JOINT EQUALIZATION AND DECODING VIA CONVEX OPTIMIZATION

A Dissertation

by

BYUNG HAK KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2012

Major Subject: Electrical Engineering

Joint Equalization and Decoding via Convex Optimization

JOINT EQUALIZATION AND DECODING VIA CONVEX OPTIMIZATION

A Dissertation

by

BYUNG HAK KIM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Henry D. Pfister
Committee Members,	Tie Liu
	Ulisses M. Braga-Neto
	Yoonsuck Choe
Head of Department,	Costas N. Georghiades

May 2012

Major Subject: Electrical Engineering

ABSTRACT

Joint Equalization and Decoding via Convex Optimization. (May 2012)

Byung Hak Kim, B.S., Korea University;

M.S., Korea University

Chair of Advisory Committee: Dr. Henry D. Pfister

The unifying theme of this dissertation is the development of new solutions for decoding and inference problems based on convex optimization methods. The first part considers the joint detection and decoding problem for low-density parity-check (LDPC) codes on finite-state channels (FSCs). Hard-disk drives (or magnetic recording systems), where the required error rate (after decoding) is too low to be verifiable by simulation, are most important applications of this research.

Recently, LDPC codes have attracted a lot of attention in the magnetic storage industry and some hard-disk drives have started using iterative decoding. Despite progress in the area of reduced-complexity detection and decoding algorithms, there has been some resistance to the deployment of turbo-equalization (TE) structures (with iterative detectors/decoders) in magnetic-recording systems because of error floors and the difficulty of accurately predicting performance at very low error rates. To address this problem for channels with memory, such as FSCs, we propose a new decoding algorithms based on a well-defined convex optimization problem. In particular, it is based on the linear-programing (LP) formulation of the joint decoding problem for LDPC codes over FSCs. It exhibits two favorable properties: provable convergence and predictable error-floors (via pseudo-codeword analysis).

Since general-purpose LP solvers are too complex to make the joint LP decoder feasible for practical purposes, we develop an efficient iterative solver for the joint LP decoder by taking advantage of its dual-domain structure. The main advantage of this approach is that it combines the predictability and superior performance of joint LP decoding with the computational complexity of TE.

The second part of this dissertation considers the matrix completion problem for the recovery of a data matrix from incomplete, or even corrupted entries of an unknown matrix. Recommender systems are good representatives of this problem, and this research is important for the design of information retrieval systems which require very high scalability. We show that our IMP algorithm reduces the well-known cold-start problem associated with collaborative filtering systems in practice.

To My Family, Friends and Colleagues

ACKNOWLEDGMENTS

"Life must be understood backwards; but... it must be lived forwards. -
Soren Kierkegaard"

I am very grateful to have had Professor Henry D. Pfister as my advisor. Dr. Pfister has truly been a friend and teacher for me during my stay at A&M. Working with Dr. Pfister has been a pleasure and there are many things that I have learned from him. First is his research philosophy: work on new problems with real potential and never be afraid to simulate and test your ideas. Second is his professional ethic, shown during our race with Flanagan's group, that advocates generous credit and acknowledgment to the work of others. Third is his sense of professionalism, based on an analogy between amateur and professional athletes, that emphasizes the importance of giving high-quality talks. I also wish to thank the other members of my dissertation committee. Professor Tie Liu provided me a firm foundation in his courses on information theory. Professor Ulisses M. Braga-Neto was always willing to provide a little more help along the way. Professor Yoonsuck Choe was very helpful as well.

There are many other colleagues and professors who helped me immeasurably during the past five years. Dr. Pascal Vontobel gave me invaluable feedback on an early draft of Chapter III and his pseducodewords website has been a truly valuable resource. Professor Tadashi Wadayama shared his knowledge and optimism about LP decoding. Sewoong Oh provided me simulation software and pointed out key references while I was completing the comparison study in Chapter V. Professor Paul Siegel, during his vist to A&M, gave me excellent career advice on how to land a job in the storage field. There were many other faculty members at A&M that helped shape my research directions. Particular thanks go to Professor Scott Miller, Professor Krishna Narayanan, Professor Jean-Francois Chamberland, Professor Byung-Jun Yoon, Professor Yong-Kyu Jung, Professor Jianhua Huang, and Professor Robert Cui.

I also wish to thank many friends who made these past five years at A&M as a pleasurable experience. My officemates and friends: Andrew Young, Arvind Yedla, Chiawen Wang, Fan Zhang, Fatemeh Hamidi Sepehr, Phong Sy Nguyen, and Yung-Yih Jian, who have filled the office with life and humor. I thank them for the many discussions and for sharing with me the experiences they brought with them from different parts of the world. I would also like to acknowledge some of my other friends who helped keep me sane these past five years: Daehyun and Jaewon for studying together from the beginning. Jongmin, Gieseung, Soonmi, Sunyeob and other church friends for making College Station like a home city to me. I give special thanks to my wonderful parents for their love, trust and support through all of this.

Finally, I would like to thank to funding agencies who supported, in part, my research: the National Science Foundation (under Grant No. 0747470) and the Texas Norman Hackerman Advanced Research Program (under Grant No. 000512-0168-2007).

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Gallager's introduction of iterative message-passing decoding for error-correcting codes, in his 1960 Ph.D. thesis, was an idea ahead of its time [1]. In the past twenty years, however, message-passing inference has become very popular in research and practice. The *sum-product algorithm* (also known as *belief propagation* (BP)) is an iterative message-passing algorithm that computes the marginal distribution of each variable in a cycle-free graphical model. It is based on an exact inference method for trees, which involves passing messages along the edges of the tree. Each node fuses messages, from all but one of its neighbors, and then propagates this information to the excluded neighbor based on the edge potential linking the two nodes. In loopy graphs, this procedure does not always converge to a fixed-point and may give inaccurate marginals when it does converge. Its main benefit is that, for the large problem sizes, the complexity grows linearly with the problem size.

A convex-programming relaxation (CPR) is an approximate method to solve intractable inference and estimation problems for more general graphs. A CPR is formulated with respect to an augmented graphical model that includes replicas of the nodes and edges of the original graph. By dualizing the constraint on the replicated variables, one obtains a relaxed, convex dual problem, which is tractable to solve. This involves using the Lagrangian decomposition technique to break up an intractable graph into tractable subgraphs, such as small blocks of nodes. Then, a distributed iterative algorithm can maximize the Lagrangian dual function via block coordinate ascent algorithm. In particular, when strong duality holds, one can re-

-This dissertation follows the style of *IEEE Trans. on Information Theory*.

cover the optimal MAP estimate. The advantage of dual methods is that they provide efficient solution methods based on BP-like distributed message-passing algorithms. Because BP does not always converge, there is growing interest in convergent iterative methods to solve these dual formulations using coordinate descent methods.

The unifying theme of this dissertation is the development of approximate solutions for new decoding and inference problems based on these two popular techniques. The first part considers the joint detection and decoding problem for low-density parity-check (LDPC) codes on finite-state channels (FSCs). Hard disk drives (or magnetic recording systems) where the required decoding error rate is too low to be verifiable by simulation are most critical applications of this research. The second part considers the matrix comletion prolem for the recovery of a data matrix from incomplete, or even corrupted information. Recommender systems are good representatives of this problem, and this research is important for the design of information retrieval systems which require very high scalability.

A. Motivation and Overview

1. Joint Detection and Decoding Problem

Iterative decoding of error-correcting codes, while introduced by Gallager in his 1960 Ph.D. thesis, was largely forgotten until the 1993 discovery of turbo codes by Berrou et al. Since then, message-passing iterative decoding has been a very popular decoding algorithm in research and practice. In 1995, the turbo decoding of a finite-state channel (FSC) and a convolutional code (instead of two convolutional codes) was introduced by Douillard et al. as *turbo equalization* (TE) and this enabled the joint-decoding of the channel and the code by iterating between these two decoders [2]. Before this, one typically separated channel decoding (i.e., estimating the channel inputs from the channel outputs) from the decoding of the error-correcting code (i.e., estimating the transmitted codeword from estimates of the channel inputs) [1, 3]. This breakthrough received immediate interest from the magnetic recording community, and TE was applied to magnetic recording channels by a variety of authors (e.g., [4, 5, 6, 7]). TE was later combined with turbo codes and also extended to low-density parity-check (LDPC) codes (and called *joint iterative decoding*) by constructing one large graph representing the constraints of both the channel and the code (e.g., [8, 9]).

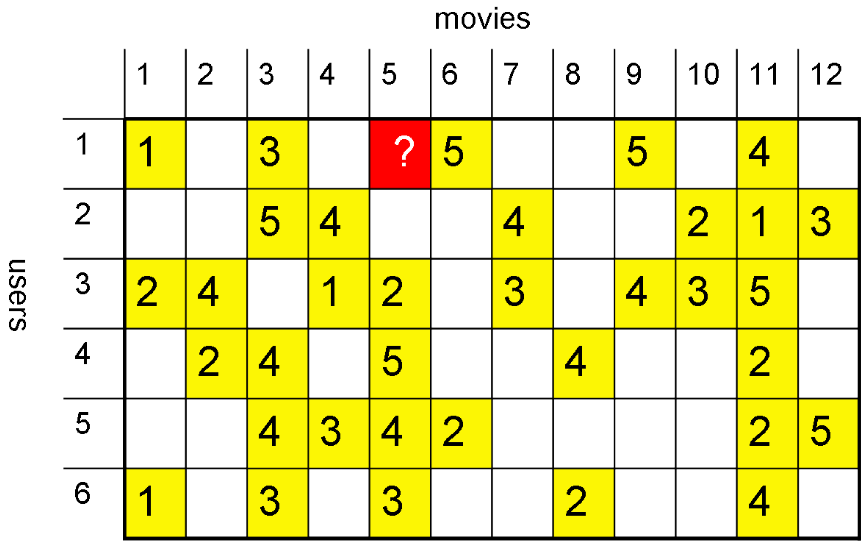
In the magnetic storage industry, error correction based on Reed-Solomon codes with hard-decision decoding has prevailed for the last 25 years. Recently, LDPC codes have attracted a lot of attention and some hard-disk drives (HDDs) have started using iterative decoding (e.g., [10, 11, 12]). Despite progress in the area of reduced-complexity detection and decoding algorithms, there has been some resistance to the deployment of TE structures (with iterative detectors/decoders) in magnetic recording systems because of error floors and the difficulty of accurately predicting performance at very low error rates. Furthermore, some of the spectacular gains of iterative coding schemes have been observed only in simulations with block-error rates above 10^{-6}. The challenge of predicting the onset of error floors and the performance at very low error rates, such as those that constitute the operating point of HDDs (the current requirement of an overall block error rate of 10^{-12}), remains an open problem. The presence of error floors and the lack of analytical tools to predict performance at very low error rates are current impediments to the application of iterative coding schemes in magnetic recording systems.

In the last five years, linear programming (LP) decoding has been a popular topic in coding theory and has given new insight into the analysis of iterative decoding algorithms and their modes of failure [13, 14, 15]. In particular, it has been observed that LP decoding sometimes performs better than iterative (e.g., sum-product) decoding in the error-floor region. We believe this stems from the fact that the LP decoder always converges to a well-defined LP optimum point and either detects decoding failure or outputs an ML codeword. For both decoders, fractional vectors, known as pseudo-codewords (PCWs), play an important role in the performance characterization of these decoders [14, 16]. This is in contrast to classical coding theory where the performance of most decoding algorithms (e.g., maximum-likelihood (ML) decoding) is completely characterized by the set of codewords.

While TE-based joint iterative decoding provides good performance close to capacity, it typically has some trouble reaching the low error rates required by magnetic recording and optical communication. To combat this, we extend LP decoding to perform joint-decoding of a binary-input FSC and an outer LDPC code. During the review process of our conference paper on this topic [17], we discovered that this LP formulation is mathematically equivalent to Flanagan's general formulation of linear-programming receivers [18, 19]. Since our main focus was different than Flanagan's, our main results and extensions differ somewhat from his. This extension had been considered as a challenging open problem in the prior works [20, 13] and the problem is well posed by Feldman in his Ph.D. thesis [13, page 146],

> *In practice, channels are generally not memoryless due to physical effects in the communication channel. ... Even coming up with a proper linear cost function for an LP to use in these channels is an interesting question. The notions of pseudocodeword and fractional distance would also need to be reconsidered for this setting.*

Other than providing satisfying answer to the above open question, our main motivation is that critical storage applications (e.g., HDDs) require block error rates that are too low to be easily verifiable by simulation. For these applications, an efficient



movies

users

	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

Fig. 1. Netflix challenge: Given a collection of ratings (yellow or light grey) between 1 to 5 that users gave to movies, predict the rating (red or dark grey) the given user would give to the given movie.

iterative solver for the joint-decoding LP would have favorable properties: error floors predictable by pseudo-codeword analysis and convergence based on a well-defined optimization problem. Therefore, we introduce a novel iterative solver for the joint LP decoding problem whose per-iteration complexity (e.g., memory and time) is similar to that of TE but whose performance appears to be superior at high SNR [17, 21].

2. Matrix Completion Problem

An important new inference problem, called the *matrix completion* problem, has recently come to light; it combines many elements of compressed sensing and collaborative filtering. This problem involves the recovery of a data matrix from incomplete (or corrupted) information and is of great practical interest over a wide range of fields [22]. The basic idea is summarized well in the following quote by Candes and Plan in [23],

> *In its simplest form, the problem is to recover a matrix from a small sample*

> *of its entries, and comes up in many areas of science and engineering including collaborative filtering, machine learning, control, remote sensing, and computer vision... Imagine now that we only observe a few entries of a data matrix. Then is it possible to accurately—or even exactly—guess the entries that we have not seen?*

In the Netflix challenge, for example, one is given a subset of large data matrix in which rows are users and columns are movies (e.g., see the Netflix Prize [24] and Fig. 1). An overwhelming portion of the user-movie matrix (e.g., 99%) is unknown and the observation matrix is very sparse because most users rate only a few movies. Randomness in the ratings process implies that one can also interpret the ratings as noisy observations of some true matrix.

The goal is to predict the rating that a user would give, to a movie he/she has not watched, based on the observed ratings. In other words, the problem is to recover missing ratings of a data matrix using the subset of observed movie ratings. In general, it would seem that this problem is difficult, if not impossible. However, if the unknown matrix has some structure, then approximate recovery is possible. Recent progress on the matrix completion problem can be largely divided into two areas:

1. The first area considers efficient models and practical algorithms. For the matrix completion problem, many researchers use models based on the assumption that the matrix has low rank. This assumption allows one to reformulate the problem into rank (or nuclear norm) minimization problem under certain incoherence assumptions [22]. For exact and approximate matrix completion, these models lead to convex relaxations, and semi-definite programming (SDP) [25, 26, 27, 28], and Bayesian-based approaches [29].

2. The second area involves exploration of the fundamental limits of these methods. Prior work has developed some precise relationships between sparse observation models and the recovery of missing entries under the restriction of low-rank matrices or clustering models [23, 30, 31, 32, 33]. This area is closely related with the practical issues known as the cold-start problem of the recommender system [34]. That is, giving recommendations to new users who have submitted only a few ratings, or recommending new items that received only a few ratings from users. In other words, how many ratings are needed to generate good recommendations?

B. Outline of Dissertation

The dissertation consists of an introduction, four self-contained chapters and conclusion.

In Chapters II, III and IV, we consider the joint-decoding problem for finite-state channels (FSCs) and low-density parity-check (LDPC) codes. In Chapter II, we introduce the joint linear-programming (LP) decoder by extending the LP decoder for binary linear codes, introduced by Feldman et al to perform joint-decoding of binary-input FSCs. In particular, we provide a rigorous definition of LP joint-decoding pseudo-codewords (JD-PCWs) that enables evaluation of the pairwise error probability between codewords and JD-PCWs in AWGN. This leads naturally to a provable upper bound on decoder failure probability. If the channel is a finite-state intersymbol interference channel, then the joint LP decoder also has the maximum-likelihood (ML) certificate property and all integer-valued solutions are codewords. In this case, the performance loss relative to ML decoding can be explained completely by fractional-valued JD-PCWs.

Chapters III and IV are devoted to developing efficient iterative solvers for the joint LP decoder introduced in Chapter II. In Chapter III, we extend the approach of iterative approximate LP decoding, proposed by Vontobel and Koetter and analyzed by Burshtein, to this problem. By taking advantage of the dual-domain structure of the joint-decoding LP, we obtain a convergent iterative algorithm for joint LP decoding whose structure is similar to BCJR-based turbo equalization (TE). The result is a joint iterative decoder whose per-iteration complexity is similar to that of TE but whose performance is similar to that of joint LP decoding. In Chapter IV, we propose a simplified joint iterative solver LP decoder whose structure is similar to SOVA-based turbo equalization (TE) with no smoothing parameters to tune. The main advantage of these decoders are that it appears to provide the predictability and superior performance of joint LP decoding with the computational complexity of TE. One expected application is coding for magnetic storage where the required block-error rate is extremely low and system performance is difficult to verify by simulation.

In Chapter V, a new message-passing (MP) method is considered for the matrix completion problem associated with recommender systems. We attack the problem using a (generative) factor graph model that is related to a probabilistic low-rank matrix factorization. Based on the model, we propose a new inference algorithm, termed IMP, for the recovery of a data matrix from incomplete observations. The algorithm is based on a clustering followed by inference via MP (IMP). The algorithm is compared with a number of other matrix completion algorithms on real collaborative filtering (e.g., Netflix) data matrices. Our results show that, while many methods perform similarly with a large number of revealed entries, the IMP algorithm outperforms all others when the fraction of observed entries is small. This is helpful because it reduces the well-known cold-start problem associated with collaborative filtering

(CF) systems in practice.

CHAPTER II

JOINT DECODING OF LDPC CODES AND FINITE-STATE CHANNELS VIA LINEAR-PROGRAMMING*

Feldman et al. introduced the LP decoder for binary linear codes in [13, 14]. It is is based on an LP relaxation of an integer program that is equivalent to ML decoding. Later, this method was extended to codes over larger alphabets [35] and to the simplified decoding of intersymbol interference (ISI) [36]. In particular, this chapter* describes an extension of the LP decoder to the joint-decoding of binary-input FSCs and defines LP joint-decoding pseudo-codewords (JD-PCWs) [17]. This extension is natural because Feldman's LP formulation of a trellis decoder is general enough to allow optimal (Viterbi style) decoding of FSCs, and the constraints associated with the outer LDPC code can be included in the same LP. This type of extension has been considered as a challenging open problem in prior works [13, 20] and was first given by Flanagan [18, 19], but was discovered independently by us and reported in [17]. In particular, Flanagan showed that any communication system which admits a sum-product (SP) receiver also admits a corresponding linear-programming (LP) receiver. Since Flanagan's approach is more general, it is also somewhat more complicated. Still, the resulting LPs are mathematically equivalent though. One benefit of restricting our attention to FSCs is that our description of the LP is based on finding a path through a trellis, which is somewhat more natural for the joint-decoding

*This chapter is in part a reprint of the material in the papers: B.-H.Kim and H. D. Pfister, "On the joint decoding of LDPC codes and finite-state channels via linear programming", in *Proc. IEEE Int. Symp. Inform. Theory*, Austin, TX, June 2010, pp. 754-758 and B.-H. Kim and H. D. Pfister, "Joint decoding of LDPC codes and finite-state channels via linear-programming", in *IEEE J. Select. Topics in Signal Processing*, pp. 1563-1576, Dec. 2011.

problem.

These LP decoders provide a natural definition of PCWs for joint-decoding, and they allow new insight into the joint-decoding problem. Joint-decoding pseudo-codewords (JD-PCWs) are defined and the decoder error-rate is upper bounded by a union bound sum over JD-PCWs. This leads naturally to a provable upper bound (e.g., a union bound) on the probability of LP decoding failure as a sum over all codewords and JD-PCWs. Moreover, we can show that all integer solutions are indeed codewords and that this joint LP decoder also has an ML certificate property. Therefore, all decoder failures can be explained by (fractional) JD-PCWs. It is worth noting that this property is not guaranteed by other convex relaxations of the same problem (e.g., see Wadayama's approach based on quadratic programming [20]).

Our primary motivation is the prediction of the error rate for joint-decoding at high SNR. The basic idea is to run simulations at low SNR and keep track of all observed codeword and pseudo-codeword errors. An estimate of the error rate at high SNR is computed using a truncated union bound formed by summing over all observed error patterns at low SNR. Computing this bound is complicated by the fact that the loss of channel symmetry implies that the dominant PCWs may depend on the transmitted sequence. Still, this technique provides a new tool to analyze the error rate of joint decoders for FSCs and low-density parity-check (LDPC) codes. Thus, novel prediction results are given in Chapter III.

A. Notation

Throughout the paper we borrow notation from [14]. Let $\mathcal{I} = \{1, \ldots, N\}$ and $\mathcal{J} = \{1, \ldots, M\}$ be sets of indices for the variable and parity-check nodes of a binary linear code. A variable node $i \in \mathcal{I}$ is connected to the set $\mathcal{N}(i)$ of neighboring parity-check nodes. Abusing notation, we also let $\mathcal{N}(j)$ be the neighboring variable nodes of a parity-check node $j \in \mathcal{J}$ when it is clear from the context. For the trellis associated with a FSC, we let $E = \{1, \ldots, O\}$ index the set of trellis edges associated with one trellis section, $\mathcal{S}$ be the set of possible states, and $\mathcal{A}$ be the set of noiseless output symbols. For each edge[1], $e \in E^N$, in the length-N trellis, the functions $t : E^N \to \{1, \ldots, N\}$, $s : E^N \to \mathcal{S}$, $s' : E^N \to \mathcal{S}$, $x : E^N \to \{0, 1\}$, and $a : E^N \to \mathcal{A}$ map this edge to its respective time index, initial state, final state, input bit, and noiseless output symbol. Finally, the set of edges in the trellis section associated with time i is defined to be $\mathcal{T}_i = \left\{e \in E^N \,|\, t(e) = i\right\}$.

B. Background: LP Decoding and Finite-State Channels

In [13, 14], Feldman et al. introduced a linear-programming (LP) decoder for binary linear codes, and applied it specifically to both LDPC and turbo codes. It is based on solving an LP relaxation of an integer program that is equivalent to maximum-likelihood (ML) decoding. For long codes and/or low SNR, the performance of LP decoding appears to be slightly inferior to belief-propagation decoding. Unlike the iterative decoder, however, the LP decoder either detects a failure or outputs a codeword which is guaranteed to be the ML codeword.

Let $\mathcal{C} \subseteq \{0, 1\}^N$ be the length-N binary linear code defined by a parity-check matrix and $\mathbf{c} = (c_1, \ldots, c_N)$ be a codeword. Let $\mathcal{L}$ be the set whose elements are the sets of indices involved in each parity check, or

$$\mathcal{L} = \{\mathcal{N}(j) \subseteq \{1, \ldots, N\} | \, j \in \mathcal{J}\} .$$

[1] In this dissertation, e is used to denote a trellis edge while e denotes the universal constant that satisfies $\ln \mathsf{e} = 1$.

Then, we can define the set of codewords to be

$$\mathcal{C} = \left\{ \mathbf{c} \in \{0,1\}^N \;\middle|\; \sum_{i \in L} c_i \equiv 0 \mod 2,\, \forall\, L \in \mathcal{L} \right\}.$$

The *codeword polytope* is the convex hull of $\mathcal{C}$. This polytope can be quite complicated to describe though, so instead one constructs a simpler polytope using local constraints. Each parity-check $L \in \mathcal{L}$ defines a local constraint equivalent to the extreme points of a polytope in $[0,1]^N$.

Definition 1. The *local codeword polytope* LCP(L) associated with a parity check is the convex hull of the bit sequences that satisfy the check. It is given explicitly by

$$\mathrm{LCP}(L) \triangleq \bigcap_{\substack{S \subseteq L \\ |S| \mathrm{odd}}} \left\{ \mathbf{c} \in [0,\, 1]^N \;\middle|\; \sum_{i \in S} c_i - \sum_{i \in L-S} c_i \leq |S| - 1 \right\}.$$

We use the notation $\mathcal{P}(H)$ to denote the simpler polytope corresponding to the intersection of local check constraints; the formal definition follows.

Definition 2. The *relaxed polytope* $\mathcal{P}(H)$ is the intersection of the LCPs over all checks and

$$\mathcal{P}(H) \triangleq \bigcap_{L \in \mathcal{L}} \mathrm{LCP}(L).$$

The LP decoder and its ML certificate property is characterized by the following theorem.

Theorem 1 ([13])**.** Consider N consecutive uses of a symmetric channel $\Pr(Y = y|C = c)$. If a uniform random codeword is transmitted and $\mathbf{y} = (y_1, \ldots, y_N)$ is received, then the LP decoder outputs $\mathbf{f} = (f_1, \ldots, f_N)$ given by

$$\arg\min_{\mathbf{f} \in \mathcal{P}(H)} \sum_{i=1}^{N} f_i \ln \left(\frac{\Pr(Y_i = y_i \,|\, C_i = 0)}{\Pr(Y_i = y_i \,|\, C_i = 1)} \right),$$

which is the ML solution if $\mathbf{f}$ is integral (i.e., $\mathbf{f} \in \{0,1\}^N$).

From simple LP-based arguments, one can see that LP decoder may also output nonintegral solutions.

Definition 3. An *LP decoding pseudo-codeword* (LPD-PCW) of a code defined by the parity-check matrix H is any *nonintegral* vertex of the relaxed (fundamental) polytope $\mathcal{P}(H)$.

We also define the finite-state channel, which can be seen as a model for communication systems with memory where each output depends only on the current input and the previous channel state instead of the entire past.

Definition 4. A *finite-state channel* (FSC) defines a probabilistic mapping from a sequence of inputs to a sequence of outputs. Each output $Y_i \in \mathcal{Y}$ depends only on the current input $X_i \in \mathcal{X}$ and the previous channel state $S_{i-1} \in \mathcal{S}$ instead of the entire history of inputs and channel states. Mathematically, we define $P\left(y, s'|x, s\right) \triangleq \Pr\left(Y_i{=}y, S_i{=}s'|X_i{=}x, S_{i-1}{=}s\right)$ for all i, and use the shorthand notation $P_0(s) \triangleq \Pr(S_0 = s)$ and

$$\begin{aligned} P\big(y_1^N, s_1^N|x_1^N, s_0\big) &\triangleq \Pr\left(Y_1^N{=}y_1^N, S_1^N{=}s_1^N|X_1^N{=}x_1^N, S_0{=}s_0\right) \\ &= \prod_{i=1}^{N} P\left(y_i, s_i|x_i, s_{i-1}\right), \end{aligned}$$

where the notation Y_i^j denotes the subvector $(Y_i, Y_{i+1}, \ldots, Y_j)$.

An important subclass of FSCs is the set of finite-state intersymbol interference channels which includes all deterministic finite-state mappings of the inputs corrupted by memoryless noise.

Definition 5. A *finite-state intersymbol interference channel* (FSISIC) is a FSC whose next state is a deterministic function, $\eta(x, s)$, of the current state s and input

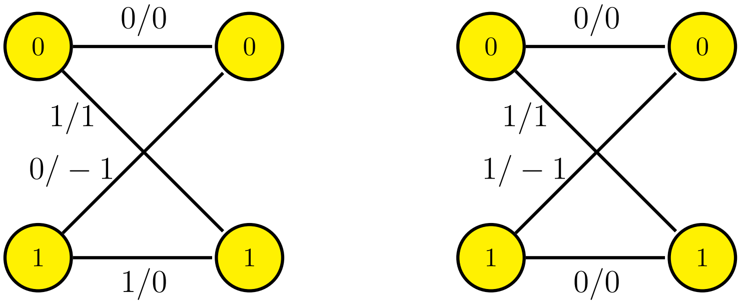


Fig. 2. State diagrams for noiseless dicode channel without (left) and with precoding (right). The edges are labeled by the input/output pair.

x. Mathematically, this implies that

$$\sum_{y\in\mathcal{Y}} P\left(y, s'|x, s\right) = \begin{cases} 1 & \text{if}\,\eta(x,s) = s' \\ 0 & \text{otherwise} \end{cases}.$$

Though our derivations are general, we use the following FSISIC examples throughout the paper to illustrate concepts and perform simulations.

Definition 6. The *dicode channel* (DIC) is a binary-input FSISIC with an impulse response of $G(z) = 1 - z^{-1}$ and additive Gaussian noise [37]. If the input bits are differentially encoded prior to transmission, then the resulting channel is called the *precoded dicode channel* (pDIC) [37]. The state diagrams of these two channels are shown in Fig. 2. For the trellis associated with a DIC and pDIC, we let $E = \{1,2,3,4\}$, $\mathcal{S} = \{0,1\}$ and $\mathcal{A} = \{-1,0,1\}$. Also, the *class-II Partial Response* (PR2) channel is a binary-input FSISIC with an impulse response of $G(z) = 1 + 2z^{-1} + z^{-2}$ and additive Gaussian noise [37, 38].

