ELE-32 Introdução a Comunicações

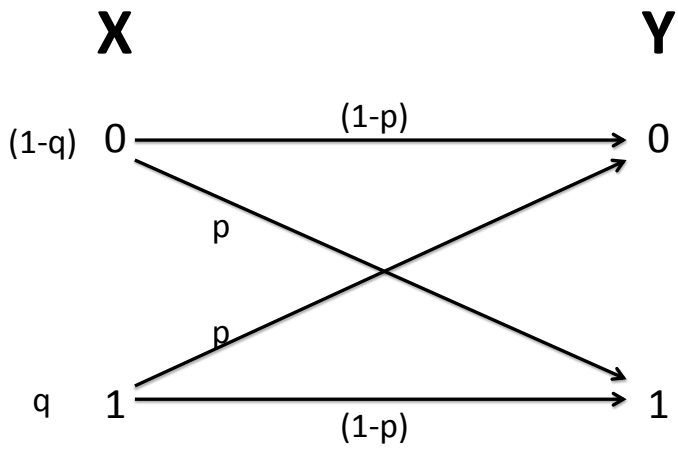
Aula 1- Códigos de Bloco

August 14, 2018

1 Introdução

Em muitas situações, a mensagem a ser transmitida por um sistema de comunicações é uma sequência de bits. Entretanto, devido a imperfeições do canal como a presença de ruído ou perda de fase, os sinais recebidos não são exatamente iguais aos transmitidos. Se a distorção for grande o suficiente, pode haver a troca do valor do bit na recepção, o que caracteriza um erro de recepção. A probabilidade de erro de bit pode depender da relação entre a potência do sinal e o ruído do canal, como será visto na teoria.

Se a probabilidade de troca independer do valor do bit transmitido, dizemos que o canal é simétrico (BSC, do inglês: *Binary Symmetric Channel*). Um modelo para este canal está na figura abaixo.



Neste modelo, a entrada X vale 0 ou 1 com probabilidade $1-q$ e q, respectivamente. A saída do canal é Y. A relação entre entrada e saída do canal é probabilística e é $P(Y=X)=1-p$ e $P(Y\neq X)=p$. É possível determinar a capacidade deste canal como sendo o máximo da informação mútua:

$$\mathcal{C} = \underset{q}{\text{argmax}}\{\mathcal{I}\{X,Y\}\} \tag{1}$$

A Teoria da Informação mostra que, se transmitirmos uma quantidade de bits de informação a uma taxa $\mathcal{R} < \mathcal{C}$, podemos ter uma probabilidade de erro de bit tão baixa quanto desejada. Mais do que isso, a Teoria da Informação diz que conseguiríamos transmitir até $C = 1 + plog_2[p] + (1-p)log_2[1-p]$ bits por uso deste canal, em média, com probabilidade de erro tão baixa quanto se queira. Para isto é necessário

que $q = 1/2$. O valor de $\mathcal{C}$ em função de p está no gráfico abaixo. Entretanto, a Teoria da Informação não diz como realizar a transmissão.

