INTRODUCTION TO COMMUNICATION SYSTEMS (CT216)

Analytical Proofs :

Lab Group - 2 : Project Group - 2

Prof. Yash Vasavada
Mentor TA : Shrey Andhariya

TOPIC : CONVOLUTIONAL CODING

Honor code :

- The work that we are presenting is our own work.
- We have not copied the work (the code,the results,etc.) that someone else has done.
- Concepts,understanding and insights we will be describing are our own.
- We make this pledge truthfully.
- We know that violation of this solemn pledge can carry grave consequences.

➢ Performance of soft decision decoding: -

- Let's say we sent a message symbolized by "c." Each bit of that message is represented as c_{jm}, where j shows the j^{th} branch and m shows the m^{th} bit of that branch.

$$r_{jm} = \sqrt{\varepsilon_c}\left(2C_{jm} - 1\right) + n_{jm}$$

c_{jm} = Bit at the m^{th} position of the j^{th} branch of the transmitted sequence

r_{jm} = Bit at the m^{th} position of the j^{th} branch of the received sequence.

ε_c = Transmitted signal energy.

n_{jm} = Noise introduced by the channel.

- The equation of Branch Metric(μ_j) for i^{th} path and j^{th} branch is,

$$\mu_j = \log(P(Y_j | C_j^{(i)}))$$

- Hence we can obtain the equation of Path Metrics($PM^{(i)}$) for the i^{th} having B branches :-

$$PM^{(i)} = \sum_{j=1}^{B} \mu_j^{(i)}$$

- The above equation of path metrics calculates Hamming distance, which is an equivalent metric for hard-decision decoding.

$$p(r_{jm} | c_{jm}^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{r_{jm} - \sqrt{\varepsilon_c}(2c_{jm} - 1)}{\sigma}\right)^2}$$

Where, $\sigma^2 = \frac{N_0}{2}$ is the variance of the noise.

Now,

$$\mu_j^{(i)} = \sum_{m=1}^{n} \log\left(P\left(r_{jm} \middle| c_{jm}^{(i)}\right)\right)$$

$$= \sum_{m=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{\left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2}{2\sigma^2}\right)}\right)$$

$$= \sum_{m=1}^{n} \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \sum_{m=1}^{n} e^{-\left(\frac{\left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2}{2\sigma^2}\right)}$$

$$= \sum_{m=1}^{n} \left(-\frac{1}{2}\right) \log(2\pi\sigma^2) - \sum_{m=1}^{n} \left(\frac{\left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2}{2\sigma^2}\right)$$

$$= \left(-\frac{n}{2}\right) \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{m=1}^{n} \left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2$$

$$= K_1 + K_2 \sum_{m=1}^{n} \left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2$$

- $CM^{(i)} = \sum_{j=1}^{B} \mu_j^{(i)}$
- $CM^{(i)} = \sum_{j=1}^{B} \left(K_1 + K_2 \sum_{m=1}^{n} \left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2\right)$
- $CM^{(i)} = BK_1 + K_2 \sum_{j=1}^{B} \sum_{m=1}^{n} \left(r_{jm}-\sqrt{\mathcal{E}_c}\left(2c_{jm}-1\right)\right)^2$

- Here $CM^{(i)}$ is the correlation metric is sum of all branch metrics up to branch B.
- We can ignore the constant terms that are common to all branch metrics (BK_1 is the common term added and K_2 is the common term multiplied by each branch metric).

So, by neglecting them, we get

- $CM^{(i)} = \sum_{j=1}^{B}\left(\sum_{m=1}^{n}\left(r_{jm} - \sqrt{\mathcal{E}_c}\left(2c_{jm} - 1\right)\right)^2\right)$

- When we're figuring out the error probability for convolutional codes, we lean on the linearity property to simplify things. Let's say we sent an all-zero sequence and now we're figuring out how likely it is to make an error by favouring a different sequence instead. The binary digits for the j^{th} branch of the code, known as C_{jm}, are sent using binary phase-shift keying (binary-PSK) and then coherently detected at the demodulator. The demodulator's output, which is what the Viterbi decoder works with, is called r_{jm}.

