Laboratorio di Teoria dell'Informazione

Simulazione di un codificatore aritmetico binario

Obiettivo: l'obiettivo di questa esercitazione è simulare un codificatore aritmetico per osservarne rispettivamente l'efficienza di codifica, i problemi di under-flow e la soluzione attraverso regole di "rinormalizzazione" degli intervalli.

Per realizzare gli esperimenti descritti nel seguito progettare un simulatore con le seguenti funzionalità:

1. procedura che emette simboli di una sorgente binaria X con P(X=0)=p e P(X=1)= 1-p
2. funzione per la codifica dei simboli binari attraverso la procedura di selezione iterativa dell'intervallo di probabilità corrispondente al simbolo emesso dalla sorgente
    a. si rappresentino gli intervalli (rappresentati come estremi sinistro e destro, oppure come un estremo e ampiezza dell'intervallo) e le probabilità in formato floating point a doppia precisione;
    b. in corrispondenza ad ogni emissione si aggiorni l'intervallo corrispondente al simbolo codificato;
    c. inserire un controllo sull'ampiezza raggiunta dall'intervallo selezionato in modo da segnalare la condizione di under flow del formato numerico scelto
    d. al termine della codifica si stimi la lunghezza L della parola binaria prodotta dal codificatore aritmetico usando la seguente approssimazione

$$L = \left\lceil log_2(\frac{1}{ampiezza}) \right\rceil + 1$$

e si determini il rate di codifica (misurato in bit per simbolo).

Esperimento 1

Si valuti l'efficienza del codificatore aritmetico al variare del numero N di simboli binari emessi dalla sorgente. In particolare si confronti l'efficienza sperimentale data da L/N con l'entropia teorica della sorgente. Si ripeta l'esperimento per vari valori di p.

Nei vari casi presi in esame valutare il numero massimo di simboli di sorgente che risulta possibile codificare prima di rivelare under flow nella rappresentazione numerica degli intervalli.

Esperimento 2

Si migliori la procedura di codifica inserendo le regole di "rinormalizzazione" R1 e R2. Tali regole permettono di operare la codifica in precisione finita raddoppiando l'intervallo di probabilità scelto ogni qual volta

1. l'estremo destro è minore di 0.5 (R1)

2. l’estremo sinistro è maggiore di 0.5 (R2)

Ogni operazione di rinormalizzazione permette inoltre di emettere un bit della parola codificata. In questo caso occorre quindi stimare il numero di bit prodotti dalla compressione come:

$$L = cnt_{R1} + cnt_{R2} + \left\lceil log_2\left(\frac{1}{ampiezza}\right)\right\rceil + 1$$

dove *cnt_R1* e *cnt_R2* indicano il numero di volte in cui è stata effettuata la rinormalizzazione R1 e R2.

Verificare che in presenza di R1 e R2 è possibile codificare sequenze molto più lunghe senza cadere nella condizione di underflow. Valutare l’efficienza di codifica ottenuta su sequenze lunghe.

Esperimento 2

Si simuli un codificatore aritmetico adattativo che stima il valore di $\tilde{p}$ dai dati precedentemente codificati. Si modifichi la procedura di codifica in modo che:

a) ad ogni iterazione aggiorni la probabilità secondo la relazione

$$\tilde{p} = \frac{1 + cnt\_0}{2 + cnt\_TOT}$$

dove *cnt_0* rappresenta i simboli 0 codificati precedentemente e *cnt_TOT* rappresenta il numero totale di simboli codificati. Alla prima iterazione entrambi i contatori sono nulli e la prima stima di probabilità corrisponde a $\tilde{p} = 0.5$.

Si valuti l’efficienza di codifica della versione adattativa e la si confronti con l’efficienza ottenuta al punto 2 per valori di p=0.7, 0.9 e 0.98.

Esperimento 3

Si utilizzi il codificatore aritmetico sviluppato al punto 3 in presenza di due contesti, ovvero usando due modelli diversi di sorgente. In particolare si codifichino 2 sorgenti (non necessariamente con lo stesso numero di simboli) con due statistiche diverse (ad esempio p=0.6 e p=0.9). Per ogni contesto è necessario mantenere i rispettivi contatori la stima della statistica dai dati. Osservare il comportamento dell’encoder e valutare sperimentalmente e analiticamente l’efficienza ottenuta in termini di rate di codifica.