ELE-32 Introdução a Comunicações

Aula 3- Códigos LDPC

May 25, 2023

1 Uma nova visão

Em vez de realizar a decodificação por síndrome, é possível visualizar um código como sendo um conjunto de restrições que devem ser simultaneamente satisfeitas. As restrições são as relações de paridade entre subconjuntos de bits, representados por exemplo no código de Hamming do primeiro laboratório como cada um dos três círculos. Podemos representar as restrições como um GRAFO bipartido com dois tipos de nós (nodes): um que representa as variáveis (v-node) e um que representa as restrições (c-node). Os nodes são ligados através de ramos que conectam um v-node a um c-node se o primeiro faz parte da restrição estabelecida pelo segundo.. O grafo pode ser organizado em duas camadas: VND(Variable Node Decoder), que contém todos os v-nodes, e CND (Check Node Decoder) que contém todos os c-nodes. Esse grafo pode ser obtido através da matriz de verificação de paridade, e vice versa.

É possível visualizar o grafo como uma democracia: os nodes são indivíduos e o grafo deve chegar a um consenso sobre qual é a palavra código dada uma palavra recebida, mesmo que alguns dos indivíduos discordem do resultado. Para isso, os nodes devem trocar mensagens relatando sua opinião local sobre a palavra código correta com base nas suas observações locais, e com o diálogo entre as partes pode-se atingir um consenso. Um node recebe e transmite mensagens de e para todos os ramos, e a mensagem transmitida por um ramo depende somente das mensagens que são recebidas nos outros ramos.

A troca de mensagens tem uma complexidade que depende do número de ramos conectado a cada ramo. Para manter a complexidade baixa, utilizamos matrizes de verificação de paridade esparsas, isto é, que tem baixa quantidade de valores diferentes de zero, resultando em nodes com baixo número de ramos. Um código definido por uma matriz de verificação de paridade esparsa é chamado de LDPC: Low Density Parity Check (code). Estes códigos são muito poderosos e são por exemplo empregados em redes 5G. Com esta visão, é suficiente construir um grafo para projetar o código LDPC.

2 Projeto e construção

O grafo contém alguns parâmetros:

- número de v-nodes: N;
- número de ramos que saem de cada v-node para a camada CND: d_v;
- número de ramos que saem de cada c-node para a camada VND: d_c;
- consequentemente, o número de c-nodes é fixo, vale M e é uma função dos números anteriores.

A taxa do código é R e pode ser determinada respondendo a seguinte pergunta: se um espaço tem dimensão N mas há $M \leq N$ restrições, quanta liberdade tenho para escolher os vetores que fazem estão inclusos no subespaço gerado por estas M restrições? Demonstra-se que a taxa depende somente de d_v e

d_c. Assim, podemos escolher d_v, d_c para definir a taxa e posteriormente escolher um valor apropriado de N para gerar um grafo que gera um código com comprimento que quisermos (dadas algumas restrições). Isso permite gerar códigos com taxas idênticas e comprimentos diferentes.

O projeto envolve a escolha correta de d_v e d_c. Pode-se ter códigos regulares ou irregulares. Para códigos regulares, d_v e d_c são constantes. Para códigos irregulares, estes valores podem variar de node para node. A escolha de d_v e d_c é feita para minimizar a probabilidade de erro. Neste laboratório, você deve escolher d_v e d_c para que a taxa do código final seja idêntica à taxa do código de Hamming do primeiro laboratório e ao mesmo tempo para que a complexidade seja a menor possível. A determinação dos valores de d_v e d_c limita os valores de N e M que podem ser escolhidos pois o número de ramos que sai da camada CND é o mesmo que entre na camada VND, e vice versa, exceto pelos ramos que vem do canal e estão conectados à camada VND

O problema é como construir efetivamente o grafo. Mesmo sabendo todos os parâmetros, é necessário conectar as camadas VND e CND. Alguns fatos são sabidos para termos um bom desempenho: não podemos ter mais de um ramo conectado um mesmo v-node com um mesmo c-node; gostaríamos que, na presença de loops dentro do grafo, eles fossem os maiores possíveis. Uma forma de construir o grafo é utilizar o algoritmo Progresive Edge Growth*, que para valores de N pequenos (menor do que 1000), oferece resultados um pouco melhores do que conectar os nodes aleatoriamente. Para os propósitos deste laboratório, é suficiente utilizar a construção aleatória, desde que os parâmetros d_v e d_c sejam respeitados para todos os nodes e que não haja mais de um ramo conectado dois nodes quaisquer.

Por último (ou primeiramente), na prática necessitamos codificar os bits de informação. É possível obter a matriz geradora, mas a complexidade do algoritmo para faze-lo é maior do que a desejada para o laboratório e desnecessária para a obtenção dos resultados. †

3 Decodificação e decisão

Há vários algoritmos que podem ser utilizados para realizar o processo de decodificação. Um algoritmo simples é o chamado bit-flipping, cujo nome é evidente a partir da descrição do mesmo:

1. Inicie o vetor $\mathbf{y}$ com os valores recebidos do canal
2. Teste todas as equações de paridade definidas pelos c-nodes
3. Conte, para cada bit de $\mathbf{y}$, o número de c-nodes com o quais esta conectado mas não satisfeitos, isto é, o somatório é diferente de zero
4. Troque o valor dos bits com o maior número de equações insatisfeitas
5. Repita os passos 2 a 4 até que:
   - todos os c-nodes tem a sua restrição satisfeitas ou
   - um número máximo de iterações seja atingido.

O número efetivo de iterações depende de quantos erros foram introduzidos pelo canal. Após a parada do processo iterativo, deve-se decidir sobre a palavra código final. No caso do algoritmo bit-flipping, a decisão é o valor final de $\mathbf{y}$.

*Vide "*Regular and Irregular Progressive Edge-Growth Tanner Graphs*" de 2005 se desejar

†Para os curiosos, veja o artigo "*Efficient encoding of low-density parity-check codes*", de 2001, que permite a codificação em tempo quase linear de uma palavra código de um código LDPC.

4 Atividades

1. Gere um programa que seja capaz de projetar a matriz de verificação de paridade para um código LDPC regular para valores arbitrários de d_v, d_c e N.

2. Utilize o programa do item acima para projetar matrizes de verificação de paridade com taxa idêntica ao código de Hamming mas com comprimentos de aproximadamente 100, 200 e 500 bits. Use o valor de N mais próximo destes valores

3. Implemente um decodificador que, com base na matriz de verificação de paridade, seja capaz de realizar o processo iterativo conforme o algoritmo bit-flipping

4. Estime a probabilidade de erro de bit de informação para os 3 sistemas encontrados considerando que a palavra código é transmitida através de um canal BSC com parâmetro $p = 0.1, 0.05, 0.002, 0.001, ...0.00001$. Dica: a probabilidade de erro de bit é uniforme para todos os bits da palavra código.

5 Pontos a serem investigados durante a realização do laboratório

1. Qual é a relação entre desempenho e probabilidade de erro de bit de informação?

2. Compare o desempenho do sistema LDPC com os sistemas dos outros laboratórios.

3. Qual é a complexidade de decodificação? Ela depende do valor de p?

4. Quais foram as dificuldades de se projetar o código LDPC?

6 Bibliografia recomendada

- J. G. Proakis, M. Salehi, "Digital Communications", 5a. edição, seção 8.11