

Applications of Low Density Graph Codes in Two
Source Coding Problems

APPLICATIONS OF LOW DENSITY GRAPH CODES
IN TWO SOURCE CODING PROBLEMS

By

ZHIBIN SUN, B.Eng., (Electrical Engineering)
McMaster University, Hamilton, ON., Canada

A Thesis

Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Applied Science

McMaster University

April 2009

Dedications:

To my family:

Jiayi Hou, Jimei Wang, Jianmin Sun and Qingyun Li

Abstract

In this thesis, we present the applications of low density graph codes in two different types of source coding problems. First, we consider asynchronous Slepian-Wolf coding where the two encoders may not have completely accurate timing information to synchronize their individual block code boundaries, and propose LDPC design in this scenario. A new information-theoretic coding scheme based on source splitting is provided, which can achieve the entire asynchronous Slepian-Wolf rate region. Unlike existing methods based on source splitting, the proposed scheme does not require common randomness at the encoder and the decoder, or the construction of super-letter from several individual symbols. We then design LDPC codes based on this new scheme, by applying the recently discovered source-channel code correspondence. Second, we consider the lossy source coding problem. In contrast with most prior work that has focused exclusively on the binary uniformly distributed source, we address the problem of lossy coding for sources with arbitrary alphabets and distributions. Built upon the idea of approximating the optimal output distribution indicated by the rate-distortion theory with a uniform distribution over a larger alphabet, we propose a multilevel coding scheme using LDGM codes that can approach the rate-distortion limit for a general source. Experimental results validate the effectiveness of both proposed methods.

Acknowledgements

In the first place I would like to express my deep appreciation to my supervisors, Dr. Jun Chen and Dr. Kon Max Wong, for their expert supervision, guidance and encouragement throughout the course of this thesis, which have been illuminating the way of my independent research.

I am very grateful to Dr. Chao Tian at AT&T Research Lab for continuously providing helpful suggestions and insightful discussions.

I want to thank Cheryl, graduate administrative assistant, and and Cosmin, research computing specialist, for their help during my study.

I also would like to thank my group members, Mingkai Shao, Min Huang and Ying Zhang, and all other friends in the department, Wenzhe Zhang, Huazhong Wang and etc, for their friendship and support which played an important role in my completing this project.

Last but not least, I am also deeply indebted to all my family members for their understanding, love and unceasing faith on me along my way.

Contents

Abstract	iii
Acknowledgements	iv
List of Figures	ix
Acronyms	x
Notations	xi
1 Introduction	1
1.1 Background	1
1.1.1 Slepian-Wolf Coding	1
1.1.2 Lossy Source Coding	3
1.2 Motivation and Contribution of the Thesis	4
1.3 Organization of the Thesis	5
2 Low Density Graph Codes and Message Passing Algorithm	6
2.1 Low Density Graph Codes	6
2.1.1 Low Density Parity Check Codes	7
2.1.2 Low Density Generator Matrix Codes	8
2.2 Message Passing Algorithm	9

2.2.1	Belief Propagation Algorithm	9
2.2.2	Survey Propagation Algorithm	11
3	Asynchronous Slepian-Wolf Coding Using LDPC Codes	12
3.1	Review of Previous Work on S-SW Coding	12
3.2	A New Source Splitting Scheme for A-SW Coding	14
3.2.1	Time-sharing and Source Splitting With Common Randomness	14
3.2.2	A New Source Splitting Scheme Without Common Randomness	17
3.2.3	Overview of the LDPC Design	21
3.3	Equivalent Channel Model and Code Design in the A-SW Setting . .	22
3.3.1	Source Coding With Decoder Side Information	23
3.3.2	Source Coding With Partial Decoder Side Information	25
3.4	Design Examples and Numerical Results	26
3.4.1	Example of Source Coding With Decoder Side Information . .	27
3.4.2	Example of Source Coding With Partial Decoder Side Information	29
3.4.3	Overall Code Performance	32
4	Lossy Source Coding for General Source Using LDGM Codes	36
4.1	Extend the Existing Method to Non-uniform Sources	36
4.2	Multilevel Coding	38
4.2.1	Basic Coding Scheme	38
4.2.2	A More Generalized Coding Scheme	40
4.2.3	Finding the Optimal Output Distribution	42
4.3	Message Passing Algorithm	42
4.3.1	Non-Uniform Binary Source with Hamming Distance	47
4.3.2	Uniform Ternary Source with Hamming Distance	50
4.3.3	Uniform Ternary Source with Non-Hamming Distance	52

4.4	Simulation Result	55
5	Conclusion and Future Work	59
5.1	Conclusion	59
5.2	Future Work	60
A	Proof of the sufficiency of linear codes in the A-SW setting	61
B	Proof of the channel correspondence to the Slepian-Wolf problem with side information	66
C	Discretized density evolution algorithm and linear programming	69
D	Calculate the channel transition probability in the dual channel model	72
	Bibliography	78

List of Figures

1.1	Slepian-Wolf coding problem	1
1.2	Achievable rate region of Slepian-Wolf coding problem	2
2.1	Factor graph of LDPC code	8
2.2	Factor graph of LDGM code	9
3.1	The difficulty of time-sharing in the asynchronous setting.	15
3.2	Illustration of the new scheme without common randomness	19
3.3	The equivalent channel for source coding with decoder side information.	24
3.4	The performances of 8 irregular LDPC codes of length 2×10^5 , with side information Y at the decoder.	29
3.5	The performances of 6 irregular LDPC codes of length 2×10^5 , with partial X side information at the decoder.	31
4.1	Performance degradation for non-uniform sources.	37
4.2	Approximating a binary output distribution with $P_{\hat{X}}(0) = 0.25$ by two uniformly distributed binary random variables.	40
4.3	Using a single LDGM code of extended length.	41
4.4	Survey propagation equations of check node and variable node.	45

4.5	survey propagation equations of source node and network node. . . .	46
4.6	The deterministic mapping for optimal output distribution $P_{\hat{X}}^*(0) = \frac{3}{16}$ and $P_{\hat{X}}^*(1) = \frac{13}{16}$	49
4.7	The deterministic mapping for optimal output distribution $P_{\hat{X}}^*(0) = \frac{5}{16}$, $P_{\hat{X}}^*(1) = \frac{5}{16}$ and $P_{\hat{X}}^*(2) = \frac{6}{16}$	51
4.8	The deterministic mapping for optimal output distribution $P_{\hat{X}}^*(0) = \frac{4}{16}$, $P_{\hat{X}}^*(1) = \frac{8}{16}$ and $P_{\hat{X}}^*(2) = \frac{4}{16}$	54
4.9	Lossy source coding of non-uniform binary source with hamming distance measure.	56
4.10	Lossy source coding of uniform ternary source with hamming distance measure.	57
4.11	Lossy source coding of uniform ternary source with non-hamming distance measure.	58

Acronyms

LDPC	Low Density Parity Check
LDGM	Low Density Generator Matrix
A-SW	Asynchronous Slepian-Wolf Problem
S-SW	Synchronous Slepian-Wolf Problem
LLR	Log Likelihood Ratio
SID	Survey Inspired Decimation
DE	Density Evolution Algorithm
BER	Bit Error Rate
AWGN	Additive White Gaussian Noise
R	Rate
C	Capacity
pmf	Probability Mass Function

Notations

\mathbb{C}	Code ensemble
H	Parity check matrix
G	Generator matrix
R	rate
\mathcal{X}	Alphabet of X
$ \mathcal{X} $	Size of alphabet \mathcal{X}
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
$[\cdot]_n$	modulo- n operation
\mathcal{A}	Set \mathcal{A}
\mathcal{A}^c	Complement set of set \mathcal{A}
\mathcal{A}/a	Set \mathcal{A} exclude a
P_e	Probability of error
\log	Logarithm function with base e
$\rho(x)$	Check node degree polynomial
$\lambda(x)$	Variable node degree polynomial

$P_{\hat{X}}^*$	The optimal output distribution calculated by the rate-distortion function
$GF(K)$	Galois field of K elements
$GF^i(K)$	Product of i Galois fields each with K elements
C_i	i -th check node
N_i	i -th network node
S_i	i -th source node
V_i	i -th variable node
$A_c(k)$	The set of check nodes connected to variable node k
$B_v(j)$	The set of variable nodes connected to check node j
$M_{V_k \rightarrow C_j}$	The messages from variable node k to check node j
$M_{C_j \rightarrow V_k}$	The message from check node j to variable node k
M_{S_l}	The message from source node l to network node l
$M_{N_l^i}$	The message from network node l to the l -th check node in i -th group
XOR	Exclusive or operation

Chapter 1

Introduction

1.1 Background

In this section we provide a review of Slepian-Wolf coding and lossy source coding problems as background materials.

1.1.1 Slepian-Wolf Coding

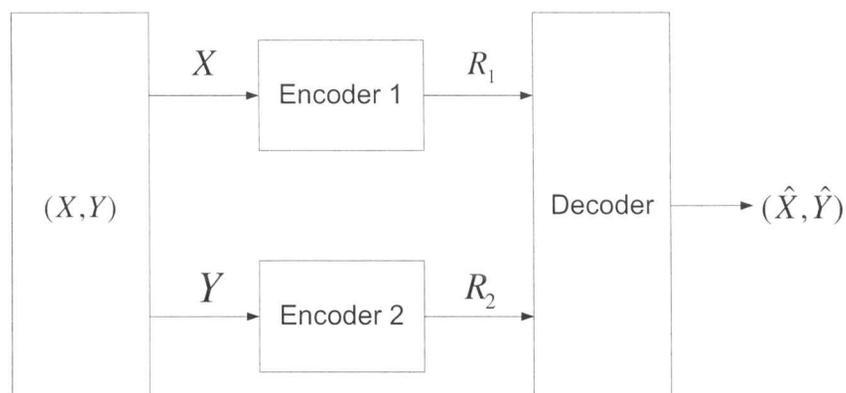


Figure 1.1: Slepian-Wolf coding problem

As illustrated in Figure 1.1, two correlated sources (X, Y) are encoded separately at two encoders and decoded together at one joint decoder. It is known that to encode a source X , R must be greater or equal to the entropy $H(X)$. Therefore, to encode X and Y separately we need $R = R_x + R_y \geq H(X) + H(Y)$. However, in the seminal work [4], Slepian and Wolf showed that it is possible to compress two dependent sources in a distributed manner at rates no larger than those needed when they are compressed jointly. More precisely, when two discrete memoryless sources X and Y jointly distributed as Q_{XY} in the alphabets \mathcal{X} and \mathcal{Y} are separately compressed using block codes of length- n at rates R_1 and R_2 respectively, both sources can be reconstructed with asymptotically diminishing error probability at a central decoder using any rates (R_1, R_2) such that

$$R_1 > H(X|Y), R_2 > H(Y|X), R_1 + R_2 > H(X, Y). \quad (1.1)$$

This rate region is illustrated in Figure 1.2. This result has been generalized in various

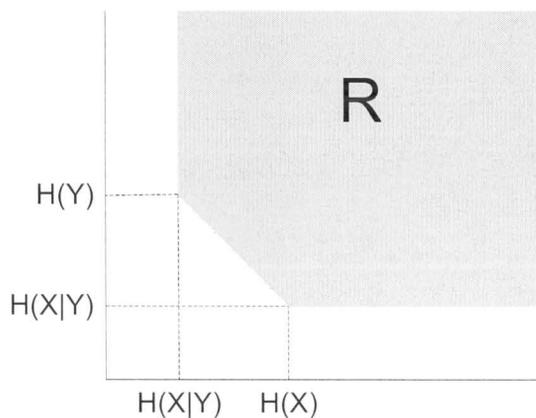


Figure 1.2: Achievable rate region of Slepian-Wolf coding problem

ways [5, 6, 7, 8], one of which is the asynchronous Slepian-Wolf (A-SW) coding scenario considered by Willems [7]. This consideration is practically important, because

although we can assume perfect synchronization in the simplistic Slepian-Wolf setting such that the length- n block codes can be applied on each corresponding block pair $\{X(t), Y(t)\}_{t=qn+1, \dots, (q+1)n}$, in practice the two encoders may not have such a perfectly accurate global clock to synchronize their block code boundaries. It was shown in [7] that even in this case, the rate region result (1.1) still holds, as long as the decoder is aware of this asynchronism.

1.1.2 Lossy Source Coding

In the lossy source coding problem, we are given a source alphabet \mathcal{X} , distribution P_X and a distortion metric $d(\cdot, \cdot)$. A source sequence x^n is drawn from the i.i.d distribution P_X , where n is the message length. We want to map each sequence x^n to a codeword in a code book containing 2^m codewords. First, one need to map the n -bits source sequence x^n to an m -bits codeword index z^m , thus, the encoding rate $R = m/n$. The codeword \hat{x}^n that corresponds to the index z^m is the decoded sequence or called reconstructed sequence. The lossy source coding problem is to find the encoding mapping and the code book satisfying the rate constraint R such that the average distortion $D = \mathbb{E}[\frac{1}{n} \sum_{i=1}^n d(x_i, \hat{x}_i)]$ is minimized.

The trade-off between the encoding rate R and the achievable minimal distortion has been proved by Shannon [3] and is expressed in the rate-distortion function as follows:

$$R(D) = \min_{P_{\hat{X}|X}} I(X; \hat{X}), \quad (1.2)$$

where the minimization is over all possible test channels $P_{\hat{X}|X}$ subject to the constraint $\mathbb{E}[d(X, \hat{X})] \leq D$.

1.2 Motivation and Contribution of the Thesis

Recently the applications of low density graph codes associated with various message passing algorithms to source coding problems have shown extremely good performances [13, 21, 22, 37]. However, these codes and the associated algorithms are not consummated in the sense that they only work under certain conditions such as the following two cases. In the recent researches of the Slepian-Wolf problem, LDPC codes have been applied. However, most code designs focus on the corner point of the achievable rate region with the joint distributed source that can be modeled as a binary symmetric channel [10, 11, 12, 13]. Moreover, the practical coding scheme and code design method are missing for the asynchronous case. The similar restriction can be seen in the researches of the lossy source coding problem in which the LDGM codes are adopted. For lossy source coding using LDGM codes, almost all the existing results focus on the binary uniform source [21, 22], and the application of the existing coding algorithm directly to binary nonuniform source incurs significant degradation. The restrictions on these source coding problems motivate us to develop a code design method and encoding algorithm that can be applied in a more general setting. In this thesis, we discuss two aforementioned source coding problems and focus on general sources. The contribution of our work is that we propose a LDPC code design method and an encoding scheme that can achieve the general rate point on the Slepian-Wolf rate region with arbitrary joint source distribution under both synchronous and asynchronous scenarios. For the lossy source coding problem, we propose a multilevel coding scheme using LDGM codes that can approach the rate-distortion limit of a general source.

1.3 Organization of the Thesis

The thesis is structured as follows:

- In Chapter 2, we present two low density graph codes, LDPC codes and LDGM codes as well as two message passing algorithms, BP and SP algorithms.
- In Chapter 3, we focus on the A-SW problem. A new information-theoretic coding scheme based on source splitting is provided. We design LDPC codes based on this new scheme, by applying the recently discovered source-channel code correspondence. Simulation results are given.
- In Chapter 4, we focus on the lossy source coding problem. A multilevel coding scheme using LDGM codes for the general source and distortion measure is proposed based on the idea of approximating the optimal output distribution indicated by the rate-distortion theory with a uniform distribution over a larger alphabet. Simulation results are given.
- In Chapter 5, we conclude this thesis and suggest the future work.

Chapter 2

Low Density Graph Codes and Message Passing Algorithm

Recently the applications of low density graph codes associated with various message passing algorithms to source coding problems such as the application of LDPC codes to the Slepian-Wolf problem [13, 37] and the application of LDGM codes to the lossy source coding problem [21, 22] have shown extremely good performances. In this chapter, we will briefly describe these two low density graph codes, LDPC and LDGM codes, as well as the message passing algorithms associated with them.

2.1 Low Density Graph Codes

Low density graph codes such as low density parity check (LDPC) codes and low density generator matrix (LDGM) codes refer to the codes that can be represented by graphical constructions. The term “low density” indicates the sparsity of the graph construction that makes the message passing algorithm efficient for decoding such a code.

2.1.1 Low Density Parity Check Codes

Low density parity check (LDPC) codes were invented by Gallager in 1960's [1]. A binary LDPC code is defined as follows:

$$\mathbb{C} = \{x \in \{0, 1\}^n : Hx = 0\}, \quad (2.1)$$

where H is an $m \times n$ sparse matrix called parity check matrix, and the code is given by its null space. The term “sparse” here means the number of 1 in the matrix H is sparse. Bipartite graphs also called factor graphs was first suggested by Tanner [2] to capture the LDPC code structure. The bipartite graph in Figure 2.1 represents the code structure of an irregular LDPC code. Each variable nodes (O) represents one bit in the codeword and each check node (\square) represent the parity check constraint specified by one row of H . If the entry (i, j) of parity check matrix H is 1, the i -th check node and j -th variable node in the factor graph is connected by an edge. These two connected nodes are called adjacent nodes. The rate of LDPC code is $r = 1 - m/n$, where m and n are the number of check nodes and variable nodes respectively ($n > m$). The degree distributions of check nodes and variable nodes in the factor graph are represented by ρ and λ , respectively.

