Linear Interactive Encoding and Decoding Schemes for Lossless Source Coding with Decoder Only Side Information

by

Jin Meng

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Near lossless source coding with side information only at the decoder, was first considered by Slepian and Wolf in 1970s, and rediscovered recently due to applications such as sensor network and distributed video coding. Suppose X is a source and Y is the side information. The coding scheme proposed by Slepian and Wolf , called SW coding, in which information only flows from the encoder to the decoder, was shown to achieve the rate $H(X|Y)$ asymptotically for stationary ergodic source pairs. As $H(X|Y)$ is the minimum achievable rate even if the encoder is informed of the side information Y, this implies the optimality of SW coding for any stationary ergodic source-side information pairs. However, it is shown by Yang and He that SW coding can not achieve this rate for most of non-ergodic source-side information pairs.

Recently, a new source coding paradigm called interactive encoding and decoding(IED) was proposed for near lossless coding with side information only at the decoder. In such paradigm, information flows in both ways, from the encoder to the decoder and vice verse, and the behaviour of the encoder also relies on the bits sent back from the decoder. The compression rate of an IED scheme is defined as average number of bits per symbol exchanged between the encoder and the decoder. In contrary to SW coding, IED can achieve the rate of $H(X|Y)$ for any stationary source-side information pairs, ergodic or not. Also it was shown that for memoryless source-side information pairs, IED schemes achieve better redundancy performance than SW coding schemes. And universal IED can be built coupled with classical universal codes for any stationary ergodic source-side information pairs, while it is well known that there does not exist a universal SW coding scheme.

The results reviewed above show that IED schemes are much more appealing than SW coding schemes to applications where the interaction between the encoder and the decoder is possible. However, the IED schemes proposed by Yang and He do not have an intrinsic structure that is amenable to design and implement in practice. Towards practical design, we restrict the encoding method to linear block codes, resulting in linear IED schemes. It is then shown that this restriction will not undermine the asymptotical performance of IED, in the sense that a sequence of linear IED schemes can be always found for any stationary ergodic source-side information pair (X, Y) to achieve the rate of conditional entropy $H(X|Y)$. Moreover, the result can also be extended to non-ergodic source-side information pairs.

Another step of practical design of IED schemes is to make the computational complexity incurred by encoding and decoding feasible. In the framework of linear IED, a scheme can be conveniently described by parity check matrices. It can be easily observed that density of these matrices (measured as the average number of non-zero entries) is directly related to the scheme's encoding and decoding complexity: the complexity increases as the density increases. Further, we get an interesting trade-off between the density of the associated parity check matrices and the resulting symbol error probability. Our analysis reveals that as long as $\epsilon_n p_n^* = \Omega(\log n/n)$, where ϵ_n and p_n^* are real numbers, one can always construct a

sequence of universal linear IED schemes $\{\mathcal{I}_n\}$ such that the average density of the parity check matrices associated with $\mathcal{I}_n$ is concentrated around $(|\mathcal{X}|-1)p_n^*$, and the resulting symbol error probability is upper bounded by $\epsilon_n + o(\epsilon_n)$. In addition, if some modification applies to linear IED schemes, it can be shown that p_n^* can be as low as $\frac{\log n}{n}$ while the symbol error probability goes to zero asymptotically.

To implement the idea of linear IED and follow the instinct provided by the result above, Low Density Parity Check(LDPC) codes and Belief Propagation(BP) decoding are utilized. Considering incremental encoding is needed in IED, a successive LDPC code based on syndrome splitting is proposed, and the splitting rule is optimized according to density evolution results. Moreover, as existing BP decoding algorithms can only apply to limited kinds of correlation between the source X and the side information Y, a new BP decoding algorithm is proposed, which applies to the case where the correlation between Y and X can be modelled as a finite state channel. It then can be shown that the existing BP algorithms, which apply to hidden markov state channels, such as GE channels and Markov Modulated Channels, are the special cases of this new algorithm. Finally, simulation results show that linear IED schemes are indeed superior to SW coding schemes.

Acknowledgements

Firstly and most importantly, I would like to express my sincerest gratitude to my supervisor Professor En-hui Yang, for his guidance and insights throughout this thesis and my education at University of Waterloo. His mathematical intuition and engineering experience is the great beacon, guiding me in study and research throughout my master period. Without his ultra-prompt and ultra-patient help, this thesis would not be possible.

I wish to thank Dr. Da-ke He, for his insightful suggestion and ultra-useful discussion. Especially for my paper submitted to alerton conference, his numerous proofreading and modification makes that paper possible.

I also wish to thank two readers of my thesis, Professor Liang-Liang Xie and Professor Mohamed Oussama Damen, for their invaluable suggesstion, which makes this thesis much better.

I am extremely grateful to the lab-mates in Multimedia Communication Lab, Dr. Haiquan Wang, Dr. Wei Sun, Dr. Xiang Yu, Dr. Xudong Ma, Miss Lin Zheng, Miss Jiao Wang, Mr. Yuhan Zhou, Mr. Krzysztof Michal Hebel. Their support and discussion are the invaluable resource of my improvement and happiness.

I am also deeply committed to my family. I thank my father and mother, for their unselfish love and constant understanding. The support of my wife, Yan Liu, is the part of reason that I made through master period. I would also like to thank my father-in-law and mother-in-law, for their considerate help during those years. Any words become meaningless when used to express my appreciation for them.

Any acknowledgement ends up with the incomplete list inevitably. Still, I wish to thank those professors, who are the instructors of class I have attended, and all the other friends of mine.

Dedication

To my parents, my wife, and all of my friends.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Classic Source Coding

1.1.1 Source Code, Compression Rate and Redundancy

The purpose of source coding, or data compression is to remove the redundancy embedded in the data and reduce the size of storage needed for the data. Usually the system of source code consists of two parts, the *encoder*, which maps the original data to a finite-length string of certain set, i.e. the *codeword*, and the *decoder*, which maps the codeword back to the domain of original data. From theoretical point of view, the data are usually modelled as the random variable, then the formal definition of *source code* is given below:

Definition 1.1.1. *A source code C for a random variable X is a mapping pair (f, g). f, which denotes encoder, is a mapping from $\mathcal{X}$, the range of X, to $\mathcal{D}^*$, the set of finite-length strings of symbols from a D-ary alphabet, while g, denoting decoder, is mapping from $\mathcal{D}^*$ to $\mathcal{X}$. Let $f(x)$ denote the codeword corresponding to x, $l(x)$ denote the length of $f(x)$, and $\hat{x}$ denote the output of decoder.*

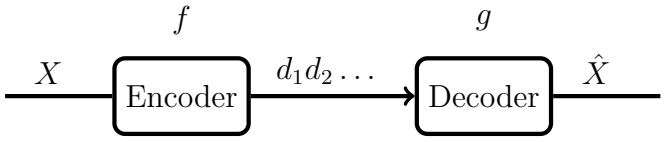

Figure 1.1: Source Code

To qualify the efficiency of source code, *expected length* is usually used.

Definition 1.1.2. *[3] The expected length $L(C)$ of a source code $C = \{f, g\}$ for random variable X with probability mass function $p(x)$ is given by*

$$\begin{aligned} L(C) &= E(f(X)) \\ &= \sum_{x \in \mathcal{X}} p(x) l(x) \end{aligned} \tag{1.1}$$

In many cases, instead of encoding and decoding one symbol, a source code C could target to a block of symbols. Formally, let n denote the length of one block, called *block length*. Then $C = \{f, g\}$, where f is a mapping from $\mathcal{X}^n$ to $\mathcal{D}^*$, while g is a mapping from $\mathcal{D}^*$ to $\mathcal{X}^n$. And the original data are denoted by $X = (X_1, X_2, \ldots, X_n)$. To better reflect the efficiency of source codes in such scenario, *compression rate* R_n is defined as expected length of the codeword per symbol.

$$R_n = \frac{1}{n} E(l(X_1, X_2, \ldots, X_n)) \tag{1.2}$$

To get block-version source code, we just have to manipulate the definition of random variable X.

Beside removing redundancy of the data, we also do not expect to lose fidelity of the data, which means that the *distortion* between the original data x and the constructed data $\hat{x}$ should satisfy some constraints. One of special cases, involved in applications widely, is the source code with lossless reconstruction. That is, $\hat{X} = X$ with probability one. In this setup, it is a well-known fact due to **First Shannon Theory** that

$$L(C) \geq H(X) \tag{1.3}$$

Therefore the redundancy of source code C is defined as

$$\mathcal{R}(C) = L(C) - H(X) \tag{1.4}$$

For block-version source codes,

$$R_n \geq \frac{1}{n} H(X_1, X_2, \ldots, X_n) \tag{1.5}$$

If the source is stationary ergodic, then the limit

$$\lim \frac{1}{n} H(X_1, X_2, \ldots, X_n) \tag{1.6}$$

exists, defined as $H(X)$. Accordingly, the *redundancy* is defined as

$$\mathcal{R}_n(C) = R_n - H(X) \tag{1.7}$$

The subscript n indicates that $\mathcal{R}_n(C)$ is a function of n.

1.1.2 Uniquely Decodable Code, Prefix Code and Kraft Inequality

In some application, even $\hat{X} = X$ with probability one is not enough. Instead, it requires $\forall x \in \mathcal{X}, \hat{x} = x$. In the other word, the encode mapping has to be one-to-one mapping.

Definition 1.1.3. *[3] A code $C = \{f, g\}$ is said to be nonsingular if every element of the range of X maps into a different string in $\mathcal{D}^*$; that is,*

$$x \neq x' \Rightarrow f(x) \neq f(x') \tag{1.8}$$

In this case, the data can be reconstructed exactly if $g = f^{-1}$ in the sense that $\forall s \in \mathcal{D}^*$, if $\exists x \in \mathcal{X}$ s.t. $s = f(x)$, then $g(s) = x$.

So far we only consider a single value of X. For sequences of symbols X, we have to extend the definition of source code.

Definition 1.1.4. *[3] The extension $C^* = \{f^*, g^*\}$ of a code $C = \{f, g\}$, where f^* is the mapping from finite-length strings of $\mathcal{X}$, to finite-length strings of $\mathcal{D}$, defined by*

$$f^*(x_1 x_2 \dots x_n) = f(x_1) f(x_2) \dots f(x_n) \tag{1.9}$$

where $f(x_1)f(x_2)\dots f(x_n)$ indicates the concatenation of the corresponding codewords, and g^* is a mapping from finite-length strings of $\mathcal{D}$ to finite-length strings of $\mathcal{X}$.

For convenience, we will sometimes write $x_m x_{m+1} \dots x_n$ as x_m^n, where $m < n$ are two integers, and x_1^n simply as x^n. Also, we have to extend the definition of *non-singularity*.

Definition 1.1.5. *[3] A code is called uniquely decodable if its extension is non-singular.*

Now we say a string A is a *prefix* of another, called B if B can be written as AC, concatenation of A and C, for some strings C.

Definition 1.1.6. *[3] A code is said to be prefix code if no codeword is a prefix of any other codeword.*

If no codeword is a prefix of another, a codeword in *prefix code* can be decoded once the end of this codeword is recognized, without referring future codewords. Therefore, the prefix code is also called the *instantaneous code*. In the contrary, if the code is not prefix, sometimes the entire string has to be considered when to decode.

There is a well-known inequality which applies to the prefix codes, called **Kraft inequality**.[3]

$$\sum_{x \in \mathcal{X}} |\mathcal{D}|^{l(x)} \leq 1 \tag{1.10}$$

where $|\cdot|$ denotes the cardinality. This inequality is then shown to be true for any *uniquely decodable* source code.

1.1.3 Universal Source Coding

Many source codes target to the sources with particular probability distribution, such as Huffman code. However, this approach may fail in certain scenarios. In one hand, probability distribution of source may be unknown. Instead, all we know is a class of distribution. Under this circumstance, a two-round coding can be applied, while the distribution is estimated during the first round, and the source is encoded during the second. Still it is much more advantageous to do one-round coding, estimating the distribution on the fly. Adaptive Arithmetic code is one of the examples. In the other hand, there may be no probability distribution underlying the data, which is just an individual sequence. For instance, Lempel Ziv code[51, 52] and Grammar based code[22, 50] are two of codes targeting to this scenario. Therefore, the *universality* is one of important aspect of source codes when applied to practice.

1.1.4 Source Coding with Side Information at Both Encoder and Decoder

For some practical situations, a sequence of the data, which is correlated with source, may be available for the encoder and the decoder, called *side information*. In such case, side information will be helpful to compress the source, and the definition of source codes is modified as

Definition 1.1.7. *A source code C for a random variable X with side information Y is a mapping pair (f, g). f, which denotes encoder, is a mapping from $\mathcal{X} \times \mathcal{Y}$, to $\mathcal{D}^*$, while g, denoting decoder, is mapping from $\mathcal{D}^* \times \mathcal{Y}$ to $\mathcal{X}$.*

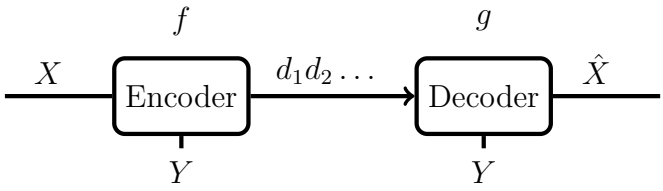

Figure 1.2: Source Code with Side Information at the Encoder and the Decoder

The definitions, extensions and properties of source code without side information in section 1.1.1 1.1.2 and 1.1.3 apply here similarly. The intuition is that with side information Y available, source codes can be designed targeting to the conditional statistics of the source X given Y, instead of statistics of the source X alone. Specifically, in the case of the lossless coding, according to **Shannon Theory**, the expected length of code $L(C)$ must satisfy

$$L(C) \geq H(X|Y) \tag{1.11}$$

For block version, the compression rate R_n must satisfy

$$R_n \geq \frac{1}{n} H(X^n|Y^n) \tag{1.12}$$

For stationary ergodic source pair, the limit

$$\lim_{n\to\infty} \frac{1}{n} H(X^n|Y^n) \tag{1.13}$$

exists, which is also the least rate that code scheme can possibly approach. As long as X and Y are not independent, $H(X|Y) < H(X)$, from which it can be seen that the side information indeed improves the efficiency of source code. Universal source code for such scenario can also be found in literature[49].

1.2 Distributed Source Coding

1.2.1 Model and Theoretical Result

Distributed Source Coding is to compress the correlated source, which are physically separated. One of its applications is sensor network, where sensors capture the image sampled in the same region but from different positions, and send the image to one common station. The formal definition of distributed source coding is addressed below:

Definition 1.2.1. *A $((2^{nR_X}, 2^{nR_Y}), n)$ distributed source code for the joint source (X, Y) consists of two encoder maps,*

$$f_1 : \mathcal{X}^n \to \{1, 2, \ldots, 2^{nR_X}\} \tag{1.14}$$
$$f_2 : \mathcal{Y}^n \to \{1, 2, \ldots, 2^{nR_Y}\} \tag{1.15}$$

and a decoder map,

$$g : \{1, 2, \ldots, 2^{nR_X}\} \times \{1, 2, \ldots, 2^{nR_Y}\} \to \mathcal{X}^n \times \mathcal{Y}^n \tag{1.16}$$

Here $f_1(X^n)$ is the index corresponding to X^n, $f_2(Y^n)$ is index corresponding to Y^n, and (R_X, R_Y) is the rate pair of the code.

The coding problem can be near lossless or lossy, which corresponds to *Slepian Wolf coding* or *Wyner-Ziv coding*. As the scope of this thesis only involves near lossless coding, we mainly focus on near lossless distributed source coding, i.e. Slepian-Wolf(SW) Coding.

"Near lossless" means that reconstructed data equals to the original data with high probability. Therefore, besides rate pair (R_X, R_Y), the probability that reconstructed data is not the same as original data is another concern.

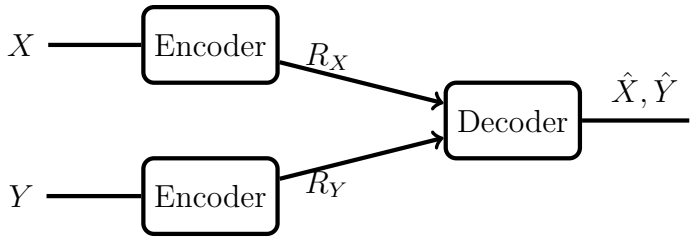

Figure 1.3: Distributed Source Coding

Definition 1.2.2. *The probability of error for a distributed source code is defined as*

$$P_e^{(n)} = P(g(f_1(X^n), f_2(X^n)) \neq (X^n, Y^n)) \tag{1.17}$$

In the case of distributed source coding, two sources are to be compressed. Therefore, the efficiency of the coding scheme is evaluated by two rates, or a rate region with two dimensions.

Definition 1.2.3. *A rate pair (R_X, R_Y) is said to be achievable for a distributed source if there exists a sequence of $((2^{nR_X}, 2^{nR_Y}), n)$ distributed source codes with probability of error $P_e^{(n)} \to 0$. The achievable rate region is the closure of the set of achievable rates.*

A naive coding scheme is to let each encoder compress each source alone without consideration of each other, as they are physically separated. Under such scheme, it is straightforward to see that $R_X \geq H(X)$ and $R_Y \geq H(Y)$. However, this is far from optimal.

Due to the work of Slepian, Wolf and Cover, the achievable rate region for stationary ergodic source has been determined as

$$\begin{aligned} R_X &\geq H(X|Y) && (1.18)\\ R_Y &\geq H(Y|X) && (1.19)\\ R_X + R_Y &\geq H(X,Y) && (1.20) \end{aligned}$$

which is quite surprising as this rate region is the same as that in the case two encoders are not separated. The achievable rate region by SW coding scheme is shown in 1.4.

This result suggests that even if the encoders are physically separated, and therefore do not know the sources of each other, the correlation between two sources can be explored at the decoder side. The idea of Slepian-Wolf coding scheme can be explained by binning argument. For one of the encoders, instead of telling the decoder what exactly the source is, it constructs bunch of bins blindly, i.e. without considering existence of the other source. Then it sends the index of bin in which the source lies. The other encoder does the same thing. For the decoder, as it knows the bin structures of both two encoders, it can construct joint bins with

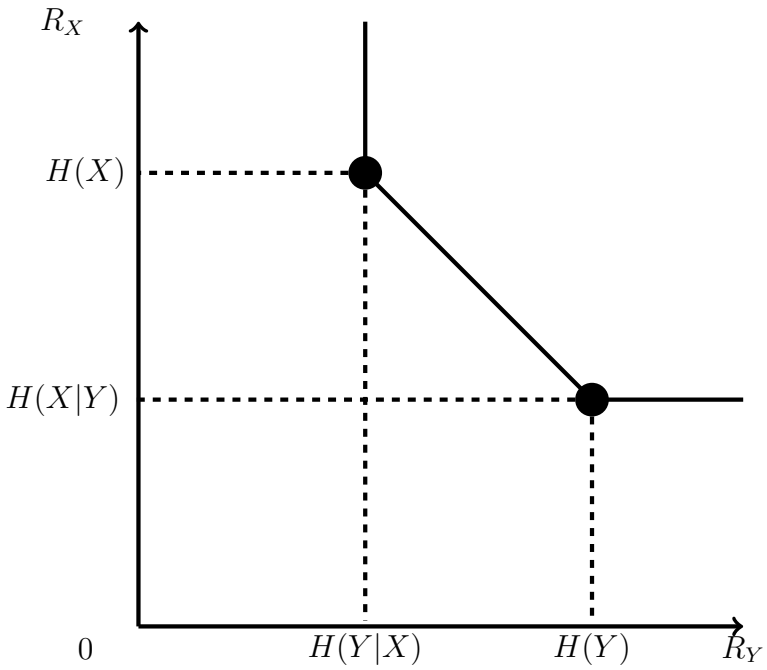

Figure 1.4: Slepian-Wolf Region

Figure 1.5: One Dimensional Bins for the Encoder

two dimensions. Then it consider the source pairs which fall in the joint bin with indices sent by those two encoders, and tries to find the pair which is most likely, considering the correlation between two sources.

