Corso di *Data Security & Privacy*[*]
Esercizi riepilogativi su Teoria dell'Informazione, codici di compressione, canali

Michele Boreale
Università di Firenze
Dipartimento di Statistica, Informatica, Applicazioni
michele.boreale@unifi.it

1 Esercizi di riepilogo

1.1 Teoria dell'Informazione

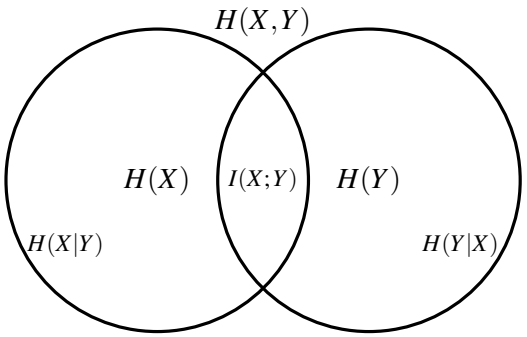
1. Due variabili aleatorie X e Y assumono entrambe valori nell'insieme $\{0,1\}$. La loro distribuzione congiunta $p(x,y)$ è data dalla seguente tabella:

	$y=0$	$y=1$
$x=0$	1/3	1/3
$x=1$	0	1/3

Calcolare le seguenti quantità: $H(X)$, $H(Y)$, $H(X|Y)$, $H(Y|X)$, $H(X,Y)$, $H(Y)-H(Y|X)$ e $I(X;Y)$. Individuare quindi ciascuna di queste quantità in un opportuno diagramma di Venn.

Traccia. La tabella rappresenta la distribuzione congiunta $p(x,y)$, cioè i valori $p(0,0)$, $p(0,1)$, $p(1,0)$, $p(1,1)$. Da questi è facile ricavare le distribuzioni marginali: ad esempio, $p(x) = p(x,0)+p(x,1)$, per $x = 0,1$. Altrettanto facile ricavare le condizionali: ad esempio, $p(x|0)$ si ottiene fissando la colonna $y = 0$ e normalizzando, dunque $p(x = 0|y = 0) = 1$ e $p(x = 1|y = 0) = 0$. Dunque è facile, una volta calcolate ad esempio $p(x)$ e $p(y|x)$ calcolare $H(X)$ e $H(Y|X)$. Le altre entropie richieste si ottengono applicando la chain rule. Il diagramma di Venn ha la forma seguente.

[*]Laurea Magistrale in Informatica, Università di Firenze, A.A. 2018-2019.



2. Sia $X \sim p(x)$, con $\mathcal{X} = \{1,2,...,m\}$. Sia dato un insieme $S \subseteq \mathcal{X}$, con $p(S) = \alpha$. Sia Y la risposta alla domanda "$X \in S$?". Calcolare quanta informazione questa domanda dà su X, cioè il valore di $I(X;Y)$.

   Supponiamo che $\mathcal{X} = \{1,2,3,4,5\}$, con la seguente distribuzione: $\mathbf{p} = (0,2,\ 0,1,\ 0,1,\ 0,25,\ 0,35)$. Scegliere S che massimizzi l'informazione fornita dalla risposta Y.

   Traccia. La risposta è $H(\alpha)$. Per il secondo quesito, si tenga presente $0 \leq H(\alpha) \leq 1$, e le condizioni per il massimo.

3. Si veda H come funzione da $\mathbb{S}^{(n)}$ a $\mathbb{R}$, dove $\mathbb{S}^{(n)}$ è l'insieme dei vettori di probabilità a n componenti. Dire quali sono i punti di minimo e di massimo di H su $\mathbb{S}^{(n)}$.

   Traccia. Si tenga presente $0 \leq H(X) \leq \log|X|$, e le condizioni per le quali si realizzano le uguaglianze.

4. Siano X,Y due v.a. Si dimostri che $H(Y|X) = 0$ se e solo se esiste una funzione $f : \mathcal{X} \to \mathcal{Y}$ tale che $Y = f(X)$.

