Robust Slepian-Wolf Coding Using Low-Density Graph Codes

Master of Applied Science (2010) McMaster University
Electrical and Computer Engineering Hamilton, Ontario

TITLE: **Robust Slepian-Wolf coding Using Low-Density Graph Codes**

AUTHOR: Min Huang
Bachelor of Engineering
McMaster University, Canada, 2008

SUPERVISOR: Dr. Jun Chen

CO-SUPERVISOR: Dr. Kon Max Wong

NUMBER OF PAGES: xii, 57

Dedications:

To my parents:

Chongrong Huang and Mingying Chen

Abstract

In this thesis, Robust Slepian-Wolf coding problem is discussed. Two correlated source sequences X^n and Y^n are encoded at separate encoders and decoded together. When the encoder of source Y is broken, another sequence X^n still can be decoded to achieve a nontrivial distortion. Further, X^n can be recovered losslessly once that broken encoder is restored. A practical coding scheme is developed using low density graph codes. Moreover, by generalizing the coding scheme of Robust Slepian-Wolf coding problem, two approaches are proposed for the Wyner-Ziv problem using the low density graph codes.

Acknowledgements

Firstly, I would like to thank my supervisors, Dr. Jun Chen and Dr. Kon Max Wong, for their guidance and encouragement. I deeply appreciate their patience, dedication and expertise, offered to me over the course of my independent research and the writing of this thesis.

Secondly, I would also like to thank my group members, Lin Song and Ying Zhang for their support, suggestions, and insightful discussions over the course of our friendship and study, and Cheryl, the graduate administrative assistant, for her help during my stay at McMaster University.

Last, but not least, I would like to extend my gratitude to my family members home in China, for their love, and for their words of kind encouragement, sent to me consistently regardless of the distance between us.

Contents

List of Figures

Acronyms

LDPC	Low Density Parity Check
LDGM	Low Density Generator Matrix
LLR	Log Likelihood Ratio
RSW	Robust Slepian-Wolf problem
BER	Bit Error Rate
KKT	Karush-Kuhn-Tucker

Notations

$\mathcal{C}$	Code ensemble
$\mathcal{C}_S$	Cosets
$\mathbf{H}$	Parity check matrix
$\mathbf{G}$	Generator matrix
R	Rate
P_{XY}	Joint distribution of (X,Y)
P_X	Distribution of X
$H(\cdot)$	Entropy of a random variable
$I(\cdot,\cdot)$	Mutual information between two random variables
$d(\cdot,\cdot)$	Distortion between two variables
$P_r[\cdot]$	Probability of an event
$\mathcal{A}$	Set $\mathcal{A}$
P_e	Probability of error
log	Logarithm function with base e
$\rho(x)$	Check node degree polynomial
$\lambda(x)$	Variable node degree polynomial
$X \leftrightarrow Y \leftrightarrow Z$	Markov chain

C_i	i-th check node
N_i	i-th network node
S_i	i-th source node
V_i	i-th variable node
$A_c(i)$	The set of check nodes connected to variable node i
$B_v(j)$	The set of variable nodes connected to check node j
$M_{V_i \to C_j}$	The messages from variable node i to check node j
$M_{C_j \to V_i}$	The message from check node j to variable node i
$M_{S_i \to N_i}$	The message from source node i to network node i
$M_{N_j \to C_i}$	The message from network node j to the check node i
$M_F(i)$	Marginal probability
XOR	Exclusive or operation

Chapter 1

Introduction

1.1 Backgroud

In this section, some brief results of lossless and lossy source coding are reviewed for a single source and two correlated sources.

1.1.1 Lossless Source Coding

1.1.1.1 Single Source

For a single information source, a well-known result shows the encoding rate R must be greater than or equal to the entropy of the source for lossless reproduction. Let X be a discrete random variable selecting the values from the set $\mathcal{A} = \{1, 2, \ldots, A\}$ with the probability distribution by $p(x) = Pr\{X = x\}, x \in \mathcal{A}$. Then a sequence of $X_1, X_2, \ldots, X_n$ is formed through i.i.d drawn from the probability distribution of X by $p(x)$ and the probability of this sequence $p(X_1, X_2, \ldots, X_n)$ is calculated by:

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i) \quad X_i \in \mathcal{A}, i = 1, 2, \ldots, n.$$

Given the definition of strongly typical sequence [1], the probability of a typical sequence is:

$$\begin{aligned} p(X_1, X_2, \ldots, X_n) &= \prod_{i=1}^{n} p(X_i) && (1.1)\\ &= \prod_{a \in A} p(a)^{n(a|\mathrm{X})} && (1.2)\\ &\approx \prod_{a \in A} p(a)^{np(a)} && (1.3)\\ &= 2^{n \sum_a p(a) \log p(a)} && (1.4)\\ &= 2^{-nH(X)} && (1.5) \end{aligned}$$

In the equation (1.2), $n(a|X^n)$ denote the number of a in the sequence X^n. The approximation step (1.3) follows from the fact that X^n is a typical sequence. The result (1.5) shows all elements of the typical set are nearly equiprobable. By the weak law of large numbers, the probability of typical set is nearly 1. So the size of the typical set is approximately $2^{nH(X)}$. If $R \geq H(X)$, a one-to-one mapping relationship is built between each typical sequence X^n generated by the source and a codeword included in the codebook with 2^{nR} codewords, which guarantees the lossless recovery of compressed source.

1.1.1.2 Slepian-Wolf Coding

The result from last part can be extended to show that compressing two independent sources X and Y requires the encoding rate $R = R_x + R_y \geq H(X) + H(Y)$ for lossless recovery. However, what is the encoding rate for two correlated sources? Slepian-Wolf coding discusses such problem: two correlated sources (X, Y) with the joint probability $p(x, y)$ are encoded separately at two encoders and decoded together at one joint decoder, and its admissible rate region R is illustrated in Figure 1.1.

As shown in figure 1.1, the two-dimensional rates (R_X, R_Y) must satisfy the following

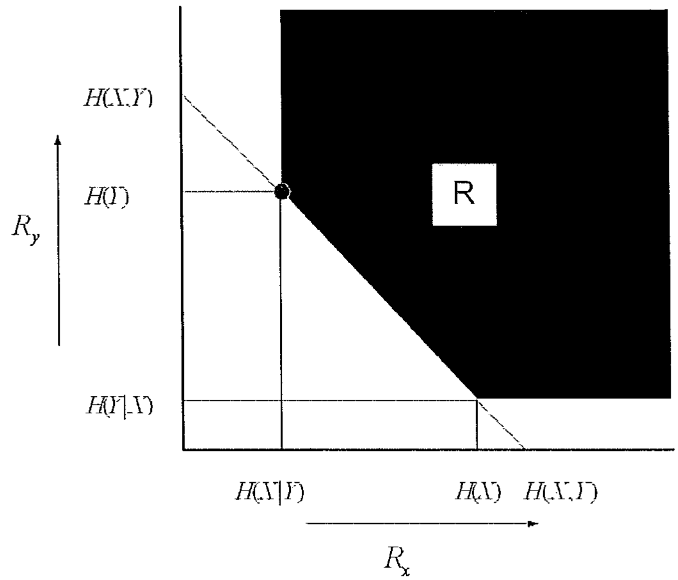

Figure 1.1: Achievable rate region of Slepian-Wolf coding

three inequalities [2]:

$$\begin{aligned} R_X &> H(X|Y) \\ R_Y &> H(Y|X) \\ R_X + R_Y &> H(X,Y) \end{aligned}$$

It is obvious that the sum of two encoding rates could be less than the sum of their entropies for lossless recovery from the three inequalities above. Let's take a closer look at the admissible rate region, two corner points $(H(X|Y), H(Y))$ and $(H(X), H(Y|X))$, which are symmetrical through replacing source X and Y, are of significant importance and the dominant face between them is achievable through the time sharing scheme. So only one corner point $(H(X|Y), H(Y))$ is analyzed and corresponding coding scheme is illustrated in Figure 1.2.

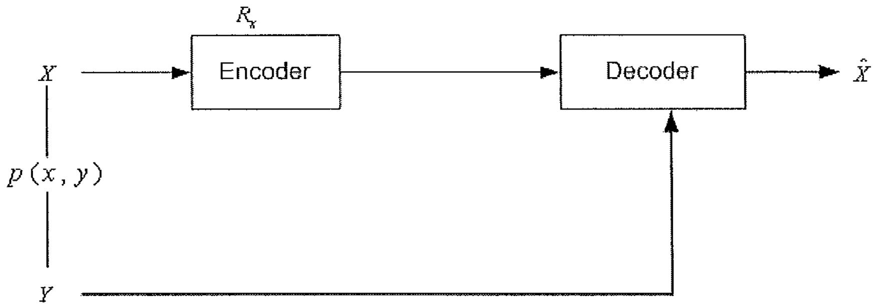

Figure 1.2: Coding Scheme of Slepian-Wolf coding problem

Firstly, the codebook is generated by using random bins. All the n-length typical sequences generated by source X are uniformly dropped into 2^{nR_x} bins, which composes the codebook. Two sequences X^n and Y^n of length n are generated from two correlated sources (X, Y) with the joint distribution $p(x, y)$. In the process of encoding, the sequence X^n is encoded with R_X to produce an index pointing to which bin the sequence X^n is in. In the process of decoding, the sequence Y^n is directly transmitted to the decoder as the side information for decoding the sequence X^n. At the decoder the side information sequence Y^n compared with those sequences in the bin specified by the index from the X encoder. If one of sequences in that bin and the side information Y^n are jointly typical sequences, then that sequence is declared as the decoded sequence $\hat{X}^n$. If the encoded rate R_X is greater than $H(X|Y)$, the decoder will recover the lossless sequence X^n with high probability. The details about the implementation of Slepian-Wolf coding with LDPC will be discussed in the chapter 3.

1.1.2 Lossy Source Coding

1.1.2.1 Single Source

Let's us discuss the simplest case first. A sequence of $X_1, X_2, \ldots, X_n$ is generated in one source through i.i.d drawn from the alphabet $\mathcal{A}$ with the probability distribution of $p(x)$. By using an encoding function $f_n : X^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$, the encoder output is an index that represents a codeword included in the code book, whose size is 2^{nR}. By using the decoding function $g_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \hat{X}^n$, the decoder output is a corresponding codeword mapped by the index from the encoder output. The distortion between sequences X^n and $\hat{X}^n$ is defined by

$$d(X^n, \hat{X}^n) = \frac{1}{n}\sum_{i=1}^{n} d(x_i, \hat{x}_i) \tag{1.6}$$

where $d(x, \hat{x})$ is a measure of the distortion of representing the symbol x by the symbol $\hat{x}$. The minimum achievable rate at distortion D is given by the classical rate-distortion theory:

$$R(D) = \min_{p(\hat{x}|x)} I(X; \hat{X}) \tag{1.7}$$

$$\text{subject to} \quad \sum_{(x,\hat{x})} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D \tag{1.8}$$

Among all conditional distributions $p(\hat{x}|x)$, a $p(\hat{x}|x)$ is found to minimize $I(X; \hat{X})$ and satisfy the expected distortion constraint. After finding the optimum $p(\hat{x}|x)$ from equation (1.7), the codebook is generated through the marginal distribution $p(\hat{x})$. Due to the fact that the sequence of $\hat{X}^n$ is i.i.d.generated, each codeword $\hat{X}^n$ can be produced with the distribution $p(\hat{x}^n) = \prod_{i=1}^{n} p(\hat{x}_i)$. To form the whole codebook simply, it is considered as a $2^{nR} \times n$ matrix and each entry is i.i.d drawn from the alphabet $\mathcal{A}$ with the probability distribution of $p(\hat{x})$. In the process of encoding, the sequence X^n is compared with each codeword in the codebook until finding one codeword which is jointly typical with X^n. Then the index of those codeword is recorded as the output of encoding. If no such codeword exists, the output of encoding is set to the index of the first codeword in the codebook. In the process of decoding, the sequence $\hat{X}^n$ is the codeword specified by that index.

