Bahl, Cocke, Jelinek and Raviv (BCJR) Algorithm

Reference: "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", IEEE Trans. Info. Theory, Vol IT-20, pp 284-287, March 1974.

The Viterbi Algorithm – MLSE – minimizes the probability of sequence (word) error

- not necessarily minimize the probability of bit(symbol) error.

Objective: To derive an optimal decoding method for linear codes which minimizes the symbol error probability.

General Problem:

Estimating the *a posteriori* probabilities (APP) of

1. The states and
2. The transitions of a

Markov source observed through a noisy discrete memoryless channel (DMC).

The source above is assumed to be a "discrete-time finite-state" Markov process (e.g. Convolutional encoder, or a finite state machine – FSM).

M states, m=0,1,…..M-1.

The state of the source at time t is denoted by S_t and its output by X_t.

A state sequence from t to t′

$S_t^{t'} = S_t, S_{t+1},\ldots,S_{t'}$.

The output sequence $X_t^{t'} = X_t, X_{t+1},\ldots,X_{t'}$.

The state transitions of the Markov source are governed by the transition probabilities

$p_t(m \mid m') = \Pr\{S_t = m \mid S_{t-1} = m')$

And the output by the probabilities

$q_t(X \mid m',m) = \Pr\{X_t = X \mid S_{t-1} = m' ; S_t = m\}$

where X belongs to some finite discrete alphabet.

The Markov source starts in the initial state $S_0 = 0$, and produces an output sequence X_1^τ ending in the terminal state $S_\tau = 0$.

X_1^τ is the input to a noisy DMC whose output is the sequence

$Y_1^\tau = Y_1, Y_2, \ldots, Y_\tau$.

The transition probabilities of the DMC are defined by R(.|.) so that for all

$1 \le t \le \tau \qquad \Pr\{Y_1^t \mid X_1^t\} = \prod_{j=1}^{t} R(Y_j \mid X_j)$

The objective of the decoder is to examine Y_1^τ and estimate the APP of the states and the transitions of the Markov source, i.e., the conditional probabilities

$$\Pr\{S_t = m \mid Y_1^\tau\} = \Pr\{S_t = m; Y_1^\tau\} / \Pr\{Y_1^\tau\} \qquad (1)$$

$$\Pr\{S_{t-1} = m'; S_t = m \mid Y_1^\tau\} = \Pr\{S_{t-1} = m'; S_t = m; Y_1^\tau\} / \Pr\{Y_1^\tau\} \qquad (2)$$

Graphical interpretation:

State diagram: Nodes are states, branches – the transitions having nonzero probabilities.

Trellis diagram: Index the states with both the time index t and state index m $\Rightarrow$ "trellis"
Shows the time progression of the state sequences.

For every state sequence S_1^τ there is a unique path through the trellis and vice versa.

If the Markov source is time variant, then we can no longer represent it by a state-transition diagram; however we can construct a trellis for its state sequences.

Associated with each node in the trellis is the corresponding APP

$\Pr\{ S_t = m \mid Y_1^\tau \}$ and

Associated with each branch in the trellis is the corresponding APP

$\Pr\{ S_{t-1} = m'; S_t = m \mid Y_1^\tau \}$.

The objective of the decoder is to examine Y_1^τ and compute these APP.

It is easier to derive the joint probabilities

$\lambda_t(m) = \Pr\{ S_t = m ; Y_1^\tau \}$ and

$\sigma_t(m',m) = \Pr\{ S_{t-1} = m'; S_t = m ; Y_1^\tau \}$.

For a given Y_1^τ , $\Pr\{ Y_1^\tau \}$ is a constant, we can divide $\lambda_t(m)$ and $\sigma_t(m',m)$ by $\Pr\{ Y_1^\tau \}$ which is equivalent to $\lambda_\tau(0)$, which is available from the decoder.

Alternatively we can normalize $\lambda_t(m)$ and $\sigma_t(m',m)$ to add up to 1 to obtain the same result.

$\Rightarrow$ Derive a method for obtaining the probabilities $\lambda_t(m)$ and $\sigma_t(m',m)$.

