Near Maximum Likelihood Sequential Search Decoding Algorithms for Binary Convolutional Codes

Shin-Lin Shieh

Directed by: Prof. Po-Ning Chen and Prof. Yunghsiang S. Han

Department of Communications Engineering,
National Chiao-Tung University

June 19, 2008
Outline

• Introduction

• Sequential Decoding Algorithms & Fano Metric

• MLSDA and the Proposed Early Elimination Scheme

• Performance Analysis & Window Size with Negligible Performance Degradation

• Decoding Complexity Analysis

• Concluding Remarks
Chapter 1:

Introduction
Viterbi decoding

- The most popular decoding algorithm for convolutional codes
- Decoding complexity increases exponentially with respect to the code constraint length
- Limited to constraint lengths of 10 or less
- When the information length is long, *path truncation* in practical decoder design
  - majority vote strategy
  - best state strategy
  - random state (fixed state) strategy
Best state strategy is commonly used

- Forney (1974) proved using *random coding argument* that a truncation window of 5.8-fold of the constraint length is enough
- Hemmati & Costello (1977) derived performance bound as a function of the truncation window for a *specific convolutional encoder* and obtained a similar conclusion
Sequential decoding (stack algorithm)

- The sequential decoding (stack algorithm or ZJ algorithm) was introduced by Zigangirov in 1966 and Jelinek in 1969.
- Decoding complexity is almost irrelevant to the code constraint length. Suitable for convolutional codes with memory order 12, 16, or even higher.
- After the development of the Viterbi algorithm in 1967, sequential decoding received little attention during the past 30 years due to its sub-optimum performance and lack of efficient hardware implementation.
- New application (Heegard 2003): Channel equalization of "super-code" considering the joint effect of multi-path channel & convolutional code.
• Maximum-Likelihood Sequential Decoding Algorithm (MLSDA):
  – proposed by Han, Chen, and Wu in 2002
  – change the decoding metric
  – achieve maximum-likelihood performance
  – decoding complexity considerably lower than Viterbi algorithm for medium to high SNRs from a software implementation standpoint
  – operate in trellis rather than tree due to the implementation of closed stack
Modification of MLSDA decoding

- Directly eliminate the top path with small level compared with the farest explored node

- Provide timely output (elimination: $\Delta$, decision: $5 \times (m + 1)$)
Performance & complexity analysis

- Level threshold $\Delta$ introduces a tradeoff between performance and complexity
- Analyze the smallest $\Delta$ value with negligible performance degradation
  - BSC: (random coding argument)
    * 2.2-fold constraint length for rate 1/2
    * 1.2-fold constraint length for rate 1/3
  - AWGN: (specific encoder)
    * 3-fold constraint length for rate 1/2
    * 2-fold constraint length for rate 1/3
- Analyze the decoding complexity using the large deviations technique & the Berry-Esseen theorem to find out how much decoding effort we can at least save
Chapter 2:

Convolutional Codes & Stack Algorithm with Fano Metric
Example for (2,1,2) codes and binary symmetric channel

Metric:
\[ M(1|1) = M(0|0) = +1 \]
\[ M(1|0) = M(0|1) = -9 \]
**Fano metric**

- Fano metric is the most common metric for sequential decoding for the past 30 years although some other metrics have been proposed.
- This metric was discovered by Fano (1963) through massive simulations.
- The bit metric for transmitting $v_j$ given receiving $r_j$ is

$$M(v_j|r_j) = \log_2 \left[ \frac{P(r_j|v_j)}{P(r_j)} \right] - R,$$

where $P(r_j|v_j)$ is the channel transition probability, $P(r_j)$ is the channel output symbol probability under equal prior, and $R$ denotes the code rate of the convolutional code.
• For example, under Binary Symmetric Channel (BSC) with crossover probability $p$,

$$M(v_j | r_j) = \begin{cases} 
\log_2(1 - p) + 1 - R, & \text{for } v_j = r_j; \\
\log_2(p) + 1 - R, & \text{for } v_j \neq r_j.
\end{cases}$$