1.1.2.2 Wyner-Ziv Coding

Wyner-Ziv coding is about the rate distortion coding with side information. Two sequences X^n and Y^n of length n are generated from two correlated sources (X, Y) with the joint distribution $p(x, y)$. The sequence X^n is encoded with the minimum rate $R_{WZ}(D)$ and decoded with the aid of the side information Y^n, which is available to the decoder, to achieve distortion D. The same distortion function (1.6) is applied. The rate distortion function with side information is [3]:

$$R_{WZ}(D) = \inf_{p(w|x)} [I(X;W) - I(Y;W)] \tag{1.9}$$

$$\text{Subject to } E[D(X,\hat{X})] \leq D, \quad \text{where } \hat{X} = f(Y,W) \tag{1.10}$$

where random variables X, Y, and W form a markov chain $Y \leftrightarrow X \leftrightarrow W$. The minimal rate $R_{WZ}(D)$ is found over conditional distribution $p(w|x)$ and functions f, so that the expected distortion is less than D. Then the codebook could be generated by the marginal distribution $p(w)$. Firstly, the whole codebook is considered as a $2^{nR_1} \times n$ matrix, where $R_1 = I(X;W)$. Each entry in the matrix is i.i.d drawn from the alphabet $\mathcal{W}$ with the probability distribution of $p(w)$ and index every codeword. After that, all the index of that codebook is dropped into 2^{nR_2} bins with the uniform distribution, where $R_2 = I(X;W) - I(Y;W)$. In the process of encoding, the source sequence X^n is compared with the codeword in the codebook until it is jointly typical with W^n. Then the index of that codeword W^n is stored. If no such codeword W^n exist, the index is set to 1. If more than one codeword W^n are jointly typical with X^n, the smallest index is stored. The output of encoder is an index of bin that contains that index of W^n. In the process of decoding, the side information Y^n is compared with the codewords specified by those indices in the bin, which is pointed by the encoder. If there is a unique codeword jointly typical with Y^n. This codeword is considered as W^n and $\hat{X}^n$ is estimated through the function $X_i = f(W_i, Y_i)$. If there is not one or more than one codewords jointly typical with Y^n. $\hat{X}^n$ is set to any codeword.

Due to the assistance of the side information, the rate $R_{WZ}(D)$ with side information is less than or equal to the one without side information. For the special case of $D = 0$, it is converted to Slepin-Wolf coding problem. Then rate required is $H(X|Y)$ bits.

1.2 Motivation and Contribution of the Thesis

Some applications of source coding have been developed using LDPC and LDGM codes with message passing algorithms. Quantizing the source of arbitrary distribution is also implemented by LDGM code with survey propagation and the corresponding simulation result [4] is extremely good. The general Slepian-Wolf coding problem has been developed by applying LDPC codes with belief propagation. For the binary source, the performance of simulation [5] is good. In this thesis, a practical scheme of Robust Slepian-Wolf coding is developed based on previous two applications and simulation is performed in a special case. After that, two approaches for Wyner-Ziv coding are derived from the application of Robust Slepian-Wolf coding. However, it is not completed due to the fact that the optimal degree distribution is still unknown.

1.3 Organization of the Thesis

The thesis is structured as follows:

- In Chapter 2, the concepts of LDGM and LDPC codes are introduced.
- In Chapter 3, the general Robust Slepian Wolf coding problem is formulated and the detailed coding scheme is developed with low density graph codes. Simulation result of a special case is provided.
- In Chapter 4, two incomplete approaches for Wyner-Ziv coding problem are developed with low density graph codes. The problem encountered in both approaches is discussed.
- In Chapter 5, this thesis is concluded and some potential methods are provided to solve the problem in Chapter 4.

Chapter 2

Low Density Graph Codes and Factor Graph

Some applications of source coding have been developed by LDPC and LDGM codes with message passing algorithms. Quantizing the source of arbitrary distribution is also implemented by LDGM code with survey propagation and the corresponding simulation result [4] is extremely good. The general Slepian-Wolf coding scheme has been developed by applying LDPC codes with belief propagation. For the binary source, the performance of simulation [5] is good. In this chapter, the concepts and some properties of LDGM and LDPC are introduced.

2.1 Low Density Generator Matrix Codes

An (n,k) binary linear block code $\mathcal{C}$ with low density generater matrix (LDGM) $\mathbf{G}$, which is a $k \times n$ matrix, is defined as follows:

$$\mathcal{C} = \{\mathbf{x} : \mathbf{x} = \mathbf{u}\mathbf{G}, \mathbf{u} \in GF^k(2)\} \tag{2.1}$$

where $\mathbf{u}$ is a $1 \times k$ information vector and $\mathbf{x}$ is a $1 \times n$ codeword and $GF(2)$ is the Galois field of two elements. The "low density" means the sparseness of ones in $\mathbf{G}$.

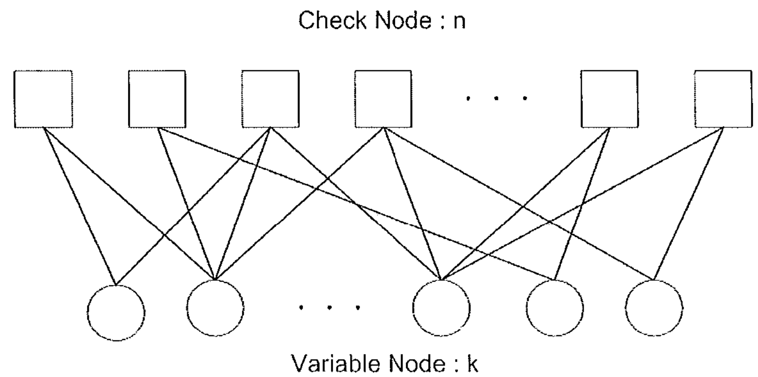

Figure 2.1: LDGM factor graph

The factor graph associated with the LDGM (2.1) is demonstrated in the Figure 2.1. Variable nodes (○) store information vector $\mathbf{u}$ and check nodes (□) store the codeword $\mathbf{x}$ based on the entire generator matrix $\mathbf{G}$. The edges between variable nodes and check nodes are drawn according to the entries $g_{i,j}$ of generator matrix $\mathbf{G}$. When the entry $g_{i,j}$ is not zero, one connection is built from the jth check node to the ith variable node.

The rate of this LDGM code is calculated by the equation (2.2). To build the irregular LDGM codes, the equations (2.3) must be satisfied. $\lambda(x)$ and $\rho(x)$ are the degree distributions in the form of polynomials for the variable nodes and check nodes, respectively. λ_i is the portion of edges on the variable node of degree i and ρ_i is the portion of edges on the check node of degree i. d_v denotes the maximum variable degree and d_c denotes the maximum check degree.

$$R = \frac{k}{n} \tag{2.2}$$

$$\frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = R \tag{2.3}$$

$$\lambda(x) = \Sigma_{i=2}^{d_v} \lambda_i x^{i-1} \tag{2.4}$$

$$\rho(x) = \Sigma_{i=2}^{d_c} \rho_i x^{i-1} \tag{2.5}$$

2.2 Low Density Parity Check Codes

Low density parity check (LDPC) codes were introduced by Gallager in the 1960s [6]. It is defined as follows:

$$\mathcal{C}_{\mathbf{s}} = \{\mathbf{x} : \mathbf{H}\mathbf{x}^T = \mathbf{s}, \mathbf{x} \in GF^n(2)\} \tag{2.6}$$

where $\mathbf{H}$ is an $(n-k) \times n$ matrix termed parity check matrix with low density referring to the sparseness of ones, s is a $1 \times (n-k)$ vector and is called syndrome. $\mathcal{C}_s$ is the coset that contains a set of x satisfying $\mathbf{H}\mathbf{x}^T = \mathbf{s}$. Furthermore, when all the syndrome bits are 0, $\mathcal{C}_{\mathbf{s}}$ is called linear code. The factor graph associated with the LDPC (2.6) is demonstrated in Figure 2.2. Variable nodes store the codeword $\mathbf{x}$ and check node store the syndrome calculated by $\mathbf{H}\mathbf{x}^T$. The edges between variable nodes and check nodes are drawn according to the entries $h_{i,j}$ of parity check matrix $\mathbf{H}$. When the entry $h_{i,j}$ is not zero, one connection is built from the jth variable node to the ith check node. The rate of this LDPC code is calculated by the equation (2.7). To build the irregular LDGM codes, the equations (2.8) must be satisfied.

$$R = \frac{n-k}{n} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} \tag{2.7}$$

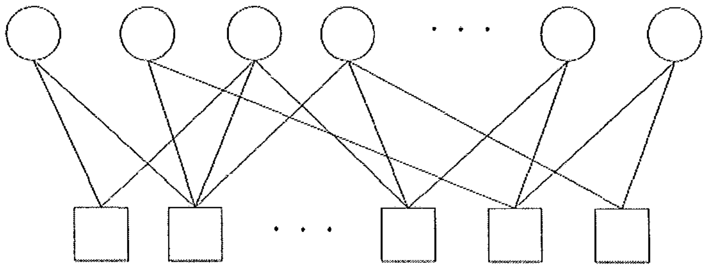

Figure 2.2: LDPC factor graph

$$1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = R \tag{2.8}$$

The definitions of $\lambda(x)$ and $\rho(x)$ are the same as in the section 2.1.

Chapter 3

Robust Slepian Wolf Coding

The general Slepian Wolf coding has been discussed in Section 1.1.1.2. Two correlated sources X and Y are encoded at two separate encoders and decoded together at a joint decoder. However, if the encoder of source Y is broken, then the joint decoder is not able to recover both sources. For enhancing the utilization of rate R_x, it is split into two parts $R_x = R_{x1} + R_{x2}$: 1) the decoder has a capacity for decoding a sequence of W^n to achieve distortion D, in the absence of the side information Y^n, with rate R_{x1}, 2) the sequence X^n is recovered losslessly at the same decoder with the side information Y^n and W^n with the rate R_{x2}. This special Slepian Wolf coding is called as Robust Slepian Wolf coding.

In this chapter, the required constraints to achieve the Robust Slepian Wolf coding will be discussed and a coding scheme is developed with low density graph codes.

3.1 Robust Slepian Wolf Coding

3.1.1 Problem Background and Formulation

As mentioned at the beginning of this chapter, Robust Slepian Wolf coding is achieved while only some constraints are satisfied. The following items list these constraints that cited from [7] and provided the corresponding explanations.

1. An auxiliary random variable W exists such that $Y \leftrightarrow X \leftrightarrow W$ forms a Markov chain.

2. $E[d(X, W)] \leq D$.

