Chapter 2

BCJR Algorithm

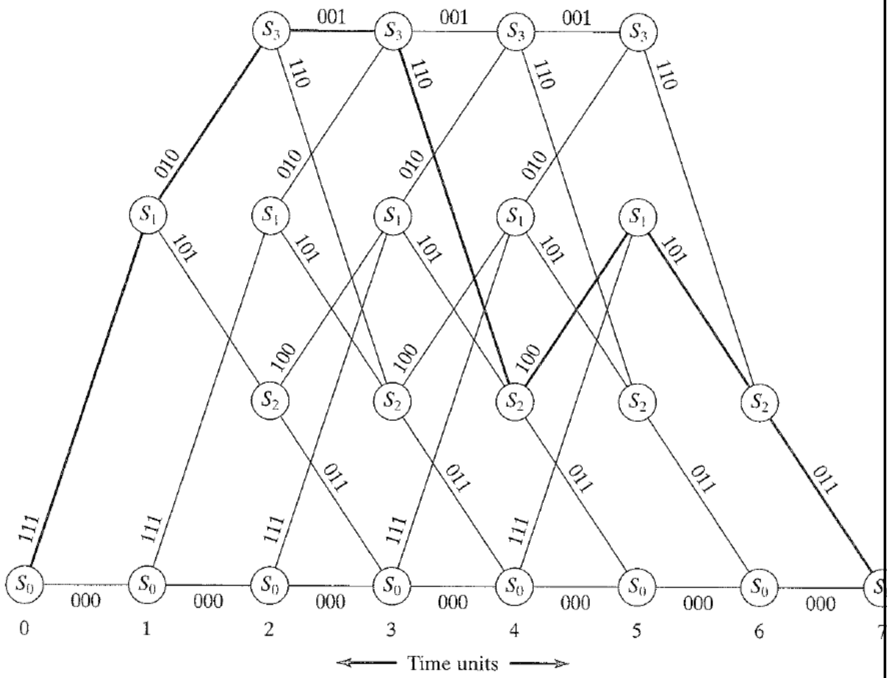
Veterbi Algorithm (revisted)

- Consider covolutional encoder with

$$\mathbb{G}(D) = [1 + D \quad 1 + D^2 \quad 1 + D + D^2]$$

- And information sequences of length h = 5
- The trellis diagram has h + m + 1 timeslots which equals 8 in our case
- Consider received sequence as

$$\mathbb{r} = (110, 110, 110, 111, 010, 101, 101).$$



- From Fig. 12.6 on the text book, we can see that

$$\hat{\mathbf{v}} = (111, 010, 110, 011, 111, 101, 011),$$

SOVA

The Soft-Output Viterbi Algorithm (SOVA) was first introduced in 1989.

We describe SOVA for convolutional code with R = 1/n on binary input, AWGN channel.

We assume that priori probabilities are not equally likely $p(u_L)$ *and* $L = 0,\ldots, h-1$.

Log-likelihood metric

- Let us define the log-likelihood ratio or the L-value of a received symbol *r* at the output of channel with binary inputs $v = \pm 1$

$$L(r) = \ln\left[\frac{p(r|v=+1)}{p(r|v=-1)}\right].$$

- Similarly the *L*-value of an information bit *u* is defined as

$$L(u) = \ln\left[\frac{p(u=+1)}{p(u=-1)}\right].$$

- Using Bay's rule if *v* is equally likely

$$L(r) = \ln\left[\frac{p(r|v=+1)}{p(r|v=-1)}\right] = \ln\left[\frac{p(v=+1|r)}{p(v=-1|r)}\right].$$

Log-likelihood metric

- A large positive value of L(r) indicates a high reliability that v = +1.
- A large negative value of L(r) indicates a high reliability that v = -1.
- A close to zero value of L(r) indicates a decision a bout the value of v based only on r is unreliable.
- The same a large positive value of L(u) indicates a high reliability that u = +1