C. Joint LP Decoding Derivation

Now, we describe *the joint LP decoder* in terms of the trellis of the FSC and the checks in the binary linear code[2]. Let N be the length of the code and $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ be the received sequence. The trellis consists of $(N+1)|\mathcal{S}|$ vertices (i.e., one for each state and time) and a set of at most $2N|\mathcal{S}|^2$ edges (i.e., one edge for each input-labeled state transition and time). The LP formulation requires one indicator variable for each edge $e \in \mathcal{T}_i$, and we denote that variable by $g_{i,e}$. So, $g_{i,e}$ is equal to 1 if the candidate path goes through the edge e in $\mathcal{T}_i$. Likewise, the LP decoder requires one cost variable for each edge and we associate the branch metric $b_{i,e}$ with the edge e given by

$$b_{i,e} \triangleq \begin{cases} -\ln P\left(y_{t(e)}, s'(e)|x(e), s(e)\right) & \text{if}\, t(e)\!>\!1 \\ -\ln\left[P\left(y_{t(e)}, s'(e)|x(e), s(e)\right) P_0\left(s(e)\right)\right] & \text{if}\, t(e)\!=\!1. \end{cases}$$

First, we define the trellis polytope $\mathcal{T}$ formally below.

Definition 7. The *trellis polytope* $\mathcal{T}$ enforces the flow conservation constraints for the channel decoder. The flow constraint for state k at time i is given by

$$\mathcal{F}_{i,k} \triangleq \left\{ \mathbf{g} \in [0,1]^{N \times O} \;\middle|\; \sum_{e:s'(e)=k} g_{i,e} = \sum_{e:s(e)=k} g_{i+1,e} \right\}.$$

Using this, the *trellis polytope* $\mathcal{T}$ is given by

$$\mathcal{T} \triangleq \left\{ \mathbf{g} \in \bigcap_{i=1}^{N-1} \bigcap_{k \in \mathcal{S}} \mathcal{F}_{i,k} \;\middle|\; \sum_{e \in \mathcal{T}_p} g_{p,e} = 1, \text{ for any}\, p \in \mathcal{I} \right\}.$$

From simple flow-based arguments, it is known that the ML edge path on trellis can be found by solving a minimum-cost LP applied to the trellis polytope $\mathcal{T}$.

[2] It is straightforward to extend this joint LP decoder to non-binary linear codes based on [35].

Theorem 2 ([13, p. 94])**.** Finding the ML edge-path through a weighted trellis is equivalent to solving the minimum-cost flow LP

$$\arg\min_{\mathbf{g}\in\mathcal{T}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e}$$

and the optimum $\mathbf{g}$ must be integral (i.e., $\mathbf{g} \in \{0,1\}^{N\times O}$) unless there are ties.

The indicator variables $g_{i,e}$ are used to define the LP and the code constraints are introduced by defining an auxiliary variable f_i for each code bit.

Definition 8. Let the code-space projection $\mathcal{Q}$, be the mapping from $\mathbf{g}$ to the input vector $\mathbf{f} = (f_1, \ldots, f_N) \in [0,1]^N$ defined by $\mathbf{f} = \mathcal{Q}(\mathbf{g})$ with

$$f_i = \sum_{e\in\mathcal{T}_i:\, x(e)=1} g_{i,e}.$$

For the trellis polytope $\mathcal{T}$, $\mathcal{P}_{\mathcal{T}}(H)$ is the set of vectors whose projection lies inside the relaxed codeword polytope $\mathcal{P}(H)$.

Definition 9. The *trellis-wise relaxed polytope* $\mathcal{P}_{\mathcal{T}}(H)$ for $\mathcal{P}(H)$ is given by

$$\mathcal{P}_{\mathcal{T}}(H) \triangleq \{\mathbf{g} \in \mathcal{T} \,|\, \mathcal{Q}(\mathbf{g}) \in \mathcal{P}(H)\}.$$

The polytope $\mathcal{P}_{\mathcal{T}}(H)$ has integral vertices which are in one-to-one correspondence with the set of trelliswise codewords.

Definition 10. The *set of trellis-wise codewords* $\mathcal{C}_{\mathcal{T}}$ for $\mathcal{C}$ is defined by

$$\mathcal{C}_{\mathcal{T}} \triangleq \left\{\mathbf{g} \in \mathcal{P}_{\mathcal{T}}(H) \,\middle|\, \mathbf{g} \in \{0,1\}^{N\times O}\right\}.$$

Finally, the joint LP decoder and its ML certificate property are characterized by the following theorem.

Theorem 3. The LP joint decoder computes

$$\arg\min_{\mathbf{g}\in\mathcal{P}_{\mathcal{T}}(H)} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e} \tag{2.1}$$

and outputs a joint ML edge-path if $\mathbf{g}$ is integral.

Proof. Let $\mathcal{V}$ be the set of valid input/state sequence pairs. For a given $\mathbf{y}$, the ML edge-path decoder finds the most likely path, through the channel trellis, whose input sequence is a codeword. Mathematically, it computes

$$\begin{aligned}
&\arg\max_{(x_1^N, s_0^N)\in\mathcal{V}} P(y_1^N, s_1^N | x_1^N, s_0) P_0\left(s(e)\right) \\
&\quad = \arg\max_{\mathbf{g}\in\mathcal{C}_{\mathcal{T}}} P_0\left(s(e)\right) \prod_{i\in\mathcal{I}} \prod_{e\in\mathcal{T}_i:\, g_{i,e}=1} P\left(y_{t(e)}, s'(e) | x(e), s(e)\right) \\
&\quad = \arg\min_{\mathbf{g}\in\mathcal{C}_{\mathcal{T}}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i:\, g_{i,e}=1} b_{i,e} \\
&\quad = \arg\min_{\mathbf{g}\in\mathcal{C}_{\mathcal{T}}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e},
\end{aligned}$$

where ties are resolved in a systematic manner and $b_{1,e}$ has the extra term $-\ln\, P_0\left(s(e)\right)$ for the initial state probability. By relaxing $\mathcal{C}_{\mathcal{T}}$ into $\mathcal{P}_{\mathcal{T}}(H)$, we obtain the desired result. □

Corollary 1. For a FSISIC[3], the LP joint decoder outputs a joint ML codeword if $\mathbf{g}$ is integral.

[3] In fact, this holds more generally for the restricted class of FSCs used in [39], which are now called unifilar FSCs because they generalize the unifilar Markov sources defined in [40].

Proof. The joint ML decoder for codewords computes

$$\begin{aligned}
&\arg\max_{x_1^N\in\mathcal{C}} \sum_{s_1^N\in\mathcal{S}^N} P(y_1^N, s_1^N | x_1^N, s_0) P_0\left(s(e)\right) \\
&= \arg\max_{x_1^N\in\mathcal{C}} \sum_{s_1^N\in\mathcal{S}^N} \prod_{i\in\mathcal{I}} P(y_i, s_{i+1} | x_i, s_i) P_0\left(s(e)\right) \\
&\overset{(a)}{=} \arg\max_{x_1^N\in\mathcal{C}} \prod_{i\in\mathcal{I}} P\left(y_i, \eta\left(x_i, s_i\right) \big| x_i, s_i\right) P_0\left(s(e)\right) \\
&\overset{(b)}{=} \arg\min_{\mathbf{g}\in\mathcal{C}_{\mathcal{T}}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e},
\end{aligned}$$

where (a) follows from Definition 5 and (b) holds because each input sequence defines a unique edge-path. Therefore, the LP joint-decoder outputs an ML codeword if $\mathbf{g}$ is integral. □

Remark 1. *If the channel is not a FSISIC (e.g., if it is a finite-state fading channel), then integer valued solutions of the LP joint-decoder are ML edge-paths but not necessarily ML codewords. This occurs because the joint LP decoder does not sum the probability of the multiple edge-paths associated with the same codeword (e.g., when multiple distinct edge-paths are associated with the same input labels). Instead, it simply gives the probability of the most-likely edge path associated that codeword.*

D. Joint LP Decoding Pseudo-codewords

Pseudo-codewords have been observed and given names by a number of authors (e.g., [41, 42, 43]), but the simplest general definition was provided by Feldman et al. in the context of LP decoding of parity-check codes [14]. One nice property of the LP decoder is that it always returns either an integral codeword or a fractional pseudo-codeword. Vontobel and Koetter have shown that a very similar set of pseudo-codewords also affect message-passing decoders, and that they are essentially fractional codewords

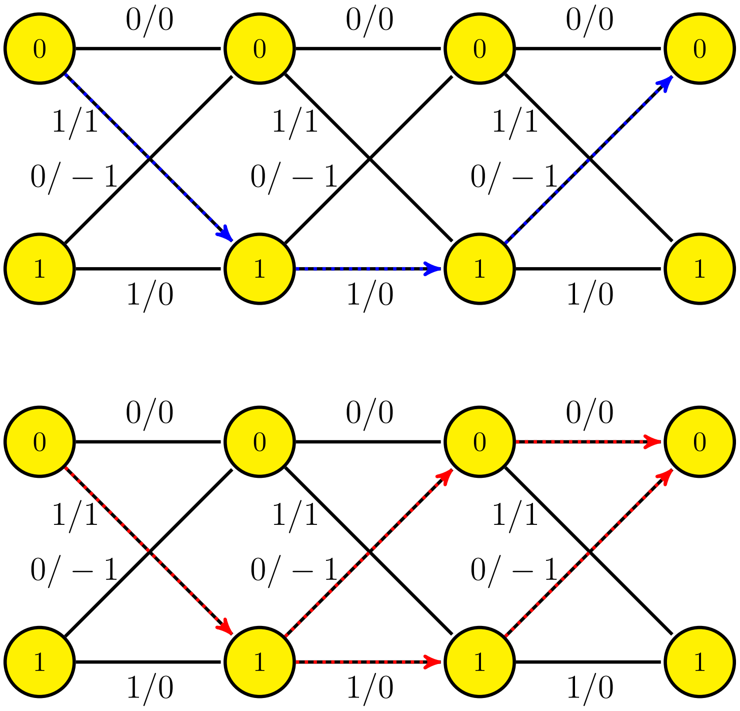


Fig. 3. Illustration of joint LP decoder outputs for the single parity-check code SPC(3,2) over DIC (starts in zero state). By ordering the trellis edges appropriately, joint LP decoder converges to either a TCW (0 1 0 0; 0 0 0 1; .0 0 1 0) (top dashed blue path) or a JD-TPCW (0 1 0 0; 0 0 .5 .5; .5 0 .5 0) (bottom dashed red paths). Using $\mathcal{Q}$ to project them into $\mathcal{P}(H)$, we obtain the corresponding SCW $(1, 1, 0)$ and JD-SPCW $(1, .5, 0)$.

that cannot be distinguished from codewords using only local constraints [16]. The joint-decoding pseudo-codeword (JD-PCW), defined below, can be used to characterize code performance at low error rates.

Definition 11. If $g_{i,e} \in \{0,\, 1\}$ for all e, then the output of the LP joint decoder is a *trellis-wise codeword* (TCW). Otherwise, $g_{i,e} \in (0,\, 1)$ for some e and the solution is called a *joint-decoding trellis-wise pseudo-codeword* (JD-TPCW); in this case, the decoder outputs "failure" (see Fig. 3 for an example of this definition).

Definition 12. For any TCW $\mathbf{g}$, the projection $\mathbf{f} = \mathcal{Q}(\mathbf{g})$ is called a *symbol-wise codeword* (SCW). Likewise, for any JD-TPCW $\mathbf{g}$, the projection $\mathbf{f} = \mathcal{Q}(\mathbf{g})$ is called a *joint-decoding symbolwise pseudo-codeword* (JD-SPCW) (see Fig. 3 for a graphical depiction of this definition).

Remark 2. *For FSISICs, the LP joint decoder has the* ML certificate *property; if the decoder outputs a SCW, then it is guaranteed to be the ML codeword (see Corollary 1).*

Definition 13. If $\mathbf{g}$ is a JD-TPCW, then $\mathbf{p} = (p_1, \ldots, p_N)$ with

$$p_i = \sum_{e \in \mathcal{T}_i} g_{i,e} a\left(e\right),$$

is called a *joint-decoding symbol-wise signal-space pseudo-codeword* (JD-SSPCW). Likewise, if $\mathbf{g}$ is a TCW, then $\mathbf{p}$ is called a *symbol-wise signal-space codeword* (SSCW).

Example 1. Consider the single parity-check code SPC(3,2). Over precoded dicode channel (starts in zero state) with AWGN, this code has five joint-decoding pseudo-codewords. A simulation was performed for joint-decoding of the SPC(3,2) on the pDIC trellis and the set of JD-TPCW, by ordering the trellis edges appropriately, was found to be

$$\begin{aligned}&\{(0\,1\,0\,0; 0\,0\,.5\,.5; 0\,.5\,.5\,0),(.5\,.5\,0\,0; .5\,0\,0\,.5; 0\,1\,0\,0),\\ &\quad(.5\,.5\,0\,0; 0\,.5\,.5\,0; 0\,0\,1\,0),(1\,0\,0\,0; .5\,.5\,0\,0; 0\,.5\,.5\,0),\\ &(.5\,.5\,0\,0; .5\,0\,0\,0; 0\,.5\,.5\,0)\}.\end{aligned}$$

Using $\mathcal{Q}$ to project them into $\mathcal{P}(H)$, we get the corresponding set of JD-SPCW

$$\{(1, .5, .5),\ (.5, .5, 1),\ (.5, .5, 0),\ (0, .5, .5),\ (.5, 0, .5)\}.$$

E. Union Bound for Joint LP Decoding

Now that we have defined the relevant pseudo-codewords, we consider how much a particular pseudo-codeword affects performance; the idea is to quantify pairwise error probabilities. In fact, we will use the insights gained in the previous section to obtain a union bound on the decoder's word-error probability and to analyze the performance of the proposed joint LP decoder. Toward this end, let's consider the pairwise error event between a SSCW $\mathbf{c}$ and a JD-SSPCW $\mathbf{p}$ first.

Theorem 4. A necessary and sufficient condition for the pairwise decoding error between a SSCW $\mathbf{c}$ and a JD-SSPCW $\mathbf{p}$ is

$$\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} b_{i,e}g_{i,e} \leq \sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} b_{i,e}\tilde{g}_{i,e}, \tag{2.2}$$

where $\mathbf{g}\in\mathcal{P}_{\mathcal{T}}(H)$ and $\tilde{\mathbf{g}}\in\mathcal{C}_{\mathcal{T}}$ are the LP variables for $\mathbf{p}$ and $\mathbf{c}$ respectively.

Proof. By definition, the joint LP decoder (2.1) prefers $\mathbf{p}$ over $\mathbf{c}$ if and only if (2.2) holds. □

For the moment, let $\mathbf{c}$ be the SSCW of FSISIC to an AWGN channel whose output sequence is $\mathbf{y}=\mathbf{c}+\mathbf{v}$, where $\mathbf{v}=(v_1,\ldots,v_N)$ is an i.i.d. Gaussian sequence with mean 0 and variance σ^2. Then, the joint LP decoder can be simplified as stated in the following theorem.

Theorem 5. Let $\mathbf{y}$ be the output of a FSISIC with zero-mean AWGN whose variance is σ^2 per output. Then, the joint LP decoder is equivalent to

$$\arg\min_{\mathbf{g}\in\mathcal{P}_{\mathcal{T}}(H)}\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i}\left(y_i-a\left(e\right)\right)^2 g_{i,e}.$$

Proof. For each edge e, the output y_i is Gaussian with mean $a\left(e\right)$ and variance σ^2, so we have $P\left(y_{t(e)},s'(e)|x(e),s(e)\right)\sim\mathcal{N}\left(a\left(e\right),\sigma^2\right)$. Therefore, the joint LP decoder computes

$$\arg\min_{\mathbf{g}\in\mathcal{P}_{\mathcal{T}}(H)}\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} b_{i,e}g_{i,e} = \arg\min_{\mathbf{g}\in\mathcal{P}_{\mathcal{T}}(H)}\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i}\left(y_i - a\left(e\right)\right)^2 g_{i,e}.$$

□

We will show that each pairwise probability has a simple closed-form expression that depends only on a generalized squared Euclidean distance $d_{gen}^2\left(\mathbf{c},\,\mathbf{p}\right)$ and the noise variance σ^2. One might notice that this result is very similar to the pairwise error probability derived in [44]. The main difference is the trellis-based approach that allows one to obtain this result for FSCs. Therefore, the next definition and theorem can be seen as a generalization of [44].

Definition 14. Let $\mathbf{c}$ be a SSCW and $\mathbf{p}$ a JD-SSPCW. Then the *generalized squared Euclidean distance* between $\mathbf{c}$ and $\mathbf{p}$ can be defined in terms of their trellis-wise descriptions by

$$d_{gen}^2\left(\mathbf{c},\,\mathbf{p}\right) \triangleq \frac{\left(\left\|\mathbf{d}\right\|^2 + \sigma_p^2\right)^2}{\left\|\mathbf{d}\right\|^2}$$

where

$$\left\|\mathbf{d}\right\|^2 \triangleq \sum_{i\in\mathcal{I}}\left(c_i - p_i\right)^2,\ \sigma_p^2 \triangleq \sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} g_{i,e}a^2\left(e\right) - \sum_{i\in\mathcal{I}} p_i^2.$$

Theorem 6. The pairwise error probability between a SSCW $\mathbf{c}$ and a JD-SSPCW $\mathbf{p}$ is

$$\Pr\left(\mathbf{c}\to\mathbf{p}\right) = Q\left(\frac{d_{gen}\left(\mathbf{c},\,\mathbf{p}\right)}{2\sigma}\right),$$

where $Q\left(x\right) = \int_x^\infty \mathsf{e}^{-t^2/2}/\sqrt{2\pi}dt$.

Proof. The pairwise error probability $\Pr\left(\mathbf{c}\to\mathbf{p}\right)$ that the LP joint-decoder will choose the pseudo-codeword $\mathbf{p}$ over $\mathbf{c}$ can be written as

$$\begin{aligned}
&\Pr\left(\mathbf{c}\rightarrow\mathbf{p}\right)\\
&\quad=\Pr\left\{\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i}g_{i,e}\left(y_i-a\left(e\right)\right)^2\leq\sum_{i\in\mathcal{I}}\left(y_i-c_i\right)^2\right\}\\
&\quad=\Pr\left\{\ \sum_i y_i\left(c_i-p_i\right)\leq\tfrac{1}{2}\left(\sum_i c_i^2-\sum_i\sum_e g_{i,e}a^2\left(e\right)\right)\ \right\}\\
&\quad\overset{(a)}{=}Q\left(\frac{\sum_i c_i\left(c_i-p_i\right)-\frac{1}{2}\left(\sum_i c_i^2-\sum_i\sum_e g_{i,e}a^2\left(e\right)\right)}{\sigma\sqrt{\sum_i\left(c_i-p_i\right)^2}}\right)\\
&\quad\overset{(b)}{=}Q\left(\frac{\left\|\mathbf{d}\right\|^2+\sigma_p^2}{2\sigma\left\|\mathbf{d}\right\|}\right)=Q\left(\frac{d_{gen}\left(\mathbf{c},\,\mathbf{p}\right)}{2\sigma}\right),
\end{aligned}$$

where (a) follows from the fact that $\sum_i y_i\left(c_i-p_i\right)$ has a Gaussian distribution with mean $\sum_i c_i(c_i-p_i)$ and variance $\sum_i(c_i-p_i)^2$, and (b) follows from Definition 14. □

The performance degradation of LP decoding relative to ML decoding can be explained by pseudo-codewords and their contribution to the error rate, which depends on $d_{gen}\left(\mathbf{c},\,\mathbf{p}\right)$. Indeed, by defining $K_{d_{gen}}(\mathbf{c})$ as the number of codewords and JD-PCWs at distance d_{gen} from $\mathbf{c}$ and $\mathcal{G}(\mathbf{c})$ as the set of generalized Euclidean distances, we can write the union bound on word error rate (WER) as

$$P_{w|\mathbf{c}}\leq\sum_{d_{gen}\in\mathcal{G}(\mathbf{c})}K_{d_{gen}}(\mathbf{c})\,Q\left(\frac{d_{gen}}{2\sigma}\right).\tag{2.3}$$

Of course, we need the set of JD-TPCWs to compute $\Pr\left(\mathbf{c}\rightarrow\mathbf{p}\right)$ with the Theorem 6. There are two complications with this approach. One is that, like the original problem [13], no general method is known yet for computing the generalized Euclidean distance spectrum efficiently. Another is, unlike original problem, the constraint polytope may not be symmetric under codeword exchange. Therefore the decoder performance may not be symmetric under codeword exchange. Hence, the decoder performance may depend on the transmitted codeword. In this case, the pseudo-codewords will also depend on the transmitted sequence.

CHAPTER III

ITERATIVE SOLVER FOR THE JOINT LP DECODER*

In the past, the primary value of linear programming (LP) decoding was as an analytical tool that allowed one to better understand iterative decoding and its modes of failure. This is because LP decoding based on standard LP solvers is quite impractical and has a superlinear complexity in the block length. This motivated several authors to propose low-complexity algorithms for LP decoding of LDPC codes in the last five years (e.g., [20, 45, 46, 47, 48, 49, 50]). Many of these have their roots in the iterative Gauss-Seidel approach proposed by Vontobel and Koetter for approximate LP decoding [45]. This approach was also analyzed further by Burshtein [49]. Smoothed Lagrangian relaxation methods have also been proposed to solve intractable optimal inference and estimation for more general graphs (e.g., [51]).

In this chapter*, we consider the natural extension of [45, 49] to the joint-decoding LP formulation developed in Chapter II. We argue that, by taking advantage of the special dual-domain structure of the joint LP problem and replacing minima in the formulation with soft-minima, we can obtain an efficient method that solves the joint LP. While there are many ways to iteratively solve the joint LP, our main goal was to derive one as the natural analogue of turbo equalization (TE). This should lead to an efficient method for joint LP decoding whose performance is similar to that of joint LP and whose per-iteration complexity similar to that of TE. Indeed, the solution we

*This chapter is in part a reprint of the material in the papers: B.-H.Kim and H. D. Pfister, "An iterative joint linear-programming decoding of LDPC codes and finite-state channels", in *Proc. IEEE Int. Conf. Commun.*, June 2011 and and B.-H. Kim and H. D. Pfister, "Joint decoding of LDPC codes and finite-state channels via linear-programming", in *IEEE J. Select. Topics in Signal Processing*, pp. 1563-1576, Dec. 2011.

Table I. Primal Problem (Problem-P)

$$\min_{\mathbf{g},\mathbf{w}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e}$$

subject to

$$\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} = 1,\ \ \forall j \in \mathcal{J},\ \ \sum_{e\in\mathcal{T}_p} g_{p,e} = 1, \text{ for any } p \in \mathcal{I}$$

$$\sum_{\mathcal{B}\in\mathcal{E}_j,\mathcal{B}\ni i} w_{j,\mathcal{B}} = \sum_{e:x(e)=1} g_{i,e},\ \ \forall i \in \mathcal{I}, j \in \mathcal{N}\left(i\right)$$

$$\sum_{e:s'(e)=k} g_{i,e} = \sum_{e:s(e)=k} g_{i+1,e},\ \ \forall i \in \mathcal{I} \setminus N,\ k \in S$$

$$w_{j,\mathcal{B}} \geq 0,\ \ \forall j \in \mathcal{J},\ \mathcal{B} \in \mathcal{E}_j,\ \ g_{i,e} \geq 0,\ \ \forall i \in \mathcal{I},\ e \in \mathcal{T}_i.$$

provide is a fast, iterative, and provably convergent form of TE whose update rules are tightly connected to BCJR-based TE. This demonstrates that an iterative joint LP solver with a similar computational complexity as TE is feasible (see Remark 3). In practice, the complexity reduction of this iterative decoder comes at the expense of some performance loss, when compared to the joint LP decoder, due to convergence issues (discussed in Section B).

Previously, a number of authors have attempted to reverse engineer an objective function targeted by turbo decoding (and TE by association) in order to discuss its convergence and optimality [52, 53, 54]. For example, [52] uses a duality link between two optimality formulations of TE: one based on Bethe free energy optimization and the other based on constrained ML estimation. This results of this section establish a new connection between iterative decoding and optimization for the joint-decoding problem that can also be extended to turbo decoding.

Table II. Dual Problem 1st Formulation (Problem-D1)

$$\max_{\mathbf{m},\mathbf{n}} \sum_{j\in\mathcal{J}} \min_{\mathcal{B}\in\mathcal{E}_j} \left[\sum_{i\in\mathcal{B}} m_{i,j} \right] + \min_{e\in\mathcal{T}_p} \left[\Gamma_{p,e} - n_{p-1,s(e)} + n_{p,s'(e)} \right]$$

subject to

$$\Gamma_{i,e} \geq n_{i-1,s(e)} - n_{i,s'(e)},\ \forall i \in \mathcal{I} \setminus p,\ e \in \mathcal{T}_i$$

and

$$n_{0,k} = n_{N,k} = 0,\ \forall k \in S,$$

where

$$\Gamma_{i,e} \triangleq b_{i,e} - \delta_{x(e)=1} \sum_{j\in\mathcal{N}(i)} m_{i,j}.$$

Table III. Dual Problem 2nd Formulation (Problem-D2)

$$\max_{\mathbf{m}} \sum_{j\in\mathcal{J}} \min_{\mathcal{B}\in\mathcal{E}_j} \left[\sum_{i\in\mathcal{B}} m_{i,j} \right] + \min_{e\in\mathcal{T}_p} \left[\Gamma_{p,e} - \overrightarrow{n}_{p-1,s(e)} + \overleftarrow{n}_{p,s'(e)} \right]$$

where $\overrightarrow{n}_{i,k}$ is defined for $i = 1, \ldots, p-1$ by

$$-\overrightarrow{n}_{i,k} = \min_{e\in s'^{-1}(k)} -\overrightarrow{n}_{i-1,s(e_i)} + \Gamma_{i,e},\ \forall k \in \mathcal{S}$$

and $\overleftarrow{n}_{i,k}$ is defined for $i = N-1, N-2, \ldots, p$ by

$$\overleftarrow{n}_{i,k} = \min_{e\in s^{-1}(k)} \overleftarrow{n}_{i+1,s'(e_{i+1})} + \Gamma_{i+1,e},\ \forall k \in \mathcal{S}$$

starting from

$$\overrightarrow{n}_{0,k} = \overleftarrow{n}_{N,k} = 0,\ \forall k \in \mathcal{S}.$$

A. Iterative Joint LP Decoding Derivation

In Chapter II, joint LP decoder is presented as an LDPC-code constrained shortest-path problem on the channel trellis. In this section, we develop the iterative solver for the joint-decoding LP. There are few key steps in deriving iterative solution for the joint LP decoding problem. For the first step, given by the primal problem (Problem-P) in Table I, we reformulate the original LP (2.1) in Theorem 3 using only equality constraints involving the indicator variables[1] $\mathbf{g}$ and $\mathbf{w}$. The second step, given by the 1st formulation of the dual problem (Problem-D1) in Table II, follows from standard convex analysis (See Appendix A). Strong duality holds because the primal problem is feasible and bounded. Therefore, the Lagrangian dual of Problem-P is equivalent to Problem-D1 and the minimum of Problem-P is equal to the maximum of Problem-D1. From now on, we consider Problem-D1, where the code and trellis constraints separate into two terms in the objective function. See Fig. 4 for a diagram of the variables involved.

The third step, given by the 2nd formulation of the dual problem (Problem-D2) in Table III, observes that forward/backward recursions can be used to perform the optimization over $\mathbf{n}$ and remove one of the dual variable vectors. This splitting is enabled by imposing the trellis flow normalization constraint in Problem-P only at one time instant $p \in \mathcal{I}$. This detail gives N different ways to write the same LP and is an important part of obtaining update equations similar to those of TE.

Lemma 1. Problem-D1 is equivalent to Problem-D2.

[1] The valid patterns $\mathcal{E}_j \triangleq \{\mathcal{B} \subseteq \mathcal{N}(j) \,|\, |\mathcal{B}| \text{ is even}\}$ for each parity-check $j \in \mathcal{J}$ allow us to define the indicator variables $w_{j,\mathcal{B}}$ (for $j \in \mathcal{J}$ and $\mathcal{B} \in \mathcal{E}_j$) which equal 1 if the codeword satisfies parity-check j using configuration $\mathcal{B} \in \mathcal{E}_j$.

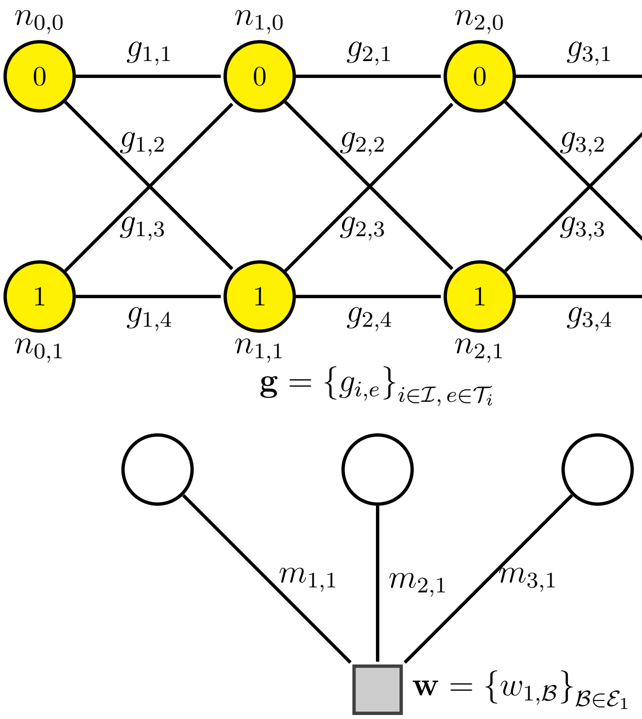


Fig. 4. Illustration of primal variables $\mathbf{g}$ and $\mathbf{w}$ defined for Problem-P and dual variables $\mathbf{n}$ and $\mathbf{m}$ defined for Problem-D1 on the same example given by Fig. 3: SPC(3,2) with DIC for $N = 3$.