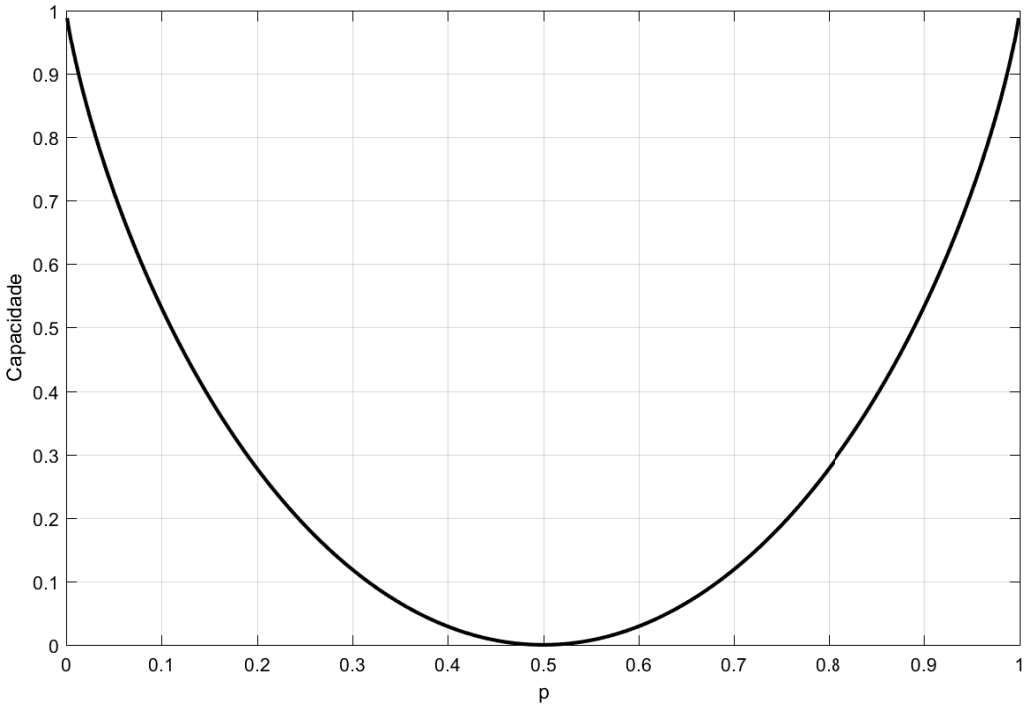


Figure 1: Capacidade do Canal BSC em função de p

Uma dúvida natural neste momento é: como transmitir uma quantidade de bits de informação menor do que 1? A resposta é a utilização de um código de canal, frequentemente categorizados em códigos de bloco (que operam sobre blocos de bits) e códigos convolucionais (que operam sobre sequências de bits). Vamos nos limitar aos códigos de bloco. As seguintes definições explicam os pontos mais relevantes para este laboratório:

- Um código de bloco binário é um conjunto de vetores binários, todos com tamanho N. Entretanto, nem todas as 2^N possibilidades de vetores binários são parte do código. Somente 2^K vetores binários são parte do código e $K \leq N$. Os vetores que fazem parte do código são chamados palavra-código.
- Logo, as palavras-código estão contidas no espaço binário com dimensão N
- Logo, é possível associar a cada palavra-código um outro vetor binário com tamanho K. Este vetor de tamanho K é chamado de palavra de informação.
- A taxa deste código é K/N, pois para cada K bits de informação, precisamos transmitir N bits de uma palavra código. A taxa não determina o tamanho da palavra código.
- A relação entre palavra de informação e palavra-código é feita através do codificador. Um mesmo código pode ter vários codificadores.
- Se $\mathbf{v}_1$ e $\mathbf{v}_2$ são palavras-código, definimos $\mathbf{v}_3 = \mathbf{v}_1 + \mathbf{v}_2$ como sendo o vetor obtido pela soma módulo 2 (XOR) dimensão a dimensão dos vetores originais.

- Um código é linear se, para qualquer par de palavras-código $\mathbf{v}_1$ e $\mathbf{v}_2$, um novo vetor $\mathbf{v}_3 = \mathbf{v}_1 + \mathbf{v}_2$ também é uma palavra código. Neste caso as palavras-código formam um sub-espaço do espaço de Hamming N dimensional.

- O peso de Hamming de um vetor é o número de 1's que ele tem. Por exemplo, o vetor [0101101011] tem peso de Hamming 6

- A distância de Hamming entre dois vetores é o número de posições em que eles diferem. A distância de Hamming entre dois vetores também pode ser vista como o peso de Hamming da soma módulo-2 dos dois vetores.

- A palavra recebida pode ser diferente da palavra-código transmitida se houver erros de transmissão. O melhor que podemos fazer* é encontrar a palavra-código mais próxima da recebida (critério MV), ou seja, a palavra-código que tem a menor distância de Hamming da palavra recebida.

- Estes códigos funcionam pela adição de redundância: para transmitir K bits de informação, utilizamos $N > K$ bits da palavra-código.

