LP Decoding for Joint Source-Channel Codes and for the Non-Ergodic Polya Channel*

Adam Cohen, Fady Alajaji, Navin Kashyap, Glen Takahara

***Abstract*— Linear programming (LP) decoding of low-density parity-check codes over discrete memoryless symmetric channels was introduced by Feldman *et al.* in [1]. Here, we extend the LP decoding paradigm by applying it to two additional scenarios: joint source-channel (JSC) coding and decoding over the infinite-memory non-ergodic binary Polya-contagion channel. Simulation results indicate that the JSC LP decoder yields significant gains over the standard LP decoder for non-uniform sources. Simulations also show that the LP decoder for the Polya channel performs moderately well in relation to the ϵ-capacity limit.**

***Index Terms* — LP decoding, joint source-channel coding, non-ergodic Polya channel, systematic and non-systematic codes.**

I. Introduction

In [1], Feldman *et al.* introduce the notion of decoding low-density parity-check (LDPC) codes over memoryless channels using linear programming (LP).[1] They present a straightforward method of defining an efficient LP relaxation of maximum-likelihood (ML) decoding of LPDC codes. LP decoding enjoys certain benefits when compared to standard belief-propagation (BP) decoding techniques; for example, the decoder always converges, and when it converges to an integer solution, it is guaranteed to yield the ML solution. Also, LP decoding is relatively easier to analyze than BP decoding. For example, LP decoding has been shown to correct a constant fraction of errors [2]. Although the algorithm is computationally demanding when compared to iterative decoding algorithms, recent improvements in efficiency have been obtained, *e.g.*, [3].

Here, we extend the LP decoding paradigm to include scenarios other than channel decoding for uniformly distributed memoryless sources transmitted over memoryless channels. First, we apply LP decoding in the context of joint source-channel (JSC) decoding by using systematic LDPC codes for non-uniform memoryless sources sent over binary memoryless channels. The LP decoder's cost function is modified to incorporate the *a-priori* source information. This scheme is further enhanced by using a non-systematic code obtained by puncturing a lower-rate systematic code and using an "extended decoding polytope." As expected, simulation results demonstrate gains vis-a-vis the standard LP decoder which does not incorporate the source statistics.

Next, we discuss the application of LP decoding to the non-ergodic infinite-memory binary Polya-contagion channel, introduced in [4]. It is shown in [4] that this channel belongs to the class of averaged channels with memory, admits a closed-form expression for its ϵ-capacity, and has a straightforward formulation for ML decoding. Thus, the infinite-memory Polya channel provides an interesting tool for modeling[2] non-ergodic fading channels since the class of averaged channels with memory has recently been actively studied in the context of (non-ergodic) wireless fading channels and their outage capacity [6]. Simulation results demonstrate that the proposed LP decoder performs well vis-a-vis the channel's ϵ-capacity.

II. LP Decoding for JSC Codes

The recent work of Feldman, Wainwright and Karger [1] shows that ML decoding of a binary linear code over a discrete memoryless channel can be formulated as an LP which can be "relaxed" to a form with easily expressible constraints and with manageable complexity. Given a length-n code C, received n-vector $\underline{y}$ and a spanning subset H of the dual code, $C^{\perp}$, relaxed LP decoding is performed by selecting the decoded vector $\underline{\hat{x}}$ to achieve the LP optimum $\max_{\underline{x}\in\mathcal{Q}(H)} \underline{\gamma}\cdot\underline{x}$, where $\underline{\gamma}=(\gamma_1,\ldots,\gamma_n)$ is defined by $\gamma_i = \log\left(\frac{P(y_i|x_i=1)}{P(y_i|x_i=0)}\right), 1\le i\le n$. The relaxed codeword polytope $\mathcal{Q}(H)$ is defined as $\mathcal{Q}(H)=\bigcap_{\underline{h}\in H}\mathcal{P}(\underline{h}^{\perp})$, where $\mathcal{P}(\underline{h}^{\perp})$ is the convex hull in $\mathbb{R}^n$ of the code $\underline{h}^{\perp}=\{\underline{c}\in\{0,1\}^n : \underline{h}\cdot\underline{c}\equiv 0 \pmod 2\}$. It is easy to show that the set of integral vertices of $\mathcal{Q}(H)$ coincides exactly with C.

