Universal Algorithms for Channel Decoding of Uncompressed Sources

Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, Krishnamurthy Viswanathan

Information and Quantum Systems Lab
HP Laboratories Palo Alto
HPL-2007-128
February 12, 2008*

Channel Decoding, Denoising, Discrete Memoryless Channels, Universal algorithms, Lossless compression, Joint source-channel decoding, Soft decoding, Belief propagation.

In many applications, an uncompressed source stream is systematically encoded by a channel code (which ignores the source redundancy) for transmission over a discrete memoryless channel. The decoder knows the channel and the code but does not know the source statistics. This paper proposes several universal channel decoders that take advantage of the source redundancy without requiring prior knowledge of the source statistics.

Internal Accession Date Only
To be published in IEEE Transactions on Information Theory, Vol. 54, No. 5, May 2008 Approved for External Publication

Universal Algorithms for Channel Decoding of Uncompressed Sources*

Erik Ordentlich† Gadiel Seroussi‡ Sergio Verdú§ Krishnamurthy Viswanathan†

February 11, 2008

Abstract

In many applications, an uncompressed source stream is systematically encoded by a channel code (which ignores the source redundancy) for transmission over a discrete memoryless channel. The decoder knows the channel and the code but does not know the source statistics. This paper proposes several universal channel decoders that take advantage of the source redundancy without requiring prior knowledge of the source statistics.

Key words and phrases: Channel Decoding, Denoising, Discrete Memoryless Channels, Universal algorithms, Lossless compression, Joint source-channel decoding, Soft decoding, Belief propagation.

I Introduction

One of the central problems formulated by Claude Shannon [1] is the reliable transmission of a redundant information source through a noisy communication channel. Shannon [1] established that if the source and channel have no memory, in the limit of long block length, encoding can be accomplished without loss of efficiency by removal of the redundancy in the source with a (channel-independent) data compressor followed by the addition of redundancy by a (source-independent) channel encoder. At the receiver, a channel decoder recovers the compressed data and feeds it to a source decompressor that finally recovers the transmitted data. A cornerstone of information theory, the source/channel *separation principle* has been shown to hold in wide generality for stationary sources and channels [2]. When the separation principle holds, the maximum rate of source symbols per channel use is equal to the ratio of channel capacity divided by the source entropy. Analogously, the separation principle also holds in wide generality when the source is to

*Parts of this paper were presented at the 2004 IEEE Int. Symp on Information Theory, Chicago, 2004

†Hewlett-Packard Laboratories, Palo Alto, CA 94304, USA

‡Mathematical Sciences Research Institute, Berkeley, CA 94720, USA. This work was essentially done while the author was with Hewlett-Packard Laboratories, Palo Alto, CA 94304, USA

§Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA

be reconstructed only within some desired distortion D, in which case the fundamental limit is the ratio of channel capacity divided by the source rate-distortion function evaluated at D.

Because of various considerations (such as legacy, layered design, and the difficulty in adapting universal data compressors to short-packet transmission systems) certain existing applications (including third-generation wireless data transmission systems) do not compress the redundant data prior to channel encoding. In other applications, such as the GSM second-generation cellular wireless system, the data compressor leaves some residual redundancy prior to channel encoding [3]. In those cases, the signal sent through the channel incorporates redundancy due to both the channel encoder and the data itself. The receiver may choose to ignore the redundancy due to the source and simply decode the data with a channel decoder. However, by exploiting the residual source redundancy at the receiver it is possible to achieve reliable communication for noisier channels than the code was designed to handle. For example, in a wireless system that transmits uncompressed data, it is possible to lower the minimum signal-to-noise ratio required for reliable communication by exploiting the redundancy in the data. Another application is in internet content distribution to receivers facing channels with a priori unknown erasure rates. The fountain codes [4, 5] are rateless codes whose decoding time depends on the channel erasure rate. By exploiting any source redundancy it is possible to further reduce the decoding time.

The idea of exploiting source redundancy in decoding goes back to Shannon [1], and one of its first practical embodiments is [3]. The approach in [3], and a number of related works [6, 7, 8, 9, 10, 11, 12, 13, 14] is to replace the maximum likelihood channel decoder by a maximum-a-posteriori (MAP) decoder (or an approximation thereof) that incorporates the statistics of the source fed to the channel encoder. Soft-channel decoders provide reliability information, i. e. (estimates of) the posterior marginals of the data, and are typically amenable to incorporate a priori probabilities of the data sequences. Notable examples are the backward-forward dynamic programming (or BCJR) algorithm [15], turbo decoding [16] and belief propagation decoding [17]. If the source is Markovian, then it is possible to take into account its structure at the decoder by augmenting the factor graph of the code (e.g. [18]). This approach faces two serious practical shortcomings: a) the decoding complexity grows exponentially with the source memory and b) knowledge of the statistics of the source is required at the decoder. These shortcomings, which are particularly detrimental in many applications, are not present, at least in principle, when the system is designed following the separation principle as linear-time source encoders/decoders (such as the Lempel-Ziv class of algorithms) are available that are universal (do not require knowledge of the source statistics either at encoder or decoder), and optimal (achieve the entropy rate of the source if it is stationary ergodic, as well as other stronger individual-sequence optimality properties) [19].

Recently, [20, 21] propose decoders that use a Krichevsky-Trofimov estimate of a biased coin in

order to take advantage of the unknown bias of the iid uncompressed encoded data. For general sources with memory, the first universal approach to harness both the redundancy in the channel code and the redundancy in the data without prior knowledge of the statistical structure of the data was introduced in the conference version of the present paper [22]. Another approach for sources with memory, employing the Burrows-Wheeler Transform and segmentation of the transformed data into piecewise-stationary memoryless sources (see also [23]), was subsequently considered in [24]. Reference [25] proposes the design of non-systematic turbo codes with a special property (quick look-in) that allows universal exploitation of the source redundancy at the receiver.