Define $\rho(x) = \sum_{i=1}^{d_c} \rho_i x^{i-1}$ and $\lambda(x) = \sum_{i=1}^{d_v} \lambda_i x^{i-1}$, where ρ_i and λ_i denote the portion of all edges connected to check nodes and variable nodes with degree i , respectively. The degree of a check or variable node is defined as the number of edges connected to this node. We use d_c to denote the maximum check node degree and d_v to denote the maximum variable node degree.

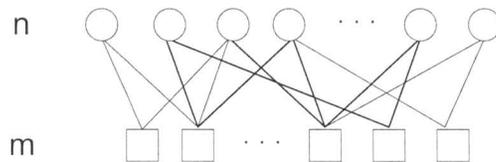


Figure 2.1: Factor graph of LDPC code

2.1.2 Low Density Generator Matrix Codes

Low density generator matrix (LDGM) codes are dual of LDPC codes. A binary LDGM code is defined as follows:

$$\mathbb{C} = \{x \in \{0, 1\}^n : x = G \cdot z\}. \quad (2.2)$$

G is an $n \times m$ sparse matrix called generator matrix, and the codeword $x \in \{0, 1\}^n$ is given by the product of G and information bits $z \in \{0, 1\}^m$. The term "sparse" here means the number of 1 in the matrix G is sparse. Bipartite graphs or factor graphs are also used to capture the code structure of LDGM codes. The bipartite graph in Figure 2.2 represents the code structure of an irregular LDGM code. Each variable nodes (O) represents one information bit and each check node (\square) represent the generation constraint specified by one row of G . If the entry (i,j) of generator matrix G is 1, the i -th check node and j -th variable node in the factor graph is connected by an edge. These two connected nodes are also called adjacent nodes. Different from LDPC codes, the rate of LDGM code is $r = m/n$, where m and n are the number of variable nodes and check nodes respectively ($n > m$). Same as LDPC codes the degree distributions of check nodes and variable nodes in the factor graph are also represented by ρ and λ , respectively.

Define $\rho(x) = \sum_{i=1}^{d_c} \rho_i x^{i-1}$ and $\lambda(x) = \sum_{i=1}^{d_v} \lambda_i x^{i-1}$, where ρ_i and λ_i denote the portion of all edges connected to check nodes and variable nodes with degree i , respectively.

The degree of a check or variable node is defined as the number of edges connected to

this node. d_c denotes the maximum check node degree and d_v denotes the maximum variable node degree.

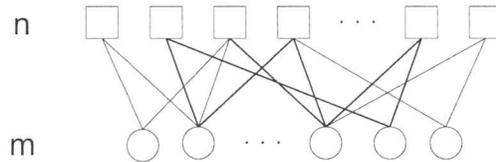


Figure 2.2: Factor graph of LDGM code

2.2 Message Passing Algorithm

Message passing algorithms are the general decoding algorithms associated with the low density graph codes. Based on the factor graph of each low density graph code, these algorithms can be summarized as that at each iteration of the algorithm messages are passed from the check nodes to the adjacent variables and also from variable nodes to the adjacent check nodes. Therefore, the message passing algorithms are also iterative algorithms. The message in terms of probabilities or log-likelihood from either check node or variable node is calculated based on the messages passed to this node from the adjacent nodes in previous iteration. In the following, two important message passing algorithms, belief propagation (BP) algorithm and survey propagation (SP) algorithm, are presented.

2.2.1 Belief Propagation Algorithm

BP algorithm is an important message passing algorithm which is presented by Gallager [1] and has been proved to have promising performance when decoding the LDPC codes in channel coding problems [34] and Slepian-Wolf problem [13].

In belief propagation algorithm, the messages are called beliefs and take the form of log-likelihood ratio (LLR). How this algorithm applies to the LDPC codes when solving the Slepian-Wolf problem with side information is briefly discussed in the following:

Sequences x^n and y^n of length n are drawn by the i.i.d joint distribution $P_{XY}(x, y)$. The syndrome sequence s^m of length m are given by $H_{m \times n} \cdot x^n$. The algorithm aim to decode sequence x^n given sequence y^n and syndrome s^m .

The LLR of X based on the observation of Y is calculated by

$$L = \log \frac{P(x_i = 0|y_i)}{P(x_i = 1|y_i)} \quad (2.3)$$

At each iteration the messages are passed from variable nodes to the adjacent check nodes first. The message sent from i -th variable node to j -th check node is calculated by

$$M_{V_i \rightarrow C_j} = L + \sum_{l \in A_c(i) \setminus C_j} M_{C_l \rightarrow V_i} \quad (2.4)$$

where $A_c(i)$ is the set of all check nodes connected to variable node i . The initial value of the message from check node to variable node is set to zero.

Then the messages are passed from check nodes to the adjacent variable nodes. The message from i -th check node to j -th variable node is calculated by

$$M_{C_i \rightarrow V_j} = 2 \tanh^{-1} \left[(1 - 2s_i) \prod_{l \in B_v(i) \setminus V_j} \tanh\left(\frac{M_{V_l \rightarrow C_i}}{2}\right) \right] \quad (2.5)$$

Once the algorithm runs to the certain number of iterations, the estimated x_i is found at each variable node by

$$\hat{x}_i = \begin{cases} 0 & \text{if } L + \sum_{l \in A_c(i)} M_{C_l \rightarrow V_i} \geq 0 \\ 1 & \text{if } L + \sum_{l \in A_c(i)} M_{C_l \rightarrow V_i} < 0 \end{cases} \quad (2.6)$$

The sequence of \hat{x}^n found from n variable nodes is the decoded sequence.

2.2.2 Survey Propagation Algorithm

Similar to the BP algorithm, SP algorithm is also a kind of message passing algorithm as well as iterative algorithm. However, it is always associated with LDGM codes. Initially it is designed to solve the satisfiability problem and has shown good performance [20]. Recently, survey propagation algorithm associated with LDGM codes has shown to be effective for lossy source coding problem [21, 22].

This algorithm resembles to BP in message passing behavior. However, the message or survey in SP algorithm is a triplet of probabilities and denoted as (M^0, M^1, M^*) in the binary case. Each element in this triplet message represents the marginal probability that the destination node is forced to be 0, 1 and free state respectively. Moreover, the messages in SP algorithm will converge after a number of iterations and a new procedure called "Survey Inspired Decimation" (SID) is introduced. When all the messages that pass along the edges converge or the number of message passing iterations reaches a threshold value, SID starts. In SID, the bias value of each variable node is computed to indicate the tendency of this node to be zero or one. Then compare these bias values with a certain threshold and fix the value of those variable nodes whose bias values are greater than the threshold value or fix the value of the variable node whose bias value is the largest if no bias value is greater than the threshold value. Remove the fixed variable nodes from the factor graph and update the source values associated with each check node. Keep on doing until all the variable node values are fixed. The sequence constructed by all values of the variable nodes in a sequential order is the encoded sequence. The decoded or reconstructed sequence can be obtained by multiplying the encoded sequence with the generator matrix G .

Chapter 3

Asynchronous Slepian-Wolf Coding Using LDPC Codes

In this chapter, we consider the code design problem in the A-SW setting. Formally, the encoders are associated with two integer-valued delay parameters d_x and d_y , respectively, in the range of 0 to $n - 1$. The values of d_x and d_y are unknown to the encoders but known to the decoder. Thus the q -th block of the source X consists of $\{X(t)\}_{t=qn+1+d_x, \dots, (q+1)n+d_x}$ and the q -th block of the source Y consists of $\{Y(t)\}_{t=qn+1+d_y, \dots, (q+1)n+d_y}$. It is easy to see that for the two corner points of the achievable rate region, this problem is not very different from that in the synchronous case. Thus our focus is on the general rate pairs, or more precisely, rate pairs on the dominant face of the rate region.

3.1 Review of Previous Work on S-SW Coding

Previous works on the synchronous Slepian-Wolf (S-SW) code design mostly focus on the corner points of the rate region [10, 11, 12, 13], with only a few exceptions [14][15].

The code design at the corner points is reasonably well understood, particularly with the recent development [16], where it was shown that when encoding Y^n with side information X^n at the decoder, the error probability of any single linear coset code is exactly the decoding error probability of the same channel code on a corresponding channel under optimal decoding or belief propagation decoding. Thus the linear Slepian-Wolf code design problem at a corner point can be conveniently converted into the code design problem for a specific channel. Though this connection was mentioned in earlier works [17, 18], it was made precise in [16] for general (non-symmetric non-binary) sources.

The code design for the S-SW problem can not be applied directly to the asynchronous case. One may wonder if the information-theoretic coding scheme given in [7] can be used for such a purpose, however, it unfortunately requires optimization of the auxiliary random variable and complex joint typicality encoding usually seen in quantization modules, thus not convenient for practical code design. The usual time-sharing approach to achieve general rate pairs in S-SW coding also does not apply in the asynchronous setting. In [19], an information-theoretic scheme based on source splitting was given to overcome this difficulty due to asynchronism, by introducing common randomness at one encoder and the decoder. However, common randomness is not desirable in practical systems and should be avoided if possible.

It is clear that a new coding approach is needed for the A-SW code design, and since Slepian-Wolf coding is well understood for the corner points of the rate region, it is also desirable to utilize these existing results. Indeed, in this thesis we first present an information-theoretic scheme based on source splitting, which does not require common randomness (or the construction of a threshold function which operates on multiple source symbols as a super-letter [15]). Then based on this coding scheme, we utilize the source-channel correspondence result in [16] to design good LDPC codes. Experimental results confirm the effectiveness of the proposed design.

It should be clarified at this point that the asynchronism here only refers to the mismatch in the block code boundaries of the two encoders, but not the timing mismatch in the sampling process. When sampling a continuous process, such timing inaccuracy may incur uncertainty in the probability distribution P_{XY} , and this problem is usually considered in the framework of universal Slepian-Wolf compression [8][9], thus is beyond the scope of this work.

3.2 A New Source Splitting Scheme for A-SW Coding

In this section, we first briefly explain the difficulty of using time-sharing in the asynchronous setting and then review how source splitting together with common randomness can be used to overcome this difficulty, as proposed in [19]. Then a new information-theoretic scheme based on source splitting is proposed, which does not require common randomness. An overview of the proposed LDPC design in the context of this new scheme will be subsequently discussed.

3.2.1 Time-sharing and Source Splitting With Common Randomness

It is easy to see that a simple time-sharing can be used to achieve any general rate pair in the achievable rate region for the S-SW problem. Essentially we only need to use the following two kinds of coding alternately: 1) first encode X directly, then encode Y with X as decoder side information, and 2) first encode Y directly, then encode X with Y as decoder side information. When the first kind of code is used with a proportion p , and the second is used with proportion $1 - p$, it is clear that the

following rate pair is achieved.

$$R_1 = pH(X) + (1-p)H(X|Y), \quad R_2 = pH(Y|X) + (1-p)H(Y). \quad (3.1)$$

An example makes clear the difficulty of using this approach in the asynchronous setting. Let us assume $p = 0.5$ and the two kinds of codes used in timesharing both are of length- n . Thus in the synchronous setting, the first kind of code is used in the even block, i.e., the $(2m)$ -th block, and the second kind of code is used in the odd block, i.e., the $(2m + 1)$ -th block. Now consider the asynchronous setting, and let $d_x = \frac{1}{2}n$ and $d_y = 0$, which are unknown to the encoders but known to the joint decoder. When the original time-sharing codes are used, it is clear that at the first half of the even blocks, source Y is encoded assuming X at the decoder, yet the source X is also encoded assuming Y at the decoder. This clearly results in decoder failure. Another way to see this is that in this portion, the sources are encoded with sum rate $H(Y|X) + H(X|Y)$, which is less than $H(X, Y)$, and thus it is impossible to ensure reliable communication; see Figure 3.1 for an illustration.

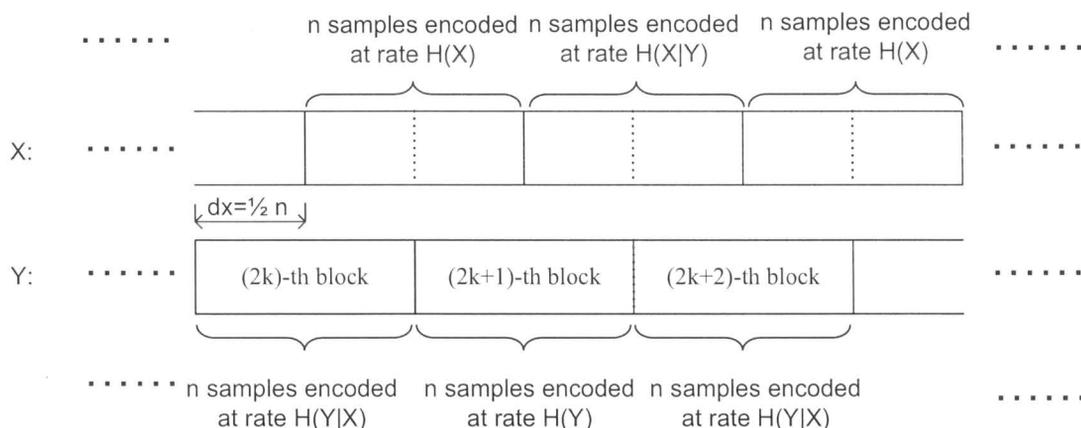


Figure 3.1: The difficulty of time-sharing in the asynchronous setting.

To overcome this difficulty, common randomness was introduced in [19] such that

source splitting can be applied. Let $\{T(t)\}_{t=1,\dots,\infty}$ be a binary discrete memoryless process distributed in $\{0,1\}$ such that at each time instance, $\Pr[T(t) = 1] = p$; it is independent of the sources (X, Y) . Let this process be available at both the encoder observing X and the decoder. Without loss of generality, let us assume $\mathcal{X} = \{1, 2, \dots, |\mathcal{X}|\}$. Now define two new sources

$$Z = X \cdot T, \quad W = X \cdot (1 - T). \quad (3.2)$$

In other words, $\{Z(t)\}_{t=1,\dots,\infty}$ is $\{X(t)\}_{t=1,\dots,\infty}$ with certain position assigned to zero, while $\{W(t)\}_{t=1,\dots,\infty}$ is $\{X(t)\}_{t=1,\dots,\infty}$ with the complement positions assigned to zero. Thus the source X is split into two sources Z and W .

With these two new sources, the original problem is transformed into Slepian-Wolf coding of sources (Z, W) and Y . The encoding can now be performed in a sequential order as follows:

1. Encode the source Z conditioned on T ;
2. Encode source Y assuming Z at the decoder;
3. Encode W assuming (Z, Y) at the decoder.

The rates at the two encoders are

$$R_1 = H(Z|T) + H(W|Z, Y), \quad R_2 = H(Y|Z). \quad (3.3)$$

Observe that

$$R_1 + R_2 = H(Z|T) + H(W|Z, Y) + H(Y|Z) = H(X, Y|T) = H(X, Y). \quad (3.4)$$

The rate pair in Eq. (3.3) is on the dominant face of the achievable rate region. By varying p from 0 to 1, all the rate pairs on the dominant face can be achieved, at least for the S-SW case.

A moment of thought should convince the readers that by applying the above block codes consecutively, the coding scheme can also be used in the asynchronous case without any change. Although this scheme can indeed achieve all rate pairs in the A-SW setting, it requires common randomness at one encoder and the decoder, which is not desirable in practice. Next we propose a new scheme which does not require this common randomness.

3.2.2 A New Source Splitting Scheme Without Common Randomness

From the source splitting scheme with common randomness afore-given, we can make the following two observations:

1. In the second coding step, for each length- n block of source Y , there are approximately $p \cdot n$ source X samples available at the decoder; furthermore, the exact positions of these X samples are known at the decoder.
2. Though the random sequence $T(t)$ can potentially split $X(t)$ at each time instance into two random variables $Z(t)$ and $W(t)$, the overall effect is in fact to split the length- n source X sequence in the time domain, such that the first item above can be made true.

Based on these two observations, we propose the following information-theoretic scheme, which does not require common randomness. Instead of the random sequence $T(t)$, let us consider a deterministic one

$$T(t) = \begin{cases} 1 & [t]_n - d_x = 1, \dots, k \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

where $\lfloor \cdot \rfloor_n$ is modulo- n operation. In other words, the first k positions in a block aligned with the length- n source X block are assigned 1, while the remaining $n - k$ positions are assigned 0. After we apply the first coding step in the original source splitting scheme, a partial X sequence is available at the decoder. Such a choice of $T(t)$ indeed approximately satisfies the first observation discussed above, regardless the exact value of d_x ; see Figure 3.2 for an illustration. At this point, the second coding step in the original source splitting scheme still can not be directly applied, since now there is no explicit source Z to simplify the coding module.

