frac{1}{2}, \quad P_C=\frac{1}{6}, \quad P_G=\frac{1}{6}, \quad P_T\frac{1}{6}$$

non ci sarebbe modo di assegnare un codice (**Finito**) in modo tale che una cifra binaria abbia il valore di un bit (solo usando grossi blocchi di simboli, si potrebbe approssimare tale comportamento).

Nell'esempio costruito, non c'e' modo di utilizzare meno di 1.75 cifre binarie per simbolo, ma potremmo rovinarci e utilizzare cifre in piu' per rappresentare il segnale.

I codici di Fano fanno questo ragionevolmente bene scomponendo l'intero insieme di simboli in sottoinsiemi successivi che siano equiprobabili, come e' necessario che sia; si puo' trovare di piu' riguardo i codici di Fano sui testi di Teoria dell'Informazione.

L'Incertezza misurata ci dice solo cio' che puo' essere fatto idealmente, a livello teorico; percio', in ultima analisi, ci dice esattamente che cosa e' impossibile fare.

Per esempio, il segnale con 1.75 bits per simbolo non potra' **MAI** essere codificato utilizzando soltanto una cifra binaria per simbolo.

Radunando le Idee

All'inizio di questo brevissimo trattato abbiamo preso l'Informazione come diminuzione di Incertezza.

Ora che abbiamo una formula generale per l'Incertezza, (***8***), possiamo esprimere l'Informazione usando questa formula.

Supponiamo che un computer contenga qualche Informazione nella sua memoria.

Se andiamo a vedere i singoli **flip-flop** *(°)*,ci troveremo ad avere una qualche **Incertezza** pari ad H_{prima} bits per flip-flop.

Supponiamo ora di eliminare parte della memoria di questo computer, quindi avremo una nuova **Incertezza**, inferiore alla precedente: H_{dopo}.

Allora il computer si ritrova ad aver perso una media di:

(***20***)

$$R = H_{prima} - H_{dopo}$$

bits di Informazione per flip-flop. Se invece eliminiamo tutta la memoria, allora $H_{dopo}=0$ e $R=H_{prima}$.

Ora, consideriamo una telescrivente che riceva caratteri da una linea telefonica.

Se non ci fossero disturbi sulla linea telefonica e nessun'altra fonte d'errore, la telescrivente stamperebbe il testo perfettamente.

Con i disturbi di linea, si crea una qualche Incertezza sul fatto che cio' che e' stato stampato sia corretto o meno.

Percio' prima che un carattere venga stampato, la telescrivente deve essere "preparata" a ricevere qualunque lettera dell'insieme delle lettere possibili, e questo stato di "pronta" ha una sua **Incertezza** H_{prima}, mentre dopo avere ricevuto ogni lettera resta una **Incertezza** H_{dopo}. Questa **Incertezza** e' basata sulla probabilita' che il simbolo appena arrivato non sia uguale al simbolo trasmesso, ed essa misura la quantita' di rumore (o disturbi di linea che dir si voglia).

Shannon ha fatto un esempio di tutto cio' nel capitolo 12 di [10] (o nelle pagine 33 e 34 di [13]).

(°)
flip-flop = **micro circuito elettronico capace di commutare tra' due stati, ovvero**:[tensione sotto una certa soglia=stato(0)], e
[tensione maggiore o uguale a quella certa soglia=stato(1)].

Un sistema avente due simboli equiprobabili che trasmetta con un "clock" di un Hertz e' in grado di inviare Informazione sul mezzo trasmissivo ad una velocita' di un bit al secondo senza errori.

Supponiamo che la probabilita' di ricevere uno zero quando uno zero e' stato trasmesso sia 0.99 e la probabilita' di ricevere un 1, quando un 1 e' stato trasmesso, sia 0.01.

"Questi numeri vengono invertiti se viene ricevuto un 1." (?)