For code rate $R = 1/2$ and crossover probability $p = 0.045$,

$$M(v_j | r_j) = \begin{cases} 
0.434, & \text{for } v_j = r_j; \\
-3.974, & \text{for } v_j \neq r_j,
\end{cases}$$

which can be scaled to

$$M(v_j | r_j) = \begin{cases} 
0.434 \times 2.30415 \approx 1, & \text{for } v_j = r_j; \\
-3.974 \times 2.30415 \approx -9, & \text{for } v_j \neq r_j.
\end{cases}$$

• Note that Fano metric can be positive or negative!
Begin

1 1 0 1 0 0 0 1 1 0 1 0 1 1

0

Diagram of nodes and connections.
Search step # 1:

1 1 0 1 0 0 0 1 1 0 1 0 1 1

-18

2
Search step # 3:

1 1 0 1 0 0 0 1 1 0 1 0 1 1

\[ -18 \]

\[ -16 \]

\[ -4 \]

\[ -4 \]
Search step # 4:

```
  1 1 0 1 0 0 0 1 1 0 1 0 1 1
```

-18

-16

-2

-22
Search step # 5:
Search step # 6:

```
1 1 0 1 0 0 0 1 1 0 1 0 1 1
```

![Diagram with values and nodes]
Search step # 7:

1 1 0 1 0 0 0 1 1 0 1 0 1 1 1

-18 -12 -18

-16 -12 -10

-22
Search step # 8:

1 1 0 1 0 0 0 1 1 0 1 0 1 1 1
Search step # 9:

1 1 0 1 0 0 0 1 1 0 1 0 1 1

-18

-16

-12

-12

-18

-22

-6
Final Output

1 1 0 1 0 0 0 1 1 0 1 0 1 1 1
Sub-optimality

• Massey (1972) proved that extending path with largest metric minimize the probability that the extending path does not belong to the optimal path.

• However, not always guarantee the finding of the global optimal path since the Fano metric may be positive.
Come back from behind due to positive metric?

1 1 0 1 0 0 0 1 1 0 1 0 1 1

-18 -12 -18 -6

-16 -12 -22
Performance with Fano metric (no quantization, no time out, perfect SNR est.)
Performance with Fano metric (quantization)
Complexity with Fano metric (code tree)

Average Decoding Complexity Per Information Bit

- (2,1,6) Viterbi = 128
- (2,1,12) Viterbi = 8192

$E_b/N_0$ vs. Average Decoding Complexity Per Information Bit
Chapter 3:

MLSDA & Early Elimination Scheme
Novel metric with ML performance & low complexity

- Han, Chen, and Wu (2002) derived another metric based on the Wagner rule.
- The path metric is defined as

\[ \mu(x_j) \triangleq (y_j \oplus x_j)|\phi_j|, \]

where \( \phi_j \) is the log-likelihood ratio, \( y_j \) is the hard decision and \( x_j \) is the encoder output.

- This metric is non-negative. It was proved that the metric is a maximum-likelihood metric.
Complexity of MLSDA

![Graph showing the average decoding complexity per information bit for different values of Eb/N0. The graph compares two codes: (2,1,10) and (2,1,6). The x-axis represents Eb/N0, ranging from 1 to 7, and the y-axis represents the average decoding complexity per information bit, ranging from 10^-1 to 10^4.]
Average decoding complexity per information bit of MLSDA

(2,1,10) Codes, AWGN, $E_b/N_0 = 3.5$ dB
Early elimination with level threshold $\Delta$

- update the farest explored level $\ell_{\text{max}}$
- if the level of the top path in stack is no larger than $(\ell_{\text{max}} - \Delta)$, directly eliminate the top path
Chapter 4:

Random Coding Argument & Performance Analysis for BSC channel
Galleger (1965) random coding exponent for \((N, K)\) block codes