Define the probability functions:

$$\alpha_t(m) = \Pr\{ S_t = m ; Y_1^t \}$$
$$\beta_t(m) = \Pr\{ Y_{t+1}^\tau \mid S_t = m \}$$

$$\gamma_t(m',m) = \Pr\{ S_t = m ; Y_t \mid S_{t-1} = m'\}$$

Now $\lambda_t(m) = \Pr\{ S_t = m ; Y_1^{\tau} \}$

$$= \Pr\{ S_t = m ; Y_1^{t} \} . \Pr\{ Y_{t+1}^{\tau} \mid S_t = m ; Y_1^{t} \}$$

where

$$\Pr\{ Y_{t+1}^{\tau} \mid S_t = m ; Y_1^{t} \} = \Pr\{ S_t = m ; Y_1^{t} ; Y_{t+1}^{\tau} \} / \Pr\{ S_t = m ; Y_1^{t} \}$$

$$= \Pr\{ S_t = m ; Y_1^{\tau} \} / \Pr\{ S_t = m ; Y_1^{t} \}$$

and $\alpha_t(m) = \Pr\{ S_t = m ; Y_1^{t} \}$

therefore,

$$\lambda_t(m) = \alpha_t(m) . \Pr\{ Y_{t+1}^{\tau} \mid S_t = m ; Y_1^{t} \}$$
$$= \alpha_t(m) . \Pr\{ Y_{t+1}^{\tau} \mid S_t = m\}$$

Markov property: if S_t is known, events after time t do not depend on Y_1^{t}.

Thus $$\lambda_t(m) = \alpha_t(m) . \beta_t(m) \qquad (3)$$

since $\beta_t(m) = \Pr\{ Y_{t+1}^{\tau} \mid S_t = m \}$.

In a similar manner,

$$\sigma_t(m',m) = \Pr\{ S_{t-1} = m'; S_t = m ; Y_1^{\tau} \}$$
$$= \Pr\{ S_{t-1} = m' ; Y_1^{t-1} \} . \Pr\{ S_t = m ; Y_t \mid S_{t-1} = m'\} . \Pr\{ Y_{t+1}^{\tau} \mid S_t = m\}$$

where

$$\Pr\{ S_t = m ; Y_t \mid S_{t-1} = m'\} = \Pr\{ S_t = m ; Y_t \mid S_{t-1} = m' ; Y_1^{t-1} \}$$
$$= \Pr\{ S_t = m ; Y_t ; S_{t-1} = m' ; Y_1^{t-1} \} / \Pr\{ S_{t-1} = m' ; Y_1^{t-1} \}$$

$$\Pr\{ Y_{t+1}^{\tau} \mid S_t = m\} = \Pr\{ Y_{t+1}^{\tau} \mid S_t = m ; Y_t ; S_{t-1} = m' ; Y_1^{t-1} \}$$
$$= \Pr\{ S_t = m ; Y_t ; S_{t-1} = m' ; Y_1^{t-1} ; Y_{t+1}^{\tau} \} / \Pr\{ S_t = m ; Y_t ; S_{t-1} = m' ; Y_1^{t-1} \}$$

the equations follow from the basic probability theory and Markov property.

Thus

$\sigma_t(m',m) = \Pr\{ S_{t-1} = m' ; Y_1^{t-1} \} . \Pr\{ S_t = m ; Y_t \mid S_{t-1} = m'\} . \Pr\{ Y_{t+1}^{\tau} \mid S_t = m\}$

$$= \alpha_{t-1}(m') . \gamma_t(m',m) . \beta_t(m) \qquad (4)$$

Now for t = 1,2,….., τ

$$\alpha_t(m) = \sum_{m'=0}^{M-1} \Pr\{S_{t-1} = m'; S_t = m; Y_1^t\}$$

$$= \sum_{m'=0}^{M-1} \Pr\{S_{t-1} = m'; Y_1^{t-1}\}.\Pr\{S_t = m; Y_t \mid S_{t-1=m'}; Y_1^{t-1}\}$$

$$= \sum_{m'=0}^{M-1} \Pr\{S_{t-1} = m'; Y_1^{t-1}\}.\Pr\{S_t = m; Y_t \mid S_{t-1=m'}\}$$

$$= \sum_{m'=0}^{M-1} \alpha_{t-1}(m'). \gamma_t(m',m) \qquad (5)$$

For t=0 we have the boundary conditions

$$\alpha_0(0) = 1, \qquad \text{and } \alpha_0(m) = 0, \text{ for } m \neq 0 \qquad (6).$$