One special case of SW coding is to consider the corner point of the achievable region, in which case one of the sources can be coded by classic source codes and reconstructed at the decoder losslessly, and then is treated as the side information of the other source. This approach is also called *Asymmetrical SW Coding*. Another perspective for this model is from source coding with side information at the encoder and the decoder considered in section 1.1.4. The modification here is that side information is no longer available to the encoder. As the rate R_X is the main concern, it is not difficult to derive from Slepian Wolf region that the achievable rate R_X must satisfy

$$R_X \geq H(X|Y) \tag{1.21}$$

Again, it is quite surprising to obverse that there is no rate-less asymptotically even if the encoder does not have the side information.

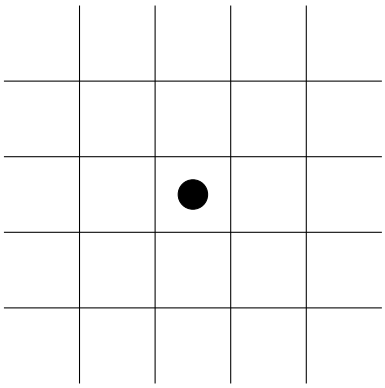

Figure 1.6: Two Dimensional Bins for the Decoder

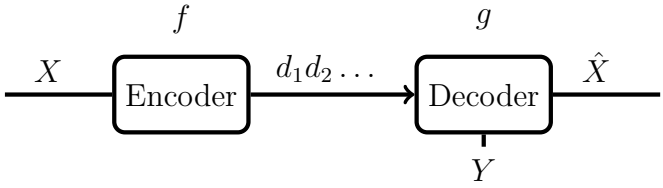

Figure 1.7: Source Code with Side Information at the Decoder Only

1.2.2 Channel Coding Approach

The method to prove the achievability of SW coding used in [40] [4] is random binning argument, illustrated above. Although this method is quite elegant for theoretical proof, it does not provide much intuition how to implement SW coding in practice. The first insight for practical SW coding is due to A.D. Wyner, who observed the relation between Asymmetrical SW coding and Channel Coding [46]. In the channel coding approach, the correlation between the source X and the side information Y is modelled by the virtual channel with its input Y and and output X. For the coding scheme, each bin is represented by a *coset* of channel code. Given

Figure 1.8: Virtual Channel from the side information Y to the source X

the source X, the encoder calculates the syndrome, or the index of coset in which X lies, and sends to the decoder. With the syndrome, or the coset X lies in and the side information Y, the decoder tries to recover X based on *Maximum Likelihood Method*. In this way, the design of a SW code is transformed to the design of a channel code, with each of its *coset code* powerful enough to correct the 'error' due to the noise of the virtual channel from Y to X. This idea was rediscovered by Pradhan and Ramchandran [33], and began to appear in practical code recently.

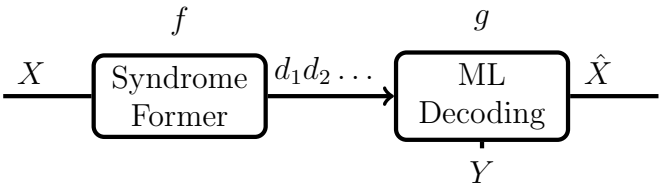

Figure 1.9: Channel Coding Approach for the Asymmetrical SW Coding

1.3 LDPC codes and Applications to SW Coding

LDPC codes and *Turbo codes* are two kinds of capacity-approach channel codes, which are also proved to achieve the SW limit. Moreover, the complexity of decoding is linear with block length if *belief propagation(BP) decoding*, or *sum product algorithm*(SPA) is used. For the practical SW coding design, LDPC codes are more popular than Turbo code. One of the reasons is that syndrome forming is more straightforward in LDPC codes than turbo codes. and another reason is that LDPC codes have better error floor performance, and the requirement of bit error is much higher in SW coding than channel coding.

1.3.1 Low-Density Parity-Check Code

Low-Density Parity-Check(LDPC) code was first invented by Gallager [13] back to 1960s, and then forgotten largely, as its complexity was considered impractical at that time. Tanner [41] proposed to represent codes as bipartite graphs and to view the iterative decoding as a message-passing algorithm on such a graph, where the bipartite graph is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V. In the framework of linear block code represented by parity check matrix, vertices of sets U and V represent the rows and columns of the parity check matrix, and vertex i in U is connected to j in V if and only if the element in i-th row and j-th column of the matrix is a non-zero entry. Generally, vertices in U are called check nodes, and V is the set of variable nodes.

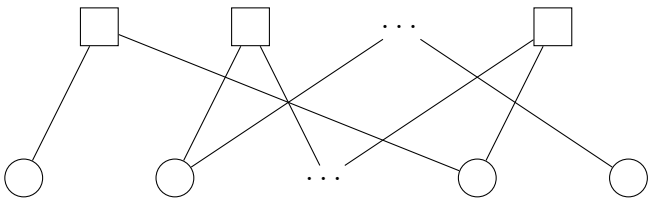

Figure 1.10: Tanner Graph

The *iterative decoding* or *message-passing algorithm* are the general terms referring to one kind of algorithms based on graphs. Actually many algorithms, such

as the *Viterbi algorithm* for convolutional codes and *BCJR algorithm*, can fit into this category, while *belief propagation* refers to the algorithm, which is also in this category, introduced by Kim and Pearl [23] to solve statistical inference problems in 1980s. Its successful implementation on turbo code ignited the related research in 1990s, and rediscovery of LDPC was credited to several research group, Mackay and Neal[27], Sipser and Spielman[39], and Wiberg[45]. *Density evolution* analysis was established by Richardson and Urbanke[34] , which founded the theoretical framework of asymptotical analysis of LDPC decoded with BP algorithm. Initially, research mainly focus on regular LDPC code, in which each row of parity check matrix has exactly d_c non-zero entries, and each column has exactly d_v non-zero entries, or in the language of graphs, each check node has d_c degrees, and each variable node has d_v degrees. Both empirical result and density evolution analysis shows that there is a small gap between achievable rate of regular LDPC code and Shannon limit. This gap can be further reduced by *irregular* LDPC code introduced by Luby et. al[26], where the degree of check(variable) node, no longer constant but varying from one to another, is represented by a distribution known as degree sequence.

1.3.2 Decoding over Finite State Channel

The concept of *finite state channel(FSC)* was established by Gallager [14], where he tried to provide a mathematical model for physical channel with memory. Essentially the history of channel inputs and outputs, as well as other parameters affecting the behaviour of channel, are described by the *state* of channel. Also there is a so called *non-anticipatory* assumption, which states for a given current input and previous state of channel, the output is statistically independent with the future inputs. Therefore, the statistical behaviour of the channel can be described by the joint distribution of the output and the state at given time conditioned on current input and previous states. If the state is represented by s_n, the input by y_n, and the output by x_n, then the channel is modelled by the distribution $p(x_n s_n | y_n s_{n-1})$.

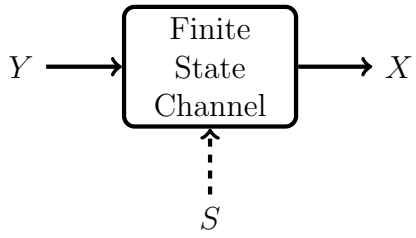

Figure 1.11: Finite State Channel

One of the special cases of finite state channel, which is also called hidden markov state model , is the case when $p(x_n s_n | y_n s_{n-1})$ can be factored into $p(x_n | y_n s_{n-1}) p(s_n | s_{n-1})$. In such scenario, the state of channel is a hidden markov process, independent with

the input of channel and un-revealed to decoder. One of the well known models in this category is the Gilbert-Elliott(GE) channel, a binary symmetric channel with crossover probability determined by the state of channel, which is either 0 or 1.

As finite state channel is of practical interest in some applications such as fading channel, many proposals of modified BP decoding for LDPC code used in FSC can be found in literature. For the best knowledge of the author, Garcia-Frias is the person who first targeted to this problem [15]. Similar algorithm targeting to more general markov noise case was proposed by Wadayama[43]. And density evolution analysis on those algorithm was given by Eckford, Kschischang, and Pasupathy[10] [9]. Fairly speaking, the model considered by those papers is restricted to hidden markov state channel.

1.3.3 Single Rate SW coding Based on LDPC Codes

The application of LDPC codes on SW coding can be found in literature [25, 36, 37, 35]. Their work targets to either asymmetrical SW coding or any achievable point of SW bound. BP decoding algorithm was modified to accommodate the situation here. However, as the original BP decoding used in these work targets to the memoryless channel in channel coding scenario, their coding schemes only work when the correlation between Y and X can be modelled by the memoryless channel.

However, the correlation between the source and the side information is much more complicated and variable than that between the channel input and output in channel coding. Therefore BP decoding algorithm working only for memoryless channel is far from enough for SW coding schemes. In [16], a BP decoding algorithm working under the assumption that there is hidden markov state channel between side information and source was proposed, which is based on the algorithm for channel coding mentioned in last section 1.3.2. But still, number of models are limited, for which there exists the algorithm that can be applied. This limitation is the central problem to push Slepian-Wolf coding into practical implementation.

1.4 Interactive Encoding and Decoding Schemes

1.4.1 Theoretical Approach – Benefit of Interaction

Interactive communication for lossless compression with the side information only at the encoder was first considered by Orlitsky[30, 31, 32]. In his setup, the decoder with the side information Y tries to learn X available at the encoder in two-way transmission, where X has to be reconstructed at the decoder with no probability of error. Note that the requirement of reconstruction is more strict than the Slepian Wolf case, where the probability of error goes to 0 asymptotically with block length. Therefore, the rate in this setup is higher than Slepian Wolf case. Meanwhile, the

idea of incremental encoding is introduced into asymmetrical SW coding by Feder and Shulman[11], where they consider the scenario that one common source is broadcast to several receivers with different side information. Coupling incremental encoding with universal fix-rate SW coding scheme proposed by Csiszar and Korner [6], Draper [7] built a universal SW coding scheme. However, as the universal coding scheme by Csiszar and Korner is only for the memoryless source pairs, so is Draper's scheme.

Recently, the concept of *interactive encoding and decoding(IED)* was formalized in [48], [47]. A special case of IED for (near) lossless one way learning (or in other words, lossless source coding) with decoder only side information is depicted in Figure 1.12 below, where X denotes a finite alphabet source to be learned at the decoder, Y denotes another finite alphabet source that is correlated with X and is only available to the decoder as the side information, and R denotes the average number bits per symbol exchanged between the encoder and the decoder measuring the performance of the IED scheme used. In view of Figure 1.12, we see that the main difference between IED and non-interactive Slepian-Wolf coding [40] lies in that IED allows the encoder and the decoder to interact until the learning (or source coding) task is accomplished.

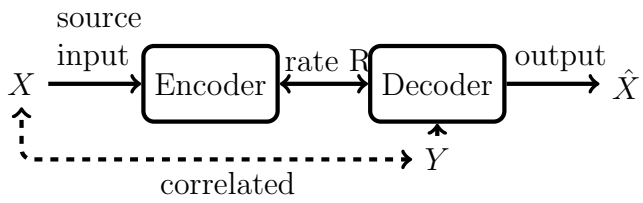

Figure 1.12: Interactive Encoding and Decoding for One Way Learning with the Side Information at the Decoder

Several important results concerning IED for (near) lossless source coding with decoder only side information were established in [48], [47]. Specifically, in comparison to non-interactive Slepian-Wolf coding [40], it was shown that IED not only delivers better first-order (asymptotic) performance for general stationary, non-ergodic source-side information pairs [48], but also achieves better second-order performance for memoryless pairs with known statistics [47]. Furthermore, in contrast to the well known fact that universal Slepian-Wolf coding does not exist, it was shown [47] that coupled with classical universal lossless codes [49], one can build IED schemes that are truly universal in the sense that they are asymptotically optimal with respect to the class of all stationary, ergodic sources-side information pairs. Essentially, the motivation to consider interactive schemes is that the feedback is embedded in many pratical communication protocols, like TCP-IP. And IED schemes provide a way to utilize this feature to further improve the efficiency and reduce complexity of the system.

1.4.2 Practical Approach – Rate-Less Slepian Wolf Coding

As the incremental encoding is embedded in interactive encoding and decoding scheme, a natural idea to implement IED is to build a *rate-less* SW coding, suitable for any compression rate of SW problem, and change the rate incrementally from low to high until the decoding is successful. Several such coding schemes are proposed for this kind of applications based on different codes and different techniques. Rate-less SW code can be built using *fountain codes* [38]. Considering the complexity of encoding and decoding practical implementation of this idea can be achieved by using *raptor codes* [12]. *Syndrome splitting* method is utilized both in *Product Accumulate codes*[2] and in LDPC codes[42] to get rate-less property, where parity check matrix is split in each time to generate a new syndrome. Also rate-less SW code can be developed based on rate-adaptive LDPC code[21], or *layered* LDPC code[8]. Note that in [8], the coding scheme is essentially *multi-rate*, instead of rate-less, in the sense it can only achieve rate of certain values. Actually generalization of this scheme to more than two rates becomes complicated and daunting.

1.5 Motivation

The results reviewed above imply that IED is much more appealing than Slepian-Wolf coding to applications where the one-way learning model depicted in Figure 1.12 fits. However, the IED schemes constructed in [48], [47] do not have an intrinsic structure that is amenable to design and implementation in practice.

1.5.1 Theoretical Side

In an attempt toward building practical IED schemes, in this thesis we consider linear IED schemes, i.e., IED schemes that use linear codes for encoding. It is shown that linear IED is indeed asymptotically optimal, or in other words, for any stationary, ergodic or not, source-side information pair (X, Y) in Figure 1.12, one can always find a sequence of linear IED schemes that is asymptotically optimal with respect (X, Y) in the sense of achieving the conditional entropy rate $H(X|Y)$ asymptotically while at the same time the error probability goes to zero asymptotically.

Another aspect of building practical IED schemes is to make the computational complexity incurred during the encoding and decoding process feasible. In practice, naturally one would like to have asymptotically optimal IED schemes that also have low encoding and decoding complexity. In the framework of linear IED, a linear IED scheme can be conveniently described by using parity-check matrices [3]. Examining the encoding and decoding process of a linear IED schemes, we observe that the density of these matrices (measured as the average number of non-zero entries) is

directly related to the scheme's encoding and decoding complexity: the complexity increases as the density increases [13]. For brevity, we shall refer to the density of the parity matrices used by a linear IED scheme simply as the density of the linear IED scheme. Subsequently, a *low-density linear IED scheme* simply refers to a linear IED scheme using parity-check matrices whose density is qualitatively low.

Our further efforts in this paper then reveal an interesting tradeoff between the density of the associated parity-check matrices and the resulting symbol error probability. Specifically, it is shown that as long as

$$\epsilon_n p_n^* = \Omega\left(\frac{\log n}{n}\right) \tag{1.22}$$

where ϵ_n and p_n^* are real numbers depending upon n, one can always find a sequence of linear IED schemes satisfying the following conditions simultaneously for any (X, Y):

1. the density of the associated parity-check matrices is concentrated around $(|\mathcal{X}| - 1)p_n^*$;
2. the rates of the IED schemes go to $H(X|Y)$ with probability one as $n \to \infty$; and
3. the resulting symbol error probability is upper bounded by $\epsilon_n + o(\epsilon_n)$.

Throughout the thesis, log denotes logarithm to 2, and ln denotes the natural logarithm. In view of the above result, we see that using the particular construction method considered in this paper, one has to balance ϵ_n and p_n^* under the constraint of (1.22). Our conjecture is that for (X, Y), (1.22) is also the necessary condition for the existence of IED schemes achieving $H(X|Y)$ at the speed of $O(1/\sqrt{n})$, which in turn reflects that the three quantities characterizing a linear IED scheme: error probability, rate, and density, are all connected. This conjecture, however, remains open at the moment of writing this thesis.

1.5.2 Practical Side

Although the complexity of encoding is manageable by considering low density linear IED schemes, complexity of decoding is impractical as it is still Maximum Likelihood Decoding(MLD). Therefore, to push the idea of IED towards a practical system, LDPC codes and BP decoding are utilized. For the encoding side, as feedback from decoder is embedded into the whole scheme, a rate-less or multi-rate SW coding scheme is expected here. In this thesis, we propose a rate-less SW coding based on LDPC codes and syndrome splitting. Different from [42], each step of splitting is optimized according to density evolution analysis in this scheme. We show that the best way to split the syndrome, or equivalently to split the row of parity check matrix, is to select the row with maximum degrees, and

split it evenly. This splitting method can be proved to be optimal in several LDPC decoding algorithms, for both symmetrical and asymmetrical channels.

For the decoder side, as the correlation between the source and the side information in SW coding is more variable than that between the channel input and output in channel coding, we propose a new BP decoding algorithm which targets to general finite-state channel models, where the state of channel is not necessarily statistically independent with the channel input. Further, we show that the algorithm for hidden markov state channel is a special case of this new algorithm.

1.6 Overview of the Thesis

The rest of the thesis is organized as follows. In chapter II, the overview of IED is presented. The formal definition and main results of IED will be introduced. In chapter III, the concept of general linear IED, and low density case will be defined, with theoretical analysis presented. In chapter IV, practical linear IED scheme based on LDPC codes and BP decoding is concerned. New rate-less SW coding scheme based on syndrome splitting and new BP decoding algorithm targeting to general finite-state channel are presented. Simulation result shows that there is indeed performance advantage of IED scheme over SW coding. Finally, the conclusion will be given in chapter V.

Chapter 2

Review of IED schemes

In this chapter, we briefly review the concept of IED, and some of the important results concerning IED for lossless source coding with decoder only side information (see Figure 1.12).

2.1 Definition

Throughout this paper, let $\mathcal{X}$ be a finite source alphabet with cardinality greater than or equal to 2. Let $\mathcal{X}^*$ be the set of all finite strings drawn from $\mathcal{X}$, including the empty string, and $\mathcal{X}^+$ be the set of all finite strings of positive length from $\mathcal{X}$. Let $\mathcal{X}^{++}$ be the set consisting of all finite dimension vectors over $\mathcal{X}^+$, i.e., $\mathcal{X}^{++} = \{(s_1, s_2, \cdots, s_k) : k \geq 1 \text{ and } s_i \in \mathcal{X}^+, i = 1, 2, \cdots, k\}$. The notation $|\mathcal{X}|$ stands for the cardinality of $\mathcal{X}$, and for any $x \in \mathcal{X}^*$, $|x|$ denotes the length of x. For any positive integer n, $\mathcal{X}^n$ denotes the set of all sequences of length n from $\mathcal{X}$. For convenience, we will sometimes write $x_m x_{m+1} \cdots x_n$ as x_m^n, where $m \leq n$ are two integers, and x_1^n simply as x^n. Similar notation applies to other countable sets and strings and random sequences drawn from them as well. In addition, random variables will be denoted by capital letters whereas their realizations will be denoted by the respective lower case letters.