- Discrete memoryless channel with input alphabet size \(I\), output alphabet size \(J\) and channel transition probability \(P_{ji}\)
- Maximum-likelihood decoding error \(P_e\) of the \((N, K)\) block code:

\[
P_e \leq \exp \left\{ -N \left[ -\rho R + E_0(\rho, \mathbf{p}) \right] \right\}
\]

for all \(0 \leq \rho \leq 1\), where \(R = \log(I^K)/N = (K/N) \log(I)\) is the code rate measured in nats per symbol, \(\mathbf{p} = (p_1, p_2, \cdots, p_I)\) is the input distribution adopted for the random selection of codewords, and

\[
E_0(\rho, \mathbf{p}) \triangleq -\log \sum_{j=1}^{J} \left( \sum_{i=1}^{I} p_i P_{ji}^{1/(1+\rho)} \right)^{1+\rho}
\]

(1)
• Gallager’s result leads to the well-known random coding exponent:

\[ E_r(R) \triangleq \max_{0 \leq \rho \leq 1} \max_p [-\rho R + E_0(\rho, p)] = \max_{0 \leq \rho \leq 1} [-\rho R + E_0(\rho)], \]

where \( E_0(\rho) \triangleq \max_p E_0(\rho, p) \) is the Gallager function.
Viterbi (1967) convolutional error exponent for time-varying convolutional codes

- Single-input \( n \)-output *time-varying convolutional codes*

- The \( n \) inner product computers may change with each new input symbol
• Viterbi showed that the ML decoding error $P_{e,c}$ for time-varying convolutional codes can be upper-bounded by:

$$P_{e,c} \leq \frac{1}{1 - 2^{-\lambda/R}} \exp[-n(m + 1)E_0(\rho)]$$

(2)

for all $0 \leq \rho \leq 1$, where $R \triangleq \log(2)/n$ is the code rate in unit of nats per symbol, and $\lambda \triangleq E_0(\rho) - \rho R$ is a constant independent of $n(m + 1)$

• $\lambda$ is required to be positive

• For symmetric channels, $E_0(\rho)$ is an increasing and concave function in $\rho$ with $E_0(0) = 0$; therefore, $E_c(R)$ can be reduced to:

$$E_c(R) \triangleq \max_{\{\rho \in [0,1] : E_0(\rho) > \rho R\}} E_0(\rho) = \begin{cases} 
R_0, & \text{if } 0 \leq R < R_0; \\
E_0(\rho^*), & \text{if } R_0 \leq R < C; \\
0, & \text{if } R \geq C,
\end{cases}$$

(3)
where $R_0 = E_0(1)$ is the cutoff rate, $C = E_0'(0)$ is the channel capacity, and $\rho^* = \rho^*(R)$ is the unique solution of $E_0(\rho) = \rho R$. It is also shown in the same work that $E_c(R)$ is a tight exponent for $R \geq R_0$. 

![Diagram showing Cross Over Probability 0.045, Capacity 0.735, Cutoff Rate 0.500 bits/symbol]
Forney (1974) window size for Viterbi decoder with truncation

- Treated the truncated convolutional code as a block code
- Upper-bounded the additional decoding error $P_{e,T}$ due to path truncation
  \[ P_{e,T} \leq \exp[-n\tau E_r(R)], \tag{4} \]
  where $\tau$ is the truncation window size
- Forney then noticed that as long as
  \[ \liminf_{n \to \infty} -\frac{1}{n} \log P_{e,T} > \limsup_{n \to \infty} -\frac{1}{n} \log P_{e,c}, \tag{5} \]
  the additional error $P_{e,T}$ due to path truncation becomes exponentially negligible with respect to $P_{e,c}$
- For $R \geq R_0$, condition (5) reduces to
  \[ \tau E_r(R) > (m + 1)E_c(R) \]
• BSC with crossover probability 0.4 (cutoff rate $R_0 = 0.0146$ bit/symbol)

$$\frac{E_c(R_0)}{E_r(R_0)} \approx \frac{0.0146}{0.0025} = 5.84$$
Random coding analysis for MLSDA with early elimination under BSC

- Additional error rate due to the transmitted path (i.e., all zero path) is early eliminated