Proof. By rewriting the inequality constraint in Problem-D1 as

$$-n_{i,s'(e_i)} \leq -n_{i-1,s(e_i)} + \Gamma_{i,e}$$

we obtain the recursive upper bound for $i = p - 1$ as

$$\begin{aligned}
&- n_{p-1,k} \\
&\leq -n_{p-2,s(e_{p-1})} + \Gamma_{p-1,e}\Big|_{s'(e_{p-1})=k} \\
&\leq -n_{p-3,s(e_{p-2})} + \Gamma_{p-2,e}\Big|_{s'(e_{p-2})=s(e_{p-1})} + \Gamma_{p-1,e}\Big|_{s'(e_{p-1})=k} \\
&\qquad\qquad\vdots \\
&\leq -n_{1,s(e_2)} + \sum_{i=2}^{p-1} \Gamma_{i,e}\Bigg|_{s'(e_{p-1})=k, s'(e_{p-2})=s(e_{p-1}), \ldots, s'(e_1)=s(e_2).}
\end{aligned}$$

This upper bound $-n_{p-1,k} \leq -\overrightarrow{n}_{p-1,k}$ is achieved by the forward Viterbi update in Problem-D2 for $i = 1, \ldots, p - 1$. Again, by expressing the same constraint as

$$n_{i-1,s(e_i)} \leq \Gamma_{i,e} + n_{i,s'(e_i)}$$

we get a recursive upper bound for $i = p + 1$. Similar reasoning shows this upper bound $n_{p,k} \leq \overleftarrow{n}_{p,k}$ is achieved by the backward Viterbi update in Problem-D2 for $i = N - 1, N - 2, \ldots, p$. See Fig. 5 for a graphical depiction of this. □

The fourth step, given by the softened dual problem (Problem-DS) in Table IV, is formulated by replacing the minimum operator in Problem-D2 with the soft-minimum operation

$$\min(x_1, x_2, \ldots, x_m) \approx -\frac{1}{K} \ln \sum_{i=1}^{m} \mathsf{e}^{-Kx_i}.$$

This smooth approximation converges to the minimum function as K increases [45]. Since the soft-minimum function is used in two different ways, we use different constants, K_1 and K_2, for the code and trellis terms. The smoothness of Problem-DS

Table IV. Softened Dual Problem (Problem-DS)

$$\max_{\mathbf{m}} \; -\frac{1}{K_1}\sum_{j\in\mathcal{J}} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} \qquad (3.1)$$

$$-\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p}\mathsf{e}^{-K_2\left\{\Gamma_{p,e}-\overrightarrow{n}_{p-1,s(e)}+\overleftarrow{n}_{p,s'(e)}\right\}}$$

where $\mathbb{1}_{\mathcal{B}}(i)$ is the indicator function of the set $\mathcal{B}$, $\overrightarrow{n}_{i,k}$ is defined for $i = 1, \ldots, p-1$ by

$$-\overrightarrow{n}_{i,k} = -\frac{1}{K_2}\ln\sum_{e_i\in s'^{-1}(k)}\mathsf{e}^{-K_2\left\{-\overrightarrow{n}_{i-1,s(e_i)}+\Gamma_{i,e}\right\}}, \qquad (3.2)$$

and $\overleftarrow{n}_{i,k}$ is defined for $i = N-1, N-2, \ldots, p$ by

$$\overleftarrow{n}_{i,k} = -\frac{1}{K_2}\ln\sum_{e_{i+1}\in s^{-1}(k)}\mathsf{e}^{-K_2\left\{\overleftarrow{n}_{i+1,s'(e_{i+1})}+\Gamma_{i+1,e}\right\}} \qquad (3.3)$$

starting from

$$\overrightarrow{n}_{0,k} = \overleftarrow{n}_{N,k} = 0, \; \forall k \in \mathcal{S}.$$

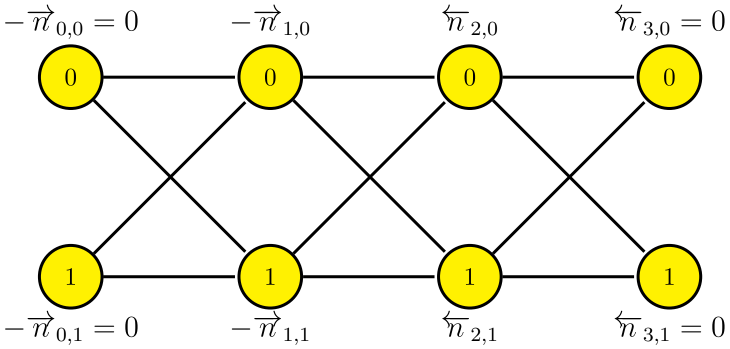

$-\overrightarrow{n}_{0,0} = 0$
$-\overrightarrow{n}_{1,0}$
$\overleftarrow{n}_{2,0}$
$\overleftarrow{n}_{3,0} = 0$
0
0
0
0
1
1
1
1
$-\overrightarrow{n}_{0,1} = 0$
$-\overrightarrow{n}_{1,1}$
$\overleftarrow{n}_{2,1}$
$\overleftarrow{n}_{3,1} = 0$

Fig. 5. Illustration of Viterbi updates in Problem-D2 on the same example given by Fig. 3: DIC for $N = 3$ with forward $\overrightarrow{\mathbf{n}}$ and backward $\overleftarrow{\mathbf{n}}$.

allows one to to take derivative of (3.1) (giving the Karush–Kuhn–Tucker (KKT) equations, derived in Lemma 2), and represent (3.2) and (3.3) using BCJR-like forward/backward recursions (given by Lemma 3).

Lemma 2. Consider the KKT equations associated with performing the minimization in (3.1) only over the variables $\{m_{p,j'}\}_{j'\in\mathcal{N}(p)}$. These equations have a unique solution given by

$$m_{p,j'} = M_{p,j'} + \frac{\gamma_p}{K_1},\;\; M_{p,j'} \triangleq \frac{1}{K_1}\ln\frac{1-l_{p,j'}}{1+l_{p,j'}}$$

for $j'\in\mathcal{N}(p)$ where

$$l_{p,j'} \triangleq \prod_{i\in\mathcal{N}(j')\setminus p}\tanh\left(\frac{K_1 m_{i,j'}}{2}\right),$$

and

$$\gamma_p \triangleq \ln\frac{\sum_{e\in\mathcal{T}_p:x(e)=0}\mathsf{e}^{-K_2\left(\Gamma_p-\overrightarrow{n}_{p-1,s(e)}+\overleftarrow{n}_{p,s'(e)}\right)}}{\sum_{e\in\mathcal{T}_p:x(e)=1}\mathsf{e}^{-K_2\left(\Gamma_p-\overrightarrow{n}_{p-1,s(e)}+\overleftarrow{n}_{p,s'(e)}\right)}}.$$

Proof. Restricting the minimization in (3.1) to the variables $\{m_{p,j'}\}_{j'\in\mathcal{N}(p)}$ gives

$$-\min_{\{m_{p,j}\}_{j\in\mathcal{N}(p)}}\left\{\frac{1}{K_1}\sum_{j\in\mathcal{N}(p)}\ln\sum_{\mathcal{B}\in\mathcal{E}_j}\mathsf{e}^{-K_1\sum_{i\in\mathcal{N}(j)}m_{i,j}\mathbb{1}_{\mathcal{B}}(i)}+\right.$$
$$\left.\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p}\mathsf{e}^{-K_2\left(\Gamma_{p,e}-\overrightarrow{n}_{p-1,s(e)}+\overleftarrow{n}_{p,s'(e)}\right)}\right\}. \tag{3.4}$$

The solution to (3.4) can be obtained by solving the KKT equations. For $p\in\mathcal{I}$, we take the first derivative with respect to $\{m_{p,j'}\}_{j'\in\mathcal{N}(p)}$ and set it to zero; this yields

$$\left(\frac{\sum_{\mathcal{B}\in\mathcal{E}_{j'},p\notin\mathcal{B}}\mathsf{e}^{-K_1\sum_{i\in\mathcal{N}(j')\setminus p}m_{i,j'}\mathbb{1}_{\mathcal{B}}(i)}}{\sum_{\mathcal{B}\in\mathcal{E}_{j'},B\ni p}\mathsf{e}^{-K_1\sum_{i\in\mathcal{N}(j')\setminus p}m_{i,j'}\mathbb{1}_{\mathcal{B}}(i)}}\right)\cdot\mathsf{e}^{K_1 m_{p,j'}} =$$
$$\left(\frac{\sum_{e\in\mathcal{T}_p:x(e)=0}\mathsf{e}^{-K_2\left(\Gamma_{p,e}-\overrightarrow{n}_{p-1,s(e)}+\overleftarrow{n}_{p,s'(e)}\right)}}{\sum_{e\in\mathcal{T}_p:x(e)=1}\mathsf{e}^{-K_2\left(\Gamma_{p,e}-\overrightarrow{n}_{p-1,s(e)}+\overleftarrow{n}_{p,s'(e)}\right)}}\right) \tag{3.5}$$

By defining $-K_1 M_{p,j'}$ as

$$\begin{aligned}
&\ln \frac{\sum_{\mathcal{B}\in\mathcal{E}_{j'},p\notin\mathcal{B}} \mathsf{e}^{-K_1\sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}\mathbb{1}_{\mathcal{B}}(i)}}{\sum_{\mathcal{B}\in\mathcal{E}_{j'},\mathcal{B}\ni p} \mathsf{e}^{-K_1\sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}\mathbb{1}_{\mathcal{B}}(i)}} \\
=&\ln \frac{\prod_{i\in\mathcal{N}(j')\backslash p}\left(1+\nu_{i,j'}\right)+\prod_{i\in\mathcal{N}(j')\backslash p}\left(1-\nu_{i,j'}\right)}{\prod_{i\in\mathcal{N}(j')\backslash p}\left(1+\nu_{i,j'}\right)-\prod_{i\in\mathcal{N}(j')\backslash p}\left(1-\nu_{i,j'}\right)} \\
=&-\ln\frac{1-l_{p,j'}}{1+l_{p,j'}},
\end{aligned} \tag{3.6}$$

where $\nu_{i,j'} \triangleq \mathsf{e}^{-K_1 m_{i,j'}}$, we can rewrite (3.5) to obtain the desired result. □

Lemma 3. Equations (3.2) and (3.3) are equivalent to the BCJR-based forward and backward recursion given by (3.7), (3.8), and (3.9).

Proof. By letting, $\alpha_i(k) \propto \mathsf{e}^{K_2 \overrightarrow{n}_{i,k}}$, $\lambda_{i+1,e} = \mathsf{e}^{-K_2\Gamma_{i+1,e}}$, and $\beta_i(k) \propto \mathsf{e}^{-K_2 \overleftarrow{n}_{i,k}}$, we obtain the desired result by normalization. □

Now, we have all the pieces to complete the algorithm. As the last step, we combine the results of Lemma 2 and 3 to obtain the iterative solver for the joint-decoding LP, which is summarized by the iterative joint LP decoding in Algorithm 1 (see Fig. 6 for a graphical depiction).

Remark 3. *While Algorithm 1 always has a bit-node update rule different from standard belief propagation (BP), we note that setting $K_1 = 1$ in the inner loop gives the exact BP check-node update and setting $K_2 = 1$ in the outer loop gives the exact BCJR channel update. In fact, one surprising result of this work is that such a small change to the BCJR-based TE update provides an iterative solver for the LP whose per-iteration complexity similar to TE. It is also possible to prove the convergence of a slightly modified iterative solver that is based on a less efficient update schedule (See Figs. 7 - 12 for details).*

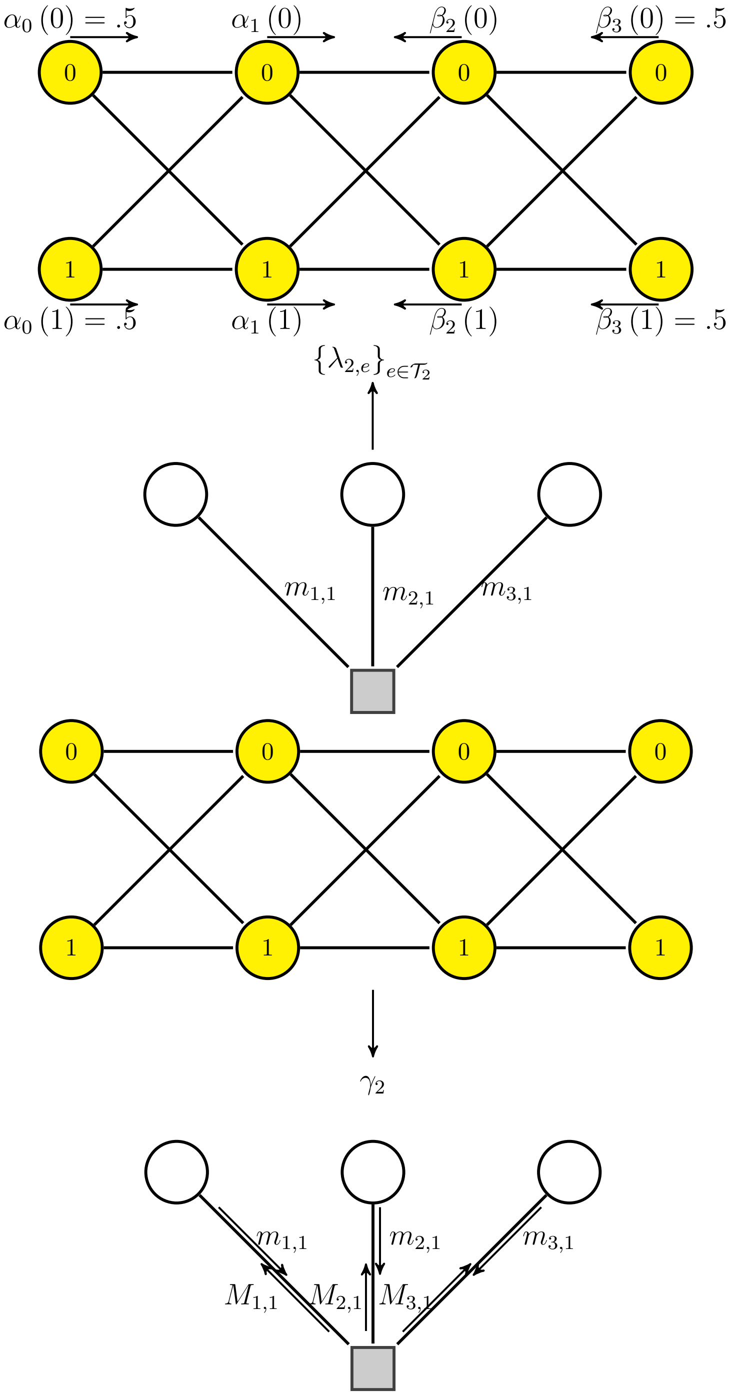


Fig. 6. Illustration of Algorithm 1 steps for $i = 2$ on the same example given by Fig. 3: Outer loop update (top) and inner loop update (bottom).

Algorithm 1 Iterative Joint Linear-Programming Decoding

```
• Step 1. Set $\ell = 0$ and initialize $m_{i,j} = 0$ for $i \in \mathcal{I}$, $j \in \mathcal{N}(i)$.
• Step 2. Update Outer Loop: For $i \in \mathcal{I}$,
    – (i) Compute bit-to-trellis message
          $\lambda_{i,e} = \mathsf{e}^{-K_2 \left\{ b_{i,e} - \delta_{x(e)=1} \sum_{j \in \mathcal{N}(i)} m_{i,j} \right\}}$
    – (ii) Compute forward/backward trellis messages
          $\alpha_{i+1}(k) = \frac{\sum_{e \in s'^{-1}(k)} \alpha_i(s(e)) \cdot \lambda_{i+1,e}}{\sum_k \sum_{e \in s'^{-1}(k)} \alpha_i(s(e)) \cdot \lambda_{i+1,e}}$    (3.7)
          $\beta_{i-1}(k) = \frac{\sum_{e \in s^{-1}(k)} \beta_i(s'(e)) \cdot \lambda_{i,e}}{\sum_k \sum_{e \in s^{-1}(k)} \beta_i(s'(e)) \cdot \lambda_{i,e}},$    (3.8)
      where $\beta_N(k) = \alpha_0(k) = 1/|\mathcal{S}|$ for all $k \in \mathcal{S}$.
    – (iii) Compute trellis-to-bit message $\gamma_i$
          $\gamma_i = \ln \frac{\sum_{e \in \mathcal{T}_i : x(e)=0} \alpha_{i-1}(s(e)) \lambda_{i,e} \beta_i(s'(e))}{\sum_{e \in \mathcal{T}_i : x(e)=1} \alpha_{i-1}(s(e)) \lambda_{i,e} \beta_i(s'(e))}$    (3.9)
• Step 3. Update Inner Loop for $\ell_{\text{inner}}$ rounds: For $i \in \mathcal{I}$,
    – (i) Compute bit-to-check msg $m_{i,j}$ for $j \in \mathcal{N}(i)$
          $m_{i,j} = \frac{\gamma_i}{K_1} - M_{i,j}$
    – (ii) Compute check-to-bit msg $M_{i,j}$ for $j \in \mathcal{N}(i)$
          $M_{i,j} = \frac{2}{K_1} \tanh^{-1} \left( \prod_{r \in \mathcal{N}(j) \setminus i} \tanh \left( \frac{K_1 m_{r,j}}{2} \right) \right)$    (3.10)
• Step 4. Compute hard decisions and stopping rule
    – (i) For $i \in \mathcal{I}$,
          $\hat{f}_i = (1 - \operatorname{sgn}(\gamma_i))/2$
    – (ii) If $\hat{\mathbf{f}}$ satisfies all parity checks or the maximum outer iteration number, $\ell_{\text{outer}}$,
      is reached, stop and output $\hat{\mathbf{f}}$. Otherwise increment $\ell$ and go to Step 2.
```

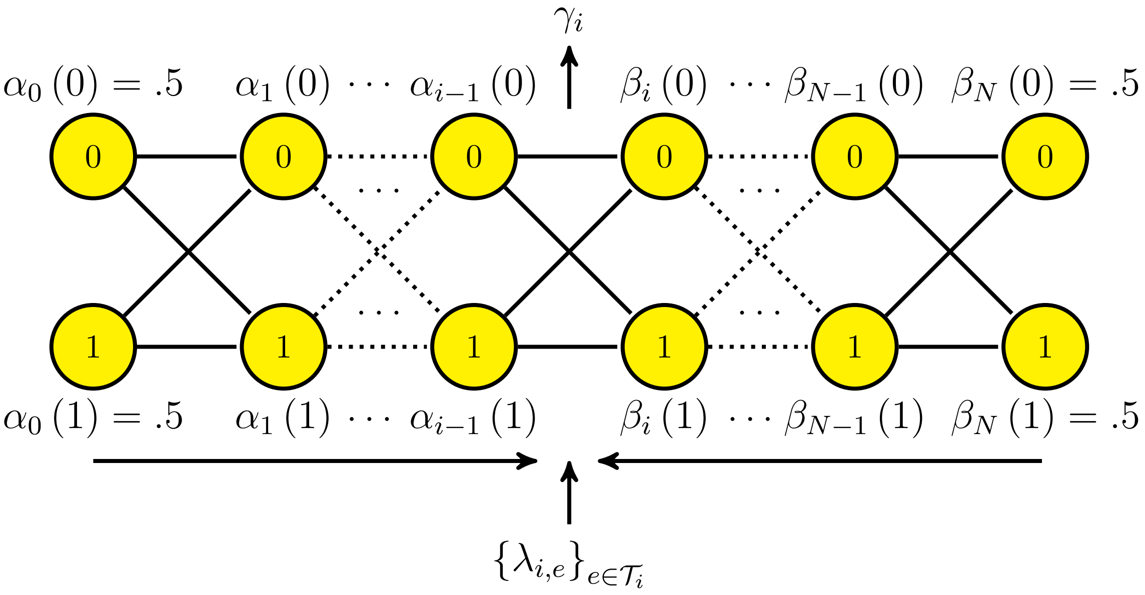

γ_i
$\alpha_0(0) = .5$
$\alpha_1(0) \cdots \alpha_{i-1}(0)$
$\beta_i(0) \cdots \beta_{N-1}(0)$
$\beta_N(0) = .5$
0
0
0
0
0
0
1
1
1
1
1
1
$\alpha_0(1) = .5$
$\alpha_1(1) \cdots \alpha_{i-1}(1)$
$\beta_i(1) \cdots \beta_{N-1}(1)$
$\beta_N(1) = .5$
$\{\lambda_{i,e}\}_{e\in\mathcal{T}_i}$

Fig. 7. Messages passing through the trellis for joint iterative MP decoding: Recursive BCJR update with forward/backward state probabilities $\alpha_i(s)$, $\beta_i(s)$ and edge-output probabilities $\lambda_{i,e}$.

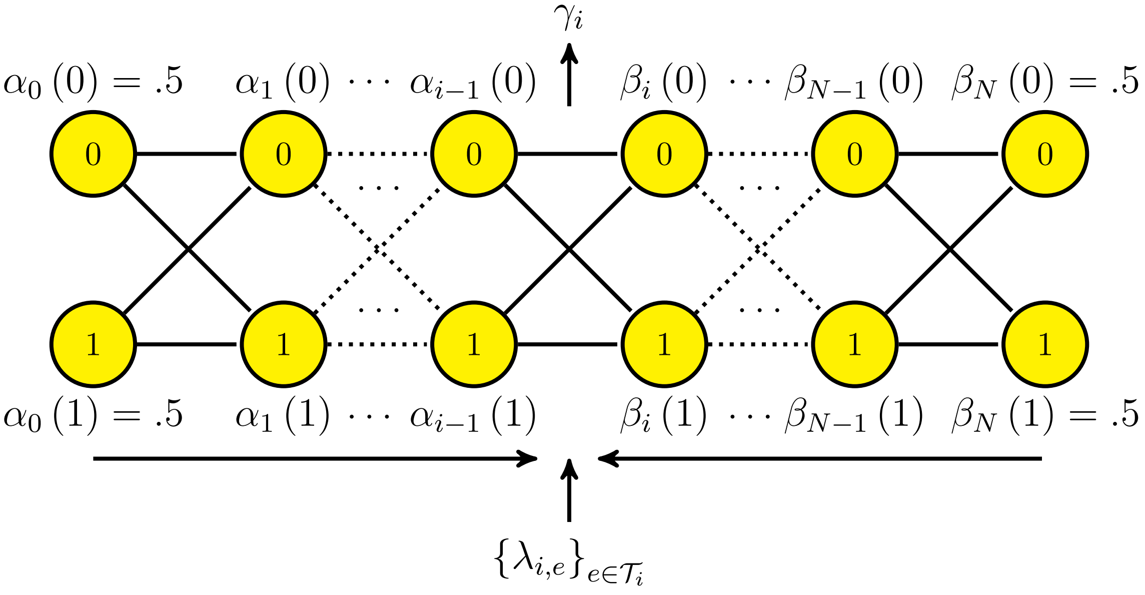

γ_i
$\alpha_0(0) = .5$
$\alpha_1(0) \cdots \alpha_{i-1}(0)$
$\beta_i(0) \cdots \beta_{N-1}(0)$
$\beta_N(0) = .5$
0
0
0
0
0
0
1
1
1
1
1
1
$\alpha_0(1) = .5$
$\alpha_1(1) \cdots \alpha_{i-1}(1)$
$\beta_i(1) \cdots \beta_{N-1}(1)$
$\beta_N(1) = .5$
$\{\lambda_{i,e}\}_{e\in\mathcal{T}_i}$

Fig. 8. Messages passing through the trellis for joint iterative LP decoding: Recursive BCJR update with different edge-output probabilities $\lambda_{i,e}$ (setting $K_1 = K_2 = 1$ gives the exact BCJR channel update).

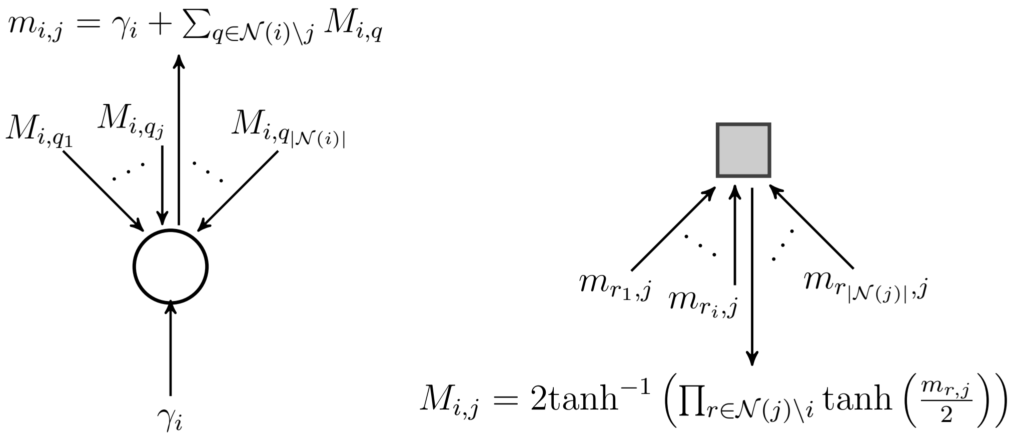

$m_{i,j} = \gamma_i + \sum_{q \in \mathcal{N}(i) \setminus j} M_{i,q}$
M_{i,q_1}
M_{i,q_j}
$M_{i,q_{|\mathcal{N}(i)|}}$
γ_i
$m_{r_1,j}$
$m_{r_i,j}$
$m_{r_{|\mathcal{N}(j)|},j}$
$M_{i,j} = 2\tanh^{-1}\left(\prod_{r \in \mathcal{N}(j) \setminus i} \tanh\left(\frac{m_{r,j}}{2}\right)\right)$

Fig. 9. Computation of code update messages for joint iterative MP decoding: Standard BP update with bit-to-check messages (left) and check-to-bit messages (right).

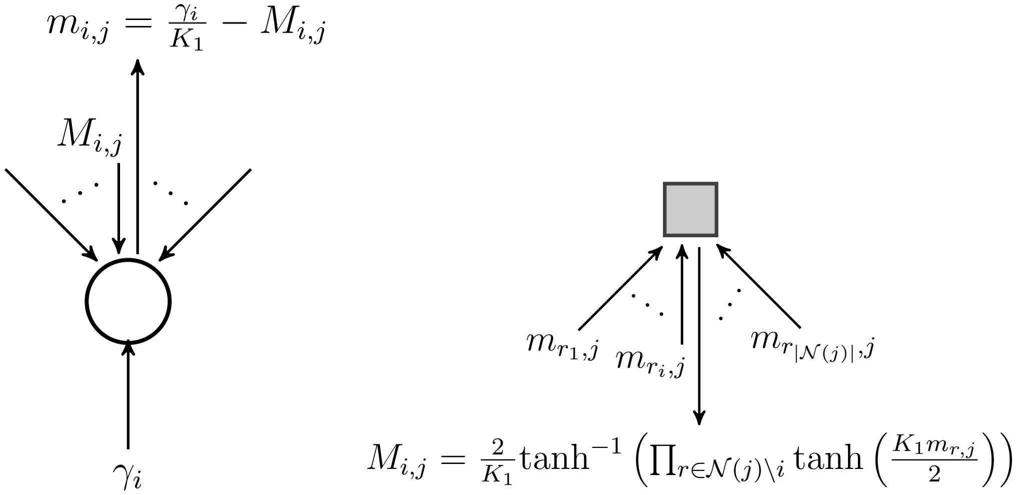

$m_{i,j} = \frac{\gamma_i}{K_1} - M_{i,j}$
$M_{i,j}$
γ_i
$m_{r_1,j}$
$m_{r_i,j}$
$m_{r_{|\mathcal{N}(j)|},j}$
$M_{i,j} = \frac{2}{K_1}\tanh^{-1}\left(\prod_{r \in \mathcal{N}(j) \setminus i} \tanh\left(\frac{K_1 m_{r,j}}{2}\right)\right)$

Fig. 10. Computation of code update messages for joint iterative LP decoding: Bit-to-check messages (left) and BP check update with hardening parameter K_1 for check-to-bit messages (right).