In this paper we propose a new approach to decoding of channel-encoded noisy uncompressed (or partially compressed) discrete sources that requires no prior knowledge of the statistical structure of the data. To motivate our general approach, consider the simple special case of the setting in which no channel code is used prior to transmission. In this case, the source is connected directly to the channel, and the task of the decoder is to "denoise" the output knowing the channel but without prior knowledge of the source. This is the problem of discrete universal denoising considered in [26], which proposed a linear-time universal algorithm, called the DUDE (Discrete Universal DEnoiser), that suffers no asymptotic penalty for universality provided that the source is stationary and the discrete memoryless channel has a full-rank transition probability matrix. Since the source is not protected against the channel noise, even an optimum nonuniversal algorithm that knows the source distribution is able to accomplish only partial denoising of the source. This setup has been applied to various practical problems [27, 28, 29] and extended in several directions: non-discrete channel output alphabets [30], unknown channel belonging to an uncertainty class [31], and channels with memory [32, 33]. The DUDE algorithm consists of the following stages:

1. Empirical conditional distributions of each channel output symbol, given a context of neighboring previous and succeeding symbols, are computed;

2. Using the memoryless and full-rank properties of the DMC, the corresponding conditional distributions of each *clean* source symbol given the corresponding *noisy* context and noisy symbol are estimated;

3. Using a distortion function, and the conditional distributions computed in step 2, a denoising table is computed which gives the denoised symbol as a function of the noisy symbol and its context.

4. The denoising table is applied to the string of channel outputs, to obtain the denoised output string.

Following up on [26], alternative algorithms to carry out Step 1 have been reported in [34, 35].

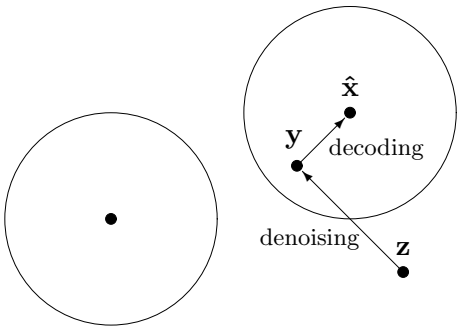

Figure 1: Combined action of denoising and error correction

The conditional marginal distributions of the source given the noisy channel outputs obtained in Step 2 can be considered "soft" information and will play a role in our development. In the sequel, we will refer to the algorithm (Steps 1 and 2) that produces such information as sDUDE. Notice that sDUDE is, in fact, a simplification of DUDE, since it outputs intermediate information and does not carry out Steps 3 and 4 above. Unlike previous works on discrete universal denoising, in this paper we modify the sDUDE to incorporate a priori beliefs of the source symbols. We refer to this "soft-in, soft-out" scheme as the ssDUDE algorithm.

Returning to the original setup where the source is not connected directly to the channel but is channel encoded, let us assume that the channel code is systematic, as is often the case in applications. A natural way to harness the redundancy of both the channel code and the source redundancy is to feed the channel decoder (in addition to the noisy parity-check symbols) not the noisy systematic information symbols but their cleaned-up version as output by the DUDE. The rationale is that if the DUDE is indeed eliminating some of the noise introduced by the channel, then replacing the noisy systematic symbols by their cleaned-up versions results in a block of symbols that should be closer to the transmitted codeword than the received block of channel outputs. The approach is illustrated in Figure 1, which shows the decoding regions (spheres) of the channel code, centered at the codewords; $\mathbf{z}$ is a noisy channel output block, $\mathbf{y}$ is the corresponding block after the systematic information part has been cleaned-up by the DUDE, and $\hat{\mathbf{x}}$ is the codeword output by the channel decoder on input $\mathbf{z}$. The figure depicts how the DUDE "brings in" noisy blocks that would otherwise have been missed by the channel decoder.

In this paper we propose six approaches achieving various performance/complexity tradeoffs:

A. The basic combined scheme outlined above will be referred to as Approach A.

B. Approach B is similar to Approach A except that we use sDUDE instead of DUDE and we now require that the channel decoder have the ability to incorporate soft information about the

marginal distribution of the inputs. For example, belief propagation decoding satisfies this requirement.

C. If the channel decoder not only accepts soft input information but also generates it (e.g. belief propagation), then it can be used in the iterative Approach C, where the first iteration proceeds as in Approach B, and then the soft output produced by the channel decoder is fed to the ssDUDE that incorporates soft information about the transmitted source. A dialog between the modules that exploit the channel code and source redundancies is established which is very much in the spirit of the turbo principle [36]. This approach uses the channel parameters and the empirical joint distribution of noisy symbols, to obtain estimates of the joint distribution of short windows of clean symbols, and these estimates are retained unchanged over multiple iterations of ssDUDE and the decoder.

D. Approach D is also an iterative approach where the first iteration proceeds as in Approach B. From then on, it postulates that the source and the channel decoder decisions at the end of each iteration are connected by a symmetric channel, and computes the marginal distribution of the each source symbol by solving a quadratic equation whose coefficients are obtained from the decoder soft output and the context-dependent counts in the hard-decision block decoded in the last iteration.

E. Approach E is also an iterative approach where the first iteration proceeds as in Approach B. From then on, the context-dependent counts in the hard-decision block decoded in the last iteration are used directly as the marginal distribution of each source symbol. As will be explained in detail in the sequel, it can also be interpreted as a simplification of Approach F below.

F. Approach F is a modification of Approach C where the joint distribution estimates are replaced in every iteration by the empirical joint distribution of short windows of symbols in the hard-decision decoded signal of the previous iteration. Approach E above can be interpreted as restricting a certain summation (marginalization) in Approach F to a single maximally weighted term.

Section II reviews the fundamentals of discrete universal denoising and presents a formal description of the DUDE and sDUDE. A description of ssDUDE is deferred to Section V. Sections III-V describe, respectively, Approaches A-F, and present the results of several experiments with real data, using different codes and channel regimes. The channel parameters, codes, and code rates are chosen to illustrate the effectiveness and wide applicability of the proposed approaches, and to

compare their relative performance. The results show that the various denoising/decoding combined schemes yield significant improvements in the residual error rates of the reconstructed data, compared to either denoising alone, or the traditional error-correction decoding alone. Also, as expected, combined schemes that exploit soft information do significantly better than schemes that do not. Section VI compares the universal Approaches A-F to some natural non-universal counterparts in the enhanced decoding of first-order Markov binary sources. Although the emphasis in this paper is on universally enhancing the decoding of existing, practical families of uncompressed channel encoding schemes, the use of synthetic sources with well-characterized entropy rates also allows us to present in Section VI some relevant fundamental performance bounds on unrestricted, possibly source dependent, encoders and decoders, both with and without compression.