Log-likelihood metric

- It can be shown that the L value is equal to (left as exercise for the students)

$$L(r) = (4E_s/N_0)r,$$

- Where $L_c = 4E_s/N_0$ is defined as channel reliability factor

BCJR algorithm

- Veterbi Algorithm minimizes the WER $P_w(E)$ that is $P(\hat{v} \neq v | r)$
- So it is minimizes the error probability between the transmitted and received codeword.
- In BCJR algorithm, we are interested in minimizing the bit error probability.
- This is done by maximizing the posteriori probability $P(\hat{u}_l = u_l | \mathbf{r})$
- That is why BCJR decoder is also called Maximum Posteriori Probability decoder (MAP)

BCJR algorithm

- We don't assume that the information bits are equally likely.
- The algorithms calculates the a posteriori L-values

$$L(u_l) \equiv \ln \left[\frac{P(u_l = +1 | \mathbf{r})}{P(u_l = -1 | \mathbf{r})} \right],$$

- Called APP L-values of each information bit, the decoder output is given by

$$\hat{u}_l = \begin{cases} +1 \text{ if } L(u_l) > 0 \\ -1 \text{ if } L(u_l) < 0 \end{cases}, \quad l = 0, 1, \cdots, h-1.$$

- We start our development of the BCJR algorithm by rewriting the APP value as

$$P(u_l = +1|\mathbf{r}) = \frac{p(u_l = +1, \mathbf{r})}{P(\mathbf{r})} = \frac{\sum_{\mathbf{u}\in\mathbf{U}_l^+} p(\mathbf{r}|\mathbf{v})P(\mathbf{u})}{\sum_{\mathbf{u}} p(\mathbf{r}|\mathbf{v})P(\mathbf{u})},$$

- Where U_L^+ is the set of all information sequences **u** such as u_l = 1, *v* is the transmitted codeword corresponding to the information sequence *u.*

- So we can rewrite the expression o f the APP L values as

$$L(u_l) = \ln\left[\frac{\sum_{\mathbf{u}\in\mathbf{U}_l^+} p(\mathbf{r}|\mathbf{v})P(\mathbf{u})}{\sum_{\mathbf{u}\in\mathbf{U}_l^-} p(\mathbf{r}|\mathbf{v})P(\mathbf{u})}\right],$$

- Where U_L^- is the set of all information sequences **u** such as u_l = -1

- The L values can be calculated using the previous formula but still it suffers from high degree of complexity.

- We can rewrite the a posteriori probability as

$$P(u_l = +1|\mathbf{r}) = \frac{p(u_l = +1, \mathbf{r})}{P(\mathbf{r})} = \frac{\sum_{(s',s)\in\Sigma_l^+} p(s_l = s', s_{l+1} = s, \mathbf{r})}{P(\mathbf{r})},$$

- Where Σ_l^+ is the set of all state pairs s_l = s' and s_{l+1} = s that corresponds to the input bit u_l = + 1.

- Reforming P(u_l= -1/r) in the same way and sub in the L value

$$L(u_l) = \ln\left\{\frac{\sum_{(s',s)\in\Sigma_l^+} p(s_l = s', s_{l+1} = s, \mathbf{r})}{\sum_{(s',s)\in\Sigma_l^-} p(s_l = s', s_{l+1} = s, \mathbf{r})}\right\},$$

- Where $\sum_l^-$ is the set of all state pairs $s_l = s'$ and $s_{l+1} = s$ that corresponds to the input bit $u_l = -1$.

- The joint pdf p(s',s,r) can be found recursively, starting from

$$p(s', s, \mathbf{r}) = p(s', s, \mathbf{r}_{t<l}, \mathbf{r}_l, \mathbf{r}_{t>l}),$$

- Where $r_{t<l}$ represents the portion of the received $\boldsymbol{r}$ before the time l and Where $\mathbf{r}_{t>l}$ represents the portion of the received r after the time l.