- Há um custo na utilização de um código. Se precisamos transmitir N bits da palavra-código para transmitir K bits de informação, a energia média por bit de informação é aumentada: $\varepsilon_b = N\varepsilon_t/K$. Um sistema não codificado em simplesmente $\varepsilon_b = \varepsilon_t$. Uma comparação justa entre o sistema codificado e o sistema não codificado exigiria que os valores de ε_b fossem iguais.

A forma mais simples de inserção de redundância é a repetição de um único bit de informação N vezes, chamada código de repetição. Há duas palavras código neste sistema: a sequencia de N zeros e a sequencia de N 1's. A distância de Hamming entre estas duas palavras é de N. Há $K = 1$ bits de informação para N bits transmitidos, a taxa do sistema é de $K/N = 1/N$ bits por uso do canal, pois são necessários transmitir N bits através do canal para enviarmos um único bit de informação. Na recepção, o processo de decisão mais simples consiste em, dado N bits recebidos, escolher pelo bit que estiver mais presente na palavra código. Assim, se menos da metade dos bits tiverem sofrido erros, o bit transmitido pode ser corretamente recebido. Por este motivo, é útil que N seja ímpar.

Se, em vez disso, transmitirmos um único bit que corresponde à soma módulo-2 de todos os K bits de informação, a taxa do sistema será de $K/(K+1)$. Independentemente do valor de K, todas as palavras-código possíveis tem um número par de 1's. O bit de redundância equivale então a um bit de verificação de paridade. Como todas as palavras são pares, a menor distância entre duas palavras-código é de 2 bits. Se recebermos uma palavra com um número de bits ímpar, sabemos que houve um erro. Não será possível, neste esquema, identificar onde foi o erro, mas pode-se usar da identificação da existência do erro para solicitar a retransmissão da mensagem.

Uma forma mais elaborada de correção de erros é através de códigos de Hamming. Neste sistema, a palavra-código é obtida através de um mapeamento de 4 bits de informação numa palavra código com 7 bits, como mostra a figura 2. Os bits de informação são b_1, b_2, b_3 e b_4. Os de paridade p_1, p_2 e p_3 são definidos de tal forma que a soma binária dos bits dentro de cada um dos círculos seja igual a 0. Como há 2^4 possíveis combinações dos bits de informação, há 2^4 palavras-código. A distância de Hamming entre elas é de 3 bits. O código de Hamming é capaz de identificar até 2 erros e de corrigir até 1, utilizando o mesmo diagrama da figura 2.

Há outras formas de se visualizar o mesmo código de Hamming. Uma delas é através da matriz geradora $\mathbf{G}$. Esta matriz tem dimensão $K \times N$. Qualquer palavra código pode ser obtida através do produto $\mathbf{u} \cdot \mathbf{G} = \mathbf{v}$, onde $\mathbf{u}$ é um vetor binário com dimensão K representando os bits de informação. Continuando o exemplo do código de Hamming, a matriz geradora do exemplo seria:

*Quando a probabilidade de erro de bit transmitido é menor do que 0.5

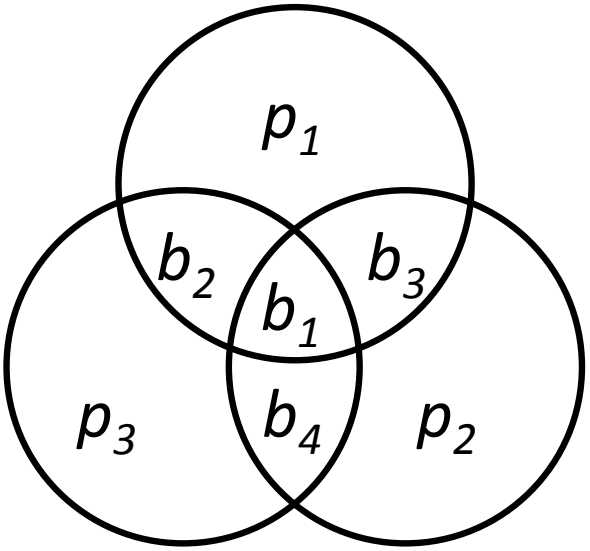


Figure 2: Código de Hamming

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{2}$$

Esta condição é semelhante a definir um hiperplano (subespaço) como todos os pontos que podem ser gerados pela combinação linear de um conjunto de vetores (linha) LI. Há várias manipulações que podem ser feitas nesta matriz mas que mantém o código. Entretanto, estas manipulações modificam a relação entre $\mathbf{u}$ e $\mathbf{v}$.