II Discrete Universal Denoising

We denote by x^n and x_m^n, respectively, the sequences $x_1, x_2, \ldots, x_n$ and $x_m, x_{m+1}, \ldots, x_n$. For a vector $\mathbf{v}$, we use the notation $\mathbf{v}[i]$ to denote the i-th entry of $\mathbf{v}$ when subscripts would result in excessively cumbersome notation.

Let $\mathcal{A}$ be a finite alphabet of cardinality $|\mathcal{A}| = M$, taking, without loss of generality, $\mathcal{A} = \{1, 2, \ldots, M\}$. We assume a given *discrete memoryless channel* (DMC) whose transition probability matrix, $\mathbf{\Pi} = \{\Pi(a,b)\}_{a,b\in\mathcal{A}}$, is known. $\Pi(a,b)$ denotes the probability of the channel producing the output symbol b when the input is a. Furthermore, we assume that the $M \times M$ matrix $\mathbf{\Pi}$ is nonsingular.[1] We also assume a given *loss function* (fidelity criterion) $\mathbf{\Lambda} : \mathcal{A}^2 \to [0, \infty)$, represented by a matrix $\mathbf{\Lambda} = \{\Lambda(a,b)\}_{a,b\in\mathcal{A}}$, where $\Lambda(a,b)$ denotes the loss incurred by estimating the symbol a with the symbol b. An example of such a loss function is the *Hamming metric*, i.e., $\Lambda(a,b) = 0$ when $a = b$, and $\Lambda(a,b) = 1$ otherwise. The examples in this paper will use the Hamming metric, although the framework applies to arbitrary cost functions.

Assume a (clean) sequence $x^n \in \mathcal{A}^n$ is transmitted over the channel, and a (noisy) sequence $z^n \in \mathcal{A}^n$ is received. An *n-block discrete denoiser* is a function $\mathbf{Y} : \mathcal{A}^n \to \mathcal{A}^n$ which, on input z^n, produces a (denoised) sequence $y^n = \mathbf{Y}(z^n)$. We let $L_{\mathbf{Y}}(x^n, z^n)$ denote the normalized cumulative loss, as measured by $\mathbf{\Lambda}$, of the denoiser $\mathbf{Y}$ when the observed sequence is $z^n \in \mathcal{A}^n$ and the underlying clean sequence is $x^n \in \mathcal{A}^n$, i.e.,

$$L_{\mathbf{Y}}(x^n, z^n) = \frac{1}{n} \sum_{i=1}^{n} \Lambda(x_i, \mathbf{Y}(z^n)[i]) \,.$$

[1] For simplicity, we assume that the output alphabet of the channel is the same as the input alphabet. The DUDE is defined in [26] more generally for channel transition matrices $\mathbf{\Pi}$ of dimensions $M \times M'$, $M' \geq M$, which are required to be of *full row rank*. All the schemes and results of this paper carry naturally to the non-square case.

When Λ is the Hamming metric, $L_{\mathbf{Y}}(x^n, z^n)$ measures the symbol error rate at the output of the denoiser.

We now describe the specific denoiser defined in [26], namely, the DUDE. Let κ denote a nonnegative integer, which will parametrize the denoiser. We assume that $2\kappa < n$. For each sequence z^n, and pair of strings $\ell^\kappa, r^\kappa \in \mathcal{A}^\kappa$, let $\mathbf{m}(z^n, \ell^\kappa, r^\kappa)$ be an M-dimensional (column) vector of integers whose components are defined as follows:

$$\mathbf{m}(z^n, \ell^\kappa, r^\kappa)[a] = \left|\{\, i \,:\, \kappa < i \le n-\kappa,\ z_{i-\kappa}^{i+\kappa} = \ell^\kappa a\, r^\kappa \,\}\right|, \quad a \in \mathcal{A}\,. \tag{1}$$

The vector $\mathbf{m}(z^n, \ell^\kappa, r^\kappa)$ is thus a histogram of the symbols in z^n occurring with a left (κ-th order) context equal to ℓ^κ and a right context equal to r^κ. Let $\boldsymbol{\pi}_a$ and $\boldsymbol{\lambda}_a$ denote the a-th columns of $\mathbf{\Pi}$ and $\mathbf{\Lambda}$, respectively, let A^T denote the transpose of a matrix A, and $A^{-T} = (A^{-1})^T$ for a nonsingular matrix A. The κ-th order DUDE, denoted $\mathbf{Y}^\kappa_{\mathsf{DUDE}}$, is defined as follows:

$$\mathbf{Y}^\kappa_{\mathsf{DUDE}}(z^n)[i] = \arg\min_{\hat{y}\in\mathcal{A}} \boldsymbol{\lambda}_{\hat{y}}^T \cdot \left(\left(\mathbf{\Pi}^{-T}\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})\right) \odot \boldsymbol{\pi}_{z_i}\right), \quad \kappa < i \le n-\kappa, \tag{2}$$

where $\odot$ denotes the component-wise or Schur product between vectors. The values output for $1 \le i \le \kappa$ and $n-\kappa \le i \le n$ are inconsequential to the asymptotic properties of the denoiser and can be set arbitrarily (e.g., for concreteness, equated to the corresponding locations in z^n). We refer to $\mathbf{Y}^\kappa_{\mathsf{DUDE}}(z^n)$ as the *κ-th order DUDE response* to z^n.

After proper normalization, the vector $\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$ in (2) can be seen as an empirical estimate of the conditional distribution, $P(Z_i|z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$, of a noisy sample given a two-sided noisy context. The vector

$$\left(\mathbf{\Pi}^{-T}\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})\right) \odot \boldsymbol{\pi}_{z_i}\,, \tag{3}$$

in turn, can be interpreted, after normalization, as an estimate of the posterior distribution of the corresponding *clean* sample X_i given the same *noisy* context and Z_i. Letting $\hat{P}_{X_i|Z^{2\kappa+1}}(\cdot|\cdot)$ denote this estimated conditional distribution, the expression (2) corresponds to a MAP estimate of X_i with respect to $\hat{P}_{X_i|Z^{2\kappa+1}}(\cdot|\cdot)$. As evident in (2), this estimation is at the heart of the DUDE, and is produced as an intermediate result by the denoiser.

