(F1X) Teoria dell'Informazione e della Trasmissione

Codifica sorgente: definizioni

Docente: *Nicolò Cesa-Bianchi* versione 4 marzo 2016

I messaggi da trasmettere sono generati da un'entità astratta chiamata sorgente. Sia $\mathcal{X}$ l'insieme finito di simboli che compongono i messaggi generati dalla sorgente. Un messaggio $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ di lunghezza n è una sequenza di n simboli sorgente. Una funzione di codifica mappa simboli sorgente in parole di codice. Una parola di codice è una sequenza di numeri dall'insieme $\{0, \ldots, D-1\}$ dei simboli di codice, dove $D > 1$ è la base del codice. Per esempio, con $D = 2$ otteniamo i codici binari. In questo senso, possiamo rappresentare una funzione di codifica per un codice sorgente con

$$c : \mathcal{X} \to \{0, \ldots, D-1\}^+$$

dove $\{0, \ldots, D-1\}^+$ rappresenta l'insieme delle sequenze su $\{0, \ldots, D-1\}$ di lunghezza maggiore o uguale a uno. Formalmente,

$$\{0, \ldots, D-1\}^+ = \bigcup_{n=1}^{\infty} \{0, \ldots, D-1\}^n \ .$$

Per esempio, $(2, 1)$ e $(4, 7, 3)$ appartengono entrambe a $\{0, \ldots, D-1\}^+$ per $D = 8$.

Esempio 1 *Dato $\mathcal{X} = \{\heartsuit, \diamondsuit, \clubsuit, \spadesuit\}$, un esempio di codice binario $c : \mathcal{X} \to \{0, 1\}^+$ è il seguente:*

$$c(\heartsuit) = 0 \quad c(\diamondsuit) = 010 \quad c(\clubsuit) = 01 \quad c(\spadesuit) = 10 \ .$$

Dato che l'obiettivo di un codice sorgente è massimizzare la compressione, siamo interessati a misurare la quantità $\ell_c(x)$ definita come la lunghezza della parola di codice $c(x)$ per il simbolo $x \in \mathcal{X}$. Nell'esempio precedente, $\ell_c(\diamondsuit) = 3$. L'obiettivo di un codice sorgente è quello di minimizzare la lunghezza media della parole di codice utilizzate per codificare i simbolo sorgente.

L'intuizione alla base della costruzione di codici sorgente è la stessa dell'alfabeto Morse: utilizzare parole di codice corte per simboli generati frequentemente dalla sorgente. Per poterla analizzare in modo rigoroso dobbiamo creare un modello formale della sorgente. La proposta di Shannon è quella di definire una distribuzione di probabilità p fissata su simboli sorgente e quindi assumere che $p(x)$ rappresenti la probabilità che la sorgente generi il simbolo $x \in \mathcal{X}$. Un **modello di sorgente** è quindi definito dalla coppia $\langle \mathcal{X}, p \rangle$.

Si noti che, in realtà, la quantità di interesse qui è la distribuzione di probabilità sui messaggi $\boldsymbol{x}$ piuttosto che sui simboli x, dato che i messaggi sono gli oggetti che vogliamo trasmettere. Data una distribuzione p sui simboli $\mathcal{X}$ definiamo quindi una distribuzione P_n sui messaggi $\mathcal{X}^n$ di lunghezza n come

$$P_n(x_1, \ldots, x_n) = p(x_1) \times \cdots \times p(x_n) \ .$$

Questa definizione di P_n corrisponde ad assumere che la sorgente generi un messaggio attraverso estrazioni indipendenti di simboli. In generale, però, questa assunzione non è molto plausibile.

Per esempio, pensiamo ad un messaggio di testo in italiano dove i simboli sorgente sono le lettere dell'alfabeto compreso spazi e punteggiatura. Chiaramente, ci sono delle forti dipendenze fra una lettera del messaggio e le lettere che le stanno attorno e tali dipendenze non sono catturate dalla P_n definita come sopra. D'altra parte, l'analisi matematica è molto facilitata dall'assunzione di indipendenza. Nel seguito, assumeremo quindi l'indipendenza dei simboli sorgente, tenendo però in mente che codici sorgente più realistici e sofisticati sono ottenuti senza questa assunzione.