We can now focus on the second coding step of encoding source Y using a length- n block code, where k out of n of the corresponding X source samples are available at the decoder, whose positions are unknown to the encoder, but known to the decoder. These k positions of source X samples can be from a single length- n source X coding block, or two separate length- n source X coding blocks, which are illustrated in Figure 3.2(a) and Figure 3.2(b), respectively.

The following (random) coding scheme for the second step is the key difference between the one in [19] and the one proposed in this work. For convenience, let us denote the set of positions (indexed) within this length- n block for which source X samples are available at the decoder as \mathcal{A} , where $\mathcal{A} \subseteq \{1, 2, \dots, n\}$. Consequently the set of the remaining positions within this block is denoted as \mathcal{A}^c .

- Random binning: each of y^n sequence is uniformly and independently assigned to one of 2^{nR_2} bin indices;
- Encoding: the encoder sends the bin index of the Y source sequence;
- Decoding: if the known length- k source X sequence from \mathcal{A} is δ_1 -typical, we then find a length- n source Y sequence in the corresponding bin, such that the following two typicality conditions hold (i) the length- k vector by collecting the Y samples at the positions corresponding to \mathcal{A} is jointly δ_1 -typical with the

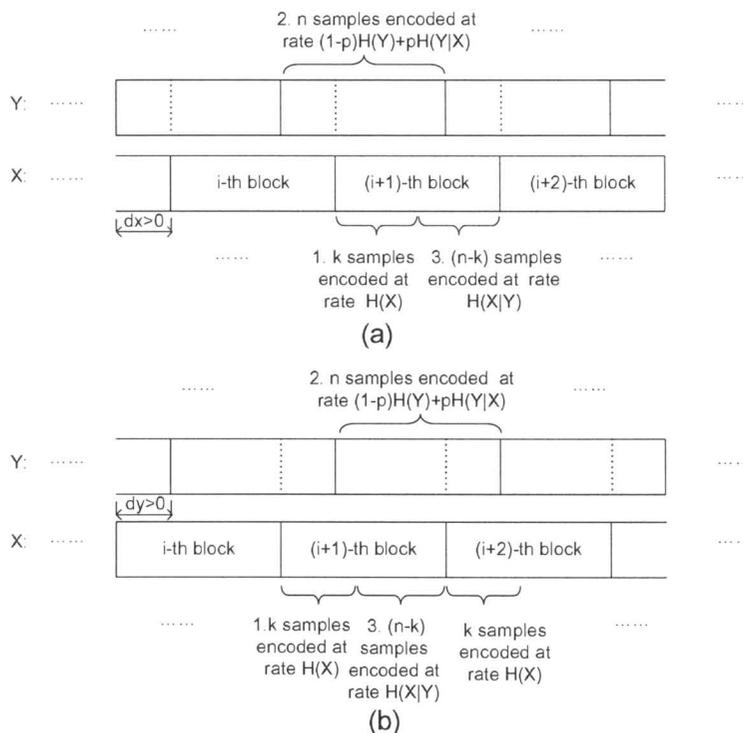


Figure 3.2: Illustration of the new scheme without common randomness

known length- k X sequence, and (ii) the length- $(n - k)$ vector by collecting the Y samples at the positions corresponding to \mathcal{A}^c is δ_2 -typical by itself.

In the above procedure, we have used the (weak) typicality definition in [36] (pp. 51 and pp. 384-385). It is easier to bound the decoding error if we view a single length- n source Y sequence as the combination of a length- k and a length- $(n - k)$ sequence. When both k and $n - k$ are sufficiently large, it is clear that with high probability, the original source sequence y^n indeed satisfy the two typicality conditions with high probability; we shall denote the probability of such error event that a source sequence fails one of the two conditions as P'_e . We only need to bound the probability that another Y sequence is decoded instead of the correct one. By the well-known

properties of the typical sequences (see [36] pp. 51-53 and pp. 385-387), we see there are less than $2^{k(H(Y|X)+2\delta_1)}$ length- k source Y sequences that are jointly δ_1 -typical with the known δ_1 -typical length- k source X , and there are less than $2^{(n-k)(H(Y)+\delta_2)}$ δ_2 -typical length- $(n-k)$ source Y sequences. Thus the decoding error in this step is bounded by

$$\begin{aligned} P_e &\leq P'_e + 2^{-nR_2} 2^{k(H(Y|X)+2\delta_1)} 2^{(n-k)(H(Y)+\delta_2)} + \delta_1 \\ &= P'_e + 2^{-n[R_2 - \frac{k}{n}(H(Y|X)+2\delta_1) - \frac{n-k}{n}(H(Y)+\delta_2)]} + \delta_1, \end{aligned} \quad (3.6)$$

where the last term δ_1 accounts for the error event that the known X sample sequence at the decoder is in fact not typical. Thus if we choose sufficiently small δ_1 and δ_2 , as long as

$$R_2 > \frac{n-k}{n}H(Y) + \frac{k}{n}H(Y|X) \approx (1-p)H(Y) + pH(Y|X),$$

where $p = \frac{k}{n}$, the decoding error vanishes as $n \rightarrow \infty$. This solves the second step in the source splitting scheme without any common randomness. It should be noted that the scheme inherently requires both k and $n-k$ to be large.

For the third coding step, the decoding may have to wait until the next Y block is decoded; see Figure 3.2(a) for an illustration. Nevertheless, it is not difficult to see that the third coding step in the original source splitting scheme can be used without much change. Thus we only need

$$R_1 > \frac{k}{n}H(X) + \frac{n-k}{n}H(X|Y) \approx pH(X) + (1-p)H(X|Y). \quad (3.7)$$

where $p = \frac{k}{n}$. For sufficiently large n , by adjusting k , all rate pair on the dominant face of the Slepian-Wolf rate region can be effectively approached.

We have shown that for a fixed set \mathcal{A} , there indeed exists a sequence of codes that can approach the Slepian-Wolf limit. However, one key requirement in the A-SW problem is that a single code has to guarantee small error probability for all the

possible cyclic shifts of \mathcal{A} induced by the asynchronism. By refining the probability bound for decoding error given above, we can indeed show that such a sequence of codes exists. However instead of proving this more technical result here, in Theorem 3.1 we provide an even stronger and relevant result that linear codes under the same source splitting paradigm can achieve the A-SW limit. Appendix A shows the proof of this theorem. This serves as a more rigorous proof, as well as the theoretical basis for designing the LDPC codes, which are indeed linear.

Theorem 3.1. *There exists a sequence of linear codes with rate approaching $pH(Y|X) + (1-p)H(Y)$, indexed by the code length n , with uniformly diminishing error probability for the above problem of source coding with partial decoder side information for all cyclic shifts.*

The following two observations are now worth noting. Firstly, the position of $T(t)$ being 1 does not need to be the first k positions. In fact, any pattern can be used with k positions assigned 1, as long as the pattern is repeated for all the blocks. More specifically, for $k/n = k'/n'$ where k' and n' are coprime of each other, we can choose $T(t)$ to be a sequence alternating between k' ones, and $n' - k'$ zeros. In the simulations given in Section 3.4, this kind of $T(t)$ sequences will always be assumed. Secondly the proposed scheme has the advantage that decoding errors do not propagate across blocks, because a single encoding (and decoding) step is essentially isolated to two consecutive blocks. The overall decoding procedure restarts when the first decoding step is used in each cycle.

3.2.3 Overview of the LDPC Design

In each step of the proposed source splitting scheme, with a given LDPC code, the encoding and decoding procedure is well known (see [13]), and thus we only need to

focus on finding good codes. The overall code design consists of finding the following three codes.

1. A lossless code for encoding length- k source X sequences. This is a well understood module, and any good lossless compression algorithm can be used.
2. An LDPC code of rate approximately $pH(Y|X) + (1 - p)H(Y)$ to encode the length- n source Y block, with length- k source X samples at the decoder, the positions of which are unknown to the encoder but known to the decoder. This step is discussed in more details in Section 3.3.
3. An LDPC code of rate approximately $H(X|Y)$ to encode the rest $n - k$ samples in the source X block, with Y block as side information at the decoder. This step is similar, and in fact simpler than the second step. We also discuss this design step using the result of [16] in Section 3.3.

3.3 Equivalent Channel Model and Code Design in the A-SW Setting

LDPC codes in conjunction with belief propagation has shown extremely good performance in channel coding, which can in fact approach the capacity of many classes of channels [31]. The application of LDPC codes to the Slepian-Wolf problem was first suggested in [32] and further investigated in [13]. These results on S-SW coding mostly focus on the case that the (symmetric) source X and Y can be understood as connected by a symmetric channel, and sources with general distribution structure were largely overlooked.

Recently, a link between Slepian-Wolf coding and channel coding, referred to as source-channel correspondence, has been established in [16]. Through this link,

coding for a corner point of the Slepian-Wolf rate region, i.e., the problem of source coding with decoder side information, can be transformed to a channel coding problem with the same error probability; consequently, given an arbitrary source Y and side information X , capacity-achieving LDPC codes for the equivalent channel can be designed using existing algorithms, which also approach the Slepian-Wolf limit of the source. The proof is briefly reviewed in the appendix B. In this section, we discuss the code design problems of step two and step three in our source splitting scheme using this link.

3.3.1 Source Coding With Decoder Side Information

We now briefly review the source-channel correspondence result in [16]. This will provide an explicit code design for the third step in the splitting scheme, i.e, encoding a block of X samples with the corresponding side information Y block at the decoder.

For notational simplicity, let us only consider the special setting when the source X is in certain finite field, but this requirement is by no means necessary; see [16]. Let the two sources in the alphabets $(\mathcal{X}, \mathcal{Y})$ be distributed as Q_{XY} , where \mathcal{X} is a certain finite field and $\mathcal{Y} = \{1, 2, \dots, J\}$. The equivalent channel [16] with input U in the alphabet \mathcal{X} and output $V = [V_1, V_2]$ is depicted in Figure 3.3, where $V_1 = U \oplus X$ and $V_2 = Y$, with the addition \oplus in the finite field \mathcal{X} . Here U is independent of X and

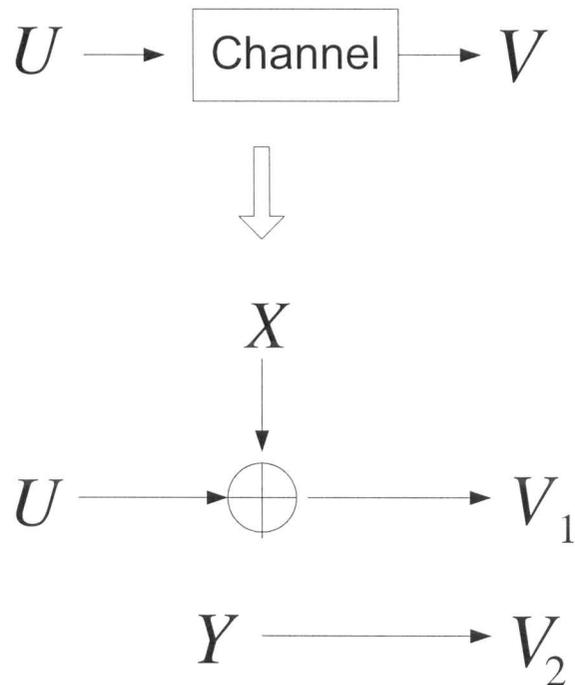


Figure 3.3: The equivalent channel for source coding with decoder side information.

Y . The mutual information between the input U and output V of this channel is

$$\begin{aligned}
 I(U; V) &= H(V) - H(V|U) \\
 &= H(U \oplus X, Y) - H(U \oplus X, Y|U) \\
 &\leq H(U \oplus X) + H(Y) - H(X, Y|U) \\
 &\leq H(U \oplus X) + H(Y) - H(X, Y) \\
 &\leq H(U \oplus X) - H(X|Y) \\
 &\leq \log |\mathcal{X}| - H(X|Y)
 \end{aligned} \tag{3.8}$$

It is clear that the capacity of the channel is achieved when U has a uniform distribution [16], resulting in

$$C = \log |\mathcal{X}| - H(X|Y). \tag{3.9}$$

Moreover, when linear codes are used, the decoding error probability of the channel coding problem and that of the source coding with decoder side information problem are exactly the same, under maximum likelihood decoder and belief propagation decoding. Because we have $H(X|Y) = \log |\mathcal{X}| - C$, if we can design the LDPC code to approach the capacity of this channel, then the same code can be used to approach the Slepian-Wolf limit, with the same error probability.

The density evolution algorithm developed in [33] can be used to design the parity check matrix for a given channel; more precisely, the degree distribution of the variable nodes and the check nodes in the factor graph [31] can be designed this way. An improved density evolution based algorithm, called discretized density evolution [34], was further developed in order to reduce the design complexity. Thus given a source and its side information structure, we can first transform the problem into an equivalent channel coding problem, then use the algorithm in [34] to design good LDPC code. This will yield good codes for the original source coding problem. We give a design example using this method in Section 3.4.

3.3.2 Source Coding With Partial Decoder Side Information

In the previous subsection, we see that the source-channel correspondence result [16] can be used to aid the design of LDPC code for the third step in the splitting scheme. For the second step, this method does not directly apply because only a partial side information sequence is available for the length- n source block.

To apply the source-channel correspondence result [16], an explicit single-letter probability structure between the source and the side information needs to be found. For this purpose, we return to the original source splitting scheme with common randomness. As we have discussed, the deterministic sequence $T(t)$ in fact approximates the effect of the randomized one. Due to this reason, we shall use the split source

$Z(t)$ defined in (3.2) to replace the length- k partial side information sequence, the positions of which are unknown to the encoder.

Now assume the alphabet is given by $\mathcal{X} = \{1, 2, \dots, |\mathcal{X}|\}$, then the joint distribution Q_{YZ} is clearly

$$Q_{YZ}(y, x) = p \cdot Q_{YX}(y, x), \quad Q_{YZ}(y, 0) = (1 - p) \cdot Q_Y(y). \quad (3.10)$$

With this explicit distribution between the source Y and the side information Z , we can now again first apply the transform to the equivalent channel, then use the design algorithm in [34] to find good LDPC codes. This is turned into an almost identical design problem as that for the third step, with the only difference being the insertion of erasure symbol 0 into the original problem. In Section 3.4, we give a detailed example on the design of such codes.

3.4 Design Examples and Numerical Results

In this section, we give several code design examples and results. We start with the third and the second steps in the splitting scheme, then the overall performance is discussed. To be more concrete, we focus on the distribution Q_{XY}

$$Q_{XY}(x, y) = \begin{cases} 0.45 & x = 1, y = 1 \\ 0.05 & x = 1, y = 2 \\ 0.09 & x = 2, y = 1 \\ 0.41 & x = 2, y = 2, \end{cases} \quad (3.11)$$

however, the design method is general and can be applied on any joint distributions. Note that we have chosen the alphabet $\{1, 2\}$ instead of the usual binary field $\{0, 1\}$ to be consistent with the discussion in the previous section. This source distribution

can not be modeled as a binary symmetric channel and thus using codes designed for binary symmetric channels is not suitable.

3.4.1 Example of Source Coding With Decoder Side Information

We first consider the design for the third step in our splitting scheme, i.e., design LDPC code to encode X with side information Y based on the equivalent channel model shown in Figure 3.3.

The initial log-likelihood ratio (LLR) messages $\log \frac{P_{U|V}(u=1|v)}{P_{U|V}(u=2|v)}$ and their associated probabilities need to be determined in order to apply the discretized density evolution (DE) [34] [35] algorithm which is introduced in the appendix C. Using the equivalent channel given in Section 3.3.1, it is straightforward to verify that these LLRs and associated probabilities are as follows, assuming an all-zero codeword (the calculation procedure is shown in the appendix D):

$$\log \frac{P_{U|V}(1|v)}{P_{U|V}(2|v)} = \begin{cases} \log 5 & \text{for } v = (1, 1) \text{ with prob. } 0.45 \\ \log \frac{5}{41} & \text{for } v = (1, 2) \text{ with prob. } 0.05 \\ \log \frac{1}{5} & \text{for } v = (2, 1) \text{ with prob. } 0.09 \\ \log \frac{41}{5} & \text{for } v = (2, 2) \text{ with prob. } 0.41 \end{cases}$$

Several good degree distributions with code rates 0.6, 0.602, 0.604, 0.607, 0.609, 0.610, 0.612, 0.614 are obtained by setting the maximum variable node degree to be 20, $\delta(e_{l-1} - e_l) = 0.001$ and $\Delta\lambda = 0.0005$, where $\delta(e_{l-1} - e_l)$ and $\Delta\lambda$ are the parameters involved in the linear programming setting of [35]. As an example, the

degree distribution of code rate 0.614 is given below:

$$\begin{aligned}\lambda(x) &= 0.213389x + 0.173764x^2 + 0.063x^3 + 0.063x^4 \\ &\quad + 0.056087x^5 + 0.036943x^6 + 0.37x^7 + 0.42x^8 + 0.314816x^{19} \\ \rho(x) &= x^6.\end{aligned}$$

In the simulation of each code, source sequences of length 10^8 are generated by the joint distribution shown in (3.11). Each sequence is divided into 500 blocks with block length $n = 2 \times 10^5$ bits. Each block is decoded with the belief propagation algorithm, for which the number of iteration is limited to 150. The same belief propagation algorithm is also used in simulations discussed in later sections.