Let $\mathcal{Y}$ denote a finite alphabet with cardinality greater than or equal to 2. Let $B = \{0, 1\}$. Suppose that the task is, with reference to Figure 1.12, to learn x^n drawn from $\mathcal{X}$ at the decoder with the helper y^n drawn from $\mathcal{Y}$. Then IED for this task can be formally defined as follows [48], [47]:

Definition 2.1.1. *An n-th order interactive encoding and decoding scheme is a triple $\mathcal{I}_n = (f_1, f_2, g)$, where*

- *$f_1 : \mathcal{X}^n \to B^+$ is a mapping from $\mathcal{X}^n$ to a binary prefix set, acting as the first round of encoding of x^n;*

- *f_2 is a mapping from $B^{++}\times\mathcal{X}^n$ to B^+ satisfying that for any $s=(s_1,s_2,\ldots,s_k)\in B^{++}$, the set $\{f_2(s,x^n)\}$ is a binary prefix set whenever $f_2(s,\cdot)$ is well defined as a function of $x^n\in\mathcal{X}^n$;*

- *together, (f_1,f_2) acts as an interactive encoder;*

- *$g=(g_1,g_2)$ is a mapping from $B^{++}\times\mathcal{Y}^n$ to $B^+\times(\mathcal{X}^n\cup\{\Lambda\})$ such that*

  1. *for any $s\in B^{++}$, the set $\{g_1(s,y^n):y^n\in\mathcal{Y}^n\}$ is a binary prefix set whenever $g_1(s,\cdot)$ as a function of $y^n\in\mathcal{Y}^n$ is well defined;*
  2. *for any $s\in B^{++}$ and $y^n\in\mathcal{Y}^n$, $g_2(s,y^n)$ is a member of $\mathcal{X}^n$ whenever the leading bit of $g_1(s,y^n)$ is 1;*

- *the mapping g acts as an interactive decoder with g_1 providing a feedback $g_1(s,y^n)$ to the interactive encoder upon receiving s and g_2 trying to decode out an estimate for x^n.*

The IED scheme $\mathcal{I}_n$ defined above works as follows. To encode x^n with y^n available only to the decoder, the encoder first encodes x^n into $s_1=f_1(x^n)$ and then sends s_1 to the decoder. Upon receiving s_1, the decoder outputs $g_1(s_1,y^n)$ (denoted as t_1) and $g_2(s_1,y^n)$, and then sends t_1 back to the encoder. If the leading bit of t_1 is 1, then the decoder declares $g_2(s_1,y^n)$ as an estimate of x^n; in this case, the IED process is deemed to terminate after one round of interaction. Otherwise, upon receiving t_1, the encoder encodes x^n again into $s_2=f_2(t_1,x^n)$ and sends s_2 to the decoder. Upon receiving s_2, the decoder outputs $g_1((s_1,s_2),y^n)$ (denoted as t_2) and $g_2((s_1,s_2),y^n)$, and then sends t_2 back to the encoder. If the leading bit of t_2 is 1, then the decoder declares $g_2((s_1,s_2),y^n)$ as an estimate of x^n, and the IED process is deemed to terminate after two rounds of interaction. Otherwise, the whole process repeats itself again and again until at some point, after receiving bitstreams $s_1,s_2,\cdots,s_j$ from the encoder, the decoder outputs and sends back $t_j=g_1((s_1,s_2,\cdots,s_j),y^n)$ the leading bit of which is 1, at which point the IED process is deemed to terminate, and $g_2((s_1,s_2,\cdots,s_j),y^n)$ is regarded as an estimate of x^n.

For any $x^n\in\mathcal{X}^n$ and $y^n\in\mathcal{Y}^n$, let $j(x^n,y^n)$ denote the number of interactions at the time when the IED process of $\mathcal{I}_n$ terminates. In general, $j(x^n,y^n)$ depends on both x^n and y^n and varies. Nonetheless, for convenience, we shall write hereafter $j(x^n,y^n)$ simply as j whenever there is no ambiguity, i.e., whenever the pair of involved sequences x^n and y^n is clear from the context. During the j rounds of interaction, the encoder sends j bitstreams $s_1,s_2,\cdots,s_j$ to the decoder; likewise, the decoder sends back j bitstreams $t_1,t_2,\cdots,t_j$ to the encoder. These two groups of bitstreams interleave with each other:

$$\begin{aligned} s_1 &= f_1(x^n) \\ t_i &= g_1((s_1,s_2,\cdots,s_i),y^n),\ \ i=1,2,\cdots,j \\ s_i &= f_2((t_1,t_2,\cdots,t_{i-1}),x^n),\ \ i=2,\cdots,j \end{aligned}$$

Therefore, the forward bit rate in bits per symbol from the encoder to the decoder is

$$r_f(x^n, y^n|\mathcal{I}_n) \quad \triangleq \quad \frac{|s_1| + |s_2| + \cdots + |s_j|}{n}$$

and the backward bit rate in bits per symbol from the decoder to the encoder is

$$r_b(x^n, y^n|\mathcal{I}_n) \quad \triangleq \quad \frac{|t_1| + |t_2| + \cdots + |t_j|}{n} \ .$$

The total number of bits per symbol exchanged between the encoder and decoder during the whole IED process of $\mathcal{I}_n$ (hereafter referred to as the total rate in bit per symbol and denoted as $r(x^n, y^n|\mathcal{I}_n)$) is then equal to

$$r(x^n, y^n|\mathcal{I}_n) \quad \triangleq \quad r_f(x^n, y^n|\mathcal{I}_n) + r_b(x^n, y^n|\mathcal{I}_n) \ .$$

A decoding error occurs if the estimate $g_2((s_1, s_2, \cdots, s_j), y^n)$ is not equal to x^n.

Let $(X, Y) = \{(X_i, Y_i)\}_{i=1}^{\infty}$ be a stationary source pair with alphabet $\mathcal{X} \times \mathcal{Y}$. When $\mathcal{I}_n$ is applied to interactively encode X^n with Y^n available only to the decoder, the resulting average forward bit rate, backward bit rate, and total rate in bits per symbol are, respectively,

$$\begin{aligned} R_f(\mathcal{I}_n) \quad &\triangleq \quad \mathbf{E}[r_f(X^n, Y^n|\mathcal{I}_n)], \\ R_b(\mathcal{I}_n) \quad &\triangleq \quad \mathbf{E}[r_b(X^n, Y^n|\mathcal{I}_n)], \end{aligned}$$

and

$$R(\mathcal{I}_n) \quad \triangleq \quad \mathbf{E}[r(X^n, Y^n|\mathcal{I}_n)].$$

Let $J(\mathcal{I}_n)$ and $P(\mathcal{I}_n)$ denote respectively the resulting average number of interactions during the whole encoding and decoding process of $\mathcal{I}_n$ and the resulting (block) error probability, i.e.,

$$\begin{aligned} J(\mathcal{I}_n) \quad &\triangleq \quad \mathbf{E}[j(X^n, Y^n)] \text{ and} \\ P(\mathcal{I}_n) \quad &\triangleq \quad \Pr\{g_2((s_1, s_2, \cdots, s_{j(X^n, Y^n)}), Y^n) \neq X^n\} \ . \end{aligned}$$

The above five quantities are interesting parameters related to the general performance of $\mathcal{I}_n$ for (X, Y).

Note that the definition of interactive encoding and decoding can be extended to more general source networks, for instance, the model shown in figure 2.1.

With respect to figure 2.1, IED is a $\mathcal{I}_n = (f_1, f_2, f_3, f_4, g)$, consisting of five parts. f_1, f_2, which specifies the encoder on X, follow the same definition above, and f_3, f_4 for the encoder on Y are defined similarly, the only difference is to change the domain of $\mathcal{X}$ in the definition of f_1, f_2 to $\mathcal{Y}$. For decoder, $g = (g_1, g_2, g_3, g_4)$ is a mapping from $B^{++} \times B^{++}$ to $B^+ \times B^+ \times (\mathcal{X}^n \cup \{\Lambda\}) \times (\mathcal{Y} \cup \{\Lambda\})$, where for any

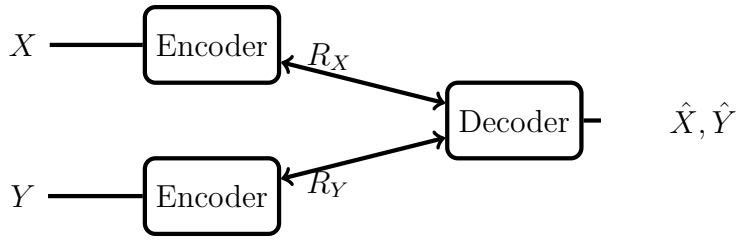

Figure 2.1: IED scheme for the separated encoders and the joint decoder

$s_0 \in B^{++}$, the set $\{g_1(s_0, s) : s \in B^{++}\}$ and $\{g_2(s, s_0) : s \in B^{++}\}$ are binary sets whenever $g_1(s_0, \cdot)$ and $g_2(\cdot, s_0)$ as a function of $s \in B^{++}$ is well defined. And for any $s_1, s_2 \in B^{++}$, $g_3(s_1, s_2)$ $(g_4(s_1, s_2))$ is a member of $\mathcal{X}^n$ $(\mathcal{Y}^n)$ whenever the leading bit of $g_1(s_1, s_2)$ $(g_2(s_1, s_2))$ is 1. Here g_1 and g_2 provide feedback for encoder on X and Y respectively, and g_3 and g_4 try to estimate X and Y. Similarly interactive encoding and decoding scheme can be extended to many other source networks.

2.2 Results

The problem of determining how small $R(\mathcal{I}_n)$ can be while keeping $P(\mathcal{I}_n)$ vanishing as $n \to \infty$ without imposing any constraint on $J(\mathcal{I}_n)$ was studied in detail in [48], [47]. A series of coding theorem was then established.

2.2.1 IED Scheme for Stationary Ergodic Source Pairs - Universality Gain

It is well known that there does not exist universal SW coding. Particularly, for asymptotical SW coding case, as the side information is not available for encoder, there is no way for the encoder to learn the conditional statistics of source given the side information, even if two-round coding is allowed. However, for memoryless source pair, if $H(X|Y)$, the theoretical limit, is known to encoder and decoder, there exists so-called universal SW coding in the sense that encoder and decoder do not have to know the statistics of source and side information except $H(X|Y)$ [5, 29]. The reason that $H(X|Y)$ is needed in those schemes is that they are fix-rate coding scheme. One way to build a truly universal scheme is to build a variable rate scheme base on those schemes, by allowing the decoder to send feedback to the encoder. However, directly basing on those scheme will end up a scheme which works only for memoryless source pairs.

Universal IED schemes for any stationary ergodic source pair proposed in [48, 47] are to couple incremental encoding and classical universal prefix source code with side information at both encoder and decoder. The basic idea is that by incremental encoding, both the encoder and the decoder probe the rate required to send source losslessly to decoder with side information, as $H(X|Y)$ is unknown,

while by classical universal prefix source code, decoder can build up a rule to decide when the information sent by encoder is enough for it to reconstruct source losslessly.

In particular, the following result was proved by using a random binning argument.

Result 1. *There exists a sequence of IED schemes*[1] $\{\mathcal{I}_n\}$ *such that for any stationary, ergodic source-side information pair* (X, Y)*,*

$$\lim_{n\to\infty} r_f(X^n, Y^n|\mathcal{I}_n) = H(X|Y) \text{ with probability one,} \tag{2.1}$$

$$r_b(X^n, Y^n|\mathcal{I}_n) = O\left(\frac{1}{\sqrt{n}}\right), \text{ and} \tag{2.2}$$

$$P(\mathcal{I}_n) = O(2^{-\sqrt{n}}). \tag{2.3}$$

Result 1 essentially states that universal IED schemes exist for the class of all stationary, ergodic source pairs. Thus by allowing interactions between the encoder and decoder, IED achieves universality gain over non-interactive Slepian-Wolf coding.

2.2.2 IED Scheme for Stationary Non-Ergodic Source Pairs - Rate Gain

The way to study non-ergodic source pairs is to use ergodic decomposition [17]. If (X, Y) is not ergodic, one can equivalently regard (X, Y) as a class of stationary ergodic source pairs indexed by a random variable Θ, which takes values in a measurable space $(\Lambda, \digamma)$ with probability distribution $\mu(\cdot)$. In other word, given value of Θ, (X, Y) is conditionally stationary ergodic. As estimating this Θ may not be possible under some circumstance, or even it is possible, the process will be extremely complicated, increasing the complexity of coding scheme dramatically, we assume that Θ is unknown to either encoder or decoder.

For any $\theta \in \Lambda$, define

$$H(\theta) \triangleq \lim_{n\to\infty} \frac{1}{n} H(X^n|Y^n, \theta) \tag{2.4}$$

and

$$H(X|Y) = \lim_{n\to\infty} \frac{1}{n} H(X^n|Y^n) \tag{2.5}$$

It then follows that

$$H(X|Y) = \mathbf{E}[H(\Theta)] = \int_\Lambda H(\theta) d\mu(\theta) \tag{2.6}$$

[1]Note that the schemes constructed in [47] are random in that they utilize a random database known to both the encoder and the decoder.

Now let us consider the distribution of source X given value of $\Theta = \theta$. Define

$$P_{X|\theta} = \{p(x^n|\theta) : x^n \in \mathcal{X}^n, n \geq 1\} \tag{2.7}$$

which represents the ergodic component of X given θ. Also define the set $[\theta]$ in $\digamma$ as

$$[\theta] \triangleq \{\eta \in \Lambda : P_{X|\eta} = P_{X|\theta}\} \tag{2.8}$$

Also define

$$[\Lambda] \triangleq \{[\theta] : \theta \in \Lambda\} \tag{2.9}$$

It is easy to see that $[\theta_1] = [\theta_2]$ if and only if θ_1 and θ_2 correspond to the same ergodic component of X. Note that $[\theta_1] = [\theta_2]$ does not imply they correspond to the same ergodic component of (X, Y). Therefore $H(\theta_1)$ may or may not equal to $H(\theta_2)$. Now we are interesting in finding the largest $H(\theta)$ given the same ergodic component of X. For each $[\theta] \in [\Lambda]$, define $H_S([\theta])$ as the conditional essential supremum of $H(\Theta)$ given $\Theta \in [\theta]$,

$$H_S([\theta]) \triangleq \inf\{\alpha : Pr\{H(\Theta) > \alpha | \Theta \in [\theta]\} = 0\} \tag{2.10}$$

As $H_S([\theta])$ is a function of θ on $(\Lambda, \digamma)$, define

$$H_S(X|Y) \triangleq \mathbf{E}[H_S([\Theta])] = \int_\Lambda H_S([\theta]) d\mu(\theta) \tag{2.11}$$

Then for asymmetrical SW and IED setup, the following results are given in [48, 47],

Result 2. *For any stationary source pair (X, Y), the minimum rate achievable asymptotically in asymmetrical SW setup $R_S(X, Y) = H_S(X|Y)$.*

Result 3. *For any stationary source pair (X, Y), the minimum rate achievable asymptotically in IED setup $R(X, Y) = H(X|Y)$.*

As for most of non-ergodic source,

$$H(X|Y) < H_S(X|Y) \tag{2.12}$$

by using Result 2 and 3, it was shown [48], [47] that for stationary, non-ergodic pair (X, Y), the best achievable rate $R(\mathcal{I}_n)$ under the condition that $P(\mathcal{I}_n)$ vanishes as $n \to \infty$ is in most cases strictly less than that afforded by Slepian-Wolf coding. In this sense, IED achieves first order performance gain over Slepian-Wolf coding.

2.2.3 IED scheme for Memoryless Source Pairs - Redundancy Gain

Now let us consider the case that the source and side information are i.i.d pairs, whose statistics are known by both the decoder and the encoder. In this case, from Slepian-Wolf coding theorem, we know that the minimum achievable rate of SW

coding is $H(X|Y)$. This limit also applies for IED, as even the encoder is informed of side information, $H(X|Y)$ is the minimum rate that can be achieved. However, if the speed that rate of coding scheme approaches this limit with respect to block length, as well as the speed that error probability tends to zero, is concerned, there are still advantages of IED over asymmetrical SW coding.

The redundancy result of asymmetrical SW coding is obtained in [18, 19].

Result 4. *Let (X, Y) be an i.i.d. source-side information pair with finite alphabet $\mathcal{X} \times \mathcal{Y}$ such that*

$$I(X;Y) > 0$$

and $P_{XY}(x, y) > 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Let $\{\epsilon_n\}$ be a sequence of positive real numbers satisfying $\epsilon_n = o\left(\sqrt{\frac{\log n}{n}}\right)$ and $-\log \epsilon_n = o(n)$. Then for sufficiently large n and any order-n SW code $\mathbb{S}_n = (f_n, g_n)$ satisfying

$$P(\mathcal{S}_n) \leq \epsilon_n$$

one has

$$R(\mathcal{S}_n) \geq H(X|Y) + \Theta\left(\sqrt{\frac{-\log \epsilon_n}{n}}\right)$$

Here $\Theta(\cdot)$ is defined as for any two functions $f(n)$ and $g(n)$ of n, $f(n) = \Theta(g(n))$ if and only if $\exists c_1, c_2$, which are constants, such that

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

At the same time, redundancy result of IED is given in [47].

Result 5. *Let (X, Y) be an i.i.d. source-side information pair with finite alphabet $\mathcal{X} \times \mathcal{Y}$ such that*

$$I(X;Y) > 0$$

and $P_{XY}(x, y) > 0$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Then there exists a sequence of IED schemes $\{\mathcal{I}_n\}$ such that for sufficiently large n,

$$\begin{aligned} R_f(\mathcal{I}_n) &\leq H(X|Y) + c\frac{(\log n)^{1/4}}{n^{3/4}} + o\left(\frac{(\log n)^{1/4}}{n^{3/4}}\right) \\ R_b(\mathcal{I}_n) &\leq c\frac{(\log n)^{1/4}}{n^{3/4}} \end{aligned}$$

and

$$P(\mathcal{I}_n) \leq 2^{-c(n \log n)^{1/4}}$$

By result 4 and 5, both the speed that rate of coding scheme approaches to $H(X|Y)$, and the speed that the error probability approaches to zero are much faster in IED than in asymmetrical SW coding. This shows that IED achieve better second order performance over SW coding, which suggests that to achieve the same rate and error probability, the required block length is less in IED scheme than in SW coding scheme. As we know that complexity of coding scheme is related to the block length, this result shows that compared to SW coding, to achieve the same rate and error probability, the IED scheme should have less complexity.

2.2.4 Converse Coding Theorem

To gain more insight of IED, a converse coding theorem is provided in [47], which states that

Result 6. *Let (X, Y) be stationary. For any sequence $\{\mathcal{I}_n\}$ of IED schemes, where $\mathcal{I}_n$ is an n-th order IED scheme,*

$$\liminf_{n\to\infty} R_f(\mathcal{I}_n) \geq H(X|Y)$$

whenever $P(\mathcal{I}_n)$ goes to 0 as $n \to \infty$.

This result suggests that the forward rate R_f in IED must be greater than $H(X|Y)$ if the error probability goes to zero. It also suggests that if total rate R of IED approaches to $H(X|Y)$, the backward rate R_b of IED must goes to zero asymptotically.

2.3 Summary

With definition formalised in [47], several results mentioned above have been obtained. These results clearly suggest that for applications where the model depicted in Figure 1.12 fits, IED is a preferred choice compared to Slepian-Wolf coding in the sense that IED has universality gain, rate gain and redundancy gain over SW coding for different scenarios. However, the random IED schemes constructed in [48], [47] do not have an intrinsic structure that is amenable to design and implementation in practice. In our attempt toward building practical IED schemes, we begin with investigating linear IED in the next chapter.

Chapter 3

Linear IED schemes with Low Density Parity Check Matrix

3.1 Linear IED

In this section, we investigate linear IED where (f_1, f_2) is a linear mapping, and g implements *maximum a posteriori*(MAP) decoding. Throughout this section, $\mathcal{X}$ is assumed to be a finite field denoted by $\mathbf{GF}(q^m)$, where q is a prime number, and m is a positive integer, and (f_1, f_2) in Definition 2.1.1 is conveniently described by using a sequence of parity-check matrices $\{\mathbf{H}^{(i)}; i = 0, 1, 2, \cdots\}$ with n columns whose elements are from $\mathbf{GF}(q^m)$:

$$\begin{aligned} s_1 &= f_1(x^n) = \mathbf{H}^{(0)} x^n \\ s_i &= f_2((t_1, t_2, \cdots, t_{i-1}), x^n) = \mathbf{H}^{(i-1)} x^n, \quad i = 2, \cdots, j, \end{aligned}$$

where we abuse the notation by regarding x^n as a column vector.