D'ora in poi identifichiamo un simbolo emesso dalla sorgente tramite la variabile casuale $X : \mathcal{X} \to \mathbb{R}$. Fissato D (base del codice) indichiamo con $\mathcal{D}$ l'insieme $\{0, \ldots, D-1\}$ dei simboli di codice con base D. Quindi una funzione di codifica, o codice, è una funzione del tipo $c : \mathcal{X} \to \mathcal{D}^+$.

Siamo pronti per definire formalmente il problema della codifica sorgente: dato un modello di sorgente $\langle \mathcal{X}, p \rangle$ e una base $D > 1$ trovare un codice $c : \mathcal{X} \to \mathcal{D}^+$ tale che il valore atteso

$$\mathbb{E}\big[\ell_c\big] = \sum_{x \in \mathcal{X}} \ell_c(x)\, p(x) \tag{1}$$

della lunghezza di parola di codice sia minimo.

Formulato in questi termini, il problema della codifica sorgente si presta ad una soluzione banale e inutile. Infatti è ovvio che il codice $c : \mathcal{X} \to \mathcal{D}^+$ tale che $c(x) = 0$ per ogni $x \in \mathcal{X}$ minimizza $\mathbb{E}\big[\ell_c\big]$ per ogni modello di sorgente. Quindi, bisogna imporre delle limitazioni sulla classe di codici che vogliamo utilizzare per risolvere (1).

Una prima limitazione è la seguente. Un codice $c : \mathcal{X} \to \mathcal{D}^+$ è **non singolare** se a simboli sorgente distinti corrispondono parole di codice distinte. Formalmente, per ogni $x, x' \in \mathcal{X}$ tale che $x \neq x'$ vale $c(x) \neq c(x')$. In altre parole, la non singolarità del codice è equivalente all'iniettività della funzione di codifica. Questa è chiaramente una proprietà minimale per un codice utilizzabile in pratica.

Ora introduciamo un concetto naturale: quello di **estensione di un codice**. L'estensione serve a definire in modo semplice la parola di codice associata ad un messaggio di una data lunghezza, ovvero ad una sequenza di simboli sorgente. Dato un codice $c : \mathcal{X} \to \mathcal{D}^+$, la sua estensione è la funzione $C : \mathcal{X}^+ \to \mathcal{D}^+$ definita come $C(x_1, \ldots, x_n) = c(x_1) \cdots c(x_n)$, dove $c(x_1) \cdots c(x_n)$ indica la sequenza ottenuta giustapponendo le parole di codice $c(x_1), \ldots, c(x_n)$.

Esempio 2 *L'estensione C del codice definito nell'Esempio 1 è tale che*

$$C(\heartsuit, \spadesuit, \clubsuit) = c(\heartsuit)c(\spadesuit)c(\clubsuit) = 01001 \ .$$

La proprietà di non singolarità non è abbastanza forte per garantire che essa venga ereditata anche dall'estensione di un codice. Infatti, l'estensione nell'Esempio 2 è tale che

$$C(\diamondsuit) = C(\clubsuit, \heartsuit) = C(\heartsuit, \spadesuit) = 010 \ .$$

Quindi mentre il codice c dell'Esempio 1 è non singolare la sua estensione C non lo è.

Motivati da questo esempio, introduciamo la nozione di codice **univocamente decodificabile**, ovvero di codice la cui estensione è non singolare. Formalmente, c è univocamente decodificabile

se C è una funzione iniettiva. In pratica questa proprietà permette di decodificare i messaggi. Infatti, se c è univocamente decodificabile allora per ogni $\boldsymbol{y} \in \mathcal{D}^+$ trovo al più un unico messaggio $\boldsymbol{x} \in \mathcal{X}^+$ (la decodifica di $\boldsymbol{y}$) tale che $C(\boldsymbol{x}) = \boldsymbol{y}$. La verifica per determinare se un dato codice c sia univocamente decodificabile è realizzata dall'algoritmo di Sardinas-Patterson in tempo $\mathcal{O}(mL)$, dove m è il numero delle parole di codice e L è la somma delle loro lunghezze.