Figure 3.4 shows the performance of these Slepian-Wolf codes under two testing scenarios. In the first test, all the testing source sequences are ϵ -jointly-typical blocks with $\epsilon = 0.001$ [36], and in the second test the testing source sequences are randomly generated by the joint distribution in (3.11) so that they can be either typical or atypical. Apparently, we expect the first test to yield better results than the second one, since the codes are specifically designed for the given distribution. In Figure 3.4, we see that the gap to the Slepian-Wolf limit of 0.57921 is 0.03 bit in the first test and 0.035 bit for the second. These results are comparable to the 0.033 and 0.06 bit gap for results on the symmetric sources using the LDPC codes in [13] [14], with code length 10^5 , bit error rate (BER) less than 10^{-5} and similar decoding algorithms; recall that the source distribution in (3.11) can not be modeled as connected by a symmetric channel, and thus it is expected to be more difficult to code.

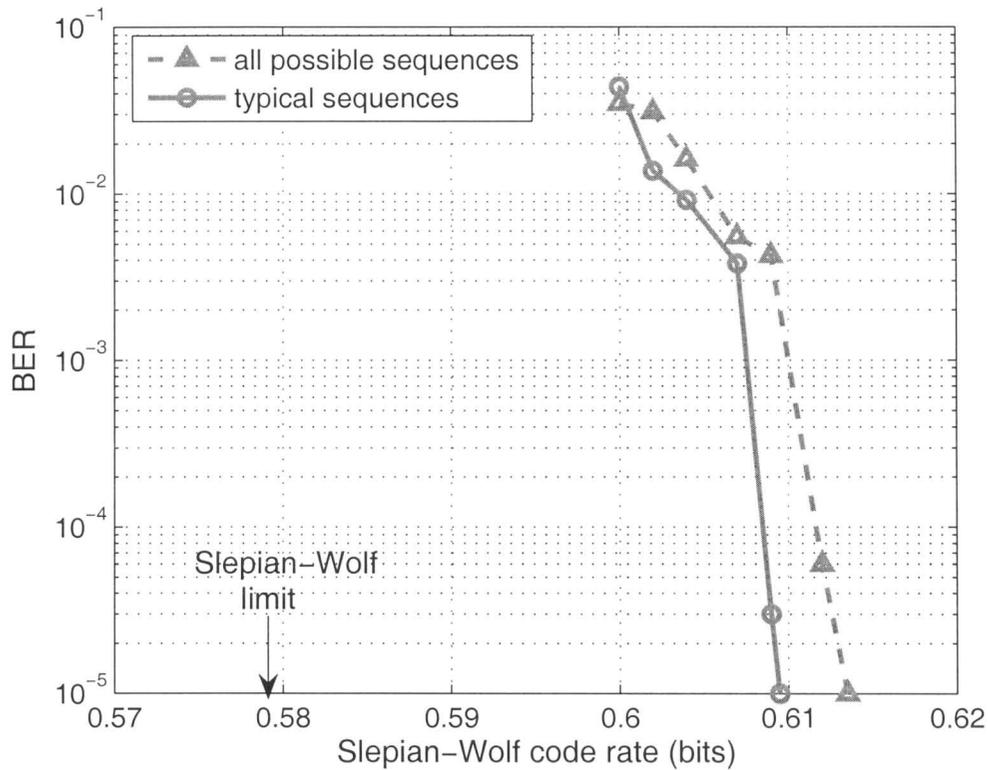


Figure 3.4: The performances of 8 irregular LDPC codes of length 2×10^5 , with side information Y at the decoder.

3.4.2 Example of Source Coding With Partial Decoder Side Information

We now consider the second step in our source splitting scheme, and focus on encoding for the mid-point on the dominant face of the Slepian-Wolf region, i.e., $p = 0.5$. We need to design code for source Y with the (randomized) partial side information Z

at the decoder, whose joint distribution is

$$Q_{YZ}(y, z) = \begin{cases} 0.27 & y = 1, z = 0 \\ 0.225 & y = 1, z = 1 \\ 0.045 & y = 1, z = 2 \\ 0.23 & y = 2, z = 0 \\ 0.025 & y = 2, z = 1 \\ 0.205 & y = 2, z = 2 \end{cases}$$

The initial (LLR) messages and their associated probabilities are as shown below:

$$\log \frac{P_{U|V}(1|v)}{P_{U|V}(2|v)} = \begin{cases} \log\left(\frac{27}{23}\right) & \text{for } v = (1, 0) \text{ with prob. } 0.27 \\ \log(9) & \text{for } v = (1, 1) \text{ with prob. } 0.225 \\ \log\left(\frac{9}{41}\right) & \text{for } v = (1, 2) \text{ with prob. } 0.045 \\ \log\left(\frac{23}{27}\right) & \text{for } v = (2, 0) \text{ with prob. } 0.23 \\ \log\left(\frac{1}{9}\right) & \text{for } v = (2, 1) \text{ with prob. } 0.025 \\ \log\left(\frac{41}{9}\right) & \text{for } v = (2, 2) \text{ with prob. } 0.205 \end{cases}$$

Under the same assumption and the same parameters setting as the previous example, we apply the discretized density evolution algorithm [34] [35], and find several LDPC codes of rates 0.806, 0.808, 0.810, 0.814, 0.816, respectively. The Slepian-Wolf limit is 0.7850, and thus these codes are less than 0.031 bit away from this lower bound. As an example, the degree distribution of code rate 0.816 is given below:

$$\begin{aligned} \lambda(x) &= 0.394235x + 0.212846x^2 + 0.011x^3 + 0.092328x^4 \\ &\quad + 0.078893x^5 + 0.210698x^{14} \\ \rho(x) &= 0.1x^2 + 0.9x^3 \end{aligned}$$

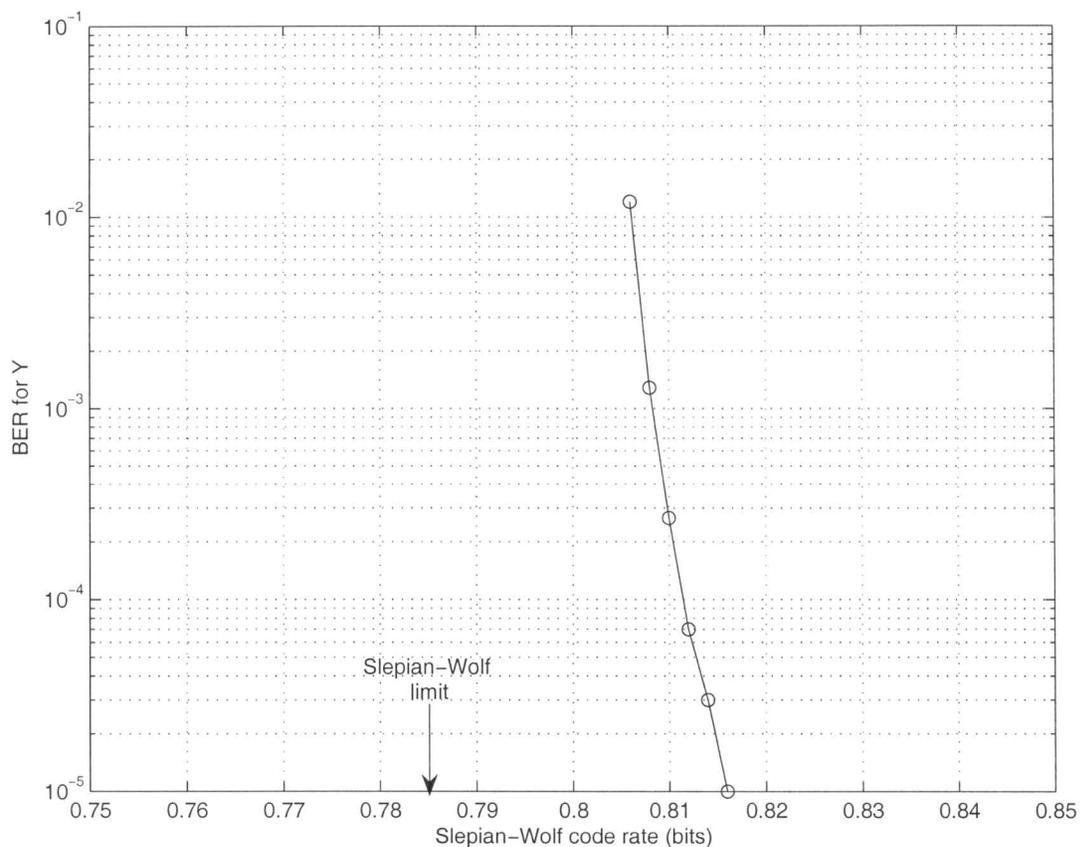


Figure 3.5: The performances of 6 irregular LDPC codes of length 2×10^5 , with partial X side information at the decoder.

Since $p = 0.5$, we can choose the pattern of $T(t)$ to be a sequence alternating between one and zero, and subsequently we can assume $d_x = 0$ without loss of generality. By doing so, for a single fixed code, all the odd d_y values induce exactly the same error probability, and all the even d_y values induce exactly the same error probability. Thus we only need to perform the simulation for the two cases $d_y = 0$ and $d_y = 1$.

The performance of these codes is shown in Figure 3.5, where the code length

is again 2×10^5 , and source sequences of length 10^8 are generated by the given joint probability without removing the atypical blocks. Results of the same code for both $d_y = 0$ and $d_y = 1$ are shown in Table 3.1, for three different codes with rates 0.812, 0.814, 0.816, respectively. It can be seen that for each code the error probability is consistent between the two d_y delay values, implying the effectiveness of this coding step in the asynchronous setting.

R_2	d_y	BER
0.812	0	7.01×10^{-5}
	1	6.12×10^{-5}
0.814	0	3.13×10^{-5}
	1	4.21×10^{-5}
0.816	0	0.98×10^{-6}
	1	1.17×10^{-5}

Table 3.1: Performances of different codes under various d_y with $p = 0.5$.

3.4.3 Overall Code Performance

We are now ready to evaluate the overall code performance in the A-SW setting. It is clear that by using variable-length codes, the first coding step in the source splitting scheme can achieve zero error with a negligible rate increase over $H(X)$ compared to the latter two steps, and thus we shall omit the error probability and assume the rate is simply $H(X)$ in the first step when calculating the overall rates and error probability.

The block length n is fixed at 3×10^5 for all simulation in this subsection and as such the length of code used in the third step is in fact $(1-p)n$. Each source sequence of length 3×10^8 is generated, and the error probability is averaged over both X and Y sequences, and thus the BER may be smaller than the corresponding ones shown in Table. 3.1.

For $p = \frac{1}{2}$, in order to drive the overall error probability smaller, we choose a slightly larger rate in the third coding step than the ones given in Section 3.4.1. We again assume $d_x = 0$, and recall for this case we only need to test the cases $d_y = 0$ and $d_y = 1$. Using the same design method we also find codes for the rate pairs associated with time sharing parameter $p = \frac{1}{3}$ and $p = \frac{2}{3}$, respectively. The resulting performances of these codes are summarized in Table 3.3, where we again assumed $d_x = 0$. The degree distributions in the second encoding step (rate 0.8902) and the third encoding step (rate 0.6185) for the case $p = \frac{1}{3}$ are:

$$\begin{aligned}\lambda(x) &= 0.480869x + 0.206888x^2 + 0.010614x^3 \\ &\quad + 0.070857x^4 + 0.094407x^5 + 0.136366x^{14} \\ \rho(x) &= 0.75x^2 + 0.25x^3.\end{aligned}$$

and

$$\begin{aligned}\lambda(x) &= 0.213124x + 0.170764x^2 + 0.06x^3 + 0.06x^4 \\ &\quad + 0.053352x^5 + 0.039944x^6 + 0.4x^7 + 0.45x^8 + 0.317816x^{19} \\ \rho(x) &= x^6.\end{aligned}$$

The degree distributions in the second encoding step (rate 0.746) and the third encoding step (rate 0.623) for the case $p = \frac{2}{3}$ are:

$$\begin{aligned}\lambda(x) &= 0.356665x + 0.248526x^2 + 0.028x^3 + 0.027575x^4 \\ &\quad + 0.173207x^5 + 0.028611x^6 + 0.072795x^{15} + 0.064621x^{16} \\ \rho(x) &= 0.7x^3 + 0.3x^4.\end{aligned}$$

and

$$\lambda(x) = 0.212855x + 0.167764x^2 + 0.057x^3 + 0.057x^4 \\ + 0.050621x^5 + 0.042944x^6 + 0.43x^7 + 0.48x^8 + 0.320816x^{19}$$

$$\rho(x) = x^6.$$

Encoding rates in step 1, 2, 3	d_y	BER
0.5, 0.812, 0.311 ($p = 1/2$)	0	3.61×10^{-5}
	1	3.21×10^{-5}
0.5, 0.814, 0.311 ($p = 1/2$)	0	1.72×10^{-5}
	1	2.26×10^{-5}
0.5, 0.816, 0.311 ($p = 1/2$)	0	6.41×10^{-6}
	1	7.35×10^{-6}
0.33, 0.8902, 0.412 ($p = 1/3$)	0	9.79×10^{-6}
	1	1.00×10^{-5}
	2	1.01×10^{-5}
0.67, 0.746, 0.208 ($p = 2/3$)	0	2.81×10^{-6}
	1	4.83×10^{-6}
	2	2.87×10^{-6}

Table 3.2: Overall code performance operating under different d_y for, $p = 1/2$, $p = 1/3$ and $p = 2/3$, respectively.

To quantify the coding efficiency without the asynchronous requirement, we include the performances for $d_y = 0$ in Table 3.2. The desired rate pair on the dominant face of the Slepian-Wolf region is shown together with the actual code rate pair (R_1, R_2) from the design in Table 3.3. The average BER is kept below 10^{-5} and the gap is measured in terms of the Euclidean distance. These results are roughly on the

same order as the best known designs [37] for symmetric sources to achieve general S-SW rate pairs, where by using IRA code with block length 10^5 , a gap of 0.039 to the Slepian-Wolf limit is reported. Thus the design proposed in this work can achieve satisfactory performance in the S-SW setting, even it is in fact designed for the more general A-SW setting.

p	Target rates	Actual rates	Gap	BER
$\frac{1}{3}$	(0.7195, 0.8551)	(0.7457, 0.8902)	0.044	6.88×10^{-6}
$\frac{1}{2}$	(0.7896, 0.7850)	(0.8108, 0.8162)	0.038	9.96×10^{-6}
$\frac{2}{3}$	(0.8597, 0.7148)	(0.8743, 0.7460)	0.034	3.50×10^{-6}

Table 3.3: Results in terms of distance to the Slepian-Wolf limit.

Chapter 4

Lossy Source Coding for General Source Using LDGM Codes

Inspired by the success of LDPC codes and belief propagation algorithm in approaching the Shannon capacity, similar techniques have been proposed for lossy source coding. In particular, LDGM codes in conjunction with variants of message-passing algorithms have shown the potential to approach the rate-distortion bound [21, 22, 23, 24, 25, 26, 27]. In this Chapter, we propose a multi-level coding scheme based on LDGM codes and SP algorithm. This scheme works for the lossy source coding problem with arbitrary source distribution and distortion measure.

4.1 Extend the Existing Method to Non-uniform Sources

All recent researches of lossy source coding are almost exclusively focused on uniformly distributed sources. It is worth noting that the extension to sources with general distributions is not straightforward. Indeed, the existing LDGM codes based methods,

when directly applied to non-uniform sources, often incur significant performance degradation. Figure 4.1 plots the rate-distortion bound for a binary source with the Hamming distortion measure as well as the empirical rate-distortion curve achieved by the algorithm given in [22]. It is easy to see from Figure 4.1 (b) that the gap between the two curves is quite significant. The simulation results indicate that the gap gets larger as the source distribution becomes more biased.