Allora l'**Incertezza** dopo aver ricevuto un simbolo e' :

$$H_{dopo} = -\{ [0.99*\log_2(0.99)] - [0.01*\log_2(0.01)] \} = 0.081$$

quindi l'attuale velocita' di trasmissione

R = 1-0.081 = 0.919 bits al secondo ([2])

La quantita' d'Informazione che otteniamo e' data dalla diminuzione di Incertezza, equazione (***20***).

Sfortunatamente molte persone hanno fatto errori soltanto perche' non avevano appreso con chiarezza questo punto.

Gli errori nascono perche' si assume implicitamente che
non ci siano disturbi nella comunicazione.

Quando non ci sono disturbi, $R=H_{prima}$, cosi' come nella memoria del computer completamente eliminata.

Cosi' e', *se non ci sono disturbi, la quantita' di Informazione comunicata eguaglia l'Incertezza che si ha prima della comunicazione.*

Quando i disturbi ci sono, ed implicitamente si assume che non ci siano, questo porta a tutta una serie di devianti filosofie.

Bisogna **SEMPRE** tener conto che i disturbi, nei mezzi trasmissivi (fisici e quindi reali), **CI SONO!**

([2]) Shannon ha usato la notazione $H_y(x)$ intendendola come Incertezza condizionale del ricevitore y data dal messaggio inviato da x, che noi abbiamo chiamato H_{dopo}. Lui ha anche usato il termine "equivocazione".

Una sottigliezza finale.

In questo breve trattato si puo' trovare strano che venga usato il termine "flip-flop". Tutto cio' perche' la parola "bit" viene internazionalmente evitata.

La ragione di questo e' che ci sono due significati di questa parola, come abbiamo accennato poco fa trattando i codici di Fano, ed e' molto meglio tenerli distinti.

I due significati della parola "bit" sono:

1. Una cifra binaria, 0 o 1. Che puo' solo essere un intero. Questi "bits" sono le unita' elementari per la memorizzazione dell'informazione (dati) nei computers.

2. Una misura di Incertezza, H, o di Informazione R. Questi possono essere numeri Reali in quanto si tratta di una media. E' la misura che Shannon ha usato per trattare di sistemi di comunicazione.

Riferimenti

[1] T.D. Schneider, G.D. Stormo, L. Gold, and A. Ehrenfeucht. 'Information content of binding sites on nucleotide sequences.' J. Mol Biol, 188:415-431, 1986.

[2] T.D. Schneider. 'Information and entropy of patterns in genetic switches.' In G.J. Erickson and C.R. Smith, editors, *Maximum-Entropy and Bayesian Methods in Science and Engineering*, volume 2, pages 147-154 Dordrecht, The Netherlands, 1988. Kluwer Academic Publishers.

[3] T.D. Schneider and G.D. Stormo. 'Execess information at bacteriophage T7 genomic promoters detected by a random cloning technique.' *Nucl. Acids Res,* 17:659-674, 1989.

[4] T.D. Schneider and R.M. Stephens, Sequence Logos: A new way to display consensus sequences. *Nucl. Acids Res,* 18:6097-6100, 1990.

[5] N.D. Herman and T.D. Schneider. 'High information conservation implies that at least three proteins bind independentli to F plasmid.' *IncD repeats J. Bact.*, 174:3558-3560, 1992.

[6] P.P. Papp, D.K. Chattoraj, and T.D. Schneider. 'Information analysis of sequences that bind the replication initiator RepA.' *J. Mol. Biol*, 233:219-230, 1993.

[7] R.M. Stephen and T.D. Schneider. 'Features of spliceosome evolution and function inferred an analysis of the information at human splice sites.' *J. Mol. Biol*, 228:1124-1136, 1992.

[8] T.D. Schneider. 'Sequnence logos, machine/channel capacity, Maxwell's demon, and molecular computers: a review of the theory of molecular machines.' *Nanotechnology,* 5:1-18, 1994.

[9] P.K. Rogan and T.D. Schneider. Using Information content and base frequencies to distinguish mutation from genetic polymorphisms in splice junction recognition sites. *Human Mutation*, 6:74-76, 1995.