   Traccia. Se $Y = f(X)$, espandere $H(Y|X)$ tenendo presente che $H(Y|X = x) = H(f(X)|X = x) = 0$ per ogni x (perché?). Viceversa, se $H(Y|X) = 0$, allora deve valere $H(Y|X = x) = 0$ per ogni x nel supporto di X: questo vuol dire che $p(y|X = x)$ concentra tutta la probabilità su un singolo elemento, y_x: poniamo $f(x) = y_x$. Per gli elementi x non nel supporto di X, si può porre $f(x)$ uguale ad un y arbitrario.

5. Un'urna contiene delle palline rosse, blu e bianche, in egual misura. Sia X l'estrazione dall'urna di n palline con rimpiazzo e Y l'estrazione dall'urna di n palline senza rimpiazzo. Chiarire, dimostrandolo, quale tra X e Y ha l'entropia maggiore.

   Traccia. Si noti che non è necessario calcolare esplicitamente né la distribuzione né l'entropia di X o Y; procedere al confronto per induzione su n.

1.2 Codici di compressione

1. Si consideri la v.a. X che assume i valori $x_1, ..., x_7$ con le probabilità 0,49, 0,26, 0,12, 0,04, 0,04, 0,03 e 0,02, rispettivamente. Si trovi un codice binario di Huffman per X e se ne calcoli la lunghezza media.

   Si ripeta l'esercizio per la variabile che assume i valori $x_1, ..., x_8$ con le probabilità 0,40, 0,26, 0,12, 0,09, 0,04, 0,04, 0,03 e 0,02, rispettivamente.

2. In un gioco di domande e risposte binarie, un giocatore deve individuare un segreto, estratto secondo una v.a. $X \sim p(x)$, a valori in $\{x_1, ..., x_m\}$. Il giocatore può porre più domande, e quando individua la risposta corretta x_i, riceve un premio prefissato $v_i \geq 0$. Tuttavia, ogni domanda posta costa 1 al giocatore. Chiamiamo *payoff* la differenza tra il premio guadagnato e il costo complessivo delle domande fatte dal giocatore.

   (a) Assumendo che il giocatore usi una strategia ottimale, dare una stima S del payoff *medio* P, a meno di uno. Cioè individuare S tale che valga $S - 1 < P \leq S$.

   (b) Si dimostri che se le v_i soddisfano la disuguaglianza di Kraft allora S è sempre non negativo – quindi il giocatore mediamente non ci rimette.

   (c) Nell'ipotesi che valga Kraft, fissata la distribuzione $\mathbf{p} = (p_1, ..., p_m)$, dire qual è la scelta di $\mathbf{v} = (v_1, ..., v_m)$ che *minimizza* S, e quindi il payoff medio per il giocatore.

   Traccia. Per il punto (a), si tenga presente che una strategia domanda/risposta è equivalente ad un albero binario etichettato, e dunque ad un codice istantaneo: una strategia ottimale corrisponde dunque ad un codice ottimo, la cui lunghezza media sarà L. Questo è dunque anche il numero medio di domande posto, e il payoff vale perciò $P = \sum_i v_i p_i - L$. Dal teorema di source coding di Shannon sappiamo che $H(\mathbf{p}) \leq L < H(\mathbf{p}) + 1$, perciò:

$$\sum_i v_i p_i - H(\mathbf{p}) - 1 < P \leq \sum_i v_i p_i - H(\mathbf{p}) .$$

   Possiamo quindi porre $S \stackrel{\text{def}}{=} \sum_i v_i p_i - H(\mathbf{p})$.