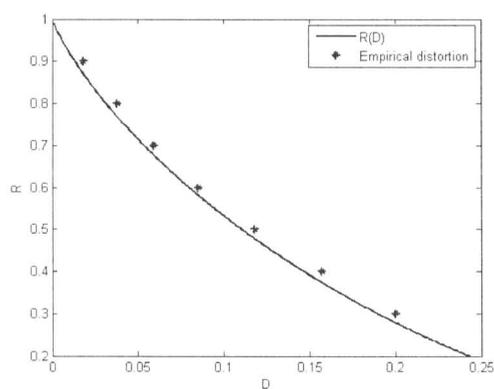
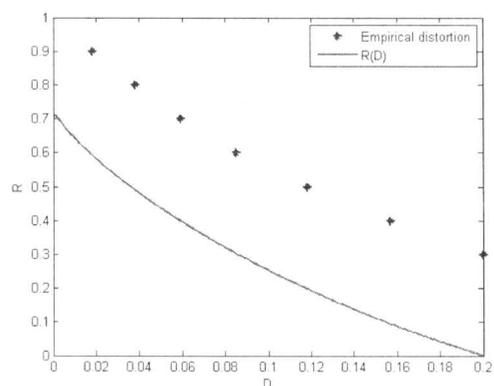
(a) $P_X(0)=0.5$ (b) $P_X(0)=0.2$

Figure 4.1: Performance degradation for non-uniform sources.

The gap between the empirical performance and the theoretical limit can be

roughly explained as follows. Given a source distribution P_X on \mathcal{X} and a distortion measure $d(\cdot, \cdot) : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow [0, d_{\max}]$, the rate-distortion function is given by

$$R(D) = \min_{P_{\hat{X}|X}} I(X; \hat{X}), \quad (4.1)$$

where the minimization is over all possible test channels $P_{\hat{X}|X}$ subject to the constraint $\mathbb{E}d(X, \hat{X}) \leq D$. Let $P_{\hat{X}|X}^*$ be the test channel that achieves the minimum in (4.1) and $P_{\hat{X}}^*$ be the output distribution induced by P_X and $P_{\hat{X}|X}^*$. The rate-distortion theory indicates that to achieve the rate-distortion bound, one has to use a codebook whose dominant codeword type is approximately $P_{\hat{X}}^*$. Note that except for certain special cases (say, uniform sources with the Hamming distortion measure), $P_{\hat{X}}^*$ is not a uniform distribution. In contrast, for most commonly used random linear code ensembles, the maximum of the expected weight enumerator is at the point corresponding to the uniform distribution. The mismatch between the optimal output distribution and the dominant weight of linear codes suggests that linear codes are in general suboptimal for lossy source coding in terms of rate-distortion tradeoff.

To circumvent the aforementioned problem while still maintaining the linear code structure, we propose a multilevel coding scheme, which is based on the idea of approximating the optimal output distribution $P_{\hat{X}}^*$ by a uniform distribution over a (possibly) larger alphabet [28].

4.2 Multilevel Coding

4.2.1 Basic Coding Scheme

The key idea of our proposed schemes is to approximate the optimal output distribution $P_{\hat{X}}^*$ by a uniform distribution over a (possibly) larger alphabet [28]. Suppose the source symbol X is a random variable over $GF(q)$ with distribution P_X

and the optimal output distribution optimized by the rate distortion function is $P_{\hat{X}}^*(0) = \frac{n_0}{2^i}, P_{\hat{X}}^*(1) = \frac{n_1}{2^i}, \dots, P_{\hat{X}}^*(q-1) = \frac{n_{q-1}}{2^i}$. Since the denominators of the optimal output distribution is 2^i , we construct i independent, uniformly distributed binary random variables Y_1, Y_2, \dots, Y_i . Then we define a deterministic mapping $f: GF^i(2) \rightarrow GF(q)$ such that $f(Y_1, Y_2, \dots, Y_i) = 0$ if the decimal value of the binary expression $y_1 y_2 \dots y_i$ is less than or equal to n_0 , and $f(Y_1, Y_2, \dots, Y_i) = k$ if the decimal value of the binary expression $y_1 y_2 \dots y_i$ is greater than $\sum_{j=0}^{k-1} n_j$ but less than or equal to $\sum_{j=0}^k n_j$, where $1 \leq k \leq q-1$. Let $\hat{X} = f(Y_1, \dots, Y_i)$. It is clear that the induced distribution of source reconstruction \hat{X} is exactly the optimal output distribution. Now associate Y_1, Y_2, \dots, Y_i with i linear codes $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_i$ respectively. Since Y_1, Y_2, \dots, Y_i are uniformly distributed, the dominant type of codewords (uniform distribution) in all linear codes can get efficiently used. Apparently, the role of the deterministic mapping $f(\cdot)$ is to convert a mismatched codebook to a codebook whose dominant type is consistent with the optimal output distribution.

The multilevel coding scheme based on the previous discussed idea [39] can be described as follows. Suppose we approximate the optimal output distribution by the uniform distribution over $GF^i(2)$ through a deterministic mapping. In this scheme, we construct i LDGM codes with the same length n . The rate for each code can be arbitrary as long as the sum-rate equals the desired code rate. In the factor graph representation, each LDGM code has the same number of check nodes and a certain number of variable nodes. We also introduce a new kind of nodes called network nodes to establish links among different LDGM codes by performing the deterministic mapping. The number of network nodes is equal to the code length n . Each network node is connected to i corresponding check nodes of the same index.

The idea is illustrated in Figure 4.2. In this example, a binary output distribution $P_{\hat{X}}$ with $P_{\hat{X}}(0) = 0.25$ is approximated by two uniformly distributed binary random variables Y_1, Y_2 .

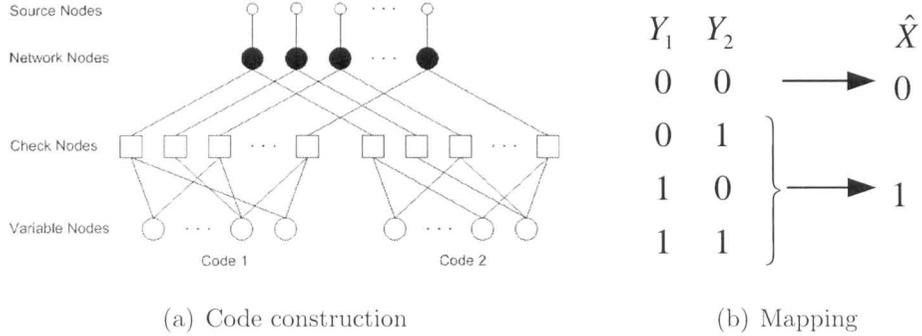


Figure 4.2: Approximating a binary output distribution with $P_{\hat{X}}(0) = 0.25$ by two uniformly distributed binary random variables.

4.2.2 A More Generalized Coding Scheme

In the aforementioned coding scheme, we have i LDGM codes with length m each, and all the codes are separated from each other in the message-passing routines. To achieve the best performance, we have to optimize the rate allocation among different linear codes, which is not an easy job. Moreover, i LDGM codes need to be generated and encoding over i separate LDGM codes is more complex. An alternative approach is to use a single linear code of extended length which can overcome the above difficulties. The underlying idea, i.e., approximating the non-uniform output distribution by a uniform distribution through deterministic mapping, remains the same.

In this scheme, instead of constructing i LDGM codes with same code length n to represent Y_1, Y_2, \dots, Y_i respectively, we construct one LDGM code of length $i \times n$ and the code rate equal to R_{desire}/i where R_{desire} is the number of the bits used to store the compressed source data divided by the number of the source bits. In the factor graph representation, we divide all the check nodes into i groups with n consecutive check nodes in each group. Network node k ($k \leq n$) connects to source node k and the k -th check nodes in each of i groups. The reconstructed symbol will be obtained

at the network node by mapping the adjacent check nodes based on the optimal output distribution. Since variable nodes are used to store the compressed bits, the information rate equals to the number of variable nodes, $i \times n \times R_{desired} / i = n \times R_{desired}$, divided by the number of source nodes, n , which is still $R_{desired}$.

Figure 4.3 shows an example of the new scheme with $i = 2$. Here we use a single LDGM code with twice the length of the source sequence. Although the rate of this LDGM code is one half of the desired rate, the actual rate of this encoding scheme is still the desired rate. The deterministic mapping is the same as the example shown in Figure 4.2.

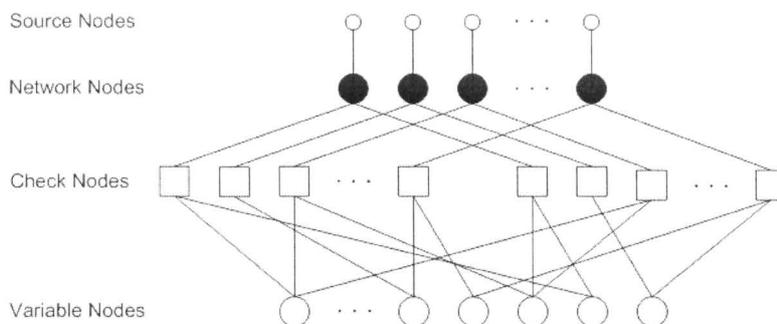


Figure 4.3: Using a single LDGM code of extended length.

Actually, this new coding scheme can be viewed as a generalization of the previous one. This generalization not only simplifies the code construction, but also improves the performance. Simulation results show that the coding scheme using single LDGM code of extended length always outperforms the coding scheme using multiple LDGM codes [39]. Therefore, in the following sections of this paper, we only discuss the generalized coding scheme.

4.2.3 Finding the Optimal Output Distribution

As we discussed above, the deterministic mapping is defined according to the optimal output distribution $P_{\hat{X}}^*$, which can be calculated as follows. Given the rate-distortion theorem

$$R(D) = \min_{P_{\hat{X}|X}} I(X; \hat{X}) \quad (4.2)$$

we minimize $I(X; \hat{X})$ over all possible test channels $P_{\hat{X}|X}$ subject to the constraints $\mathbb{E}d(X, \hat{X}) \leq D$ and $P_{\hat{X}}^*$ must be a multiple of $\frac{1}{2^i}$. The second constraint is introduced because we use i binary random variables to approximate the $P_{\hat{X}}^*$. After the optimal $P_{\hat{X}|X}$ is found, $P_{\hat{X}}^*$ is induced by P_X and $P_{\hat{X}|X}$. Apparently, $P_{\hat{X}|X}$ varies at different distortion requirements which results in different code rates. The optimal distributions of three different sources and distortion measures are calculated and shown in the following section.

4.3 Message Passing Algorithm

After constructing the factor graph, the remaining problem is to find an efficient algorithm to map the source sequences to the codewords. To solve this problem, we propose a message-passing algorithm, which is inspired by the powerful survey propagation algorithm [29, 30].

Let the check nodes and variable nodes be denoted as $\{C_1, C_2, \dots\}$ and $\{V_1, V_2, \dots\}$. Assume the source sequence is $\{x_1, x_2, \dots\}$ and each source symbol is drawn from the source alphabet $\mathcal{X} = \{0, 1, \dots, (|\mathcal{X}| - 1)\}$ by a given distribution. Denote the source nodes and the network nodes by $\{S_1, S_2, \dots\}$ and $\{N_1, N_2, \dots\}$ respectively. The source nodes store the source sequence that need to be compressed and the output sequence \hat{X}^n will be constructed at the network nodes by mapping the values of

adjacent check nodes. Let $A_c(k)$ be the set of check nodes connected to the variable node k and $B_v(j)$ be the set of variable nodes connected to the check node j . The messages from the variable node k to check node j consists of three components: $M_{V_k \rightarrow C_j}^0$, $M_{V_k \rightarrow C_j}^1$ and $M_{V_k \rightarrow C_j}^*$ denoting the probabilities that variable node k forces check node j to be 0, 1 or free state respectively. The message from the check node j to variable node k also consists of three components: $M_{C_j \rightarrow V_k}^0$, $M_{C_j \rightarrow V_k}^1$ and $M_{C_j \rightarrow V_k}^*$ denoting the probabilities that check node j forces variable node k to be 0, 1 or free state respectively. The message from source node l to the network node l consists of $|\mathcal{X}|$ components: $M_{S_l}^0, M_{S_l}^1, \dots, M_{S_l}^{(|\mathcal{X}|-1)}$ denoting the probabilities that source node l forces network node l to be 0, 1, $\dots, (|\mathcal{X}|-1)$ respectively. The message from network node l to the l -th check node in q -th group consists of two components: $M_{N_l^q}^0$ and $M_{N_l^q}^1$ denoting the probabilities that network node l forces l -th check node in q -th group to be 0, 1 respectively.

The message-passing routine runs as follows:

Step 1. Each check node sends message to the adjacent variable nodes and network node. The initial message sent in the first iteration is $(0.5, 0.5, 0)$.

Step 2. Each network node and variable node calculate the message to be passed to the adjacent check nodes based on the message they received, then send the message back to the check nodes.

Step 3. Check nodes receive the messages from network nodes and variable nodes. Calculate the new message to be sent out and check whether this new message converge with the message sent in the previous iteration or not. If the messages converge at all check nodes or the number of iteration reaches the maximum value (i.e. 100), then go to step 4, otherwise go to step 1.

Step 4. Calculate the marginal distribution (using the message equations from variable node to check node but without the exclusion of C_j in $A_c(k)$ in the product operation) of each variable node. Find those variable nodes whose bias value is greater than a certain threshold (i.e. $|P(0) - P(1)|$ is greater than a certain value) and fix the value of these variables accordingly. If there is no variable node whose bias value is greater than the threshold then we fix the value of the variable node whose bias value is the largest.

Step 5. Remove the variable nodes whose value are fixed from the factor graph. Check whether all the variable nodes are fixed or not, if all the variable nodes are fixed then go to step 6, otherwise go to step 1.

Step 6. Calculate the value of each check node (“XOR” of all the adjacent variable node values). At each network node, construct the decoded symbol using the adjacent check node values according to the deterministic mapping.

The message calculation equations of variable nodes and check nodes are shown in Figure 4.4. The message vector in any direction on any edge needs to be normalized so that the sum is equal to one since it represents the probability. c is the value of check node C_j (“XOR” of all the adjacent variable nodes value). δ and W_{sou} are the parameters that can be adjusted.

However, the equations for messages from source nodes to network nodes depend on the source alphabet size, number of check node groups i and the specific deterministic mapping being used. The messages passed from network node l to its adjacent check nodes consist two components, $M_{N_i}^0$ and $M_{N_i}^1$ that represent the probabilities of the network node l force the destination check node to be 0 or 1, respectively. These two components are calculated by marginalizing the probability of the l -th

Variable node to check node

$$\begin{aligned}
M_{V_k \rightarrow C_j}^0 &= \prod_{l \in A_c(k) \setminus \{C_j\}} (1 - M_{C_l \rightarrow V_k}^1) - \prod_{l \in A_c(k) \setminus \{C_j\}} (1 - M_{C_l \rightarrow V_k}^1 - M_{C_l \rightarrow V_k}^0) \\
M_{V_k \rightarrow C_j}^1 &= \prod_{l \in A_c(k) \setminus \{C_j\}} (1 - M_{C_l \rightarrow V_k}^0) - \prod_{l \in A_c(k) \setminus \{C_j\}} (1 - M_{C_l \rightarrow V_k}^1 - M_{C_l \rightarrow V_k}^0) \\
M_{V_k \rightarrow C_j}^* &= \prod_{l \in A_c(k) \setminus \{C_j\}} (1 - M_{C_l \rightarrow V_k}^1 - M_{C_l \rightarrow V_k}^0)
\end{aligned} \tag{4.3}$$

Check node to variable node

$$\begin{aligned}
M_{C_j \rightarrow V_k}^0 &= \frac{1}{2} [(M_{N_l \rightarrow C_j}^0 + M_{N_l \rightarrow C_j}^1) \prod_{l \in B_v(j) \setminus \{V_k\}} (M_{V_l \rightarrow C_j}^0 + M_{V_l \rightarrow C_j}^1) \\
&\quad + (M_{N_l \rightarrow C_j}^0 - M_{N_l \rightarrow C_j}^1) \prod_{l \in B_v(j) \setminus \{V_k\}} (M_{V_l \rightarrow C_j}^0 - M_{V_l \rightarrow C_j}^1)] \\
M_{C_j \rightarrow V_k}^1 &= \frac{1}{2} [(M_{N_l \rightarrow C_j}^0 + M_{N_l \rightarrow C_j}^1) \prod_{l \in B_v(j) \setminus \{V_k\}} (M_{V_l \rightarrow C_j}^0 + M_{V_l \rightarrow C_j}^1) \\
&\quad - (M_{N_l \rightarrow C_j}^0 - M_{N_l \rightarrow C_j}^1) \prod_{l \in B_v(j) \setminus \{V_k\}} (M_{V_l \rightarrow C_j}^0 - M_{V_l \rightarrow C_j}^1)] \\
M_{C_j \rightarrow V_k}^* &= 1 - M_{C_j \rightarrow V_k}^0 - M_{C_j \rightarrow V_k}^1
\end{aligned} \tag{4.4}$$

Check node to network node

$$\begin{aligned}
M_{C_j \rightarrow N_i}^0 &= \frac{1}{2} \left[\prod_{l \in B_v(j)} (M_{V_l \rightarrow C_j}^0 + M_{V_l \rightarrow C_j}^1) + \prod_{l \in B_v(j)} (M_{V_l \rightarrow C_j}^0 - M_{V_l \rightarrow C_j}^1) \right] \\
M_{C_j \rightarrow N_i}^1 &= \frac{1}{2} \left[\prod_{l \in B_v(j)} (M_{V_l \rightarrow C_j}^0 + M_{V_l \rightarrow C_j}^1) - \prod_{l \in B_v(j)} (M_{V_l \rightarrow C_j}^0 - M_{V_l \rightarrow C_j}^1) \right] \\
M_{C_j \rightarrow N_i}^* &= 1 - M_{C_j \rightarrow N_i}^0 - M_{C_j \rightarrow N_i}^1
\end{aligned} \tag{4.5}$$

*If all $V_l \in B_v(j)$ are fixed, then

$$\begin{aligned}
M_{C_j \rightarrow N_i}^0 &= [(1 - c) \exp(\delta) + c \exp(-\delta)] / (\exp(\delta) + \exp(-\delta) + W_{sou}) \\
M_{C_j \rightarrow N_i}^1 &= [c \exp(\delta) + (1 - c) \exp(-\delta)] / (\exp(\delta) + \exp(-\delta) + W_{sou}) \\
M_{C_j \rightarrow N_i}^* &= W_{sou} / (\exp(\delta) + \exp(-\delta) + W_{sou})
\end{aligned} \tag{4.6}$$

Figure 4.4: Survey propagation equations of check node and variable node.