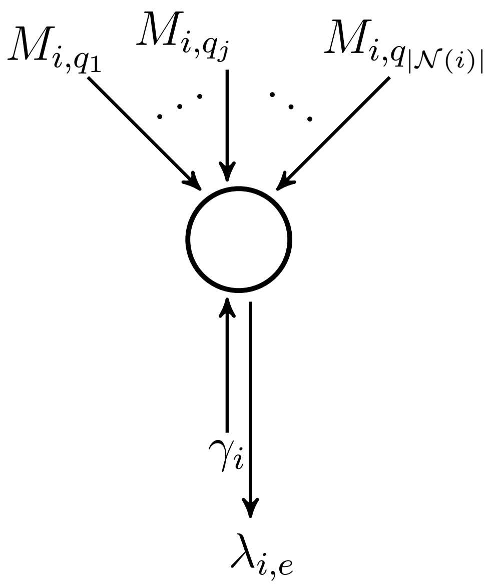


Fig. 11. Bit-to-trellis messages for joint iterative MP decoding: Passing extrinsic information with $\lambda_{i,e} = \mathsf{e}^{-\left\{b_{i,e}-\delta_{x(e)=0}\left(\sum_{q\in\mathcal{N}(i)} M_{i,q}-\gamma_i\right)\right\}}$.

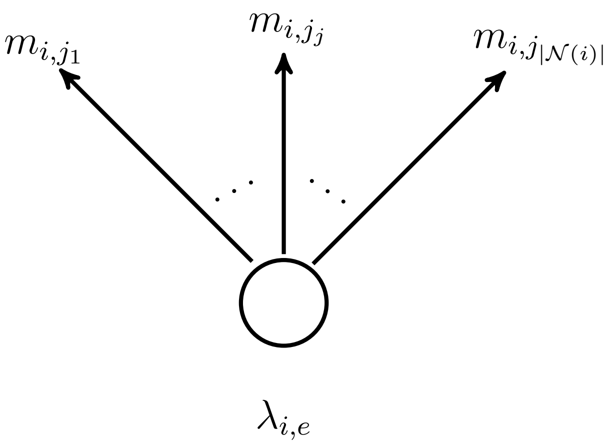


Fig. 12. Bit-to-trellis messages for joint iterative LP decoding: Passing with $\lambda_{i,e} = \mathsf{e}^{-K_2\left(b_{i,e}-\delta_{x(e)=1}\sum_{j\in\mathcal{N}(i)} m_{i,j}\right)}$.

B. Convergence Analysis

This section considers the convergence properties of Algorithm 1. Although simulations have not shown any convergence problems with Algorithm 1 in its current form, our proof requires a modified update schedule that is less computationally efficient. Following Vontobel's approach in [45], which is based on general properties of Gauss-Seidel-type algorithms for convex minimization, we show that the modified version Algorithm 1 is guaranteed to converge. Moreover, a feasible solution to Problem-P can be obtained whose value is arbitrarily close to the optimal value of Problem-P.

The modified update rule for Algorithm 1 consists of cyclically, for each $p = 1, \ldots, N$, computing the quantity γ_p (via step 2 of Algorithm 1) and then updating $m_{p,j}$ for all $j \in \mathcal{N}(p)$ (based on step 3 of Algorithm 1). The drawback of this approach is that one BCJR update is required for each bit update, rather than for N bit updates. This modification allows us to interpret Algorithm 1 as a Gauss-Seidel-type algorithm. We believe that, at the expense of a longer argument, the convergence proof can be extended to a decoder which uses windowed BCJR updates (e.g., see [55]) to achieve convergence guarantees with much lower complexity. Regardless, the next few lemmas and theorems can be seen as a natural generalization of [45, 49] to the joint-decoding problem.

Proposition 1. Consider the problem

$$\min_{x \in \mathcal{X}} f(x)$$

where $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_m$ and each $\mathcal{X}_i$ is a closed convex subset of $\mathbb{R}^{n_i}$. The vector x is partitioned so $x = (x_1, x_2, \ldots, x_m)$ with $x_i \in \mathbb{R}^{n_i}$. Suppose that f is *continuously differentiable and convex* on $\mathcal{X}$ and that, for every $x \in \mathcal{X}$ and every

$i = 1, \ldots, m$, the problem

$$\min_{\xi_i \in \mathcal{X}_i} f\left(x_1, \, \ldots \, , \, x_{i-1}, \, \xi_i, \, x_{i+1}, \, \ldots \, , \, x_m\right)$$

has a *unique minimum*. Now, consider the sequence $x^{k+1} = \left(x_1^{k+1}, \, \ldots \, , \, x_m^{k+1}\right)$ defined by

$$x_i^{k+1} = \arg\min_{\xi_i \in \mathcal{X}_i} f\left(x_1^{k+1}, \, \ldots \, , \, x_{i-1}^{k+1}, \, \xi_i, \, x_{i+1}^{k}, \, \ldots \, , \, x_m^{k}\right),$$

for $i = 1, \ldots, m$. Then, every limit point of this sequence minimizes f over $\mathcal{X}$.

Lemma 4. Assume that all the rows of H have Hamming weight at least 3. Then, the modified Algorithm 1 converges to the maximum of the Problem-DS.

Proof. To characterize the convergence of the iterative joint LP decoder, we consider the modification of Algorithm 1 with cyclic updates. The analysis follows [45] and uses the proposition about *convergence of block coordinate descent methods* from [56, p. 247]. By using Proposition 1, we will show that the modified Algorithm 1 converges. Define $\mathbf{m}_i = \left\{m_{i,j}\right\}_{j \in \mathcal{N}(i)}$ and

$$\begin{aligned} f\left(\mathbf{m}\right) \triangleq & f\left(\mathbf{m}_1, \, \ldots \, , \, \mathbf{m}_N\right) \\ = & \frac{1}{K_1} \sum_{j \in J} \ln \sum_{\mathcal{B} \in \mathcal{E}_j} \mathsf{e}^{-K_1 \left\{\sum_{i \in \mathcal{N}(j)} m_{i,j} \mathbb{1}_{\mathcal{B}}(i)\right\}} + \\ & \frac{1}{K_2} \ln \sum_{e \in \mathcal{T}_p} \mathsf{e}^{-K_2 \left\{\Gamma_{p,e} - \overrightarrow{n}_{p-1,s(e_p)} + \overleftarrow{n}_{p,s'(e_p)}\right\}}. \end{aligned}$$

Let us consider cyclic coordinate decent algorithm which minimizes f cyclically with respect to the coordinate variable. Thus $\mathbf{m}_1$ is changed first, then $\mathbf{m}_2$ and so forth through $\mathbf{m}_N$. Then (3.1), (3.2), and (3.3) are equivalent to for each $p \in \mathcal{I}$ with proper

Table V. Softened Primal Problem (Problem-PS)

$$\min_{\mathbf{g},\mathbf{w}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e} - \frac{1}{K_1} \sum_{j\in\mathcal{J}} H(w_j) - \frac{1}{K_2} H(g_p)$$

subject to the same constraints as Problem-P.

$\mathcal{X}_p$ as

$$\begin{aligned}
&\min_{\xi_p\in\mathcal{X}_p} f\left(\mathbf{m}_1,\,\ldots\,,\,\mathbf{m}_{p-1},\,\xi_p,\,\mathbf{m}_{p+1},\,\ldots\,,\,\mathbf{m}_N\right)\\
&= \min_{\xi_p\in\mathcal{X}_p} \frac{1}{K_1} \sum_{j\in J} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\xi_{p,j}\mathbb{1}_{\mathcal{N}(j)}(p)\mathbb{1}_{\mathcal{B}}(p)+\sum\limits_{i\in\mathcal{N}(j)\backslash p} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}}\\
&+\frac{1}{K_2} \ln \sum_{e\in\mathcal{T}_p} \exp\left\{-K_2\left(b_{p,e} - \sum_{j\in\mathcal{N}(p)} \xi_{p,j}\delta_{x(e_p)=1}\right)\right.\\
&+\ln \sum_{\{e_1,\ldots,e_{p-1}\}} \mathsf{e}^{-K_2\left\{n_{1,s(e_2)}-\sum\limits_{i=2}^{p-1} b_{i,e}+\sum\limits_{i=2}^{p-1}\sum\limits_{j\in\mathcal{N}(i)} m_{i,j}\delta_{x(e_i)=1}\right\}}\\
&\left.+\ln \sum_{\{e_{p+1},\ldots,e_N\}} \mathsf{e}^{-K_2\left\{\sum\limits_{i=p+1}^{N} b_{i,e}-\sum\limits_{i=p+1}^{N}\sum\limits_{j\in\mathcal{N}(i)} m_{i,j}\delta_{x(e_i)=1}\right\}}\right\}.
\end{aligned}$$

Using the properties of log-sum-exp functions (e.g., see [57, p. 72]), one can verify that f is continuously differentiable and convex. The minimum over ξ_p for all $p \in \mathcal{I}$ is uniquely obtained because of the unique KKT solution in Lemma 2. Therefore, we can apply the Proposition 1 to achieve the desired convergence result under the modified update schedule. It is worth mentioning that the Hamming weight condition prevents degeneracy of Problem-DS based on the fact that, otherwise, some pairs of bits must always be equal. □

Next, we introduce the softened primal problem (Problem-PS) in Table V, using the definitions $w_j \triangleq \{w_{j,\mathcal{B}}\}_{\mathcal{B}\in\mathcal{E}_j}$ and $g_p \triangleq \{g_{p,e}\}_{e\in\mathcal{T}_p}$. Using standard convex analysis (See Appendix B), one can show that Problem-PS is the Lagrangian dual of Problem-

DS and that the minimum of Problem-PS is equal to the maximum of Problem-DS. In particular, Problem-PS can be seen as a maximum-entropy regularization of Problem-DS that was derived by smoothing dual problem given by Problem-D2. Thus, Algorithm 1 is dually-related to an interior-point method for solving the LP relaxation of joint ML decoding on trellis-wise polytope using the entropy function (for x in the standard simplex)

$$H(x) \triangleq -\sum_{i} x_i \ln x_i \tag{3.11}$$

as a barrier function (e.g., see [51, p. 126]) for the polytope.

Remark 4. *By taking sufficiently large K_1 and K_2, the primal LP of joint LP decoder in Problem-P, emerges as the "zero temperature" limit of the approximate LP relaxations given by Problem-PS [45, 51]. Also, Problem-PS can be seen as a convex free-energy minimization problem [51].*

Next, we develop a relaxation bound, given by Lemma 5 and Lemma 6 to quantify the performance loss of Algorithm 1 (when it converges) in relation to the joint LP decoder.

Lemma 5. Let P^* be the minimum value of Problem-P and $\tilde{P}$ be the minimum value of Problem-PS. Then, one finds that

$$0 \leq \tilde{P} - P^* \leq \delta N,$$

where

$$\bar{\mathcal{N}} \triangleq \frac{\sum_{j\in\mathcal{J}} |\mathcal{N}(j)|}{N},\ R \triangleq 1 - \frac{M}{N}$$

and

$$\delta \triangleq \frac{\left(1 - R + \bar{\mathcal{N}}\right) \ln 2}{K_1} + \frac{\ln O}{K_2 N}.$$

Proof. Denote the optimum solution of Problem-P by $\mathbf{g}^*$ and $\mathbf{w}^*$ and the optimum solution of Problem-PS by $\tilde{\mathbf{g}}$ and $\tilde{\mathbf{w}}$. Since $\mathbf{g}^*$ and $\mathbf{w}^*$ are the optimal with respect to the Problem-P, we have

$$P^* = \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} g^*_{i,e} \leq \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} \tilde{g}_{i,e} = \tilde{P}. \tag{3.12}$$

On the other hand, $\tilde{\mathbf{g}}$ and $\tilde{\mathbf{w}}$ are the optimal with respect to the Problem-PS, we have

$$\begin{aligned} &\sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} \tilde{g}_{i,e} - \frac{1}{K_1} \sum_{j \in \mathcal{J}} H(\tilde{w}_j) - \frac{1}{K_2} H(\tilde{g}_p) \\ &\qquad \leq \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} g^*_{i,e} - \frac{1}{K_1} \sum_{j \in \mathcal{J}} H(w^*_j) - \frac{1}{K_2} H(g^*_p), \end{aligned}$$

where $H(\cdot)$ is the entropy defined by (3.11). Rewrite this gives

$$\begin{aligned} &\sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} \tilde{g}_{i,e} \\ &\leq \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} g^*_{i,e} + \frac{1}{K_1} \left(\sum_{j \in \mathcal{J}} H(\tilde{w}_j) - \sum_{j \in \mathcal{J}} H(w^*_j) \right) \\ &+ \frac{1}{K_2} \left(H(\tilde{g}_p) - H(g^*_p) \right) \\ &\leq \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{T}_i} b_{i,e} g^*_{i,e} + \frac{1}{K_1} \sum_{j \in \mathcal{J}} H(\tilde{w}_j) + \frac{1}{K_2} H(\tilde{g}_p). \end{aligned} \tag{3.13}$$

The last inequality is due to nonnegativity of entropy. Using Jensen's inequality, we obtain

$$\begin{aligned} \sum_{j \in \mathcal{J}} H(\tilde{w}_j) &\leq \sum_{j \in \mathcal{J}} \ln |\mathcal{E}_j| = \sum_{j \in \mathcal{J}} \left(|\mathcal{N}(j)| - 1 \right) \ln 2 \\ &= N \left(1 - R + \bar{\mathcal{N}} \right) \ln 2 \end{aligned} \tag{3.14}$$

and

$$H(\tilde{g}_p) \leq \ln O. \tag{3.15}$$

By substituting (3.14) and (3.15) to (3.13), we have

$$\tilde{P} - P^* \leq \frac{N\left(1 - R + \bar{\mathcal{N}}\right)\ln 2}{K_1} + \frac{\ln O}{K_2}. \tag{3.16}$$

Combining (3.12) and (3.16) gives the result. □

Lemma 6. For any $\epsilon > 0$, sufficiently many iterations of the modified Algorithm 1 produces a feasible solution for Problem-DS that satisfies the KKT conditions within ϵ. If $\epsilon < 1/6$, then one can construct a solution $(\tilde{\mathbf{g}}_\epsilon, \tilde{\mathbf{w}}_\epsilon)$ for Problem-PS that is feasible and whose value $\tilde{P}_\epsilon$ satisfies

$$0 \leq \tilde{P}_\epsilon - \tilde{P} \leq \delta N,$$

where

$$\delta \triangleq \frac{\left(1 - R + \bar{\mathcal{N}}\right)\ln 2}{K_1} + \epsilon\left(\frac{3}{N}\sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l}|b_{l,e}| + C\right).$$

Proof. Proof follows from careful modification of the arguments in [49, p. 4840-4841]. For the coordinate-descent solution of Problem-DS, minimizing over the p-th block gives

$$-\min_{\{m_{p,j}\}_{j\in\mathcal{N}(p)}} \frac{1}{K_1}\sum_{j\in\mathcal{N}(p)} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} \tag{3.17}$$

subject to

$$\Gamma_{p,e} = \overrightarrow{n}_{p-1,s(e)} - \overleftarrow{n}_{p,s'(e)}, \, \forall e \in \mathcal{T}_p.$$

The solution can be obtained by applying the KKT conditions and this yields

$$\frac{\sum_{e:x(e)=1}\lambda_{p,e}}{1 - \sum_{e:x(e)=1}\lambda_{p,e}} = \mathsf{e}^{K_1(M_{p,j} - m_{p,j})}. \tag{3.18}$$

Given a feasible solution of the modified Algorithm 1, we define

$$\begin{aligned}\lambda_i^j &\triangleq \sum_{e:x(e)=1} \lambda_{i,e}^j = \frac{1}{1+\mathsf{e}^{K_1(m_{i,j}-M_{i,j})}},\\ \lambda_i &\triangleq \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} \lambda_i^j = \sum_{e:x(e)=1} \lambda_{i,e}\end{aligned}$$

with

$$\lambda_{i,e} \triangleq \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} \lambda_{i,e}^j$$

and

$$\epsilon \triangleq \max_{i\in\mathcal{I}} \max_{j\in\mathcal{N}(i)} |\lambda_i^j - \lambda_i|.$$

Suppose we stop iterating when $\epsilon \leq \frac{1}{6}$ and define

$$\begin{aligned}\hat{\lambda}_i &\triangleq (1-6\epsilon)\,\lambda_i + 6\epsilon \sum_{e:x(e)=1} \frac{1}{|E|}\\ &= (1-6\epsilon)\,\lambda_i + 3\epsilon = \sum_{e:x(e)=1} \hat{\lambda}_{i,e},\end{aligned}$$

where

$$\hat{\lambda}_{i,e} \triangleq (1-6\epsilon)\,\lambda_{i,e} + \frac{6\epsilon}{|E|}.$$

First, we claim that $\hat{\lambda} \triangleq \left\{\hat{\lambda}_i\right\} \in \mathcal{P}(H)$. This is because setting

$$w_{j,\mathcal{B}} \triangleq \frac{\mathsf{e}^{-K_1 \sum_{l\in\mathcal{N}(j)} m_{l,j} \mathbb{1}_{\mathcal{B}}(l)}}{\sum_{\mathcal{B}'\in\mathcal{E}_j} \mathsf{e}^{-K_1 \sum_{l\in\mathcal{N}(j)} m_{l,j} \mathbb{1}_{\mathcal{B}'}(l)}} \tag{3.19}$$

obviously satisfies for $\forall j \in \mathcal{J}$

$$w_{j,\mathcal{B}} \geq 0,\ \ \forall \mathcal{B} \in \mathcal{E}_j,\ \ \sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} = 1$$

and satisfies for $\forall i \in \mathcal{I},\ j \in \mathcal{N}\left(i\right)$

$$\sum_{\mathcal{B}\in\mathcal{E}_j,\mathcal{B}\ni i} w_{j,\mathcal{B}} = \frac{\sum_{\mathcal{B}\in\mathcal{E}_j,\mathcal{B}\ni i} \mathsf{e}^{-K_1 \sum_{l\in\mathcal{N}(j)} m_{l,j} \mathbb{1}_{\mathcal{B}}(l)}}{\sum_{\mathcal{B}'\in\mathcal{E}_j} \mathsf{e}^{-K_1 \sum_{l\in\mathcal{N}(j)} m_{l,j} \mathbb{1}_{\mathcal{B}'}(l)}} = \lambda_i^j.$$

From [49, p. 4841], it follows that $\tilde{\lambda} \in \mathcal{P}(H)$. Next, we show that $\left\{\hat{\lambda}_{i,e}\right\} \in \mathcal{T}$. Note that defining

$$\lambda_{i,e} \triangleq \frac{\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}{\sum_{e\in\mathcal{T}_i}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}$$

implies that (by (3.5))

$$\frac{\sum_{e:x(e)=1}\lambda_{i,e}}{1-\sum_{e:x(e)=1}\lambda_{i,e}} = \mathsf{e}^{K_1(M_{p,j}-m_{p,j})},$$

obviously satisfies for $\forall i \in \mathcal{I}$

$$\lambda_{i,e} \geq 0,\ \ \forall e \in \mathcal{T}_i,\ \sum_{e\in\mathcal{T}_i}\lambda_{i,e} = 1$$

and for $\forall i \in \mathcal{I} \setminus N,\ \ k \in S$ by (3.2) and (3.3)

$$\begin{aligned}\sum_{e:s'(e)=k}\lambda_{i,e} &= \frac{\sum_{e:s'(e)=k}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}{\sum_{e\in\mathcal{T}_i}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}\\ &= \sum_{e:s(e)=k}\lambda_{i+1,e}.\end{aligned}$$

Furthermore,

$$\sum_{e\in\mathcal{T}_i}\hat{\lambda}_{i,e} = (1-6\epsilon)\sum_{e\in\mathcal{T}_i}\lambda_{i,e} + 6\epsilon\sum_{e\in\mathcal{T}_i}\frac{1}{|E|} = 1,$$

$$\begin{aligned}\sum_{e:s'(e)=k}\hat{\lambda}_{i,e} &= (1-6\epsilon)\sum_{e:s'(e)=k}\lambda_{i,e} + 6\epsilon\sum_{e:s'(e)=k}\frac{1}{|E|}\\ &= (1-6\epsilon)\sum_{e:s(e)=k}\lambda_{i+1,e} + 6\epsilon\sum_{e:s(e)=k}\frac{1}{|E|}\\ &= \sum_{e:s(e)=k}\hat{\lambda}_{i+1,e},\end{aligned}$$

and by Definition 7, $\hat{\lambda} \in \mathcal{P}(H)$. Therefore, we conclude that $\left\{\hat{\lambda}_{i,e}\right\} \in \mathcal{P}_{\mathcal{T}}(H)$ is feasible in Problem-P. From [49, p. 4855], it follows that there exist feasible $\hat{w}_j$

vectors associated with $\left\{\hat{\lambda}_{i,e}\right\}$. Furthermore for $\forall i \in \mathcal{I}, j \in \mathcal{N}(i)$

$$\begin{aligned}
\sum_{\mathcal{B}\in\mathcal{E}_j,\mathcal{B}\ni i} w_{j,\mathcal{B}} &= \frac{\sum_{\mathcal{B}\in\mathcal{E}_j,\mathcal{B}\ni i} \mathsf{e}^{-K_1\sum_{l\in\mathcal{N}(j)} m_{l,j}\mathbb{1}_{\mathcal{B}}(l)}}{\sum_{\mathcal{B}'\in\mathcal{E}_j} \mathsf{e}^{-K_1\sum_{l\in\mathcal{N}(j)} m_{l,j}\mathbb{1}_{\mathcal{B}'}(l)}} \\
&= \frac{1}{1+\frac{\sum_{e:x(e)=0}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}{\sum_{e:x(e)=1}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}} = \sum_{e:x(e)=1} g_{i,e},
\end{aligned}$$

where the second equality follows from (3.5). Also, for $\forall i \in \mathcal{I}\setminus N$, $k \in S$

$$\sum_{e:s'(e)=k} g_{i,e} = \frac{\sum_{e:s'(e)=k}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}}{\sum_{e\in\mathcal{T}_i}\mathsf{e}^{-K_2\left\{\Gamma_{i,e}-\overrightarrow{n}_{i-1,s(e_i)}+\overleftarrow{n}_{i,s'(e_i)}\right\}}} = \sum_{e:s(e)=k} g_{i+1,e},$$

where the second equality follows from (3.2) and (3.3). Thus, $\mathbf{w}$ and $\mathbf{g}$ are feasible in Problem-P.

Next, define the solution vector λ with

$$\lambda_{i,j} \triangleq \frac{1}{1+\mathsf{e}^{K_1(m_{i,j}-M_{i,j})}}$$

and

$$\epsilon \triangleq \max_{i\in\mathcal{I}} \max_{j,j'\in\mathcal{N}(i)} |\lambda_{i,j}-\lambda_{i,j'}|.$$

Denote the minimum value of Problem-PS by $\tilde{P}$. Then by the Lagrange duality we can upper bound $\tilde{P}_\epsilon - \tilde{P}$ with

$$\begin{aligned}
&\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} b_{i,e}\hat{\lambda}_{i,e} - \frac{1}{K_1}\sum_{j\in\mathcal{J}} H(\hat{w}_j) - \frac{1}{K_2}H(\hat{\lambda}_p) - \tilde{P} \\
&\le \sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} b_{i,e}\hat{\lambda}_{i,e} - \frac{1}{K_1}\sum_{j\in\mathcal{J}} H(\hat{w}_j) - \frac{1}{K_2}H(\hat{\lambda}_p) \\
&\quad + \frac{1}{K_1}\sum_{j\in\mathcal{J}} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} \\
&\overset{(a)}{\le} \frac{1}{K_1}\sum_{j\in\mathcal{J}}[H(w_j) - H(\hat{w}_j)] - \frac{1}{K_2}H(\hat{\lambda}_p) \\
&\quad + \epsilon\left(3\sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l}|b_{l,e}| + CN\right) \\
&\le \frac{1}{K_1}\sum_{j\in\mathcal{J}} H(w_j) + \epsilon N\left(\frac{3}{N}\sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l}|b_{l,e}| + C\right),
\end{aligned}$$

where (a) is given by rewriting (3.19) as

$$\begin{aligned}
&\frac{1}{K_1}\sum_{j\in\mathcal{J}} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} \\
&= \frac{1}{K_1}\sum_{j\in\mathcal{J}} H(w_j) - \sum_{j\in\mathcal{J}}\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} \sum_{l\in\mathcal{N}(j)} m_{l,j}\mathbb{1}_{\mathcal{B}}(l) \\
&\le \frac{1}{K_1}\sum_{j\in\mathcal{J}} H(w_j) - \sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l} b_{l,e}\hat{\lambda}_{l,e} + \epsilon\left(3\sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l}|b_{l,e}| + CN\right).
\end{aligned}$$

The last step of this equation follows from

$$\begin{aligned}
&\sum_{j\in\mathcal{J}}\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} \sum_{l\in\mathcal{N}(j)} m_{l,j}\mathbb{1}_{\mathcal{B}}(l) \\
&\quad= \sum_{l\in\mathcal{I}}\sum_{j\in\mathcal{N}(l)} m_{l,j}\lambda_l^j \\
&\quad\geq \sum_{l\in\mathcal{I}}\sum_{j\in\mathcal{N}(l)} m_{l,j}\left(\lambda_l-\epsilon\right) \\
&\quad\geq \sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l}\left(\delta_{x(e)=1}\sum_{j\in\mathcal{N}(l)} m_{l,j}\right)\lambda_{l,e} - \epsilon\sum_{l\in\mathcal{I}}\sum_{j\in\mathcal{N}(l)} |m_{l,j}| \\
&\quad\geq \sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l} b_{l,e}\lambda_{l,e} - \epsilon CN \\
&\quad\geq \sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l} b_{l,e}\hat{\lambda}_{l,e} - 3\epsilon\sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l} |b_{l,e}| - \epsilon CN.
\end{aligned}$$

In the above equation, the details of the last two inequalities are not included due to space limitations, but they can be derived using arguments very similar to [49, p. 4840-4841]. For any $\delta > 0$, after sufficiently many iterations with sufficiently large K_1 and K_2 until ϵ becomes sufficiently small, the modified Algorithm 1 outputs $\tilde{\mathbf{g}}_\epsilon$ with the minimum value $\tilde{P}_\epsilon$ which satisfies

$$\frac{\tilde{P}_\epsilon - \tilde{P}}{N} \leq \frac{\delta}{2}$$

by Lemma 4. By combining with Lemma 5, the final $\tilde{\mathbf{g}}_\epsilon$ satisfies the given relaxation bound as

$$\frac{\tilde{P}_\epsilon - P^*}{N} = \frac{\tilde{P}_\epsilon - \tilde{P}}{N} + \frac{\tilde{P} - P^*}{N} \leq \delta.$$

□

Lastly, we obtain the desired conclusion, which is stated as Theorem 7.

Theorem 7. *For any $\epsilon > 0$, sufficiently many iterations of the modified Algorithm 1* produces a feasible solution for Problem-DS that satisfies the KKT conditions within ϵ. If $\epsilon < 1/6$, then one can construct a solution $(\tilde{\mathbf{g}}_\epsilon, \tilde{\mathbf{w}}_\epsilon)$ for Problem-P that is feasible and whose value P_ϵ^* satisfies

$$0 \leq P_\epsilon^* - P^* \leq \delta N,$$

where

$$\delta \triangleq \frac{2\left(1 - R + \bar{\mathcal{N}}\right)\ln 2}{K_1} + \frac{\ln O}{K_2 N} + \epsilon\left(\frac{3}{N}\sum_{l\in\mathcal{I}}\sum_{e\in\mathcal{T}_l}|b_{l,e}| + C\right).$$

Proof. Combining results of Lemma 4, Lemma 5, and Lemma 6, we obtain the desired error bound. □

Remark 5. *The modified (i.e., cyclic schedule) Algorithm 1 is guaranteed to converge to a solution whose value can be made arbitrarily close to P^*. Therefore, the joint iterative LP decoder provides an approximate solution to Problem-P whose value is governed by the upper bound in Theorem 7. Algorithm 1 can be further modified to be of Gauss-Southwell type so that the complexity analysis in [49] can be extended to this case. Still, the analysis in [49], although a valid upper bound, does not capture the true complexity of decoding because one must choose $\delta = o\left(1/N\right)$ to guarantee that the iterative LP solver finds the true minimum. Therefore, the exact convergence rate and complexity analysis of Algorithm 1 is left for future study. In general, the convergence rate of coordinate-descent methods (e.g., Gauss-Seidel and Gauss-Southwell type algorithms) for convex problems without strict convexity is an open problem [58].*

C. Error Rate Prediction and Validation

In this section, we validate the proposed joint-decoding solution and discuss some implementation issues. Then, we present simulation results and compare with other approaches. In particular, we compare the performance of the joint LP decoder and joint iterative LP decoder with the joint iterative message-passing decoder on two finite-state intersymbol interference channels (FSISCs) described in Definition 6. For preliminary studies, we use a (3, 5)-regular binary LDPC code on the precoded dicode channel (pDIC) with length 155 and 455. For a more practical scenario, we also consider a (3, 27)-regular binary LDPC code with length 4923 and rate 8/9 on the class-II Partial Response (PR2) channel used as a partial-response target for perpendicular magnetic recording. All parity-check matrices were chosen randomly except that double-edges and four-cycles were avoided. Since the performance depends on the transmitted codeword, the WER results were obtained for a few chosen codewords of fixed weight. The weight was chosen to be roughly half the block length, giving weights 74, 226, and 2462 respectively.