Then the Robust Slepian Wolf coding problem is formulated as:

$$R_{RSW}(D) = \min_{p(w|x)} I(X;W) + H(X|Y,W) \tag{3.1}$$

$$\text{subject to} \quad E[d(X,W)] \leq D \tag{3.2}$$

$$Y \leftrightarrow X \leftrightarrow W \tag{3.3}$$

In the equation (3.1), the first term is derived from quantizing the sequence X^n to W^n with the distortion D using rate R_{x1} and the second term is derived from recovering the sequence X^n with both side information W^n and Y^n using rate R_{x2}. If $R_{RSW} = H(X|Y)$, then one more constraint meeds to be added, that is Y and W are independent. This can be shown using the following argument. Firstly, the encoding rate of Robust Slepian Wolf Coding is calculated as follows:

$$R_{RSW}(D) = I(X;W) + H(X|W,Y) \tag{3.4}$$

$$= H(X) - H(X|W) + H(X|W,Y) \tag{3.5}$$

$$= H(X) - I(X;Y|W) \tag{3.6}$$

$$= H(X) - I(X;Y) + I(W;Y) \tag{3.7}$$

$$= H(X|Y) + I(W;Y) \tag{3.8}$$

The property of Markov chain $Y \leftrightarrow X \leftrightarrow W$ is applied from (3.6) to (3.7).

$$I(Y;W|X) = 0 \tag{3.9}$$

$$I(Y;X,W) = I(Y;W) + I(Y;X|W) \tag{3.10}$$

$$= I(Y;X) + I(Y;W|X) \tag{3.11}$$

$$I(Y;X|W) = I(Y;X) - I(Y;W) \tag{3.12}$$

Then, it is simplified until the final result (3.8) that must be less than or equal to the limited rate $H(X|Y)$ of Slepian Wolf Coding .

$$\begin{aligned} R_{RSW}(D) &\leq R_{SW} \\ H(X|Y) + I(W;Y) &\leq H(X|Y) \\ I(W;Y) &\leq 0 \end{aligned}$$

From the fundamental information theory, the mutual information is greater than or equal to 0. So the mutual information between W and Y can only be 0. Note that $I(W;Y) = 0$ if and only if W and Y are independent. So W and Y must be independent. Due to the fact that the sequence W^n is output from quantizing X^n, X and W are not independent. In order to satisfy those two relationships, the alphabet set size of X must be greater than that of Y, i.e. $|\mathcal{X}| > |\mathcal{Y}|$. Under the constrain $R_{RSW} = H(X|Y)$, one can readily formulate the following optimization problem.

$$\min E[d(X,W)] \tag{3.13}$$

$$\text{subject to} \quad \text{Y and W are independent} \tag{3.14}$$

According to the discussion about those constraints, the coding scheme of Fig.3.1 is developed. Two sequences X^n and Y^n are generated with a joint distribution $p(x,y)$. Then, the sequence X^n is quantized with the rate R_{x1} and the corresponding output of quantizer is the sequence W^n with the distortion D between them. After that,

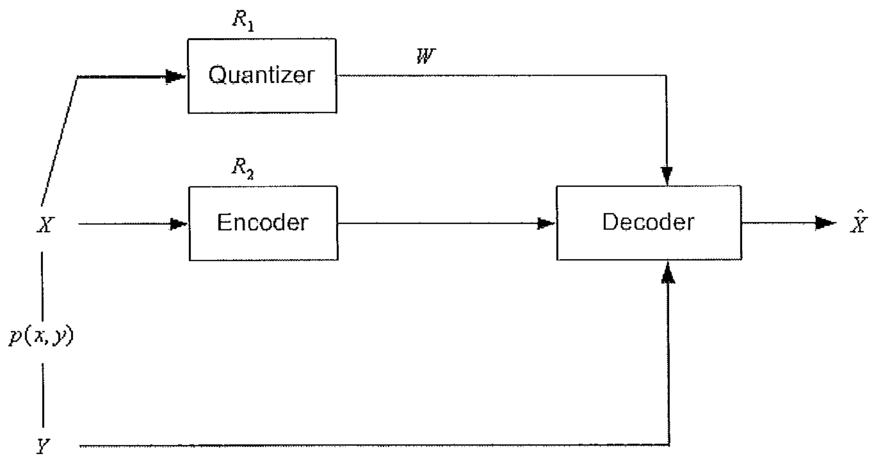

Figure 3.1: Coding Scheme of Robust Slepian-Wolf Coding Problem

a codebook is constructed through dropping all the typical sequences generated by source X into $2^{nR_{x2}}$ bins uniformly. The output of encoding the sequence X^n is an index pointing to one of those bins. In the decoder, it receives two sequences Y^n and W^n, which are compared with the sequences in the bin pointed by the index of encoding the sequence X^n. If one of sequences is jointly typical with Y^n and W^n, that sequence is assigned as the decoded sequence $\hat{X}^n$. If no such sequence exists, one sequence is chosen from that bin as the decoded sequence randomly. If the encoding rate R_{x2} is greater than or equal to $H(X|Y,W)$, the decoder will recover the lossless sequence X^n with high probability. The sum $(R_{x1}+R_{x2})$ of rates in these two steps equals to the $H(X|Y)$ and does not exceed the limited rate of the general Slepian Wolf coding.

3.2 General Case

3.2.1 Test Channels

After building the coding scheme, the quantization between sequence X^n and W^n depends on the conditional distribution $p(w_j|x_i)$ and the distribution $p(x_i)$ of the source X. The problem of minimizing distortion D is converted to one linear programming problem.

$$\text{minimize} \qquad D = \sum_{i=0}^{|\mathcal{X}|-1} \sum_{j=0, w_j \neq x_i}^{|\mathcal{W}|-1} p(w_j|x_i)p(x_i)d(x_i, w_i) \tag{3.15}$$

$$\text{subject to} \qquad p(y_k) - p(y_k|w_j) = 0 \tag{3.16}$$

$$\sum_{j=0}^{|\mathcal{W}|-1} p(w_j|x_i) - 1 = 0 \tag{3.17}$$

$$p(w_j|x_i) - 1 \leq 0 \tag{3.18}$$

$$-p(w_j|x_i) \leq 0 \tag{3.19}$$

Given the distribution $p(x_i)$, the objective function (3.15) describes that the minimum distortion D is found through all possible $p(w_j|x_i)$, which satisfy all the constraints listed last section. The equation (3.16) describes the independence between Y and W. The equation (3.17), the inequalities (3.18) and (3.19) describe the properties of the conditional distribution $p(w_j|x_i)$: the sum of all $p(w_j|x_i)$ with x_i must be 1 and each $p(w_j|x_i)$ must be in the range of $[0, 1]$. The test channels graph is shown as in the Figure 3.2 and $|\mathcal{X}| = |\mathcal{W}| > |\mathcal{Y}|$.

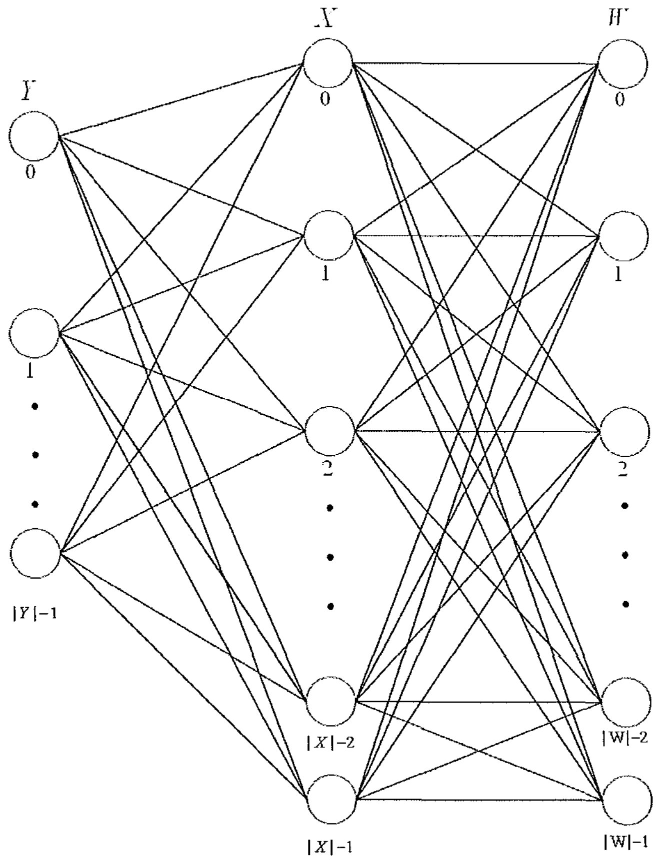

Figure 3.2: General Case of Slepian-Wolf Coding Mapping

3.2.2 Coding

Once the conditional probabilities $p(w_j|x_i)$ are fixed by solving the linear programming problem (3.15). The code scheme could be implemented through the LDGM and LDPC codes, and the corresponding factor graph is drawn as follows:

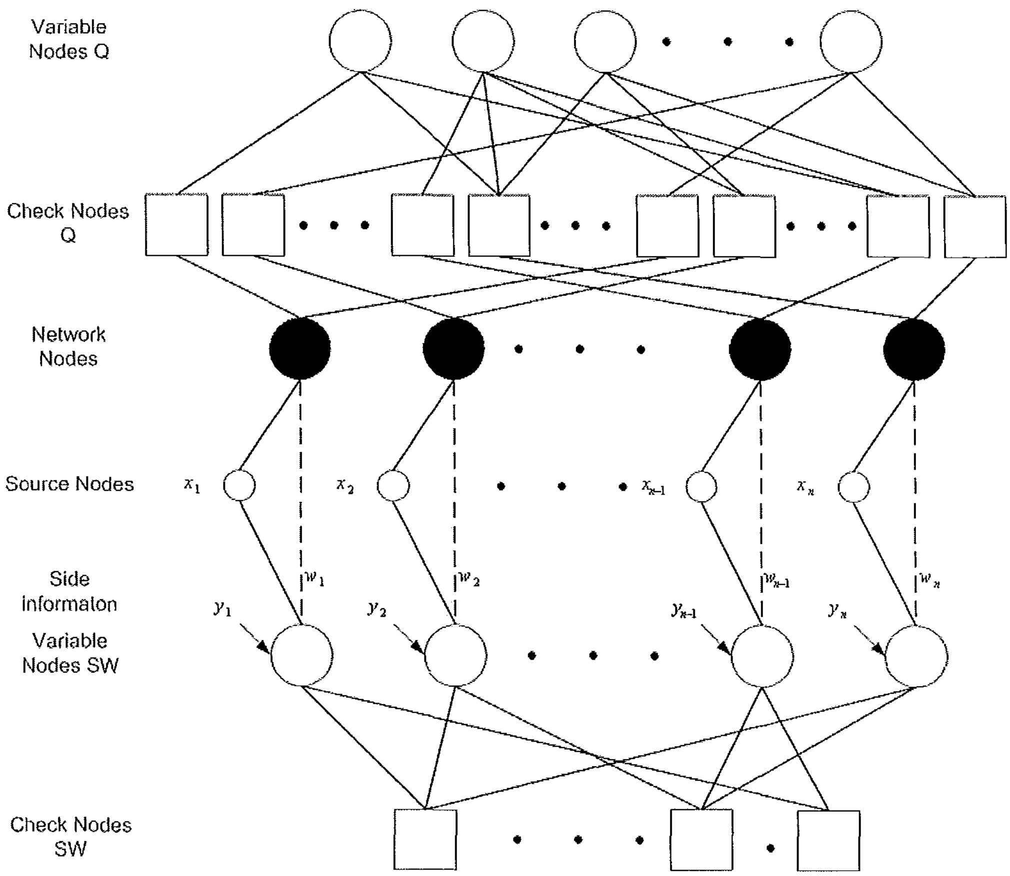

Figure 3.3: Factor Graph of Robust Slepian-Wolf Coding Scheme

There are totally 6 layers in this factor graph. The first 4 four layers construct a LDGM factor graph for quantizing the source sequence X^n. The last 3 layers construct a LDPC factor graph for encoding and decoding X^n. Due to the fact that both LDGM and LDPC factor graph have the variable nodes and check nodes, the notations "Q" and "SW" are added to distinguish.

3.2.2.1 Quantization and Encoding

In the Sun's thesis [4], the quantization of one source uniform and nonuniform distribution is researched using LDGM codes with survey propagation. The encoding simulation is very good. So this method is adopted to quantize X to W with the rate R_{x1}. According to the marginal distribution $p(w)$, a deterministic mapping is built between the network node and check nodes Q. This deterministic mapping could be considered as a function $f_M : Y^M \rightarrow \mathcal{A} = \{1, 2, \ldots, A\}$. There are totally 2^M possible binary sequences generated by the uniform distribution. Based on the distribution $p(w)$, m_i sequences are assigned to the value w. So that the following equation is satisfied.

$$\frac{m_i}{2^M} \approx p(w) = Pr\{W = w\}, w \in \mathcal{A} \tag{3.20}$$

Besides, each sequence is only allowed to assign to one value in order to construct a deterministic mapping. Then each network node connects to M check nodes in the fact graph. Once the values of check nodes are fixed, the value of network node is calculated by the function f_M. To pass message along the factor graph, the f_M is also used from network nodes to check nodes shown as equation (??).

The first four layers in the Figure 3.3 constructs the factor graph of quantization. The sequence of message-passing is listed as follows and the corresponding the calculation of message are shown as Figures 3.7, 3.5 and 3.6.