In the following, we assume a non-uniform, memoryless source with the probability of 1 and 0 being p_1 and $p_0=1-p_1$, respectively. The source is channel-coded using a systematic (n,k) LDPC code and transmitted over a memoryless binary symmetric channel (BSC), with $\underline{y}$ being the received vector.

Decoding Systematic Codes: The optimal decoding rule (in the sense of minimizing the codeword error probability) is the maximum *a-posteriori* probability (MAP) rule. The following set equality can be shown, and hence MAP decoding can be performed by selecting the decoded vector to be in the set on the right hand side of

$$\arg\max_{\underline{x}\in C} P(\underline{x})P(\underline{y}|\underline{x}) \quad = \quad \arg\max_{\underline{x}\in C} \underline{\gamma'}\cdot\underline{x},$$

where $\underline{\gamma'}=(\gamma'_1,\ldots,\gamma'_n)$ is defined as

$$\gamma'_i = \begin{cases} \log\left(\frac{p_1}{p_0}\right)+\gamma_i, & \text{for } 1\le i\le k \\ \gamma_i, & \text{for } k<i\le n. \end{cases}$$

*This work was supported in part by NSERC of Canada. All authors are with the Department of Mathematics and Statistics, Queen's University, Kingston, ON K7L 3N6, Canada. (email:{adamcohen, fady, nkashyap, takahara}@mast.queensu.ca).

[1] LP is used throughout to stand for both "Linear Programming" and "Linear Program," and this will be clear from the context.

[2] Note that the finite-memory version of the Polya channel and its generalization based on a finite queue have been recently shown to accurately model ergodic Rician fading channels, e.g., see [5].

and γ_i is the same as above.

The cost function $\underline{\gamma}'$ is linear in the variables x_i and accounts for the *a-priori* probability of a given codeword. Note that this cost function depends on the code being systematic. Given a cost function of this form, we can convert the MAP problem to an LP by constructing the relaxed codeword polytope, $\mathcal{Q}(H)$, with H corresponding to the rows of a parity-check matrix of C, and optimizing over that polytope. So, we can select a vector $\hat{\underline{x}}$ in the set

$$\arg\max_{\underline{x}\in\mathcal{Q}(H)} \underline{\gamma}' \cdot \underline{x}. \tag{1}$$

If $\hat{\underline{x}}$ is an integral vector, then it is selected as the decoded vector, and it is known that this is the MAP codeword. This follows as the integral vertices of $\mathcal{Q}(H)$ coincide exactly with C. In the case that $\hat{\underline{x}}$ is non-integral, we simply round to get the nearest integral vector as our decoding selection. Rounding $\hat{\underline{x}}$ is a heuristic alternative to declaring decoding failure.

Extended Polytope: The proposed technique works; however, it has been demonstrated (see [7],[8] and other works) that in scenarios with strong non-uniformity at the source, systematic codes perform worse than non-systematic codes.

We next modify the above scheme in an attempt to account for this. One way to incorporate the *a-priori* codeword information into an LP decoder without transmitting a systematic code is to encode using a systematic code of rate lower than desired, and then puncture the systematic bits before transmission. More precisely, suppose that we wish to use an LDPC code of rate $R = \frac{k}{n}$, and blocklength n. First, we select a systematic $(n+k,k)$ LDPC code, $\widetilde{C}$. Now, suppose we encode a source word $\underline{s}$ using $\widetilde{C}$, but, before transmission, we strip away the first k (systematic) bits from the encoded block. That is, we transmit only the last n bits of the encoded block. Given that the n-vector $\underline{y}$ is received, MAP decoding can be achieved by selecting an $(n+k)$-vector $\hat{\underline{x}}$ in the set

$$\arg\max_{\underline{x}\in\widetilde{C}} \prod_{i=1}^{k} P(x_i) \prod_{i=k+1}^{n+k} P(y_{i-k}|x_i) \quad = \quad \arg\max_{\underline{x}\in\widetilde{C}} \underline{\gamma}^* \cdot \underline{x},$$

where $\underline{\gamma}^* = (\gamma_1^*, \ldots, \gamma_{n+k}^*)$ is defined as

$$\gamma_i^* = \begin{cases} \log\left(\frac{p_1}{p_0}\right), & \text{for } 1 \le i \le k \\ \log\left(\frac{P(y_{i-k}|x_i=1)}{P(y_{i-k}|x_i=0)}\right), & \text{for } k < i \le n+k. \end{cases}$$