<p>Source node to network node</p> $M_{S_l}^k = \exp(\gamma) / (\exp(\gamma) + \sum_{m \in \mathcal{X}: m \neq x_l} \alpha_m \exp(-\gamma)) \quad k = x_l$ $M_{S_l}^k = \alpha_k \exp(-\gamma) / (\exp(\gamma) + \sum_{m \in \mathcal{X}: m \neq x_l} \alpha_m \exp(-\gamma)) \quad k \neq x_l$ <p style="text-align: right;">(4.7)</p> <p>Network node to check node</p> $M_{N_l^q}^0 = \sum_{(Y_1, \dots, Y_i): Y_q=0} (M_{S_l}^k \prod_{m=1: m \neq q}^i M_{C_l^m \rightarrow N_l}^{Y_m}) \quad k = f(Y_1, \dots, Y_i)$ $M_{N_l^q}^1 = \sum_{(Y_1, \dots, Y_i): Y_q=1} (M_{S_l}^k \prod_{m=1: m \neq q}^i M_{C_l^m \rightarrow N_l}^{Y_m}) \quad k = f(Y_1, \dots, Y_i)$ <p style="text-align: right;">(4.8)</p>

Figure 4.5: survey propagation equations of source node and network node.

check node being 0 or 1 in i -th group given the message from l -th source node and the messages from l -th check node in other groups that are also connected to network node l . The marginalization can be done by traversing all the deterministic mapping cases. Therefore, the equations of these two messages vary from case to case. The general forms of calculation equations are shown in Figure 4.5. The message vector also needs to be normalized for the same reason as the equations in Figure 4.4. α_k and γ are the parameters that control the strength of the impact of source symbols on the reconstructed symbols at network nodes, and they need to be adjusted in order to keep good performance of the algorithm at different coding rates. x_l is the l -th source symbol and $k \in \mathcal{X}$. $f(Y_1, \dots, Y_i)$ is the deterministic mapping function of i binary random variable and Y_m is the value of m -th random variable. $M_{C_l^m \rightarrow N_l}^{Y_m}$ denotes the probability that the value of l -th check node in the m -th group to be Y_m .

To verify the effectiveness of the proposed multilevel coding scheme and the message passing algorithm for different sources and distortion measures, we consider the following three cases: non-uniform binary source with hamming distance, uniform ternary source with hamming distance and uniform ternary source with non-hamming

distance.

4.3.1 Non-Uniform Binary Source with Hamming Distance

Given a non-uniform binary source $\mathcal{X} = \{0, 1\}$ and we want to approximate it with i uniform binary random variables. The distortion is measured using the hamming distance, i.e. the distance between two bits is 0 if they are identical and 1 otherwise. Based on the general form equation (Eq. (4.7)), the message from source node l to network node l are derived and shown as follows:

$$\begin{aligned} M_{S_l}^0 &= ((1 - x_l) \exp(\gamma) + \alpha x_l \exp(-\gamma)) / (\exp(\gamma) + \alpha \exp(-\gamma) + W_{sou}) \\ M_{S_l}^1 &= (x_l \exp(\gamma) + \alpha(1 - x_l) \exp(-\gamma)) / (\exp(\gamma) + \alpha \exp(-\gamma) + W_{sou}) \end{aligned}$$

where x_l is the l -th source symbol. γ and α are the control parameters that can be adjusted.

We consider the binary source with distribution $P_X(0) = 0.25$, and use 4 binary variables to approximate the output distribution. Apply Eq. (4.2), we find the optimal output distributions $P_{\hat{X}}^*$ among different rate regions and show them in Table 4.1. Suppose the encoding rate is chosen to be 0.4 bit/symbol, according to Table.4.1, the optimal output distribution is $P_{\hat{X}}^*(0) = \frac{3}{16}$ and $P_{\hat{X}}^*(1) = \frac{13}{16}$. The deterministic mapping is shown in Figure 4.6.

The messages from network node l to the l -th check node in i -th group are composed of two probabilities denoted as $(M_{N_i}^0, M_{N_i}^1)$. $(M_{C_i \rightarrow N_i}^0, M_{C_i \rightarrow N_i}^1)$ denotes the message from the l -th check node in i -th group to network node l . The message from source node l to network node l is $(M_{S_l}^0, M_{S_l}^1)$. The message passed from the l -th network node to the adjacent check node in i -th group is calculated based on the deterministic mapping. According to Eq. (4.8) and Figure 4.6, the message passed from the l -th network node to the l -th check node in 1st group is derived by setting

Rate region	$(P_{\hat{X}}^*(0), P_{\hat{X}}^*(1))$
0.479 - 1.000	$(\frac{4}{16}, \frac{12}{16})$
0.228 - 0.478	$(\frac{3}{16}, \frac{13}{16})$
0.096 - 0.227	$(\frac{2}{16}, \frac{14}{16})$
0.000 - 0.095	$(\frac{1}{16}, \frac{15}{16})$

Table 4.1: The optimal output distribution $P_{\hat{X}}^*$ at different rates for a non-uniform binary source with $P(0) = 0.25$.

$q = 1$ and $i = 4$. Eq. (4.9) shows the explicit form of this message, where $M_{N_i}^0$ and $M_{N_i}^1$ need to be normalized after calculation.

Y_1	Y_2	Y_3	Y_4	\hat{X}
0	0	0	0	} 0
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	} 1
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Figure 4.6: The deterministic mapping for optimal output distribution $P_{\hat{X}}^*(0) = \frac{3}{16}$ and $P_{\hat{X}}^*(1) = \frac{13}{16}$.

$$\begin{aligned}
M_{N_l}^0 &= M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
M_{N_l}^1 &= M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1
\end{aligned} \tag{4.9}$$

4.3.2 Uniform Ternary Source with Hamming Distance

Given a ternary source $\mathcal{X} = \{0, 1, 2\}$ with uniform distribution $P_X(0) = P_X(1) = P_X(2) = \frac{1}{3}$, we want to approximate the output distribution with i binary random variables. The distortion is measured using the hamming distance, i.e. the distance between two symbol is 0 if they are identical and 1 otherwise. Since source alphabet is ternary ($|\mathcal{X}| = 3$), the message passed from source node l to network node l has three components: $M_{S_l}^0$, $M_{S_l}^1$ and $M_{S_l}^2$. Based on Eq. (4.7), we set $\alpha_k = \alpha$ if $k \neq x_l$ because the distances from x_l to any value of k are the same, then the message equations from source node l to network node l are derived as follows:

$$\begin{aligned} M_{S_l}^k &= \exp(\gamma)/(\exp(\gamma) + 2\alpha \exp(-\gamma)), & k = x_l \\ M_{S_l}^k &= \alpha \exp(-\gamma)/(\exp(\gamma) + 2\alpha \exp(-\gamma)), & k \neq x_l \end{aligned}$$

where x_l is the l -th source symbol. γ and α are the control parameters that can be adjusted.

Note that, since the input is uniformly distributed, according to the rate distortion function, the optimal output is also uniformly distributed for any rate. This implies that we need to map i binary random variables to one ternary variable uniformly, which is apparently not possible. Therefore, we have to approximate the uniform ternary distribution. For example, suppose we use 4 random variables to approximate. We map the first 5 combinations to $\hat{X} = 0$, the next 5 combinations to $\hat{X} = 1$ and the rest 6 combinations to $\hat{X} = 2$. We denote this mapping as (5,5,6) and it is shown in Figure 4.7. The message sent from network node l to the l -th check node in i -th group is calculated by finding the marginal distribution of Y_i based on the deterministic mapping. According to Eq. (4.8) and Figure 4.7, the message passed from the l -th network node to the l -th check node in 1st group is derived by setting $q = 1$ and $i = 4$. Eq. (4.10) shows the explicit form of this message, where $M_{N_l}^0$ and $M_{N_l}^1$ need to be normalized after calculation.

Y_1	Y_2	Y_3	Y_4	\hat{X}
0	0	0	0	} 0
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	} 1
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	} 2
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	} 2
1	1	0	1	
1	1	1	0	
1	1	1	1	

Figure 4.7: The deterministic mapping for optimal output distribution $P_{\hat{X}}^*(0) = \frac{5}{16}$, $P_{\hat{X}}^*(1) = \frac{5}{16}$ and $P_{\hat{X}}^*(2) = \frac{6}{16}$.

$$\begin{aligned}
M_{N_l}^0 &= M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
M_{N_l}^1 &= M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1
\end{aligned} \tag{4.10}$$

4.3.3 Uniform Ternary Source with Non-Hamming Distance

Given a ternary source $\mathcal{X} = \{0, 1, 2\}$ with uniform distribution $P_X(0) = P_X(1) = P_X(2) = \frac{1}{3}$, we want to approximate the output distribution with i binary random variables. The distortion is measured using a non-hamming distance which is the absolute value of the symbol difference. Precisely, the distance is 0 if two symbols are identical, the distance between 0 and 1 is 1, and the distance between 0 and 2 is 2.

Since only the distortion measure is changed from the ternary hamming case, the messages passed from source node to network node still contains three components: $M_{S_l}^0$, $M_{S_l}^1$ and $M_{S_l}^2$, but with different the message equations. Based on Eq. (4.7), we let α_k be α and β if the distance between k and x_l is 1 and 2 respectively, such that α and β control the different impact of the source symbol on the reconstructed symbol at the network node to be distance 1 apart and distance 2 apart. Since we encourage the reconstructed value having small distortion with the source value, α is always set to be larger than β . The message equations are derived and shown as follows:

$$\begin{aligned} M_{S_l}^k &= \exp(\gamma)/(\exp(\gamma) + (\alpha + \beta)\exp(-\gamma)), & k = x_l \\ M_{S_l}^k &= \alpha \exp(-\gamma)/(\exp(\gamma) + (\alpha + \beta)\exp(-\gamma)), & |k - x_l| = 1 \\ M_{S_l}^k &= \beta \exp(-\gamma)/(\exp(\gamma) + (\alpha + \beta)\exp(-\gamma)), & |k - x_l| = 2 \end{aligned}$$

where x_l is the l -th source symbol. γ , α and β are the control parameters that can be adjusted.

We use 4 binary variables to approximate the output distribution. Apply Eq. (4.2), we found the optimal output distributions $P_{\hat{X}}^*$ among different rate regions and shown in Table 4.2.

Suppose the encoding rate is chosen to be 0.4 bit/symbol, according to Table 4.2, the optimal output distribution is $P_{\hat{X}}^*(0) = \frac{4}{16}$, $P_{\hat{X}}^*(1) = \frac{8}{16}$ and $P_{\hat{X}}^*(2) = \frac{4}{16}$. The deterministic mapping is shown in Figure 4.8. The message sent from network node

rate region	$(P_{\hat{X}}^*(0), P_{\hat{X}}^*(1), P_{\hat{X}}^*(2))$
0.43 - 1.58	$(\frac{5}{16}, \frac{6}{16}, \frac{5}{16})$
0.22 - 0.43	$(\frac{4}{16}, \frac{8}{16}, \frac{4}{16})$
0.13 - 0.22	$(\frac{3}{16}, \frac{10}{16}, \frac{3}{16})$
0 - 0.13	$(\frac{2}{16}, \frac{12}{16}, \frac{2}{16})$

Table 4.2: The optimal output distribution $P_{\hat{X}}^*$ along the rate distortion curve

l to the adjacent check node in i -th group is calculated based on the deterministic mapping using the similar marginalizing method in the previous two cases. According to Eq. (4.8) and Figure 4.8, the message passed from the l -th network node to the l -th check node in 1st group is derived by setting $q = 1$ and $i = 4$. Eq. (4.11) shows the explicit form of this message, where $M_{N_l^1}^0$ and $M_{N_l^1}^1$ need to be normalized after calculation.

$$\begin{aligned}
M_{N_l^1}^0 &= M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^0 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
M_{N_l^1}^1 &= M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^1 \times M_{C_l^2 \rightarrow N_l}^0 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^0 \times M_{C_l^4 \rightarrow N_l}^1 \\
&\quad + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^0 + M_{S_l}^2 \times M_{C_l^2 \rightarrow N_l}^1 \times M_{C_l^3 \rightarrow N_l}^1 \times M_{C_l^4 \rightarrow N_l}^1
\end{aligned} \tag{4.11}$$

Y_1	Y_2	Y_3	Y_4		\hat{X}
0	0	0	0	}	0
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0	}	1
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0	}	2
1	1	0	1		
1	1	1	0		
1	1	1	1		

Figure 4.8: The deterministic mapping for optimal output distribution $P_{\hat{X}}^*(0) = \frac{4}{16}$, $P_{\hat{X}}^*(1) = \frac{8}{16}$ and $P_{\hat{X}}^*(2) = \frac{4}{16}$.

4.4 Simulation Result

We examined the performance of the multilevel coding scheme for the three cases described in the previous section. The algorithms are implemented in C. The degree distribution of LDGM codes are obtained from the LPDCopt website [40] and optimized for the AWGN channel.

The simulation results are shown in Figure 4.9,4.10,4.11. We test the performances of the LDGM codes with our encoding algorithm under two different setting of block length, 1000 and 10000 respectively. The optimal output distribution is approximated using 4 random variables. Therefore, the length of the LDGM codes used in all the test cases is 4000 or 40000. The damping method is used in the message passing algorithm if the messages do not converge after 30 iterations. The decimation threshold is set to 0.9 and the number of iteration is 100 used in the encoding algorithm. For each simulation case 1000 source sequences are tested, and the simulation result are obtained by averaging over these 1000 source sequences. The resulting distortions are close to the theoretical lower bound in all these three cases, thus, validate our encoding scheme.

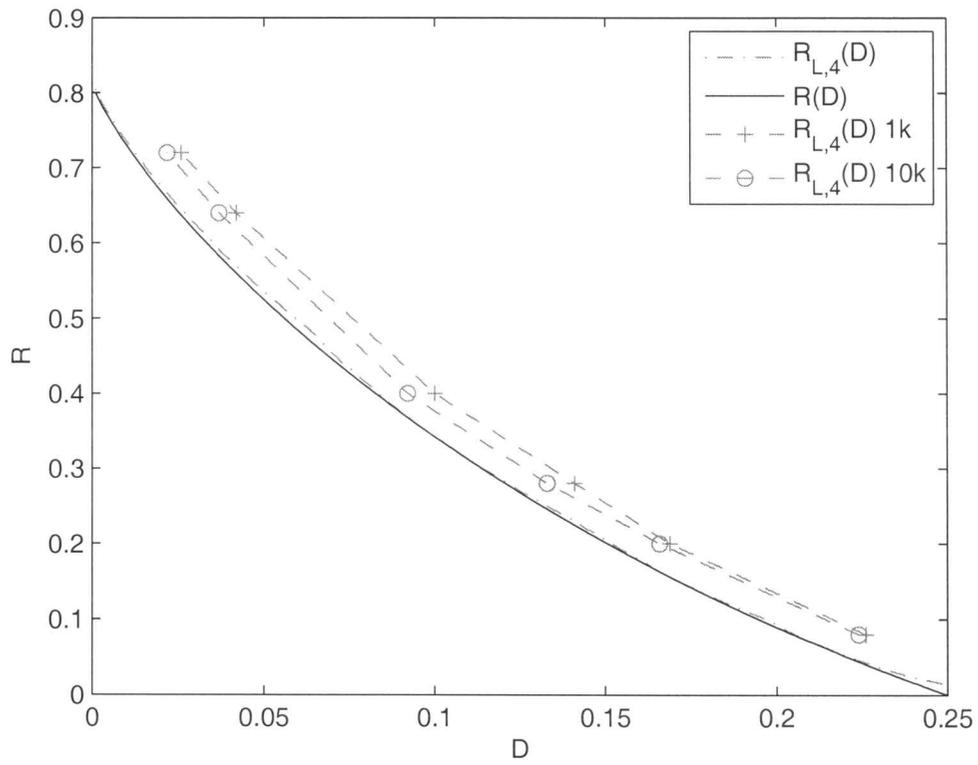


Figure 4.9: Lossy source coding of non-uniform binary source with hamming distance measure.

