Codifica di Huffman

Metodo per costruire codici prefissi ottimali (ma non sempre assolutamente ottimi) con approccio bottom-up.

Si costruisce un albero binario in cui le foglie contengono i simboli e la loro frequenza, e ciascun nodo interno mantiene un insieme dei caratteri che si trovano nei suoi discendenti e la somma delle loro frequenze. L'albero viene creato come segue:

- si inserisce in una lista una foglia per ciascun carattere (con la relativa frequenza);
- si uniscono i due nodi con frequenza minore, impostando la somma delle frequenze dei due come frequenza del nuovo nodo, e l'unione delle due liste come suo insieme dei caratteri;
- si ripete il passo precedente finché non si ha una sola radice.

Codifica

Per ciascun carattere da codificare si ripete la seguente procedura, a partire dalla radice dell'albero:

- se il nodo corrente è una foglia si passa al carattere successivo;
- se il carattere è contenuto nell'insieme del figlio sinistro del nodo corrente, si stampa 0 e si scende nel sottoalbero sinistro;
- se è contenuto nel sottoalbero destro, si stampa 1 e si scende a destra.

Decodifica

Per ogni bit in input, a partire dalla radice:

- se il bit è 0 si scende nel sottoalbero sinistro, altrimenti nel destro;
- se il nuovo nodo corrente è una foglia si stampa il carattere associato e si torna alla radice.

Svantaggi

- richiede la conoscenza dell'albero per la decodifica;
- se non si conoscono le frequenze a priori la codifica avviene in due passi (anche se ci sono varianti dinamiche che costruiscono l'albero mentre calcolano le frequenze);
- se le probabilità sono molto sbilanciate il codice è poco efficiente. Per esempio, $p_1 = 0.9 \implies I(\sigma_1) = 0.152$, ma $l_1 = 1$. Si può migliorare considerando l'estensione n-esima della sorgente o con codifiche che non traducono simboli in parole di codice (e.g. codifica aritmetica).