Example. Consider a binary symmetric channel (BSC) with cross-over probability δ, $0 < \delta < \frac{1}{2}$ and the Hamming cost function. In this case, we have

$$\mathbf{\Pi} = \begin{pmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{pmatrix}, \quad \mathbf{\Pi}^{-1} = \frac{1}{1-2\delta}\begin{pmatrix} 1-\delta & -\delta \\ -\delta & 1-\delta \end{pmatrix}, \quad \mathbf{\Lambda} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

For simplicity, let $\mathbf{n}_i = \mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$. Substituting the values of $\mathbf{\Pi}^{-1}$ and $\mathbf{\Lambda}$ into (2) yields, following simple algebraic manipulations,

$$\mathbf{Y}^\kappa_{\mathsf{DUDE}}(z^n)[i] = \begin{cases} z_i, & \dfrac{\mathbf{n}_i[z_i]}{\mathbf{n}_i[\bar{z}_i]} \ge \dfrac{2\delta(1-\delta)}{1-2\delta(1-\delta)}, \\ \bar{z}_i, & \text{otherwise,} \end{cases} \tag{4}$$

where $\bar{z}_i$ denotes the binary complement of z_i. In words, for each bit z_i in the noisy sequence, the DUDE counts how many bits occurring within the same double-sided context are equal to z_i. If the ratio of the number of such occurrences to those of the value $\bar{z}_i$ is below the threshold $2\delta(1-\delta)/(1-2\delta(1-\delta))$, then z_i is deemed to be an error introduced by the BSC, and is flipped. The corresponding estimate of the posterior distribution of X_i given the noisy context and Z_i is given, using simplified notation, by

$$\left(\begin{array}{c} \hat{P}\left(x_i=0 \mid z_{i-\kappa}^{i+\kappa}\right) \\ \hat{P}\left(x_i=1 \mid z_{i-\kappa}^{i+\kappa}\right) \end{array}\right) = \left(\begin{array}{c} \Big((1-\delta)\mathbf{n}_i[0]-\delta\mathbf{n}_i[1]\Big)\Big((1-\delta)\bar{z}_i+\delta z_i\Big) \\ \Big(-\delta\mathbf{n}_i[0]+(1-\delta)\mathbf{n}_i[1]\Big)\Big(\delta\bar{z}_i+(1-\delta)z_i\Big) \end{array}\right)\gamma\,, \tag{5}$$

where γ is an appropriate normalization constant, and we slightly abuse notation and interpret binary symbols z_i as $\{0,1\}$-valued integers.

It is shown in [26] that if Z^n is the output of a DMC with transition probability matrix $\mathbf{\Pi}$ and input x^n then the normalized loss of the κ-th order DUDE response using a context parameter κ_n that increases sufficiently slowly with n, converges, with probability 1, to that incurred by the best stationary sliding window denoiser, with window size $2\kappa_n+1$, optimized with full knowledge of x^n and z^n. The latter is equivalent to a genie–aided denoiser for which the estimate $\hat{P}_{X_i|Z^{2\kappa_n+1}}$ relied upon in (2) is replaced with the corresponding posterior derived from the empirical joint distribution of $2\kappa_n+1$ blocks in the pair x^n and z^n. This establishes the universality of the DUDE in a so-called *semi-stochastic* setting, where x^n is a given individual sequence, and all the randomness in the problem resides in the channel. Furthermore, in a fully stochastic setting, the semi-stochastic results are leveraged in [26] to show that if the channel input X^n is a stationary process, the normalized loss of the DUDE response converges, with probability one, to the normalized loss of the optimal Bayes response, corresponding to the MAP estimation in (2) but with respect to the actual source and channel distributions.

The DUDE algorithm can be generalized [27, 28, 34, 35] from the foregoing one-dimensional signal and context setting to operate on multi-dimensional signals, such as images, with i taking values in a general index set, and the values of an arbitrary finite group of symbols neighboring z_i, denoted by $\boldsymbol{\eta}_i$, replacing $(z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$ as the context. For example, given a two-dimensionally indexed signal $\{z_{i,j}\}$, the context corresponding to a symbol with index (i,j) might be $\boldsymbol{\eta}_{i,j} = (z_{i-1,j}, z_{i+1,j}, z_{i,j-1}, z_{i,j+1})$. The generalization includes also one-dimensional schemes with asymmetric contexts of the form $(\ell^{\kappa_1}, r^{\kappa_2})$ in which κ_1 and κ_2 are not necessarily equal. We will denote by $K = |\boldsymbol{\eta}_i|$ the overall size (number of neighboring samples) in $\boldsymbol{\eta}_i$, not including the center sample z_i. In the one-dimensional symmetric case, for instance, we have $K = 2\kappa$.

The generalized DUDE operates according to (2) with $\mathbf{m}(\cdot)$ redefined for each context value $\boldsymbol{c}$ as

$$\mathbf{m}(z^n, \boldsymbol{c})[a] = \left|\{\, i \,:\, \boldsymbol{\eta}_i = \boldsymbol{c}, z_i = a\}\right|, \quad a \in \mathcal{A}\,.$$

The interpretation of the corresponding version of (3), after normalization, as an estimate of the posterior distribution of X_i given Z_i and the noisy generalized context $\boldsymbol{\eta}_i$, applies as well. Analogous optimality results hold for the generalized DUDE with respect to corresponding context-based genie aided denoisers in semi-stochastic and fully-stochastic settings. Similarly, the decision rule (2) holds for generalized contexts, and, in particular, in the simplified rule (4) for the binary case, we will denote the counters of symbols in context as $\mathbf{n}_i^{\mathbf{c}}[a],\ a \in \{0,1\}$.