Now, if we select $\widetilde{H} \subset \widetilde{C}^{\perp}$ corresponding to the rows of a parity-check matrix of $\widetilde{C}$, we can construct the relaxed polytope $\mathcal{Q}(\widetilde{H})$, as before. Decoding can then be performed by selecting some $\hat{\underline{x}}$ in the set

$$\arg\max_{\underline{x}\in\mathcal{Q}(\widetilde{H})} \underline{\gamma}^* \cdot \underline{x}. \tag{2}$$

As before, if $\hat{\underline{x}}$ is an integral vector, then it is known to be the MAP solution. Essentially, the “extended codeword polytope”, $\mathcal{Q}(\widetilde{H})$, allows the systematic bits (and hence *a-priori* information) to be associated with the punctured codewords during decoding without having to transmit them.

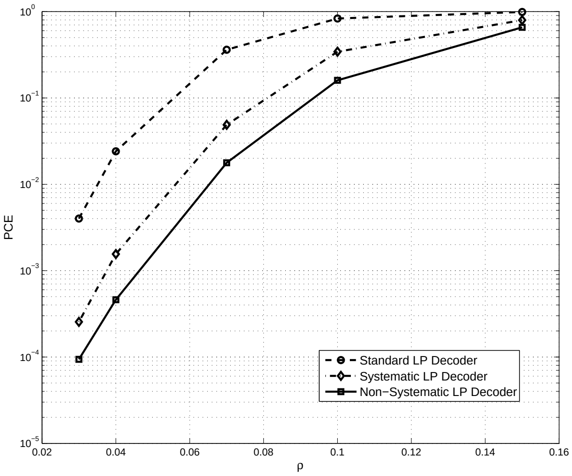

Fig. 1. The top curve shows the PCE performance of a systematic $(200,100)$ LDPC code with the standard LP decoder. The middle curve shows the same code with the decoder (1). The bottom curve shows a systematic $(300,100)$ LDPC code with the first 100 (systematic) bits punctured and decoded using the “extended polytope” decoder (2). All three schemes have an effective redundant rate of $\frac{1}{2}$, $p_1 = 0.9$, and are transmitted over the BSC with bit error rate (BER) ρ.

Simulation Results: The decoders (1) and (2) were implemented and evaluated in terms of the probability of codeword error (PCE). Fig. 1 compares the two methods. At a PCE of 10^{-2}, we see a 20% gain (in terms of the channel bit error rate (BER) ρ) in favour of the “extended polytope” decoder (2) as compared to decoder (1). It should be noted that for less biased values of p_1 (e.g., $p_1 = 0.7$), the punctured code in fact performs worse than the systematic code [9]. Fig. 1 also displays the performance of an LDPC code using the standard LP decoder which does not exploit the non-uniformity of the source distribution. We note that incorporating the source distribution into the decoder yields significant gains, as expected.

III. LP Decoding for the Polya Channel

The infinite-memory Polya-contagion communication channel was introduced in [4, Sections II-V]. It is a non-ergodic binary channel in which the noise is modeled by Polya’s urn scheme for the spread of disease in a population. Specifically, the channel is described by $y_i = c_i \oplus z_i$, where addition is modulo-2, and c_i, z_i and y_i are the input, noise and output bits, respectively at time i. The noise process $\{z_i\}_{i=1}^{\infty}$, generated by Polya’s contagion urn scheme, is a stationary binary process with correlation coefficient $\frac{\delta}{1+\delta}$ and BER $P(z_i = 1) = \rho$, where $\delta \ge 0$ and $0 < \rho < 1$ [4]. If $\delta = 0$, $\{z_i\}$ becomes memoryless and the channel reduces to a BSC with cross-over ρ. For $\delta > 0$, the process $\{z_i\}$ is non-ergodic as its sample average converges (almost surely) to a Beta-distributed random variable with parameters $\frac{\rho}{\delta}$ and $\frac{1-\rho}{\delta}$.