Our purpose in this section is to show that for the lossless source coding problem depicted in Figure 1.12, there is no (asymptotic) performance loss by restricting to linear IED. To this end, we construct a sequence of linear IED schemes $\{\mathcal{I}_n\}$. Before describing how $\mathcal{I}_n$ works, we need a few more definitions. Let $(X, Y) = \{(X_i, Y_i)\}_{i=0}^{\infty}$ be a stationary, ergodic source-side information pair with alphabet $\mathcal{X} \times \mathcal{Y}$. Let $\{C_n\}_{n=1}^{\infty}$ denote a sequence of (classical) prefix codes, where C_n is a mapping from $\mathcal{X}^n \times \mathcal{Y}^n$ to $\{0, 1\}^*$. For each code C_n in this sequence, we define a normalized length function $h_n(\cdot|\cdot)$ such that $nh_n(x^n|y^n)$ denotes the number of bits when C_n applies to encode x^n from $\mathcal{X}$ given side information sequence y^n from $\mathcal{Y}$. As in [47], we assume that

$$\lim_{n\to\infty} h_n(X^n|Y^n) = H(X|Y) \text{ with probability one} \tag{3.1}$$

for any stationary, ergodic pair (X, Y). In other words, $\{C_n\}$ is assumed to be universal with respect to the class of all stationary, ergodic source pairs.

We are now ready to describe our linear IED schemes $\{\mathcal{I}_n\}$. Let Δ be a positive integer to be specified later. To encode a source sequence X^n with decoder only side information sequence Y^n, $\mathcal{I}_n$ works as follows.

1: $l \leftarrow 0$.
2: **while** Encoder does not receive 1 **do**
3: Encoder selects a $\Delta \times n$ matrix $\mathbf{H}^{(l)}$ whose elements are randomly generated from $\mathcal{X}$ by using a uniform distribution (the sequence $\mathbf{H}^{(0)}, \mathbf{H}^{(1)}, ...$ is known to both Encoder and Decoder before the encoding process begins), and sends the syndrome $S_l = \mathbf{H}^{(l)} X^n$ by using $\Delta \log |\mathcal{X}| = m\Delta \log q$ bits.
4: Upon receiving the syndrome S_l, Decoder tries to find the sequence $\hat{X}^n$ as follows.

$$\hat{X}^n = \underset{v^n \in \mathcal{X}^n:\, \mathbf{H}^{(i)} v^n = S_i,\, i=0,\cdots,l}{\arg\min} h_n(v^n|Y^n). \tag{3.2}$$

5: **if** $h_n(\hat{X}^n|Y^n) \leq \frac{l\Delta \log |\mathcal{X}|}{n}$ **then**
6: Decoder reconstructs X^n as $\hat{X}^n$, and sends bit 1 to Encoder.
7: **else**
8: Decoder sends bit 0 to Encoder.
9: **end if**
10: $l \leftarrow l + 1$.
11: **end while**

Analyzing the performance of $\mathcal{I}_n$, we need the following lemma [47].

Lemma 1. *For any $y^n \in \mathcal{Y}^n$ and any $0 \leq \alpha \leq \log |\mathcal{X}|$, define $A^\alpha(y^n) \stackrel{\Delta}{=} \{x^n \in \mathcal{X}^n : h_n(x^n|y^n) \leq \alpha\}$. Then*

$$|A^\alpha(y^n)| \leq 2^{n\alpha}.$$

Proof. Since the code $\mathcal{C}_n$ is prefix code, it satisfies Kraft inequality. Therefore, $\forall y^n$,

$$\begin{aligned} 1 &\geq \sum_{x^n \in \mathcal{X}^n} 2^{-nh_n(x^n|y^n)} \\ &\geq \sum_{x^n \in A^\alpha(y^n)} 2^{-nh_n(x^n|y^n)} \\ &\geq \sum_{x^n \in A^\alpha(y^n)} 2^{-n\alpha} \\ &\geq |A^\alpha(y^n)| 2^{-n\alpha} \end{aligned}$$

which finishes the proof. □

Then we arrive at the following theorem.

Theorem 1. *For any stationary, ergodic source-side information pair (X, Y),*

$$\lim_{n\to\infty} r_f(\mathcal{I}_n) = H(X|Y) \text{ with probability one,} \tag{3.3}$$

$$r_b(\mathcal{I}_n) = O(\frac{1}{\sqrt{n}}), \ \ and \tag{3.4}$$

$$P(\mathcal{I}_n) \leq O(2^{-\sqrt{n}}). \tag{3.5}$$

Proof. We begin the proof with selecting $\Delta \sim \sqrt{n}$. For notational convenience, we also assume that Δ divides n. In view of **Step 3** of the IED scheme $\mathcal{I}_n$ constructed above, the maximum number of feedback bits is $\frac{n}{\Delta} \sim \sqrt{n}$. Thus, we have already established (3.33).

To prove (3.32), we consider two cases: $\hat{X}^n = X^n$ and $\hat{X}^n \neq X^n$. In the case where $\hat{X}^n = X^n$, we see that

$$r_f(X^n, Y^n|\mathcal{I}_n) \leq h_n(X^n|Y^n) + \frac{\Delta}{n}. \tag{3.6}$$

In the case where $\hat{X}^n \neq X^n$, we have

$$\begin{aligned} r_f(X^n, Y^n|\mathcal{I}_n) &\leq h_n(\hat{X}^n|Y^n) + \frac{\Delta}{n} \\ &\leq h_n(X^n|Y^n) + \frac{\Delta}{n}, \end{aligned} \tag{3.7}$$

where the last inequality follows from (3.29). Putting (3.1), (3.36), and (3.37) together, and recalling our selection of $\Delta \sim \sqrt{n}$, we have proved (3.32).

It remains to upper bound $P(\mathcal{I}_n)$. To this end, let us define a sequence of indicator random variables $\{E_l\}_{l=0}^{\frac{n}{\Delta}}$ as follows. For $0 \leq l \leq n/\Delta$,

$$E_l \stackrel{\Delta}{=} \begin{cases} 1 & \text{if there exists a sequence } v^n \in \mathcal{X}^n \text{ such that } v^n \neq x^n, \\ & \mathbf{H}^{(i)}(x^n - v^n) = 0 \text{ for } 0 \leq i \leq l, \text{ and } h(v^n|Y^n) \leq \frac{l\Delta \log|\mathcal{X}|}{n} \\ 0 & \text{otherwise} \end{cases}.$$

In view of **Step 3** of $\mathcal{I}_n$, we see that for any non-zero vector $\tilde{x}^n$,

$$Pr\{\mathbf{H}^{(i)}\tilde{x}^n = 0\} = \frac{1}{|\mathcal{X}|^{\Delta}}. \tag{3.8}$$

Further, since the error event $\hat{X}^n \neq X^n$ implies $E_l = 1$ at some l, we have by union bound that

$$P(\mathcal{I}_n) \leq \sum_{l=0}^{\frac{n}{\Delta}} Pr\{E_l = 1\}. \tag{3.9}$$

It follows from Lemma 1 and (3.39) that

$$\begin{aligned} Pr\{E_l = 1|Y^n = y^n\} &\leq |A^{\frac{l\Delta \log|\mathcal{X}|}{n}}(y^n)| \prod_{i=0}^{l} \frac{1}{|\mathcal{X}|^{\Delta}} \\ &\leq 2^{-\Delta \log|\mathcal{X}|}, \end{aligned}$$

which, together with (3.40), implies

$$P(I_n) \quad \leq \quad 2^{-\Delta \log |\mathcal{X}| + \log(\frac{n}{\Delta}+1)} = O(2^{-\sqrt{n}}), \tag{3.10}$$

where the last equality is due to our selection of $\Delta \sim \sqrt{n}$. This completes the proof of Theorem 1. □

Remark 1. *In view of Theorem 1, Result 1, and the converse coding theorems in [47], we see that linear IED is indeed asymptotically optimal for any stationary, ergodic source pairs.*

This asymptotic optimality can be further extended to non-ergodic pairs by using an argument similar to that used in [48].

Theorem 2. *For any stationary source-side information pair (X, Y), there exists a sequence of linear IED scheme $\{\mathcal{I}_n\}$, such that $\forall \epsilon > 0$, $\exists N$ satisfying $n > N$,*

$$\begin{aligned} R(\mathcal{I}_n) &\leq H(X|Y) + \epsilon \\ P(\mathcal{I}_n) &\leq \epsilon \end{aligned}$$

Remark 2. *This theorem is the direct result of theorem 1 and ergodic decomposition. Essentially by theorem 1, linear IED scheme can achieve $H(\theta)$ for ergodic component of (X, Y) corresponding to θ. Details omitted here can be found in the full paper [28].*

3.2 Low-Density Linear IED

In view of the encoding and decoding process of the IED scheme $\mathcal{I}_n$ constructed in Section 3.1 above, we see that the encoding complexity and the decoding complexity of $\mathcal{I}_n$ are directly related to the density of the parity-check matrices $\mathbf{H}^{(0)}, \mathbf{H}^{(1)}, \cdots$: the lower the density, the lower the complexity. Clearly there is a lower bound on the density of linear IED schemes that needs to be maintained without negatively affecting the asymptotic optimality. The purpose of this section is then to investigate the performance of low-density linear IED schemes. Specifically, we modify the linear IED schemes in Section 3.1 and show that under the constraint that the total rate in bits per symbol goes to $H(X|Y)$ asymptotically for any stationary, ergodic pair (X, Y), there exists an interesting tradeoff between symbol error probability and density.

Our modification of the linear IED scheme $\mathcal{I}_n$ in Section 3.1 is as follows. Instead of using a uniform distribution over $\mathcal{X}$ to draw elements in $\mathbf{H}^{(l)}$ independently, $l \geq 0$, we use the following distribution $\mathbf{p}_{n,l}$. Let x denote a symbol in $\mathcal{X}$.

$$\mathbf{p}_{n,l}(x) = \begin{cases} p_{n,l}^0 & \text{if } x = 0 \\ p_{n,l}^* & \text{otherwise} \end{cases}, \tag{3.1}$$

where $p^*_{n,l}$ is a real number in $[0,1]$ and $p^0_{n,l} = 1-(|\mathcal{X}|-1)p^*_{n,l}$. Thus if $p^*_{n,l}$ is small, $\mathbf{p}_{n,l}$ is biased toward 0, and as a consequence the density of $\mathbf{H}^{(l)}$ is low. Throughout this paper, we assume that $p^*_{n,l} \leq \frac{1}{|\mathcal{X}|}$.

Note that each element in $\mathbf{H}^{(i)}$, $i \geq 0$, is randomly drawn from $\mathcal{X}$ by using the probability distribution $\mathbf{p}_{n,i}$. Using this fact, we prove the following lemma.

Lemma 2. *For any sequence $x^n \in \mathcal{X}^n$,*

$$Pr\{\mathbf{H}^{(i)}x^n = 0\} = \left[\frac{1+(|\mathcal{X}|-1)(1-|\mathcal{X}|p^*_{n,i})^{w(x^n)}}{|\mathcal{X}|}\right]^{\Delta},$$

where $w(x^n)$ denotes the weight of x^n, i.e., $w(x^n) \stackrel{\Delta}{=} |\{i : x_i \neq 0\}|$.

Proof. Fix $x^n \in \mathcal{X}^n$. We write the weight $w(x^n)$ of x^n simply as w for brevity. Recall that each element in $\mathbf{H}^{(i)}$ is drawn independently from $\mathcal{X}$ by using the probability distribution $\mathbf{p}_{n,i}$. Thus in order to calculate the probability of the event $\mathbf{H}^{(i)}x^n = 0$, it suffices to calculate the probability of the following event

$$U_1 + U_2 + \cdots + U_w = 0, \tag{3.2}$$

where $\{U_k\}_{k=1}^w$ is a sequence of independent copies of a random variable U whose distribution P_U over $\mathcal{X}$ is equal to $\mathbf{p}_{n,i}$. Note that in (3.2), the addition is defined over the finite field $\mathcal{X} = GF(q^m)$.

Let $S = \sum_{k=1}^w U_k$. Then the distribution P_S of the random variable S is equal to the w-th order convolution of P_U. In the following, we shall use the standard discrete fourier transform (DFT) method to facilitate the calculation of P_S from P_U.

Let $\mathcal{F}_U$ denote the DFT of P_U defined as follows. For any $a \in \mathcal{X}$,

$$\mathcal{F}_U(a) = \sum_{b\in\mathcal{X}} P_U(b) e^{-\frac{2\pi j}{q} a\cdot b},$$

where $j = \sqrt{-1}$, and $a \cdot b$ denotes the dot product of a and b regarded as two length-m vectors over $GF(q)$, that is, $a = (a_1, a_2, \cdots, a_m)$, $b = (b_1, b_2, \cdots, b_m)$, and $a \cdot b = \sum_{i=1}^m a_i b_i$, where a_i and b_i, $1 \leq i \leq m$, are all from $GF(q)$. Similarly, let $\mathcal{F}_S$ denote the DFT of P_S. It follows immediately from the convolution theorem that for any $a \in \mathcal{X}$,

$$\mathcal{F}_S(a) = \left(\mathcal{F}_U(a)\right)^w, \text{ and} \tag{3.3}$$

$$P_S(0) = \frac{1}{|\mathcal{X}|}\sum_{a\in\mathcal{X}} \mathcal{F}_S(a). \tag{3.4}$$

Let us now calculate $\mathcal{F}_U$. Observe that P_U can be written as the linear sum of a uniform distribution P_0 and a unitary distribution P_1, both over $\mathcal{X}$, i.e.,

$$P_U = p^*_{n,i}P_0 + (p^0_{n,i} - p^*_{n,i})P_1, \tag{3.5}$$

where for any $a \in \mathcal{X}$, $P_0(a) = \frac{1}{|\mathcal{X}|}$, and

$$P_1(a) = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}.$$

The DFTs of P_0 and P_1, denoted by $\mathcal{F}_0$ and $\mathcal{F}_1$, respectively, can be easily calculated by definition: for any $a \in \mathcal{X}$,

$$\mathcal{F}_0(a) = \begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases},$$

and $\mathcal{F}_1(a) = 1$. Thus, it follows from the linearity of DFT and (3.5) that

$$\begin{aligned} \mathcal{F}_U &= p^*_{n,i}\mathcal{F}_0 + (p^0_{n,i} - p^*_{n,i})\mathcal{F}_1 \\ &= \begin{cases} 1 & \text{if } a = 0 \\ p^0_{n,i} - p^*_{n,i} = 1 - |\mathcal{X}|p^*_{n,i} & \text{otherwise} \end{cases}. \end{aligned} \tag{3.6}$$

Combining (3.3), (3.4), and (3.6) leads to

$$P_S(0) = \frac{1 + (|\mathcal{X}| - 1)(1 - |\mathcal{X}|p^*_{n,i})^w}{|\mathcal{X}|}, \tag{3.7}$$

which, together with the observation that $\mathbf{H}^{(i)}$ has Δ independent rows, implies

$$Pr\{\mathbf{H}^{(i)}x^n = 0\} = \left[\frac{1 + (|\mathcal{X}| - 1)(1 - |\mathcal{X}|p^*_{n,i})^w}{|\mathcal{X}|}\right]^\Delta.$$

This completes the proof of Lemma 2. □

In the following theorem, we analyze the performance of the modified (low-density) linear IED schemes $\{\mathcal{I}_n\}$. Note that for the purpose of characterizing the tradeoff between symbol error probability and density, instead of $P(\mathcal{I}_n)$, we are interested in the symbol error probability $D(\mathcal{I}_n)$ defined as the normalized Hamming distance between the decoder output $\hat{X}^n$ and the source sequence X^n resulting from using $\mathcal{I}_n$ to encode and decode X^n with the decoder only side information sequence Y^n.

Theorem 3. *Let $\{\epsilon_n\}$ be a sequence of real numbers. Suppose that $\mathcal{I}_n$ is constructed by using the distributions $\mathbf{p}_{n,0}, \mathbf{p}_{n,1}, \cdots$ defined by (3.1) such that*

$$\lim_{n\to\infty} \sum_{i=0}^{n/\Delta} \log(1 + (|\mathcal{X}| - 1)(1 - |\mathcal{X}|p^*_{n,i})^{n\epsilon_n}) < \log|\mathcal{X}|. \tag{3.8}$$

Then for any stationary, ergodic source-side information pair (X, Y)

$$\lim_{n\to\infty} r_f(\mathcal{I}_n) = H(X|Y) \text{ with probability one,} \tag{3.9}$$

$$r_b(\mathcal{I}_n) = O(\frac{1}{\sqrt{n}}), \text{ and} \tag{3.10}$$

$$D(\mathcal{I}_n) \le \epsilon_n + O(2^{-\sqrt{n}}). \tag{3.11}$$

Proof. Recall that $\Delta \sim \sqrt{n}$. Equations (3.9) and (3.10) can then be proved by using an argument similar to that used in the proof of Theorem 1. Thus it remains to upper bound the symbol error probability $D(\mathcal{I}_n)$ to complete the proof of Theorem 3.

For any real number $\alpha \in [0,1]$ and any sequence $x^n \in \mathcal{X}^n$, define

$$B^{\alpha}(x^n) \stackrel{\Delta}{=} \left\{ v^n \in \mathcal{X}^n : \frac{1}{n} \left| \{ i : v_i \neq x_i \} \right| \leq \alpha \right\}.$$

Observe that one of the following events must happen: $\hat{X}^n \in B^{\epsilon_n}(X^n)$ or $\hat{X}^n \notin B^{\epsilon_n}(X^n)$. In the case where the event $\hat{X}^n \in B^{\epsilon_n}(X^n)$ happens, we immediately have

$$D(\mathcal{I}_n) \leq \epsilon_n. \tag{3.12}$$

In the following, we shall focus on upper bounding $D(\mathcal{I}_n)$ in the case where $\hat{X}^n \notin B^{\epsilon_n}(X^n)$.