Outra forma é definir o código como sendo todas as palavras-código que satisfazem $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$, ou seja, a projeção da palavra-código nos vetores (coluna) que formam a matriz H é zero. Esta condição é semelhante à condição que define um plano como sendo ortogonal a um vetor. A matriz $\mathbf{H}$ é chamada de matriz de verificação de paridade. Entretanto, esta matriz não estabelece a relação entre $\mathbf{u}$ e $\mathbf{v}$. Assim como para a matriz geradora, há várias matrizes $\mathbf{H}$ que definem o mesmo código. Elaborando um pouco chegamos em:

$$\begin{gathered} \mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}, \forall \mathbf{v} \in C \\ \mathbf{u} \cdot \mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}, \forall \mathbf{u} \\ \rightarrow \mathbf{G} \cdot \mathbf{H}^T = \mathbf{0} \end{gathered} \tag{3}$$

Quando $\mathbf{G}$ pode ser escrito como $\mathbf{G} = [\mathbf{I} \quad \mathbf{P}]_{K \times N}$, podemos obter $\mathbf{H}$ facilmente como $\mathbf{H} = [-\mathbf{P}^T \quad \mathbf{I}]_{(N-K) \times N}$. No nosso exemplo teríamos:

$$\mathbf{H}^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

A matriz de verificação de paridade pode ser usada para realizar a decodificação dos vetores recebidos da seguinte forma:

1. Transmitimos a palavra-código $\mathbf{v}$
2. Há erros de transmissão de forma que recebemos $\mathbf{r} = \mathbf{v} + \mathbf{e}$, onde a soma é modulo 2 e $\mathbf{e}$ é um vetor binário que vale 1 onde há erros de transmissão.
3. Ao testar a palavra recebida, obtemos $\mathbf{s} = \mathbf{r} \cdot \mathbf{H} = (\mathbf{v} + \mathbf{e})\mathbf{H}^T = \mathbf{v} \cdot \mathbf{H}^T + \mathbf{e} \cdot \mathbf{H}^T = \mathbf{0} + \mathbf{e} \cdot \mathbf{H}^T$. Como $\mathbf{H}^T$ tem dimensão $(K) \times (N - K)$, há $(N - K)$ vetores $\mathbf{s}$ distintos. Os valores de $\mathbf{s}$ são chamados de síndrome.
4. Associamos para cada vetor $\mathbf{s}$ um padrão de erro $\mathbf{e}'$. Há vários valores de $\mathbf{e}$ que geram o mesmo $\mathbf{s}$. Entre todos os possíveis, selecionamos previamente o valor de $\mathbf{e}'$ com o menor peso de Hamming, pois este é o mais provável.
5. Tentamos corrigir os erros de transmissão decidindo que a palavra-código transmitida foi $\mathbf{r} + \mathbf{e}'$.
6. Caso $\mathbf{e} = \mathbf{e}'$, decidiríamos corretamente por $\mathbf{r} + \mathbf{e}' = \mathbf{v} + \mathbf{e} + \mathbf{e}' = \mathbf{v} + \mathbf{e} + \mathbf{e} = \mathbf{v}$, pois $\mathbf{e} + \mathbf{e} = \mathbf{0}$.
7. Caso $\mathbf{e} \neq \mathbf{e}'$, decidiríamos erroneamente por $\mathbf{r} + \mathbf{e}' \neq \mathbf{v}$. Neste caso teríamos erros de transmissão de informação.
8. Extraímos os bits de informação da palavra-código $\mathbf{r} + \mathbf{e}'$