- Considering i=0 as all zero path. The path matric for all zero paths (C_{jm} = 0) is

$$CM^{(0)} = \sum_{i=1}^{B}\sum_{m=1}^{n}(\sqrt{\varepsilon_c}\left(2C_{jm} - 1\right) + n_{jm})\left(2C_{jm}{}^{(i)} - 1\right)$$

$$CM^{(0)} = \sum_{j=1}^{B}\sum_{m=1}^{n}(\sqrt{\varepsilon_c}(2(0) - 1) + n_{jm})(2(0) - 1)$$

$$CM^{(0)} = \sum_{j=1}^{B}\sum_{m=1}^{n}(-\sqrt{\varepsilon_c} + n_{jm})(-1)$$

$$CM^{(0)} = \sqrt{\varepsilon_c}nB - \sum_{j=1}^{B}\sum_{m=1}^{n} n_{jm}$$

- P_2(d) = It is the first event probability when an incorrect path merges with all zero paths for the first time.
- Suppose the incorrect path differs from the all zero path in d bits (i.e. there are d bits in the incorrect path).
- Considering the two paths, the i^{th} path and the all zero path (i=0).
- Hence, we can say for the correct path $CM^{(i)} < CM^{(0)}$, but an error will occur when the $CM^{(i)} \geq CM^{(0)}$.

$$\therefore P_2(\mathrm{d}) = P\left(CM^{(i)} \geq CM^{(0)}\right)$$

$$= P\left(CM^{(i)} - CM^{(0)} \geq 0\right)$$

- Now, we have the equation of $CM^{(0)}$ and $CM^{(i)}$ as below,

$$CM^{(0)} = \sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm}(2c_{jm}^{(0)} - 1)$$

$$CM^{(i)} = \sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm}(2c_{jm}^{(i)} - 1)$$

- Now applying it in above equation,

$$P\left(\sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm}\left(2c_{jm}^{(i)} - 1\right) - \sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm}(2c_{jm}^{(0)} - 1) \geq 0\right)$$
$$= P\left(2\sum_{j=1}^{B} \sum_{m=1}^{n} r_{jm}\left(c_{jm}^{(i)} - c_{jm}^{(0)}\right) \geq 0\right)$$

- We have assumed above that both paths differed by d bits, so $c_{jm}^{(i)} - c_{jm}^{(0)}$ **is zero** for all other bits.
- So, the above formula will be reduced to,

$$P\left(\sum_{l=1}^{d} r_l{}' \geq 0\right)$$

Here, l is the set of differing d bits and set $\{r_l'\}$ represents input to the decoder for these d bits.

- How does the equation of Expectation and variance come?

Expectation, $\boldsymbol{E\{r_l'\} = E\{\sqrt{\varepsilon_c}(0-1) + n_{jm}\}}$

$$\boldsymbol{E\{r_l'\} = E\{-\sqrt{\varepsilon_c} + n_{jm}\}}$$

$$\boldsymbol{E\{r_l'\} = E\{-\sqrt{\varepsilon_c}\} + E\{n_{jm}\}}$$

$$\boldsymbol{E\{r_l'\} = -\sqrt{\varepsilon_c}}$$

Variance, $\boldsymbol{var\{r_l'\} = var\{-\sqrt{\varepsilon_c} + n_{jm}\}}$

$$\boldsymbol{var\{r_l'\} = 0 + var\{n_{jm}\}}$$

$$\boldsymbol{var\{r_l'\} = \frac{N_0}{2}}$$

- Here, $\{r_l'\}$ ~ $\mathrm{N}(-\sqrt{\varepsilon_c}, N_0/2)$ is an Identically and Independently Gaussian distributed variable.
- Now, (using central limit theorem)

$$\boldsymbol{\sum_{l=1}^{d} r_l' \sim N\left(-d\sqrt{\varepsilon_c}, \frac{dN_0}{2}\right)}$$