- Now application of Bay's rule

$$\begin{aligned} p(s', s, \mathbf{r}) &= p(\mathbf{r}_{t>l} | s', s, \mathbf{r}_{t<l}, \mathbf{r}_l) p(s', s, \mathbf{r}_{t<l}, \mathbf{r}_l) \\ &= p(\mathbf{r}_{t>l} | s', s, \mathbf{r}_{t<l}, \mathbf{r}_l) p(s, \mathbf{r}_l | s', \mathbf{r}_{t<l}) p(s', \mathbf{r}_{t<l}) \\ &= p(\mathbf{r}_{t>l} | s) p(s, \mathbf{r}_l | s') p(s', \mathbf{r}_{t<l}), \end{aligned}$$

- Defining

$$\begin{aligned} \alpha_l(s') &\equiv p(s', \mathbf{r}_{t<l}) \\ \gamma_l(s', s) &\equiv p(s, \mathbf{r}_l | s') \\ \beta_{l+1}(s) &\equiv p(\mathbf{r}_{t>l} | s), \end{aligned}$$

- So the joint pdf can be rewritten as

$$p(s', s, \mathbf{r}) = \beta_{l+1}(s) \gamma_l(s', s) \alpha_l(s').$$

- We can write expression for $\alpha_{l+1}(s)$ as

$$\alpha_{l+1}(s) = \sum_{s' \in \sigma_l} \gamma_l(s', s) \alpha_l(s'),$$

- Where σ_l is the set of all states at time l.

- $\beta_l(s')$ can be written as

$$\beta_l(s') = \sum_{s \in \sigma_{l+1}} \gamma_l(s', s)\beta_{l+1}(s),$$

- Where σ_{l+1} is the set of all states at time l+1.
- The forward recursion starts from

$$\alpha_0(s) = \begin{cases} 1, & s = \mathbb{0} \\ 0, & s \neq \mathbb{0} \end{cases},$$

- And the backward recursion starts from
-

$$\beta_K(s) = \begin{cases} 1, & s = \mathbb{0} \\ 0, & s \neq \mathbb{0} \end{cases},$$

- We can write the branch metric as

$$\begin{aligned} \gamma_l(s', s) &= p(s, \mathbb{r}_l | s') = \frac{p(s', s, \mathbb{r}_l)}{P(s')} \\ &= \left[\frac{P(s', s)}{P(s')}\right]\left[\frac{p(s', s, \mathbb{r}_l)}{P(s', s)}\right] \\ &= P(s|s')p(\mathbb{r}_l|s', s) = P(u_l)p(\mathbb{r}_l|\mathbb{v}_l), \end{aligned}$$

- Which yields

$$\gamma_l(s', s) = P(u_l)p(\mathbb{r}_l|\mathbb{v}_l) = P(u_l)\left(\sqrt{\frac{E_s}{\pi N_0}}\right)^n e^{-\frac{E_s}{N_0}||\mathbb{r}_l - \mathbb{v}_l||^2},$$

- We can drop the constant to achieve

$$\gamma_l(s', s) = P(u_l)e^{-E_s/N_0||\mathbb{r}_l - \mathbb{v}_l||^2}.$$

- The priori probability can be written as

$$
\begin{aligned}
P(u_l = \pm 1) &= \frac{[P(u_l = +1)/P(u_l = -1)]^{\pm 1}}{\{1 + [P(u_l = +1)/P(u_l = -1)]^{\pm 1}\}} \\
&= \frac{e^{\pm L_a(u_l)}}{\{1 + e^{\pm L_a(u_l)}\}} \\
&= \frac{e^{-L_a(u_l)/2}}{\{1 + e^{-L_a(u_l)}\}} e^{u_l L_a(u_l)/2} \\
&= A_l e^{u_l L_a(u_l)/2},
\end{aligned}
$$