The performance of the three algorithms was assessed based on the following implementation details.

Joint LP Decoder: Joint LP decoding is performed in the dual domain because this is much faster than the primal domain when using MATLAB. Due to the slow speed of LP solver, simulations were completed up to a WER of roughly 10^{-4} on the three different non-zero LDPC codes with block lengths 155 and 455 each. To extrapolate the error rates to high SNR (well beyond the limits of our simulation), we use a simulation-based semi-analytic method based on the truncated union bound, (2.3), as discussed in Chapter II. The idea is to run a simulation at low SNR and keep track of all observed codeword and pseudo-codeword (PCW) errors and a truncated union bound is computed by summing over all observed errors. The truncated union bound is obtained by computing the generalized Euclidean distances associated with all decoding errors that occurred at some low SNR points (e.g., WER of roughly than 10^{-1}) until we observe a stationary generalized Euclidean distance spectrum. It is quite easy, in fact, to store these error events in a list which is finally pruned to avoid overcounting. Of course, low SNR allows the decoder to discover PCWs more rapidly than high SNR and it is well-known that the truncated bound should give a good estimate at high SNR if all dominant joint decoding PCWs have been found (e.g., [59, 60]). One nontrivial open question is the feasibility and effectiveness of enumerating error events for long codes. In particular, we do not address how many instances must be simulated to have high confidence that all the important error events are found so there are no surprises at high SNR.

Joint Iterative LP Decoder: Joint iterative decoding is performed based on the Algorithm 1 on all three LDPC codes of different lengths. For block lengths 155 and 455, we chose the codeword which shows the worst performance for the joint LP decoder experiments. We used a simple scheduling update scheme: variables are updated according to Algorithm 1 with cyclically with $\ell_{\text{inner}} = 2$ inner loop iterations for each outer iteration. The maximum number of outer iterations is $\ell_{\text{outer}} = 100$, so the total iteration count, $\ell_{\text{outer}}\ell_{\text{inner}}$, is at most 200. The choice of parameters are $K_1 = 1000$ and $K_2 = 100$ on the LDPC codes with block lengths 155 and 455. For the LDPC code with length 4923, K_2 is reduced to 10. To prevent possible underflow or overflow, a few expressions must be implemented carefully. When

$$K_1 \min_{r \in \mathcal{N}(j)\setminus i} |m_{r,j}| \geq 35,$$

a well-behaved approximation of (3.10) is given by

$$\begin{aligned}&\left[\frac{1}{K_1}\ln\left\{\left(2+2\sum_{r\in\mathcal{N}(j)\setminus i}\mathsf{e}^{-K_1\left(|m_{r,j}|-\min_{r\in\mathcal{N}(j)\setminus i}|m_{r,j}|\right)}\right)\right\}\right.\\&\qquad\qquad\qquad\qquad\left.-\min_{r\in\mathcal{N}(j)\setminus i}|m_{r,j}|\right]\left(\prod_{r\in\mathcal{N}(j)\setminus i}\operatorname{sgn}\left(m_{r,j}\right)\right),\end{aligned}$$

where $\operatorname{sgn}(x)$ is the usual sign function. Also, (3.9) should be implemented as

$$\begin{aligned}&\max_{e\in\mathcal{T}_i:x(e)=0}\left\{\bar{\alpha}_{i-1}\left(s(e)\right)+\bar{\lambda}_{i,e}+\bar{\beta}_i\left(s'(e)\right)\right\}\\&\qquad-\max_{e\in\mathcal{T}_i:x(e)=1}\left\{\bar{\alpha}_{i-1}\left(s(e)\right)+\bar{\lambda}_{i,e}+\bar{\beta}_i\left(s'(e)\right)\right\}\\&\qquad+\log\left[\sum_{e\in\mathcal{T}_i:x(e)=0}\bar{\alpha}_{i-1}\left(s(e)\right)+\bar{\lambda}_{i,e}+\bar{\beta}_i\left(s'(e)\right)-\right.\\&\qquad\qquad\left.\max_{e\in\mathcal{T}_i:x(e)=0}\left\{\bar{\alpha}_{i-1}\left(s(e)\right)+\bar{\lambda}_{i,e}+\bar{\beta}_i\left(s'(e)\right)\right\}\right]\\&\qquad-\log\left[\sum_{e\in\mathcal{T}_i:x(e)=1}\bar{\alpha}_{i-1}\left(s(e)\right)+\bar{\lambda}_{i,e}+\bar{\beta}_i\left(s'(e)\right)-\right.\\&\qquad\qquad\qquad\left.\max_{e\in\mathcal{T}_i:x(e)=1}\left\{\bar{\alpha}_{i-1}\left(s(e)\right)+\bar{\lambda}_{i,e}+\bar{\beta}_i\left(s'(e)\right)\right\}\right],\end{aligned}$$

where $\bar{\alpha}_i\left(k\right)\triangleq\ln\alpha_i\left(k\right)$, $\bar{\beta}_i\left(k\right)\triangleq\ln\beta_i\left(k\right)$ and $\bar{\lambda}_{i,e}\triangleq\ln\lambda_{i,e}$.

Joint Iterative Message-Passing Decoder: Joint iterative message decoding is performed based on the state-based algorithm described in [55] on all three LDPC codes of different lengths. To make a fair comparison with the Joint Iterative LP Decoder, the same maximum iteration count is used and the same codewords are transmitted.

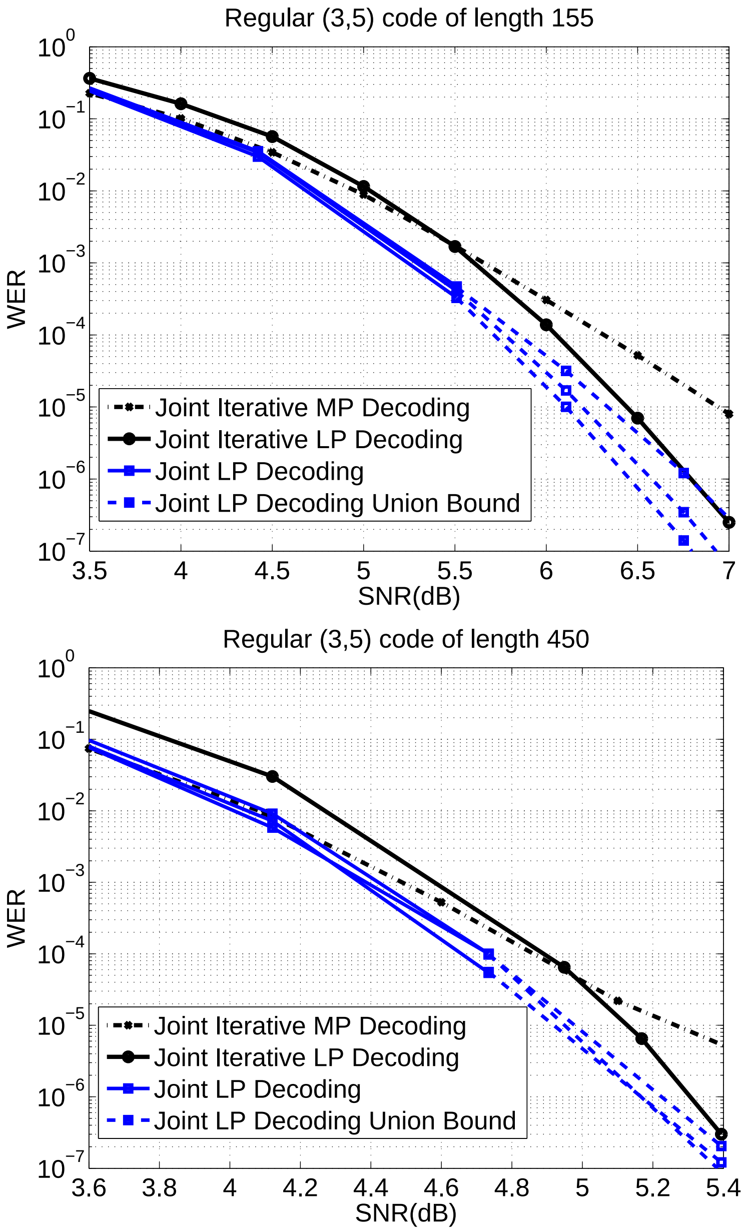


Fig. 13. Comparison between joint LP decoding, joint iterative LP decoding, and joint iterative message-passing (MP) decoding on the pDIC with AWGN for random (3,5) regular LDPC codes of length $N = 155$ (top) and $N = 450$ (bottom). The joint LP decoding experiments were repeated for three different non-zero codewords and depicted in three different curves. The dashed curves are computed using the union bound in Equation (2.3) based on JD-PCWs observed at 3.46 dB (left) 2.67 dB (right). Note that SNR is defined as channel output power divided by σ^2.

1. Results

Fig. 13 compares the results of all three decoders and the error-rate estimate given by the union bound method discussed in Chapter II. The solid lines represent the simulation curves while the dashed lines represent a truncated union bound for three different non-zero codewords. Surprisingly, we find that joint LP decoder outperforms joint iterative message passing decoder by about 0.5 dB at WER of 10^{-4}. We also observe that that joint iterative LP decoder loses about 0.1 dB at low SNR. This may be caused by using finite values for K_1 and K_2. At high SNR, however, this gap disappears and the curve converges towards the error rate predicted for joint LP decoding. This shows that joint LP decoding outperforms belief-propagation decoding for short length code at moderate SNR with the predictability of LP decoding. Of course, this can be achieved with a computational complexity similar to turbo equalization.

One complication that must be discussed is the dependence on the transmitted codeword. Computing the bound is complicated by the fact that the loss of channel symmetry implies that the dominant PCWs may depend on the transmitted sequence. It is known that long LDPC codes with joint iterative decoding experience a concentration phenomenon [55] whereby the error probability of a randomly chosen codeword is very close, with high probability, to the average error probability over all codewords. This effect starts to appear even at the short block lengths used in this example. More research is required to understand this effect at moderate block lengths and to verify the same effect for joint LP decoding.

Fig. 14 compares the joint iterative LP decoder and joint iterative message-passing decoder in a practical scenario. Again, we find that the joint iterative LP decoder provides gains over the joint iterative message-passing decoder at high SNR. The slope difference between the curves also suggests that the performance gains of

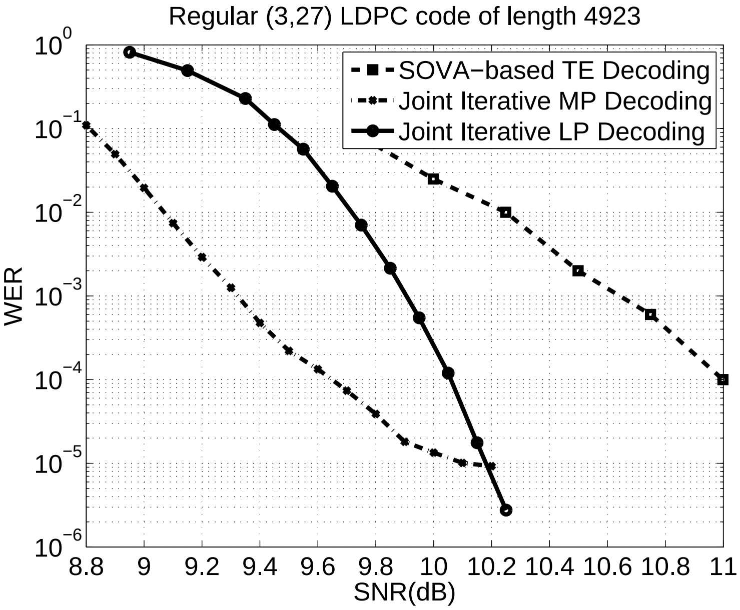


Fig. 14. Comparison between joint iterative LP decoding, joint iterative MP decoding and soft-output Viterbi algorithm (SOVA)-based TE decoding (taken from [38]) on the PR2 channel with AWGN for random (3,27) regular LDPC codes of length $N = 4923$. Note that SNR is defined as channel output power divided by σ^2.

joint iterative LP decoder will increase with SNR. This shows that joint iterative LP decoding can provide performance gains at high SNR with a computational complexity similar to that of turbo equalization.

CHAPTER IV

SIMPLIFIED ITERATIVE LP DECODING FOR BINARY PROBLEMS

A. Motivation

In Chapter III, smoothed Lagrangian relaxation methods were shown to greatly reduce the computational complexity of the joint LP solver and combine the predictability of LP decoding with a computational complexity similar to turbo equalization (TE). In addition, they provide provable convergence guarantees with performance gains over TE in the error-floor region. Inspired by these gains, we reconsider the problem of iterative linear-programming (LP) decoding for low-density parity-check (LDPC) codes and develop computationally simplified solutions.

The main idea is applying block-coordinate maximization algorithms directly to the dual problem. This direct approach is similar in spirit to [61, 62]. But, our results and methods are customized for decoding problems [45, 49, 63]. The primary result is a simplified joint iterative LP decoder with SOVA-based channel update and min-sum-like code update. Moreover, it has no smoothing parameters to tune. We anticipate this approach will lead eventually to a compact analysis of a bound on the iteration complexity (or convergence rate).

B. Derivation of Simplified Iterative Solver for the LP Decoder

First, we derive a a block coordinate ascent algorithm that cyclically, for $p = 1, \ldots, n$, maximizes over the block $\mathbf{m}_p \triangleq \{m_{p,j}\}_{j \in \mathcal{N}(p)}$ of dual-domain variables for the LP decoder. For fixed $p \in \mathcal{I}$, one can separate the objective function of the dual problem

Algorithm 2 Simplified Iterative Linear-Programming Decoding

- Step 1. Set $\ell = 0$ and initialize $m_{i,j} = \frac{\gamma_i}{|\mathcal{N}(i)|}$ for $i \in \mathcal{I}$, $j \in \mathcal{N}(i)$.
- Step 2. For $i \in \mathcal{I}$,
  - (i) Compute bit-to-check msg $m_{i,j}$ for $j \in \mathcal{N}(i)$

    $$m_{i,j} = \frac{\gamma_i + \sum_{j' \in \mathcal{N}(i)} M_{i,j'}}{|\mathcal{N}(i)|} - M_{i,j}$$

  - (ii) Compute check-to-bit msg $M_{i,j}$ for $j \in \mathcal{N}(i)$

    $$M_{i,j} = \left(\prod_{i \in \mathcal{N}(j) \backslash p} \operatorname{sgn}(m_{i,j}) \right) \min_{i' \in \mathcal{N}(j) \backslash p} |m_{i',j}| \tag{4.1}$$

- Step 3. Compute hard decisions and stopping rule
  - (i) For $i \in \mathcal{I}$,

    $$\hat{f}_i = \left(1 - \operatorname{sgn}\left(\gamma_i + \sum_{j' \in \mathcal{N}(i)} M_{i,j'} \right) \right) / 2$$

  - (ii) If $\hat{\mathbf{f}}$ satisfies all parity checks or the maximum iteration number, $\ell_{\max}$, is reached, stop and output $\hat{\mathbf{f}}$. Otherwise increment ℓ and go to Step 2.

Table VI. Dual Problem for the LP Decoder (Problem-D)

$$\max_{\mathbf{m}} \sum_{j \in \mathcal{J}} \min_{\mathcal{B} \in \mathcal{E}_j} \left[\sum_{i \in \mathcal{B}} m_{i,j} \right]$$

subject to

$$\sum_{j \in \mathcal{N}(i)} m_{i,j} = \gamma_i, \; \forall i \in \mathcal{I}$$

(i.e., Problem-D in Table VI) into two parts with

$$\begin{aligned}
\max_{\mathbf{m}_p} g\left(\mathbf{m}\right) &\triangleq \max_{\mathbf{m}_p} g\left(\mathbf{m}_1, \ldots, \mathbf{m}_N\right) \\
&= \max_{\mathbf{m}_p} \sum_{j\in\mathcal{J}} \min_{\mathcal{B}\in\mathcal{E}_j} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j} \mathbb{1}_{\mathcal{B}}(i) \right\} \\
&= \sum_{j\in\mathcal{J}\setminus\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_j} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j} \mathbb{1}_{\mathcal{B}}(i) \right\} + \max_{\mathbf{m}_p} \sum_{j\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_j} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j} \mathbb{1}_{\mathcal{B}}(i) \right\}.
\end{aligned}$$

Since the first term does not depend on $\mathbf{m}_p$, we can focus on the second term. Using the simple equality

$$\sum_{i\in\mathcal{N}(j)} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) = m_{p,j}^{(t)} \mathbb{1}_{\mathcal{B}}(p) + \sum_{i\in\mathcal{N}(j)\setminus p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i),$$

one can rewrite the second term as

$$\begin{aligned}
&\max_{\mathbf{m}_p} \sum_{j\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_j} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \\
&= \max_{\mathbf{m}_p} \sum_{j\in\mathcal{N}(p)} \min \left[\min_{\mathcal{B}\in\mathcal{E}_j : p\in\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \min_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right] \\
&= \max_{\mathbf{m}_p} \sum_{j\in\mathcal{N}(p)} \min \left[m_{p,j}^{(t)} + \min_{\mathcal{B}\in\mathcal{E}_j, p\in\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)\setminus p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \min_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)\setminus p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right].
\end{aligned}$$

Because the sum of the minimums is less than the minimum of the sum, one has the upper bound

$$\begin{aligned}
\max_{\mathbf{m}_p} \min \Bigg[&\sum_{j\in\mathcal{N}(p)} \left\{ m_{p,j}^{(t)} + \min_{\mathcal{B}\in\mathcal{E}_j, p\in\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)\setminus p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right\}, \\
&\sum_{j\in\mathcal{N}(p)} \left\{ \min_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)\setminus p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right\} \Bigg].
\end{aligned}$$

From the constraint on $\sum_{j \in \mathcal{N}(p)} m_{p,j}$ and the fact that the inner expression no longer depends on $\mathbf{m}_p$, we obtain

$$\min \left[\gamma_p + \sum_{j \in \mathcal{N}(p)} \min_{\mathcal{B} \in \mathcal{E}_j, p \in \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j) \backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \right.$$
$$\left. \sum_{j \in \mathcal{N}(p)} \min_{\mathcal{B} \in \mathcal{E}_j, p \notin \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j) \backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right],$$

Proposition 2. Maximizing $g\left(\mathbf{m}\right)$ over $\mathbf{m}_p$ yields the following ascent step

$$M_{p,j}^{(t+1)} \triangleq \min_{\mathcal{B} \in \mathcal{E}_j, p \in \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j) \backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} - \min_{\mathcal{B} \in \mathcal{E}_j, p \notin \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j) \backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}$$

and

$$m_{p,j}^{(t+1)} \triangleq \frac{\gamma_p + \sum_{j' \in \mathcal{N}(p)} M_{p,j'}^{(t+1)}}{|\mathcal{N}(p)|} - M_{p,j}^{(t+1)}.$$

Proof. First, we verify that this choice satisfies the constraint with equality by observing that

$$\sum_{j \in \mathcal{N}(p)} m_{p,j}^{(t+1)} = \sum_{j \in \mathcal{N}(p)} \left[\frac{\gamma_p + \sum_{j' \in \mathcal{N}(p)} M_{p,j'}^{(t+1)}}{|\mathcal{N}(p)|} - M_{p,j}^{(t+1)} \right] = \gamma_p.$$

To see that it also achieves the upper bound with equality, we compute

$$\sum_{j \in \mathcal{N}(p)} \min_{\mathcal{B} \in \mathcal{E}_j} \left\{ \sum_{i \in \mathcal{N}(j)} m_{i,j}^{(t+1)} \mathbb{1}_{\mathcal{B}}(i) \right\}$$
$$= \sum_{j \in \mathcal{N}(p)} \min \left[\min_{\mathcal{B} \in \mathcal{E}_j : p \in \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j)} m_{i,j}^{(t+1)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \min_{\mathcal{B} \in \mathcal{E}_{j'}, p \notin \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j)} m_{i,j}^{(t+1)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right]$$
$$= \sum_{j \in \mathcal{N}(p)} \min \left[m_{p,j}^{(t+1)} + \min_{\mathcal{B} \in \mathcal{E}_j, p \in \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j) \backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \right.$$
$$\left. \min_{\mathcal{B} \in \mathcal{E}_j, p \notin \mathcal{B}} \left\{ \sum_{i \in \mathcal{N}(j) \backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right],$$

Using the definition of $M_{p,j}^{(t+1)}$, one can rewrite this as

$$\begin{aligned}
&\sum_{j\in\mathcal{N}(p)} \min\left[m_{p,j}^{(t+1)} + M_{p,j}^{(t+1)}, 0\right] + \sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \\
&= \sum_{j\in\mathcal{N}(p)} \min\left[m_{p,j}^{(t+1)} + M_{p,j}^{(t+1)} + \frac{\sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}}{|\mathcal{N}(p)|}, \right. \\
&\qquad\qquad\qquad \left. \frac{\sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}}{|\mathcal{N}(p)|} \right] \\
&= \frac{1}{|\mathcal{N}(p)|} \sum_{j\in\mathcal{N}(p)} \min\left[\gamma_p + \sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\in\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \right. \\
&\qquad\qquad\qquad \left. \sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right] \\
&= \min\left[\gamma_p + \sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\in\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\}, \right. \\
&\qquad\qquad \left. \sum_{j'\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_{j'}, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j')\backslash p} m_{i,j'}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \right].
\end{aligned}$$

□

Proposition 3. The quantity $M_{p,j}^{(t+1)}$ can be computed efficiently using

$$M_{p,j}^{(t+1)} \quad = \quad \left(\prod_{i\in\mathcal{N}(j)\backslash p} \operatorname{sgn}\left(m_{i,j}^{(t)}\right) \right) \min_{i'\in\mathcal{N}(j)\backslash p} |m_{i',j}^{(t)}|.$$

Proof. From [45], we have

$$\min\left(x_1,\, x_2,\, \ldots\,,\, x_m\right) = \lim_{K\to\infty} -\frac{1}{K} \ln \sum_{i=1}^{m} \mathsf{e}^{-Kx_i}.$$

Therefore, we can write $M_{p,j}^{(t+1)}$ as

$$\begin{aligned} M_{p,j}^{(t+1)} &= \min_{\mathcal{B}\in\mathcal{E}_j, p\in\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)\backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} - \min_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \left\{ \sum_{i\in\mathcal{N}(j)\backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i) \right\} \\ &= \lim_{K\to\infty} -\frac{1}{K} \ln \frac{\sum_{\mathcal{B}\in\mathcal{E}_j, p\in\mathcal{B}} \mathsf{e}^{-K\sum_{i\in\mathcal{N}(j)\backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i)}}{\sum_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \mathsf{e}^{-K\sum_{i\in\mathcal{N}(j)\backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i)}}. \end{aligned}$$

Next, we rewrite $\ln \sum_{\mathcal{B}\in\mathcal{E}_j, p\in\mathcal{B}} \mathsf{e}^{-K\sum_{i\in\mathcal{N}(j)\backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i)} / \sum_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \mathsf{e}^{-K\sum_{i\in\mathcal{N}(j)\backslash p} m_{i,j}^{(t)} \mathbb{1}_{\mathcal{B}}(i)}$ as

$$\begin{aligned} &\ln \frac{\sum_{\mathcal{B}\in\mathcal{E}_j, p\in\mathcal{B}} \prod_{i\in\mathcal{N}(j)\backslash p} x_i^{\mathbb{1}_{\mathcal{B}}(i)}}{\sum_{\mathcal{B}\in\mathcal{E}_j, p\notin\mathcal{B}} \prod_{i\in\mathcal{N}(j)\backslash p} x_i^{\mathbb{1}_{\mathcal{B}}(i)}} \\ &= \ln \frac{\prod_{i\in\mathcal{N}(j)\backslash p} (1+x_{i,j}) + \prod_{i\in\mathcal{N}(j)\backslash p} (1-x_{i,j})}{\prod_{i\in\mathcal{N}(j)\backslash p} (1+x_{i,j}) - \prod_{i\in\mathcal{N}(j)\backslash p} (1-x_{i,j})} \\ &= \ln \frac{1+\prod_{i\in\mathcal{N}(j)\backslash p} \left(\frac{1-x_{i,j}}{1+x_{i,j}}\right)}{1-\prod_{i\in\mathcal{N}(j)\backslash p} \left(\frac{1-x_{i,j}}{1+x_{i,j}}\right)}, \end{aligned}$$

where $x_i \triangleq \mathsf{e}^{-Km_{i,j}^{(t)}}$. Equivalently, we obtain

$$\begin{aligned} &\ln \frac{1+\prod_{i\in\mathcal{N}(j)\backslash p} \tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)}{1-\prod_{i\in\mathcal{N}(j)\backslash p} \tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)} \\ &\overset{(a)}{=} 2\tanh^{-1}\left(\prod_{i\in\mathcal{N}(j)\backslash p} \tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)\right) \\ &\overset{(b)}{=} 2\,\mathrm{sgn}\left(\prod_{i\in\mathcal{N}(j)\backslash p} \tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)\right) \tanh^{-1}\left(\left| \prod_{i'\in\mathcal{N}(j)\backslash p} \tanh\left(\frac{Km_{i',j}^{(t)}}{2}\right)\right| \right) \\ &\overset{(c)}{=} \left(\prod_{i\in\mathcal{N}(j)\backslash p} \mathrm{sgn}\left(m_{i,j}^{(t)}\right)\right) \left[2\tanh^{-1}\left(\prod_{i'\in\mathcal{N}(j)\backslash p} \tanh\left(\frac{K\left|m_{i',j}^{(t)}\right|}{2}\right)\right)\right], \end{aligned}$$

where (a) follows from $\tanh^{-1}(x) = (1/2)\ln((1+x)/(1-x))$, (b) and (c) follows from the fact that $\tanh(x)$ and $\tanh^{-1}(x)$ are monotonically increasing and have odd symmetry, implying

$$\tanh^{-1}(x) = \operatorname{sgn}(x)\tanh^{-1}(|x|),$$

$$\begin{aligned}\operatorname{sgn}\left(\prod_{i\in\mathcal{N}(j)\backslash p}\tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)\right) &= \prod_{i\in\mathcal{N}(j)\backslash p}\operatorname{sgn}\left(\tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)\right)\\ &= \prod_{i\in\mathcal{N}(j)\backslash p}\operatorname{sgn}\left(\frac{Km_{i,j}^{(t)}}{2}\right) = \prod_{i\in\mathcal{N}(j)\backslash p}\operatorname{sgn}\left(m_{i,j}^{(t)}\right)\end{aligned}$$

and

$$\left|\prod_{i\in\mathcal{N}(j)\backslash p}\tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)\right| = \prod_{i\in\mathcal{N}(j)\backslash p}\left|\tanh\left(\frac{Km_{i,j}^{(t)}}{2}\right)\right| = \prod_{i\in\mathcal{N}(j)\backslash p}\tanh\left(\frac{K\left|m_{i,j}^{(t)}\right|}{2}\right).$$

Finally, one gets

$$\begin{aligned}M_{p,j}^{(t+1)} &= -\left(\prod_{i\in\mathcal{N}(j)\backslash p}\operatorname{sgn}\left(m_{i,j}^{(t)}\right)\right)\lim_{K\to\infty}\left[\frac{2}{K}\tanh^{-1}\left(\prod_{i'\in\mathcal{N}(j)\backslash p}\tanh\left(\frac{K\left|m_{i',j}^{(t)}\right|}{2}\right)\right)\right]\\ &\overset{(a)}{=} \left(\prod_{i\in\mathcal{N}(j)\backslash p}\operatorname{sgn}\left(m_{i,j}^{(t)}\right)\right)\min_{i'\in\mathcal{N}(j)\backslash p}|m_{i',j}^{(t)}|,\end{aligned}$$

where (a) holds because

$$\begin{aligned}&\frac{2}{K}\tanh^{-1}\left(\prod_{i'\in\mathcal{N}(j)\backslash p}\tanh\left(\frac{K\left|m_{i',j}^{(t)}\right|}{2}\right)\right)\\ &= \frac{1}{K}\ln\frac{1+\prod_{i'\in\mathcal{N}(j)\backslash p}\mathsf{e}^{K\left|m_{i',j}^{(t)}\right|}}{\sum_{i'\in\mathcal{N}(j)\backslash p}\mathsf{e}^{K\left|m_{i',j}^{(t)}\right|}}\\ &= -\min_{i'\in\mathcal{N}(j)\backslash p}|m_{i',j}^{(t)}| + \frac{1}{K}\ln\frac{\mathsf{e}^{\ln\left(1+\prod_{i'\in\mathcal{N}(j)\backslash p}\mathsf{e}^{K\left|m_{i',j}^{(t)}\right|}\right)}}{\sum_{i'\in\mathcal{N}(j)\backslash p}\mathsf{e}^{K\left(\left|m_{i',j}^{(t)}\right|-\min_{i'\in\mathcal{N}(j)\backslash p}|m_{i',j}^{(t)}|\right)}}.\end{aligned}$$

□

The resulting algorithm is summarized in Algorithm 2.