- We can convert it into a standard normal distribution $Z \sim N(0,1)$

$$\boldsymbol{P\left(\sum_{l=1}^{d} r_l' \geq 0\right)}$$

$$\boldsymbol{= P\left(\frac{\sum_{l=1}^{d} r_l' - (-d\sqrt{\mathcal{E}_c})}{\left(\sqrt{\frac{d}{2}N_0}\right)} \geq \frac{0 - (-d\sqrt{\mathcal{E}_c})}{\left(\sqrt{\frac{d}{2}N_0}\right)}\right)}$$

$$\boldsymbol{= P\left(Z \geq \frac{0 - (-d\sqrt{\mathcal{E}_c})}{\left(\sqrt{\frac{d}{2}N_0}\right)}\right)}$$

$$= \boldsymbol{P}\left(\boldsymbol{Z} \geq \frac{\sqrt{\mathbf{2}\boldsymbol{d}\mathcal{E}_{\boldsymbol{c}}}}{\sqrt{\boldsymbol{N}_{\mathbf{0}}}}\right)$$

$$= \boldsymbol{P}\left(\boldsymbol{Z} \geq \sqrt{\frac{\mathbf{2}\boldsymbol{d}\mathcal{E}_{\boldsymbol{c}}}{\boldsymbol{N}_{\mathbf{0}}}}\right)$$

$$= \boldsymbol{Q}\left(\sqrt{\frac{\mathbf{2}\boldsymbol{d}\mathcal{E}_{\boldsymbol{c}}}{\boldsymbol{N}_{\mathbf{0}}}}\right)$$

$$\because \left[\boldsymbol{Q}(\boldsymbol{x}) = \frac{\mathbf{1}}{\sqrt{\mathbf{2}\boldsymbol{\pi}}}\int_{\boldsymbol{x}}^{\infty} \boldsymbol{e}^{-\frac{\boldsymbol{x}^{2}}{2}}\,\boldsymbol{dx}\right]$$

Where,

$\mathcal{E}_c$ = energy per transmitted code

$\mathcal{E}_b$ = energy per transmitted bit

N_0 = the number of bits

- Now, $SNR(\boldsymbol{\gamma_b}) = \frac{\boldsymbol{\varepsilon_b}}{\boldsymbol{N_0}}$
- And, $\boldsymbol{\varepsilon_b} = \frac{\boldsymbol{\varepsilon_c}}{\boldsymbol{R_c}}$

  $\therefore \left(\frac{\boldsymbol{\varepsilon_b}}{\boldsymbol{N_0}} = \frac{\boldsymbol{\varepsilon_c}}{\boldsymbol{R_c N_0}} = \boldsymbol{\gamma_b}\right)$

- $\mathrm{R_c} = \text{Code Rate}$

$$\boxed{\boldsymbol{P}_2(\boldsymbol{d}) = \boldsymbol{Q}\left(\sqrt{\mathbf{2}\boldsymbol{\gamma_b R_c d}}\right)} \qquad \ldots.\boldsymbol{equation}(\mathbf{1})$$

- Now there can be multiple incorrect paths that can merge with the all-zero path.

- Summing up the error probability over all possible path distances. So, we obtain the upper bound on a first event error probability in the form

$$\boldsymbol{P_e \leq \sum_{d=d_{free}}^{\infty} a_d P_2(d)}$$

- a_d = Number of paths having distance d from the all zero path
- Now, $P_2(d) = Q\left(\sqrt{2\gamma_b R_c d}\right)$

$$\boxed{\boldsymbol{P_e \leq \sum_{d=d_{free}}^{\infty} a_d Q\left(\sqrt{2\gamma_b R_c d}\right)}} \qquad \ldots.\boldsymbol{equation(2)}$$