For convenience, let us define a sequence of indicator random variables $\{E_l\}_{l=0}^{\frac{n}{\Delta}}$ as follows. For $0 \leq l \leq \frac{n}{\Delta}$,

$$E_l = \begin{cases} 1 & \text{if there exists a sequence } v^n \text{ from } \mathcal{X} \text{ such that } v^n \neq X^n,\ v^n \in B^{\epsilon_n}(X^n), \\ & \mathbf{H}^{(i)}(X^n - v^n) = 0 \text{ for } 0 \leq i \leq l, \text{ and } h_n(v^n|Y^n) \leq \frac{l\Delta \log|\mathcal{X}|}{n} \\ 0 & \text{otherwise} \end{cases}.$$

Clearly, the right-hand-side of the equation in Lemma 2 is decreasing as the function of $w(x^n)$. It follows from this observation, Lemma 2, and the definition of $B^{\epsilon_n}(x^n)$ that for any $x^n \in \mathcal{X}^n$,

$$Pr\{\mathbf{H}^{(i)}(x^n - v^n) = 0\} \leq \left[\frac{1 + (|\mathcal{X}| - 1)(1 - |\mathcal{X}| p^*_{n,i})^{n\epsilon_n}}{|\mathcal{X}|} \right]^{\Delta}, \tag{3.13}$$

which, together with Lemma 1, leads to

$$\begin{aligned} & Pr\{E_l = 1 | X^n = x^n, Y^n = y^n\} \\ \leq & \left| A^{\frac{l\Delta \log|\mathcal{X}|}{n}}(y^n) \cap B^{\epsilon_n}(x^n) \right| \prod_{i=0}^{l} \left[\frac{1 + (|\mathcal{X}| - 1)(1 - |\mathcal{X}| p^*_{n,i})^{n\epsilon_n}}{|\mathcal{X}|} \right]^{\Delta} \\ \leq & 2^{\Delta(-\log|\mathcal{X}| + \sum_{i=0}^{l} \log(1 + (|\mathcal{X}|-1)(1-|\mathcal{X}| p^*_{n,i})^{n\epsilon_n}))}. \end{aligned} \tag{3.14}$$

Since (3.8) implies that when n is sufficiently large, there exists a small real constant $\delta > 0$ such that for all $0 \leq l \leq n/\Delta$,

$$\sum_{i=0}^{l} \log(1 + (|\mathcal{X}| - 1)(1 - |\mathcal{X}| p^*_{n,i})^{\epsilon_n n}) \leq \log|\mathcal{X}| - \delta. \tag{3.15}$$

Putting (3.15) back into (3.14), we have

$$Pr\{E_l = 1 | X^n = x^n, Y^n = y^n\} \leq 2^{-\delta\sqrt{n}\log|\mathcal{X}|},$$

which in turn implies

$$Pr\{E_l = 1\} \leq 2^{-\delta\sqrt{n}\log|\mathcal{X}|}. \tag{3.16}$$

It then follows from the union bound that under the condition $\hat{X}^n \notin B^{\epsilon_n}(X^n)$,

$$D(\mathcal{I}_n) \;\leq\; (\frac{n}{\Delta}+1)2^{-\delta\sqrt{n}\log|\mathcal{X}|} = O(2^{-\sqrt{n}}). \tag{3.17}$$

Because of (3.12) and (3.17), this completes the proof of Theorem 3. □

Let us take a closer look at equation (3.8). Note that $\ln(1+x) \leq x$ for all $|x| < 1$. Thus when $p^*_{n,i}$ is small for all i,

$$\begin{aligned} &\sum_{i=0}^{n/\Delta} \log(1 + (|\mathcal{X}|-1)(1-|\mathcal{X}|p^*_{n,i})^{n\epsilon_n}) \\ \leq\; &\sum_{i=0}^{n/\Delta} \frac{1}{\ln 2}(|\mathcal{X}|-1)\exp(-n\epsilon_n|\mathcal{X}|p^*_{n,i}). \end{aligned} \tag{3.18}$$

Accordingly it is sufficient to require

$$\sum_{i=0}^{n/\Delta} \exp(-n\epsilon_n|\mathcal{X}|p^*_{n,i}) \leq \frac{\ln 2(\log|\mathcal{X}|)}{|\mathcal{X}|-1}. \tag{3.19}$$

In a special case where $p^*_{n,i} = p^*_n$ for all i, we have

$$\exp(-n\epsilon_n|\mathcal{X}|p^*_n) \leq \frac{\ln 2(\log|\mathcal{X}|)}{(\frac{n}{\Delta}+1)(|\mathcal{X}|-1)} \tag{3.20}$$

which, together with $\Delta \sim \sqrt{n}$, leads to

$$\epsilon_n p^*_n = \Omega(\frac{\log n}{n}). \tag{3.21}$$

Remark 3. *Equations (3.21) and (3.11) together quantify the tradeoff between the symbol error probability and the density of the linear IED scheme $\mathcal{I}_n$. In other words, in order to maintain asymptotic optimality, one has to balance the computational complexity quantified by p^*_n and the resulting symbol error probability according to (3.11) under our construction of $\mathcal{I}_n$.*

Equation (3.21) is a sufficient condition under which the construction of asymptotic optimal linear IED schemes with the desired density is possible. Though we conjecture that for (X,Y), (3.21) must be held by linear IED schemes achieving $H(X|Y)$ at the speed of $O(1/\sqrt{n})$, establishing such a necessary condition is still an open problem at the moment of writing this paper. Finally we note that the above conjecture reflects our belief that the three fundamental quantities characterizing an IED scheme: error probability, rate, and density, are all connected.

Another remark here is that the result also can be extended into stationary nonergodic source -side information pairs, with the same technology mentioned in section C) 3.1.

From the theorem above, essentially, as long as $p_*^n = \omega(\frac{\log n}{n})$, the bit error probability ϵ will go to zero asymptotically. Even if p_*^n is on the order of $\frac{\log n}{n}$, it can still be shown that the bit error probability goes to zero. However, we have to make a small modification of coding scheme at this point. Instead of starting from step 1, the encoding goes to $r\sqrt{n}$ step at the beginning, which means that the parity-check matrix has non-negligible rate r in the first interaction. The rate r is the initial rate of this interactive encoding and decoding scheme.

Corollary 1. *Let*

$$p_*^n = \frac{1}{2|\mathcal{X}|\epsilon}\frac{\ln n - 2\ln\beta}{n} \tag{3.22}$$

where $\epsilon \leq \frac{1}{2}$ and β is a constant set to satisfy criterion (3.20)*, i.e.*

$$\beta = \frac{\ln 2(\log|\mathcal{X}| - C_0)}{|\mathcal{X}| - 1} \tag{3.23}$$

Also for large n, there exists a constant $0 < \theta \leq 1$, such that

$$p_*^n \geq \frac{\theta}{2|\mathcal{X}|\epsilon}\frac{\ln n}{n} \tag{3.24}$$

Then if the initial rate r satisfies

$$r \geq \max\{f_1(\epsilon), f_2(\epsilon)\} \tag{3.25}$$

where

$$f_1(\epsilon) = \frac{H(\epsilon)}{\ln\left(\frac{|\mathcal{X}|}{1+(|\mathcal{X}|-1)e^{-\frac{\theta}{2\sqrt{\epsilon}}}}\right)} \tag{3.26}$$

and

$$f_2(\epsilon) = \frac{3|\mathcal{X}|}{2(|\mathcal{X}|-1)}\left(\sqrt{\epsilon} + \frac{2\epsilon}{\theta}\right) \tag{3.27}$$

for the code in the first interaction

$$Pr\{D \leq \epsilon n\} = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right) \to 0 \tag{3.28}$$

where D is the minimum hamming distance of this code.

Proof. By union bound,

$$\begin{aligned} Pr\{D \leq n\epsilon\} &\leq \sum_{l=1}^{n\epsilon}\binom{n}{l}\left(\frac{1+(|\mathcal{X}|-1)(1-|\mathcal{X}|p_n^*)^l}{|\mathcal{X}|}\right)^{nr} \\ &\leq \sum_{l=1}^{n\epsilon}\binom{n}{l}\left(\frac{1+(|\mathcal{X}|-1)(1-\frac{\theta}{2\epsilon}\frac{\ln n}{n})^l}{|\mathcal{X}|}\right)^{nr} \\ &\leq \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}}\cdot + \sum_{l=\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon}\cdot \end{aligned} \tag{3.29}$$

where

$$\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \cdot = \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \left(\begin{array}{c} n \\ l \end{array} \right) \left(\frac{1 + (|\mathcal{X}| - 1)(1 - \frac{\theta}{2\epsilon}\frac{\ln n}{n})^l}{|\mathcal{X}|} \right)^{nr} \tag{3.30}$$

and

$$\sum_{l=\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \cdot = \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \left(\begin{array}{c} n \\ l \end{array} \right) \left(\frac{1 + (|\mathcal{X}| - 1)(1 - \frac{\theta}{2\epsilon}\frac{\ln n}{n})^l}{|\mathcal{X}|} \right)^{nr} \tag{3.31}$$

For $\sqrt{\epsilon}\frac{n}{\ln n} \leq l \leq n\epsilon$,

$$\begin{aligned} \left(1 - \frac{\theta}{2\epsilon}\frac{n}{\ln n}\right)^l &\leq \exp\left(-\frac{\theta}{2\epsilon}\frac{l \ln n}{n}\right) \\ &\leq \exp\left(-\frac{\theta}{2\sqrt{\epsilon}}\right) \end{aligned} \tag{3.32}$$

Therefore

$$\sum_{\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \cdot \leq \left[\sum_{\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \left(\begin{array}{c} n \\ l \end{array} \right) \right] \left(\frac{1 + (|\mathcal{X}| - 1)\exp\left(-\frac{\theta}{2\sqrt{\epsilon}}\right)}{|\mathcal{X}|} \right)^{nr} \tag{3.33}$$

Note that

$$\begin{aligned} \sum_{\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \left(\begin{array}{c} n \\ l \end{array} \right) &\leq \sum_{l=1}^{n\epsilon} \left(\begin{array}{c} n \\ l \end{array} \right) \\ &= \left(\begin{array}{c} n \\ n\epsilon \end{array} \right) \sum_{l=0}^{n\epsilon-1} \frac{\left(\begin{array}{c} n \\ n\epsilon - l \end{array} \right)}{\left(\begin{array}{c} n \\ n\epsilon \end{array} \right)} \\ &= \left(\begin{array}{c} n \\ n\epsilon \end{array} \right) \sum_{l=0}^{n\epsilon-1} \prod_{i=1}^{l} \frac{n\epsilon - i + 1}{n - n\epsilon + i} \\ &\leq \left(\begin{array}{c} n \\ n\epsilon \end{array} \right) \sum_{l=0}^{n\epsilon-1} \left(\frac{n\epsilon}{n - n\epsilon + 1} \right)^l \\ &\leq \left(\begin{array}{c} n \\ n\epsilon \end{array} \right) \sum_{l=0}^{n\epsilon-1} \left(\frac{\epsilon}{1 - \epsilon} \right)^l \\ &\leq \left(\begin{array}{c} n \\ n\epsilon \end{array} \right) \frac{1 - \epsilon}{1 - 2\epsilon} \end{aligned} \tag{3.34}$$

Note that at the beginning, we assume $\epsilon < \frac{1}{2}$.

From striling formula, we know that

$$\left(\begin{array}{c} n \\ n\epsilon \end{array} \right) \leq \frac{1}{\sqrt{2\pi n\epsilon(1 - \epsilon)}} \exp nH(\epsilon) \tag{3.35}$$

where

$$H(\epsilon) = -\epsilon \ln \epsilon - (1-\epsilon)\ln(1-\epsilon)$$

Therefore

$$\sum_{\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \cdot \leq \frac{1}{1-2\epsilon}\sqrt{\frac{1-\epsilon}{2\pi n\epsilon}} \exp n \left(H(\epsilon) + r \ln \left(\frac{1 + (|\mathcal{X}|-1)\exp\left(-\frac{\theta}{2\sqrt{\epsilon}}\right)}{|\mathcal{X}|} \right) \right) \tag{3.36}$$

It is not difficult to observe that as long as r satisfies

$$r \geq \frac{H(\epsilon)}{\ln\left(\frac{|\mathcal{X}|}{1+(|\mathcal{X}|-1)\exp\left(-\frac{\theta}{2\sqrt{(\epsilon)}}\right)}\right)} = f_1(\epsilon) \tag{3.37}$$

then

$$\sum_{\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \cdot \leq C_1 \frac{1}{\sqrt{n}}$$

where

$$C_1 = \frac{1}{1-2\epsilon}\sqrt{\frac{1-\epsilon}{2\pi\epsilon}}$$

Now we would like to bound $\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \cdot$. Before that, two inequalities are proved here. One is that

$$\exp(-x) \leq 1 - \frac{x}{1+x} \tag{3.38}$$

for any $x > 0$, which comes from

$$\exp(x) > 1 + x \tag{3.39}$$

And the other inequality

$$\left(\begin{array}{c} n \\ l \end{array} \right) \leq \left(\frac{en}{l}\right)^l \tag{3.40}$$

for $1 \leq l \leq n$, is a little tricky. We prove it by induction. It is easy to see this inequality holds for $l = 1$, which is

$$n \leq ne$$

as $e > 1$. Now suppose that

$$\left(\begin{array}{c} n \\ k \end{array} \right) \leq \left(\frac{en}{k}\right)^k$$

then

$$\begin{aligned}\binom{n}{k+1} &= \binom{n}{k}\frac{n-k}{k+1} \\ &\le \left(\frac{en}{k}\right)^k \frac{n-k}{k+1} \\ &= \left(\frac{en}{k+1}\right)^{k+1} \frac{n-k}{n}\frac{1}{e}\left(1+\frac{1}{k}\right)^k \\ &\le \left(\frac{en}{k+1}\right)^{k+1}\end{aligned}$$

where the last step is due to that $\left(1+\frac{1}{k}\right)^k$ is a increasing function of k, and it is well known that

$$\lim_{k\to\infty}\left(1+\frac{1}{k}\right)^k = e$$

Now for $1 \le l \le \sqrt{\epsilon}\frac{n}{\ln n}$, we have

$$\begin{aligned}\left(1-\frac{\theta}{2\epsilon}\frac{\ln n}{n}\right)^l &\le \exp\left(-\frac{\theta}{2\epsilon}\frac{l\ln n}{n}\right) \\ &\le 1-\frac{\frac{\theta}{2\epsilon}\frac{l\ln n}{n}}{1+\frac{\theta}{2\epsilon}\frac{l\ln n}{n}} \\ &\le 1-\frac{\frac{\theta}{2\epsilon}\frac{l\ln n}{n}}{1+\frac{\theta}{2\sqrt{\epsilon}}} \\ &= 1-\frac{l\ln n}{n}\frac{1}{\sqrt{\epsilon}+\frac{2\epsilon}{\theta}} \qquad (3.41)\end{aligned}$$

Therefore

$$\begin{aligned}\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \cdot &\le \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \binom{n}{l}\left(\frac{1+(|\mathcal{X}|-1)\left(1-\frac{l\ln n}{n}\frac{1}{\sqrt{\epsilon}+\frac{2\epsilon}{\theta}}\right)}{|\mathcal{X}|}\right)^{nr} \\ &= \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \binom{n}{l}\left(1-\frac{l\ln n}{n}\frac{|\mathcal{X}|-1}{|\mathcal{X}|\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right)}\right)^{nr} \\ &\le \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \binom{n}{l}\exp\left(-l\ln n\frac{(|\mathcal{X}|-1)r}{|\mathcal{X}|\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right)}\right) \\ &\le \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \left(\frac{en}{l}\right)^l \exp\left(-l\ln n\frac{(|\mathcal{X}|-1)r}{|\mathcal{X}|\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right)}\right) \\ &= \sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \left(\frac{e}{l}\right)^l \exp\left(-l\ln n\left(\frac{(|\mathcal{X}|-1)r}{|\mathcal{X}|\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right)}-1\right)\right) \qquad (3.42)\end{aligned}$$

when

$$\frac{(|\mathcal{X}|-1)r}{|\mathcal{X}|\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right)} > 1$$

i.e. $\exists\delta$ such that

$$r = \frac{|\mathcal{X}|(1+\delta)}{(|\mathcal{X}|-1)}\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right)$$

then

$$\begin{aligned}\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \cdot &\leq \exp(-\delta\ln n)\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}}\left(\frac{e}{l}\right)^l \\ &= \left(\frac{1}{n}\right)^{\delta}\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}}\left(\frac{e}{l}\right)^l \end{aligned} \tag{3.43}$$

where

$$\begin{aligned}\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}}\left(\frac{e}{l}\right)^l &\leq \sum_{l=1}^{\infty}\left(\frac{e}{l}\right)^l \\ &= e+\left(\frac{e}{2}\right)^2+\sum_{l=3}^{\infty}\left(\frac{e}{l}\right)^l \\ &\leq e+\left(\frac{e}{2}\right)^2+\sum_{l=0}^{\infty}\left(\frac{e}{3}\right)^l \\ &= e+\left(\frac{e}{2}\right)^2+\frac{e}{3-e} \triangleq C_2 \end{aligned} \tag{3.44}$$

which means

$$\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \cdot \leq C_2\left(\frac{1}{n}\right)^{\delta} \tag{3.45}$$

Now let $\delta = \frac{1}{2}$, then as long as

$$r \geq \frac{2|\mathcal{X}|}{3(|\mathcal{X}|-1)}\left(\sqrt{\epsilon}+\frac{2\epsilon}{\theta}\right) = f_2(\epsilon)$$

then

$$\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}}\left(\frac{e}{l}\right)^l \leq C_2\frac{1}{\sqrt{n}} \tag{3.46}$$

The bounds on $\sum_{l=1}^{\sqrt{\epsilon}\frac{n}{\ln n}} \cdot$ and $\sum_{\sqrt{\epsilon}\frac{n}{\ln n}}^{n\epsilon} \cdot$ together complete the proof. □

Remark 4. *Note that through the interaction, when number of rows of parity check matrix used in IED scheme increases, the hamming distance of corresponding linear code will not decrease. In other word, the hamming distance of the linear code at the first interaction is the least hamming distant of linear codes used in IED schemes. As the corollary shows that the least hamming distant is greater than* $n\epsilon$ *with probability one in the modified interactive encoding and decoding scheme, together with the proof of theorem 3, it essentially shows that the bit error probability goes to 0 in this modified version of IED schemes. Moreover, from equation (*(3.25)*) (3.26) and (3.27), as well as the fact that* $f_1(\epsilon), f_2(\epsilon) \to 0$ *when* $\epsilon \to 0$ *,the initial rate can be very close to 0 as long as* ϵ *is close to 0. The density of matrix* p_*^n *will increase as* ϵ *decreases. But* p_*^n *will be on the order of* $\frac{\log n}{n}$ *if* ϵ *is a constant respective to block length n. Also, for the source-side information pairs of non-negligible conditional entropy rate, it is always possible to set initial rate far smaller than the entropy rate such that the probability that a sequence of pair produced by this source can be encoded and decoded successfully by a parity-check matrix with rate less than initial rate goes to zero. In the other word, rate loss due to the modification of coding scheme goes to zero.*

Chapter 4

Linear IED schemes with LDPC code and BP decoding

In the previous chapter, we illustrate that the linear IED scheme, specifically the low density IED scheme can achieve SW bound universally with asymptotically zero bit error probability. Although low density parity check matrix used in the scheme reduces the complexity of encoding, complexity of decoding, which is assumed ML decoding, is still too high. Therefore, for the practical coding scheme, LDPC code and BP decoding is utilized, to guarantee the complexity of coding scheme linear with block length.

4.1 Practical Coding Scheme – LDPC Approach

The practical coding scheme is as follows: In each step i, both the encoder and the decoder know the low density matrix H_i. For the encoder, it obtains syndrome by $S_i = H_i * X$, and sends S_i to the decoder, while the decoder tries to solve the problem $\max_{H_i * X = S_i} Pr\{X|Y\}$ [25] using the sum product algorithm. If the decoder succeeds to converge, and the output passes through the threshold criterion, then it sends back one bit 1, indicating the success of decoding and terminating the coding procedure for this block. Otherwise, it sends back one bit 0, indicating the failure of this step, and the encoder will enlarge the matrix and the whole procedure goes to next step. Below is the schematic graph.

4.1.1 Interactive Encoding

As Syndrome Encoding is to perform $S = H * X$, and this operation can be done efficiently, the main issue of encoding is how to get the matrix H_i. One way is to generate a $n \times n$ matrix H_n, where n is the block length, and each step select several rows of H_n. However, this approach is not good for the reason that it will change the distribution of check node degrees, and variable node degrees in each step, and

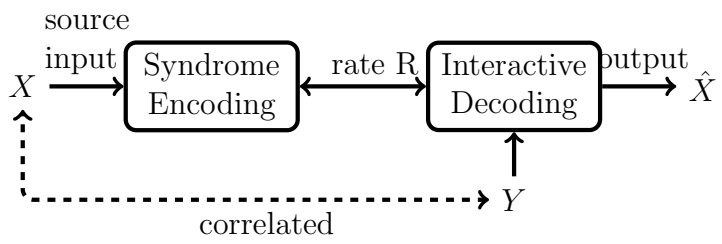

Figure 4.1: Practical Coding Scheme

to keep the distribution optimized for each step is not traceable. Instead, we keep the variable node degrees unchanged, and split several check nodes to generate new syndromes. The procedure follows:

1: Given a Low Density Parity Check Matrix H.
2: **while** more syndromes are needed **do**
3: Find the check node C_m with maximum degrees in current H.
4: Partition the neighbours of C_m into two sets, S_1 and S_2 such that $||S_1| - |S_2|| \leq 1$.
5: Delete the edges between C_m and variable nodes in S_2.
6: Create a new check node C_n, with neighbours set S_2, and corresponding syndrome.
7: Update H according to the change of its tanner graph.
8: **end while**

Figure 4.2: Splitting Syndrome Encoding

Remark 1. *Note that this Splitting Syndrome Encoding algorithm is similar to LDPC Accumulate Codes in [42]. But there are two major difference. Fist of all, our coding scheme design follows the theoretical result obtained in [48], [47] and previous chapter. Therefore, number of syndrome split in each step is designed carefully such that rate-less between the total rate exchanged between encoder and decoder and bit error probability is low. However, in [42], actually one syndrome is split in each step. There is no guarantee that bit error probability is low, and the total rate exchanged between encoder and decoder is more than twice of the conditional entropy of source given side information. Second, splitting method is designed according to density evolution analysis. Actually every time the algorithm selects the check node with maximum degrees and split its neighbours as even as possible. The rational behind it is that in this way $\rho'(1)$ is reduced most, where $\rho(x)$ is check node distribution of LDPC codes[26]. More specificly,*

$$\begin{aligned} \rho'(1) &= \sum_j (j-1)\rho_j \\ &= \sum_j j\rho_j - 1 \end{aligned} \tag{4.1}$$

while if a check node with degree j_0 is split to two check nodes, with degree j_1 and j_2, then

$$\begin{aligned}\hat{\rho}'(1) &= \rho'(1) - \frac{j_0^2 - j_1^2 - j_2^2}{N} \\ &= \rho'(1) - \frac{2j_1 j_2}{N} \end{aligned} \tag{4.2}$$

where $j_0 = j_1 + j_2$ and N is the total number of edges. Apparently, the way to split above enlarge the amount of reduction of $\rho'(1)$ most. From density evolution analysis, $\rho'(1)$ is critical to the performance of codes using message passing algorithm, whether message is discrete or continuous.