- Unlike previous results, we are going to compare metric with un-equal length
• Node $C$ is expanded prior to node $B$ is equivalent to that

$$\mu \left( \mathbf{x}(\ell n-1) \right) \geq \mu \left( \mathbf{x}(\ell_{\text{max}}n-1) \right),$$

which in turn is equivalent to

$$(1 - \epsilon)^n(\ell_{\text{max}}-\ell) \cdot \Pr \left( \mathbf{r}(\ell n-1) \mid \mathbf{x}(\ell n-1) \right) \leq \Pr \left( \mathbf{r}(\ell_{\text{max}}n-1) \mid \mathbf{x}(\ell_{\text{max}}n-1) \right)$$

(7)

• The probability of early elimination is bounded by

$$P_{e,E} \leq K_n \cdot 2^{-\Delta n[-\rho R + \rho \log_2(1-\epsilon)/(1+\rho) + E_1(\rho)]},$$

(8)

where $K_n = \frac{2^{-\lambda/R}}{1-2^{-\lambda/R}}$ is a constant, independent of $\Delta$

• The error exponent

$$E_{el}(R) \triangleq \max_{\{\rho \in [0,1] : E_0(\rho) > \rho R\}} \left[ -\rho R + \rho \log_2(1-\epsilon)/(1+\rho) + E_1(\rho) \right]$$
Following similar argument, we conclude that the additional error due to early elimination in the MLSDA becomes exponentially negligible if

\[
\Delta \cdot E_{el}(R) > (m + 1)E_c(R)
\]

\[
\iff \quad \frac{\Delta}{m + 1} > \frac{E_c(R)}{E_{el}(R)} \tag{9}
\]

for convolutional code rates above the channel cutoff rate \( R_0 \).

For example,

\[
\Delta > \frac{0.4996}{0.2271} \times (m + 1) \approx 2.1999(m + 1) \text{ for rate } 1/2 \text{ codes}; \tag{10}
\]

\[
\Delta > \frac{0.3342}{0.2816} \times (m + 1) \approx 1.1868(m + 1) \text{ for rate } 1/3 \text{ codes.} \tag{11}
\]
BSC with crossover probability 0.045

Cross Over Probability 0.045, Cutoff Rate 0.5 bit/symbol

- Random Convolutional Code $E_c(R)$
- MLSDA Early Elimination $E_{el}(R)$
**BSC with crossover probability 0.095**

Cross Over Probability 0.095, Cutoff Rate 0.334 bit/symbol

Graph showing error exponent against rate for different codes.
Simulation results for (2,1,12) codes under BSC ($\Delta > 2.2 \times 13 = 28.6$)
Simulation results for (3,1,8) codes under BSC ($\Delta > 1.2 \times 9 = 10.8$)
Chapter 5:

Performance Analysis to a Specific Encoder under AWGN channel
Main idea of this chapter

- Find the BLER of ML decoder
- Analyze the additional BLER (as a function of early elimination threshold $\Delta$) due to the introduce of early elimination in MLSDA
- Find the smallest value of $\Delta$ such that the additional BLER is small compared with the BLER of ML decoder
Block error rate upper bound for finite-length convolutional codes with ML decoder

- The weight enumerator function (WEF) $A(x)$ is defined as

$$A(x) = \sum_{d=d_{\text{free}}}^{\infty} A_d x^d,$$

where $d_{\text{free}}$ is the free distance and $A_d$ denote the number of codewords with weight $d$

- For example, the WEF for (2,1,6) convolutional code with generator polynomial [554,744] (in oct) is

$$A(x) = 11x^{10} + 38x^{12} + 193x^{14} + 1331x^{16} + 7275x^{18} + 40406x^{20} + \cdots$$
• Define the *first event error* is made at time unit $t$ if the correct path is eliminated (in Viterbi decoder) for the first time at time unit $t$ in favor of a competitor path

• For infinite-length convolutional code under binary-input AWGN channel, the first event error probability $P_{ev}$ can be obtained

$$P_{ev} \leq \sum_{d=d_{free}}^{\infty} A_d Q \left( \sqrt{\frac{2dRE_b}{N_0}} \right)$$

(12)