[10] C.E. Shannon. A mathematical theory of communication. *Bell System Tec.*, 27:379-423, 623-656, 1948.

[11] J.R. Pierce. An Introduction to Information Theory: *Symbols, Signals and noise.* Dover Publications, Inc, New York, second edition, 1980.

[12] W. Sacco, W. Copes, C.Sloyer, and R. Stark. *Information Theory: Saving Bits.* Janson Publication, Inc, Dedham, MA, 1988.

[13] C.E. Shannon. *Claude Elwood Shannon: Collected Papers.* IEEE Press, Pisacataway, NJ, 1993.

[14] M. Tribus. *Thermostatics and Thermodynamics.* D. Van Nostrand Company, Inc, Princeton,N.J., 1961.

APPENDICE: Un Tutorial sui Logaritmi

Capire la Funzione Log(x) (logaritmo di x).

Nell'operazione matematica dell'addizione noi solitamente prendiamo due numeri e li associamo per crearne un terzo:

$$1+1=2. \qquad (21)$$

Possiamo ripetere questa operazione:

$$1+1+1=3. \qquad (22)$$

La moltiplicazione e' l'operazione che estende questo concetto:

$$3 \times 1 = 3. \qquad (23)$$

Allo stesso modo, possiamo ripetere la moltiplicazione:

$$2 \times 2 = 4. \qquad (24)$$

e.....

$$2 \times 2 \times 2 = 8. \qquad (25)$$

L'estensione della moltiplicazione e' l'elevamento a potenza:

$$2 \times 2 = 2^2 = 4. \qquad (26)$$

e.....

$$2 \times 2 \times 2 = 2^3 = 8. \qquad (27)$$

Questo si legge: "due elevato alla terza e' otto". Essendo che l'elevamento a potenza "conta" semplicemente il numero di moltiplicazioni, gli esponenti si possono sommare:

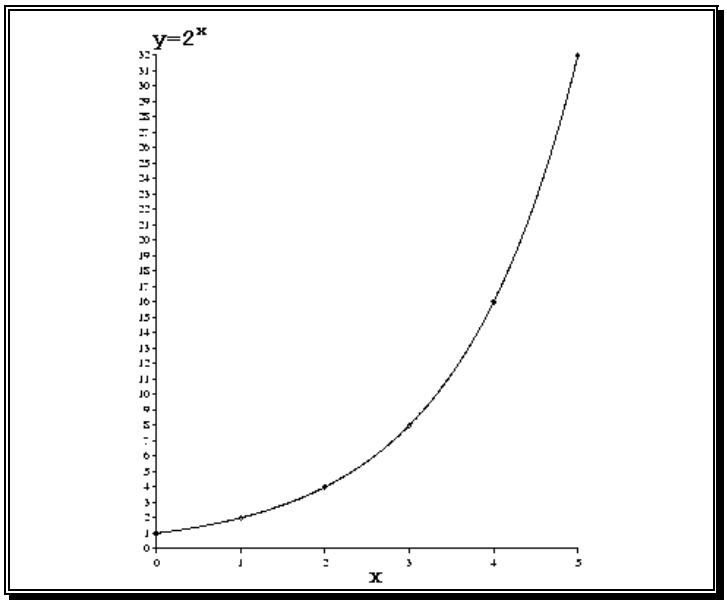
$$2^2 \times 2^3 = 2^{2+3} = 2^5. \qquad (28)$$

Il numero "2" si dice base della potenza.

Se eleviamo l'esponente a un altro esponente i valori si moltiplicano:

$$(2^2)^3 = 2^2 \times 2^2 \times 2^2 = 2^{2+2+2} = 2^6. \qquad (29)$$

La funzione esponenziale $y = 2^x$ e' mostrata in questo grafico[3]:



[3] Il programma usato per creare questo grafico e' reperibile via ftp anonimo dal file:

ftp://ftp.ncifcrf.gov/pub/delila/expgraph.p

Ora, pensiamo di avere un numero e di voler sapere quante volte dobbiamo moltiplicare un 2 per ottenere questo numero.
Per esempio, poniamo di usare 2 come base, quanti 2 devono essere moltiplicati tra' loro per ottenere 32?
Cioe' vogliamo risolvere questa equazione:

$$2^B = 32. \qquad (30)$$

Ovviamente, $2^5 = 32$, cosi' $B = 5$.