C. Derivation of Simplified Iterative Solver for the Joint LP Decoder

Minimizing Problem-D2 (in Chapter III) over the p-th block gives

$$\max_{\mathbf{m}_p} \sum_{j\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_j} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i) \right\}$$

subject to

$$\Gamma_{p,e} = \overrightarrow{n}_{p-1,s(e)} - \overleftarrow{n}_{p,s'(e)},\ e \in \mathcal{T}_p$$

where $\overrightarrow{n}_{i,k}$ is defined for $i = 1, \ldots, p-1$ by

$$-\overrightarrow{n}_{i,k} = \min_{e\in s'^{-1}(k)} -\overrightarrow{n}_{i-1,s(e_i)} + \Gamma_{i,e},\ \forall k \in \mathcal{S}$$

and $\overleftarrow{n}_{i,k}$ is defined for $i = N-1, N-2, \ldots, p$ by

$$\overleftarrow{n}_{i,k} = \min_{e\in s^{-1}(k)} \overleftarrow{n}_{i+1,s'(e_{i+1})} + \Gamma_{i+1,e},\ \forall k \in \mathcal{S}$$

starting from

$$\overrightarrow{n}_{0,k} = \overleftarrow{n}_{N,k} = 0,\ \forall k \in \mathcal{S}.$$

By defining

$$\gamma_p \triangleq \frac{\sum_{e\in\mathcal{T}_p}\left(b_{p,e} - \overrightarrow{n}_{p-1,s(e)} + \overleftarrow{n}_{p,s'(e)}\right)}{|\left\{e \in \mathcal{T}_p \,|x(e) = 1\right\}|}$$

and

$$\Gamma_{p,e} = b_{i,e} - \delta_{x(e)=1} \sum_{j\in\mathcal{N}(i)} m_{i,j},$$

this problem can be written equivalently as

$$\max_{\mathbf{m}_p} \sum_{j\in\mathcal{N}(p)} \min_{\mathcal{B}\in\mathcal{E}_j} \left\{ \sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i) \right\}$$

Algorithm 3 Simplified Iterative Joint Linear-Programming Decoding

- Step 1. Set $\ell = 0$ and initialize $m_{i,j} = 0$ for $i \in \mathcal{I}$, $j \in \mathcal{N}\left(i\right)$.
- Step 2. Update Outer Loop: For $i \in \mathcal{I}$,
  - (i) Compute bit-to-trellis message

    $$\Gamma_{i,e} = b_{i,e} - \delta_{x(e)=1} \sum_{j \in \mathcal{N}(i)} m_{i,j}$$

  - (ii) Compute forward/backward trellis messages

    $$-\overrightarrow{n}_{i+1,k} = \min_{e \in s'^{-1}(k)} -\overrightarrow{n}_{i,s(e_i)} + \Gamma_{i+1,e} \qquad (4.2)$$

    $$\overleftarrow{n}_{i-1,k} = \min_{e \in s^{-1}(k)} \overleftarrow{n}_{i,s'(e_{i+1})} + \Gamma_{i,e}, \qquad (4.3)$$

    where $\overrightarrow{n}_{0,k} = \overleftarrow{n}_{N,k} = 0$ for all $k \in \mathcal{S}$.
- Step 3. Update Inner Loop for ℓ_{inner} rounds: For $i \in \mathcal{I}$,
  - (i) Compute trellis-to-bit message γ_i

    $$\gamma_i \;\; = \;\; \frac{\sum_{e \in \mathcal{T}_i} \left(b_{i,e} - \overrightarrow{n}_{i-1,s(e)} + \overleftarrow{n}_{i,s'(e)}\right)}{\left|\left\{e \in \mathcal{T}_i \,|x(e) = 1\right\}\right|}$$

  - (ii) Compute bit-to-check msg $m_{i,j}$ for $j \in \mathcal{N}\left(i\right)$

    $$m_{i,j} = \frac{\gamma_i + \sum_{j' \in \mathcal{N}(i)} M_{i,j'}}{|\mathcal{N}(i)|} - M_{i,j}$$

  - (iii) Compute check-to-bit msg $M_{i,j}$ for $j \in \mathcal{N}\left(i\right)$

    $$M_{i,j} = \left(\prod_{i \in \mathcal{N}(j) \backslash p} \operatorname{sgn}\left(m_{i,j}\right)\right) \min_{i' \in \mathcal{N}(j) \backslash p} |m_{i',j}| \qquad (4.4)$$

- Step 4. Compute hard decisions and stopping rule
  - (i) For $i \in \mathcal{I}$,

    $$\hat{f}_i = \left(1 - \operatorname{sgn}\left(\gamma_i\right)\right)/2$$

  - (ii) If $\hat{\mathbf{f}}$ satisfies all parity checks or the maximum outer iteration number, ℓ_{outer}, is reached, stop and output $\hat{\mathbf{f}}$. Otherwise increment ℓ and go to Step 2.

subject to

$$\sum_{j \in \mathcal{N}(p)} m_{p,j} = \gamma_p.$$

To obtain a simple and efficient algorithm, we again use a block coordinate ascent strategy to this problem by following the same argument in Section B. The resulting algorithm is summarized in Algorithm 3.

CHAPTER V

IMP: A MESSAGE-PASSING ALGORITHM FOR MATRIX COMPLETION*

This chapter* considers an important subclass of the matrix completion problem where the entries (drawn from a finite alphabet) are modeled by a (generative) factor graph. Based on this factor graph model, we propose a message-passing (MP) based algorithm, termed IMP, to estimate missing entries. This algorithm seems to share some of the desirable properties demonstrated by MP in its successful application to modern coding theory [64]. The IMP algorithm tries to combine the benefits of soft clustering of users/movies into groups and message-passing based on the unknown groups to make predictions. In addition, simulation results for cold-start settings (i.e., less than 0.5% randomly sampled entries) show that the cold start problem is reduced greatly by IMP in comparison to other methods on real collaborative filtering (or Netflix) data matrices.

A. Factor Graph Model

Consider a collection of N users and M movies when the set O of user-movie pairs have been observed. The main theoretical question is, “How large should the size of O be to estimate the unknown ratings within some distortion δ?”. Answers to this question certainly require some assumptions about the movie rating process. So we begin differently by introducing a probabilistic model for the movie ratings. The basic idea is that *hidden* variables are introduced for users and movies, and that the movie

*This chapter is in part a reprint of the material in the paper: B.-H.Kim, A. Yedla, and H. Pfister, "IMP: A message-passing algorithm for matrix completion",in *Proc. Int. Symp. on Turbo Codes & Iterative Inform. Proc.*, Brest, France,Sept. 2010, pp. 469-473.

ratings are conditionally independent given these hidden variables. It is convenient to think of the hidden variable for any user (or movie) as the *user group* (or *movie group*) of that user (or movie) and this can be viewed as a simplistic assumption about the psychological nature of movie preferences [65, 66]. In this context, the rating associated with a user-movie pair depends only on the user group and the movie group. [67]

Since the number of movie groups are very small compared to the number of movies, this idea is similar to mapping movies to a low-dimensional movie group. Each movie group may correspond to a genre (e.g., comedy, drama, action, ...). Each user group tries to capture sets of users that have similar taste in movies. For example, a movie may be classified as a comedy, and a user may be classified as a comedy lover. The model may use 20 to 40 such groups to locate each movie and user in a multidimensional space. It then predicts a user's rating of a movie according to the movie's rating on the dimensions that person cares about most since similar user/movie map to similar groups in the low-dimensional (group) space.

The goal is to design a probabilistic mapping such that reflects group associations in the low-dimensional (group) space. Let there be g_u user groups, g_v movie groups, and define $[k] \triangleq \{1, 2, \ldots, k\}$. The user group of the n-th user, $U_n \in [g_u]$, is a discrete random variable drawn from $\Pr(U_n = u) \triangleq p_U(u)$ and $\mathbf{U} = U_1, U_2, \ldots, U_N$ is the user group vector. Likewise, the movie group of the m-th movie, $V_m \in [g_v]$, is a discrete random variable drawn from $\Pr(V_m = v) \triangleq p_V(v)$ and $\mathbf{V} = V_1, V_2, \ldots, V_M$ is the movie group vector. Then, the rating of the m-th movie by the n-th user is a discrete random variable $R_{nm} \in \mathcal{R}$ (e.g., Netflix uses $\mathcal{R} = [5]$) drawn from $\Pr(R_{nm} = r | U_n = u, V_m = v) \triangleq w(r|u, v)$ and the rating R_{nm} is *conditionally independent* given the user group U_n and the movie group V_m. Let $\mathbf{R}$ denote the rating matrix and the observed submatrix be $\mathbf{R}_O$ with $O \subseteq [N] \times [M]$. In this setup, some of the entries in the rating

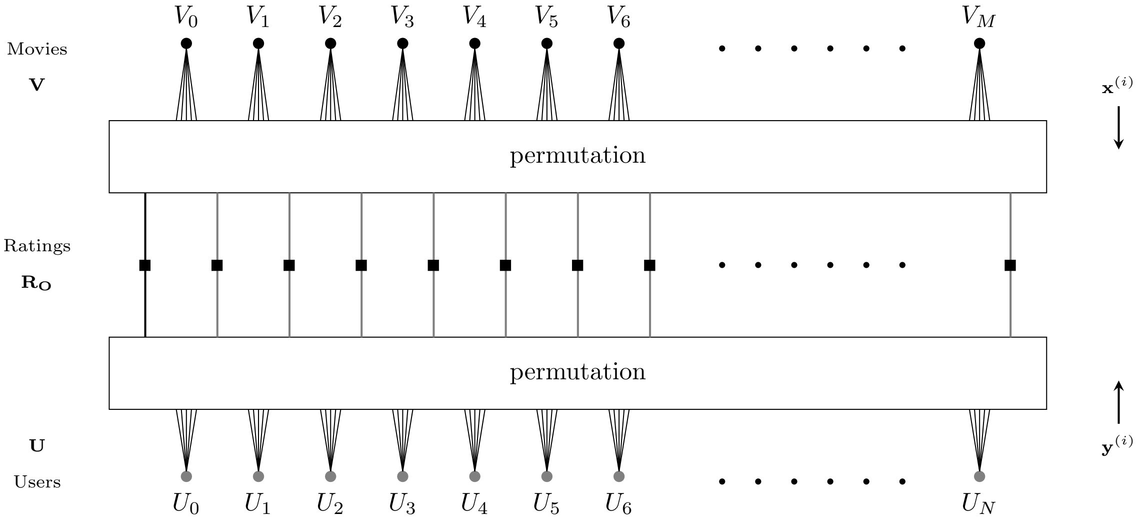


Fig. 15. The factor graph model for the matrix completion problem. The graph is sparse when there are few ratings. Edges represent random variables and nodes represent local probabilities. The node probability associated with the ratings implies that each rating depends only on the movie group (top edge) and the user group (bottom edge). Synthetic data can be generated by picking i.i.d. random user/movie groups and then using random permutations to associate groups with ratings. Note $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ are the messages from movie to user and user to movie during iteration i for the Algorithm 4.

matrix are observed while others must be predicted. The conditional independence assumption in the model implies that

$$\Pr\left(\mathbf{R}_O|\mathbf{U},\mathbf{V}\right) \triangleq \prod_{(n,m)\in O} w\left(R_{nm}|U_n,V_m\right).$$

Specifically, we consider the factor graph (composed of 3 layers, see Fig. 15) as a randomly chosen instance of this problem based on this probabilistic model. The key assumptions are that these layers separate the influence of user groups, movie groups, and observed ratings. A random permutation is used to map the edges attached to user nodes to the edges attached to movie nodes.

This model attempts to exploit correlation in the ratings based on similarity between users (and movies). It also tries to include the noisy rating process in the model and reduce the impact of corrupted ratings on prediction by dimension reduction. These advantages allows one to approximates real Netflix data generation process more closely than other simpler factor models. In fact this model can be seen as a generalization of [29] and [32]. It is also important to note that this is a *probabilistic generative model* which generalizes the clustering model in and also allows one to evaluate different learning algorithms on synthetic data and compare the results with theoretical bounds (see Section F for details).

B. The IMP Algorithm

1. Initializing $w(r|u,v)$ for Group Ratings

The IMP algorithm requires reasonable initial estimates, of the observation model $w(r|u,v)$, to get started. To get these estimates, we cluster users (and movies) first. The basic method uses a variable-dimension vector quantization (VDVQ) clustering algorithm and the standard codebook splitting approach known as the generalized Lloyd algorithm (GLA) to generate codebooks whose size is any power of 2 [68]. Though our approach was motivated by the VDVQ clustering algorithm, it turns out to be equivalent to soft K-means clustering with an appropriate distance measure. So we will refer VDVQ clustering as soft K-means clustering.

The soft K-means clustering algorithm is based on the alternating minimization of the average distance between users (or movies) and codebooks (that contain no missing data). This leads to alternating application of *nearest neighbor* and *centroid* rules. The distance is computed only on the elements both vectors share. In the case of users, one can think of this Algorithm 5 as a "K-critics" algorithms which tries to design K critics (i.e., people who have seen every movie) that cover the space of all user tastes and each user is given a soft "degree of assignment (or soft group

Algorithm 4 IMP Algorithm

Step I: Initialization of $w(r|u,v)$ via Algorithm 5 and randomized initialization of user/movie group probabilities $\mathbf{x}_{m\to n}^{(0)}(v)$ and $\mathbf{y}_{n\to m}^{(0)}(u)$.

Step II: Recursive update for user/movie group probabilities

$$\mathbf{y}_{n\to m}^{(i+1)}(u) \propto \mathbf{y}_{n}^{(0)}(u) \prod_{k\in\mathcal{V}_n\backslash m} \sum_{v} w\left(r|u,v\right) \mathbf{x}_{k\to n}^{(i)}(v)$$

$$\mathbf{x}_{m\to n}^{(i+1)}(v) \propto \mathbf{x}_{m}^{(0)}(v) \prod_{k\in\mathcal{U}_m\backslash n} \sum_{u} w\left(r|u,v\right) \mathbf{y}_{k\to m}^{(i)}(u)$$

Step III: Update $w(r|u,v)$ and output probability of rating R_{nm} given observed ratings

$$\hat{p}_{R_{nm}|\mathbf{R}_O}^{(i+1)}(r) \propto \sum_{u,v} \mathbf{y}_{n\to m}^{(i+1)}(u) w\left(r|u,v\right) \mathbf{x}_{m\to n}^{(i+1)}(v)$$

membership)" to each of the critics which can take on values between 0 and 1. After soft-clustering users/movies each into user/movie groups, we estimate $w(r|u,v)$ by computing the soft frequency of each rating for each user-movie group pair.

2. Message-Passing Updates of Group Vectors

Using the model from Section A, we describe how message-passing can be used for the prediction of hidden variables based on observed ratings. Ideally, we could perform exact inference of our factor graph model. But exact learning and inference for this model is intractable, so we turn to approximate message-passing algorithms (e.g., the sum-product algorithm) [69]. The basic idea is that the local neighborhood of any node in the factor graph is tree-like (see [70] for details). For iteration i, we simplify notation by denoting the message from movie m to user n by $\mathbf{x}_{m\to n}^{(i)}$ and the message from user n to movie m by $\mathbf{y}_{n\to m}^{(i)}$. The iteration is initialized with

$$\mathbf{x}_{m\to n}(v)=\mathbf{x}_m(v)=p_V(v),\ \mathbf{y}_{n\to m}(u)=\mathbf{y}_n(u)=p_U(u).$$

The set of all users who rated movie m is denoted $\mathcal{U}_m$ and the set of all movies whose rating by user n was observed is denoted $\mathcal{V}_n$. The exact update equations are given in Algorithm 4. The group probabilities are randomly initialized by assuming that the initial group (of the user and movie) probabilities are uniform across all groups.

3. Approximate Matrix Completion

Since the primary goal is the prediction of hidden variables based on observed ratings, the IMP algorithm focuses on estimating the distribution of each hidden variable given the observed ratings. In particular, the outputs of the algorithm (after i iterations) are estimates of the distributions for R_{nm}, U_n, and V_m. They are denoted, respectively, as

$$\begin{aligned}
\hat{p}^{(i+1)}_{R_{nm}|\mathbf{R}_O}(r) &\propto \sum_{u,v} \mathbf{y}^{(i+1)}_{n\to m}(u) w\left(r|u,v\right) \mathbf{x}^{(i+1)}_{m\to n}(v)\\
\hat{p}^{(i+1)}_{U_n|\mathbf{R}_O}(u) &\propto \mathbf{y}^{(0)}_n(u) \prod_{k\in\mathcal{V}_n} \sum_{v} w\left(r|u,v\right) \mathbf{x}^{(i)}_{k\to n}(v)\\
\hat{p}^{(i+1)}_{V_m|\mathbf{R}_O}(v) &\propto \mathbf{x}^{(0)}_m(v) \prod_{k\in\mathcal{U}_m} \sum_{u} w\left(r|u,v\right) \mathbf{y}^{(i)}_{k\to m}(u).
\end{aligned}$$

Using these, one can minimize various types of prediction error. For example, minimizing the mean-squared prediction error results in the conditional mean estimate (see Fig. 16)

$$\hat{r}^{(i)}_{n,m} = \sum_{r\in\mathcal{R}} r\, \hat{p}^{(i)}_{R_{nm}|\mathbf{R}_O}(r).$$

Algorithm 5 Initializing Group Ratings (shown only for users)

Step I: Initialization

Let $i = j = 0$ and $c_m^{(0,0)}(0)$ be the average rating vector of users for movie m.

Step II: Splitting of critics

Set

$$c_m^{(i+1,j)}(u) = \begin{cases} c_m^{(i,j)}(u) & u = 0, \ldots, 2^i - 1 \\ c_m^{(i,j)}(u - 2^i) + z_m^{(i+1,j)}(u) & u = 2^i, \ldots, 2^{i+1} - 1 \end{cases}$$

where the $z_m^{(i+1,j)}(u)$ are i.i.d. random variables with small variance.

Step III: Recursive soft K-means clustering for $c_m^{(i,j)}(u)$ for $j = 1, \ldots, J$.

1. Each user is assigned a soft group membership $\pi_n(u)$ to each of the critics using

$$\pi_n^{(i,j)}(u) \propto \exp\left(-\beta\sqrt{\frac{1}{|\mathcal{V}_n|}\sum_{m\in\mathcal{V}_n}\left(c_{m,n}^{(i,j)}(u) - R_{nm}\right)^2}\right)$$

where $\mathcal{V}_n = \{m \in [M] \mid (n, m) \in O\}$ and $g_u = 2^{i+1}$.

2. Update all critics as

$$c_m^{(i,j+1)}(u) \propto \sum_n \pi_n^{(i,j)}(u)\, c_m^{(i,j)}(u).$$

Step IV: Repeat Steps II and III until the desired number of critics g_u is obtained.

Step V: Estimate of $w(r|u, v)$

After clustering users/movies each into user/movie groups with the soft group membership $\pi_n(u)$ and $\tilde{\pi}_m(v)$, compute the soft frequencies of ratings for each user/movie group pair as

$$w(r|u, v) \propto \sum_{(n,m)\in O : R_{nm} = r} \pi_n(u)\, \tilde{\pi}_m(v)\,.$$

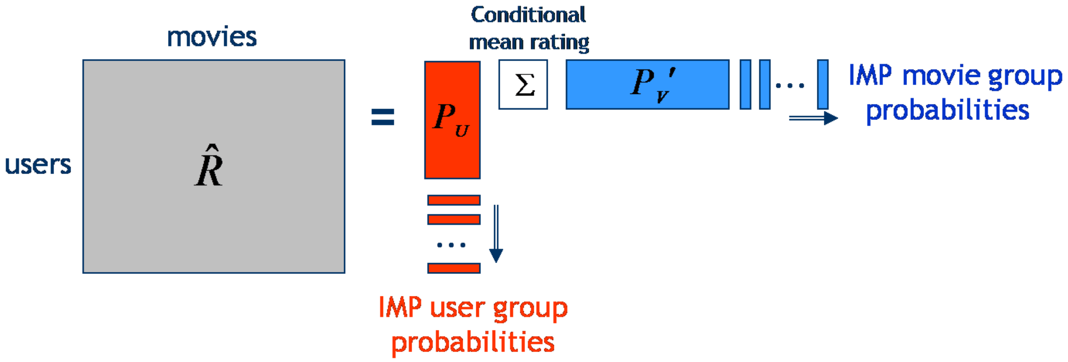


Fig. 16. Minimum mean square estimator (MMSE) estimates $\hat{R}$ can be written as a matrix factorization. Each element of Σ represents the conditional mean rating of $w\,(r|u, v)$ given u, v and each row of P_U/P_V represents a user/movie group probabilities. In contrast to the basic low-rank matrix model, we add non-negativity (to Σ, P_U and P_V) and normalization constraints (to both P_U and P_V).

C. The IEM Algorithm

We reformulate the problem in a standard variational Expectation Maximization (EM) framework and propose iterative EM based algorithm, termed IEM, by minimizing an upper bound on the free energy [71]. In other words, we view the problem as maximum-likelihood parameter estimation problem where $p_{U_n}(\cdot)$, $p_{V_m}(\cdot)$, and $p_{R|U,M}(\cdot|\cdot)$ are the model parameters θ and $\mathbf{U}, \mathbf{V}$ are the missing data. For each of these parameters, the i-th estimate is denoted $f_n^{(i)}(u)$, $h_m^{(i)}(v)$, and $w^{(i)}(r|u, v)$. Let $O \subseteq [N] \times [M]$ be the set of user-movie pairs that have been observed. As the first step, we specify a complete data likelihood as

$$\Pr\left(R_{nm} = r_{n,m}, U_n = u_n, V_m = v_m\right) = w\left(r_{n,m}|u_n, v_m\right) f_n\left(u_n\right) h_m\left(v_m\right).$$

Then, we can write the complete data (negative) log-likelihood as

$$\begin{aligned} R^c(\theta) &= -\ln \prod_{(n,m)\in O} \Pr\left(R_{nm} = r_{n,m}, U_n = u_n, V_m = v_m\right) \\ &= -\ln \prod_{(n,m)\in O} w\left(r_{n,m}|u_n, v_m\right) f_n\left(u_n\right) h_m\left(v_m\right). \end{aligned}$$

Using a variational approach, this can be upper bounded by

$$\sum_{(n,m)\in O} D\left(Q_{U_n,V_n|R_{nm}}(\cdot,\cdot|r_{n,m})||\hat{p}_{U_n,V_m|R_{nm}}(\cdot,\cdot|r_{n,m})\right),$$

where we introduce the variational probability distributions $Q_{U_n,V_m|R_{nm}}\left(u,v|r\right)$ that satisfy

$$\sum_{u,v} Q_{U_n,V_m|R_{nm}}\left(u,v|r\right) = 1$$

and let

$$\hat{p}_{U_n,V_m|R_{nm}}(u,v|r) = \frac{w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right)}{\sum_{u',v'} w\left(r_{n,m}|u',v'\right) f_n\left(u'\right) h_m\left(v'\right)}.$$

The IEM algorithm now consists of two steps that are performed in alternation with a Q distribution to approximate a general distribution.

1. E-step

Since the states of the latent variables are not known, we introduce a variational probability distribution

$$Q_{U_n,V_m|R_{nm}}\left(u,v|r\right) \text{ subject to } \sum_{u,v} Q_{U_n,V_m|R_{nm}}\left(u,v|r\right) = 1$$

Algorithm 6 IEM Algorithm

Step I: Initialization of $w(r|u,v)$ via Algorithm 5 and randomized initialization of user/movie group probabilities $f_n^{(0)}(u)$ and $h_m^{(0)}(v)$.

Step II: Recursive update for user/movie group probabilities and $w(r|u,v)$

$$f_n^{(i+1)}(u) \propto \sum_{m\in\mathcal{V}_n} f_n^{(i)}\left(u\right) \sum_{v\in[g_m]} w^{(i)}\left(r_{n,m}|u,v\right) h_m^{(i)}(v)$$

$$h_m^{(i+1)}(v) \propto \sum_{n\in\mathcal{U}_m} h_m^{(i)}(v) \sum_{u\in[g_u]} w^{(i)}\left(r_{n,m}|u,v\right) f_n^{(i)}\left(u\right)$$

$$w^{(i+1)}\left(r|u,v\right) \propto \sum_{(n,m):r_{n,m}=r} w^{(i)}\left(r_{n,m}|u,v\right) f_n^{(i+1)}(u) h_m^{(i+1)}(v)$$

Step III: Output probability of rating R_{nm} given observed ratings

$$\hat{p}^{(i+1)}_{R_{nm}|\mathbf{R}_O}(r) \propto \sum_{u,v} f_n^{(i+1)}(u) h_m^{(i+1)}(v) w^{(i+1)}\left(r|u,v\right)$$

$$\hat{p}^{(i+1)}_{U_n|\mathbf{R}_O}(u) = f_n^{(i+1)}(u), \;\; \hat{p}^{(i+1)}_{V_m|\mathbf{R}_O}(v) = h_m^{(i+1)}(v)$$

for all observed pairs (n,m). Exploiting the concavity of the logarithm and using Jensen's inequality, we have

$$\begin{aligned}
R\left(\theta\right) &= -\sum_{(n,m)\in O} \ln \sum_{u,v} \Pr\left(R_{nm} = r_{n,m}, U_n = u_n, V_m = v_m\right) \\
&= -\sum_{(n,m)\in O} \ln \sum_{u,v} Q_{U_n,V_m|R_{nm}}\left(u,v|r\right) \frac{w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right)}{Q_{U_n,V_m|R_{nm}}\left(u,v|r\right)} \\
&\leq -\sum_{(n,m)\in O} \sum_{u,v} Q_{U_n,V_m|R_{nm}}\left(u,v|r\right) \ln \frac{w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right)}{Q_{U_n,V_m|R_{nm}}\left(u,v|r\right)} \\
&\triangleq \bar{R}\left(\theta\,|Q\right) - \sum_{(n,m)\in O} H\left(Q\left(\cdot|u,v,r\right)\right) \\
&\triangleq R\left(\theta;\, Q\right)
\end{aligned}$$

To compute the tightest bound given parameters $\hat{\theta}$ i.e., we optimize the bound w.r.t the Qs using

$$\nabla_Q \left[R\left(\theta;\, Q\right) + \sum_{(n,m)\in O} \sum_{u,v} \lambda_{u,v}\, Q \right] = 0.$$

These yield posterior probabilities of the latent variables,

$$\hat{p}_{U_n,V_m|R_{nm}}(u,v|r;\hat{\theta}) = Q^{*}_{U_n,V_m|R_{nm}}\left(u,v|r;\hat{\theta}\right) = \frac{w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right)}{\sum_{u',v'} w\left(r_{n,m}|u',v'\right) f_n\left(u'\right) h_m\left(v'\right)}.$$

Also note that we can get the same result by Gibbs inequality as

$$\begin{aligned} R\left(\theta\right) &\le -\sum_{(n,m)\in O} \sum_{u,v} Q_{U_n,V_m|R_{nm}}\left(u,v|r\right) \ln \frac{w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right)}{Q_{U_n,V_m|R_{nm}}\left(u,v|r\right)} \\ &= \sum_{(n,m)\in O} D\left(Q_{U_n,V_n|R_{nm}}(\cdot,\cdot|r_{n,m}) || \hat{p}_{U_n,V_m|R_{nm}}(\cdot,\cdot|r_{n,m})\right). \end{aligned}$$

2. M-step

Obviously the posterior probabilities need only to be computed for pairs $(n,\, m)$ that have actually been observed. Thus optimize

$$\begin{aligned} &\bar{R}\left(\theta,\hat{\theta}\right) \\ &= -\sum_{(n,m)\in O} \sum_{u,v} Q^{*}_{U_n,V_m|R_{nm}}\left(u,v|r;\hat{\theta}\right) \ln w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right) \\ &= -\sum_{(n,m)\in O} \sum_{u,v} \frac{w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right)}{\sum_{u',v'} w\left(r_{n,m}|u',v'\right) f_n\left(u'\right) h_m\left(v'\right)} \ln w\left(r_{n,m}|u,v\right) f_n\left(u\right) h_m\left(v\right) \end{aligned}$$

with respect to parameters θ which leads to the three sets of equations for the update of

$$w\left(r|u,v\right),\ f_n\left(u\right),\ h_m\left(v\right).$$

Moreover, for large scale problems, to avoid computational loads of each step, combining both E and M steps by plugging Q function into M-step gives more tractable EM Algorithm. The resulting equations are presented in Algorithm 6.