4.1.2 Density Evolution Analysis

For the irregular LDPC code[26], let $(\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_{|\Lambda|}})$ and $(\rho_{j_1}, \rho_{j_2}, \ldots, \rho_{j_{|\Omega|}})$ describe the variable node and check node degree distribution, where $\lambda_i(\rho_i)$ represents the fraction of edges connecting to the variable(check) node with degree i, and $\Lambda = \{i_1, i_2, \ldots, i_{|\Lambda|}\}(\Omega = \{j_1, j_2, \ldots, j_{|\Omega|}\})$ represents the set of variable(check) degree with nonzero probability. Therefore,

$$\sum_{i\in\Lambda} \lambda_i = \sum_{j\in\Omega} \rho_j = 1 \tag{4.3}$$

As the total number of edge is determined by matrix, whether from viewpoint of variable node or check node, so automatically there is a constrain:

$$\frac{1}{R}\sum_{i\in\Lambda} \lambda_i / i = \sum_{j\in\Omega} \rho_j / j \tag{4.4}$$

where R is the ratio of number of rows and columns of matrix, which corresponds to compression rate.

Now if we define $\lambda(x)$ and $\rho(x)$ as

$$\lambda(x) = \sum_{i\in\Lambda} \lambda_i x^{i-1} \tag{4.5}$$

$$\rho(x) = \sum_{j\in\Omega} \rho_j x^{j-1} \tag{4.6}$$

Note usually it is assumed that $\min_{i\in\Lambda} i > 1$ and $\min_{j\in\Omega} j > 1$, as check node with degree 1 is only useful for a single bit, and we can simply expurgate this row from the parity check matrix. For variable node with degree 1, this may involve constant error probability no matter how many iterations it is applied.

Density evolution essentially is to analyse the performance of decoding for particular ensemble of codes, which is determined by check and variable node degrees. Based on density evolution, we investigate several well-known message passing algorithm for LDPC decoding, with discrete or continuous alphabet, and show the role of $\rho'(1)$ in code performance.

Discrete Alphabets

In this kind of decoding algorithms, we investigate two algorithms proposed by Gallager[13], called Gallager Algorithm I and Gallager Algorithm II. Detail of this algorithm will be found in Gallager's paper cited above. For binary symmetrical channel, density evolution shows that the bit error probability in i-th iteration p_i satisfies the following recursive function:

$$\begin{aligned} p_{i+1} &= p_0 - p_0\lambda\left(\frac{1+\rho(1-2p_i)}{2}\right) \\ &\quad + (1-p_0)\lambda\left(\frac{1-\rho(1-2p_i)}{2}\right) \end{aligned} \tag{4.7}$$

where $p_0 = \epsilon$, the transmission error of channel.

For Gallager Algorithm II, the density evolution shows that:

$$\begin{aligned} &p_{i+1} = p_0 - \\ &p_0 \sum_{j\in\Lambda}\lambda_j \sum_{t=b_{i,j}}^{j-1}\left(\begin{array}{c} j-1 \\ t \end{array}\right) g(\rho(1-2p_i), t, j) + (1-p_0) \\ &\sum_{j\in\Lambda}\lambda_j \sum_{t=b_{i,j}}^{j-1}\left(\begin{array}{c} j-1 \\ t \end{array}\right) g(-\rho(1-2p_i), t, j) \end{aligned} \tag{4.8}$$

where

$$b_{i,j} = \lceil (j-1+z_i)/2 \rceil \tag{4.9}$$

$$z_i = \frac{\log((1-p_0)/p_0)}{\log((1+\rho(1-2p_i))/(1-\rho(1-2p_i)))} \tag{4.10}$$

$$g(y,t,j) = \left(\frac{1+y}{2}\right)^t \left(\frac{1-y}{2}\right)^{j-1-t} \tag{4.11}$$

Now let us consider p_{i+1} a function of $\rho_i = \rho(1-2p_i)$, for both of the algorithms above we have the following result:

Proposition 1. *p_{i+1} is a decreasing function of ρ_i.*

Proof. For non-relaxed version, that is equation (4.7), it is quite straightforward, as $0 \le \rho_i \le 1$ and $\lambda(x)$ is an increasing function in $(0,1)$. For relaxed version, it is not very obvious. In fact, given $j, t \ge b_{i,j}$, $g(y,t,j)$ is an increasing function of y in $(-1,1)$, and then proposition 1 is proved. Now let us differential the $g(y,t,j)$,

$$\begin{aligned} \frac{dg(y,t,j)}{dy} &= \frac{1}{2}\left(\frac{1+y}{2}\right)^{t-1}\left(\frac{1-y}{2}\right)^{j-2-t} \\ &\quad \times \left(t-(j-1)\frac{1+y}{2}\right) \end{aligned} \tag{4.12}$$

Here $-1 \le y \le 1$, therefore $0 \le \frac{1\pm y}{2} \le 1$, and $t \ge b_{i,j} \ge j-1$. Proof is completed. □

Therefore, to increase the speed of convergence of decoding algorithm, ρ_i should be as large as possible, given the same p_i. Consider the case when p_i is very small, and ρ_i is determined by the behaviour of the function ρ around 1. Now apply Taylor expansion to ρ at 1, we get

$$\rho_i = \rho(1-2p_i) = \rho(1) + \rho'(1)(-2p_i) + o(p_i) \tag{4.13}$$

where

$$\rho'(1) = \left.\frac{d\rho(x)}{dx}\right|_{x=1} = \sum_{j\in\Omega}(j-1)\rho_j = \sum_{j\in\Omega} j\rho_j - 1 \tag{4.14}$$

As $\rho(1) = 1$ for any check node degree distribution, this means that it is better to select check node distribution such that $\rho'(1)$ is as small as possible.

Continuous Alphabets – Belief Propagation

When the message is continuously distributed, usually it is impossible to find the density evolution function directly for the error probability of message[34].

Instead, the recursive function is between the distribution or density of message in each iteration. For any binary memoryless symmetric channel, we have stability conditions, the sufficiency part of which states decoding will succeed if $r\lambda'(0)\rho'(1) \leq 1$, where

$$r = \int P_0(x) e^{-\frac{x}{2}} dx \tag{4.15}$$

and $P_0(x)$ is the initial L-density, or density of initial Log-Likelihood Ratio. Obviously if $\lambda(x)$ is fixed, $\rho'(1)$ should be as small as possible for the sake of correction ability of codes.

For asymmetric memoryless channels[44], sufficient stability condition tells us that error threshold ϵ^* is the smallest strictly positive root of the equation:

$$\lambda(\rho'(1)\epsilon)r = \epsilon \tag{4.16}$$

where

$$r = \int \langle P_0(x)\rangle e^{-\frac{1}{2}} dx \tag{4.17}$$

where $\langle P_0(x)\rangle$ is the initial L-density averaging over all possible codewords. Again considering $\rho'(1)$, we have

Proposition 2. *ϵ^* is a decreasing function of $\rho'(1)$.*

Proof. Suppose $\rho_1'(1) > \rho_2'(1)$, we have to show that $\epsilon_1^* < \epsilon_2^*$. Now we reformulate the equation by substituting $\bar{\epsilon} = \rho'(1)\epsilon$:

$$\rho'(1)r\lambda(\bar{\epsilon}) = \bar{\epsilon} \tag{4.18}$$

or

$$\rho'(1) = \frac{\bar{\epsilon}}{r\lambda(\bar{\epsilon})} \tag{4.19}$$

Now we can consider $\rho'(1)$ the function of $\bar{\epsilon}$. As the smallest variable degree is larger than 1, then $\frac{\lambda(\bar{\epsilon})}{\bar{\epsilon}}$ is still polynomial. In addition, $r > 0$ and coefficients of λ are all positive, then $\rho'(1)$ is decreasing function of $\bar{\epsilon}$. If $\rho_1'(1) > \rho_2'(1)$, then $\bar{\epsilon_1} < \bar{\epsilon_2}$. Then

$$\begin{aligned} \frac{\epsilon_1^*}{\epsilon_2^*} &= \frac{\rho_2'(1)}{\rho_1'(1)} \frac{\bar{\epsilon_1}}{\bar{\epsilon_2}} \\ &= \frac{\lambda(\bar{\epsilon_1})}{\lambda(\bar{\epsilon_2})} \\ &< 1 \end{aligned} \tag{4.20}$$

The first equation is from the relation between ϵ^* and $\bar{\epsilon}$, and the second from (4.18). The inequality is due to the fact that $\lambda(x)$ is an increasing function. □

In summary, for all those analysis, $\rho'(1)$ should be as small as possible. In fact, this also implies that even check distribution is optimal, given fixed variable node degree λ and rate R. Suppose $C = \frac{1}{R}\sum \frac{\lambda_i}{i}$, we know that in addition to the constraint $\sum \rho_j = 1$, the check node degree ρ also has to satisfy $C = \sum \frac{\rho_j}{j}$. Then we have the following result:

Proposition 3. *If C is an integer, then the case that check node degrees all equal to C yields the best performance in the sense mentioned above.*

Proof. This is a straightforward result apply Cauchy-Schwartz inequality. That is,

$$\begin{aligned} \left(\sum_{j\in\Omega} j\rho_j\right)\left(\sum_{j\in\Omega} \frac{\rho_j}{j}\right) &\geq \left[\sum_{j\in\Omega} \sqrt{(j\rho_j)}\sqrt{\left(\frac{\rho_j}{j}\right)}\right]^2 \\ &= 1 \end{aligned} \tag{4.21}$$

which means that $\sum_{j\in\Omega} j\rho_j \geq C$, and equality is satisfied if and only if j^2 is constant. This indicates that $\Omega = \{C\}$. □

When C is not an integer, it is impossible to make all check nodes degrees even. But it turns out that it should be as even as possible.

Proposition 4. *If C is not an integer, the best performance is achieved when $\Omega^*(C) = \{C_z, C_z + 1\}$, where $C_z = \lfloor C \rfloor$.*

Proof. First of all, it is obvious that $C \geq 1$, as

$$\frac{1}{C} = \sum_{j\in\Omega} \frac{\rho_j}{j} \leq \sum_{j\in\Omega} \rho_j = 1 \tag{4.22}$$

where the inequality is due to the fact that j is degree of check node, which is integer and larger than 1.

Then it can be shown that it is always possible to choose $\Omega^*(C)$ no matter what C is. In fact, for Ω and ρ, there are those two constraints (4.3) and (4.4). Therefore, if we substitute $\Omega^*(C)$ as Ω into those two constraints, we can solve the linear equations, and get

$$\rho_{C_z} = C_z\left(\frac{C_z+1}{C}-1\right) \tag{4.23}$$

$$\rho_{C_z+1} = (C_z+1)\left(1-\frac{C_z}{C}\right) \tag{4.24}$$

It is obvious that $0 \le \rho_{C_z}, \rho_{C_z+1} \le 1$ when $C \ge 1$. In this case,

$$\sum_{j\in\Omega} j\rho_j = C + \frac{C_m(1-C_m)}{C} \tag{4.25}$$

where $C_m = C - C_z$.

In general, suppose $\Omega = \{j_1, \ldots, j_d\}$ is fixed, whose elements are in ascend order. Then the problem turns out to be

$$\min_{\sum \rho_j=1, \sum \rho_j/j=\frac{1}{C}} \sum_{j\in\Omega} j\rho_j = \min\{\vec{\rho}^T\vec{J} | \vec{\rho} \ge 0, \vec{\rho}^T A = \vec{e}^T\} \tag{4.26}$$

where

$$A = \begin{pmatrix} 1 & \frac{1}{j_1} \\ 1 & \frac{1}{j_2} \\ \vdots & \vdots \\ 1 & \frac{1}{j_d} \end{pmatrix}, \vec{e} = \begin{pmatrix} 1 \\ \frac{1}{C} \end{pmatrix}$$

According to duality theorem of linear programming, this minimization equals to

$$\begin{aligned} \max\{\vec{e}^T x | Ax \le \vec{J}\} &= \max_{x_1+\frac{x_2}{j_i}\le j_i} \left(x_1 + \frac{x_2}{C}\right) \\ &= \frac{1}{C}\max_{j_i x_1 + x_2 \le j_i^2} (Cx_1 + x_2) \end{aligned} \tag{4.27}$$

Due to complementary slackness phenomenon, suppose $\vec{\rho_0}$ is the optimal vector, then $\rho_j \ne 0$ implies $j * x_1 + x_2 = j^2$, and as every $j \in \Omega$ is different with each other, any row of A is linearly independent with each other, which means there are at most two ρ_j is not zero. According to constraint (4.4), if C is not integer, there should be at least two $\rho_j \ne 0$, and at least one larger than C, at least one smaller

than C. Then we get that to minimize the objective function, we should pick some j_p and j_q, with $j_p < C$ and $j_q > C$. Solve the linear equation, we have

$$\min_{\sum \rho_j = 1, \sum \rho_j / j = \frac{1}{C}} j\rho_j = j_p + j_q - \frac{1}{C} j_p j_q \tag{4.28}$$

With $j_p < C$ and $j_q > C$, this function is a decreasing function with j_p and increasing function with j_q. Therefore to minimize this function given Ω, we should pick

$$j_p = \max\{j : j \in \Omega, j < C\} \tag{4.29}$$
$$j_q = \min\{j : j \in \Omega, j > C\} \tag{4.30}$$

Meanwhile we show that $\Omega^*(C)$ is indeed the optimal. □

Remark 2. *This proof has show us not only which check degree is optimal, but also how to choose check degrees when set of possible check node degrees has been given. Note that this result has been mentioned in [26] without proof.*

Therefore from this point of view, the best way to select node and partition is to try to make check node degree as even as possible in the structure of splitting.

4.2 Model-Training Interactive Decoding Algorithm

4.2.1 General Form of Decoding

Conventional iterative decoding algorithm of LDPC codes for memoryless channel bases on factor graph. General form is as follows:[24] where □ represents check

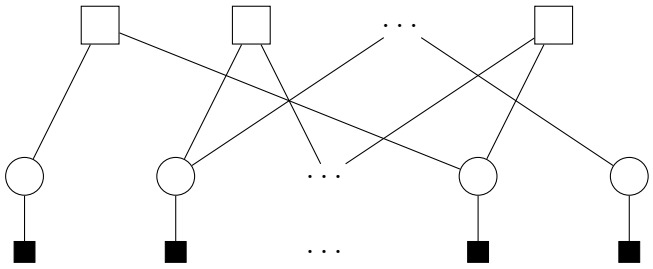

Figure 4.3: General LDPC Decoding for memoryless channels

node, ○ represents variable node, and ■ represents cost function for each variable node.

For arbitrary distribution, we propose a general form called model-training. "Model Training" tries to enforce the statistical property of model into decoding procedure. It is a multiple input multiple output function. Its input is the vector of marginal distribution calculated by factor graph in current step, and its output is the vector of marginal distribution, which is embedded some model constraints.

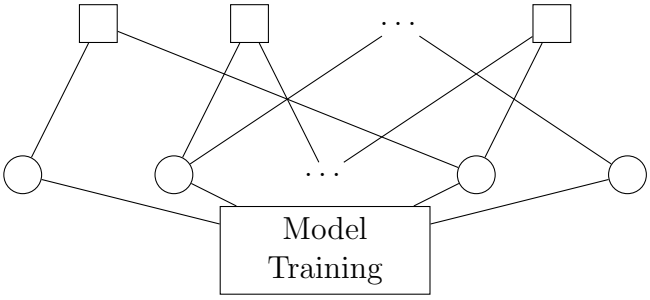

Figure 4.4: General LDPC Decoding for general channels

4.2.2 Non-anticipatory Finite State Channel - Channel with Memory

Finite state channel was first defined by Gallager[14], depicted in figure 4.5. The

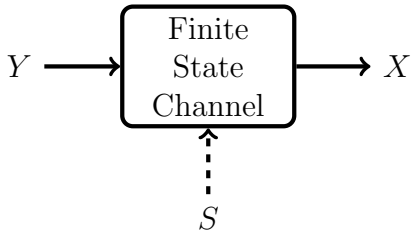

Figure 4.5: Finite State Channel

motivation to define finite state channel is to deal with channel with memory, i.e. output of channel X_i statistically depends current input Y_i, past input Y_1^{i-1} and past output X_1^{i-1}. Non-anticipatory assumption is that current output X_i is independent of future inputs Y_{i+1}^{∞} given current input Y_i and input output history Y_1^{i-1}, X_i^{i-1}, i.e. $Y_{i+1}^{\infty} \rightarrow (Y_1^i, X_1^{i-1}) \rightarrow X_i$ forms a markov chain. To generalise this notation, we use state of channel s_i. which denotes the effect of history on current output. However, to accommodate more general model, e.g. fading channel, the state can be related to, but not necessarily determined by the history of input and output. Therefore channel is defined by the conditional probability $Pr\{x_i s_{i+1} | y_i s_i\}$. As history of input and output have been taken into account in state, we have $(Y_1^{i-1}, X_1^{i-1}, S_1^{i-1}) \rightarrow (Y_i, S_i) \rightarrow (X_i, S_{i+1})$ as a markov chain. With non-anticipatory assumption, we require that $(Y_1^{i-1}, X_1^{i-1}, S_1^{i-1}, Y_{i+1}^{\infty}) \rightarrow (Y_i, S_i) \rightarrow (X_i, S_{i+1})$ forms a markov chain.

In decoding design for this model, the intuition is to utilize forward and backward propagation. And the essential problem is to how to calculate the backward transition probability efficiently, as we already know the forward transition probability is $Pr\{X_i, S_{i+1}|Y_i, S_i\}$. This problem will be answered by the following two propositions.

Proposition 5. *Given output history Y_1^{i-1} and initial channel state s_1, the current output x_i and state s_{i+1} depends only on current input y_i, and independent with future input Y_{i+1}^{∞}. In other word, $Y_{i+1}^{\infty} \to (Y_1^i, s_1) \to (X_i, S_{i+1})$ forms a markov chain.*

Proof. From definition,

$$\begin{aligned}
& Pr\{x_i s_{i+1}|Y_1^{\infty} s_1\} \\
= & \sum_{X_1^{i-1} S_2^i} Pr\{X_1^i S_2^{i+1}|Y_1^{\infty} s_1\} \\
= & \sum_{X_1^{i-1} S_2^i} \prod_{j=1}^{i} Pr\{x_j s_{j+1}|X_1^{j-1} S_1^j Y_1^{\infty}\} \\
= & \sum_{X_1^{i-1} S_2^i} \prod_{j=1}^{i} Pr\{x_j s_{j+1}|X_1^{j-1} S_1^j Y_1^{i}\} \\
= & \sum_{X_1^{i-1} S_2^i} Pr\{X_1^i S_2^{i+1}|Y_1^{i} s_1\} \\
= & Pr\{x_i s_{i+1}|Y_1^{i} s_1\} \qquad (4.31)
\end{aligned}$$

□

We can see that given input Y_1^n, x_i, s_{i+1} only depends on s_i. Actually it can be shown that similar property holds in the reverse direction.