Per riuscire a far questo, i matematici hanno costruito una nuova funzione chiamata Logaritmo:

$$\log_2(32) = 5. \qquad (31)$$

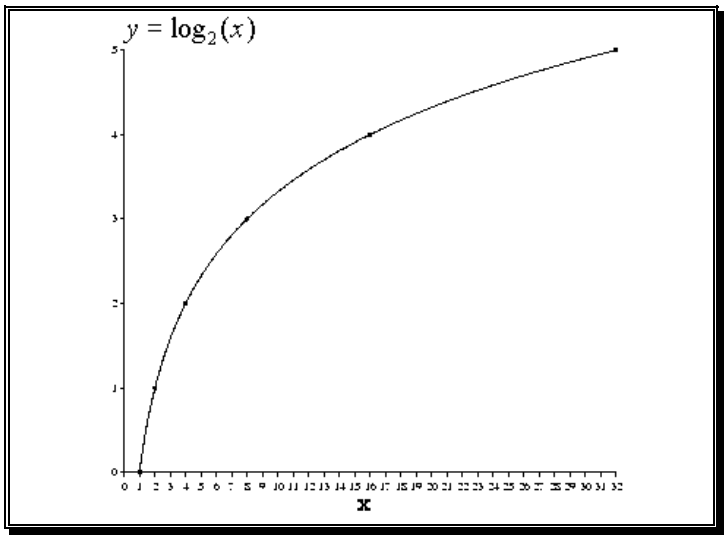
Che si legge "il logaritmo in base 2 di 32 e' 5".
E' la "funzione inversa" dell'elevamento a potenza:

$$2^{\log_2(\alpha)} = \alpha \qquad (32)$$

e.....

$$\log_2(2^{\alpha}) = \alpha \qquad (33)$$

La funzione logaritmica $y = \log_2(x)$ e' mostrata in questo grafico[4]:



[4] Il programma usato per creare questo grafico e' reperibile via ftp anonimo dal file:

ftp://ftp.ncifcrf.gov/pub/delila/expgraph.p

Questo grafico e' stato creato semplicemente scambiando la x con la y nel grafico esponenziale, che sarebbe come ruotare quest'ultimo su di un asse posto a 45°. Da notare in particolare che $\log_2(1) = 0$ e $\log_2(0) = -\infty$

La legge dell'Addizione.

Consideriamo questa equazione:

$$2^{a+b} = 2^a \times 2^b \qquad (34)$$

che e' soltanto una generalizzazione dell'equazione (*28*).

Prendiamo il logaritmo di entrambi i membri:

$$\log_2(2^{a+b}) = \log_2(2^a \times 2^b) \qquad (35)$$

Elevamento a potenza e logaritmo sono l'una l'inversa dell'altra, percio' possiamo "far crollare" la parte sinistra e cosi' ottenere:

$$a+b=\log_2(2^a\times 2^b) \qquad (36)$$

Ora facciamoci furbi, poniamo $\log_2(x)=a$ e $\log_2(y)=b$:

$$\log_2(x)+\log_2(y)=\log_2(2^{\log_2(x)}\times 2^{\log_2(y)}) \qquad (37)$$

Nuovamente, elevamento a potenza e logaritmo sono l'una l'inversa dell'altra, percio' possiamo "far crollare" le due potenze nella parte destra:

$$\log_2(x)+\log_2(y)=\log_2(x\times y) \qquad (38)$$

Questa e' la proprieta' additiva a cui Shannon era interessato.