The sDUDE algorithm is a soft-output version of the DUDE which outputs, for each i, an estimate of the conditional probability of X_i given the noisy symbols $Z_{i-\kappa}^{i+\kappa}$. These probabilities can then serve as inputs to soft input channel decoders, as will be the case in the sequel. In such applications, it would not be ideal for the sDUDE to directly output $\hat{P}_{X_i|Z^{2\kappa+1}}(\cdot|z_{i-\kappa}^{i+\kappa})$, as obtained by normalizing (3), since this vector may have negative components, as potentially introduced by the term $\mathbf{\Pi}^{-T}\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$. While the DUDE operation is still well defined in such cases, negative (or even zero) estimates for probabilities can pose serious problems for downstream soft input channel decoding. In the binary case, for example, a zero conditional probability of a symbol being 1 may force a soft-input decoder to output 0 for that symbol, irrespective of any additional information from the channel code redundancy. To avoid these difficulties, the output for i of sDUDE is set to

$$\psi\left(\mathbf{\Pi}^{-T}\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})\right) \odot \boldsymbol{\pi}_{z_i}\,, \tag{6}$$

where $\psi(\cdot)$ is a smoothing function that outputs a vector that is close to its input after normalization and has sufficiently positive components. There are many potentially good choices for $\psi(\cdot)$. In this paper, we focus on soft decoding of *binary* sources and channels, and for the corresponding two dimensional input and output vectors, the following (sum preserving) choice for ψ was found to give good results:

$$\psi([x,y]^T) = \begin{cases} [1, x+y-1]^T & \text{if } x < 1 \\ [x,y]^T & \text{if } 1 < x \text{ and } 1 < y \\ [x+y-1, 1]^T & \text{if } y < 1 \end{cases} \tag{7}$$

Note that the sum of the components of $\mathbf{\Pi}^{-T}\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$ is equal to the sum of the components of $\mathbf{m}(z^n, z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa})$, which is the total number of occurrences of the context $z_{i-\kappa}^{i-1}, z_{i+1}^{i+\kappa}$. Thus $\psi([x,y]^T)$, as defined in (7), ensures that each symbol value has an estimated "clean" symbol count of at least 1, per context. Other values for this minimum count could be used, but a value of 1 was found to work well.

For ease of exposition, in the sequel, the various DUDE-enhanced decoders are described in a one-dimensional setting and the term DUDE refers to a denoiser that operates according to (2), for each i, while sDUDE is a modification that outputs (6), with $\psi(\cdot)$ as in (7). Though not explicitly presented, the DUDE-enhanced decoders generalize easily to multi-dimensional settings, with the DUDE and sDUDE steps replaced by their appropriate multi-dimensional generalizations, along the

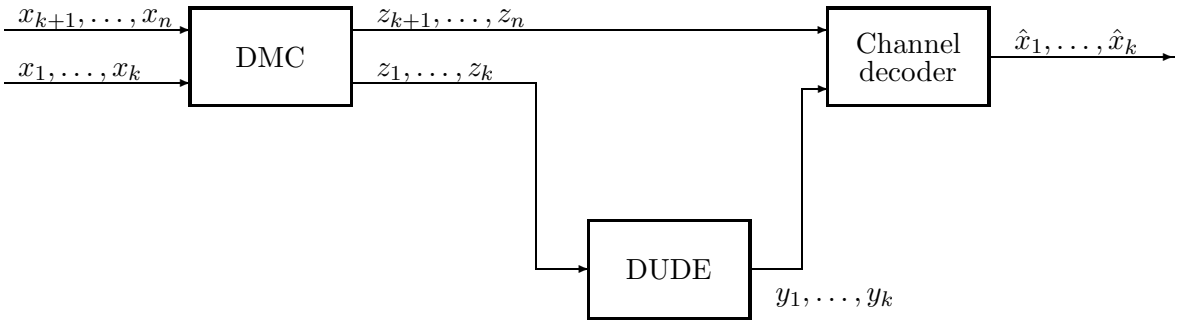

Figure 2: Approach A

lines described above. We note that some of the experimental results on binary images presented in the sequel are based on such generalizations.

Choice of context size. It follows from the results of [26] that a choice of $\kappa = \kappa_n$, with κ_n increasing slowly enough with n, or the corresponding choice of overall context size in the case of the generalized DUDE schemes, guarantees asymptotic convergence of the DUDE to optimal denoising performance. The sufficient condition found in [26] still leaves a very broad range of choices of context size; various heuristics for choosing a good context size for the DUDE have proven very effective in practice. These heuristics are based on using an observable parameter as a proxy for the denoising performance, and optimizing the context size based on the proxy. A heuristic described in [26] suggests using the *compressibility* of the denoised sequence (using a universal compressor), and is based on the empirical observation that when the sequence is denoised with the optimal value of the context size, the denoised sequence exhibits a local minimum in compression ratio. This heuristic has proven effective, in practice, in finding the best values of the context size for a wide range of practical data sets, and was adopted also for the empirical examples described later in this paper. More principled techniques, based on an unbiased estimator of the DUDE loss, are described in [34].

III Approach A: DUDE-enhanced decoding

III-1 The combined scheme

Our first approach is illustrated in Figure 2. The error-correcting code is a systematic code of block length n whose symbols belong to a discrete finite alphabet $\mathcal{A}$. The sequences $x_1^k \in \mathcal{A}^k$ and $x_{k+1}^n \in \mathcal{A}^{n-k}$ denote the noiseless systematic information and parity check symbols respectively. The sequence $z_1^n \in \mathcal{A}^n$ consists of the corresponding noisy symbols obtained when x_1^n is transmitted over a discrete memoryless channel. The DUDE module takes z_1^k as input and outputs the block of symbols $y_1^k \in \mathcal{A}^k$, which is the denoised version of z_1^k . The channel decoder takes (y_1^k, z_{k+1}^n)

as input, and outputs the decoded block $\hat{x}_1^k$, which is the system's reconstruction of the noiseless information symbols. To improve denoising efficiency, the DUDE module can use statistics from multiple code blocks to derive the denoising table for z_1^k. All of the experiments in the sequel employ this technique, in which *all* code blocks corresponding to a data set are used to build statistics. In practice, the accumulation of statistics can be limited to occur in a causal or limited delay fashion across previously observed code blocks. Various fading-memory and periodic reset strategies can be employed to fine-tune this mechanism. In addition, further fine-tuning of denoiser parameters, such as context size and assumed channel parameters, can be achieved by monitoring the end-to-end performance of the denoiser/decoder system as reflected in information provided by the decoder, such as current rate of nonzero block syndromes, or rate of block decoding failures. The parameters of the denoiser could be dynamically adjusted, seeking to decrease these rates.

Throughout this paper, the criterion for assessing the performance of the proposed schemes, including Approach A, is symbol (bit) error rate. If the target were block error rate instead, it might impact the choice of the constituent channel codes and channel decoders in the various approaches, but it would not change the role or operation of the DUDE. The latter would continue to target the reduction of symbol error rate in the systematic portion of the received codeword, thereby reducing the effective noise handled by the channel decoder, which, in turn, could target block error rate.