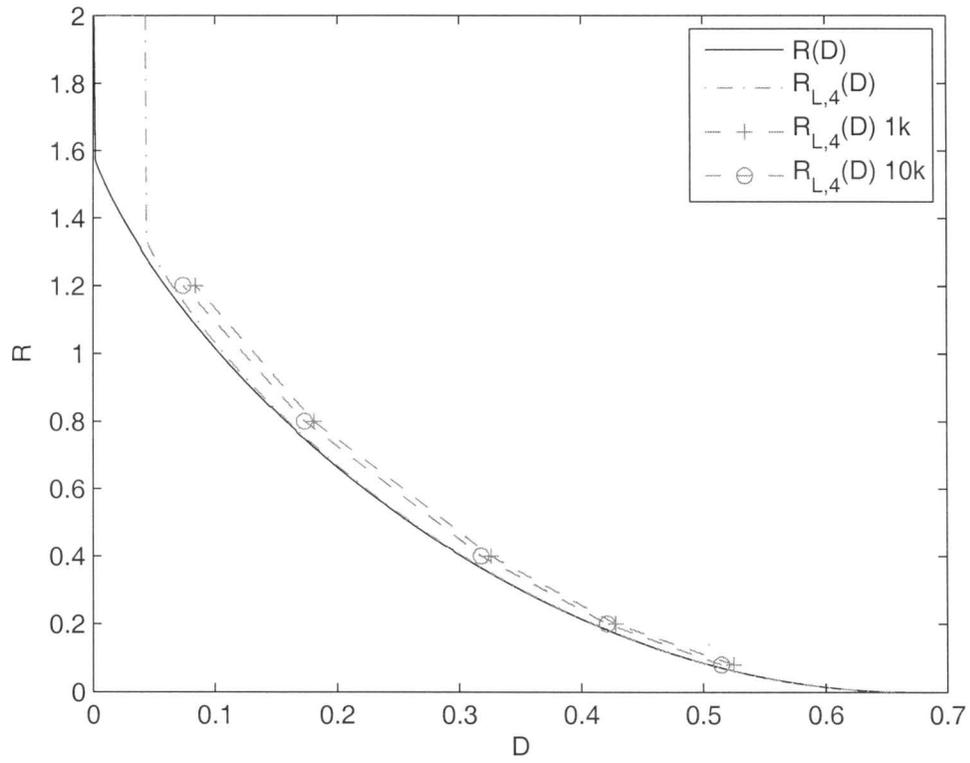


Figure 4.10: Lossy source coding of uniform ternary source with hamming distance measure.

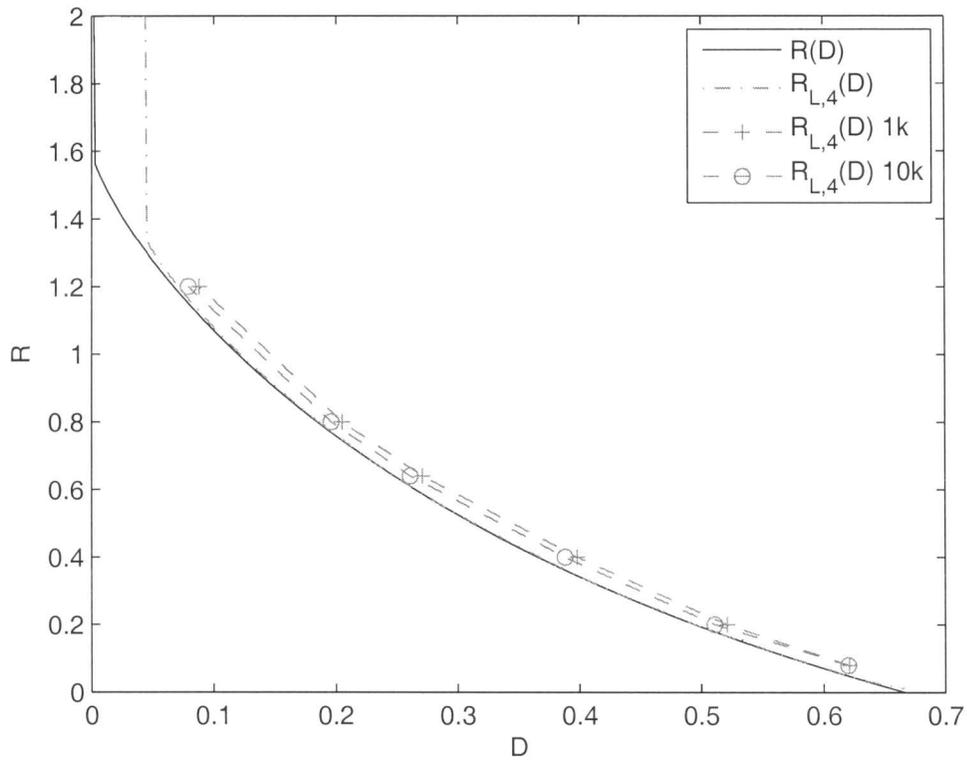


Figure 4.11: Lossy source coding of uniform ternary source with non-hamming distance measure.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we have briefly discussed LDPC and LDGM codes as well as their associated message passing algorithms. After that we present the applications of these two kinds of codes in two source coding problems. First, we introduced a new information-theoretic scheme based on source splitting for the asynchronous Slepian-Wolf problem. Combined with the source-channel correspondence result, the proposed LDPC design leads to promising performance which is validated by simulations. The advantage of the proposed method is that it does not require common randomness and super-letter construction, and it can utilize existing code design results for the corner points of S-SW problem. Second, we introduce a new multilevel coding scheme and message passing algorithm using LDGM codes based on the idea of approximating the optimal output distribution indicated by the rate-distortion theory with a uniform distribution over a (possibly) larger alphabet. The simulation results show the advantage of this proposed scheme; specifically, we can approach the rate-distortion bound closely even if the source alphabet is non-binary and the distribution is non-uniform. This

can not be accomplished using the previous LDGM codes based algorithm.

5.2 Future Work

Following this thesis, we can extend our work along several directions. Whenever an implementation contains any of two aforementioned problems as a building block, our coding scheme and codes design method can be directly applied. For example, our works can be extended in the implementation of Wyner-Ziv coding problem which is a problem of lossy source coding with side information. The optimal coding scheme for Wyner-Ziv problem is that doing the quantization (lossy source coding) of the source and side information firstly and doing the lossless coding of the quantized source with the quantized side information (Slepian-Wolf coding) secondly. Therefore, by combining our presented two applications in this thesis, we can implement the Wyner-Ziv coding for arbitrary sources. Moreover, following the same way, we also can implement the Berger-Tung coding problem in which we consider lossy source coding of correlated sources.

Appendix A

Proof of the sufficiency of linear codes in the A-SW setting

In this appendix, we prove the sufficiency of linear codes in the A-SW setting. Since the sufficiency for the first and third steps is well known, we only need to focus on the second step, more precisely as follows: encoding a length- n source Y sequence, where $k = np$ out of n of the corresponding X side information samples are available at the decoder, whose positions (the non-erased pattern within a block due to shifts of a given $T(t)$) are unknown to the encoder, but known to the decoder. We have the following theorem.

Theorem A.1. *There exists a sequence of linear codes with rate approaching $pH(Y|X) + (1 - p)H(Y)$, indexed by the code length n , with uniformly diminishing error probability for the above problem of source coding with partial decoder side information for all cyclic shifts.*

We prove this theorem using the method of types [8], and particularly the techniques in [38]. The type of a sequence $x^n \in \mathcal{X}^n$ is the distribution P_X on \mathcal{X} by

$$P_X(a) \triangleq \frac{1}{n} N(a|x^n) \text{ for every } a \in \mathcal{X},$$

where $N(a|x^n)$ is the number of occurrences of symbol a in the sequence x^n . Denote the set of types for length- n sequences in the alphabet \mathcal{X} as $\mathcal{P}_n(\mathcal{X})$. The set of length- n sequences of type P will be denoted as \mathcal{T}_P^n . Similar concepts can be defined for conditional types, but are omitted here for brevity. We need the following elementary results in [8].

Lemma A.1. *The number of different types of sequences in \mathcal{X}^n is less than $(n+1)^{|\mathcal{X}|}$, i.e., $|\mathcal{P}_n(\mathcal{X})| \leq (n+1)^{|\mathcal{X}|}$.*

Lemma A.2. *For any type P_X of sequences in \mathcal{X}^n , $|\mathcal{T}_{P_X}^n| \leq 2^{nH(P_X)}$. Similarly, for any conditional type $P_{Y|X}$ of sequences in \mathcal{Y}^n , $|\mathcal{T}_{P_{Y|X}}^n(x^n)| \leq 2^{nH(P_{Y|X})}$ for any x^n of the consistent type. Here we use the notation $H(P_{Y|X})$ to denote the conditional entropy of the given joint type.*

Let Q be an arbitrary distribution on \mathcal{X} and Q^n be the corresponding product distribution on \mathcal{X}^n . We have the following simple identity [38],

$$Q^n(x^n) = 2^{-n(D(P_X||Q)+H(P_X))} \quad (\text{A.1})$$

for any $P_X \in \mathcal{P}_n(\mathcal{X})$ and $x^n \in \mathcal{T}_{P_X}^n$.

Proof of Theorem A.1. Consider the source sequence Y^n and the side information X^k . Divide Y^n into two parts with length k and $n-k$, respectively, the first of which is aligned with the known side information sequence X^k . We associate the first part with a generic random variable Y_1 and the second part with a generic random variable Y_2 . It is understood that the sequence associated with Y_1 is of length k while the sequence associated with Y_2 is of length $n-k$.

Let us consider constructing linear encoding function $f(\cdot)$ using a $n \times l$ parity check matrix with entries independently and uniformly selected from \mathcal{Y} , where \mathcal{Y} is assumed to be a finite field. For each joint type pair

$$(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) \in (\mathcal{P}_k(\mathcal{Y} \times \mathcal{Y} \times \mathcal{X}), \mathcal{P}_{n-k}(\mathcal{Y} \times \mathcal{Y})), \quad (\text{A.2})$$

we also write the sub-type associated with (Y_1, X) as P_{Y_1X} , and write the sub-type associated with Y_2 also as P_{Y_2} .

For each joint type pair in (A.2), let $N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2})$ denote the number of sequences (y_1^k, x^k, y_2^{n-k}) with $(y_1^k, x^k) \in \mathcal{T}_{P_{Y_1X}}^k$ and $y_2^{n-k} \in \mathcal{T}_{P_{Y_2}}^{n-k}$, such that for some $(\tilde{y}_1^k, \tilde{y}_2^{n-k}) \neq (y_1^k, y_2^{n-k})$ with $y_1^k \tilde{y}_1^k x^k \in \mathcal{T}_{P_{Y_1\tilde{Y}_1X}}^k$ and $P_{y_2^{n-k} \tilde{y}_2^{n-k}} \in \mathcal{T}_{P_{Y_2\tilde{Y}_2}}^{n-k}$, and furthermore, the relation $f(y_1^k y_2^{n-k}) = f(\tilde{y}_1^k \tilde{y}_2^{n-k})$ holds. Due to the random construction of the linear function $f(\cdot)$, it is straightforward to see that for two distinct sequences, we have

$$\Pr[f(y_1^k y_2^{n-k}) = f(\tilde{y}_1^k \tilde{y}_2^{n-k})] = |\mathcal{Y}|^{-l}. \quad (\text{A.3})$$

It follows that for any joint type pair in (A.2), we have

$$\mathbb{E} N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) \stackrel{(a)}{\leq} |\mathcal{T}_{P_{Y_1X}}^k| |\mathcal{T}_{P_{Y_2}}^{n-k}| 2^{kH(P_{\tilde{Y}_1|Y_1X})} 2^{(n-k)H(P_{\tilde{Y}_2|Y_2})} |\mathcal{Y}|^{-l} \quad (\text{A.4})$$

$$\leq |\mathcal{T}_{P_{Y_1X}}^k| |\mathcal{T}_{P_{Y_2}}^{n-k}| 2^{kH(P_{\tilde{Y}_1|X})} \times 2^{(n-k)H(P_{\tilde{Y}_2})} |\mathcal{Y}|^{-l}, \quad (\text{A.5})$$

where in (a) we have used Lemma A.2, and (b) is due to conditioning reduces entropy.

By applying Markov's inequality and defining $R = \frac{l}{n} \log |\mathcal{Y}|$, we have

$$\Pr \left\{ N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) \geq |\mathcal{T}_{P_{Y_1X}}^k| |\mathcal{T}_{P_{Y_2}}^{n-k}| 2^{-n[R - \frac{k}{n}H(P_{\tilde{Y}_1|X}) - \frac{n-k}{n}H(P_{\tilde{Y}_2}) - \delta_n]} \right\} \leq 2^{-n\delta_n}. \quad (\text{A.6})$$

As afore-mentioned, there are at most a total of n side information erasure patterns (within the cyclic group), and the number of types are bounded by Lemma A.1, thus probability of the event E_0 , that for some type pair and some side information erasure pattern the condition in the brace of (A.6) is satisfied, is bounded by

$$\Pr(E_0) \leq n(k+1)^{|\mathcal{X}||\mathcal{Y}|^2} (n-k+1)^{|\mathcal{Y}|^2} 2^{-n\delta_n}, \quad (\text{A.7})$$

which is strictly less than one for sufficiently large n , by choosing δ_n appropriately such $\delta_n \rightarrow 0$ as $n \rightarrow \infty$; such a sequence of δ_n indeed exists by the δ -convention in [8]

(p. 34). It is thus seen that there exists $f(\cdot)$ such that for sufficiently large n

$$N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) \leq |\mathcal{T}_{P_{Y_1X}}^k| |\mathcal{T}_{P_{Y_2}}^{n-k}| 2^{-n[R - \frac{k}{n}H(P_{\tilde{Y}_1|X}) - \frac{n-k}{n}H(P_{\tilde{Y}_2}) - \delta_n]}, \quad (\text{A.8})$$

for all joint type pairs in (A.2) and all possible erasure patterns. By observing

$$N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) \leq |\mathcal{T}_{P_{Y_1X}}^k| |\mathcal{T}_{P_{Y_2}}^{n-k}|. \quad (\text{A.9})$$

we may write without loss of generality that

$$N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) \leq |\mathcal{T}_{P_{Y_1X}}^k| |\mathcal{T}_{P_{Y_2}}^{n-k}| 2^{-n[R - \frac{k}{n}H(P_{\tilde{Y}_1|X}) - \frac{n-k}{n}H(P_{\tilde{Y}_2}) - \delta_n]^+}. \quad (\text{A.10})$$

We shall use the linear code $f(\cdot)$ with the above property (A.10) as the encoding function. The decoder now chooses $(\hat{y}_1^k, \hat{y}_2^{n-k})$ such that $\frac{k}{n}H(P_{\tilde{Y}_1|X}) + \frac{n-k}{n}H(P_{\tilde{Y}_2})$ is minimized. The decoding error probability can be bounded as

$$P_e \stackrel{(a)}{\leq} \sum N_f(P_{Y_1\tilde{Y}_1X}, P_{Y_2\tilde{Y}_2}) 2^{-k[D(P_{Y_1X} \| Q_{YX}) + H(P_{Y_1X})]} \times 2^{-(n-k)[D(P_{Y_2} \| Q_Y) + H(P_{Y_2})]} \quad (\text{A.11})$$

$$\stackrel{(b)}{\leq} \sum 2^{-n[R - \frac{k}{n}H(P_{\tilde{Y}_1|X}) - \frac{n-k}{n}H(P_{\tilde{Y}_2}) - \delta_n]^+} \times 2^{-kD(P_{Y_1X} \| Q_{YX})} 2^{-(n-k)D(P_{Y_2} \| Q_Y)} \quad (\text{A.12})$$

$$\stackrel{(c)}{\leq} (k+1)^{|\mathcal{X}||\mathcal{Y}|^2} (n-k+1)^{|\mathcal{Y}|^2} 2^{-nE_1}, \quad (\text{A.13})$$

where (a) is by (A.1) and by taking summation over joint type pairs in (A.2) such that

$$\frac{k}{n}H(P_{\tilde{Y}_1|X}) + \frac{n-k}{n}H(P_{\tilde{Y}_2}) \leq \frac{k}{n}H(P_{Y_1|X}) + \frac{n-k}{n}H(P_{Y_2}), \quad (\text{A.14})$$

because only this case may lead to a decoding error; in (b) we used (A.10) and Lemma A.2; in (c) we define

$$E_1 = \min \left[\left| R - \frac{k}{n}H(P_{\tilde{Y}_1|X}) - \frac{n-k}{n}H(P_{\tilde{Y}_2}) - \delta_n \right|^+ + \frac{k}{n}D(P_{Y_1X} \| Q_{YX}) + \frac{n-k}{n}D(P_{Y_2} \| Q_Y) \right], \quad (\text{A.15})$$

and the minimization is over the joint type pairs in (A.2) such that (A.14) holds. By choosing R such that

$$R > \frac{k}{n}H(Y|X) + \frac{n-k}{n}H(Y), \quad (\text{A.16})$$

it is clear that E_1 is bounded below from zero when n is sufficiently large, which completes the proof. \square

Appendix B

Proof of the channel correspondence to the Slepian-Wolf problem with side information

X and Y are two correlated sources with joint distribution $P_{XY}(x, y)$ in the alphabet \mathcal{X} and \mathcal{Y} . We assume the \mathcal{X} is in $GF(K)$ and $\mathcal{Y} = \{1, 2, \dots, J\}$. The linear code we will be using is in $GF(K)$, and is specified by the parity check matrix H . From the given distribution P_{XY} , we construct a channel of one input U with $\mathcal{U} = \mathcal{X}$, and two outputs $V = [V_1, V_2]$ with $\mathcal{V}_1 = \mathcal{X}$ and $\mathcal{V}_2 = \mathcal{Y}$, such that

$$V = [V_1, V_2] = [U + W_1, W_2] \tag{B.1}$$

where the addition is in $GF(K)$. The distribution of the noise W is given by

$$P_W(W_1 = a, W_2 = b) = P_{XY}(a, b) \tag{B.2}$$

for any $(a, b) \in \mathcal{X} \times \mathcal{Y}$.