Proposition 6. *Given input history and initial state (Y_1^{i-1}, S_1) and current input and next state (Y_i, S_{i+1}), current output and state (X_i, S_i) is independent with other future input, output and state $(Y_{i+2}^n, S_{i+2}^n, X_{i+1}^n)$. In the other word, $(Y_{i+2}^n, S_{i+2}^n, X_{i+1}^n) \to (Y_1^{i+1}, S_1, S_{i+1}) \to (X_i, S_i)$ forms a markov chain.*

Proof. From definition,

$$\begin{aligned}
& Pr\{x_i s_i|Y_1^n, s_1, S_{i+1}^{n+1}, X_{i+1}^n\} \\
= & \frac{Pr\{X_i^n S_i^{n+1}|Y_1^n, s_1\}}{Pr\{X_{i+1}^n S_{i+1}^n|Y_1^n, s_1\}} \\
= & \frac{Pr\{s_i|Y_1^n, s_1\} \prod_{j=i-1}^{n-1} Pr\{x_{j+1} s_{j+2}|s_{j+1} y_{j+1}\}}{Pr\{s_{i+1}|Y_1^n, s_1\} \prod_{j=i}^{n-1} Pr\{x_{j+1} s_{j+2}|s_{j+1} y_{j+1}\}} \\
= & \frac{Pr\{s_i|Y_1^{i-1}, s_1\}}{Pr\{s_{i+1}|Y_1^i, s_1\}} Pr\{x_i s_{i+1}|y_i s_i\} \\
= & Pr\{x_i s_i|Y_1^i, s_1, s_{i+1}\} \qquad (4.32)
\end{aligned}$$

□

Now if we consider linear block code, usually decoder does not know the exact value of s_1, which is channel state at the beginning of this block. However, as decoder has the output before, we can see that from decoder point of view, s_1 is independent with Y_1^n, and from the proof above, we can get

$$Pr\{x_i s_{i+1}|Y_1^n\} = Pr\{x_i s_{i+1}|Y_1^i\} \tag{4.33}$$

and

$$\begin{aligned} &Pr\{x_i s_i|Y_1^n, S_{i+1}^{n+1}, X_{i+1}^n\} \\ &= Pr\{x_i s_i|Y_1^i, s_{i+1}\} &(4.34)\\ &= \frac{Pr\{s_i|Y_1^{i-1}\}}{Pr\{s_{i+1}|Y_1^i\}} Pr\{x_i s_{i+1}|y_i s_i\} &(4.35)\end{aligned}$$

Remark 3. *marginalization, where in principle both forward and backward direction the sequence forms markov chain. Equation* (4.34) *provides the rational that we can also do the same thing. Note that for backward propagation, instead of* (Y_{i+1}, S_{i+2})*, we should use* (Y_1^{i+1}, S_{i+2}) *as condition. The forward and backward transition probability is related in equation* (4.35)*, which also provides the efficient way to calculate the backward transition probability. It will be shown that the complexity of calculation is linear with block length afterward.*

Now we can describe the algorithm for finite state channel. The message passing diagram is shown in figure 4.6, where $\boxed{\text{I}}$ depicts interfere node. In this diagram, message passed between check node an variable node is identical to the standard BP decoding, while message passed between variable node and inference node, as well as among inference nodes, is specified as follows. Suppose $\vec{q} = \{q_i(x_i), 1 \le i \le n\}$ is the message from variable node to interfere node, and $\vec{p} = \{p_i(x_i), 1 \le i \le n\}$ is the message in the reverse direction. Two additional vectors will be defined to represent message between interfere nodes. $\vec{\alpha} = \{\alpha_i(s_i), 1 \le i \le n\}$ and $\vec{\beta} = \{\beta_i(s_i), 1 \le i \le n\}$. For one of inference node, inference propagation can be depicted by figure 4.7.

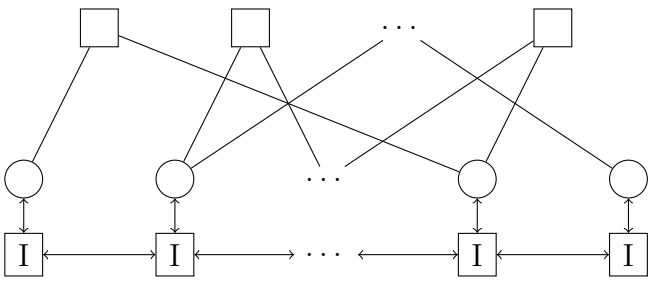

Figure 4.6: Message passing diagram of LDPC Decoding over finite-state channels

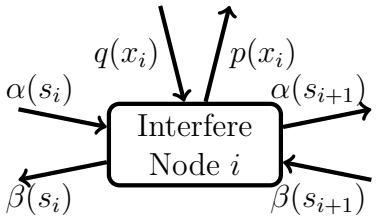

Figure 4.7: Interfere Propagation

The message related to this inference node is calculated by equations below.

$$\begin{aligned}
\alpha_{i+1}(s_{i+1}) &= \sum_{s_i}\sum_{x_i} Pr\{x_i s_{i+1}|y_i s_i\} \\
&\quad \times \alpha_i(s_i) q_i(x_i) \qquad (4.36) \\
\beta_i(s_i) &= \sum_{s_{i+1}}\sum_{x_i} Pr\{x_i s_i|Y_1^i, s_{i+1}\} \\
&\quad \times \beta_{i+1}(s_{i+1}) q_i(x_i) \qquad (4.37) \\
p_i(x_i) &= \kappa_i \sum_{s_i}\sum_{s_{i+1}} Pr\{x_i s_{i+1}|y_i s_i\} \\
&\quad \times \alpha_i(s_i)\beta_{i+1}(s_{i+1}) \qquad (4.38)
\end{aligned}$$

where κ_i is a normalising parameter, which guarantees $p_i(x_i)$ is a probability distribution. This is similar to forward-backward algorithm. Actually for backward part, if (4.34) is substituted into (4.37), we have

$$\begin{aligned}
\frac{\beta_i(s_i)}{Pr\{s_i|Y_1^i\}} &= \sum_{s_{i+1}}\sum_{x_i} Pr\{x_i s_{i+1}|y_i s_i\} \\
&\quad \frac{\beta_{i+1}(s_{i+1})}{Pr\{s_{i+1}|Y_1^i\}} q_i(x_i) \qquad (4.39)
\end{aligned}$$

and if we define $\overline{\beta_i(s_i)} = \frac{\beta_i(s_i)}{Pr\{s_i|Y_1^i\}}$, then (4.37) and (4.38) can be written as

$$\begin{aligned}
\overline{\beta_i(s_i)} &= \sum_{s_{i+1}}\sum_{x_i} Pr\{x_i s_{i+1}|y_i s_i\} \\
&\quad \times \overline{\beta_{i+1}(s_{i+1})} q_i(x_i) \qquad (4.40) \\
p_i(x_i) &= \kappa_i \sum_{s_i}\sum_{s_{i+1}} Pr\{x_i s_{i+1}|y_i s_i\} \\
&\quad \times \alpha_i(s_i) Pr\{s_{i+1}|Y_1^i\} \overline{\beta_{i+1}(s_{i+1})} \qquad (4.41)
\end{aligned}$$

In the following we will investigate several special cases of finite state channel, and existing BP decoding algorithm for those cases. It will show that those existing algorithm will degrade to the algorithm proposed here in those special cases.

4.2.3 Memoryless Channel

If channel is memoryless, $Pr\{x_i s_{i+1}|y_i s_i\} \propto Pr\{x_i|y_i\}$ with the same factor no matter s_{i+1} and s_i are. Then according to (4.41), $p_i(x_i) \propto Pr\{x_i|y_i\}$, and if κ_i is chosen properly, $p_i(x_i) = Pr\{x_i|y_i\}$, and this algorithm reduces to sum product algorithm for memoryless channel.

4.2.4 GE channel

For GE channel, channel state is independent with input, therefore $Pr\{x_i s_{i+1}|y_i s_i\} = Pr\{x_i|y_i s_i\}Pr\{s_{i+1}|s_i\}$. According to (4.36) (4.40) and (4.41), we have

$$\begin{aligned}
\alpha_{i+1}(s_{i+1}) &= \sum_{s_i} Pr\{s_{i+1}|s_i\}\alpha_i(s_i) \\
&\quad \times \sum_{x_i} q_i(x_i) Pr\{x_i|y_i s_i\} \qquad (4.42) \\
\overline{\beta_i(s_i)} &= \sum_{x_i} q_i(x_i) Pr\{x_i|y_i s_i\} \\
&\quad \times \sum_{s_{i+1}} Pr\{s_{i+1}|s_i\}\overline{\beta_{i+1}(s_{i+1})} \qquad (4.43) \\
p_i(x_i) &= \sum_{s_i}\sum_{s_{i+1}} Pr\{x_i|y_i s_i\} Pr\{s_{i+1}|s_i\} \\
&\quad \times \alpha_i(s_i) Pr(s_{i+1})\overline{\beta_{i+1}(s_{i+1})} \qquad (4.44)
\end{aligned}$$

Here (4.42) and (4.43) are exactly same to LDPC decoding algorithm for GE channel[], and (4.44) is different only by the item $Pr\{s_{i+1}\}$. However, when channel enters into stationary state, $Pr\{s_i\}$ will be a constant for any i. Therefore this algorithm reduces into normalised version of LDPC algorithm for GE channel.

4.2.5 Output Markov Channel

Now we consider the case when state is determined by previous k output, that is, $s_i = X_{i-k}^{i-1}$. Then from (4.36) (4.37) and (4.38), we will get

$$\begin{aligned}
\alpha_{i+1}(X_{i+1-k}^{i}) &= \sum_{x_{i-k}} Pr\{x_i|y_i, X_{i-k}^{i-1}\} \\
&\quad \times \alpha_i(X_{i-k}^{i-1}) q_i(x_i) \qquad (4.45) \\
\beta_i(X_{i-k}^{i-1}) &= \sum_{x_i} Pr\{x_{i-k}|Y_1^i, X_{i-k+1}^{i}\} \\
&\quad \times \beta_{i+1}(X_{i+1-k}^{i}) q_i(x_i) \qquad (4.46) \\
p_i(x_i) &= \sum_{X_{i-k}^{i-1}} Pr\{x_i|y_i, X_{i-k}^{i-1}\} \\
&\quad \times \alpha_i(X_{i-k}^{i-1}) \beta_{i+1}(X_{i+1-k}^{i}) \qquad (4.47)
\end{aligned}$$

where

$$Pr\{x_{i-k}|Y_1^i, X_{i-k+1}^{i}\} = \frac{Pr\{X_{i-k}^{i-1}|Y_1^{i-1}\}}{Pr\{X_{i-k+1}^{i}|Y_1^{i}\}} Pr\{x_i|y_i, X_{i-k}^{i-1}\} \qquad (4.48)$$

4.2.6 Complexity of the Algorithm

As we know that inference propagation within LDPC tanner graph has the complexity of $\mathcal{O}(n)$, and n is block length. And we can see from equations of forward-backward propagation, the propagation itself also has the complexity of $\mathcal{O}(n)$. Now the issue is how to calculate $Pr\{s_{i+1}|Y_1^i\}$. Note that

$$Pr\{s_{i+1}|Y_1^i\} = \sum_{s_i} Pr\{s_i|Y_1^{i-1}\} Pr\{s_{i+1}|y_i, s_i\} \qquad (4.49)$$

we can calculate those probabilities sequentially, and complexity is again $\mathcal{O}(n)$.

Remark 4. *Note that the derivation of algorithm does not involve the statistical property of Y. Even for non-stationary non-ergodic Y, this algorithm also works.*

4.3 Experiment Results

4.3.1 Complexity of Interactive Coding - Incremental Message Passing Algorithm and Block-wise Adaptive Encoding

Theoretically, in each interaction, message passing algorithm will output a block of symbols, which satisfies the check node constraint. Then by comparing its likelihood

with threshold, decision whether more bits are necessary for decoding is made. For LDPC code, usually when the bits are not enough to estimate source from side information, decoding simply fails, which means that within maximum number of iterations allowed message passing algorithm fails to output such block of symbols satisfying the check node constraint. In such case, the decoding complexity will increase linearly with number of interactions. However, number of interactions also affects the whole performance of coding, and should be reasonably large. Therefore, in each interaction instead of waiting for decoding failure, we design the technique of detecting the failure. Essentially, both the failure and success are detected during iterative decoding in each interaction. For failure detection, in each iteration of message passing algorithm, number of unsatisfied check nodes is counted, and if this number is unchanged for several round of iterations, this is the sign of decoding failure, then decoder claims the failure, and send back bit 0 to ask more bits from encoder. For detecting success, decoder is looking for high confidential bits (variable with high Log-Likelihood Ratio) during the initial several iteration, and if those bits are present, decoding will continue, otherwise, decoding is terminated and decoder asks more bits from encoders. And in each interaction, the iteration of decoding does not start from the very beginning. Instead, decoder will utilize the message vector from the last interaction. By doing this, the complexity will be approximately the same as the case without interaction. Experiment shows that the average total iteration of one block over all interaction turns out to be about 100 for small conditional entropy region(0.25 to 0.4), and 300 for high conditional entropy region(0.75-0.9).

The model of interactive coding is the master-slave type. In practical system, this is not necessary. Encoder can also adapt the property of source, block-wisely. Essentially, initial rate the encoder transmits in the first interaction can be any number as long as it is above the rate of base code. From previous blocks, encoder can estimate the range of number of interaction required, and set the initial rate at the bottom of that range. In this way, the number of both interactions and total iterations will be reduced.

4.3.2 Symmetrical Memoryless Model

Experiment settings are as follows: Block Length = 8000. The three different base LDPC codes are used here, whose variable node degrees distributions target to channel rate 0.25, 0.5, and 0.7 respectively. Therefore, the corresponding SW rates are 0.75, 0.5 and 0.3. The variable node degrees distributions are obtained in [1], and SW rates of base codes are 0.375, 0.25 and 0.15. And Codes are generated by Progressive Edge-Growth Method [20]. Therefore there is no need to specify the

check node degrees distributions. The variable node degrees used are:

$$\begin{aligned}\lambda_{0.15}(x) &= 0.132994x + 0.125532x^2 + 0.163834x^4 \\ &\quad + 0.148921x^5 + 0.428719x^{19} \qquad (4.50)\\ \lambda_{0.25}(x) &= 0.23802x + 0.20997x^2 + 0.03492x^3 + 0.12015x^4 \\ &\quad + 0.01587x^6 + 0.0048x^{13} + 0.376269x^{14} \qquad (4.51)\\ \lambda_{0.375}(x) &= 0.351551x + 0.221134x^2 + 0.215355x^5 + 0.00260617x^6 \\ &\quad + 0.0439016x^{12} + 0.127313x^{14} + 0.0381391x^{15} \qquad (4.52)\end{aligned}$$

where the subscript indicates the SW rate of base codes.

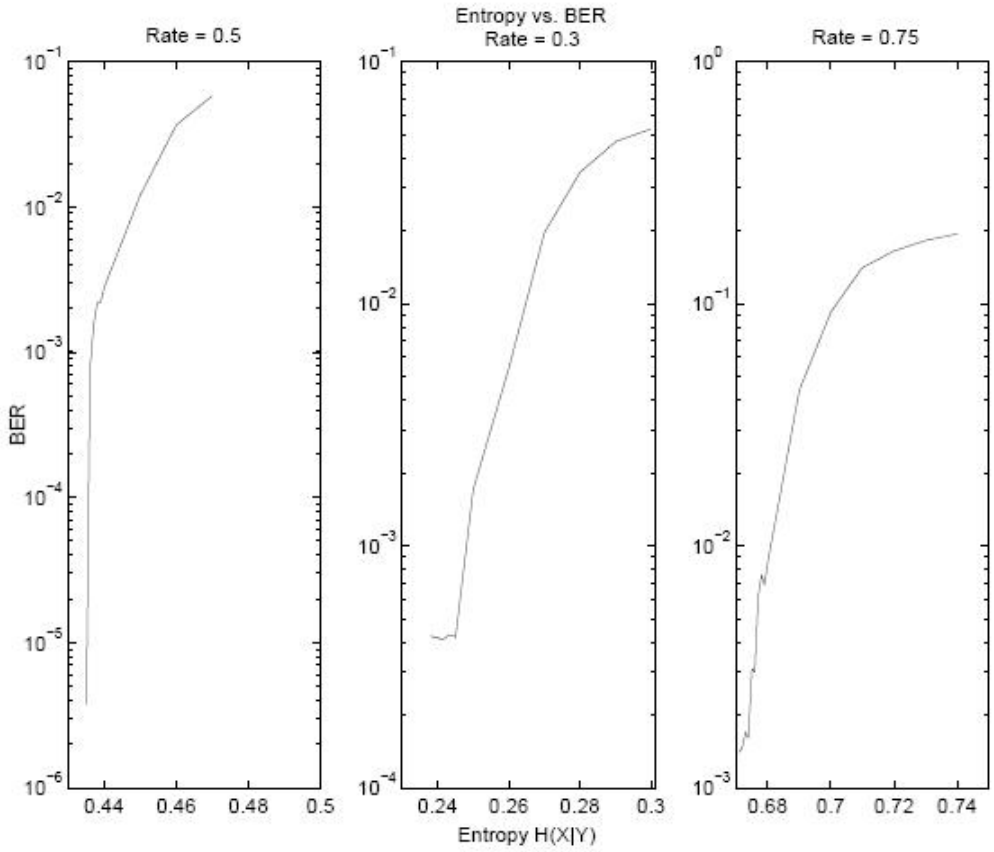

Figure 4.8: Performance of LDPC code, Length 8000

Figure 4.8 shows the performance of those three codes. Note that for channel rate = 0.3(SW rate = 0.7), when gap between theoretical limit and compression rate is less than 0.06, errors blow up, larger than 10^{-4}. Same thing happens for the code of channel rate = 0.5(SW rate=0.5), when gap is less than 0.06, and for channel rate=0.75(SW rate=0.25), when gap is less than 0.07.

Now let us look at our coding scheme with feedback. Note that if there is feedback, decoder can send information to encoder to indicate which code is suitable

for the current source. Therefore, we have flexibility to choose LDPC for different rate region. The performance is shown in Figure 4.9.