- The L value depend of the value of u, thus

$$
\begin{aligned}
\gamma_l(s', s) &= A_l e^{u_l L_a(u_l)/2} e^{-(E_s/N_0)||\mathbf{r}_l - \mathbf{v}_l||^2}, \\
&= A_l e^{u_l L_a(u_l)/2} e^{(2E_s/N_0)(\mathbf{r}_l \cdot \mathbf{v}_l) - ||\mathbf{r}_l||^2 - ||\mathbf{v}_l||^2} \\
&= A_l e^{-(||\mathbf{r}_l||^2 + n)} e^{u_l L_a(u_l)/2} e^{(L_c/2)(\mathbf{r}_l \cdot \mathbf{v}_l)} \\
&= A_l B_l e^{u_l L_a(u_l)/2} e^{(L_c/2)(\mathbf{r}_l \cdot \mathbf{v}_l)}, \quad l = 0, 1, \cdots, h-1, \\
\gamma_l(s', s) &= P(u_l) e^{-(E_s/N_0)||\mathbf{r}_l - \mathbf{v}_l||^2} \\
&= e^{-(E_s/N_0)||\mathbf{r}_l - \mathbf{v}_l||^2}, \\
&= B_l e^{(L_c/2)(\mathbf{r}_l \cdot \mathbf{v}_l)}, \quad l = h, h+1, \cdots, K-1,
\end{aligned}
$$

- Again if we drop the constants

$$
\begin{aligned}
\gamma_l(s', s) &= e^{u_l L_a(u_l)/2} e^{(L_c/2)(\mathbf{r}_l \cdot \mathbf{v}_l)}, \quad l = 0, 1, \cdots, h-1, \\
\gamma_l(s', s) &= e^{(L_c/2)(\mathbf{r}_l \cdot \mathbf{v}_l)}, \quad l = h, h+1, \cdots, K-1,
\end{aligned}
$$

- Using the log-domain enable using

$$\max^*(x, y) \equiv \ln(e^x + e^y) = \max(x, y) + \ln(1 + e^{-|x-y|})$$

- And the log-domain metrics are

$$\gamma_l^*(s', s) \equiv \ln\ \gamma_l(s', s) = \begin{cases} \frac{u_l L_a(u_l)}{2} + \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l, & l = 0, 1, \cdots, h-1, \\ \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l, & l = h, h+1, \cdots, K-1, \end{cases}$$

$$\alpha_{l+1}^*(s) = \max{}^*_{s' \in \sigma_l}[\gamma_l^*(s', s) + \alpha_l^*(s')], \quad l = 0, 1, \cdots, K-1,$$

$$\alpha_0^*(s) \equiv \ln\ \alpha_0(s) = \begin{cases} 0, & s = \mathbb{0} \\ -\infty, & s \neq \mathbb{0}, \end{cases}$$

$$\beta_l^*(s') = \max{}^*_{s \in \sigma_{l+1}}[\gamma_l^*(s', s) + \beta_{l+1}^*(s)], \quad l = K-1, K-2, \cdots, 0,$$

$$\beta_K^*(s) \equiv \ln\ \beta_K(s) = \begin{cases} 0, & s = \mathbb{0} \\ -\infty, & s \neq \mathbb{0}. \end{cases}$$

- Writing the expression for the pdf p(*s'*, *s*, *r*) and the APP L-value *L*(u_l) as:

- $$p(s', s, \mathbf{r}) = e^{\beta_{l+1}^*(s) + \gamma_l^*(s', s) + \alpha_l^*(s')}$$

$$L(u_l) = \ln \left\{ \sum_{(s', s) \in \Sigma_l^+} e^{\beta_{l+1}^*(s) + \gamma_l^*(s', s) + \alpha_l^*(s')} \right\} - \ln \left\{ \sum_{(s', s) \in \Sigma_l^-} e^{\beta_{l+1}^*(s) + \gamma_l^*(s', s) + \alpha_l^*(s')} \right\}$$