1. Initialize the vector message $M_{C_{Qi} \rightarrow V_{Qj}}$, $M_{C_{Qi} \rightarrow N_j}$ and $M_{S_i \rightarrow N_i}$. Jump to the step 3

2. Send the vector message $M_{C_{Qi} \rightarrow V_{Qj}}$ out from the check node Q C_{Qi} to the variable node Q V_{Qj}. Denote $A_{vQ}(i)$ is the set of variable nodes Q connected to the check

node Q C_{Qi}. Send the vector message $M_{C_{Qi}\to N_l}$ out from the check node Q C_{Qi} to the network node N_l.

3. Send the vector message $M_{V_{Qj}\to C_{Qi}}$ out from the variable node Q V_{Qj} to the check node Q C_i. Denote $C_{cQ}(j)$ is the set of check nodes Q connected to the variable node Q V_{Qj}. Send the vector message $M_{N_j\to C_{Qi}}$ out from the network node N_j to the check node Q C_{Qi}. Denote $D_{cQ}(j)$ is the set of check nodes connected to the network node N_j.

4. Go back the step 2 until both vector messages $M_{C_{Qi}\to V_{Qj}}$ and $M_{C_{Qi}\to N_j}$ converge or the number of iteration reaches 150 times

5. Calculate the marginal distribution M_{V_j} in each variable node Q. Set the values of some variable nodes whose bias $|M^0_{V_j} - M^1_{V_j}|$ are greater than source threshold, which usually is greater than 0.9. If there is no such bias, the value of one variable node which has the biggest bias is set. Then, remove those variable nodes from the factor graph.

6. Go back the step 2 until all the values of variable nodes are set.

7. Calculate the values of the check nodes by $\mathbf{uG}$. $\mathbf{u}$ is a vector representing the values of variable nodes.

8. Calculate the values of the network nodes according to the deterministic mapping between network nodes and check nodes.

After quantizing the sequence X^n to the sequence W^n, the sequence X^n is compressed into an index of length $k = nR_{x2} = nH(X|Y, W)$. In the process of encoding, the value of sequence X^n is passed into the variable nodes SW firstly, then the values of syndrome stored in check nodes, which is called the encoded index of sequence of

Initial vector message at the check node

$$\begin{aligned} M^0_{CQi\to VQj} &= 0.5 \\ M^1_{CQi\to VQj} &= 0.5 \\ M^0_{CQi\to N_j} &= 0.5 \\ M^1_{CQi\to N_j} &= 0.5 \end{aligned} \tag{3.21}$$

Check node to variable node

$$\begin{aligned} M^0_{CQi\to VQj} &= 0.5[1 + (M^0_{N_l\to CQi} - M^1_{N_l\to CQi}) \prod_{V_{Qm}\in A_{vQ}(i)\backslash\{V_{Qj}\}} (M^0_{V_{Qm}\to CQi} - M^1_{V_{Qm}\to CQi})] \\ M^1_{CQi\to VQj} &= 0.5[1 - (M^0_{N_l\to CQi} - M^1_{N_l\to CQi}) \prod_{V_{Qm}\in A_{vQ}(i)\backslash\{V_{Qj}\}} (M^0_{V_{Qm}\to CQi} - M^1_{V_{Qm}\to CQi})] \end{aligned} \tag{3.22}$$

Check node to network node

$$\begin{aligned} M^0_{CQi\to N_l} &= 0.5[1 + \prod_{V_{Qm}\in A_{vQ}(i)} (M^0_{V_{Qm}\to CQi} - M^1_{V_{Qm}\to CQi})] \\ M^1_{CQi\to N_l} &= 0.5[1 - \prod_{V_{Qm}\in A_{vQ}(i)} (M^0_{V_{Qm}\to CQi} - M^1_{V_{Qm}\to CQi})] \end{aligned} \tag{3.23}$$

Figure 3.4: Calculation of the Message in Check Node Q

X^n, are calculated by $\mathbf{Hx}^T$, where compression rate $R_{x2} = \frac{k}{n}$. For the binary sequence, the value in each check node is computed by operating XOR with all values stored in the variable nodes connected to that check node.

Variable node to check node

$$
\begin{aligned}
M^0_{V_{Qj}\to C_{Qi}} &= \prod_{C_{Qm}\in B_{cQ}(j)\setminus\{C_{Qi}\}} (M^0_{C_{Qm}\to V_{Qj}}) \\
M^1_{V_{Qj}\to C_{Qi}} &= \prod_{C_{Qm}\in B_{cQ}(j)\setminus\{C_{Qi}\}} (M^1_{C_{Qm}\to V_{Qj}})
\end{aligned}
\tag{3.24}
$$

Marginal distribution in Variable node

$$
\begin{aligned}
M^0_{Vj} &= \prod_{C_{Qm}\in B_{cQ}(j)} (M^0_{C_{Qm}\to V_{Qj}}) \\
M^1_{Vj} &= \prod_{C_{Qm}\in B_{cQ}(j)} (M^1_{C_{Qm}\to V_{Qj}})
\end{aligned}
\tag{3.25}
$$

Figure 3.5: Calculation of the Message in Variable Node Q

3.2.2.2 Decoding

As illustrated in Figure 3.3, the factor graph of decoding includes the last 4 layers. Both source nodes and side information nodes connect to variable nodes one by one. The edges between check nodes and variable nodes are built through a $k \times n$ parity check matrix $\mathbf{H}$ described in the section 2.2.

A lossless sequence X^n is required to recover. In the decoding process, the belief propagation is adopted. The output sequence X^n from the decoder is generated as follows and the detail calculations of message are listed in Figure 3.7.

1. Initialize the vector message M_{V_i} in the form of $(M^0_{V_i}, M^1_{V_i}, \ldots, M^{|\mathcal{X}|-1}_{V_i})$ in each variable node based on the values of both side information Y and W and the conditional probability $p(x_i|y_i, w_i)$, and set received vector message $M_{C_j\to V_i}$ to 1.

2. Send the vector message $M_{V_{SWi}\to C_{SWj}}$ out from the variable node SW V_{SWi} to the check node SW C_{SWj}. Denote $A_{cSW}(i)$ is the set of check nodes connected to the variable node SW V_{SWi}.

Source node to network node

$$M^z_{S_i \to N_i} = \exp(\gamma)/(\exp(\gamma) + \sum_{m \in \mathcal{X}: m \neq x_i} \alpha_m \exp(-\gamma)) \qquad z = x_i$$
$$M^z_{S_i \to N_i} = \alpha_k \exp(-\gamma)/(\exp(\gamma) + \sum_{m \in \mathcal{X}: m \neq x_i} \alpha_m \exp(-\gamma)) \qquad z \neq x_i \tag{3.26}$$

Network node to check node

$$M^0_{N_j \to C_{Qi}} = \sum_{z \in \mathcal{X}} M^z_{S_j \to N_j} \prod_{C_{Ql} \in D_{cQ}(j) \setminus \{C_{Qi}\}, Y_{C_{Qj}} = 0, z = f_M(Y_1, Y_2, \ldots, Y_m, \ldots, Y_M)} M^{Y_m}_{C_{Ql} \to N_j}$$
$$M^1_{N_j \to C_{Qi}} = \sum_{z \in \mathcal{X}} M^z_{S_j \to N_j} \prod_{C_{Ql} \in D_{cQ}(j) \setminus \{C_{Qi}\}, Y_{C_{Qj}} = 1, z = f_M(Y_1, Y_2, \ldots, Y_m, \ldots, Y_M)} M^{Y_m}_{C_{Ql} \to N_j} \tag{3.27}$$

Figure 3.6: Calculation of the Message in Network Node

3. Send vector message $M_{C_{SWj} \to V_{SWi}}$ out from the check node SW C_{SWj} to the variable node SW V_{SWi} . Denote $B_{vSW}(j)$ is the set of variable nodes connected to the check node SW C_{SWj}.

4. Go back the step 2 until the number of iteration reaches 150 times

5. Estimate x_i according to the final decision rule:

$$x_i = m, \text{where } M^m_F(i) = \max\{M^1_F(i), M^2_F(i), \ldots, M^Z_F(i)\} \tag{3.28}$$

Initial vector message at the variable node

$$M^z_{V_{SWi}} = p(x_i = z|y_i, w_i) \qquad z \in \mathcal{X}$$
$$M^z_{C_{SWj} \to V_{SWi}} = 1 \qquad z \in \mathcal{X} \tag{3.29}$$

Variable node to check node

$$M^z_{V_{SWi} \to C_{SWj}} = \frac{M^z_{V_{SWi}} \prod_{C_{SWm} \in A_{cSW}(i) \backslash \{C_{SWj}\}} M^z_{C_{SWm} \to V_{SWi}}}{\sum_{z=0}^{Z} M^z_{V_{SWi}} \prod_{C_{SWm} \in A_{cSW}(i) \backslash \{C_{SWj}\}} M^z_{C_{SWm} \to V_{SWi}}} \tag{3.30}$$

Check node to variable node

$$M^z_{C_{SWj} \to V_{SWi}} = \frac{\sum_{V_{SWm} \in B_{vSW}(j) \backslash \{V_{SWi}\}, \sum v_{SWm} = s_i - z} \prod M^{v_{SWm}}_{V_{SWm} \to C_{SWj}}}{\sum_{z=0}^{Z} \sum_{V_{SWm} \in B_{vSW}(j) \backslash \{V_{SWi}\}, \sum v_{SWm} = s_i - z} \prod M^{v_{SWm}}_{V_{SWm} \to C_{SWj}}} \tag{3.31}$$

Final decision rule in the variable node

$$M^z_F(i) = M^z_{V_{SWi}} \prod_{C_{SWm} \in A_{cSW}(i)} M^z_{C_{SWm} -> V_{SWi}} \tag{3.32}$$

Figure 3.7: Calculation of the Message in Variable Node and Check Node SW

3.3 Special Case

We consider a special case given in Figure 3.8 and meanings of notations in that figure are listed in the table 3.1.

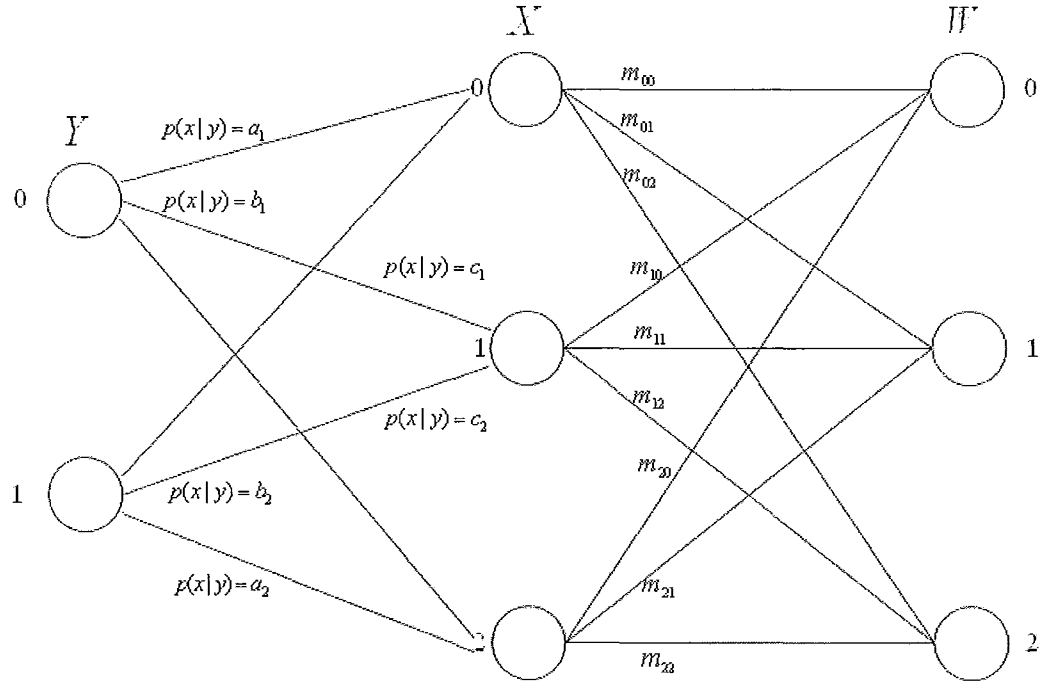

Figure 3.8: A Special Case of Robust Slepian-Wolf Coding Mapping

$a_1 = p(x = 0 \mid y = 0)$	$b_1 = p(x = 1 \mid y = 0)$	$c_1 = p(x = 2 \mid y = 0)$
$a_2 = p(x = 0 \mid y = 1)$	$b_2 = p(x = 1 \mid y = 1)$	$c_2 = p(x = 2 \mid y = 1)$
$m_{00} = p(w = 0 \mid x = 0)$	$m_{01} = p(w = 1 \mid x = 0)$	$m_{02} = p(w = 2 \mid x = 0)$
$m_{10} = p(w = 0 \mid x = 1)$	$m_{11} = p(w = 1 \mid x = 1)$	$m_{12} = p(w = 2 \mid x = 1)$
$m_{20} = p(w = 0 \mid x = 2)$	$m_{21} = p(w = 1 \mid x = 2)$	$m_{22} = p(w = 2 \mid x = 2)$