No nosso exemplo, a síndrome obtida quando temos um erro de transmissão no terceiro bit pode ser obtida pelo produto $[0010000]\mathbf{H}^T = [110]$. Assim, sempre que encontrarmos esta síndrome, assumiríamos que houve um erro de transmissão no terceiro bit da palavra recebida. As vezes o sistema irá corrigir erros de transmissão dos bits de informação e as vezes haverá erros. De qualquer forma espera-se que a probabilidade de erro **dos bits de informação** seja menor do que p

Podemos ainda ter uma terceira representação através de um grafo que representa a condição matemática $\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}$, como indicado abaixo. Os círculos são bits e os quadrados são equações de paridade: a soma de todos os bits que participam daquela equação deve valor 0 (modulo 2). Esta estrutura é muito utilizada em métodos de decodificação de canal contemporâneos que utilizam propagação de probabilidades. Há métodos em inteligência artificial semelhantes.

2 Atividade

O objetivo do laboratório de hoje é implementar cinco componentes:

1. Um codificador de canal para o código de Hamming como descrito.
2. Um canal BSC com parâmetro p
3. Um decodificador de canal para o codificador do item 1
4. Um código e codificador de canal criado pelos próprios alunos, com taxa semelhante ao do código de Hamming, mas com palavras-código de tamanho maior. Não é suficiente simplesmente repetir o código de Hamming, ou algo semelhante.
5. Um decodificador para o codificador do item 4

Após a implementação destes componentes, o objetivo é obter as curvas de probabilidade de erro de bit de informação em função do parâmetro p para o código de Hamming e para o código criado pelos alunos. O processo para se estimar a probabilidade de erro é a seguinte:

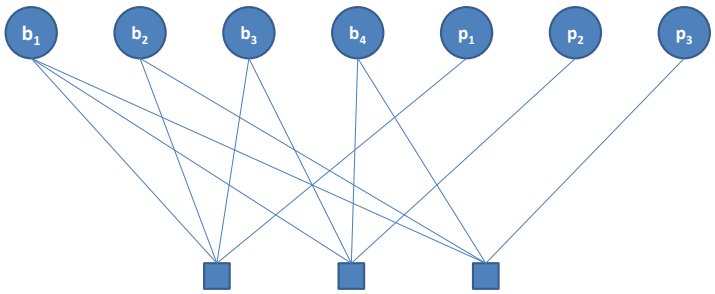


Figure 3: Grafo correspondente ao código de Hamming

1. Escolha um valor de p. Utilize $p = 0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, ...$
2. Gere aleatoriamente cerca de 1 milhão de bits de informação
3. Divida em grupos de K bits. Haverá L grupos
4. Gera as L palavras-código correspondentes a cada um dos grupos do item 3
5. Simule o efeito do canal BSC trocando o valor dos bits das palavras código do item anterior com probabilidade p para todos os bits de todas as L palavras-código, gerando assim L palavras recebidas.
6. Realize o processo de detecção por síndrome para cada uma das L palavras recebidas.
7. Realize o processo de estimação sobre os bits de informação.
8. Compare as estimativas do item anterior com os bits gerados no item 2. O % de diferenças é a probabilidade de erro de bit de informação.
9. Retorne ao passo 1 até que todos os valores de p sejam processados. Limite-se a valores de $p > 10^{-6}$

3 Perguntas a serem respondidas no relatório

O relatório deve conter, além do que os alunos consideram necessário, as resposta das seguintes perguntas:

1. Qual foi a maior dificuldade de implementar o decodificador para o código de Hamming?
2. Qual foi o método utilizado para encontrar o código maior? Este método é extensível para qualquer tamanho de bloco?
3. Qual é a relação medida entre o tamanho do bloco e o desempenho?
4. Qual é a complexidade de codificação e decodificação do seu sistema?

4 Referências

- https://en.wikipedia.org/wiki/Forward_error_correction
- https://en.wikipedia.org/wiki/Block_code
- https://en.wikipedia.org/wiki/Hamming_code
- Digital Communications 5th edition, Proakis&Salehi, capítulo 7