   Per il punto (b), esprimere il guadagno medio del giocatore, $\sum_i p_i v_i$, come una cross-entropy $H(\mathbf{p}||\mathbf{q})$ per una opportuna $\mathbf{q}$. Più precisamente, posto $q_i \stackrel{\text{def}}{=} 2^{-v_i}/c$, dove $c = \sum_i 2^{-v_i}$, si verifica che

$$p_i v_i = p_i(\log(\frac{1}{q_i}) + \log\frac{1}{c}) .$$

Da cui sommando su tutti gli i, e ponendo $\mathbf{q} = (q_1, ..., q_m)$

$$\begin{aligned}\sum_i p_i v_i &= \sum_i p_i \log(\frac{1}{q_i}) + \log\frac{1}{c} \\ &= H(\mathbf{p}||\mathbf{q}) + \log\frac{1}{c} \\ &= H(\mathbf{p}) + D(\mathbf{p}||\mathbf{q}) + \log\frac{1}{c}\end{aligned}$$

dove nell'ultimo passaggio abbiamo applicato la linking identity. Per definizione della stima del payoff, S, otteniamo

$$S = D(\mathbf{p}||\mathbf{q}) + \log\frac{1}{c}. \tag{1}$$

Se le v_i soddisfano la disuguaglianza di Kraft, abbiamo che $c \leq 1$. Inoltre, per la disuguaglianza di Gibbs, $D(\mathbf{p}||\mathbf{q}) \geq 0$. Otteniamo dunque che $S \geq 0$.

Per il punto (c), vediamo da (1) che S è minimizzata, assumendo sempre che $\mathbf{v}$ rispetti Kraft, se $\mathbf{q} = \mathbf{p}$, quando otteniamo $S = 0$. Questo vuol dire

$$v_i = -\log p_i$$

per ogni $p_i > 0$, e v_i un numero arbitrario altrimenti.

1.3 Canali

1. Ricorrendo ai risultati enunciati in classe, si calcoli la capacità del canale con $\mathcal{X} = \mathcal{Y} = \{0,1,2\}$ e

$$p(y|x) = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \\ 1/2 & 0 & 1/2 \end{bmatrix}.$$

Traccia. La matrice è debolmente simmetrica.

2. Un canale prende in input $X \in \{0,1\}$ e dà in uscita un simbolo $Y = X + Z$, dove $Z \in \{0,a\}$ è una fonte di rumore indipendente da X, e a è un parametro reale fissato. Si calcoli la capacità di questo canale. Conviene distinguere vari casi per a: $a = 0$, $a = 1$, $a = -1$ e $a =$ altri valori.

Traccia. Il caso $a = 0$ è banale, e la capacità vale $C = 1$. Nel caso $a = 1$, abbiamo $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{0,1,2\}$ e la forma della matrice è

$$p(y|x) = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 \end{bmatrix}.$$

Per una generica v.a. X data come ingresso al canale e distribuita secondo $\mathbf{p} = (\lambda, 1-\lambda)$, calcoliamo $I(X;Y)$:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(\lambda) - H(X|Y=1)\Pr(Y=1) \\ &= H(\lambda) - H(X|Y=1)\frac{1}{2} \\ &= H(\lambda) - H(\lambda)\frac{1}{2} \\ &= \frac{H(\lambda)}{2}. \end{aligned}$$

Considerando la forma di $H(\lambda)$, abbiamo che $I(X;Y)$ è massima se $\lambda = \frac{1}{2}$, quando $H(\lambda) = 1$. Dunque otteniamo $C = \frac{1}{2}$.

Il caso $a = -1$ è del tutto analogo. Infine si vede facilmente che per gli altri valori di a, la matrice è tale per cui $C = 1$.