• For finite-length convolutional code, the above upper bound is still valid

• With finite information length $L$, we obtain the BLER upper bound $P_B$ as

$$P_B = L \cdot P_{ev}$$

(13)
(2,1,6) convolutional codes with $L = 200$
(2,1,10) convolutional codes with $L = 200$

![Graph showing error rate vs. Eb/N0 for (2,1,10) Code with Message Length 200. The graph includes two lines: one for BLER UB = 200 * NER UB and another for BLER Simulation. The x-axis represents Eb/N0 ranging from 0 to 4.5, while the y-axis represents error rate ranging from $10^{-6}$ to $10^{1}$. The graph shows a downward trend as Eb/N0 increases.]
Probability for path $0_{(n\ell-1)}$ eliminated by $x_{(n(\ell+\Delta)-1)}$

- Let $x_{(n(\ell+\Delta)-1)}$ be the path with weight $d_1$ in the first $n\ell$ bits and weight $d_2$ in the last $n\Delta$ bits.
• Under the condition that 0 is transmitted, \( \{r_j\}_{j=1}^N \) is i.i.d. Gaussian with mean \( \mathcal{E} \) and variance \( N_0/2 \)

\[
\Pr\left(0_{(n\ell-1)} \text{ is EE due to } x_{(n(\ell+\Delta)-1)}\right)
= \Pr\left(0 \leq \sum_{j=1}^{d_1} (-r_j) + \sum_{j=d_1+1}^{d_1+d_2} (-r_j^+) + \sum_{j=d_1+d_2+1}^{d_1+n\Delta} (-r_j^-)\right),
\]

where

\[
r_j^+ = \begin{cases} 
  r_j, & \text{if } r_j > 0; \\
  0, & \text{elsewise,}
\end{cases}
\]

and

\[
r_j^- = \begin{cases} 
  0, & \text{if } r_j > 0; \\
  -r_j, & \text{elsewise.}
\end{cases}
\]
Moment generating bound

- The moment generating functions $M(t)$, $M_+(t)$, and $M_-(t)$ are as follows:

  \[
  M(t) = \exp \left( \frac{N_0 t^2}{4} - \mathcal{E} t \right)
  \]

  \[
  M_+(t) = Q \left( \frac{\mathcal{E}}{\sqrt{N_0/2}} \right) + \exp \left( \frac{N_0 t^2}{4} - \mathcal{E} t \right) Q \left( \frac{N_0 t - 2\mathcal{E}}{\sqrt{2N_0}} \right)
  \]

  \[
  M_-(t) = 1 - Q \left( \frac{\mathcal{E}}{\sqrt{N_0/2}} \right) + \exp \left( \frac{N_0 t^2}{4} + \mathcal{E} t \right) Q \left( \frac{N_0 t + 2\mathcal{E}}{\sqrt{2N_0}} \right)
  \]

- The probability is therefore upper bound by the moment generating bound

  \[
  \Pr \left( 0 \leq \sum_{j=1}^{d_1} (-r_j) + \sum_{j=d_1+1}^{d_1+d_2} (-r_j^+) + \sum_{j=d_1+d_2+1}^{d_1+n\Delta} (-r_j^-) \right) \leq [M(t)]^{d_1} \cdot [M_+(t)]^{d_2} \cdot [M_-(t)]^{n\Delta-d_2}, \quad \text{for } t > 0.
  \]
Weight distribution function

- Let $P_{\ell+\Delta}$ be the set of paths of length $n(\ell + \Delta)$ that diverge from $0_{(n\ell-1)}$ at level 0 and never return to all zero state.

- Define the weight distribution function for all paths in $P_{\ell+\Delta}$ by

$$W(\ell, \Delta) = \sum_{d_1,d_2} A_{d_1,d_2}(\ell, \Delta) \cdot w_1^{d_1} \cdot w_2^{d_2}, \quad (18)$$

where $A_{d_1,d_2}(\ell, \Delta)$ is the number of paths with weight $d_1$ in the first $n\ell$ bits and weight $d_2$ in the last $n\Delta$ bits.