The baseline denoising/channel decoding scheme of this section was tested on various data sets, under different noise regimes. The results, showing the effectiveness of the scheme, are presented in Sections III-4 and III-3. First, we describe the data sets, which are used throughout the remainder of the paper.

III-2 Test data sets

Three test data sets are used in the examples given in this and subsequent sections of the paper. They are briefly described below, together with the DUDE settings used to denoise their BSC-corrupted versions.

Data Set 1 (Figure 3). A 896×1160 binary (B/W) half-toned rendering of a continuous tone grayscale picture. A one-dimensional DUDE scheme with contexts of overall size of $K = 12$ and $K = 14$ was used to denoise this image, with the smaller context sizes used at the higher noise levels.

Data Set 2 (Figure 4). The first page of a scanned version of [1]. The dimensions of this B/W image are 1800×2104. This image is best denoised with a two-dimensional DUDE scheme, where the context $\boldsymbol{\eta}_{i,j}$ of a bit $z_{i,j}$ is given by a collection of K bits that correspond to the pixels neighboring the pixel corresponding to $z_{i,j}$. All experiments were run with the context size $K = 12$ so that $\boldsymbol{\eta}_{i,j} = \{z_{i\pm1,j\pm1}, z_{i\pm2,j}, z_{i,j\pm2}\}$.

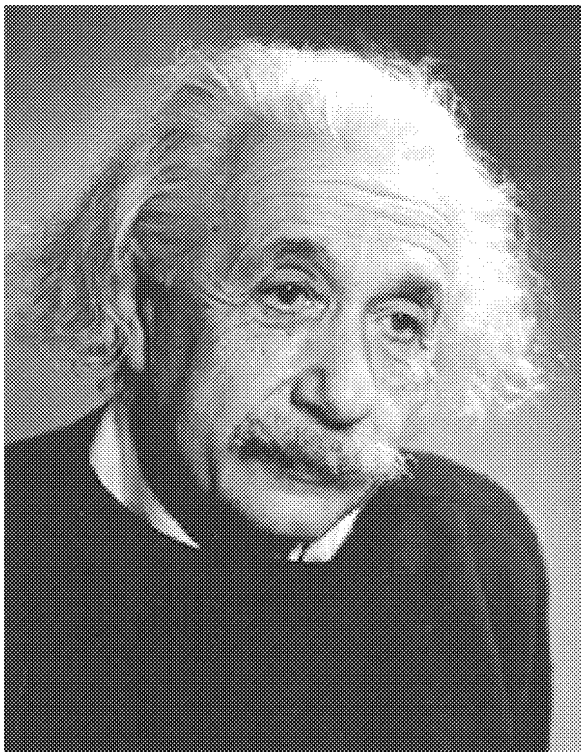

Figure 3: Half-toned binary image (Data Set 1)

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one *selected from a set* of possible messages. The system must be designed to operate for each possible selection, not just the one which will actually be chosen since this is unknown at the time of design.

If the number of messages in the set is finite then this number or any monotonic function of this number can be regarded as a measure of the information produced when one message is chosen from the set, all choices being equally likely. As was pointed out by Hartley the most natural choice is the logarithmic function. Although this definition must be generalized considerably when we consider the influence of the statistics of the message and when we have a continuous range of messages, we will in all cases use an essentially logarithmic measure.

The logarithmic measure is more convenient for various reasons:

1. It is practically more useful. Parameters of engineering importance

[1] Nyquist, H., "Certain Factors Affecting Telegraph Speed," *Bell System Technical Journal*, April 1924, p. 324; "Certain Topics in Telegraph Transmission Theory," *A. I. E. E. Trans.*, v. 47, April 1928, p. 617.

[2] Hartley, R. V. L., "Transmission of Information," *Bell System Technical Journal*, July 1928, p. 535.

Published in THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 27, pp. 379-423, 623-656, July, October, 1948

MONOGRAPH B-1598
Reissued December, 1957

Figure 4: Scanned binary image (Data Set 2)

Notice that although both of the above data sets represent binary images, their fine structures are very different. As noted in [26] and in [27], although effective denoisers for scanned B/W images were known before [26], none of the previously known schemes was effective for half-toned images such as that of Figure 3.

Data set 3. A large text file containing HTML language code, downloaded from one of the popular web portals serving web-enabled wireless devices [37]. The file size is approximately 140 Mbits. Since the file contents consists mostly of printable text, it would be natural to regard it as a string over an alphabet of 8-bit symbols ($M = 256$), and apply a DUDE scheme over the same alphabet, rather than a binary one. However, our experiments on this file showed that binary DUDE schemes were as effective in denoising it as schemes defined over the larger alphabet. At lower noise levels ($\delta \leq 0.050$), a plain binary denoiser that ignores the character structure suffices. The preferred context size in this regime was $K = 24$. At higher noise levels, a "character-aware" binary scheme provided the best performance. This scheme uses a different statistical model for each bit position in a character, and asymmetric context patterns with the right-side context always aligned to a character boundary. The best context sizes in this regime varied from $K = 23$ to $K = 16$, with K decreasing toward the highest noise levels tested. The decrease in optimal context size as the noise level increases is well predicted by the theoretical analysis of [26], which exhibits a *model cost* term that increases exponentially in K, with coefficients that increase with δ.

III-3 Experimental results: the high noise regime

Table 1 presents results of experiments with a scheme combining DUDE with a $(255, 155)$ Reed-Solomon (RS) code over GF(256) [38]. The data sets were parsed into 8-bit symbols (in the case of the binary images, after raster-scanning and packing binary pixels into octets) and 155-symbol blocks, and each block was encoded with a standard RS encoder. The encoded data was fed into a BSC, and the channel output was processed according to the scheme of Figure 2, using a binary DUDE and a standard RS decoder matching the encoder. The decoder uses a standard full-error correction algorithm, correcting all patterns of up to 50 symbols. Most uncorrectable error patterns are detected by the decoder, in which case the data is left untouched.