- Using the following math expression

$$\max{}^*(x, y, z) \equiv \ln(e^x + e^y + e^z) = \max{}^*[\max{}^*(x, y), z],$$

- We can formula the L value as

$$L(u_l) = \max{}^*_{(s',s)\in\Sigma_l^+}[\beta_{l+1}^*(s) + \gamma_l^*(s', s) + \alpha_l^*(s')] \\ - \max{}^*_{(s',s)\in\Sigma_l^-}[\beta_{l+1}^*(s) + \gamma_l^*(s', s) + \alpha_l^*(s')].$$

Steps of Log-Domain BCJR algorithm

- Step1: calculate the forward and backward metrics using

$$\alpha_0^*(s) \equiv \ln \alpha_0(s) = \begin{cases} 0, & s = \mathbb{0} \\ -\infty, & s \neq \mathbb{0}, \end{cases}$$

$$\beta_K^*(s) \equiv \ln \beta_K(s) = \begin{cases} 0, & s = \mathbb{0} \\ -\infty, & s \neq \mathbb{0}. \end{cases}$$

- Step 2 Compute the branch metric using

$$\gamma_l^*(s', s) \equiv \ln \gamma_l(s', s) = \begin{cases} \frac{u_l L_a(u_l)}{2} + \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l, & l = 0, 1, \cdots, h-1, \\ \frac{L_c}{2}\mathbf{r}_l \cdot \mathbf{v}_l, & l = h, h+1, \cdots, K-1, \end{cases}$$

Steps of Log-Domain BCJR algorithm

- Step3: calculate the forward metrics using

$$\alpha^*_{l+1}(s) = \max{}^*_{s' \in \sigma_l}[\gamma^*_l(s', s) + \alpha^*_l(s')], \quad l = 0, 1, \cdots, K-1,$$

- Step 4 Compute the backward metric using

$$\beta^*_l(s') = \ln \sum_{s \in \sigma_{l+1}} e^{[\gamma^*_l(s', s) + \beta^*_{l+1}(s)]}$$

- Step 5 compute the APP-L values using

$$L(u_l) = \max{}^*_{(s', s) \in \Sigma^+_l}[\beta^*_{l+1}(s) + \gamma^*_l(s', s) + \alpha^*_l(s')] - \max{}^*_{(s', s) \in \Sigma^-_l}[\beta^*_{l+1}(s) + \gamma^*_l(s', s) + \alpha^*_l(s')].$$

Steps of Log-Domain BCJR algorithm

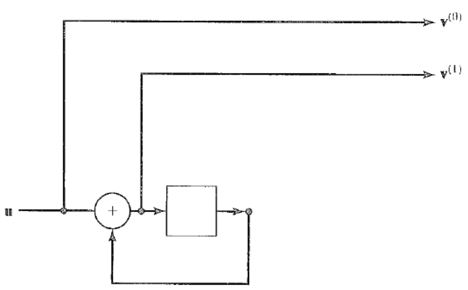
- Step6: (Optional) compute the hard decisions using

$$\hat{u}_l = \begin{cases} +1 \text{ if } L(u_l) > 0 \\ -1 \text{ if } L(u_l) < 0 \end{cases}, \quad l = 0, 1, \cdots, h-1.$$

Example

- We will consider the BCJR decoding of a (2, 1, 1) systematic Recursive Convolutional code on AWGN with generator matrix

$$\mathbb{G}(D) = [1 \;\; 1/(1+D)]$$

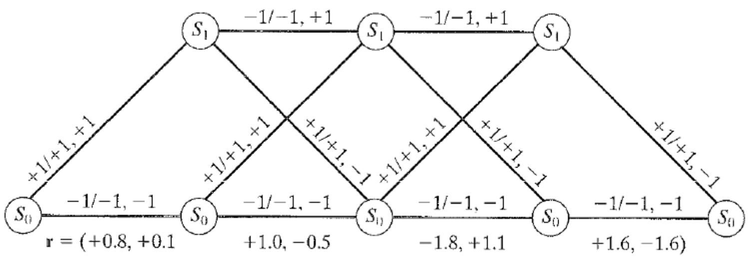


(a)

- Let u = (u0, u1, u2, u3) denote the input vector of length 4 and v = (v0, v1, v2, v3) denotes the codeword of length 8.