La regola del "tirare in avanti". Dall'equazione (***32***):

$$\alpha=2^{\log_2(\alpha)} \qquad (39)$$

Eleviamo entrambi i membri ad u:

$$\alpha^u=(2^{\log_2(\alpha)})^u \qquad (40)$$

Ora possiamo unire gli esponenti moltiplicandoli come abbiamo fatto in (***29***):

$$\alpha^u=2^{u\log_2(\alpha)}. \qquad (41)$$

Infine, prendiamo il logaritmo in base 2 di entrambi i membri e facciamo "crollare" la parte destra:

$$\log_2(\alpha^u)=u\log_2(\alpha) \qquad (42)$$

Che puo' essere ricordata come la regola che permette di "tirare" l'esponente avanti dall'interno del logaritmo.

Come convertire tra basi diverse. Le calcolatrici e i computers solitamente non calcolano il logaritmo in base 2, ma possiamo usare una furbizia per convertire nella base desiderata (nel nostro caso la base 2) il risultato ottenuto in una base qualsiasi. Cominciamo ponendo:

$$x=\log_a(a)/\log_a(b) \qquad (43)$$

Modifichiamola cosi':

$$\log_a(a) = x \log_a(b) \qquad (44)$$

Ora usiamo un "tira avanti ribaltato" (!):

$$\log_a(a) = \log_a b^x \qquad (45)$$

e lasciamo "cadere" i logaritmi:

$$a = b^x \ . \qquad (46)$$

Prendiamo ora la base dei logaritmi β:

$$\log_\beta(a) = \log_\beta b^x \qquad (47)$$

che si semplifica come:

$$\log_\beta(a) = x. \qquad (48)$$

Ma noi sappiamo, dall'equazione (**43**), che x e':

$$\log_\beta(a) = \log_a(a) \Big/ \log_a(\beta) \qquad (49)$$

La regola di conversione per ottenere il logaritmo in base 2 partendo da una qualsiasi base x e':

$$\log_2(a) = \log_x(a) \Big/ \log_x(2) \qquad (50)$$

Notiamo che dal momento che la x non appare nella parte sinistra dell'equazione non importa che tipo di logaritmo abbiamo a disposizione, perche' possiamo sempre ottenerlo in un'altra base usando questa equazione! Provate questo esempio sulla vostra calcolatrice:

$$\log_2(32) = \frac{\log_{qualsiasi!}(32)}{\log_{qualsiasi!}(2)} \ . \qquad (51)$$

Otterrete '5'.

Trucchetti con le potenze di 2. Nei calcoli impariamo che la base naturale dei logaritmi e'
$\varepsilon = 2.718281828459045.....$
Che impressione vi fa memorizzare questo numero? Notate che dopo il 2.7 abbiamo due 1828 seguiti da un triangolo 45°-90°-45°.

I calcoli in questa base possono essere fatti molto facilmente da un computer o calcolatrice, ma per molte persone risultano difficili da fare mentalmente.
In contraddizione, le potenze di 2 sono facili da memorizzare e ricordare:

Scelte	Bits
M	B
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1024	10

dove $2^B = M$ e $\log_2(M) = B$.

Possiamo utilizzare questa tabella ed un trucchetto per dare una veloce, sia pur approssimata, stima di logaritmi di numeri piuttosto alti.
Notiamo che

$$2^{10} = 1024 \approx 1000 = 10^3 \quad (52)$$

Percio' per calcolare il logaritmo in base 2 di 4×10^6, procediamo cosi':

$$.\log_2(4\times10^6) = \log_2(4) + \log_2(10^6) \quad (53)$$

$$= 2 + \log_2(10^3 \times 10^3) \quad (54)$$

$$= 2 + \log_2(10^3) + \log_2(10^3) \quad (55)$$

$$\approx 2 + \log_2(2^{10}) + \log_2(2^{10}) \quad (56)$$

$$\approx 2 + 10 + 10 \quad (57)$$

$$\approx 22 \quad (58)$$

Il valore vero e' 21.93 !!

Simone Baldi, Martedi' 12 Marzo 1996.