- For example, for $(2,1,6)$ convolutional code with $g(x) = [554, 744]$,

$$W(2, 3) = 2w_1^3w_2 + 3w_1^3w_2^2 + 6w_1^3w_2^3 + 4w_1^3w_2^4 + w_1^3w_2^6.$$  

- Therefore, for each early elimination threshold $\Delta$

$$\Pr \left( 0_{(n\ell-1)} \text{ is EE} \right) \leq \sum_{d_1,d_2} A_{d_1,d_2}(\ell, \Delta) \cdot \min_{t>0} \left[ M(t)^{d_1} \cdot M_+(t)^{d_2} \cdot M_-(t)^{n\Delta-d_2} \right]$$
(2,1,6) Code with Message Length 200

Block Error Rate $\frac{E}{N_0}$

- BLER UB = BLER ML + NER UB $\Delta = 21$
- BLER UB = BLER ML + NER UB $\Delta = 22$
- BLER UB = BLER ML + NER UB $\Delta = 23$
- BLER ML Bler UB
Block Error Rate

$\frac{E_b}{N_0}$

(2,1,10) Code with Message Length 200

$\Delta = 31$

$\Delta = 32$

$\Delta = 33$

$\Delta = 34$

BLER UB = BLER ML + NER UB
(2,1,6), Message Length 200

Block Error Rate vs. $E_b / N_0$ for different values of $\Delta$:
- $\Delta=16$
- $\Delta=18$
- $\Delta=20$
- ML
(2,1,10), Message Length 200

Block Error Rate vs. $E_b/N_0$ for different values of $\Delta$: 24, 26, 28, and ML.
Table 1: Suggested $\Delta$ values for rate one-half convolutional codes

<table>
<thead>
<tr>
<th>Memory Order $m$</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generator Polynomial</td>
<td>554</td>
<td>561</td>
<td>4672</td>
<td>42554</td>
<td>56721</td>
<td>716502</td>
</tr>
<tr>
<td></td>
<td>744</td>
<td>753</td>
<td>7542</td>
<td>77304</td>
<td>61713</td>
<td>514576</td>
</tr>
<tr>
<td>Minimum Distance $d_{\text{min}}$</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>Sufficiently Large $\Delta$ by UB</td>
<td>23</td>
<td>29</td>
<td>33</td>
<td>38</td>
<td>42</td>
<td>46</td>
</tr>
<tr>
<td>Sufficiently Large $\Delta$ by Simulation</td>
<td>20</td>
<td>24</td>
<td>28</td>
<td>32</td>
<td>36</td>
<td>40</td>
</tr>
</tbody>
</table>
Chapter 6:

Complexity Analysis Based on Large Deviations Technique & Berry-Esseen Theorem
The Berry-Esseen theorem

- The large deviations technique is generally applied to compute the exponent of an exponentially decaying probability mass.

- The Berry-Esseen inequality to evaluate the subexponential detail of the concerned probability to obtain a better complexity upper bound.

- Let $\{X_i\}_{i=1}^n$ be independent zero-mean random variables. For every $a \in \mathbb{R}$,

$$
\left| \Pr \left\{ \frac{1}{s_n} (X_1 + \cdots + X_n) \leq a \right\} - \Phi(a) \right| \leq C' \frac{r_n}{s_n^3},
$$

(19)

where $\Phi(\cdot)$ represents the unit Gaussian cdf, $s_n^2$ and $r_n$ are, respectively, the sums of the marginal variances and the marginal absolute third moments, and the Berry-Esseen coefficient, $C'$, is an absolute constant.
Upper bound for sum of i.i.d. non-Gaussian rv

**Lemma 1 (main idea)** Let $Y_n = \sum_{i=1}^{n} X_i$ be the sum of i.i.d. random variables. Then, for every $\theta < 0$,

$$\Pr \{ Y_n \leq -n\alpha \} \leq A_n(\theta, \alpha)e^{\theta\alpha n}M^n(\theta),$$

where $M(\theta) \triangleq E[e^{\theta X_1}]$ and

$$A_n(\theta, \alpha) = \min\{B_n(\theta, \alpha), 1\} \approx \min\{2C\frac{\rho(\theta)}{\sigma^3(\theta)\sqrt{n}}, 1\}.$$