2 Esercizi di approfondimento

2.1 Sequenze tipiche

Fissiamo $n \geq 1$. Sia $B = x_1, x_2, ..., x_n$ una sequenza di n lettere estratte da un alfabeto $\mathcal{X}$. Il *tipo* della sequenza B, denotato con t_B, è la distribuzione empirica che essa determina su $\mathcal{X}$. Ovvero, per ogni $x \in \mathcal{X}$, denotando con $f(x)$ la frequenza di x in B, si pone:

$$t_B(x) \stackrel{\text{def}}{=} \frac{f(x)}{n}.$$

(a) Sia p una distribuzione su $\mathcal{X}$, e supponiamo che $B = X_1, ..., X_n$, con le X_i estratte i.i.d. secondo p. Dimostrare che la probabilità dell'intera sequenza B sotto p puo essere scritta come:

$$p(B) = 2^{-n(H(t_B)+D(t_B||p))}.$$

(b) Sia t il tipo di una certa sequenza. Consideriamo tutte le sequenze di lunghezza n che hanno lo stesso tipo t, ovvero

$$\mathcal{B}_t \stackrel{\text{def}}{=} \{B : t_B = t\}.$$

Sfruttando il punto precedente, dimostrare che

$$|\mathcal{B}_t| \leq 2^{nH(t)}$$

(Si noti che deve essere $1 \geq t(\mathcal{B}_t)$, da cui ...).

(c) Dato un certo tipo t, e una qualsiasi distribuzione di probabilità p su $\mathcal{X}$, dimostrare che è possibile maggiorare la probabilità che venga estratta una qualsiasi sequenza di tipo t come segue

$$p(\mathcal{B}_t) \leq 2^{-nD(t||p)}.$$

2.2 Strategie e codici

In un gioco sono date n monete, etichettate con numeri da 1 a n. Il giocatore A, non visto dal giocatore B, può:

(a) sostituire una moneta i scelta a caso con una moneta falsa, identica all'aspetto ma più leggera; oppure

(b) sostituire una moneta i scelta a caso con una moneta falsa, identica all'aspetto ma più pesante; oppure

(c) non effettuare alcuna sostituzione.

Il giocatore B deve individuare l'azione compiuta da A. Allo scopo, B ha a sua disposizione una bilancia con cui confrontare il peso di gruppi di monete: ad ogni pesata, la bilancia può dire che è più pesante il piatto di destra, oppure quello di sinistra, oppure che i due piatti hanno ugual peso.

(a) Dare un esempio di strategia con $n = 5$ monete. La strategia corrisponde ad un albero ternario, dove: i nodi interni sono etichettati con coppie del tipo (S,T), con S e T sottoinsiemi disgiunti di monete che rappresentano il contenuto dei piatti sinistro e destro; e gli archi sono etichettati con i simboli L, R oppure $=$, che rappresentano i possibili risultati di una pesata. Le foglie sono etichettate con le possibili mosse del giocatore A, tratte dall'insieme

$$\mathcal{X} = \{(1,+),(1,-),(2,+),(2,-),...,(n,+),(n,-),nil\}$$

dove $(i,+)$ indica la sostituzione della moneta i con una più pesante, $(i,-)$ la sostituzione della moneta i con una più leggera, e *nil* indica nessuna sostituzione.

(b) Calcolare la lunghezza (= n. di pesate) *massima* (cioè l'altezza dell'albero) e quella *media* della strategia definita al punto precedente.

(c) Dare un limite inferiore, in funzione di n, al numero di *massimo* di pesate in una qualsiasi strategia che porta ad individuare l'azione compiuta da A.

(d) Dare un limite inferiore, in funzione di n, al numero di *medio* di pesate in una qualsiasi strategia che porta ad individuare la mossa compiuta da A.

Traccia. Per il punto (c), posto k uguale all profondità dell'albero che definisce la strategia, sfruttare la relazione che esiste tra profondità di un albero ternario e numero delle sue foglie. In alternativa, vedere l'azione compiuta da A come una v.a. X che può assumere uno tra i $2n+1$ valori di $\mathcal{X}$ con probabilità uniforme; e le pesate fatte da B come una sequenza di v.a. ternarie $Y_1,...,Y_k$. Per definizione, $H(X|Y_1,...,Y_k) = 0$. Quindi usare un po' di (dis)uguaglianze note per H per dare una limitazione inferiore per k.