In [4], a formulation for ML decoding over the Polya channel was derived in which either the minimum or maximum Hamming distance codeword (with respect to the received vector) is selected depending on the channel parameters and the distances of the minimum and maximum distance codewords to the received vector. The following Lemma provides a simplification of this formulation for linear codes containing the all-ones codeword.

Lemma: *For a linear code containing the all-ones codeword, if $\rho \le 0.5$ then ML decoding over the Polya channel reduces to minimum Hamming distance decoding.*

The proof is omitted, but the result is obtained by a relatively straightforward extension of the original formulation in [4]. It is possible to formulate minimum distance decoding using an LP. Selecting the decoded vector $\underline{\hat{x}}$ in the set $\arg\min_{\underline{x}\in C} \underline{\gamma}^{+} \cdot \underline{x}$ is equivalent to minimum distance decoding [10], where $\gamma_i^{+} = -1$ if $y_i = 1$, and $\gamma_i^{+} = 1$ if $y_i = 0$.

Having a linear cost function corresponding to minimum distance decoding, we can formulate a relaxed LP decoder corresponding to ML decoding for the Polya channel for linear codes with the all-ones codeword. First, we select an (n, k) LDPC code containing the all-ones codeword (*i.e.*, an LDPC code with even row degree), C, and $H \subset C^{\perp}$ corresponding to the rows of a parity-check matrix of C, and construct its relaxed codeword polytope, $\mathcal{Q}(H)$, as described in Section II. Then, minimum distance decoding can be expressed as an LP by selecting $\underline{\hat{x}}$ in the set $\arg\min_{\underline{x}\in \mathcal{Q}(H)} \underline{\gamma}^{+} \cdot \underline{x}$. The vector $\underline{\hat{x}}$ is rounded (as there is the possibility of obtaining a non-integral vertex) and the result is used as the decoded vector.

Simulation Results: Since there are no known simulation results for decoding over the infinite-memory Polya channel, it is natural for us to compare simulation results to the theoretical performance limit, which is the channel ϵ-capacity.[3]

For a given $\epsilon > 0$, the ϵ-capacity, C_ϵ, of a channel is defined as the maximum ϵ-achievable rate. That is, the maximum rate, R, for which there exist, given sufficiently large block length, codes having rate arbitrarily close to R and probability of error at most ϵ [4]. A closed-form expression for the ϵ-capacity, C_ϵ, of the infinite-memory Polya channel is given in [4, Equation (8)] as a function of ϵ, ρ and δ. For the purpose of comparison, given ρ, δ and rate R', we can determine the ϵ for which R' is the maximum achievable rate (i.e., the ϵ-capacity) by solving $C_\epsilon = R'$ over ϵ. This value of ϵ is thus a theoretical lower bound on the PCE of a rate-R' code over the infinite-memory Polya channel with parameters ρ and δ.

An issue that needs to be carefully considered is exactly how one should simulate communications over a non-ergodic channel. Simulating over a single instance[4] of the channel is certainly not correct, as this would demonstrate only one of many possible behaviours of the channel. Instead, results were obtained by repeatedly initializing the channel and transmitting a fixed number of codewords in each instance. This allowed for the overall (Beta) distribution of possible channel outcomes to be explored.

Fig. 2 compares the theoretical limits to our simulation results over a range of values of the channel BER ρ and for $\delta = 2$. A (200,100) LDPC code under the decoding rule described above was used. For $\delta = 2$ and at a PCE of 10^{-2}, the proposed decoder performs within 39% of the ϵ-capacity in terms of the BER ρ. For comparison purposes, we also include the code performance over the BSC (*i.e.*, when $\delta = 0$), corresponding to the situation where an ideal (infinite-depth) interleaver is applied to the channel. From Fig. 2 we observe that for moderate to high BERs (approximately $\rho > 0.05$), interleaving will result in worse performance (in addition to introducing large encoding/decoding delay); hence, exploiting the channel's memory in the LP decoder can result in improved performance over interleaving.

