Exercise 7.1 More properties of linear codes (all)

Show the following properties of a (binary) linear $[n, k]$-code C.

a.) The minimal distance $d(C)$ is the minimal Hamming weight of all (non-zero) codewords.

Solution: For a binary linear $[n, k]$-code C, for any $c, c' \in C$, we have $x = c \oplus c' \in C$. Minimal distance:

$$\begin{aligned} d(C) &= \min_{c,c' \in C, c \neq c'} \delta(c, c') = \min_{c,c' \in C, c \neq c'} |\{i : c_i \neq c'_i\}| \\ &= \min_{c,c' \in C, c \neq c'} |\{i : c_i \oplus c'_i = 1\}| = \min_{x \in C, x \text{ non-zero}} |\{i : x_i \neq 0\}|. \end{aligned}$$

Thus, the minimal distance is the minimal Hamming weight of all non-zero codewords.

b.) If H is the parity check matrix of C, then $d(C)$ equals the number of columns of H that are linearly dependent.

Solution: Let $H = [h_1^T, \ldots, h_n^T] \in \{0,1\}^{(n-k)\times n}$, where $h_i^T \in \{0,1\}^{(n-k)\times 1}$. Let $c = [c_1, \ldots, c_n]^T$ be the codeword with minimal Hamming weight. Since H is the parity check matrix of C, then

$$0 = Hc = \sum_{i=1}^{n} h_i^T c_i = \sum_{i:c_i \neq 0}^{n} h_i^T = 0.$$

Thus, $d(C)$ is the number of columns of H that are linearly dependent.

c.) Prove that (after permuting the coordinates if necessary) C has a generator matrix of the form $G = [I_k \ G']^T$ where I_k is the $k \times k$ identity matrix, and where G' is some $k \times (n-k)$ matrix.

Solution: Since k is the minimal number of codewords needed for a basis and k columns of generator matrix $G \in \{0,1\}^{n\times k}$ span C, then G must contain a $k \times k$ identity matrix I_k. Then by permuting the columns of G, we can always write the generating matrix as

$$G = \begin{bmatrix} I_k \\ G' \end{bmatrix}.$$

Exercise 7.2 Modified linear codes (all)

Some of the following operations on rows or columns of the generator matrix G or the parity-check matrix H may decrease the minimum distance of a linear block code? Which of the operations below can cause a reduction in the minimum weight? **Note: Here G is a $n \times k$ matrix.**

a.) Exchanging two rows of G.

Solution: No.

b.) Exchanging two rows of H.

Solution: No.

c.) Exchanging two columns of G.

Solution: No.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Figure 1: A typical Sudoku and its solution (from Wikipedia)

d.) Exchanging two columns of H.

Solution: No.

e.) Deleting a row of G.

Solution: Yes. Since the length n of the codeword is shortened and the minimum weight $d \leq n - k+1$.

f.) Deleting a row of H.

Solution: Yes, n is unchanged but $n - k$ is reduced and thus k increases. This may cause a reduction in d.

g.) Deleting a column of G and the corresponding column of H.

Solution: Not valid. After deleting a column of G and H, they cannot yield a valid code.

h.) Adding a column to G and a corresponding column to H.

Solution: Adding a column to G means will make G become a $n \times (k+1)$ matrix. Adding a corresponding column to H will make H become a $(n-k) \times (n+1)$ matrix. This cannot yield a valid code.

i.) Adding one column of H to another column of H.

Solution: Yes. From Problem (b) in Exercise 6.2, for example, we have a code such that $d(C) = 3$, $h_1^T + h_2^T + h_3^T = \mathbf{0}$ and h_2^T,h_3^T are linearly independent. If we add h_3^T to h_2^T and denote the new column 2 as $\tilde{h}_2^T$, then $\tilde{h}_2^T - h_3^T = 0$ and the new code C' may have minimum distance $d(C')$ less or equal to 2.

Exercise 7.3 Sudoku and the belief propagation algorithm (all)

Sudoku is a classical mathematical puzzle in which a player is asked to fill in missing numbers in a 9×9 array where each of the nine rows, nine columns, and nine 3×3 sub-arrays consists of numbers $\{1, \ldots, 9\}$. An example is given in Figure 1.

a.) Denote the configuration of a Sudoku by $\{x_{i,j}\}_{i,j}$ where, for each $(i,j) \in \{1, \ldots, 9\}^2$, $x_{i,j} \in \{1, \ldots, 9\}$ is the number at the (i,j)-th location. Define the function $g : \{1, \ldots, 9\}^{81} \to \{0, 1\}$ as

$$g(x_{i,j} : i,j \in \{1, \ldots, 9\}) = \begin{cases} 1 & \text{if } \{x_{i,j}\}_{i,j} \text{ composes a valid Sudoku} \\ 0 & \text{otherwise} \end{cases}.$$