- We assume Es/ N0 = $\frac{1}{4}$(-6.02) dB

- The received vector r = (+0.8, +0.1; +1.0, -0.5; -1.8, +1.1; 1.6, -1.6)



(b)

- The rate of the terminated code is R = h/N =3/8
- .
- Eb/ N0 = Es/RN0 = 2/3

- Assuming that the information bits are equaly likely La(ul) = 0
- Lc = 4 Eb/N0 = 1

$$\gamma_0^*(S_0, S_0) = \frac{-1}{2}L_a(u_0) + \frac{1}{2}\mathbf{r}_0 \cdot \mathbf{v}_0$$
$$= \frac{1}{2}(-0.8 - 0.1) = -0.45$$
$$\gamma_0^*(S_0, S_1) = \frac{+1}{2}L_a(u_0) + \frac{1}{2}\mathbf{r}_0 \cdot \mathbf{v}_0$$
$$= \frac{1}{2}(0.8 + 0.1) = 0.45$$

$$\gamma_1^*(S_0, S_0) = \frac{-1}{2}L_a(u_1) + \frac{1}{2}\mathbf{r}_1 \cdot \mathbf{v}_1$$
$$= \frac{1}{2}(-1.0 + 0.5) = -0.25$$
$$\gamma_1^*(S_0, S_1) = \frac{+1}{2}L_a(u_1) + \frac{1}{2}\mathbf{r}_1 \cdot \mathbf{v}_1$$
$$= \frac{1}{2}(1.0 - 0.5) = 0.25$$
$$\gamma_1^*(S_1, S_1) = \frac{-1}{2}L_a(u_1) + \frac{1}{2}\mathbf{r}_1 \cdot \mathbf{v}_1$$
$$= \frac{1}{2}(-1.0 - 0.5) = -0.75$$
$$\gamma_1^*(S_1, S_0) = \frac{+1}{2}L_a(u_1) + \frac{1}{2}\mathbf{r}_1 \cdot \mathbf{v}_1$$
$$= \frac{1}{2}(1.0 + 0.5) = 0.75$$

$$\gamma_2^*(S_0, S_0) = \frac{-1}{2}L_a(u_2) + \frac{1}{2}\mathbf{r}_2 \cdot \mathbf{v}_2$$
$$= \frac{1}{2}(1.8 - 1.1) = 0.35$$
$$\gamma_2^*(S_0, S_1) = \frac{+1}{2}L_a(u_2) + \frac{1}{2}\mathbf{r}_2 \cdot \mathbf{v}_2$$
$$= \frac{1}{2}(-1.8 + 1.1) = -0.35$$
$$\gamma_2^*(S_1, S_1) = \frac{-1}{2}L_a(u_2) + \frac{1}{2}\mathbf{r}_2 \cdot \mathbf{v}_2$$
$$= \frac{1}{2}(1.8 + 1.1) = 1.45$$
$$\gamma_2^*(S_1, S_0) = \frac{+1}{2}L_a(u_2) + \frac{1}{2}\mathbf{r}_2 \cdot \mathbf{v}_2$$
$$= \frac{1}{2}(-1.8 - 1.1) = -1.45$$

$$\gamma_3^*(S_0, S_0) = \frac{+1}{2}\mathbf{r}_3 \cdot \mathbf{v}_3$$
$$= \frac{1}{2}(-1.6 + 1.6) = 0$$
$$\gamma_3^*(S_1, S_0) = \frac{+1}{2}\mathbf{r}_3 \cdot \mathbf{v}_3$$
$$= \frac{1}{2}(1.6 + 1.6) = 1.60.$$