- Note: Chernoff bound $e^{\theta\alpha n}M^n(\theta)$
Upper bound for sum of two i.i.d. sequences

Lemma 2 Let \( Y_n = \sum_{i=1}^{n} X_i \) be the sum of independent random variables \( \{X_i\}_{i=1}^{n} \), among which \( \{X_i\}_{i=1}^{d} \) are identically Gaussian distributed with positive mean \( \mu \) and non-zero variance \( \sigma^2 \), and \( \{X_i\}_{i=d+1}^{n} \) have common marginal distribution as \( \min\{X_1, 0\} \). Let \( \gamma \triangleq (1/2)(\mu^2/\sigma^2) \). Then

\[
\Pr \left\{ Y_n \leq 0 \right\} \leq \mathcal{B}(d, n-d, \gamma),
\]
where

\[ B(d, n - d, \gamma) = \begin{cases} 
\Phi(-\sqrt{2\gamma n}), & \text{if } d = n; \\
\Phi\left(-\frac{(n-d)\mu+d\sqrt{2\gamma}}{\sqrt{d}}\right) + \tilde{A}_{n-d}(\lambda) \\
\times \left[ \Phi(-\lambda)e^{-\gamma}e^{\lambda^2/2} + \Phi(\sqrt{2\gamma}) \right]^{n-d} \\
\times e^{d(-\gamma+\lambda^2/2)}\Phi\left(\frac{(n-d)\mu+\lambda d}{\sqrt{d}}\right), & \text{if } 1 > \frac{d}{n} \geq 1 - \frac{\sqrt{4\pi \gamma e^\gamma}}{1+\sqrt{4\pi \gamma e^\gamma}\Phi(\sqrt{2\gamma})}; \\
1, & \text{otherwise.}
\end{cases} \]

- Main idea of proof

\[
\Pr(Y_n \leq 0) = \Pr\{X_1 + \cdots + X_d + X_{d+1} + \cdots + X_n \leq 0\} \\
= \int_{-\infty}^{\infty} \Pr\{X_{d+1} + \cdots + X_n \leq -x\} \frac{1}{\sqrt{2\pi d\sigma^2}} e^{-\frac{(x-d\mu)^2}{2d\sigma^2}} dx \\
= \cdots \text{ (apply Lemma 1 for non-Gaussian sequence)}
\]
• If Chernoff bound is used instead, $\tilde{A}_{n-d}(\lambda) = 1$.

• For large $n$, $\tilde{A}_{n-d}(\lambda) < 1$ and the complexity upper bound improves.

\[
\frac{d}{n} = 0.2
\]
Computation complexity for MLSDA

- Denote by $s_j(\ell)$ the node that is located at level $\ell$ and corresponds to state index $j$
- Let $S_j(\ell)$ be the set of paths that end at node $s_j(\ell)$
- Let $\mathcal{H}_j(\ell)$ be the set of the Hamming weights of the paths in $S_j(\ell)$
- Denote the minimum Hamming weight in $\mathcal{H}_j(\ell)$ by $d_j^*(\ell)$
• As an example, $S_3(3)$ equals $\{111010001, 000111010\}$ in the following figure, which results in $\mathcal{H}_3(3) = \{5, 4\}$ and $d^*_3(3) = 4$. 
• Let $x^*$ label the minimum-metric code path for a given log-likelihood ratio $\phi$

Path $s_j(\ell)$ is extended

\[
\Leftrightarrow M(x_{(\ell_{n-1})}|\phi_{(\ell_{n-1})}) \leq M(x^*|\phi) \\
\Rightarrow M(x_{(\ell_{n-1})}|\phi_{(\ell_{n-1})}) \leq M(0|\phi).
\]

