

Algoritmo di Viterbi

Implementa decodifica a massima verosimiglianza per codici convoluzionali. Questo significa selezionare, tra i percorsi possibili nel diagramma a traliccio, quello con la minima distanza dalla sequenza ricevuta.

Ricevuta la sequenza y , l'algoritmo assegna ad ogni transizione da $s_i^{(t)}$ a $s_j^{(t+1)}$ un costo che è la distanza di Hamming tra l'output di quella transizione e la t -esima codeword di y .

Poi, partendo da $s_1^{(1)}$ e aumentando t fino a $\frac{|y|}{N}$, si valutano i percorsi verso gli stati al tempo $t + 1$ che si ottengono a partire dai nodi raggiunti a t ; per ogni i si mantiene solo il percorso di costo minimo che va in $s_i^{(t+1)}$.

L'algoritmo termina quando si esaurisce la sequenza in input. Si mantiene il percorso di costo minimo tra quelli che arrivano all'ultima colonna, e:

- leggendo le etichette si trova la sequenza inviata con maggiore probabilità;
- leggendo gli input degli archi (linea continua o tratteggiata se $k = 1$) si decodifica;
- il costo del cammino è il numero di errori corretti.

Complessità

È necessario memorizzare $2^{k(N-1)}$ cammini sopravvivenenti (tanti quanti gli stati), perciò il costo dell'algoritmo è esponenziale in k ed N , che quindi sono molto limitati.

Variante approssimata

Anziché processare l'intera sequenza ricevuta, si emette il risultato con un *ritardo di decisione* limitato, tipicamente $5N$, che è sufficiente ad ottenere il risultato ottimo nella maggior parte dei casi comuni.

Errori di decodifica

Se viene scelto un cammino non corretto, solitamente si torna su quello corretto dopo qualche passo. Ciò porta ad errori di decodifica a *burst*.