Now compute the log-domain forward metrics

$$\alpha_1^*(S_0) = [\gamma_0^*(S_0, S_0) + \alpha_0^*(S_0)] = -0.45 + 0 = -0.45$$
$$\alpha_1^*(S_1) = [\gamma_0^*(S_0, S_1) + \alpha_0^*(S_0)] = 0.45 + 0 = 0.45$$
$$\alpha_2^*(S_0) = \max{}^*\{[\gamma_1^*(S_0, S_0) + \alpha_1^*(S_0)], [\gamma_1^*(S_1, S_0) + \alpha_1^*(S_1)]\}$$
$$= \max{}^*\{[(-0.25) + (-0.45)], [(0.75) + (0.45)]\}$$
$$= \max{}^*(-0.70, +1.20) = 1.20 + \ln(1 + e^{-|-1.9|}) = 1.34$$

$$\alpha_2^*(S_1) = \max{}^*\{[\gamma_1^*(S_0, S_1) + \alpha_1^*(S_0)], [\gamma_1^*(S_1, S_1) + \alpha_1^*(S_1)]\}$$
$$= \max{}^*(-0.20, -0.30) = -0.20 + \ln(1 + e^{-|0.1|}) = 0.44.$$

- Similarly compute the log-domain backward metrics

$$\beta_3^*(S_0) = [\gamma_3^*(S_0, S_0) + \beta_4^*(S_0)] = 0 + 0 = 0$$
$$\beta_3^*(S_1) = [\gamma_3^*(S_1, S_0) + \beta_4^*(S_0)] = 1.60 + 0 = 1.60$$
$$\beta_2^*(S_0) = \max{}^*\{[\gamma_2^*(S_0, S_0) + \beta_3^*(S_0)], [\gamma_2^*(S_0, S_1) + \beta_3^*(S_1)]\}$$
$$= \max{}^*\{[(0.35) + (0)], [(-0.35) + (1.60)]\}$$
$$= \max{}^*(0.35, 1.25) = 1.25 + \ln(1 + e^{-|-0.90|}) = 1.59$$
$$\beta_2^*(S_1) = \max{}^*\{[\gamma_2^*(S_1, S_0) + \beta_3^*(S_0)], [\gamma_2^*(S_1, S_1) + \beta_3^*(S_1)]\}$$
$$= \max{}^*(-1.45, 3.05) = 3.05 + \ln(1 + e^{-|-4.5|}) = 3.06$$

$$\beta_1^*(S_0) = \max{}^*\{[\gamma_1^*(S_0, S_0) + \beta_2^*(S_0)], [\gamma_1^*(S_0, S_1) + \beta_2^*(S_1)]\}$$
$$= \max{}^*(1.34, 3.31) = 3.44$$
$$\beta_1^*(S_1) = \max{}^*\{[\gamma_1^*(S_1, S_0) + \beta_2^*(S_0)], [\gamma_1^*(S_1, S_1) + \beta_2^*(S_1)]\}$$
$$= \max{}^*(2.34, 2.31) = 3.02.$$