Similarly, for t=1,2,3,….,τ-1

$$\beta_t(m) = \sum_{m'=0}^{M-1} \Pr\{S_{t+1} = m'; Y_{t+1}^t \mid S_t = m\}$$

$$= \sum_{m'=0}^{M-1} \Pr\{S_{t+1} = m'; Y_{t+1} \mid S_t = m\}.\Pr\{Y_{t+2}^t \mid S_{t+1} = m'\}$$

$$= \sum_{m'=0}^{M-1} \beta_{t+1}(m'). \gamma_{t+1}(m,m') \qquad (7)$$

The appropriate boundary conditions are

$$\beta_\tau(0) = 1, \text{ and } \beta_\tau(m) = 0, \text{ for } m \neq 0. \tag{8}$$

(5) and (7) show that $\alpha_t(m)$ and $\beta_\tau(m)$ are recursively obtainable.

Now,

$$\gamma_t(m',m) = \sum_X \Pr\{S_t = m \mid S_{t-1} = m'\}.\Pr(X_t = X \mid S_{t-1} = m', S_t = m\}.\Pr\{Y_t \mid X\}$$

$$= \sum_X p_t(m \mid m').q_t(X \mid m', m).R(Y_t \mid X) \tag{9}$$

where the summation in (9) is over all possible output symbols X.

Now the operation of the decoder for computing $\lambda_t(m)$ and $\sigma_t(m',m)$.

1. $\alpha_0(m)$ and $\beta_\tau(m)$, m=0,1,….,M-1 are initialized according to (6) and (8).

2. As soon as Y_t is received, the decoder computes $\gamma_t(m',m)$ using (9) and $\alpha_t(m)$ using (5). The obtained values for $\alpha_t(m)$ are stored for all t and m.

3. After the complete sequence Y_1^τ is received the decoder recursively computes $\beta_t(m)$ using (7). When the $\beta_t(m)$ have been computed, they can be multiplied by the appropriate $\alpha_t(m)$ and $\gamma_t(m',m)$ to obtain $\lambda_t(m)$ and $\sigma_t(m',m)$ using (3) and (4).

We can obtain the probability of any event that is a function of the states by summing the appropriate $\lambda_t(m)$; likewise , the $\sigma_t(m',m)$ can be used to obtain the probability of any event which is a function of transitions.

Summary:

- Calculation of α - "forward recursion"
- Calculation of β - "backward recursion"
- Probability of input information – by summing $\lambda_t(m)$
- Probability of coded information – by summing $\sigma_t(m',m)$

Application to Convolutional Codes

A binary k_0/n_0 Convolutional encoder. The input to the encoder at time t is the block $I_t = (i_{(t)}^{(1)}, i_{(t)}^{(2)}, \ldots, i_{(t)}^{(k_0)})$ and the corresponding output is $X_t = (x_{(t)}^{(1)}, x_{(t)}^{(2)}, \ldots, x_{(t)}^{(n_0)})$.

The state of the encoder (contents of the register, ν most recent input blocks)

$S_t = (s_{(t)}^{(1)}, s_{(t)}^{(2)}, \ldots, s_{(t)}^{(k_0\nu)}) = (I_t, I_{t-1}, \ldots, I_{t-v+1})$.

The encoder starts in state $S_0 = \mathbf{0}$. Information sequence I_1^T is the input to the encoder, followed by v blocks of all-zero inputs ,i.e., $I_{T+1}^{\tau} = \mathbf{0,0,\ldots,0}$ where $\tau = T+v$, so that $S_\tau = \mathbf{0}$.

The transition probabilities $p_t(m|m')$ of the trellis are governed by the input statistics. Generally we assume all input sequences equally likely for $t \leq T$, and since there are 2^{k0} possible transitions out of each state , $p_t(m|m') = 2^{-k0}$ for each of these transitions.

For $t > T$, only one transition is possible out of each state, and this has probability 1.

