A maximum likelihood estimator of can be chosen that is differentiable and

\[ \theta \text{ is the natural parameter} \]

\[ x \text{ is the natural statistic} \]

\[ h(x) \text{ is the nuisance statistic} \]

Maximum likelihood estimation of can be done by

\[ \text{V. Invar. Algoston} \]
Binary Symmetric Channel

code vector $x$ and received vector $y$ are $1 \times N$
binary sequences.

$$p(y|\bar{x}) = \prod_{i=1}^{N} p(y_i|x_i)$$

so

$$\ln(p(y|x)) = \sum_{i=1}^{N} \ln p(y_i|x_i)$$

where

$$p(y_i|x_i) = \begin{cases} p & \text{if } y_i \neq x_i \\ 1-p & \text{if } y_i = x_i \end{cases}$$

Assume a Hamming distance of $d$ between $y$ and $x$

$$\ln p(y|x) = d \ln p + (N-d) \ln(1-p)$$

$$= d \ln \left(\frac{p}{p-1}\right) + N \ln(1-p)$$

Assume $p < \frac{1}{2}$ and $N \ln(1-p)$ is constant $\theta$

so we will choose $\hat{x}$ that minimizes Hamming distance
between $\hat{x}$ and $y$.
Viterbi Algorithm

VA is a maximum-likelihood decoder which is optimum for WGN. We choose the path through the codetree or trellis that minimizes Hamming distance.

Let's do the first example

Code word: 01, 00, 01, 00, 00

Choose a survivor

2 paths intersect at a node so we can choose one survivor for 4 without knowing future path.
Step 1:

- $i = 2$
- $\text{level} = 2$

Reflected line = 0, dashed line = 1

Step 2:

- $i = 4$

Diagram with nodes and connections.
Figure 8.12 Trellis for the convolutional encoder of Fig. 8.10a.

Figure 8.13 A portion of the central part of the trellis for the encoder of Fig. 8.10a.
Note that the memory is $M = K - 1$, where $K$ is the constraint length of the code.

The great advantage of the Viterbi algorithm is that (for a constraint length $K$, code rate $r = kn$, convolutional code) the number of operations performed in decoding $L$ bits is $L2^{u(k-1)}$, which is linear in $L$. However, the number of operations performed per decoded bit is an exponential function of the constraint length $K$. This exponential dependence on $K$ limits the utilization of the Viterbi algorithm as a practical decoding technique to relatively short constraint-length codes (typically, in the range of 7 to 11).

**EXAMPLE 9**

Suppose that the encoder of Fig. 8.10a generates an all-zero sequence that is sent over a binary symmetric channel, and that the received sequence is (0100010000 ... ). There are two errors in the received sequence due to noise in the channel: one in the second location and the other in the sixth location. We wish to show that this double-error pattern is correctable through the application of the Viterbi decoding algorithm.

In Fig. 8.15a, we show the result of applying step 1 of the algorithm for level $j = 2$. There are four paths (survivors), one for each of the four states of the encoder. The figure also includes the metric of each path.

In the left side of Fig. 8.15b, we show the two paths entering each of the four states at level $j = 3$, together with their individual metrics. In the right side of this figure, we show the four survivors that result from applying step 2 of the algorithm for $j = 3$. In Figs. 8.15c and 8.15d, we show the corresponding results obtained from application of step 2 of the algorithm for level $j = 4$ and $j = 5$, respectively.

Examining the four survivors in Fig. 8.15d, we see that the all-zero path has the smallest metric. This clearly shows that the all-zero sequence is the maximum likelihood choice of the Viterbi decoding algorithm, which agrees exactly with the transmitted sequence.

**EXAMPLE 10**

Suppose next that the received sequence is (1100010000 ... ), which contains three errors compared to the transmitted all-zero sequences.

In Fig. 8.16a, we show the paths (survivors) that result from applying Step 1 of the Viterbi decoding algorithm for $j = 2$.

In Figs. 8.16b and 8.16c, we show the survivors that result from applying step 2 of the algorithm for $j = 3$ and $j = 4$, respectively.

We see that in this example the correct path has been eliminated by level $j = 3$. Clearly, a triple-error pattern is uncorrectable by the Viterbi algorithm when applied to a convolutional code of rate $1/2$ and constraint length $K = 3$.

The explanation for why this convolutional code can correct up to two errors in the received sequence (as in Example 9), but fails when there are three errors (as in Example 10), will be presented in the next section.

### 8.7 DISTANCE PROPERTIES OF CONVOLUTIONAL CODES

The performance of a convolutional code depends not only on the decoding algorithm used but also on the distance properties of the code. In this context, the most important single measure of a convolutional code’s ability to combat
Figure 8.15  Illustrating steps in the Viterbi algorithm.