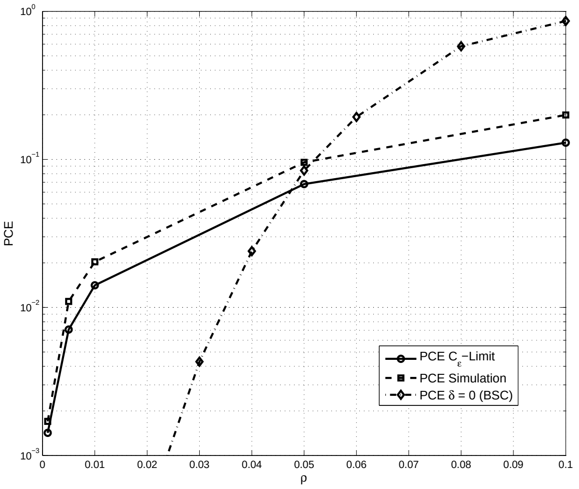

Fig. 2. Simulation vs. ϵ-capacity curves for a systematic (200,100) LDPC code over the Polya channel with $\delta = 2$ and BER ρ. Simulation over the BSC (*i.e.*, the Polya channel with $\delta = 0$), which represents the ideally-interleaved Polya channel.

Finally, we mention that LDPC codes without the all-ones codeword have been tested over the Polya channel using a modified LP decoder which uses approximate minimum and maximum distance decoding (based on the ML decoding formulation in [4]) and have demonstrated superior performance when compared to using approximate minimum distance decoding with codes containing the all-ones codeword [9].

Future work may include the comparison of our JSC LP decoder with BP JSC decoding [8] and the design of a BP type decoder for the non-ergodic Polya channel. Another interesting future direction is the study of JSC LP decoding over the non-ergodic Polya channel.

References

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

[2] J. Feldman, T. Malkin, R. Servedio, C. Stein, and M. Wainwright, "LP decoding corrects a constant fraction of errors." Chicago, U.S.A.: Proc. IEEE Int. Symp. Inform. Theory, 2004, p. 69.

[3] M. H. Taghavi and P. Siegel, "Adaptive linear programming decoding," Proc. ISIT-2006, Seattle, USA, pp. 1374–1378, 2006.

[4] F. Alajaji and T. Fuja, "A communication channel modeled on contagion," *IEEE Tras. Inform. Theory*, vol. 60, no. 6, pp. 2035–2041, 1994.

[5] L. Zhong, F. Alajaji and G. Takahara, "A model for correlated Rician fading channels based on a finite queue," *IEEE Trans. Veh. Technol.*, vol. 57, no. 1, pp. 79-89, Jan. 2008.

[6] M. Effros, A. Goldsmith, and Y. Liang, "Capacity definitions of general channels with receiver side information." Nice, France: Proc. IEEE Int. Symp. Inform. Theory, June 2007, pp. 921–925.

[7] G. C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Transmission of nonuniform memoryless sources via nonsystematic turbo codes," *IEEE Trans. Commun.*, vol. 52, no. 8, pp. 1344–1354, 2004.

[8] A. Alloum, J. Boutros, G. I. Shamir, and L. Wang, "Non-systematic LDPC codes for redundant data." Monticello, IL: Proc. 43rd Allerton Conf. Commun., Control, Computing, 2006, pp. 1879–1888.

[9] A. Cohen, "LP decoding for non-uniform sources and binary channels with memory," Master's thesis, Queen's University, 2008.

[10] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. dissertation, Massachusetts Institute of Technology, 2003.

[3] Note that since the infinite-memory Polya channel is a (non-ergodic) averaged channel with BSC components governed by the Beta distribution, its channel capacity is zero; however, its ϵ-capacity is strictly positive for $\epsilon > 0$ and strictly decreasing to zero as $\epsilon \searrow 0$ [4].

[4] By "single instance" we mean a single realization of the infinite-memory non-ergodic Polya noise process.