The output X_t is a deterministic function of the transition so that, there is a 0-1 probability distribution $q_t(X|m',m)$ over the alphabet of binary n-tuples. For the time invariant codes $q_t(.|.)$ is independent of t.

If the output sequence is sent over a **memoryless** channel (DMC, AWGN) or where the output is conditionally independent of other outputs given the current input (ISI in AWGN) with transition probabilities r(.|.), the derived block transition probabilities are

$$R(Y_t|X_t) = \prod_{j=1}^{n_0} r(y^{(j)} \mid x_t^{(j)})$$

Where $Y_t = = (y_{(t)}^{(1)}, y_{(t)}^{(2)}, \ldots, y_{(t)}^{(n_0)})$ is the block received by the receiver at time t.

To minimize the symbol probability of error we must determine the most likely input digits $i_t^{(j)}$ from the received sequence Y_1^τ.

Let $A_t^{(j)}$ be the set of states S_t such that $s_t^{(j)} = 0$. Note that $A_t^{(j)}$ is not dependent on t. Then we have

$$s_t^{(j)} = i_t^{(j)}, \quad j=1,2,\ldots, k_0$$

which implies $\quad \Pr\{ i_t^{(j)}=0 \,;\, Y_1^\tau \} = \sum_{s_t \in A_t^{(j)}} l_t(m)$

Normalizing by $\Pr\{Y_1^\tau\} = \lambda_\tau(0)$

$$\Pr\{ i_t^{(j)}=0 \mid Y_1^\tau \} = \frac{1}{l_t(0)} \sum_{s_t \in A_t^{(j)}} l_t(m)$$

We decode $i_t^{(j)}=0$ if $\Pr\{ i_t^{(j)}=0 \mid Y_1^\tau \} \geq 0.5$, otherwise $i_t^{(j)}=1$.

Another equivalent way would be to compute the log likelihood ratio (LLR)

$$\text{LLR} = \text{Log}\, \frac{\Pr\{i_t^{(j)}=0 \mid Y_1^t\}}{\Pr\{i_t^{(j)}=1 \mid Y_1^t\}} = Log \frac{\sum_{s_t \in A_t^{(j)}, m} l_t(m)}{\sum_{s_t \in A_t^{(j)c}, m'} l_t(m')} \quad \geq \quad 0 \text{ then } i_t^{(j)}=0.$$

Sometimes, it is of interest to determine the APP of the encoder output digits, i.e., Pr{ $x_t^{(j)}$ =0 |Y_1^{τ}}.

Let $B_t^{(j)}$ be the set of transitions S_{t-1}=m′ to S_t = msuch that the jth output digit $x_t^{(j)}$ on that transition is 0. $B_t^{(j)}$ is independent of t for time invariant codes. Then,

$$\Pr\{ x_t^{(j)} = 0 ; Y_1^{\tau} \} = \sum_{(m',m)\in B_t^{(j)}} s_t(m',m)$$

Which can be normalized to give Pr{ $x_t^{(j)}$ =0 |Y_1^{τ}}.

Therefore we can state the following.

"We can obtain the probability of any event that is a function of the states, e.g., probability of input symbols, by summing the appropriate l_t*(m); likewise , the* s_t*(m ¢m) can be used to obtain the probability of any event which is a function of transitions, e.g. probability of output symbols".*

Elements of Turbo (Iterative) Decoding

A. Analysis of Log Likelihood Ratio (LLR) of Systematic Convolutional Codes (in AWGN, Rate ($1/n_0$) type)

The transition probability is given by

$$p(y_{(t)}^{(j)} \mid x_{(t)}^{(j)}) = \frac{1}{\sqrt{2ps^2}} \exp\{-\frac{1}{2s^2}(y_t^{(j)} - x_t^{(j)})^2\}$$

Now the $\quad \text{LLR} = Log\dfrac{\Pr\{i_t = 0 \mid Y_1^t\}}{\Pr\{i_t = 1 \mid Y_1^t\}} = Log\dfrac{\sum\limits_{s_t \in A_t, m} l_t(m)}{\sum\limits_{s_t \in A_t^C, m'} l_t(m')}$

$$= \text{Log}\ \frac{\sum\limits_{s_t \in A_t, m, i_t = 0} l_t(m)}{\sum\limits_{s_t \in A_t^C, m', i_t = 1} l_t(m')}$$