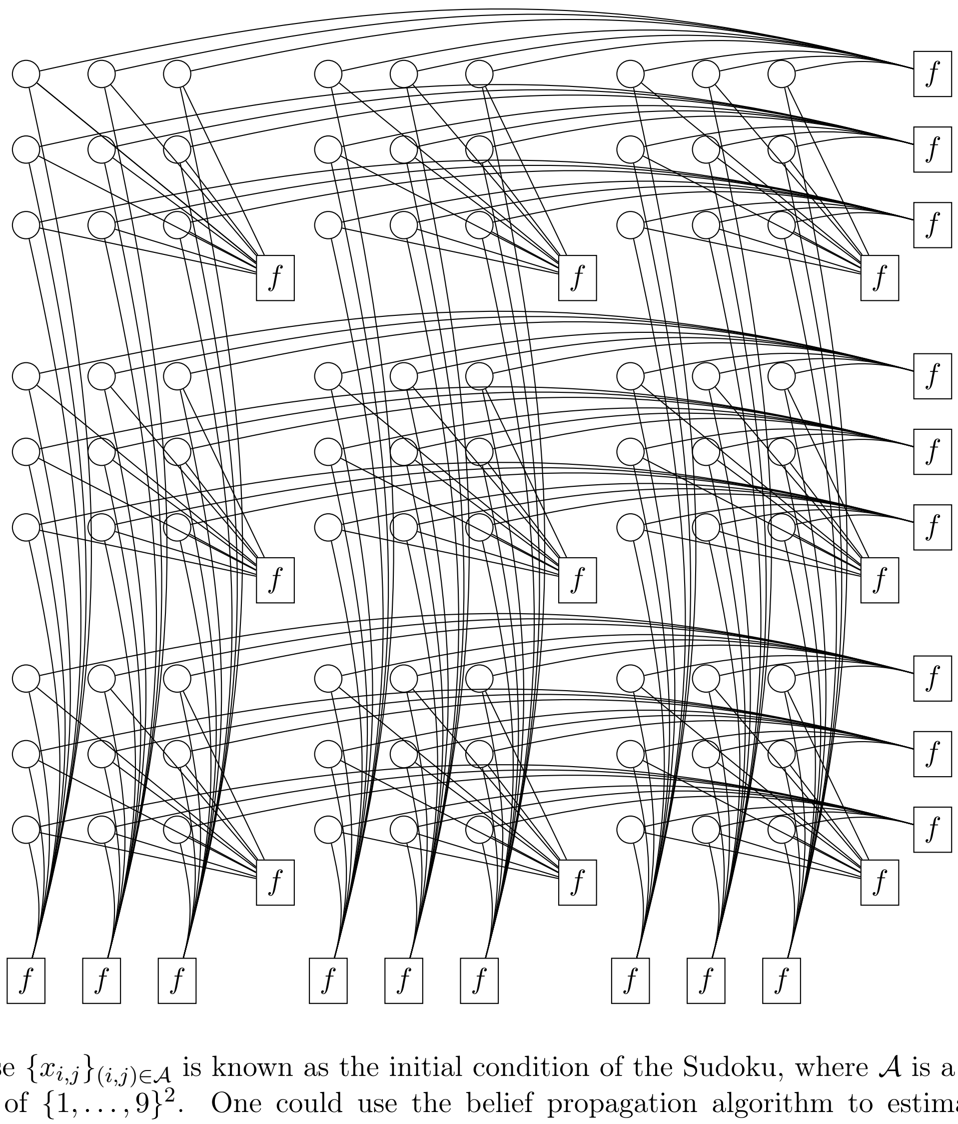
Represent g as a factorization of f over different arguments, where $f : (x_1, \dots, x_9) \mapsto 1\{\{x_1, \dots, x_9\} = \{1, \dots, 9\}\}$.

Solution:

$$g(\boldsymbol{x}) = \prod_{a=1}^{9} f(x_{a,j} : j \in \{1, \dots, 9\}) \cdot \prod_{b=1}^{9} f(x_{i,b} : i \in \{1, \dots, 9\})$$
$$\cdot \prod_{c_1=0}^{2} \prod_{c_2=0}^{2} f(x_{i,j} : i \in \{3c_1 + 1, \dots, 2c_1 + 3\}, j \in \{3c_2 + 1, \dots, 3c_2 + 3\}).$$

b.) Draw the factor graph corresponding to the factorization in a.).

Solution:



c.) Suppose $\{x_{i,j}\}_{(i,j)\in\mathcal{A}}$ is known as the initial condition of the Sudoku, where $\mathcal{A}$ is a proper subset of $\{1, \dots, 9\}^2$. One could use the belief propagation algorithm to estimate the remaining positions. To do so, a partially finished MATLAB program has been provided. Please fill in the gaps in the program and run the program to solve the embedded Sudoku.

Solution: Please refer to the file ‘Sudoku_sol.m’.