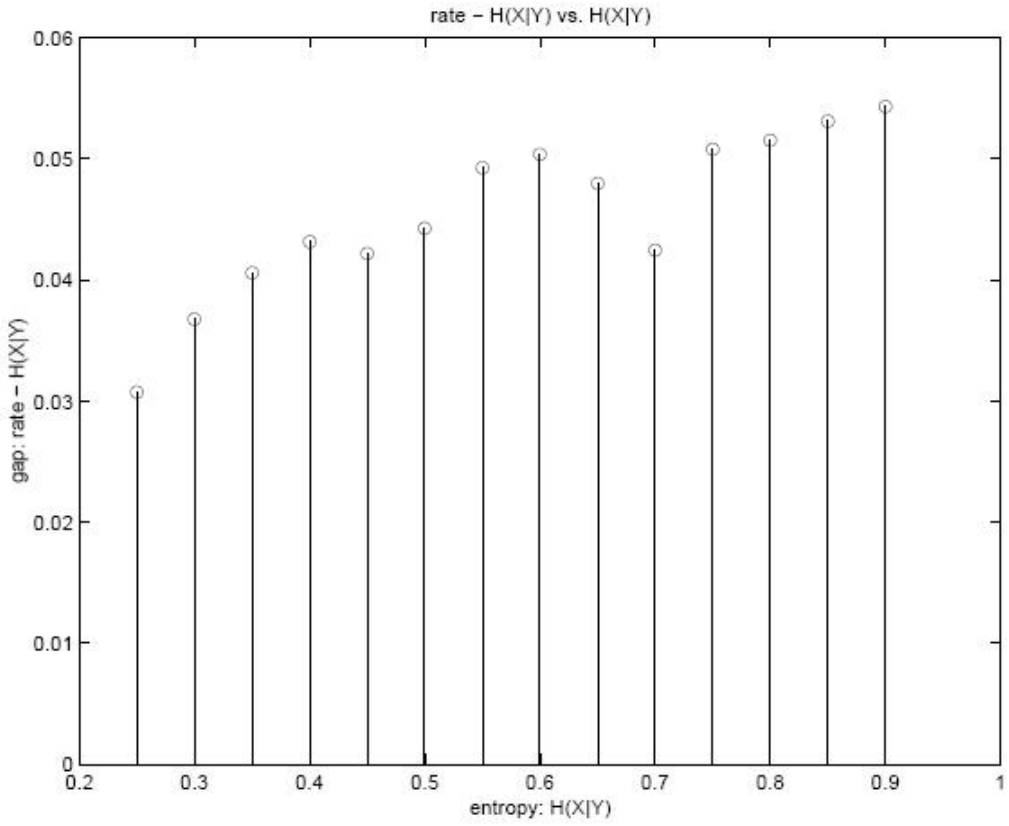

Figure 4.9: Performance Interactive Coding Scheme, Length 8000

Here we do not show error probability. Actually except for SW rate=0.25, which has errors at 10^{-5} level, rest of them has no error within 10^6 bits simulated, which means they should have error probability less than 10^{-6}. For SW rate = 0.3, 0.5 and 0.75, gaps are 0.367, 0.443 and 0.510, which are much better for regular coding scheme. More importantly, this coding scheme adapts all the rate, and note that when the compression rate is calculated, both forward and backward bits are counted. Figure 4.10 is another way to show the performance.

4.3.3 Asymmetrical Memoryless Model

Here are some results when channel between Y and X are asymmetrical, but still memoryless. Base code is generated according to variable node distribution of channel rate = 0.5(SW rate=0.5), and rate of base code is 0.25. Performance is shown in Table 4.1.

We can see that when the entropy is approaching to 0.5, and channel turns more "symmetrical", the result is better. This is the property of this particular LDPC

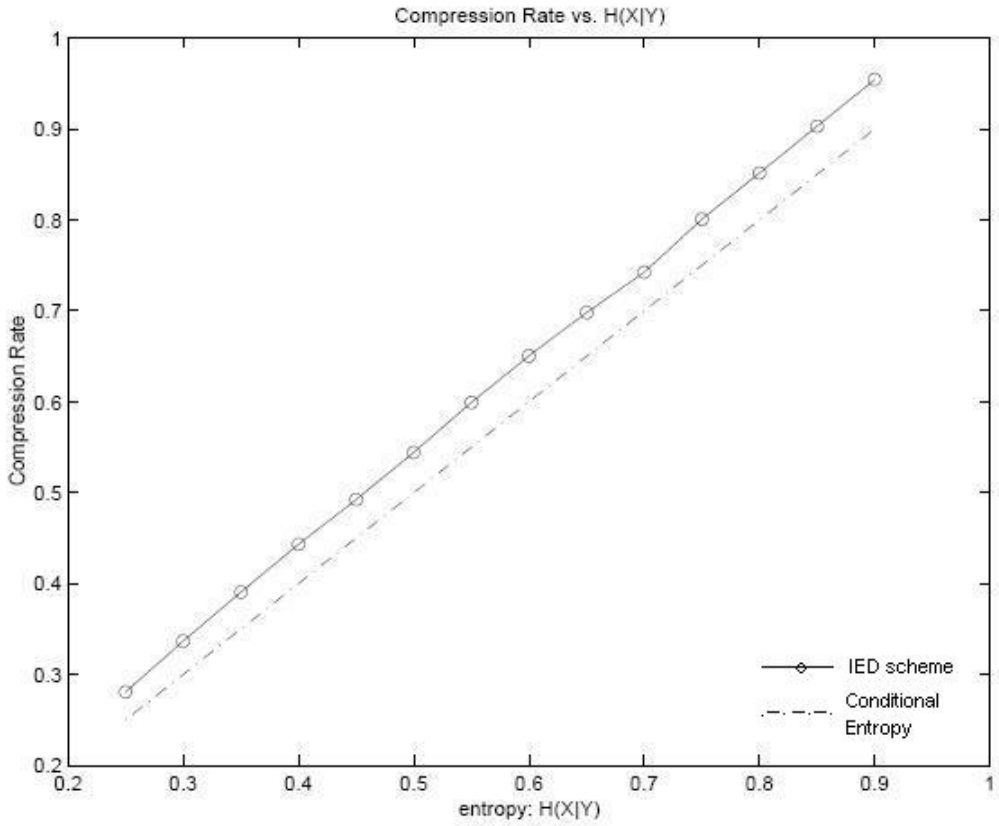

Figure 4.10: Performance showed in another way, length 8000

code. The coding scheme accommodate the "asymmetric" very well. Here 10^5 bits are simulated and there is no error, which implies the error should at least at the level of 10^{-5}.

4.3.4 First Order Output-Markov Source

Here we assume Y is i.i.d sequence. In this assumption, it is not hard to see that X is first-order Markov Source itself. Actually

$$Pr\{X_i|X_{i-1}\} = \sum_{y_i} Pr\{X_i|y_i, X_{i-1}\} \tag{4.53}$$

Therefore entropy can be calculated as

$$\begin{aligned} H(X_i|Y_i, X_{i-1}) &= \sum_{y_i, x_{i-1}} Pr(y_i)\pi(x_{i-1}) \\ &\quad \times H(X_i|Y_i = y_i, X_{i-1} = x_{i-1}) \end{aligned} \tag{4.54}$$

where π is stationary distribution of x.

$Pr\{0\|0\}$	$Pr\{0\|1\}$	Entropy	Rate	Gap
0.95	0.15	0.3834	0.4441	0.0607
0.94	0.14	0.4045	0.4604	0.0559
0.93	0.13	0.4234	0.4744	0.0510
0.92	0.12	0.4403	0.4856	0.0453
0.91	0.11	0.4555	0.4977	0.0422

Table 4.1: Asymmetrical Memoryless Channel Model

p_{00}	p_{01}	p_{10}	p_{11}	Entropy	Gap
0.95	0.05	0.85	0.15	0.3942	0.0543
0.94	0.06	0.86	0.14	0.4134	0.0481
0.93	0.07	0.87	0.13	0.4303	0.0484
0.92	0.08	0.88	0.12	0.4451	0.0440
0.91	0.09	0.89	0.11	0.4580	0.0405
0.95	0.15	0.85	0.05	0.4251	0.0462
0.94	0.14	0.86	0.06	0.4380	0.0443
0.93	0.13	0.87	0.07	0.4486	0.0441
0.92	0.12	0.88	0.08	0.4573	0.0412
0.91	0.11	0.89	0.09	0.4640	0.0387

Table 4.2: Performance of First Order Markov Source Case

The result is shown in Table 4.2, where $Pr\{Y = 0\} = 0.7$:

where $p_{st} = Pr\{X_i = 0|Y_i = s, X_{i-1} = t\}$. First five results are in the case there are symmetrical channel from Y_i to X_i, with different cross error probability according to different value of X_{i-1}. Last five results are the case when the channel is asymmetrical.

It can be seen that when the entropy is approaching to 0.5, and X_i has less dependence on X_{i-1}, gap is smaller. Here 10^5 bits are simulated and there is no error, which implies the error should at least at the level of 10^{-5}.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Distributed source coding, which is formalised conceptually in 1970s and rediscovered in 1990s due to its application such as sensor network etc. is considered in this thesis. Particularly the thesis focuses on one of the special cases where side information is only available at the decoder and the source has to be reconstructed near losslessly. This problem has been settled theoretically by Slepian and Wolf during 1970s in the classical setup, where only one way transmission, i.e. from the encoder to the decoder, is allowed. Its practical code scheme design and implementation, according to the idea transforming it to channel coding problem first proposed by Wyner, are reviewed. In contrary to the Slepian Wolf setup, a new approach, called interactive encoding and decoding, was formalised conceptually by Yang and He, while the idea can be traced back to 1980s. The difference between Slepian Wolf coding and IED is that two way transmission is allowed in latter. While several theoretical results of IED has been derived by Yang and He, the proof largely bases on random binning argument, which does not provide the answer how to design a practical code scheme. Meanwhile, this idea has been utilized, although not accurately, in the research of rate-less SW coding. This thesis is essentially in the framework of IED. The purpose is to consider the complexity of IED scheme, and build a practical code scheme. In order to achieve this purpose, different approaches are used.

From theoretical side, we restrict code scheme to the scope of linear block code, resulting in linear IED. It then shows that this restriction does not undermine the performance of IED asymptotically, in the sense that a sequence of linear IED schemes will be always found that asymptotically the performance of code scheme, evaluated by compression rate R_n and error probability, is the same as IED scheme. Spefically, it shows that linear IED scheme can approach $H(X|Y)$ for any stationary source-side information pair (X, Y), ergodic or not, with asymptotically zero error probability. With the consideration that $H(X|Y)$ is also the minimum rate achievable by any IED scheme in such setup, we reach the conclusion made above.

To further investigate the complexity of linear IED scheme, a parameter p_n^*, called the density of linear IED scheme, is introduced, which is defined as the probability that each element of parity check matrix used in IED is non-zero. p_n^* is shown to be related to the complexity of encoding and decoding in the code scheme. With another parameter P_n^e, the bit error probability concerned, an interesting result is obtained that as long as p_n^* and ϵ_n satisfy

$$\epsilon_n p_n^* = \Omega\left(\frac{\log n}{n}\right)$$

then a sequence of linear IED schemes, with density p_n^* can be found such that compression rate approaches $H(X|Y)$ asymptotically, with P_n^e upper-bounded by ϵ_n. This event shows that there is an interesting trade-off between bit error probability and density of code scheme given that $H(X|Y)$ is approached asymptotically. Moreover, it is shown that density can be as low as $\frac{\log n}{n}$, with bit error probability still going to 0, if linear IED scheme is modified in the way that it starts from non-negligible initial rate. With the results summarised above, we conclude the theoretical approach.

From practical side, LDPC codes and BP decoding are utilized. We improve one of rate-less SW coding schemes based on the syndrome splitting in the way that splitting method in each step is optimized according to density evaluation analysis. More important issue of practical implementation is to deal with the case that the correlation between source and side information can not be modelled simply, for example by memoryless channel or hidden markov state channel, which are already considered in literature. Towards this destination, we propose "Model Training" BP decoding algorithm for general cases, and design a BP decoding used in the finite state channel case accordingly. Those existing algorithms, for memoryless or hidden markov state channel, can be shown to be the special cases of our algorithm. Finally, with the simulation results that clearly demonstrate the advantages of IED scheme over SW coding scheme, we conclude practical approach.

5.2 Future Work

The results of this thesis are just starting point towards practical Interactive Encoding and Decoding schemes. Several critical problems have to be answered before an efficient implementation is found.

1. As the linear IED scheme is just one of approaches to design practical IED schemes, are there any other ways which are more efficient and easier to design?

2. The scope of the thesis is limited to the case of side information at the decoder, how to extend the approach used in this thesis to the case of two encoders and one decoder, and more general source networks?

3. The relation between the density of code scheme and the bit error probability is the achievability result, how about converse theorem? i.e. Is this relation necessary to be satisfied for all linear IED schemes?

4. Syndrome splitting method and LDPC codes are used to form a rate-less SW code, which is the incremental encoding problem in IED scheme. Is there any other way to design incremental encoding?

5. If LDPC codes and BP decoding algorithm are utilized in IED scheme, how to design BP decoding to solve the problem of more general correlation between the source and side information?

References

[1] A. Amraouli. Lthc: Ldpcopt. URL http://lthcwww.epfl.ch/research/ldpcopt.

[2] J. Chen, A. Khisti, D. M. Malioutov, and J. S. Yedidia. Distributed source coding using serially-concatenated-accumulate codes. In *Proc. Information Theory Workshop*, Oct. 2004.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, NY, 1991.

[4] T.M. Cover. A proof of the data compression theorem of slepian and wolf for ergodic sources. *IEEE Trans. Inform. Theory*, 21:226–228, Mar. 1975.

[5] I. Csiszar and J. Korner. Towards a general theory of source networks. *IEEE Trans. Inform. Theory*, 26(2):155–165, Mar. 1980.

[6] I. Csiszar and J. Korner. *Information Theory, Coding Theorems for Discrete Memoryless Systems*, chapter 3. Akademiai Kiado, 1981.

[7] S. C. Draper. Universal incremental slepian-wolf coding. In *Proc. 43rd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 2004.

[8] A. Eckford and W. Yu. Rateless slepian-wolf codes. In *Proc. Asilomar*, Pacific Grove, CA, Nov. 2005.

[9] A. D. Eckford. *Low-Density Parity-Check Codes for Gilbert-Elliott and Markov-Modulated Channels*. PhD thesis, University of Toronto, 2004.

[10] A. W. Eckford, F. R. Kschischang, and S. Pasupathy. Analysis of low-density parity-check codes for the gilbert-elliott channel. *IEEE Trans. Inform. Theory*, 51(11):3872–3889, Nov. 2005.

[11] M. Feder and N. Shulman. Source broadcasting with unknown amount of receiver side information. In *Proc. Inform. Theory Workshop*, Bangalore, India, Oct. 2002.

[12] M. Fresia and L. Vandendorpe. Distributed source coding using raptor codes. In *Proc. Global Telecom.*, Nov. 2007.

[13] R. G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963.

[14] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley and Sons, Inc., 1968.

[15] J. Garcia-Frias. Decoding of low-density parity-check codes over finite-state binary markov channels. *IEEE Trans. Commun.*, 52(11):1840–1843, Nov. 2004.

[16] J. Garcia-Frias and W. Zhong. Ldpc codes for compression of multi-terminal sources with hidden markov correlation. *IEEE Comm. Letters*, 7:115–117, Mar. 2003.

[17] R. M. Gray and L. D. Davisson. The ergodic decomposition of stationary discrete random processes. *IEEE Trans. Inform. Theory*, 20:625–636, 1974.

[18] D.-K. He, L. A. Lastras-Montano, and E.-H. Yang. A lower bound for variable rate slepian-wolf coding. In *Proc. ISIT*, 2006.

[19] D.-K. He, L. A. Lastras-Montano, and E.-H. Yang. On the relationship between redundancy and decoding error in slepian-wolf coding. In *Proc. ITW*, 2006.

[20] X. Y. Hu, E. E. Eleftheriou, and D. M. Arnold. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inform. Theory*, IT-51:386–398, Jan. 2005.

[21] J. Jiang, D. He, and A. Jagmohan. Rateless slepian-wolf coidng based on rate adaptive low-density-parity-check codes. In *Proc. ISIT*, Nice, France, June 2007.

[22] J. C. Kieffer and E.-H. Yang. Grammar based codes: A new class of universal lossless source codes. *IEEE Trans. Inform. Theory*, 46:737–754, 2000.

[23] J. H. Kim and J. Pearl. A computational model for causal and diagnostic reasoning in inference systems. *IJCAI*, pages 190–193, 1983.

[24] F. R. Kschischang, B. J. Frey, and H. A. Leoliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, IT-47:498–519, Feb. 2001.

[25] A. D. Liveris, Z. Xiong, and C. N. Georghiades. Compression of binary sources with side information at the decoder using ldpc codes. *IEEE Comm. Letters*, 6:440–442, Oct. 2002.

[26] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE. Trans. Inform. Theory*, 47(2):585–598, Feb. 2001.

[27] D. J. C. Mackay and R. M. Neal. Near shannon limit performance of low density parity check codes. *Electronics Letters*, 32:1645–1646, 1996.

[28] J. Meng, E.-H. Yang, and D.-K. He. Interactive encoding and decoding for lossless source coding with side information at the decoder: Low-density linear schemes. in preparation for submission to *IEEE Trans. on Inform. Theory*, 2008.

[29] Y. Oohama and T. S. Han. Universal coding for the slepian-wolf data compression system and the strong converse theorem. *IEEE Trans. Inform. Theory*, 40(6):1908–1919, Nov. 1994.

[30] A. Orlitsky. Worst-case interactive communication i: Two messages are almost optimal. *IEEE. Trans. Inform. Theory*, 36(5):1111–1126, Sep. 1990.

[31] A. Orlitsky. Worst-case interactive communication ii: Two messages are not optimal. *IEEE. Trans. Inform. Theory*, 37(4):995–1005, July 1991.

[32] A. Orlitsky. Average-case interactive communication. *IEEE. Trans. Inform. Theory*, 38(5):1534–1547, Sep. 1992.

[33] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes(discus): Design and construction. *IEEE Trans. Inform. Theory*, 49:626–643, Mar. 2003.

[34] T. J. Richardson and R. L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE. Trans. Inform. Theory*, 2:599–618, Feb. 2001.

[35] M. Sartipi and F. Fekri. Distributed source coding in wireless sensor networks using ldpc coding: The entire slepian-wolf rate region. In *Proc. Wireless Communications and Networking Conference*, 2005.

[36] D. Schonberg, K. Ramchandran, and S. S. Pradhan. Ldpc codes can approach the slepian-wolf bound for general binary sources. In *Proc. of fortieth Annual Allerton Conference*, Urbana-Champaign, IL, Oct. 2002.

[37] D. Schonberg, K. Ramchandran, and S. S. Pradhan. Distributed code constructions for the entire slepian-wolf rate region for arbitarily correlated sources. In *Proc. IEEE Data Compression Conference*, 2004.

[38] A. Shokrollahi and N. N. Bertrand. Fountain codes for the slepian-wolf problem. In *Proc. Allerton*, Monicello, IL, Sep. 2006.

[39] M. Sipser and D. A. Spielman. Expander codes. *IEEE Trans. Inform. Theory*, 42:1710–1722, Nov. 1996.

[40] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, IT-19:471–480, 1973.

[41] R. M. Tanner. A recursive approach to low complexity codes. *IEEE Trans. Inform. Theory*, 27:533–547, 1981.

[42] D. Varodayan, A. Aaron, and B. Girod. Rate-adaptive distributed source coding using low-density parity-check codes. In *Proc. Asilomar*, Pacific Grove, CA, July 2005.

[43] T. Wadayama. An iterative decoding algorithm of low density parity check codes for hidden markov noise channels. In *Proc. IEEE Int. Symp. Information Theory and Its Application*, Honolulu, HI, Nov. 2000.

[44] C. C. Wang, S. R. Kullkarni, and H. V. Poor. Density evolution for asymmetric memoryless channels. *IEEE Trans. Inform. Theory*, IT-51:4216–4236, Dec. 2005.

[45] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, 1996.

[46] A.D. Wyner. Recent results in the shannon theory. *IEEE Trans. Inform. Theory*, 20:2–10, Jan. 1974.

[47] E.-H. Yang and D.-K. He. Interactive encoding and decoding for one way learning: Near lossless recovery with side information at the decoder. to be submitted to *IEEE Trans. on Inform. Theory*, 2008.

[48] E.-H. Yang and D.-K. He. On interactive encoding and decoding for lossless source coding with decoder only side information. In *Proc. ISIT'08*, 2008.

[49] E.-H. Yang, A. Kaltchenko, and J. C. Kieffer. Universal lossless data compression with side information by using a conditional mpm grammar transform. *IEEE Trans. Inform. Theory*, 47:2130–2150, 2001.

[50] E.-H. Yang and J. C. Kieffer. Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform-part one: Without context models. *IEEE Trans. Inform. Theory*, 46:755–788, 2000.

[51] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE. Trans. Inform. Theory*, 23:337–343, 1977.

[52] J. Ziv and A. Lempel. Compression of individual sequences via variable rate coding. *IEEE. Trans. Inform. Theory*, 24:530–536, 1978.