Using $\lambda_t(m) = \alpha_t(m)\beta_t(m)$ and $\alpha_t(m) = \sum\limits_{m'=0}^{M-1} a_{t-1}(m').\, g_t(m', m)$

$$\text{LLR} = \text{Log}\ \frac{\sum\limits_{s_t \in A_t, m, i_t = 0} \sum\limits_{m''=0}^{M-1} a_{t-1}(m'').\, g_t(m'', m).b_t(m)}{\sum\limits_{s_t \in A_t^C, m', i_t = 1} \sum\limits_{\hat{m}=0}^{M-1} a_{t-1}(\hat{m}).\, g_t(\hat{m}, m').b_t(m')}$$

Since the encoder is systematic, i.e., $i_t = 0 \Rightarrow x_t^{(1)} = -1$ for those transitions

$R(Y_t \mid X_t) = p(y_t^{(1)} \mid x_t^{(1)} = -1).\ \prod\limits_{j=2}^{n_0} p(y_t^{(j)} \mid x_t^{(j)})$, $p_t(m|m') = 1/2$

Similarly for $i_t = 1 \Rightarrow x_t^{(1)} = 1$ and for them

$R(Y_t \mid X_t) = p(y_t^{(1)} \mid x_t^{(1)} = 1).\ \prod\limits_{j=2}^{n_0} p(y_t^{(j)} \mid x_t^{(j)})$, $p_t(m|m') = 1/2$

Therefore,

$$\text{LLR} = \text{Log}\,\frac{p(y_t^{(1)} \mid x_t^{(1)} = -1)}{p(y_t^{(1)} \mid x_t^{(1)} = 1)} \quad +$$

$$\text{Log}\;\frac{\displaystyle\sum_{s_t \in A_t\,, m, i_t = 0} \sum_{m''=0}^{M-1} a_{t-1}(m'').\prod_{j=2}^{n_0} p(y_t^{(j)} \mid x_t^{(j)}).b_t(m)}{\displaystyle\sum_{s_t \in A_t^c, m', i_t = 1} \sum_{\hat{m}=0}^{M-1} a_{t-1}(\hat{m}).\prod_{k=2}^{n_0} p(y_t^{(k)} \mid x_t^{(k)}).b_t(m')}$$

The second term is a function of the redundant information introduced by the encoder. In general it has the same sign as x_t. This quantity represents the "**extrinsic**" information supplied by the decoder and does not depend on the decoder input $x_t^{(1)}$.

This property will be used for decoding "parallel concatenated codes – **TURBO** codes".

In the first equation since we consider states with $i_t = 0$ in the numerator and the states with $i_t = 1$ respectively, we can rewrite it in the following was as well.

$$\text{LLR} = \text{Log}\;\frac{\displaystyle\sum_{(m'',m)\; i_t = 0} a_{t-1}(m'').\,g_t(m'',m).b_t(m)}{\displaystyle\sum_{(\hat{m},m')\; i_t = 1} a_{t-1}(\hat{m}).\,g_t(\hat{m},m').b_t(m')}$$

$$= \text{Log}\,\frac{p(y_t^{(1)} \mid x_t^{(1)} = -1)}{p(y_t^{(1)} \mid x_t^{(1)} = 1)} \;\; +$$

$$\text{Log}\;\frac{\displaystyle\sum_{(m'',m)\; i_t = 0} a_{t-1}(m'').\prod_{j=2}^{n_0} p(y_t^{(j)} \mid x_t^{(j)}).b_t(m)}{\displaystyle\sum_{(\hat{m},m')\; i_t = 1} a_{t-1}(\hat{m}).\prod_{k=2}^{n_0} p(y_t^{(k)} \mid x_t^{(k)}).b_t(m')}$$

Thus the extrinsic part can be represented by

$$\text{Log}\;\frac{\displaystyle\sum_{(m'',m)\; i_t = 0} a_{t-1}(m'').\, g_e(m'',m).b_t(m)}{\displaystyle\sum_{(\hat{m},m')\; i_t = 1} a_{t-1}(\hat{m}).\, g_e(\hat{m},m').b_t(m')}$$