Table 3.1: Notation in General Mapping Graph

For this special case, the linear program problem is rewritten as follows:

$$\begin{aligned} \text{minimize} \qquad D &= (m_{01} + m_{02})[a_1 p + c_2(1 - p)] \\ &+ (m_{10} + m_{12})[b_1 p + b_2(1 - p)] \\ &+ (m_{20} + m_{21})[c_1 p + a_2(1 - p)] \end{aligned} \tag{3.33}$$

subject to

$$h_1(M) = (1 - m_{01} - m_{02})(a_1 - c_2) + m_{10}(b_1 - b_2) + m_{20}(c_1 - a_2) = 0 \tag{3.34}$$

$$h_2(M) = m_{01}(a_1 - c_2) + (1 - m_{10} - m_{12})(b_1 - b_2) + m_{21}(c_1 - a_2) = 0 \tag{3.35}$$

$$f_1(M) = m_{01}^2 - m_{01} \leq 0 \tag{3.36}$$

$$f_2(M) = m_{02}^2 - m_{02} \leq 0 \tag{3.37}$$

$$f_3(M) = m_{10}^2 - m_{10} \leq 0 \tag{3.38}$$

$$f_4(M) = m_{12}^2 - m_{12} \leq 0 \tag{3.39}$$

$$f_5(M) = m_{20}^2 - m_{20} \leq 0 \tag{3.40}$$

$$f_6(M) = m_{21}^2 - m_{21} \leq 0 \tag{3.41}$$

Function $h(M)$ is derived from the independence between Y and W and Function $f(M)$ derives from the properties of the conditional probabilities $p(w_j|x_i)$. In the equation (3.33), hamming distortion is applied.

$$d(x_i, w_i) = \begin{cases} 0 & \text{if } x_i = w_i \\ 1 & \text{if } x_i \neq w_i \end{cases} \quad (3.42)$$

3.3.1 Symmetric Case

Firstly, the simplest case is analyzed. Assume the conditional probabilities $p(x|y)$ between the sources X and Y are symmetric,

$$a = a_1 = a_2 \quad (3.43)$$

$$b = b_1 = b_2 \quad (3.44)$$

$$c = c_1 = c_2 \quad (3.45)$$

then this linear programming problem is simplified further.

$$\begin{aligned} \text{minimize} \quad D &= m_{01}[ap + (1-p)c] + m_{02}[ap + (1-p)c] \\ &+ m_{10}b + m_{12}b \\ &+ m_{20}[cp + (1-p)a] + m_{21}[cp + (1-p)a] \end{aligned} \quad (3.46)$$

$$\text{subject to} \quad (m_{00} - m_{20})(a - c) = 0 \quad (3.47)$$

$$(m_{01} - m_{21})(a - c) = 0 \quad (3.48)$$

$$(m_{02} - m_{22})(a - c) = 0 \quad (3.49)$$

Note that the condition $a \neq c$ must be held to avoid the violation of $|\mathcal{X}| > |\mathcal{Y}|$. Once $a = c$ exists, the probabilities from y_i to x_0 and x_2 are the same. So that these two bits x_0 and x_2 are equivalent and are merged to one bit. The alphabet set size of X decreases to 2, which is the same as that of Y.

Given the following definitions:

$$\begin{aligned}
m_{00} = m_{20} &= A \\
m_{01} = m_{21} &= B \\
m_{02} = m_{22} &= C \\
ap + c - cp = E_1 &\geq 0 \\
cp + a - ap = E_2 &\geq 0 \\
D = A \times (E_2 - E_1) &+ B \times E_2
\end{aligned}$$

There are 3 kinds of solutions in this problem based on the relationship between E_1 and E_2.

Case: $E_1 > E_2$	Case: $E_1 = E_2$	Case: $E_1 < E_2$
$A = 1$	$A + C = 1$	$A = 0$
$B = 0$	$B = 0$	$B = 0$
$C = 0$		$C = 1$

From the above solution, it is obvious that the cases $E_1 > E_2$ and $E_1 < E_2$ are similar and symmetric. So only solutions of two cases $E_1 > E_2$ and $E_1 = E_2$ are displayed in Figure 3.9 and Figure 3.10, respectively.

As observed from those figures above, the X in the decoder can be estimated by both side information Y and W directly, without the index from encoding X. These mappings reveal that the total rate of Robust Slepain-Wolf coding is consumed in the quantizer $R_{x1} = H(X|Y)$ and $R_{x2} = 0$ in the symmetric case.

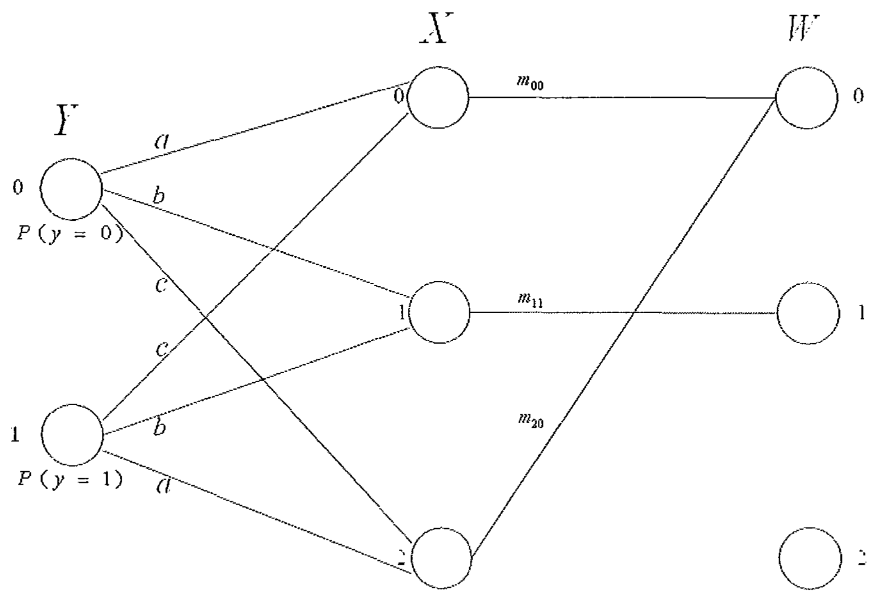

Figure 3.9: Robust Slepian-Wolf Coding Mapping with $E_1 > E_2$

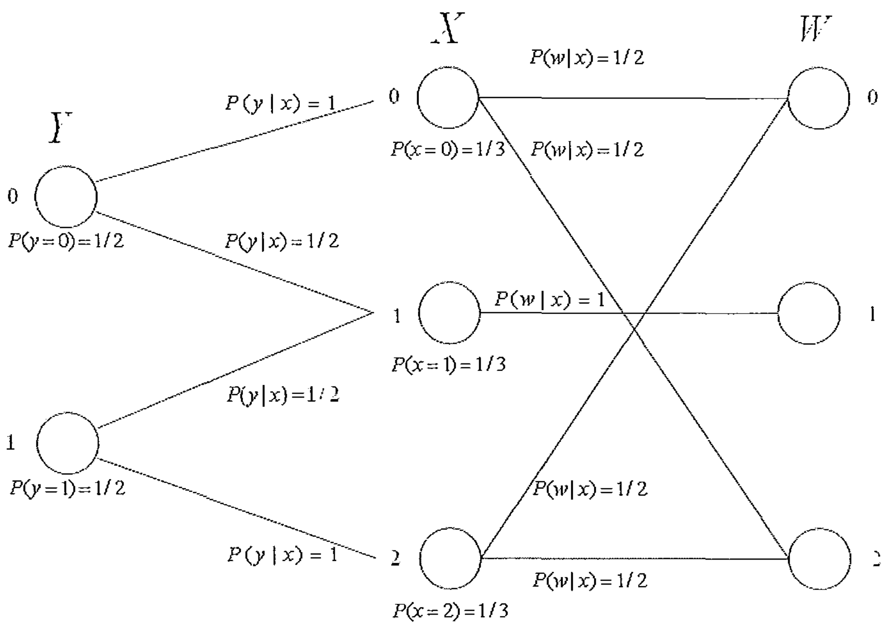

Figure 3.10: Robust Slepian-Wolf Coding Mapping with $E_1 = E_2$

$a_1 = \frac{1}{2}$	$b_1 = \frac{1}{2}$	$c_1 = 0$
$a_2 = \frac{2}{3}$	$b_2 = \frac{1}{3}$	$c_2 = 0$

Table 3.2: Values of the Conditional Probabilities $p(x_j|y_i)$

$m_{00} = 0$	$m_{01} = 0$	$m_{02} = 1$
$m_{10} = 0$	$m_{11} = 1$	$m_{12} = 0$
$m_{20} = 0$	$m_{21} = \frac{1}{4}$	$m_{22} = \frac{3}{4}$

Table 3.3: Optimum Mapping Values

3.3.2 Asymmetric Case

In this section, an asymmetric case is analyzed. The value of conditional probabilities $p(x_i|y_j)$ is set in the following table 3.2. The mapping values in the table 3.3 between X and W is found by applying the numerical method and proved that they satisfy the KKT condition [8]. As discussed in Appendix A, the corresponding λ and ν of this optimization problem exist and is listed as follows:

$$\begin{aligned}
\nu_1 &= \frac{3}{5} & \nu_2 &\in (-\frac{3}{5}, \frac{2}{5}) \\
\lambda_1 &= \frac{1}{2} - \frac{1}{2}\nu_2 & \lambda_2 &= \frac{1}{5} - \frac{1}{2}\nu_2 \\
\lambda_3 &= \frac{3}{10} - \frac{1}{6}\nu_2 & \lambda_4 &= \frac{3}{10} \\
\lambda_5 &= \frac{2}{5} + \frac{2}{3}\nu_2 & \lambda_6 &= 0
\end{aligned}$$

According to the conditional probability $p(x|y)$ and the optimum mapping, the mapping diagram Figure 3.11 is drawn. In this figure, when both y and w are 1, x has two possible outputs 1 or 2. So the estimated $\hat{X}$ can no longer be obtained through Y and X directly, the encoding index of X is required to losslessly recover X.

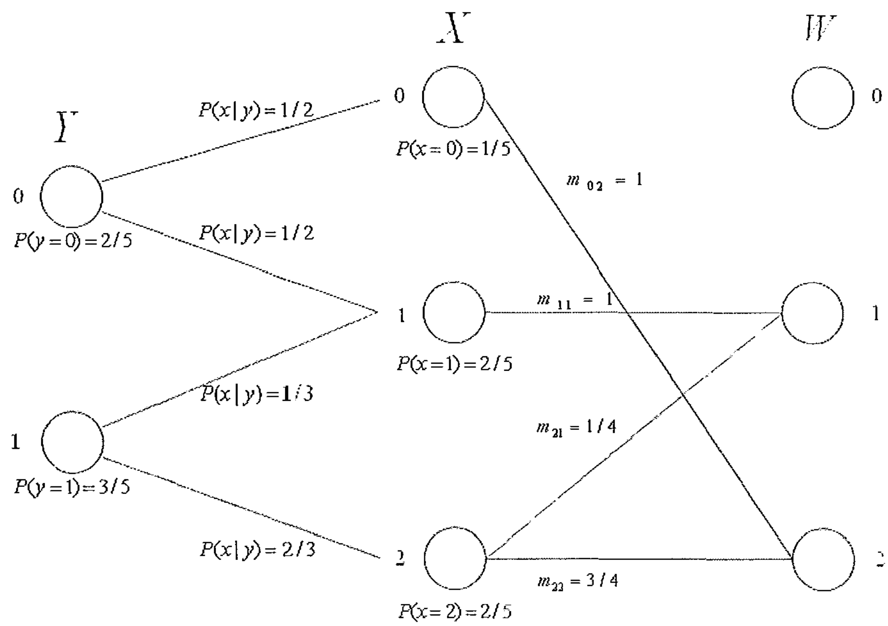

Figure 3.11: Asymmetrical Case of Robust Slepian-Wolf Coding Mapping

3.3.3 Coding

After building the mapping of Markov chain $Y \leftrightarrow X \leftrightarrow W$, the detailed implementation is developed by applying LDGM and LDPC. The factor graph is shown in the Figure 3.3.