In Table 1, we compare the performance, in terms of output bit error rate (BER), of the combined denoiser/decoder scheme against that of the channel decoder alone, or the denoiser alone, at various values of the BER of the BSC. The results are also plotted in graphical form in Figure 5. As expected, the performance of the error-correction decoder is independent of the data, up to statistical random sample variations. The performance of the denoiser, and of the combined scheme, on the other hand, are strongly data-dependent. In all cases, however, the combined scheme significantly outperforms either the decoder or the denoiser alone, except in the very high

	Data set 1 (Fig. 3)			Data set 2 (Fig. 4)			Data set 3 (HTML)		
channel δ	Decoded	Denoised	Denoised/ Decoded	Decoded	Denoised	Denoised/ Decoded	Decoded	Denoised	Denoised/ Decoded
0.020	0.0005	0.0075	0.0000	0.0005	0.0018	0.0000	0.0005	0.0083	0.0000
0.025	0.0078	0.0092	0.0000	0.0078	0.0022	0.0000	0.0078	0.0110	0.0001
0.030	0.0240	0.0109	0.0001	0.0238	0.0026	0.0000	0.0238	0.0139	0.0007
0.040	0.0399	0.0143	0.0038	0.0399	0.0035	0.0002	0.0400	0.0204	0.0090
0.050	0.0500	0.0180	0.0132	0.0500	0.0043	0.0012	0.0500	0.0259	0.0224
0.060	0.0600	0.0219	0.0206	0.0600	0.0051	0.0031	0.0600	0.0333	0.0330
0.080	0.0801	0.0301	0.0300	0.0800	0.0070	0.0065	0.0800	0.0494	0.0494
0.100	0.1000	0.0398	0.0398	0.1001	0.0092	0.0091	0.1000	0.0662	0.0662
0.200	0.2001	0.1131	0.1131	0.2002	0.0272	0.0272	0.2000	0.1508	0.1508

Table 1: Results for Approach A with a $(255, 155)$ RS code over GF(256)

noise region, where the error rate overwhelms the RS decoder, and all improvement in BER is due to the denoiser. Under the assumption that the BSC is a binary-quantized Gaussian channel, the combined denoiser/decoder scheme yields, at an output BER of 10^{-3}, a coding gain ranging from 0.8dB for Data Set 3 to 1.8dB for Data Set 2, over the channel decoder alone.

The BER values in these experiments are relatively high, and the code utilized is, accordingly, of relatively high redundancy. The low noise/low redundancy regime is studied next in Section III-4.

III-4 The low noise regime

In this section, we consider data corrupted by channels of relatively low BER (ranging from 10^{-4} to 10^{-2}), and the channel code is a high-rate $(255, 235)$ Reed-Solomon code. The setting is otherwise identical to that of Section III-3. Results for Data Set 3 are presented in Table 2.

The results show the expected advantage of the combined denoiser/decoder scheme over the RS decoder alone for channel BERs from around $2{\cdot}10^{-3}$ and above, but the advantage narrows at lower BERs, and the combined scheme is actually *worse* than the decoder alone at BER level 10^{-3} and below (see boldface entries in Table 2). This deterioration occurs despite the fact that the DUDE is doing its job, and, as the table shows, the BER at the output of the denoiser is significantly lower than the channel BER even in the "problematic" region.

Although the observed deterioration might seem counterintuitive at first, its cause is not hard to find: the DUDE introduces memory into the channel "seen" by the decoder. The composite channel is no longer characterized by the marginal BER alone, which is nevertheless the loss function that the denoiser is attempting to minimize. Intuitively, the action of the DUDE "clusters" errors in a way that makes a certain fraction of the codewords suffer heavy corruption, while leaving (maybe many more) codewords less corrupted than the average. However, reducing the number of errors

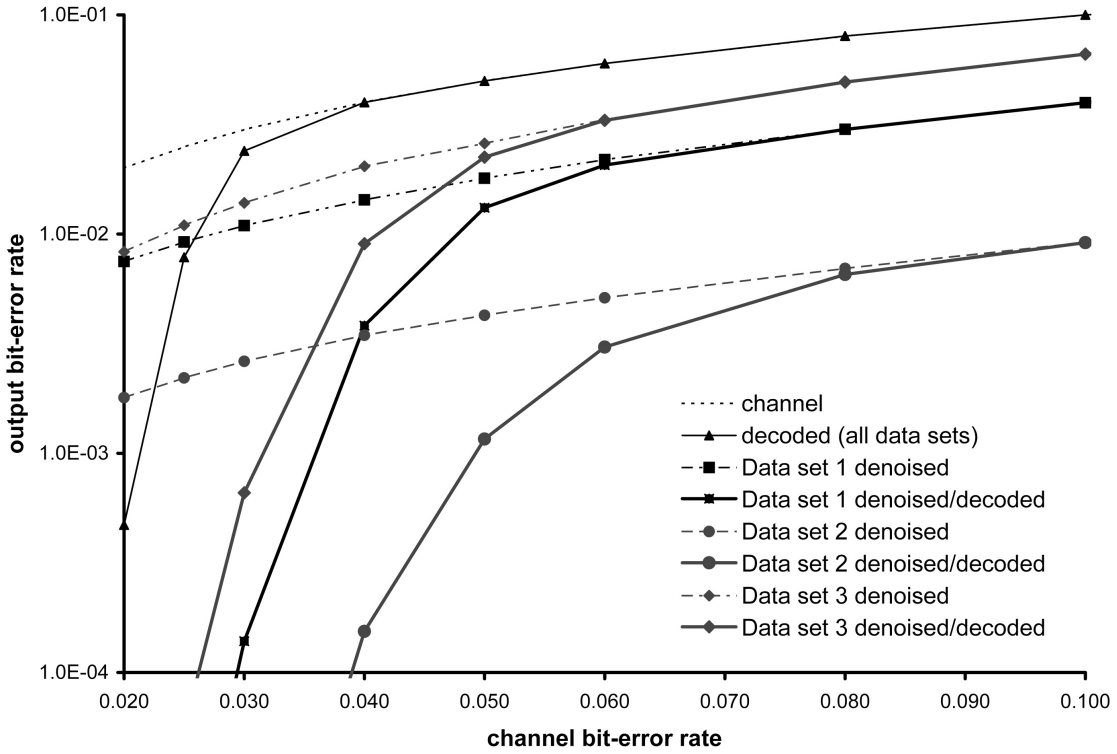

Figure 5: Plot of output BER vs. channel BER for Approach A on test data sets 1–3

channel δ	Decoded	Denoised	Denoised/ Decoded
7.50e-04	**4.309e-09**	3.275e-04	**2.449e-08**
1.00e-03	**3.679e-08**	4.189e-04	**4.616e-08**
1.25e-03	3.984e-07	5.034e-04	7.749e-08
1.50e-03	1.575e-06	5.899e-04	1.304e-07
1.75e-03	5.870e-06	6.731e-04	2.057e-07
2.00e-03	1.629e-05	7.619e-04	4.812e-07
2.50e-03	7.932e-05	9.343e-04	1.645e-06
3.00e-03	2.502e-04	1.107e-03	4.751e-06
4.00e-03	1.133e-03	1.450e-03	2.327e-05
5.00e-03	2.711e-03	1.810e-03	7.605e-05
6.00e-03	4.555e-03	2.180e-03	1.770e-04
7.00e-03	6.237e-03	2.561e-03	3.410e-04
8.00e-03	7.657e-03	2.939e-03	5.950e-04