Per il punto (d), notare che ogni strategia, descritta come albero, corrisponde ad un codice istantaneo *ternario* C, cioè un codice su un alfabeto di codifica di tre lettere $\{L,R,=\}$ (anziché $\{0,1\}$). La lunghezza media della strategia è dunque la lunghezza media del codice, $L(C)$. I risultati visti per i codici binari si generalizzano al caso ternario semplicemente considerando la base 3, anziché 2. In particolare, indicando con $H_3(\cdot)$ la funzione entropia con i log presi in base 3, avremo che

$$L(C) \geq H_3(\frac{1}{2n+1}, \ldots, \frac{1}{2n+1}) = \cdots,$$

2.3 La capacità del piccione viaggiatore

Un comandante che assedia un forte usa dei piccioni viaggiatori per comunicare con gli alleati. Ogni piccione porta una lettera (8 bit). Viene liberato un piccione ogni 5 minuti. Ogni piccione impiega tre minuti per raggiungere la destinazione. Si calcoli la capacità in bit/ora di questo collegamento, nei seguenti due casi.

(a) I piccioni raggiungono tutti la destinazione.

(b) I nemici abbattono una frazione α dei piccioni, e sostituiscono ogni piccione abbattuto con uno che porta una lettera scelta a caso. Nei calcoli, non è necessario espandere la funzione entropia binaria $B(\lambda) = H(\lambda, 1-\lambda)$.

Traccia. Per il caso (b), si usi il risultato sui canali simmetrici dimostrato in classe. Può risultare utile tenere presente la seguente uguaglianza, che vale per una generica distribuzione $\mathbf{p} = (p_1, ..., p_n)$:

$$H(p_1, ..., p_n) = H(p_1) + (1-p_1)H(\frac{p_2}{1-p_1}, ..., \frac{p_n}{1-p_1}).$$

Per la dimostrazione di quest'ultima, sia X una v.a. distribuita come $\mathbf{p}$ e si definisca la v.a. ausiliaria $Y = 1$ se $X = x_1$, $Y = 0$ altrimenti: si espanda $H(X,Y)$ usando la chain rule.

3 Esercizi di programmazione

3.1 Test di ipotesi

Dato un testo (sequenza di lettere) $x_1, x_2, x_3, ..., x_n$, si vuole stabilire quale di due ipotesi alternative sia quella corretta:

- H_0: testo casuale;
- H_1: testo in lingua Inglese, o ottenuto da un testo in lingua Inglese tramite shift encryption.

Programmare una procedura Python che risolve tale problema applicando un test di verifica di ipotesi. Allo scopo, considerare la sequenza data $x_1, x_2, ..., x_n$ come ottenuta da estrazioni i.i.d. Come riferimento, si usi la tabella delle frequenze alla voce *letter frequency* di Wikipedia. Si ignorino spazi e punteggiatura. La procedura restituisce non solo l'ipotesi ritenuta corretta, ma anche una stima delle probabilità di errore α_n e β_n. Testare la procedura su 100 blocchi di testo, di cui la metà verificano H_0 e l'altra metà H_1. Riassumere i risultati ottenuti in una tabella, indicando la percentuale di falsi positivi e falsi negativi ottenuta e confrontandoli con le stime teoriche di α_n e β_n.

3.2 Codici di Huffman

Programmare in Python l'algoritmo di Huffman. La procedura accetta una lista di lettere (l'alfabeto), la relativa distribuzione di probabilità e restituisce un codice istantaneo ottimo per la distribuzione data. Programmare anche una procedura di decodifica che, preso come argomento un generico codice prefix-free (non necessariamente Huffman) e una stringa binaria, proceda alla decodifica della stringa.

3.3 Codici Lempel-Ziv (LZ)

Programmare in Python l'algoritmo di codifica e quello di decodifica di LZ, come visti a lezione. Mettere a confronto i risultati di Huffman e LZ su 10 blocchi di testo di lingua Inglese, di lunghezza almeno 1000 caratteri ciascuno. In particolare, riportare qual è la percentuale di compressione media ottenuta nei due casi.