The encoder of the S-W codes forms the syndrome by $s^m = H_{m \times n} \cdot x^n$, and the decoder notes the syndrome s^m and y^n such that it tries to decode x^n . The goal now

is to find the decoding error using such a scheme.

Let us consider the channel coding problem with afore given channel. We can form the syndrome at the decoder by using

$$s^m = H_{m \times n} \cdot (U^n + W_1^n) = H_{m \times n} \cdot U^n + H_{m \times n} \cdot W_1^n = H_{m \times n} \cdot W_1^n \quad (\text{B.3})$$

let us use syndrome decoding in this case, and the problem becomes finding the most likely noise component W_1^n , given its syndrome and the second component W_2^n . Such a decoding algorithm can be formed that for each pair of (W_1^n, W_2^n) , a decoded \hat{w}_1^n is chosen. An error occurs if $\hat{w}_1^n \neq w_1^n$.

Notice that for any pair of (x^n, y^n) , we know the syndrome of x^n , and the second component y^n . Then we can run the syndrome decoding algorithm for the channel coding. An error occurs if $\hat{x}^n \neq x^n$. Since the joint distribution $P_{XY}(x^n, y^n)$ is the same as $P_W(x^n, y^n)$ as specified, the probability of error is the same as follows

$$P_e^{S-W} = \sum_{(x^n, y^n)} P_{XY}(x^n, y^n) l_{err}(x^n, y^n) = \sum_{(w_1^n, w_2^n)} P_W(w_1^n, w_2^n) l_{err}(w_1^n, w_2^n) = P_e^C \quad (\text{B.4})$$

where l_{err} is the indicator function that an error occurs for this specific pair if using syndrome decoding algorithm. It shows that the error probability using this code on Slepian-Wolf problem is exactly the same as in the channel coding problem if syndrome decoding is used. Thus, a good channel coding algorithm is also good in the Slepian-Wolf setting.

Now we shall show that the capacity of this channel is achieved by the uniform distribution.

$$\begin{aligned} I(U; V_1, V_2) &= H(V_1, V_2) - H(W_1, W_2) = H(W_1 + U|W_2) - H(W_1|W_2) \\ &= H(W_1 + U|W_2) - H(X|Y) \leq \log |\mathcal{U}| - H(X|Y). \end{aligned} \quad (\text{B.5})$$

if U is a uniform distribution, then

$$\log |\mathcal{U}| = H(U|W_1, W_2) = H(U + W_1|W_1, W_2) \leq H(U + W_1|W_2) \leq \log |\mathcal{U}|. \quad (\text{B.6})$$

Thus, $H(W_1 + U|W_2) = \log |\mathcal{U}|$ and subsequently

$$C = \max_{P_U} I(U; V_1, V_2) = \log |\mathcal{U}| - H(X|Y) \quad (\text{B.7})$$

Now it is clear that the rate of the Slepian-Wolf code with a capacity achieving code for the afore given channel is $\log |\mathcal{U}| - C = H(X|Y)$.

Appendix C

Discretized density evolution algorithm and linear programming

The expected behavior of belief propagation when used to decode LDPC codes can be analyzed by tracing the density function of the messages under the tree assumption of the codes construction. This idea is well captured and presented formally in an algorithm called "density evolution" (DE) algorithm [31][33]. This algorithm gives an expectation of the performance of LDPC codes used in channel coding problem. Urbanke *et al.* [34] developed an improved algorithm called "discretized density evolution" (discretized DE) algorithm. The discretized DE algorithm shares the same idea of DE algorithm, however, reduces the computational complexity. In the following we will briefly discuss the discretized DE algorithm.

The degree polynomials $\lambda(x)$ and $\rho(x)$ (see section 2.1.1) are given to specify a random ensemble of irregular LDPC codes. In discretized DE algorithm, all the messages are quantized into the closest level with the quantization interval Δ .

Based on the messages calculating equations (see Eq(2.4) and Eq(2.5)), the probability mass function (pmf) of message from variable node to check node, p_v , is

calculated by

$$p_v = p_{u0} * (\otimes^{d_v-1} p_u) \quad (\text{C.1})$$

where p_{u0} is the pmf of the message from the observation bit that associated with the variable node. This pmf, p_{u0} , is the input to this algorithm and calculated by the given channel parameters. The pmfs given in section 3.4 are p_{u0} s. P_u denotes the pmf of the message from check node to variable node. $*$ and \otimes^{d_v-1} denote the discrete convolution and $d_v - 1$ times discrete convolution respectively.

The pmf p_u is calculated by

$$p_u = \mathcal{R}^{d_c-1}(p_v) \quad (\text{C.2})$$

where \mathcal{R} is a complicated operation on p_v .

Combining with the degree polynomials $\lambda(x)$ and $\rho(x)$, the discretized DE formula is

$$p_u^{(l+1)} = \rho(p_{u0} * \lambda(p_u^{(l)})) \quad (\text{C.3})$$

where l is the number of iteration and the initial value of p_u is that the probability of $u = 0$ is 1.

The algorithm is run by assuming the code sent is an all zero codeword. Thus, it will stop when the probability of message u larger than 0 is 1 or a number sufficiently close to 1 after several iterations. Then, the code specified by the degree polynomials $\lambda(x)$ and $\rho(x)$ is expected to have good performance when transmitting through this channel. The details of the algorithm can be found in [34]. The discretized DE is an algorithm to test the performance of a pair of degree distributions. Then how to design the degree distributions that have a good performance under the test of discretized DE is described in [35] using linear programming.

This linear programming based method is used to optimize the node degree distributions iteratively. Initially, we choose the node degree distributions represented

by $\lambda(x)$ and $\rho(x)$ that result the code rate lower than the desired rate or the capacity of the channel. Then run the linear programming to modify the degree distributions and increase the code rate according to the following constraints:

- the new $\lambda(x)$ must be a distribution so that $\lambda(1) = 1$ and $\lambda_i \geq 0$ for $2 \leq i \leq d_v$.
- the new $\lambda(x)$ has not significantly difference from the old one.
- the new $\lambda(x)$ must produce smaller probability of error than the old one.

Combining these two recursive methods, given the channel characteristics, we can design the good node degree distributions following the aforementioned procedure.

Appendix D

Calculate the channel transition probability in the dual channel model

Suppose X and Y are two joint distributed sources with $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and joint distribution $P_{XY}(x, y)$, we calculate the probability $P_{U|V}(u|v)$ in the channel model shown in Figure 3.3 of Chapter 3 as follows:

$$\begin{aligned} P_{U|V}(u = 0|(v_1 = 0, v_2 = 0)) &= \frac{P_{UV}(u = 0, (v_1 = 0, v_2 = 0))}{P_V((v_1 = 0, v_2 = 0))} \\ &= \frac{P_U(u = 0) \cdot P_{V|U}((v_1 = 0, v_2 = 0)|u = 0)}{P_V((v_1 = 0, v_2 = 0))} \end{aligned} \tag{D.1}$$

Since $P_U(u = 0) = 0.5$

$$P_{U|V}(u = 0|(v_1 = 0, v_2 = 0)) = \frac{0.5 \cdot P_{V|U}((v_1 = 0, v_2 = 0)|u = 0)}{P_V((v_1 = 0, v_2 = 0))} \tag{D.2}$$

Since U is given, we know that $v_1 = x \oplus u$ and $v_2 = y$. Therefore, $x = v_1 \oplus u$ and $y = v_2$. we have:

$$P_{V|U}((v_1, v_2)|u) = P_{XY}(x = v_1 \oplus u, y = v_2)$$

So that

$$P_{V|U}((v_1 = 0, v_2 = 0)|u = 0) = P_{XY}(x = 0, y = 0) \quad (\text{D.3})$$

Since v_1 and v_2 are independent and $P_{V_1}(v_1 = 0) = 0.5$, we have:

$$\begin{aligned} P_V(v_1 = 0, v_2 = 0) &= P_{V_1}(v_1 = 0) \cdot P_{V_2}(v_2 = 0) \\ &= 0.5 \cdot P_{V_2}(v_2 = 0) \end{aligned} \quad (\text{D.4})$$

Now plug Eq.(D.3) and Eq.(D.4) into Eq.(D.2), we have:

$$\begin{aligned} P_{U|V}(u = 0|(v_1 = 0, v_2 = 0)) &= \frac{0.5 \cdot P_{XY}(x = 0, y = 0)}{0.5 \cdot P_{V_2}(v_2 = 0)} \\ &= \frac{P_{XY}(x = 0, y = 0)}{P_Y(y = 0)} \end{aligned} \quad (\text{D.5})$$

Given the joint probability $P_{XY}(x, y)$, we can find $P_Y(y)$ by marginalizing over X .

$$P_Y(y) = \sum_{x \in X} P_{XY}(x, y) \quad (\text{D.6})$$

Finally we find that $P_{U|V}(u = 0|(v_1 = 0, v_2 = 0)) = P_{X|Y}(x = 0|y = 0)$. Using the same procedure, we can derive all the possible probability $P_{U|V}(u|(v_1, v_2))$.

Bibliography

- [1] R. G. Gallager, *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.
- [2] M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533-547, Sept. 1981.
- [3] C.E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, part 4:142-163, 1959.
- [4] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp.471-480, Jul. 1973.
- [5] R. F. Ahlswede and J. Korner, "Source coding with side information and a converse for degraded broadcast channels," *IEEE Trans. Inform. Theory*, vol. 21, no. 6, pp. 629-637, Nov. 1975.
- [6] T. M. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inform. Theory*, vol. 21, no. 2, pp. 226-228, Mar. 1975.
- [7] F. Willems, "Totally asynchronous Slepian-Wolf data compression", *IEEE Transactions on Inform. Theory*, vol. 34, no. 1, pp. 35-44, Jan. 1988.

-
- [8] I. Csiszar and J. Korner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic, 1981.
- [9] I. Csiszar and J. Korner, "Towards a general theory of source networks," *IEEE Transactions on Inform. Theory*, vol. 26, no. 2, pp. 155–165, Mar. 1980.
- [10] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communications Letters*, 5:417-419, Oct.2001.
- [11] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," *IEEE GLOBECOM*, pp.1400-1404, Nov. 2001.
- [12] A. Aaron and B. Girod, "Compression with side information using turbo codes," *Data Compression Conference*, pp. 252-261, Apr. 2002.
- [13] A. D. Liveris, Z. Xiong, C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, pp. 440-442, Oct. 2002.
- [14] D. Schonbery, S. S. Pradhan, and K. Ramchandran, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," *Data Compression Conference*, pp. 292, 2004.
- [15] T. P. Coleman, A. H. Lee, M. Medard, M. Effros, "Low-complexity approaches to Slepian-Wolf near-lossless distributed data compression," *IEEE Trans. Inform. Theory*, vol. 52, no. 8, pp. 3546-3561, Aug. 2006.
- [16] J. Chen, D.-k He, E.-h Yang, "On the codebook-level duality between Slepian-Wolf Coding and channel coding," *Information Theory and Applications Workshop*, 2007.

- [17] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, no. 1, pp. 2–10, Jan. 1974.
- [18] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.
- [19] B. Rimoldi and R. Urbanke, "Asynchronous Slepian-Wolf coding via source-splitting," *IEEE Int.Symp.on Info.Theory*, Ulm, Germany, July 1997.
- [20] A. Braunstein, M. Mézard and R. Zecchina, "Survey propagation: an algorithm for satisfiability," *Random Structures and Algorithms*, vol. 27, pp. 201-206, Sep. 2005.
- [21] E. Martinian, and M. Wainwright, "Low density codes achieve the rate-distortion bound," *Proc. Data Compression Conference*, pp. 153-162, Mar. 28-30, 2006.
- [22] M. Wainwright and E. Maneva, "Lossy source encoding via message-passing and decimation over generalized codewords of ldgm codes," in *Proceedings of International Symposium on Information Theory*, pp. 1493-1497, Sept. 4-9, 2005.
- [23] S. Cilibertia, M. Mézard, and R. Zecchina, "Lossy data compression with random gates," *Phys. Rev. Lett.*, vol. 95, no. 038701, 2005.
- [24] G. Caire, S. Shamai, and S. Verdú, "A new data compression algorithm for sources with memory based on error correcting," in *Proceedings of Information Theory Workshop*, pp. 291-295. Mar. 31 - Apr. 4, 2003.
- [25] E. Martinian and J. Yedidia, "Iterative quantization using codes on graphs," *Proc. Allerton Conf. Comm., Control, and Computing*, Monticello, IL, Oct. 2003.

- [26] Y. Matsunaga and H. Yamamoto, "A coding theorem for lossy data compression by ldpc codes," in *Proceedings of International Symposium on Information Theory*, p. 461, Jun. 2002.
- [27] T. Filler and J. Fridrich, "Binary quantization using belief propagation with decimation over factor graphs of ldgm codes," *Proc. Allerton Conf. Comm., Control, and Computing*, Monticello, IL, Sept. 2007.
- [28] J. Chen, D.-k He, and A. Jagmohan, "Achieving the rate-distortion bound with linear codes," in *Proceedings of Information Theory Workshop*, pp. 662-667 Sept. 2-6 2007.
- [29] A. Braunstein, M. Mézard, M. Weigt, and R. Zecchina, "Constraint satisfaction by survey propagation," *Phys. Rev. Lett.*, vol. 89, no. 268701, 2002.
- [30] A. Braunstein, M. Mézard, and R. Zecchina, "Survey propagation: an algorithm for satisfiability," *Random Structures and Algorithms*, vol. 27, no. 2, pp. 201-226, Mar. 2005.
- [31] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619-637, Feb. 2001.
- [32] T. Murayama, "Statistical mechanics of linear compression codes in network communication," *Europhysics Lett.*, preprint, 2001.
- [33] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, 2001.

- [34] S.-Y. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, pp. 58-60, Feb. 2001.
- [35] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA, 2000.
- [36] T. M. Cover, J. A. Thomas, *Elements of Information Theory*, New York:Wiley, 1991.
- [37] V. Stankovic, A. D. Liveris, Z. Xiong and C. N. Georghiades, "On code design for the Slepian-Wolf problem and lossless multiterminal networks," *IEEE Transactions on Information Theory*, vol. 52, no. 4, Apr. 2006.
- [38] I. Csiszar, "Linear codes for sources and source networks: error exponents, universal coding," *IEEE Transactions on Information Theory*, vol. 28, no. 4, pp. 585-592, Jul. 1982.
- [39] Z. Sun, M. Shao, J. Chen, K.M. Wong and X. Wu, "Lossy source coding for non-uniform sources using LDGM codes," *Proceedings of 24th Biennial Symposium on*, 24-26 June 2008 Page(s):76 - 79
- [40] <http://lthcwww.epfl.ch/research/ldpcopt>