- Considering the upper-bound of the Q function

$$\boldsymbol{Q\left(\sqrt{2\gamma_b R_c d}\right) \leq e^{-\gamma_b R_c d}}$$

- Replacing the Q function with the upper bound in the equation(2)

$$\boldsymbol{P_e \leq \sum_{d=d_{free}}^{\infty} a_d\ e^{-\gamma_b R_c d}}$$

- Replacing D by $e^{-\gamma_b R_c}$ and obtaining the final equation of $\boldsymbol{P_e}$

$$\boldsymbol{P_e \leq \sum_{d=d_{free}}^{\infty} a_d\ D^d}$$

$$where\ , \boldsymbol{D = e^{-\gamma_b R_c}}$$

➢ Transfer Function

$$T(D) = \sum_{d=d_{free}}^{\infty} a_d \, D^d$$

- Where D is the hamming distance
- Also, we can write $P_e \leq T(D)$ theoretically but practically $P_e < T(D)$

$$\therefore P_e \leq T(D)$$

$$where\ D = e^{-\gamma_b R_c}$$

- Now we need to determine the bit error probability.
- Considering the transfer function $T(D, N)$ where f(d) indicates the number of bit errors in selecting an incorrect path that merges with the all zero path at some node B.

$$T(D, N) = \sum_{d=d_{free}}^{\infty} a_d \; D^d \; N^{f(d)}$$

- Taking the derivative of T(D,N) w.r.t N and then placing N=1.
- If we multiply $P_2(d)$ by number of incorrectly decoded information bit (f(d)), and then sum up all the possible path we obtain BER(P_b).

$$\left(\frac{d}{dN} T(D, N)\right)_{N=1} = \sum_{d=d_{free}}^{\infty} a_d \; D^d \; f(d)$$

$$= \sum_{d=d_{free}}^{\infty} B_d \; D^d$$

$$where\ , B_d = \; a_d \, f(d)$$

- Now the probability of bit error rate, P_b
- Bit error probability for k = 1 is upper bounded by,

$$P_b < \sum_{d=d_{free}}^{\infty} a_d f(d) P_2(d)$$

$$P_b < \sum_{d=d_{free}}^{\infty} B_d P_2(d)$$

$$P_b < \sum_{d=d_{free}}^{\infty} B_d Q\left(\sqrt{2\gamma_b R_c d}\right)$$

- Now,

$$Q\left(\sqrt{2\gamma_b R_c d}\right) \le D^d$$

$$P_b < \sum_{d=d_{free}}^{\infty} B_d D^d = \frac{dT(D,N)}{dN}$$

$$P_b < \frac{dT(D,N)}{dN}$$

- Bit error probability is upper bounded by the differentiation of transfer function w.r.t N

$$P_b < \left.\frac{dT(D,N)}{dN}\right|_{N=1, D=e^{-\gamma_b R_c}}$$

➢ Performance of Hard decision decoding: -

- We denote "p" as the probability of a bit flip occurring in the binary symmetric channel.
- We start with the assumption that the all-zero path has been transmitted. Our objective is to determine the first-event error probability, denoted as $P_2(d)$.
- To achieve this, we examine the output of a specific path and compare it to an all-zero sequence at a particular node B. Node B is located at a distance d from the all-zero path.

- When **d= odd**

$$\boldsymbol{P_2(d) = \sum_{k=\frac{(d+1)}{2}}^{d} \binom{d}{k} p^k (1-p)^{d-k}}$$

This is because when d is odd from $k = \frac{(d+1)}{2}$ to d we can only detect the incorrect path.

- From the concept of Hamming distance we can only correct t_c(error correction capacity) = $\left\lfloor \frac{d-1}{2} \right\rfloor$, so a number of errors greater than t_c cannot be corrected and produces an error.
- When d = even:

$$\boldsymbol{P_2(d) = \sum_{k=d/2+1}^{d} \binom{d}{k} p^k (1-p)^{d-k} + \frac{1}{2}\binom{d}{d/2} p^{d/2}(1-p)^{d/2}}$$

The first term is similar to the term when d is odd, the second term is when we have d distance it is in the middle and can't decide what path to detect so ½ chance of both paths can be detected.