3.3.3.1 Quantization with LDGM

According to the equation (3.27) and the constraints of deterministic function, the alphabet set size of quantized sequence must be the same as one of source sequence. So the mapping shown in in Figure 3.12 is modified based on the Figure 3.11. In this new mapping, the conditional probability $p(w = 2|x = 0) = 1$ is replaced by $p(w = 0|x = 0) = 1$, so the alphabet set size of W is still ternary. After quantization, all the symbol "0" in W^n are converted to 2. The deterministic mapping is built

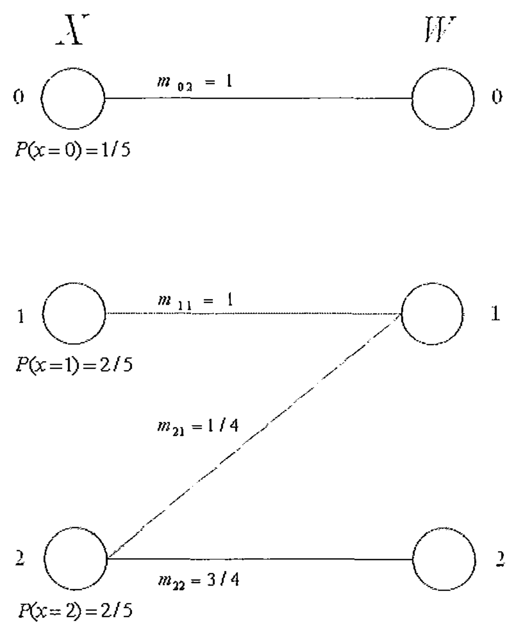

Figure 3.12: Grouping Side information

in the network node as shown in Figure 3.13 by the distribution of W. From the optimal mapping in figure 3.12 between X and W, the conditional probabilities of $p(w = 0|x = 0)$ and $p(w = 1|x = 1)$ are 1. So the penalties from $x = 0$ to $w = 0$ and from $x = 1$ to $w = 1$ must be set to 0. However, it is impossible to reach 0 in the penalty equation (3.26) so that some inevitable errors are generated. For this issue, the extra rate R_e is used to store the index of those bits of x and force them into mapping the correct w.

3.3.3.2 Decoding with LDPC

The process of decoding is almost the same as that in section 3.2.2.2. According to the mapping in Figure 3.11, x only requires to be distinguished when y equals to 1 and w equals to 1. So other combinations between Y and W could be grouped into one set as shown in Figure 3.14.

Z_1	Z_2	Z_3	Z_4	W
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	2
1	1	0	0	2
1	1	0	1	2
1	1	1	0	2
1	1	1	1	2

Figure 3.13: Mapping of Network Node in LDGM

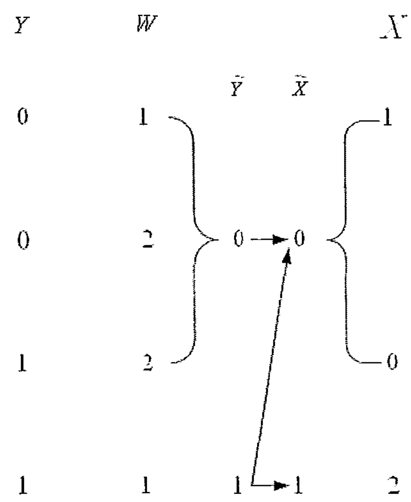

Figure 3.14: Grouping Side information

After grouping the side information, the format of message is changed to LLR (Log-likelihood Ratio) instead of the vector message for the binary source and the corresponding calculation of message is listed in figure 3.15. The final decision rule in the variable node is also modified as

$$\hat{x}_i = \begin{cases} 0 & \text{if } M_F(i) \geq 0 \\ 1 & \text{if } M_F(i) < 0 \end{cases} \tag{3.50}$$

3.3.4 Simulation Result

Two kinds of sequences from source X with different lengths are simulated for the coding scheme discussed above. Their performances are demonstrated in Figure 3.16. The red line and blue line represent the simulated results of the sequences of length $10,000$ and $100,000$, respectively. For each length, 1000 sequences are tested to average their performance. The total rate R is composed of three parts R_1 used in quantization, R_e used for fixing the inevitable error in quantization and R_2 used for encoding the index of $\mathbf{X}$. During the simulation, the maximum number of the

Initial vector message at the variable node

$$M_{V_i} = \log \frac{p(x_i = 0|y_i, w_i)}{p(x_i = 1|y_i, w_i)} \qquad M_{C_j \to V_i} = 0 \tag{3.51}$$

Variable node to check node

$$M_{V_i \to C_j} = M_{V_i} + \sum_{C_m \in A_C(i) \setminus \{C_j\}} M_{C_m \to V_i} \tag{3.52}$$

Check node to variable node

$$\tanh(\frac{M_{C_j \to V_i}}{2}) = (1 - 2s_j) \prod_{V_m \in B_V(j) \setminus \{V_i\}} \tanh(\frac{M_{V_m \to C_j}}{2}) \tag{3.53}$$

Marginal distribution in the variable node

$$M_F(i) = M_{V_i} + \prod_{C_m \in A_C(i)} M_{C_m -> V_i} \tag{3.54}$$

Figure 3.15: Calculation of the Message in Variable Node and Check Node in the Format of LLR

inevitable error bit is 10 when the length of test sequence is 10,000, so R_e is set to 0.014. Similarly, the maximum number of the inevitable error bit is 20 when the length of test sequence is 100,000, so R_e is set to 0.00332. Comparing two lines in the Figure 3.16, the line representing the performance of the sequence of the sequence of length 100,000 is obviously closer to the Slepian-Wolf limit.

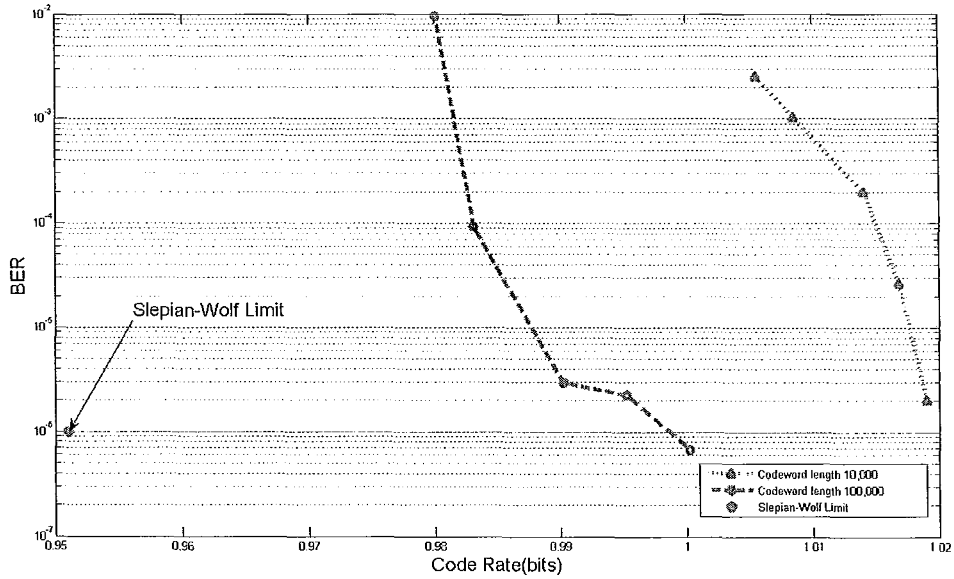

Figure 3.16: Simulated Result of Robust Slepian-Wolf Coding Problem

Chapter 4

Two Incomplete Approaches for Wyner-Ziv Coding Using Low Density Graph Codes

In Chapter 1, the coding problem of two correlated sources X and Y is introduced. Slepian-Wolf coding problem is analyzed and is implemented by using Low Density Graph Codes in previous chapter. The coding scheme for Wyner-Ziv coding problem is developed based on one of the Slepian-Wolf schemes. As illustrated in Figure 4.1, the step of encoding includes two parts: 1) The sequence X^n of length n is quantized into the sequence W^n with the distortion d. 2) A codebook is constructed as a $2^{I(X;W)} \times n$ matrix and each entry in the matrix is i.i.d drawn from the alphabet with the probability distribution of $p(w)$. All codewords in the codebook are uniformly dropped into $2^{R_{WZ}(d)}$ bins. The quantized sequence W is encoded to an index to point to one bin. In the decoder, it receives sequences Y^n, which is compared with the sequences in the bin pointed by the index of encoding the sequence W^n. If one of sequences is jointly typical with Y^n, that sequence is assigned as the decoded

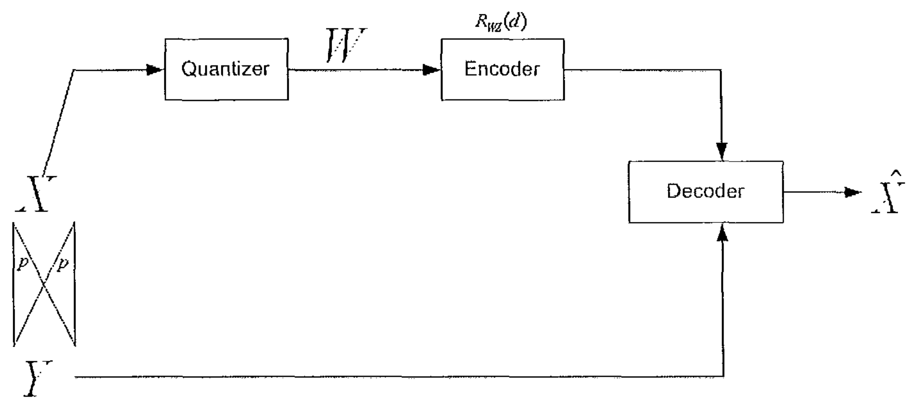

Figure 4.1: Coding Scheme of Wyner-Ziv

sequence $\hat{X}^n$. If no such sequence exists, one sequence is chosen from that bin as the decoded sequence randomly. If the encoding rate R_d is greater than or equals to $I(X;W) - I(Y;W)$, the decoder will recover the sequence $\hat{X}^n$ with distortion with high probability.

In this chapter, we only discuss one special case doubly symmetric binary sources X and Y. The size of alphabet set of X and Y is binary, and sequences X^n and Y^n are generated from those two sources by a crossover probability p, which is in the range of $[0, \frac{1}{2}]$. The minimum achievable rate region is found in Appendix B. The following two approaches are developed for the case $d < d_c$.

4.1 First Approach Using LDGM and LDPC

The idea of first approach is from the Robust Slepian-Wolf coding problem, so they have the similar factor graphs. The sequence X^n is quantized into W^n by using LDGM. Then W^n is encoded into an index of length $nR_{WZ}(d)$ through LDPC. At last, combining that LDGM and LDPC is to decode the sequence $\hat{X}^n$. As shown in the Figure 4.2, there are totally 5 layers. The first 3 layers construct the factor graph of LDGM used to quantize the sequence X^n. The last 3 layers construct the factor graph of LDPC used to encode the sequence W^n. The entire factor graph is used to decode the sequence $\hat{X}^n$.