- Finally we compute the app L -values

$$L(u_0) = [\beta_1^*(S_1) + \gamma_0^*(S_0, S_1) + \alpha_0^*(S_0)] - [\beta_1^*(S_0) + \gamma_0^*(S_0, S_0) + \alpha_0^*(S_0)]$$
$$= (3.47) - (2.99) = +0.48$$
$$L(u_1) = \max{}^*\{[\beta_2^*(S_0) + \gamma_1^*(S_1, S_0) + \alpha_1^*(S_1)], [\beta_2^*(S_1) + \gamma_1^*(S_0, S_1) + \alpha_1^*(S_0)]\}$$
$$- \max{}^*\{[\beta_2^*(S_0) + \gamma_1^*(S_0, S_0) + \alpha_1^*(S_0)], [\beta_2^*(S_1) + \gamma_1^*(S_1, S_1) + \alpha_1^*(S_1)]\}$$
$$= \max{}^*[(2.79), (2.86)] - \max{}^*[(0.89), (2.76)]$$
$$= (3.52) - (2.90) = +0.62$$

$$L(u_2) = \max^*\{[\beta_3^*(S_0) + \gamma_2^*(S_1, S_0) + \alpha_2^*(S_1)], [\beta_3^*(S_1) + \gamma_2^*(S_0, S_1) + \alpha_2^*(S_0)]\}$$
$$- \max^*\{[\beta_3^*(S_0) + \gamma_2^*(S_0, S_0) + \alpha_2^*(S_0)], [\beta_3^*(S_1) + \gamma_2^*(S_1, S_1) + \alpha_2^*(S_1)]\}$$
$$= \max^*[(-1.01), (2.59)] - \max^*[(1.69), (3.49)]$$
$$= (2.62) - (3.64) = -1.02.$$

Example

In the Max-log-MAP algorithm, the branch metrics remain as computed in (12.137). The approximation $\max^*(x, y) \approx \max(x, y)$ of (12.133) affects the computation of the forward and backward metrics (see (12.138) and (12.139), respectively), as follows:

$$\alpha_2^*(S_0) = \max(-0.70, +1.20) = 1.20$$
$$\alpha_2^*(S_1) = \max(-0.20, -0.30) = -0.20$$
$$\beta_2^*(S_0) = \max(0.35, 1.25) = 1.25$$
$$\beta_2^*(S_1) = \max(-1.45, 3.05) = 3.05$$
$$\beta_1^*(S_0) = \max(1.34, 3.31) = 3.31$$
$$\beta_1^*(S_1) = \max(2.34, 2.31) = 2.34.$$

- Finding the L-values

$$L(u_0) = [\beta_1^*(S_1) + \gamma_0^*(S_0, S_1) + \alpha_0^*(S_0)] - [\beta_1^*(S_0) + \gamma_0^*(S_0, S_0) + \alpha_0^*(S_0)]$$
$$= (2.79) - (2.86) = -0.07$$
$$L(u_1) = \max\{[\beta_2^*(S_0) + \gamma_1^*(S_1, S_0) + \alpha_1^*(S_1)], [\beta_2^*(S_1) + \gamma_1^*(S_0, S_1) + \alpha_1^*(S_0)]\}$$
$$- \max\{[\beta_2^*(S_0) + \gamma_1^*(S_0, S_0) + \alpha_1^*(S_0)], [\beta_2^*(S_1) + \gamma_1^*(S_1, S_1) + \alpha_1^*(S_1)]\}$$
$$= \max[(2.79), (2.86)] - \max[(0.89), (2.76)]$$
$$= (2.86) - (2.76) = +0.1000$$
$$L(u_2) = \max\{[\beta_3^*(S_0) + \gamma_2^*(S_1, S_0) + \alpha_2^*(S_1)], [\beta_3^*(S_1) + \gamma_2^*(S_0, S_1) + \alpha_2^*(S_0)]\}$$
$$- \max\{[\beta_3^*(S_0) + \gamma_2^*(S_0, S_0) + \alpha_2^*(S_0)], [\beta_3^*(S_1) + \gamma_2^*(S_1, S_1) + \alpha_2^*(S_1)]\}$$
$$= \max[(-1.65), (2.45)] - \max[(1.55), (2.85)]$$
$$= (2.45) - (2.85) = -0.4000. \qquad (12.143)$$

- Then he hard decision outputs of the Max-log-MAP decoder is
- u = (-1, +1, -1)