- Using upper bounds to simplify the equation we get,

$$\boldsymbol{P_2(d) = \sum_{k=\frac{(d+1)}{2}}^{d} \binom{d}{k} p^k (1-p)^{d-k}}$$

$$\leq \sum_{k=\frac{(d+1)}{2}}^{d} \binom{d}{k} p^{d/2}(1-p)^{d/2}$$

$$= p^{d/2}(1-p)^{d/2} \sum_{k=\frac{(d+1)}{2}}^{d} \binom{d}{k}$$

$$\leq p^{d/2}(1-p)^{d/2} \sum_{k=0}^{d} \binom{d}{k}$$

$$\leq \ 2^d p^{d/2}(1-p)^{d/2}$$

For Even d,

$$\mathbf{P} = 1/2 \binom{d}{d/2} p^{d/2}(1-p)^{d/2} + \sum_{e=(d+1)/2}^{d} \binom{d}{e} p^e (1-p)^{d-e}$$

$$< \sum\nolimits_{e=d/2}^{d} \binom{d}{e} p^e (1-p)^{d-2}$$

$$< \sum\nolimits_{e=d/2}^{d} \binom{d}{e} p^{d/2}(1-p)^{d/2}$$

$$= \ p^{d/2}(1-p)^{d/2} \sum\nolimits_{e=d/2}^{d} \binom{d}{e}$$

$$< \ p^{d/2}(1-p)^{d/2} \sum\nolimits_{e=0}^{d} \binom{d}{e}$$

$$= \ 2^d p^{d/2}(1-p)^{d/2}$$

- So, finally we will get the simplified term as follows,

$$P_2(d) < [4p(1-p)]^{d/2}$$

- If we solve the equation for d=even we will get a similar equation for $\boldsymbol{P_2(d)}$.
- Now summing up all the first event error probabilities we get the overall first event probability $\boldsymbol{P_e}$

$$\boldsymbol{P_e < \sum_{d=d_{free}}^{\infty} a_d[4p(1-p)]^{d/2}}$$

- We can now infer transfer function **T(D)** as following,

$$\boldsymbol{P_e < T(D)\ where, D = \sqrt{4p(1-p)}}$$

- From the definitions of transfer functions mentioned above,

$$\boldsymbol{P_e < T(D)\ where, D = \sqrt{4p(1-p)}}$$

- Considering the transfer function $\boldsymbol{T(D,N)}$ **where** f(d) indicates the number of bit errors in selecting an incorrect path that merges with the all zero path at some node B.

$$\boldsymbol{T(D,N) = \sum_{d=d_{free}}^{\infty} a_d\ D^d\ N^{f(d)}}$$

- Taking the derivative of T(D,N) w.r.t N and then setting N=1.

$$\boldsymbol{\frac{d}{dN}T(D,N)|_{(N=1)} = \sum_{d=d_{free}}^{\infty} a_d\ D^d\ f(d)}$$

$$\boldsymbol{= \sum_{d=d_{free}}^{\infty} B_d\ D^d}$$

$$\boldsymbol{where\ , B_d = a_d\, f(d)}$$

- Consider the probability of bit error rate, $\boldsymbol{P_b}$.

- Useful measure for the performance of Convolution code is given by bit error probability.
- $\boldsymbol{P_b}$ is the bit error probability.
- Bit error probability for k = 1 is upper bounded by,

$$\boldsymbol{P_b} < \sum_{\boldsymbol{d=\,d_{free}}}^{\infty} \boldsymbol{B_d P_2(d)}$$

- Bit error probability is upper bounded by the differentiation of transfer function w.r.t N.

$$\boldsymbol{P_b} < \left.\frac{\boldsymbol{dT(D,N)}}{\boldsymbol{dN}}\right|_{\boldsymbol{N=1,D=\sqrt{4p(1-p)}}}$$

Thank You