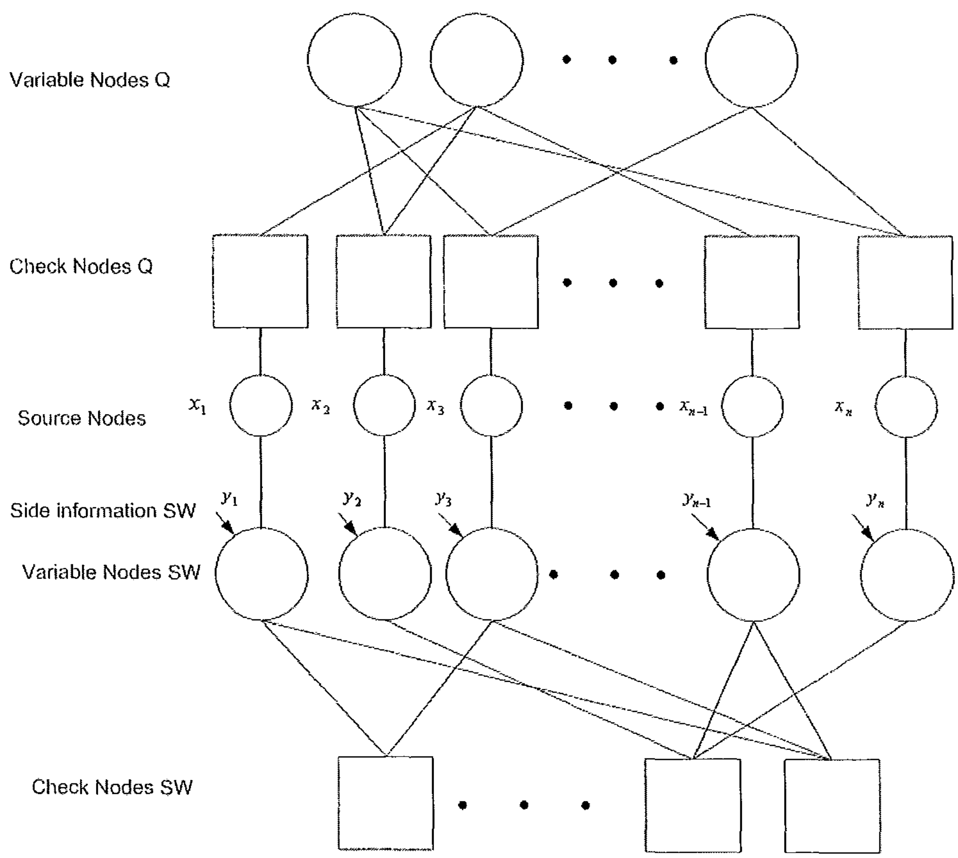

Figure 4.2: Factor Graph of First Approach

The encoding and decoding of this approach are analyzed in following subsections.

4.1.1 Encoding

The detailed process of quantization using LDGM code has been described in the section 3.2.2.1 and it is applied directly in this part. So there is no more discussion about it. After obtaining the quantized sequence W^n, the values of W^n are passed into variable nodes in the fourth layer. Then an index of W^n is calculated by the $\mathbf{H}\mathbf{w}^T$, where $\mathbf{H}$ is $nR_{WZ}(d) \times n$ parity check matrix.

4.1.2 Decoding

Due to using the side information Y^n and an index of W^n in the decoding process, the message in the form of $LLR = \log \frac{p(w=0|y)}{p(w=1|y)}$ is passed, where the conditional probability $p(w|y)$ is found through the Markov chain $Y \leftrightarrow X \leftrightarrow W$. Here, it provides the reason why the sequence W^n can not be decoded by using only LDPC, not like general Slepain-Wolf coding problem. The Wyner-Ziv rate $R_{WZ}(d)$ is less than or equal to the Slepian-Wolf rate $H(W|Y)$ and the number of bins is , which is less than or equals to $2^{H(W|Y)}$. The LDGM constraint is added to ensure the codebook generated by the marginal distribution $p(z)$, dropped the indices of the codewords in that codebook into $2^{nR_{WZ}(d)}$ bins, and identity the codeword from the other sequences jointly typical with Y^n in the bin pointed by the index from encoding the sequence W^n. So the factor graph of LDGM code used in the quantization is connected with the one of LDPC code as the constraint. Now, the decoder sequence $\hat{X}^n$ is not only in the bin pointed by the index of W^n, but also is the output of quantizing X^n. The is why the factor graph combined by both LDPC and LDGM codes is required to decode the sequence $\hat{X}^n$.

The detailed message-passing algorithm in the factor graph is presented as follows:

1. Initialize the $M_{V_{SWi}}$ computed in the equation (4.2) in the form of LLR (Log-likelihood Ratio) in each variable node SW based on the value of side information Y and the conditional probability $p(w_i|y_i)$. Set all received messages $M_{C_{SWj} \to V_{SWi}}$ and $M_{S_i \to V_{SWi}}$to 0

2. Send message $M_{V_{SWi} \to C_{SWj}}$ computed in the equation (4.4) out from the variable node SW V_{SWi} to the check node SW C_{SWj}. Denote $A_{cSW}(i)$ as the set of check nodes SW connected to the variable node V_{SWi}

3. Send message $M_{V_{SWi} \to S_i}$ computed in the equation (4.5) out from the variable node SW V_{SWi} to the source node S_i. Then send $M_{S_i \to C_{Qi}}$ computed in the equation (4.8) out from the source node S_i to the check node Q C_{Qi} with the format of message converted from LLR to a set of $p(w = 0|y)$ and $p(w = 1|y)$.

4. Send message $M_{C_{Qi} \to V_{Qj}}$ computed in the equation (4.9) out from the check node Q C_{Qi} to the variable node Q V_{Qj}. Denote $B_{vQ}(i)$ is the set of variable nodes Q connected to the check node C_{Qi}.

5. Send message $M_{C_{SWj} \to V_{SWi}}$ computed in the equation (4.9) out from the check node C_{SWj} to the variable node V_{SWi} . Denote $B_{vSW}(j)$ is the set of variable nodes connected to the check node C_{SWj}.

6. Send message $M_{V_{Qj} \to C_{Qi}}$ out from the variable node Q V_{Qj} to the check node C_{Qi} . Denote $A_{cQ}(j)$ is the set of check nodes Q connected to the variable node V_{Qj}. Then send $M_{C_{Qi} \to S_i}$ out from the check node Q C_{Qi} to the source node S_i and convert the message format from a set of $p(w = 0|y)$ and $p(w = 1|y)$ to LLR.

7. Go back the step 2 until the number of iteration reaches 150 times

8. Estimate x_i according to the final decision rule: where $M_F(i)$ computed in the equation (4.7)

$$\hat{x}_i = \begin{cases} 0 & \text{if } M_F(i) \geq 0 \\ 1 & \text{if } M_F(i) < 0 \end{cases} \tag{4.1}$$

Initial message at the variable node SW

$$\begin{aligned} M_{V_{SWi}} &= \log \frac{p(w_i = 0|y_i)}{p(w_i = 1|y_i)} \\ M_{C_{SWj} \to V_{SWi}} &= 0 \\ M_{S_i \to V_{SWi}} &= 0 \end{aligned} \tag{4.2}$$

Source node to Variable Nodes SW

$$M_{S_i \to V_{SWi}} = \log \frac{M^0_{C_{Qi} \to S_i}}{M^1_{C_{Qi} \to S_i}} \tag{4.3}$$

Variable node SW to check node SW

$$M_{V_{SWi} \to C_{SWj}} = M_{S_i \to V_{SWi}} + M_{V_{SWi}} + \sum_{C_{SWm} \in A_{cSW}(i) \backslash \{C_{SWj}\}} M_{C_{SWm} \to V_{SWi}} \tag{4.4}$$

Variable node SW to source node

$$M_{VSWi \to S_i} = \sum_{C_{SWm} \in A_{cSW}(i)} M_{C_{SWm} -> V_{SWi}} \tag{4.5}$$

Check node SW to variable node SW

$$\tanh\left(\frac{M_{C_{SWj} \to V_{SWi}}}{2}\right) = (1 - 2s_j) \prod_{V_{SWm} \in B_{vSW}(j) \backslash \{V_{SWi}\}} \tanh\left(\frac{M_{V_{SWm} \to C_{SWj}}}{2}\right) \tag{4.6}$$

Final decision rule in the variable node

$$M_F(i) = M_{V_i} + \sum_{C_m \in A_C(i)} M_{C_m -> V_i} \tag{4.7}$$

Figure 4.3: Calculation of the Message in Variable Node SW and Check Node SW

According to the rate $R_{WZ}(d)$, the optimal degree distributions of check node SW and variable node SW are found in the website [9], However, the degree in the check nodes is too high. So that one problem is met in step 5 that the message $M_{C_{SWj}\rightarrow V_{SWi}}$, which is calculated by in the equation 4.6, is close to 0 and $M_{V_{SWi}\rightarrow C_{SWj}}$ never changes in each iteration.

4.2 Second Approach Using LDPC

Quantization in the second approach is implemented by LDPC instead of LDGM in order to modify the degree distribution of variable nodes and check nodes. The whole factor graph is shown in the Figure 4.5. There are 2 parity check matrixes in this factor graph. $\mathbf{H_1}$ is used to quantize the X^n to W^n and $\mathbf{H_2}$ is used to encode W^n to an index of length $nR_{WZ}(d)$. After that, one combines two matrices to $\mathbf{H} = [\frac{\mathbf{H_1}}{\mathbf{H_2}}]$ for decoding the $\hat{X}^n$.

4.2.1 Encoding

Quantization is implemented by LDPC with survey propagation. The sequence W^n must satisfy $\mathbf{H_1}(W^n)^T = \mathbf{0}$ and all values of syndrome in the check nodes are 0. Encoding is accomplished using survey propagation on the factor graph obtained from Figure 4.5 by removing check nodes $\mathbf{H_2}$.

1. Initialize the vector message $M_{S_i \to V_i}$ computed in the equation (4.12)and message.

2. Send message $M_{V_i \to C_j}$ computed in the equation (4.13) out from the variable node V_i to the check node C_j. Denote $A_c(i)$ as the set of check nodes connected to the variable node V_i.

3. Send message $M_{C_j \to V_i}$ computed in the equation (4.14) out from the check node C_j to the variable node V_i . Denote $B_v(j)$ as the set of variable nodes connected to the check node C_j.

4. Go back the step 2 until the vector message $M_{C_j \to V_i}$ computed in the equation (4.14) or the number of iteration reaches the 150 times

5. Calculate the marginal distribution M_{V_j} computed in the equation (4.15) in each variable node. Set the values of some variable nodes whose bias $|M^0_{V_j} - M^1_{V_j}|$ are greater than one threshold, which usually is less than 0.1. If there is no such bias, the value of one variable node which has the biggest bias is set. Then, remove those variable nodes from the factor graph.

6. Go back the step 2 until all the values of variable nodes are set.

Once the sequence W^n is generated, $\mathbf{H_2}(W^n)^T$ produces an index of length $nR_{WZ}(d)$. However, the index used by decoding is composed of $\mathbf{H_1}(W^n)^T$ and $\mathbf{H_2}(W^n)^T$. As we know, $\mathbf{H_1}(W^n)^T$ always is $\mathbf{0}$, so the decoder is assumed to know $\mathbf{H_1}(W^n)^T$ already. Therefore, the coding rate is $R_{WZ}(d)$.

4.2.2 Decoding

In the process of decoding, the sequence $\hat{X}^n$ is decoded by the side information Y^n and the index from $\mathbf{H}(W^n)^T$. The message passed along the factor graph $\mathbf{H}$. Hence, the process of decoding is exactly the same as Slepian-Wolf coding problem. The procedure can be refereed in the section 3.3.3.2. In this approach, although the degree in the variable node is increase, the degree in the check node referring $\mathbf{H_2}$ is unchanged. Therefore, the same problem exist. The message $M_{C_j \to V_i}$ in the $\mathbf{H_2}$ is close to 0 in each iteration.