Remark 6. *The results show that this variational approach gives the equivalent update rule as the standard EM framework with a simpler derivation which guarantees convergence to local minima. This IEM algorithm, in fact, extends Thomas Hofmann's work and generalizes probabilistic matrix factorization (PMF) results [72, 73] and uses alternating steps of KL divergence minimization to estimate the underlying generative model [74]. Its main drawback is that it is difficult to analyze because the effects of initial conditions and local minima can be very complicated. Though the idea is similar to an IMP update, the resulting equations are different and seem to perform much worse.*

D. Generalization Error Bound

In this section, we consider bounds on generalization from partial knowledge of the (binary-rating) matrix for collaborative filtering application. The tighter bound implies one can use most of known ratings for learning the model completely. Since computation of $\mathbf{R}$ can be viewed as the product of three matrices, we consider the simplified class of tri-factorized matrices χ_{g_u,g_v} as,

$$\left\{X|X = U^T WV, U \in [0,1]^{g_u\times N} , V \in [0,1]^{g_v\times M} , W \in \{\pm1\}^{g_u\times g_v}\right\}.$$

We bound the overall distortion between the entire predicted matrix X and the true matrix Y as a function of the distortion on the observed set of size $|O|$ and the error ϵ. Let $y \in \{\pm1\}$ be binary ratings and define a zero-one sign agreement distortion as

$$d(x, y) \triangleq \begin{cases} 1 & \text{if } xy \le 0 \\ 0 & \text{otherwise} \end{cases}.$$

Also, define the average distortion over the entire prediction matrix as

$$D\left(X,\,Y\right) \triangleq \sum_{(n,m)\in[N]\times[M]} d\left(x,\,y\right)/NM$$

and the averaged observed distortion as

$$D_O\left(X,\,Y\right) \triangleq \sum_{(n,m)\in O} d\left(x,\,y\right)/|O|.$$

Theorem 8. For any matrix $Y \in \{\pm 1\}^{N\times M}$, $N,\,M > 2$, $\delta > 0$ and integers g_u and g_v, with probability at least $1-\delta$ over choosing a subset O of entries in Y uniformly among all subsets of $|O|$ entries $\forall X \in \chi_{g_u,g_v}$, $|D\left(X,\,Y\right) - D_O\left(X,\,Y\right)|$ is upper bounded by

$$\sqrt{\left\{(Ng_u + Mg_v + g_ug_v)\ \ln\frac{12eM}{\min(g_u,\,g_v)} - \ln\delta\right\}/2|O|} \triangleq h\left(g_u,\,g_v,\,N,\,M,\,|O|\right).$$

Proof. This proof follows arguments of the generalization error in [75]. First, fix Y as well as $X \in R^{N\times M}$. When an index pair $(n,\,m)$ is chosen uniformly random, $\mathrm{d}\left(x_{n,m},\,y_{n,m}\right)$ is a Bernoulli random variable with probability $D\left(X,\,Y\right)$ of being one. If the entries of O are chosen independently random, $|O|D_O\left(X,\,Y\right)$ is binomially distributed with parameters $|O|D\left(X,\,Y\right)$ and $|O|\epsilon$. Using Chernoff's inequality, we get

$$\begin{aligned}\Pr\left(D\left(X,\,Y\right) \geq D_O\left(X,\,Y\right) + \epsilon\right) &= \Pr\left(|O|D_O\left(X,\,Y\right) \leq |O|D\left(X,\,Y\right) - |O|\epsilon\right)\\ &\leq \mathsf{e}^{-2|O|\epsilon^2}.\end{aligned}$$

Now note that $d\left(x,\,y\right)$ only depends on the sign of xy, so it is enough to consider equivalence classes of matrices with the same sign patterns. Let $f\left(N,\,M,\,g_u,\,g_v\right)$ be the number of such equivalence classes. For all matrices in an equivalence class, the random variable $D_O\left(X,\,Y\right)$ is the same. Thus we take a union bound of the events

$\{X|D\,(X,\,Y) \geq D_O\,(X,\,Y) + \epsilon\}$ for each of these $f\,(N,\,M,\,g_u,\,g_v)$ random variables with the bound above and $\epsilon = \sqrt{(\ln\,f\,(N,\,M,\,g_u,\,g_v) - \ln\delta)/(2|O|)}$, we have

$$\Pr\left(\exists X \in \chi_{g_u,g_v}\, D\,(X,\,Y) \geq D_O\,(X,\,Y) + \sqrt{\frac{\ln\,f\,(N,\,M,\,g_u,\,g_v) - \ln\delta}{2|O|}}\right) \leq \delta.$$

Since any matrix $X \in \chi_{g_u,g_v}$ can be written as $X = U^T G V$, to bound the number of sign patterns of X, $f\,(N,\,M,\,g_u,\,g_v)$, consider $Ng_u + Mg_v + g_u g_v$ entries of U, G, V as variables and the NM entries of X as polynomials of degree three over these variables as

$$x_{n,m} = \sum_{k=1}^{g_u}\sum_{l=1}^{g_v} u_{k,n} \cdot g_{k,l} \cdot v_{l,m}.$$

By the use of the bound in Lemma 7, we obtain

$$\begin{aligned} f\,(N,\,M,\,g_u,\,g_v) &\leq \left(\frac{4\mathsf{e}\cdot 3\cdot NM}{Ng_u + Mg_v + g_u g_v}\right)^{Ng_u+Mg_v+g_ug_v} \\ &\leq \left(\frac{12\mathsf{e}M}{\min(g_u,\,g_v)}\right)^{Ng_u+Mg_v+g_ug_v}. \end{aligned}$$

This bound yields a factor of $\ln\,12\mathsf{e}M/\min(g_u,\,g_v)$ in the bound and establishes the theorem. □

Lemma 7 ([76])**.** Total number of sign patterns of r polynomials, each of degree at most d, over q variables, is at most $(8\mathsf{e}dr/q)^q$ if $2r > q > 2$. Also, total number of sign patterns of r polynomials with $\{\pm 1\}$ coordinates, each of degree at most d, over q variables, is at most $(4\mathsf{e}dr/q)^q$ if $r > q > 2$.

Remark 7. *There are two implications of the Theorem. 8 in terms of the five parameters: g_u, g_v, N, M, $|O|$. For fixed group numbers g_u and g_v, as number of users N and movies M increases, to keep the bound tight, number of observed ratings $|O|$ also needs to grow in the same order. For a fixed sized matrix, when the choice of g_u and/or g_v increases, $|O|$ needs to grow in the same order to prevent over-learning the* *model. Also, as $|O|$ increases, we could increase the value of g_u and/or g_v.*

E. Density Evolution (DE) Analysis

DE is a well-known technique for analyzing probabilistic message-passing inference algorithms that was originally developed to analyze belief-propagation decoding of error-correcting codes and was later extended to more general inference problems [77]. It works by tracking the distribution of the messages passed on the graph under the assumption that the local neighborhood of each node is a tree. While this assumption is not rigorous, we consider that, in Fig. 15, the outgoing edges from each user node are attached to movie nodes via random permutations. This is identical to the model used for irregular LDPC codes [78]. While this assumption is not rigorous, it is motivated by the following lemma. We consider the factor graph for a randomly chosen instance of this problem. The key assumption is that the outgoing edges from each user node are attached to movie nodes via a random permutation. This is identical to the model used for irregular LDPC codes.

Lemma 8. Let $\mathcal{N}_l(v)$ denote the depth-l neighborhood (i.e., the induced subgraph including all nodes within l steps from v) of an arbitrary user (or movie) node v. Let the problem size N become unbounded with $M = \beta N$ for $\beta < 1$, maximum degree d_N, and depth-l_N neighborhoods. One finds that if

$$\frac{(2l_N + 1)\ln d_N}{\ln N} < 1 - \delta,$$

for some $\delta > 0$ and all N, then the graph $\mathcal{N}_l(v)$ is a tree w.h.p. for almost all v as $N \to \infty$.

Proof. The proof follows from a careful treatment of standard tree-like neighborhood arguments. Starting from any node v, we can recursively grow $\mathcal{N}_{i+1}(v)$ from $\mathcal{N}_i(v)$ by adding all neighbors at distance $i+1$. Let A_i be the number of outgoing edges from $\mathcal{N}_i(v)$ to the next level and $b_1^{(i)}, \ldots, b_n^{(i)}$ be the degrees of the n_i available nodes that can be chosen in the next level. The probability that the graph remains a tree is

$$p\left(A_i, \mathbf{b}^{(i)}\right) = \frac{\sum_{S \subset [n], |S| = A_i} \prod_{s \in S} b_s^{(i)}}{\binom{\sum_{j=1}^{n} b_j^{(i)}}{A_i}},$$

where the numerator is the number of ways that the A_i edges can attach to distinct nodes in the next level and the denominator is the total number of ways that the A_i edges may attach to the available nodes. Using the fact that the numerator is an unnormalized expected value of the product of A_i b's drawn without replacement, we can lower bound the numerator using

$$\sum_{S \subset [n], |S| = A_i} \prod_{s \in S} b_s^{(i)} \geq \binom{n_i}{A_i} \left(\bar{b}_i - \frac{(d-1)A_i}{n_i}\right)^{A_i} \geq \frac{(n_i - A_i)^{A_i}}{A_i!} \left(\bar{b}_i - \frac{(d-1)A_i}{n_i}\right)^{A_i}.$$

This can be seen as lower bounding the expected value of A_i b's drawn from with replacement from a distribution with a slightly lower mean. Upper bounding the denominator by $(n_i \bar{b}_i)^{A_i}/A_i!$ gives

$$\begin{aligned} p\left(A_i, \mathbf{b}^{(i)}\right) &\geq \frac{(n_i - A_i)^{A_i} A_i! \left(\bar{b}_i - \frac{(d-1)A_i}{n_i}\right)^{A_i}}{\left(n_i \bar{b}_i\right)^{A_i} A_i!} \\ &= \left(1 - \frac{A_i}{n_i}\right)^{A_i} \left(1 - \frac{(d-1)A_i}{\bar{b}_i n_i}\right)^{A_i} \\ &\geq \left(1 - \frac{A_i^2}{n_i} - \frac{A_i^2(d-1)}{\bar{b}_i n_i}\right). \end{aligned}$$

Now, we can take the product from $i = 0, \ldots, l-1$ to get

$$\begin{aligned}
\Pr\left(\mathcal{N}_l(v) \text{ is a tree}\right) &= \prod_{i=0}^{l-1} \Pr\left(\mathcal{N}_{i+1}(v) \text{ is a tree} | \mathcal{N}_0(v), \ldots, \mathcal{N}_i(v) \text{ are trees}\right) \\
&\geq \prod_{i=0}^{l-1} \left(1 - \frac{A_i^2}{n_i} - \frac{A_i^2(d-1)}{\overline{b}_i n_i}\right) \\
&\geq 1 - \sum_{i=0}^{l-1} \left(\frac{A_i^2}{n_i} + \frac{A_i^2(d-1)}{\overline{b}_i n_i}\right) \\
&\geq 1 - \left(1 + \frac{1}{d^2-1}\right)\left(\frac{d^{2l}}{\beta N - d^l} + \frac{d^{2l}(d-1)}{\beta N - d^l}\right) \\
&\geq 1 - \left(1 + \frac{1}{d^2-1}\right)\frac{d^{2l+1}}{\beta N - d^l},
\end{aligned}$$

because $A_i \leq d^{i+1}$, $\sum_{i=0}^{l-1} A_i^2 \leq d^2 d^{2l}/(d^2-1) = d^{2l}\left(1 + 1/(d^2-1)\right)$, and $n_i \geq \beta N - \sum_{j=0}^{i} d^j \geq \beta N - d^{i+1}$. Examining the expression

$$\ln \frac{d^{2l+1}}{\beta N - d^l} \leq (2l_N + 1)\ln d_N - \ln N + O(1) \leq -\delta \ln N + O(1)$$

shows that the probability of failure is $O\left(N^{-\delta}\right)$. Let Z be a r.v. whose value is the number of user nodes whose depth-l neighborhood is not a tree. We can upper bound the expected value of Z with

$$E[Z] \leq \frac{d^{2l+1}}{\Theta(N) - d^l} N \leq \frac{O\left(N^{-\delta}\right)}{\Theta\left(N\right) - O\left(N^{1/2}\right)} N = O\left(N^{1-\delta}\right).$$

With Markov's inequality, one can show that

$$\Pr\left(Z \geq N^{1-\delta/2}\right) \leq \frac{E[Z]}{N^{1-\delta/2}} \leq \frac{O\left(N^{1-\delta}\right)}{N^{1-\delta/2}}.$$

Therefore, the depth-l neighborhood is a tree (w.h.p. as $N \to \infty$) for all but a vanishing fraction of user nodes. □

For this problem, the messages passed during inference consist of belief functions for user groups (e.g., passed from movie nodes to user nodes) and movie groups (e.g., passed form user nodes to movie nodes). The message set for user belief functions is $\mathcal{M}_u = \mathcal{P}([g_u])$, where $\mathcal{P}(S)$ is the set of probability distributions over the finite set S. Likewise, the message set for movie belief functions is $\mathcal{M}_v = \mathcal{P}([g_v])$. The decoder combines d user (resp. movie) belief-functions $a_1(\cdot), \ldots, a_d(\cdot) \in \mathcal{M}_u$ (resp. $b_1(\cdot), \ldots, b_d(\cdot) \in \mathcal{M}_v$) using

$$F_d\left(a_1, r_1, ..., a_d, r_d; b\right) \triangleq \frac{b(v) \prod_{j=1}^{d} \sum_u a_j(u) w\left(r_j|u, v\right)}{\sum_v b(v) \prod_j \sum_u a_j(u) w\left(r_j|u, v\right)}$$

$$G_d\left(b_1, r_1, ..., b_d, r_d; a\right) \triangleq \frac{a(u) \prod_{j=1}^{d} \sum_v b_j(v) w\left(r_j|u, v\right)}{\sum_u a(u) \prod_j \sum_v b_j(v) w\left(r_j|u, v\right)}.$$

Since we need to consider the possibility that the ratings are generated by a process other than the assumed model, we must also keep track of the true user (or movie) group associated with each belief function. Let $\mu^{(i)}(u, A)$ (resp. $\nu^{(i)}(v, B)$) be the probability that, during the i-th iteration, a randomly chosen user (resp. movie) message is coming from a node with true user group u (resp. movie group v) and has a user belief function $a(\cdot) \in A \subseteq \mathcal{M}_u$ (resp. movie belief function $b(\cdot) \in B \subseteq \mathcal{M}_v$). The DE update equations for degree d user and movie nodes, in the spirit of [77], are shown in equations (5.1) and (5.2) where $I(x \in A)$ is defined as a indicator function

$$I(x \in A) = \begin{cases} 1 & \text{if}\, x \in A \\ 0 & \text{if}\, x \notin A \end{cases}.$$

Like LDPC codes, we expect to see that the performance of Algorithm 4 depends crucially on the degree structure of the factor graph. Therefore, we let Λ_j (resp. Γ_j) be the fraction of user (resp. movie) nodes with degree j and define the edge degree distribution to be $\lambda_j = \Lambda_j j / \sum_{k \geq 1} \Lambda_k k$ (resp.$\rho_j = \Gamma_j j / \sum_{k \geq 1} \Gamma_k k$). Averaging over the degree distribution gives the final update equations

$$
\begin{aligned}
&\mu_d^{(i+1)}(u,B)\\
&\quad= \int \sum_{r_1,\ldots,r_d} I\left(G\left((b_1,r_1),\ldots,(b_d,r_d);a\right)\in B\right)\mu^{(0)}(u,da)\prod_{j=1}^{d}\sum_{v}\nu^{(i)}(v,db_j)\,w\left(r_j|u,v\right)
\end{aligned}
\tag{5.1}
$$

$$
\begin{aligned}
&\nu_d^{(i+1)}(v,A)\\
&\quad= \int \sum_{r_1,\ldots,r_d} I\left(F\left((a_1,r_1),\ldots,(a_d,r_d);b\right)\in A\right)\nu^{(0)}(v,db)\prod_{j=1}^{d}\sum_{u}\mu^{(i)}(u,da_j)\,w\left(r_j|u,v\right)
\end{aligned}
\tag{5.2}
$$

$$
\begin{aligned}
\mu^{(i+1)}(u,B) &= \sum_{d\geq 1}\lambda_d\mu_d^{(i+1)}(u,B)\\
\nu^{(i+1)}(v,A) &= \sum_{d\geq 1}\rho_d\nu_d^{(i+1)}(v,A).
\end{aligned}
$$

We anticipate that this analysis will help us understand the IMP algorithm's observed performance for large problems based on the success of DE for channel coding problems.

F. Simulation Results with Real Data Matrices

1. Details of Training

The key challenge of matrix completion problem is predicting the missing ratings of a user for a given item based only on very few known ratings in a way that minimizes some per-letter metric $d(r,r')$ for ratings. To provide further insights into the proposed factor graph model and the IMP algorithm, we compared our results against three other algorithms: OptSpace [27], SET [28] and SVT [25]. Due to time and space constraints, we have chosen three algorithms among all the available algorithms. OptSpace and the more recent SET appear to be the best (this is also apparent from experimental results), and can handle reasonably large matrix sizes. In some cases, the programs are publicly available (e.g., [27, 25]) and others (e.g., [28]) have been obtained from their respective authors.

To make a fair comparison between different algorithms/models whose complexity varies widely, we have created two smaller submatrices from the real Netflix dataset:

- **Netflix Data Matrix 1** is a matrix given by the first 5,000 movies and users. This matrix contains 280,714 user/movie pairs. Over 15% of the users and 43% of the movies have less than 3 ratings.

- **Netflix Data Matrix 2** is a matrix of 5,035 movies and 5,017 users by selecting some 5,300 movies and 7,250 users and avoiding movies and users with less than 3 ratings. This matrix contains 454,218 user/movie pairs. Over 16% of the users and 41% of the movies have less than 10 ratings.

To provide further insights into the quality of the proposed factor graph model and suboptimality of the algorithms by comparison with the theoretical lower bounds, we generated two synthetic datasets from the above partial matrices. The synthetic datasets are generated once with the learned density $\hat{p}^{(i)}_{R_{nm}|\mathbf{R}_O}(r)$, $\hat{p}^{(i)}_{U_n|\mathbf{R}_O}(u)$, and $\hat{p}^{(i)}_{V_m|\mathbf{R}_O}(v)$ and then randomly subsampled as the partial Netflix datasets.

- **Synthetic Dataset 1** is generated after learning Netflix Dataset 1 with $g_u = g_v = 8$.

- **Synthetic Dataset 2** is generated after learning Netflix Dataset 2 with $g_u = g_v = 16$.

Also, we hide 1,000 randomly selected user/movie entries as a validation set S. The performance is evaluated using the root mean squared error (RMSE) of prediction on this set defined by

$$\sqrt{\sum_{(n,m)\in S} \left(\hat{r}_{n,m} - r_{n,m}\right)^2 / \, |S|}.$$

We primarily focused on the RMSE as a function of the average number of observation ratings per user (i.e., how many ratings, $|O|$, are needed to get each algorithm in shape). Simulations were performed in the very small sample regime (e.g., much less than 0.5% of ratings) by varying the randomly selected average number of observed ratings per user between 1 and 30 and the average results are shown in Fig. 17 and Fig. 18. Note that the choice of parameters for each algorithm (e.g., g_u and g_v for IMP and rank for others) was optimized over the validation set S by running each algorithm multiple times. For IMP, we used hard K-means clustering (i.e., soft K-means clustering with large β) for Algorithm 5 Step III to improve the speed of $w(r|u, v)$ initialization. Also, to make a fair comparison with algorithms that provide unbounded predictions, we clip the out-of-range predictions (i.e., ratings greater than 5 or less than 1), if there are any.

2. Discussion

Our results do shed some light on the performance of recommender systems based on the MP framework. First, we have verified that IMP really does improve the cold-start problem. From simulation results on Netflix submatrices in Fig. 17, we clearly see while other matrix completion algorithms perform similarly with large amounts of revealed entries, the IMP algorithm can estimate the matrix very well only after a few observed entries. The performance of other algorithms for users with fewer than 5 ratings is generally poorer than that of the simple movie average

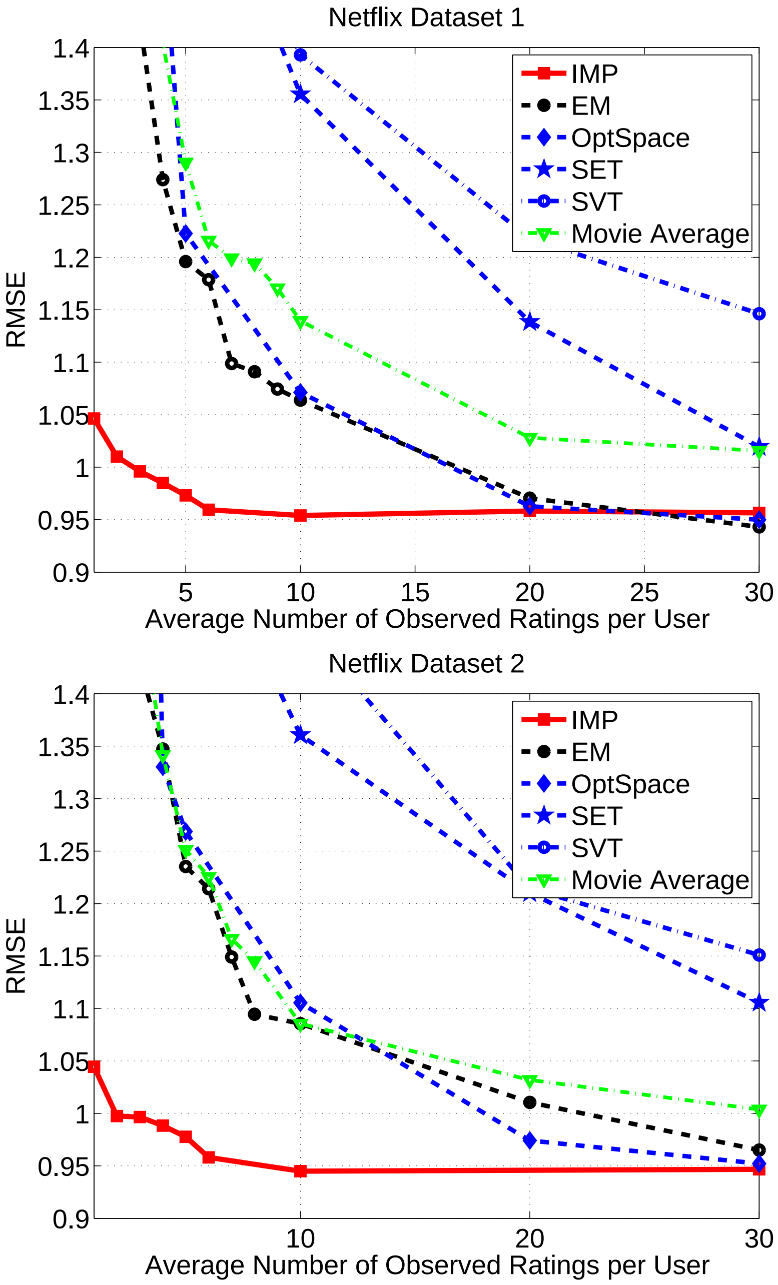


Fig. 17. Remedy for the Cold-Start Problem: RMSE performance is compared with other different competing algorithms [27, 28, 25] on the validation set versus the average number of observations per user for Netflix submatrices.

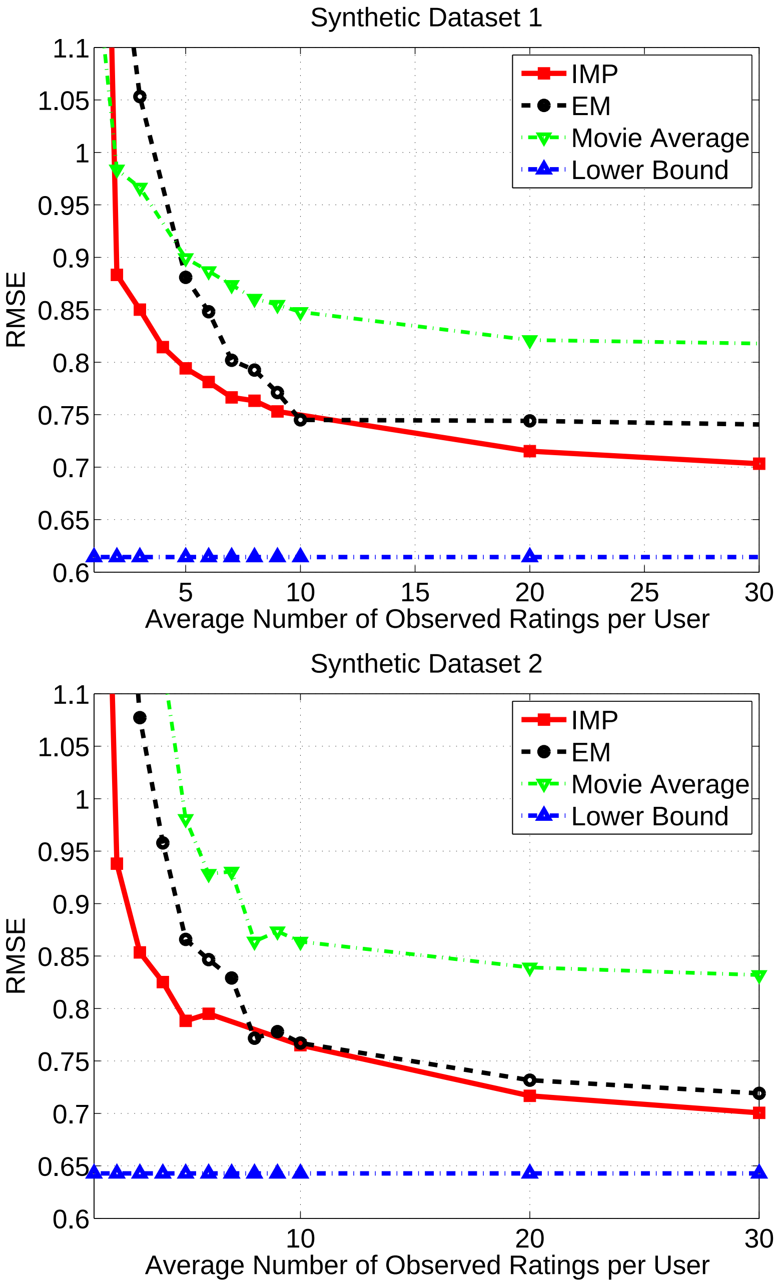


Fig. 18. Each plot shows the RMSE on the validation set versus the average number of observations per user for synthetic datasets. Performance is compared with an (analytical) lower bound on RMSE assuming known user and movie group.

algorithm that uses the average rating for each movie as the prediction. The IMP algorithm, however, performs considerably better on users with a very few ratings. This better threshold performance (see the steep RMSE decay) of the IMP algorithm in comparison to other algorithms helps to reduce the cold start problem. It is worth noting that the simple K-means clustering (used for $w(r|u, v)$ initialization) performs worse than movie average in the small sample regime (due to space limits, this curve is not shown). This implies that the improvement of IMP for the cold start problem comes from the MP update steps and not the clustering initialization. We believe this will be a major benefit of MP approaches to standard CF problems. Other than these important advantages, each output group has generative nature with explicit semantics. In other words, after learning the density, we can use them to generate synthetic data with clear meanings. These benefits do not extend to general low-rank matrix models easily.

CHAPTER VI

CONCLUSION

In this dissertation, we have developed convex optimization based algorithms to address joint detection and decoding problem and message passing based algorithm to address matrix completion problem. Our approach to both of these problems is inspired by methods of convex optimization, information and coding theory. Let us now highlight what we consider the most interesting aspects of our research and summarize a few of the other directions for further work.

A. Joint Detection and Decoding Problem

1. Summary

- To address this problem for channels with memory, such as finite-state channels (FSCs), we propose new decoding algorithms based on a well-defined convex optimization problem. In particular, it is based on Linear-programing (LP) formulation of joint decoding for LDPC codes on FSCs and shows two favorable properties: guranteed convergence and predictable error-floors via pseudo-codeword analysis. An important aspect of this method is the application of convex-programming relaxation to the problem of decoding an error-correcting code in more general channels.

- Since general-purpose LP solvers are highly complex for the joint LP decoding problem, we develop an efficient iterative solver for the joint LP decoder. To handle the fact that the Lagrangian dual function is non-differentiable, we use smoothed Lagrangian relaxation methods and maximize the smoothed dual function by block coordinate ascent on the Lagrangian multipliers. This leads to an iterative solver for the joint LP decoder that is closely related to BCJR-based turbo equalization (TE). This iterative algorithm shows the predictability and superior error-floor performance of joint LP decoding with the computational complexity of conventional TE. Essentially, we can achieve these benefits by small change to the current TE implementations.

- Lastly, we derive a block-coordinate ascent algorithm to maximize Lagrangian dual function. One motivation for this approach is to accelerate the rate to convergence and so as to more efficiently decode. It is also hoped this approach will later prove useful in practical use by providing a desirable iteration complexity bound, possibly leading to reduced computational complexity in applications where these effects are important.

2. Possible Extensions

Universal Joint LP Decoding

- A restrictive assumption made so far is that the coefficients of the FSCs are known. Indeed, running current iterative joint LP algorithm requires knowledge of channel coefficient. The idea is to formulate the universal LP joint decoder for joint channel parameter estimation and decoding for FSISI by letting $\mathbf{y}$ be the output of a finite-state ISI channel (FSISI) with zero-mean AWGN whose variance is σ^2 per output as

$$\min_{\mathbf{g}\in\mathcal{P}_{\mathcal{T}}(\mathbf{H}),\mathbf{a}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} \left(y_i - a_e\right)^2 g_{i,e} \tag{6.1}$$

subject to $a_{e_i} = a_e$ for $i \in \mathcal{I}$. This LP formulation will lead to an iterative algorithm for joint channel parameter estimation and decoding for FSISI that is analogue to EM-based algorithm.