Therefore,

\[
\Pr \{ \text{node } s_j(\ell) \text{ is extended by the MLSDA} \} \\
\leq \max_{d \in H_j(\ell)} \Pr \left\{ r_1 + \cdots + r_d + \sum_{j=\ell n}^{N-1} \min(r_j, 0) \leq 0 \right\} \\
= \Pr \left\{ r_1 + \cdots + r_{d^*_j(\ell)} + \sum_{j=\ell n}^{N-1} \min(r_j, 0) \leq 0 \right\} \\
\leq B \left( d^*_j(\ell), N - \ell n, \frac{kL}{N} \gamma_b \right).
\]
Theorem 1 (Complexity of the MLSDA) Consider an \((n, k, m)\) binary convolutional code transmitted via an AWGN channel. The average number of branch metric computations evaluated by the MLSDA, denoted by \(L_{\text{MLSDA}}(\gamma_b)\), is upper-bounded by

\[
L_{\text{MLSDA}}(\gamma_b) \leq 2^k \sum_{\ell=0}^{L-1} \sum_{j=0}^{2^m-1} B\left(d_j^*(\ell), N - \ell n, \frac{kL}{N} \gamma_b \right),
\]

where if \(\mathcal{H}_j(\ell)\) is empty, implying the non-existence of state \(j\) at level \(\ell\), then \(B(d_j^*(\ell), N - \ell n, kL\gamma_b/N) = 0\).
Computation complexity for MLSDA with early elimination

- MLSDA with early elimination, in the “normal case”

\[ \Pr \{ \text{node } s_j(\ell) \text{ is extended by the MLSDA with EE in the normal case} \} \leq B \left( d_j^*(\ell), \Delta n, \frac{kL}{N} \gamma_b \right). \]
• "Abnormal case": The first expanded path $x^o_{((\ell+\Delta)n-1)} > 0_{((\ell+\Delta)n-1)}$
  - $x^o_{((\ell+\Delta)n-1)}$ may not be the best path at the same level
  - abnormal case $\Rightarrow$ early elimination of $0_{((\ell+\Delta)n-1)}$
  - probability can be upper bounded by the BLER of MLSDA with EE

• Overall complexity upper bound

\[
L_{EE}(\gamma_b) \leq 2^k \sum_{\ell=0}^{L-1} \sum_{j=0}^{2^m-1} \left[ P_{EE}(\gamma_b, \Delta) + B \left( d^*_j(\ell), \Delta n, \frac{kL}{N} \gamma_b \right) \right],
\]

\[
= L \cdot 2^{k+m} \cdot P_{EE}(\gamma_b, \Delta) + 2^k \sum_{\ell=0}^{L-1} \sum_{j=0}^{2^m-1} B \left( d^*_j(\ell), \Delta n, \frac{kL}{N} \gamma_b \right),
\]

where $P_{EE}(\gamma_b, \Delta)$ denotes the block error rate of MLSDA with early elimination window $\Delta$ under SNR $\gamma_b$. 
Complexity UB vs simulation result for (2,1,10) code with $L = 100$

(2,1,10) Convolutional codes, $\Delta = 30$

- UB MLSDA
- Sim MLSDA
- UB MLSDA with EE
- UB MLSDA with EE ignoring BLER term
- Simulation
Complexity UB vs simulation result for (2,1,10) code with SNR = 3.5 dB
Simulation result for complexity vs memory order

- MLSDA SNR = 3 dB
- MLSDA with EE SNR = 3 dB
- MLSDA SNR = 4 dB
- MLSDA with EE SNR = 4 dB
- MLSDA SNR = 5 dB
- MLSDA with EE SNR = 5 dB
Chapter 7:

Concluding Remarks
Early elimination is proposed to reduce decoding complexity of MLSDA.

Performance analyses suggest the sufficient threshold value for negligible performance degradation.

Complexity analysis confirms the large complexity reduction compared with the original MLSDA.

The performance and complexity analytic results are further justified by simulation results.
Acknowledgement

- Industrial Technology Research Institute (ITRI)
- Sunplus Technology Co., Ltd. (Sunplus)
- Sunplus mMobile Inc. (SunplusMM)
Thank You for Your Attention!

Questions & Answers