Source node to Check node Q

$$M^0_{S_i \to C_{Qi}} = \frac{M_{VSWi \to S_i}}{1 - M_{V_{SWi} \to S_i}}$$
$$M^1_{S_i \to C_{Qi}} = \frac{1}{1 - M_{V_{SWi} \to S_i}} \tag{4.8}$$

Check node Q to variable node Q

$$M^0_{C_{Qi} \to V_{Qj}} = \frac{1}{2}[1 + (M^0_{S_i \to C_{Qi}} - M^1_{S_i \to C_{Qi}}) \prod_{m \in B_{vQ}(i) \backslash \{V_{Qj}\}} (M^0_{V_{Qm} \to C_{Qi}} - M^1_{V_{Qm} \to C_{Qi}})]$$
$$M^1_{C_{Qi} \to V_{Qj}} = \frac{1}{2}[1 - (M^0_{S_i \to C_{Qi}} - M^1_{S_i \to C_{Qi}}) \prod_{m \in B_{vQ}(i) \backslash \{V_{Qj}\}} (M^0_{V_{Qm} \to C_{Qi}} - M^1_{V_{Qm} \to C_{Qi}})] \tag{4.9}$$

Variable node Q to Check node Q

$$M^0_{V_{Qj} \to C_{Qi}} = \prod_{m \in A_{cQ}(j) \backslash \{C_{Qm}\}} M^0_{C_{Qm} \to V_{Qj}}$$
$$M^1_{V_{Qj} \to C_{Qi}} = \prod_{m \in A_{cQ}(j) \backslash \{C_{Qm}\}} M^1_{C_{Qm} \to V_{Qj}} \tag{4.10}$$

Check node Q to Source Node

$$M^0_{C_{Qi} \to S_i} = \prod_{m \in B_{vQ}(i)} M^0_{V_{Qm} \to C_{Qi}}$$
$$M^1_{C_{Qi} \to S_i} = \prod_{m \in B_{vQ}(i)} M^1_{V_{Qm} \to C_{Qi}} \tag{4.11}$$

Figure 4.4: Calculation of the Message in Variable Node Q and Check Node Q, M_0 and M_1 have to be normalized to sum to 1

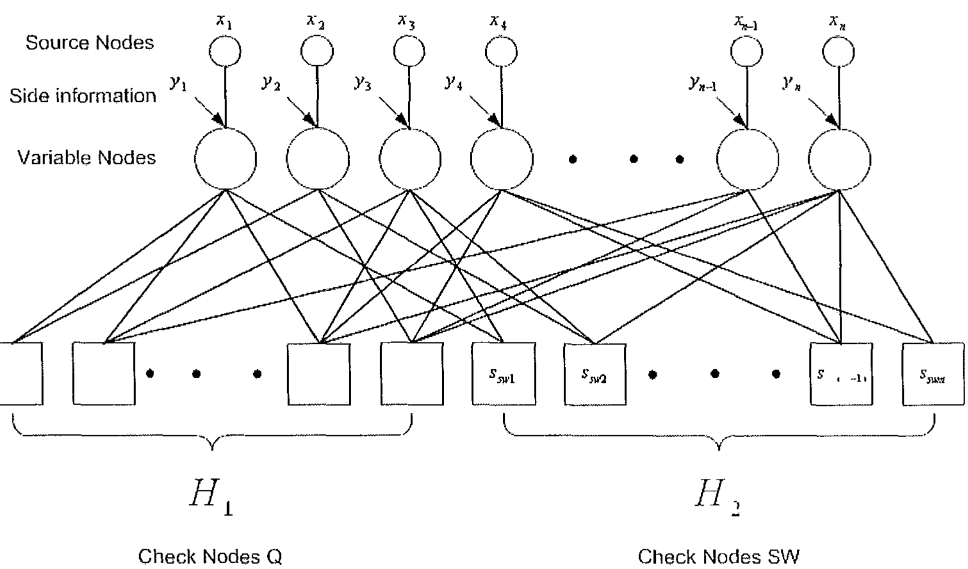

Figure 4.5: Factor Graph of Second Approach

Source node to Variable node

$$\begin{aligned} M^{z}_{S_i \to V_i} &= \exp(\gamma)/(\exp(\gamma) + \sum_{m \in \mathcal{X}: m \neq x_i} \alpha_m \exp(-\gamma)) \qquad z = x_i \\ M^{z}_{S_i \to V_i} &= \alpha_k \exp(-\gamma)/(\exp(\gamma) + \sum_{m \in \mathcal{X}: m \neq x_i} \alpha_m \exp(-\gamma)) \qquad z \neq x_i \end{aligned} \tag{4.12}$$

Variable node to check node

$$\begin{aligned} M^{0}_{V_j \to C_i} &= M^{0}_{S_j \to V_j} \prod_{C_m \in A_c(j) \backslash \{C_i\}} (M^{0}_{C_m \to V_j}) \\ M^{1}_{V_j \to C_i} &= M^{1}_{S_j \to V_j} \prod_{C_m \in A_c(j) \backslash \{C_i\}} (M^{1}_{C_m \to V_j}) \end{aligned} \tag{4.13}$$

Check node to variable node

$$\begin{aligned} M^{0}_{C_i \to V_j} &= 0.5[1 + \prod_{V_m \in B_v(i) \backslash \{V_j\}} (M^{0}_{V_m \to C_i} - M^{1}_{V_m \to C_i})] \\ M^{1}_{C_i \to V_j} &= 0.5[1 - \prod_{V_m \in B_v(i) \backslash \{V_j\}} (M^{0}_{V_m \to C_i} - M^{1}_{V_m \to C_i})] \end{aligned} \tag{4.14}$$

Marginal distribution in variable node

$$\begin{aligned} M^{0}_{V_j} &= M^{0}_{S_j \to V_j} \prod_{C_m \in A_c(j)} (M^{0}_{C_m \to V_j}) \\ M^{1}_{V_j} &= M^{1}_{S_j \to V_j} \prod_{C_m \in A_c(j)} (M^{1}_{C_m \to V_j}) \end{aligned} \tag{4.15}$$

Figure 4.6: Calculation of the Message in Variable Node and Check Node, M_0 and M_1 have to be normalized to sum to 1

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, some basic concepts of source coding have been introduced and the low density graph codes (LDPC and LDGM) have been briefly discussed. A coding scheme of Robust Slepian-Wolf coding problem is implemented in two steps: 1) an LDGM code is applied for quantizing the source sequence, then an LDPC code is used for encoding. 2)In the process of decoding, only LDPC codes is used. Two kind of sequences of different length is simulated and the corresponding result is good. After that, two incomplete approaches are developed from the coding scheme of Robust Slepian-Wolf coding problem in order to achieve the rate of Wyner-Ziv problem. However, it is blocked by one problem that the optimal degree distribution is not found.

5.2 Future Work

The future work focuses on finding the optimal degree distribution, so that those two incomplete approaches could work, especially the second one. Right now, the entire parity matrix $\mathbf{H} = [\frac{\mathbf{H}_1}{\mathbf{H}_2}]$ is generated by $\mathbf{H}_1$ and $\mathbf{H}_2$ separately. Although two matrixes $\mathbf{H}_1$ and $\mathbf{H}_2$ satisfy the constraint in each part, the whole matrix does not ensure the constraint of decoding. Because the decoder requires the entire matrix $\mathbf{H}$, the entire parity matrix $\mathbf{H}$ must be generated to satisfy the constraints of quantization and decoding at same time. It is probably implemented by density evolution [10], [11], [12].

Appendix A

K.K.T conditions

All the optimal solution of the linear program problem 3.33 must satisfy the following K.K.T conditions:

$$L(M) = D(M) + \sum_{i=1}^{6} \lambda_i f_i(M) + \sum_{i=1}^{2} \nu_i h_i(M)$$

$$\begin{aligned}
f_i(M^*) &\leq 0, i = 1, ...6 \\
h_1(M^*) &= (1 - m_{01}^* - m_{02}^*)(a_1 - c_2) + m_{10}^*(b_1 - b_2) + m_{20}^*(c_1 - a_2) = 0 \\
h_2(M^*) &= m_{01}^*(a_1 - c_2) + (1 - m_{10}^* - m_{12}^*)(b_1 - b_2) + m_{21}^*(c_1 - a_2) = 0 \\
\lambda_i &\geq 0, i = 1, ...6 \\
\lambda_i \times f_i(M^*) &= 0, i = 1, ...6
\end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial m_{01}^*} &= [a_1 p + c_2(1-p)] + 2\lambda_1 m_{01}^* - \lambda_1 - \nu_1(a_1 - c_2) + \nu_2(a_1 - c_2) = 0 \\
\frac{\partial L}{\partial m_{02}^*} &= [a_1 p + c_2(1-p)] + 2\lambda_2 m_{02}^* - \lambda_2 - \nu_1(a_1 - c_2) = 0 \\
\frac{\partial L}{\partial m_{10}^*} &= [b_1 p + b_2(1-p)] + 2\lambda_3 m_{10}^* - \lambda_3 - \nu_1(b_1 - b_2) - \nu_2(b_1 - b_2) = 0 \\
\frac{\partial L}{\partial m_{12}^*} &= [b_1 p + b_2(1-p)] + 2\lambda_4 m_{12}^* - \lambda_4 - \nu_2(b_1 - b_2) = 0 \\
\frac{\partial L}{\partial m_{20}^*} &= [c_1 p + a_2(1-p)] + 2\lambda_5 m_{20}^* - \lambda_5 - \nu_1(c_1 - a_2) = 0 \\
\frac{\partial L}{\partial m_{21}^*} &= [c_1 p + a_2(1-p)] + 2\lambda_6 m_{21}^* - \lambda_6 + \nu_2(c_1 - a_2) = 0
\end{aligned}$$

Appendix B

Minimum Rate of Doubly Symmetric Binary Source with the Distortion d

Two sequences X^n and Y^n are generated from two binary sources X and Y by a crossover probability p, which is in the range of $[0, \frac{1}{2}]$. X^n is encoded to an index of length $nR(d)$. Y^n is sent to the decoder directly as the side information. The minimum rate $R(d)$ [3] with distortion d is found as:

$$R(d) = \begin{cases} g(d) & \text{if } d < d_c \\ \frac{g(d_c)(p-d)}{p-d_c} & \text{if } d_c \le d \le p \\ 0 & \text{if } d \ge p \end{cases} \tag{B.1}$$

where

$$g(d) = h(p * d) - h(d), 0 \le d \le p \tag{B.2}$$

where $h(k) = -k \log k - (1-k)\log(1-k)$, $0 \le k \le 1$ for $0 \le u, v \le 1$

$$p * d = p(1-d) + d(1-p) \tag{B.3}$$

d_c is solved by the following equation.

$$\frac{g(d_c)}{d_c - p} = g'(d_c) \tag{B.4}$$

Bibliography

[1] T. Cover and J. Thomas, *Elements of information theory.* wiley, 2006.

[2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[3] A. Wyner, "The rate-distortion function for source coding with side information at the decoder\ 3-II: General sources," *Information and Control*, vol. 38, no. 1, pp. 60–80, 1978.

[4] Z. Sun, "Applications of low density graph codes in two source coding problems," Master's thesis, McMaster University, 2009.

[5] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoderusing ldpc codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, 2002.

[6] R. Gallager, "Low-density parity-check codes," *Information Theory, IRE Transactions on*, vol. 8, no. 1, pp. 21–28, 1962.

[7] T. Berger and R. Yeung, "Multiterminal source encoding with encoder breakdown," *IEEE Transactions on Information Theory*, vol. 35, no. 2, pp. 237–244, 1989.

[8] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge Univ Pr, 2004.

[9] A. Amraoui and R. Urbanke, "Ldpcopt." http://ipgdemos.epfl.ch/ldpcopt/index.php, 2009.

[10] Q. Wang and C. He, "Practical dirty paper coding with nested binary ldgm-ldpc codes." 2009.

[11] S. Chung, *On the construction of some capacity-approaching coding schemes.* PhD thesis, Citeseer, 2000.

[12] S. Chung, G. Forney Jr, T. Richardson, R. Urbanke, A. Inc, and M. Chelmsford, "On the design of low-density parity-check codes within 0.0045 dB ofthe Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, 2001.