Joint Decoding for Two-dimensional (2D) ISI Channel

- Dramatic increase in the demand for data storage over the past decade has fueled new interest in increasing the capacity of magnetic and optical storage. While both of the these storage media are physically 2D, current storage techniques store the data in one-dimension (1D). Unlike the 1D case, the lack of an optimal detector for the channel makes this problem very challenging. Joint LP decoding idea proposed for 1D-FSC can be leveraged for 2D-ISI. The idea is to formulate a similar LP based on pairwise potentials of the factor graph. We believe the resulting iterative algorithm will be similar to the iterative multistrip joint-decoding method with expected gains in error-floor.

B. Matrix Completion Problem

1. Summary

- In collaborative filtering applications, matrix completion problem is studied from a graphical models perspective. Exact learning of the model parameters is intractable for such models, we use a factor graph model to characterize the probability distribution underlying the collaborative filtering dataset. Main benefits of the factor-graph model is that an establishment of a generative model for data matrices and exploiting sparse observations which reduce complexity. Then, a message passing based algorithm, dubbed IMP, is introduced to infer the underlying distribution from the observed entries. IMP combines clustering with message passing and we attempt partial performance analysis of IMP algorithm via density evolution.

2. Possible Extensions

Convex-programming Relaxations Framework

- Major drawback of the current approach is that it mixes clustering and message-passing. Therefore it is difficult to analyze the algorithm's behavior. To get a fully iterative solution which combines clustering via message-passing, we can express the factor graph model for matrix completion problem in convex optimization framework as

  $$\max_{\mathbf{X},\mathbf{Y}} \left\|\mathbf{X}^T\mathbf{R}_O\mathbf{Y}\right\|_F^2$$

  subject to

  $$\mathbf{X}\mathbf{1}_{g_u} = \mathbf{1}_N, \;\; \mathbf{Y}\mathbf{1}_{g_v} = \mathbf{1}_M$$

  and

  $$\mathbf{X} \geq 0, \;\; \mathbf{Y} \geq 0,$$

  then using a similar approach developed for joint detection and decoding problem, we hope to obtain a fully distributed algorithm that is amenable to analysis.

REFERENCES

[1] R. G. Gallager, “Low-density parity-check codes,” Ph.D. dissertation, M.I.T., Cambridge, MA, 1960.

[2] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference: Turbo equalization,” *Eur. Trans. Telecom.*, vol. 6, no. 5, pp. 507–511, Sept. – Oct. 1995.

[3] R. R. Müller and W. H. Gerstacker, “On the capacity loss due to separation of detection and decoding,” *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1769–1778, Aug. 2004.

[4] W. E. Ryan, “Performance of high rate turbo codes on a PR4-equalized magnetic recording channel,” in *Proc. IEEE Int. Conf. Commun.*, Atlanta, GA, June 1998, pp. 947–951.

[5] L. L. McPheters, S. W. McLaughlin, and E. C. Hirsch, “Turbo codes for PR4 and EPR4 magnetic recording,” in *Proc. Asilomar Conf. on Signals, Systems & Computers*, vol. 2, Pacific Grove, CA, Nov. 1998, pp. 1778–1782.

[6] M. Öberg and P. H. Siegel, “Performance analysis of turbo-equalized dicode partial-response channel,” in *Proc. 36th Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, Sept. 1998, pp. 230–239.

[7] M. Tüchler, R. Koetter, and A. Singer, “Turbo equalization: Principles and new results,” *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[8] B. M. Kurkoski, P. H. Siegel, and J. K. Wolf, “Joint message-passing decoding of LDPC codes and partial-response channels,” *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1410–1422, June 2002.

[9] G. Ferrari, G. Colavolpe, and R. Raheli, *Detection Algorithms for Wireless Communications, with Applications to Wired and Storage Systems.* West Sussex, England: John Wiley & Sons, Ltd, 2004.

[10] A. Dholakia, E. Eleftheriou, T. Mittelholzer, and M. Fossorier, "Capacity-approaching codes: Can they be applied to the magnetic recording channel?" *IEEE Commun. Magazine*, vol. 42, no. 2, pp. 122 –130, Feb. 2004.

[11] A. Anastasopoulos, K. Chugg, G. Colavolpe, G. Ferrari, and R. Raheli, "Iterative detection for channels with memory," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1272–1294, June 2007.

[12] A. Kavčić and A. Patapoutian, "The read channel," *Proceedings of the IEEE*, vol. 96, no. 11, pp. 1761–1774, Nov. 2008.

[13] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. dissertation, M.I.T., Cambridge, MA, 2003.

[14] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 954–972, March 2005.

[15] [Online]. Available: http://www.PseudoCodewords.info

[16] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," Dec. 2005, [Online]. Available: http://arxiv.org/abs/cs/0512078.

[17] B.-H. Kim and H. D. Pfister, "On the joint decoding of LDPC codes and finite-state channels via linear programming," in *Proc. IEEE Int. Symp. Inform. Theory*, Austin, TX, June 2010, pp. 754–758.

[18] M. F. Flanagan, “Linear-programming receivers,” in *Proc. 47th Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, Sept. 2008, pp. 279–285.

[19] ——, “A unified framework for linear-programming based communication receivers,” Feb. 2009, [Online]. Available: http://arxiv.org/abs/0902.0892.

[20] T. Wadayama, “Interior point decoding for linear vector channels based on convex optimization,” *IEEE Trans. Inform. Theory*, vol. 56, no. 10, pp. 4905–4921, Oct. 2010.

[21] B.-H. Kim and H. D. Pfister, “An iterative joint linear-programming decoding of LDPC codes and finite-state channels,” in *Proc. IEEE Int. Conf. Commun.*, Kyoto, June 2011, pp. 1–6.

[22] E.J.Candes and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2008.

[23] E. Candes and Y. Plan, “Matrix completion with noise,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, June 2010.

[24] [Online]. Available: http://www.netflixprize.com

[25] J. Cai, E. Candes, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” Oct. 2008, [Online]. Available: http://arxiv.org/abs/0810.3286.

[26] K. Lee and Y. Bresler, “ADMIRA: Atomic decomposition for minimum rank approximation,” *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4402–4416, Sept. 2010.

[27] R. Keshavan, A. Montanari, and S. Oh, “Matrix completion from noisy entries,” June 2009, [Online]. Available: http://arxiv.org/abs/0906.2027.

[28] W. Dai and O. Milenkovic, “SET: An algorithm for consistent matrix completion,” Sept. 2009, [Online]. Available: http://arxiv.org/abs/0909.2705.

[29] R. Salakhutdinov and A. Mnih, “Bayesian probabilistic matrix factorization using Markov chain Monte Carlo,” in *Proceedings of the International Conference on Machine Learning*, vol. 25, 2008, pp. 880–887.

[30] E. Candes and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” March 2009, [Online]. Available: http://arxiv.org/abs/0903.1476.

[31] R. Keshavan, S. Oh, and A. Montanari, “Matrix completion from a few entries,” Sept. 2009, [Online]. Available: http://arxiv.org/abs/0901.3150v4.

[32] S. Aditya, O. Dabeer, and B. Dey, “A channel coding perspective of collaborative filtering,” *IEEE Trans. Inform. Theory*, vol. 57, no. 4, pp. 2327–2341, 2011.

[33] S. Vishwanath, “Information theoretic bounds for low-rank matrix completion,” Jan. 2010, [Online]. Available: http://arxiv.org/abs/1001.2331.

[34] A. Schein, A. Popescul, L. Ungar, and D. Pennock, “Methods and metrics for cold-start recommendations,” in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM New York, NY, 2002, pp. 253–260.

[35] M. F. Flanagan, V. Skachek, E. Byrne, and M. Greferath, “Linear-programming decoding of nonbinary linear codes,” *IEEE Trans. Inform. Theory*, vol. 55, no. 9, pp. 4134–4154, Sept. 2009.

[36] M. H. Taghavi and P. H. Siegel, “Graph-based decoding in the presence of ISI,” *IEEE Trans. Inform. Theory*, vol. 57, no. 4, pp. 2188–2202, April 2011.

[37] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, “Codes for digital recorders,” *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2260–2299, Oct. 1998.

[38] S. Jeon, X. Hu, L. Sun, and B. Kumar, “Performance evaluation of partial response targets for perpendicular recording using field programmable gate arrays,” *IEEE Trans. Magn.*, vol. 43, no. 6, pp. 2259–2261, 2007.

[39] J. Ziv, “Universal decoding for finite-state channels,” *IEEE Trans. Inform. Theory*, vol. 31, no. 4, pp. 453–460, July 1985.

[40] R. Ash, *Information Theory.* Mineola, New York: Dover, 1990.

[41] N. Wiberg, “Codes and decoding on general graphs,” Ph.D. dissertation, Linköping University, Linköping, 1996.

[42] C. Di, D. Proietti, E. Telatar, T. J. Richardson, and R. Urbanke, “Finite-length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

[43] T. Richardson, “Error floors of LDPC codes,” in *Proc. 42nd Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, 2003, pp. 1426–1435.

[44] G. D. Forney, Jr., R. Koetter, F. R. Kschischang, and A. Reznik, “On the effective weights of pseudocodewords for codes defined on graphs with cycles,” in *Codes, Systems and Graphical Models*, ser. IMA Volumes Series, B. Marcus and J. Rosenthal, Eds., vol. 123. New York: Springer, 2001, pp. 101–112.

[45] P. O. Vontobel and R. Koetter, “Towards low-complexity linear-programming decoding,” in *Proc. Int. Symp. on Turbo Codes & Related Topics*, Munich, Germany, April 2006, pp. 1–9.

[46] P. O. Vontobel, “Interior-point algorithms for linear-programming decoding,” in *Proc. 3rd Annual Workshop on Inform. Theory and Its Appl.*, San Diego, CA, Feb. 2008, pp. 433–437.

[47] M. Taghavi and P. Siegel, “Adaptive methods for linear programming decoding,” *IEEE Trans. Inform. Theory*, vol. 54, no. 12, pp. 5396–5410, Dec. 2008.

[48] T. Wadayama, “An LP decoding algorithm based on primal path-following interior point method,” in *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, June 2009, pp. 389–393.

[49] D. Burshtein, “Iterative approximate linear programming decoding of LDPC codes with linear complexity,” *IEEE Trans. Inform. Theory*, vol. 55, no. 11, pp. 4835–4859, Nov. 2009.

[50] M. Punekar and M. F. Flanagan, “Low complexity linear programming decoding of nonbinary linear codes,” in *Proc. 48th Annual Allerton Conf. on Commun., Control, and Comp.*, Monticello, IL, Sept. 2010, pp. 6–13.

[51] J. K. Johnson, “Convex relaxation methods for graphical models: Lagrangian and maximum entropy approaches,” Ph.D. dissertation, M.I.T., Cambridge, MA, 2008.

[52] P. Regalia and J. Walsh, “Optimality and duality of the turbo decoder,” *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1362–1377, 2007.

[53] F. Alberge, “Iterative decoding as Dykstra’s algorithm with alternate I-projection and reverse I-projection,” in *Proc. Eur. Signal Process. Conf.*, Lausanne, Switzerland, 2008.

[54] J. Walsh and P. Regalia, “Belief propagation, Dykstra’s algorithm, and iterated information projections,” *IEEE Trans. Inform. Theory*, vol. 56, no. 8, pp. 4114–4128, 2010.

[55] A. Kavčić, X. Ma, and M. Mitzenmacher, “Binary intersymbol interference channels: Gallager codes, density evolution and code performance bounds,” *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1636–1652, July 2003.

[56] D. Bertsekas, *Nonlinear Programming.* Belmont, MA: Athena Scientific Belmont, 1995.

[57] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, Cambridge, 2004.

[58] Z. Q. Luo and P. Tseng, “On the convergence of the coordinate descent method for convex differentiable minimization,” *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, Jan. 1992.

[59] X. Hu, Z. Li, V. Kumar, and R. Barndt, “Error floor estimation of long LDPC codes on magnetic recording channels,” *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1836–1839, June 2010.

[60] P. Lee, L. Dolecek, Z. Zhang, V. Anantharam, B. Nikolic, and M. Wainwright, “Error floors in LDPC codes: Fast simulation, bounds and hardware emulation,” in *Proc. IEEE Int. Symp. Inform. Theory*, Toronto, Canada, July 2008, pp. 444–448.

[61] A. Globerson and T. Jaakkola, “Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations,” *Advances in Neural Information Processing Systems*, vol. 21, pp. 553–560, 2007.

[62] D. Sontag, A. Globerson, and T. Jaakkola, “Introduction to dual decomposition for inference,” in *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S. J. Wright, Eds. Cambridge, MA: MIT Press, 2012, pp. 219–254.

[63] B.-H. Kim and H. D. Pfister, “Joint decoding of LDPC codes and finite-state channels via linear-programming,” *IEEE J. Select. Topics in Signal Processing*, vol. 5, no. 8, pp. 1563–1576, Dec. 2011.

[64] R. G. Gallager, *Low-Density Parity-Check Codes.* Cambridge, MA: M.I.T. Press, 1963.

[65] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, “Group formation in large social networks: membership, growth, and evolution,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, Aug. 2006, pp. 44–54.

[66] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, “Feedback effects between similarity and social influence in online communities,” in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2008, pp. 160–168.

[67] B.-H. Kim, A. Yedla, and H. Pfister, “IMP: A message-passing algorithm for matrix completion,” in *Proc. Int. Symp. on Turbo Codes & Iterative Inform. Proc.*, Brest, France, Sept. 2010, pp. 469–473.

[68] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Norwell, MA: Kluwer Academic Publishers, 1992.

[69] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519,

Feb. 2001.

[70] B.-H. Kim, A. Yedla, and H. D. Pfister, "Message-passing inference on a factor graph for collaborative filtering," April 2010, [Online]. Available: http://arxiv.org/abs/1004.1003.

[71] R. Neal and G. E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," *Learning in Graphical Models*, vol. 89, pp. 355–368, 1998.

[72] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. of Uncertainty in Artificial Intelligence*, Stockholm, Aug. 1999, pp. 289–296.

[73] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1257–1264, 2008.

[74] I. Csiszar and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, vol. Supplement, no. 1, pp. 205–237, 1984.

[75] N. Srebro, N. Alon, and T. Jaakkola, "Generalization error bounds for collaborative prediction with low-rank matrices," *Advances in Neural Information Processing Systems*, vol. 17, pp. 1321–1328, 2005.

[76] N. Alon, "Tools from higher algebra," *Handbook of Combinatorics*, vol. 2, pp. 1749–1783, 1995.

[77] A. Montanari, "Estimating random variables from random sparse observations," Sept. 2007, [Online]. Available: http://arxiv.org/abs/0709.0145.

[78] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

APPENDIX A

DERIVATION OF DUAL OF THE JOINT LP DECODING PROBLEM

In this section, we show how to derive the dual of the joint LP decoding problem in Problem-P (in Table I) using the technique of Lagrangian relaxation. Consider the primal LP,

$$\min_{\mathbf{g},\mathbf{w}} \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e}$$

subject to

$$\sum_{\mathcal{B}\in\mathcal{E}_j : i\in\mathcal{B}} w_{j,\mathcal{B}} = \sum_{e:x(e)=1} g_{i,e}, \ \ \forall i \in \mathcal{I}, j \in \mathcal{N}(i)$$

$$\sum_{e:s'(e)=k} g_{i,e} = \sum_{e:s(e)=k} g_{i+1,e}, \ \ \forall i \in \mathcal{I} \setminus N, \ k \in S$$

$$\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} = 1, \ \ \forall j \in \mathcal{J}, \ \ \sum_{e\in\mathcal{T}_p} g_{p,e} = 1, \text{ for any } p \in \mathcal{I}$$

$$w_{j,\mathcal{B}} \geq 0, \ \ \forall j \in \mathcal{J}, \mathcal{B} \in \mathcal{E}_j, \ \ g_{i,e} \geq 0, \ \ \forall i \in \mathcal{I}, \ e \in \mathcal{T}_i.$$

We introduce the Lagrange multipliers $m_{i,j}$ and $n_{i,k}$ for the first two constraints and c_j and r for the last two constraints. For this problem, the Lagrangian dual function $h(\mathbf{c}, \mathbf{m}, \mathbf{n}, r)$ is given by

$$\begin{aligned}\inf_{\mathbf{g}\geq\mathbf{0},\mathbf{w}\geq\mathbf{0}} \Bigg[\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i} b_{i,e}g_{i,e} + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{N}(i)} m_{i,j}\left(-\sum_{e:x(e)=1} g_{i,e} + \sum_{\mathcal{B}\in\mathcal{E}_j:i\in\mathcal{B}} w_{j,B}\right)\\ + \sum_{i\in\mathcal{I}}\sum_{k\in S} n_{i,k}\left(-\sum_{e\in\mathcal{T}_{i+1}} g_{i+1,e}\delta_{s(e)=k} + \sum_{e\in\mathcal{T}_i} g_{i,e}\delta_{s'(e)=k}\right)\\ + \sum_{j\in\mathcal{J}} c_j\left(-\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} + 1\right) + r\left(-\sum_{e\in\mathcal{T}_p} g_{p,e} + 1\right)\Bigg]\end{aligned}$$

where $n_{N,k} = 0$, $\forall k \in S$. Rearranging the objective, we get

$$\begin{aligned}\sum_{j\in\mathcal{J}} c_j + r + \inf_{\mathbf{g}\geq\mathbf{0}}\Bigg[\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{T}_i}\Bigg(b_{i,e} - \sum_{k\in S} n_{i-1,s}\delta_{s(e)=k} + \sum_{k\in S} n_{i,s}\delta_{s'(e)=k}\\ - \sum_{j\in\mathcal{N}(i)} m_{i,j}\delta_{x(e)=1} - r\delta_{i=p}\Bigg) g_{i,e}\Bigg] + \inf_{\mathbf{w}\geq\mathbf{0}}\left[\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}}\sum_{j\in\mathcal{J}}\left(\sum_{i\in\mathcal{B}} m_{i,j} - c_j\right)\right]\end{aligned}$$

where $n_{0,k} = n_{N,k} = 0$, $\forall k \in S$. Finally, we obtain the dual objective as

$$\begin{cases}\sum_{j\in\mathcal{J}} c_j + r, & \text{if } b_{i,e} \geq n_{i-1,s(e)} - n_{i,s'(e)} + \sum_{j\in\mathcal{N}(i)} m_{i,j}\delta_{x(e)=1} + r\delta_{i=p},\ \forall i\in\mathcal{I},\, e\in\mathcal{T}_i\\ & \quad n_{0,k} = n_{N,k} = 0,\ \forall k\in S \text{ and } c_j \leq \sum_{i\in\mathcal{B}} m_{i,j}\ \forall j\in\mathcal{J},\, \mathcal{B}\in\mathcal{E}_j\\ -\infty, & \text{otherwise}\end{cases}$$

The dual optimization problem is then

$$\max_{\mathbf{m},\mathbf{n}} \sum_{j\in\mathcal{J}} c_j + r$$

subject to

$$r \leq b_{p,e} - \sum_{j\in\mathcal{N}(p)} m_{p,j}\delta_{x(e)=1} - n_{p-1,s(e)} + n_{p,s'(e)},\ e\in\mathcal{T}_p$$

$$b_{i,e} - \sum_{j \in \mathcal{N}(i)} m_{i,j} \delta_{x(e)=1} \geq n_{i-1,s(e)} - n_{i,s'(e)}, \ \ \forall i \in \mathcal{I} \setminus p, \, e \in \mathcal{T}_i$$

$$c_j \leq \sum_{i \in B} m_{i,j} \, \forall j \in \mathcal{J}, \, \mathcal{B} \in \mathcal{E}_j$$

and

$$n_{0,k} = n_{N,k} = 0, \, \forall k \in S.$$

This linear program can be expressed as Problem-D1 given in Table II.

APPENDIX B

DERIVATION OF DUAL OF THE SOFTENED DUAL PROBLEM

We can analogously derive the dual of the softened dual problem in Problem-DS (in Table IV). Rewrite the Problem-DS as

$$-\min_{\mathbf{m},\mathbf{n}} \frac{1}{K_1}\sum_{j\in\mathcal{J}} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} + \frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p} \mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}}$$

subject to

$$\Gamma_{i,e} \geq n_{i-1,s(e)} - n_{i,s'(e)},\ \forall i\in\mathcal{I}\setminus p,\ e\in\mathcal{T}_i$$

and

$$n_{0,k} = n_{N,k} = 0,\ \forall k\in S,$$

where

$$\Gamma_{i,e} \triangleq b_{i,e} - \delta_{x(e)=1}\sum_{j\in\mathcal{N}(i)} m_{i,j}.$$

We introduce the Lagrange multipliers $g_{i,e}$ for the first constraint. For this problem, the Lagrangian dual function $h(\mathbf{g})$ is given by

$$\inf_{\mathbf{m}\geq\mathbf{0},\mathbf{n}\geq\mathbf{0}} \Bigg[\frac{1}{K_1}\sum_{j\in\mathcal{J}} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} + \frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p} \mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}} + \sum_{i\in\mathcal{I}\setminus p}\sum_{e\in\mathcal{T}_i} g_{i,e}\left(-\Gamma_{i,e}+n_{i-1,s(e)}-n_{i,s'(e)}\right)\Bigg].$$

where $n_{0,k} = n_{N,k} = 0$, $\forall k\in S$. Rearranging the objective, we get

$$
\begin{aligned}
&- \sum_{i\in\mathcal{I}\setminus p}\sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e} \\
&\quad + \inf_{\mathbf{m}\geq\mathbf{0}} \Bigg[\frac{1}{K_1}\sum_{j\in J}\ln\sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} + \sum_{i\in\mathcal{I}\setminus p}\sum_{e\in\mathcal{T}_i} g_{i,e}\delta_{x(e)=1}\sum_{j\in\mathcal{N}(i)} m_{i,j} \\
&\quad + \inf_{\mathbf{n}\geq\mathbf{0}} \Bigg[\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p}\mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}} + \sum_{i\in\mathcal{I}\setminus p}\sum_{e\in\mathcal{T}_i} g_{i,e}\left(n_{i-1,s(e)}-n_{i,s'(e)}\right)\Bigg]\Bigg]
\end{aligned}
$$

We analytically solve the minimization with respect to $\mathbf{n}$ as

$$
\begin{aligned}
&\inf_{\mathbf{n}\geq\mathbf{0}} \Bigg[\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p}\mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}} + \sum_{i\in\mathcal{I}\setminus p}\sum_{e\in\mathcal{T}_i} g_{i,e}\left(n_{i-1,s(e)}-n_{i,s'(e)}\right)\Bigg] \\
&= \sum_{i\in\mathcal{I}}\inf_{\mathbf{n}_i}\Bigg[\left(\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_i}\mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}}\right)\delta_{i=p} + \sum_{e\in\mathcal{T}_i} g_{i,e}\left(n_{i-1,s(e)}-n_{i,s'(e)}\right)\mathbb{1}_{\mathcal{I}\setminus p}(i)\Bigg] \\
&= \sum_{i\in\mathcal{I}}\inf_{\mathbf{n}_i}\Bigg[\left(\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_i}\mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}}\right)\delta_{i=p} \\
&+ \sum_{k\in S} n_{i,k}\left(-\sum_{e:s(e)=k} g_{i,e} + \sum_{e:s'(e)=k} g_{i-1,e}\right)\mathbb{1}_{\mathcal{I}\setminus p}(i)\Bigg] \\
&= \inf_{\mathbf{n}_p}\left[\frac{1}{K_2}\ln\sum_{e\in\mathcal{T}_p}\mathsf{e}^{-K_2\left\{\Gamma_{p,e}-n_{p-1,s(e)}+n_{p,s'(e)}\right\}}\right], \text{ if } \sum_{e:s(e)=k} g_{i,e} = \sum_{e:s'(e)=k} g_{i-1,e},\ \forall i\in\mathcal{I}\setminus p,\ k\in S \\
&\overset{(a)}{=} A
\end{aligned}
$$

where (a) follows from the strong duality argument as in [57, Example 5.5, p. 254]) to obtain

$$
A = \max_{\mathbf{g}_p} - \sum_{e\in\mathcal{T}_p} b_{p,e} g_{p,e} + \sum_{e\in\mathcal{T}_p} g_{p,e}\delta_{x(e)=1}\sum_{j\in\mathcal{N}(p)} m_{p,j} - \frac{1}{K_2}\sum_{e\in\mathcal{T}_p} g_{p,e}\ln g_{p,e}
$$

subject to

$$
\sum_{e:s(e)=k} g_{i,e} = \sum_{e:s'(e)=k} g_{i+1,e},\ \forall i\in\mathcal{I}\setminus N,\ k\in S
$$

and

$$\sum_{e\in\mathcal{T}_p} g_{p,e} = 1 \,\text{and}\, g_{i,e} \geq 0, \;\; \forall i \in \mathcal{I},\, e \in \mathcal{T}_i.$$

Then again, we analytically solve the minimization with respect to $\mathbf{m}$ as

$$\begin{aligned}
&\inf_{\mathbf{m}\geq\mathbf{0}} \left[\frac{1}{K_1} \sum_{j\in J} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} + \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} g_{i,e}\delta_{x(e)=1} \sum_{j\in\mathcal{N}(i)} m_{i,j} \right] \\
&= \sum_{j\in J} \inf_{\mathbf{m}_j} \left[\frac{1}{K_1} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{-K_1\left\{\sum_{i\in\mathcal{N}(j)} m_{i,j}\mathbb{1}_{\mathcal{B}}(i)\right\}} + \sum_{i\in\mathcal{N}(j)} \sum_{e\in\mathcal{T}_i} g_{i,e}\delta_{x(e)=1} m_{i,j} \right] \\
&= \sum_{j\in J} \inf_{\mathbf{m}_j} \left[\frac{1}{K_1} \ln \sum_{\mathcal{B}\in\mathcal{E}_j} \mathsf{e}^{K_1 \sum_{i\in\mathcal{N}(j)} \left\{\sum_{e\in\mathcal{T}_i} g_{i,e}\delta_{x(e)=1} - \mathbb{1}_{\mathcal{B}}(i)\right\} m_{i,j}} \right] \\
&\overset{(c)}{=} \sum_{j\in J} B_j,
\end{aligned}$$

where (c) follows from the strong duality argument as in [57, Example 5.5, p. 254]) to obtain

$$B_j = \max_{\mathbf{w}_j} -\frac{1}{K_1} \sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} \ln\, w_{j,\mathcal{B}}$$

subject to

$$\sum_{\mathcal{B}\in\mathcal{E}_j : i\in\mathcal{B}} w_{j,\mathcal{B}} = \sum_{e:x(e)=1} g_{i,e} \,\forall i \in \mathcal{N}(j),\; \sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} = 1, \,\text{and}\, w_{j,B} \geq 0,$$

Finally, the dual optimization problem is

$$- \quad \max_{\mathbf{g},\mathbf{w}} - \sum_{i\in\mathcal{I}} \sum_{e\in\mathcal{T}_i} b_{i,e} g_{i,e} \quad - \quad \frac{1}{K_1} \sum_{j\in J} \sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} \ln\, w_{j,\mathcal{B}} \quad - \quad \frac{1}{K_2} \sum_{e\in\mathcal{T}_p} g_{p,e} \ln\, g_{p,e}$$

subject to

$$\sum_{\mathcal{B}\in\mathcal{E}_j : i\in\mathcal{B}} w_{j,\mathcal{B}} = \sum_{e:x(e)=1} g_{i,e}, \;\; \forall i \in \mathcal{I}, j \in \mathcal{N}\left(i\right)$$

$$\sum_{e:s'(e)=k} g_{i,e} = \sum_{e:s(e)=k} g_{i+1,e}, \;\; \forall i \in \mathcal{I} \setminus N,\, k \in S$$

$$\sum_{\mathcal{B}\in\mathcal{E}_j} w_{j,\mathcal{B}} = 1,\ \ \forall j \in \mathcal{J},\ \ \sum_{e\in\mathcal{T}_p} g_{p,e} = 1, \text{ for any } p \in \mathcal{I}$$

$$w_{j,\mathcal{B}} \geq 0,\ \ \forall j \in \mathcal{J},\, \mathcal{B} \in \mathcal{E}_j,\ \ g_{i,e} \geq 0,\ \ \forall i \in \mathcal{I},\, e \in \mathcal{T}_i$$

and can be expressed as Problem-PS given in Table V.

VITA

Byung Hak Kim received his B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, Korea, in 2000 and 2002, respectively. In Fall 2011, Mr. Kim graduated with his Ph.D. degree in electrical engineering from Texas A&M University, College Station.

During the summer of 2007, he interned at Qualcomm, Inc. in San Diego. In 2012, he will join the data storage signal processing group at Marvell Semiconductor, Inc. in Santa Clara as a senior DSP engineer. His current research interests include information theory and channel coding with applications in wireless communications, data storage, and statistical inference. He received the Graduate Thesis Award for the best master's thesis in engineering at Korea University in 2002.

Mr.Kim can be reached at Zachry Engineering Center 241A, Texas A&M University, College Station, TX 77843-3128. His email is bhkim@tamu.edu.