Table 2: Results for Approach A with a $(255, 235)$ RS code on Data Set 3 (HTML).

in codewords that were correctable to start with does not help the decoder, whereas making some codewords uncorrectable will defeat it, resulting in an overall higher output BER. The effect was not observed in the high noise regime, where the high-redundancy RS code is fairly powerful in correcting bursts of errors of density well above the average, and whose length exceeds the DUDE memory (of which the overall context size is a good reference measure).

Ideally, in the system of Figure 1, the loss function minimized by the DUDE should be adapted to the characteristics of the channel decoder. Although implementing this adaptation optimally and precisely seems impractical, a closer look at the nature of the residual errors of the DUDE will allow us to address the problem in practice. Errors remaining at the DUDE's output are of two kinds: *errors of omission* are original channel errors that the DUDE failed to correct; *errors of commission* are symbols that went through the channel unscathed, but the DUDE decided to flip anyway, following its decision rule (4). Clearly, it is the errors of commission that are defeating the decoder and the overall system performance; the DUDE does not increase the local density of errors by making errors of omission. Thus, the output of the DUDE would be better matched to the channel decoder if it were possible to penalize errors of commission more than errors of omission in the loss function minimized by the denoiser. It turns out that a minor modification of DUDE can implement this unequal loss function.

Consider a loss function Λ with three inputs that assigns different losses to errors of commission and errors of omission as follows:

$$\Lambda(x,\hat{x},z) = \begin{cases} \ell_o & x \neq \hat{x}, \hat{x} = z \\ \ell_c & x \neq \hat{x}, \hat{x} \neq z \\ 0 & x = \hat{x} \end{cases}$$

Letting $\hat{P}(\cdot|\boldsymbol{\eta}_i)$ denote the DUDE-estimated posterior given the context of z_i (but not the noisy symbol z_i itself), the Bayes optimal decision rule is

$$\mathbf{Y}(z_i) = \begin{cases} z_i, & \hat{P}(z_i|\boldsymbol{\eta}_i)\ell_c(1-\delta) > \hat{P}(\bar{z}_i|\boldsymbol{\eta}_i)\ell_o\delta\,, \\ \overline{z}_i, & \text{otherwise.} \end{cases}$$

Denoting $\mathbf{n}_i = \mathbf{m}(z^n, \boldsymbol{\eta}_i)$, and substituting the formula for the estimated posterior

$$\hat{P}(z_i|\boldsymbol{\eta}_i) \sim (1-\delta)\mathbf{n}_i[z_i] - \delta\mathbf{n}_i[\bar{z}_i]$$

(cf. (5)), we obtain, after simplification, the decision rule

$$\mathbf{Y}(z_i) = \begin{cases} z_i, & \dfrac{\mathbf{n}_i[z_i]}{\mathbf{n}_i[\overline{z}_i]} > \dfrac{\delta(1-\delta)}{\alpha(1-\delta)^2 + (1-\alpha)\delta^2}\,, \\ \overline{z}_i, & \text{otherwise.} \end{cases} \tag{8}$$

	DUDE with $\delta' \leq \delta$					DUDE with $\delta' = \delta$	
channel δ	δ'/δ	ℓ_c/ℓ_o	Decoded	Denoised	Denoised/ Decoded	Denoised	Denoised/ Decoded
7.50e-04	0.50	∞	4.309e-09	5.955e-04	8.210e-10	3.275e-04	2.449e-08
1.00e-03	0.60	5	3.679e-08	6.002e-04	4.309e-09	4.189e-04	4.616e-08
1.25e-03	0.62	4	3.984e-07	6.895e-04	2.098e-08	5.034e-04	7.749e-08
1.50e-03	0.75	2	1.575e-06	6.812e-04	8.769e-08	5.899e-04	1.304e-07
1.75e-03	0.75	2	5.870e-06	7.807e-04	2.481e-07	6.731e-04	2.057e-07
2.00e-03	0.75	2	1.629e-05	8.771e-04	4.295e-07	7.619e-04	4.812e-07

Table 3: Results for Approach A with a $(255, 235)$ RS code on Data Set 3 (HTML) with DUDE tuned for a weighted loss function. Results for the standard DUDE are also listed, for comparison.

where $\alpha = \ell_c/(\ell_c + \ell_o)$. For $\alpha = 1/2$ (or $\ell_o = \ell_c$), we recover the original DUDE decision rule (4). As we are interested in the case $\ell_c \geq \ell_o$, the parameter α will vary in the range $\frac{1}{2} \leq \alpha < 1$, with $\alpha \to 1$ as $\ell_c/\ell_o \to \infty$. It is readily verified that for *any* value of α in this range, the rule (8) is equivalent to one of the form of (4), for some value $\delta' \leq \delta$ substituted for δ. Thus, a version of DUDE that penalizes errors of commission can be implemented by simply using the standard DUDE algorithm, but tuning it for a channel parameter δ' below the true channel parameter δ, i.e., a more "conservative" DUDE. In practice, rather than setting an arbitrary ratio ℓ_c/ℓ_o, the best operating value for the ratio $\delta'/\delta \leq 1$ can be tuned by monitoring the performance of the channel decoder, and adjusting the parameter so that the rate of decoding failures is minimized.

Results for a denoising/decoding scheme incorporating the DUDE tuned for unequal error weighting are shown in Table 3 for the problematic range of Table 2, and in Figure 6 for the whole range of the table. In Table 3, we list the ratio between the parameter δ' used by the DUDE to the actual channel parameter δ, and the corresponding ratio ℓ_c/ℓ_o by which errors of commission are over-weighted.

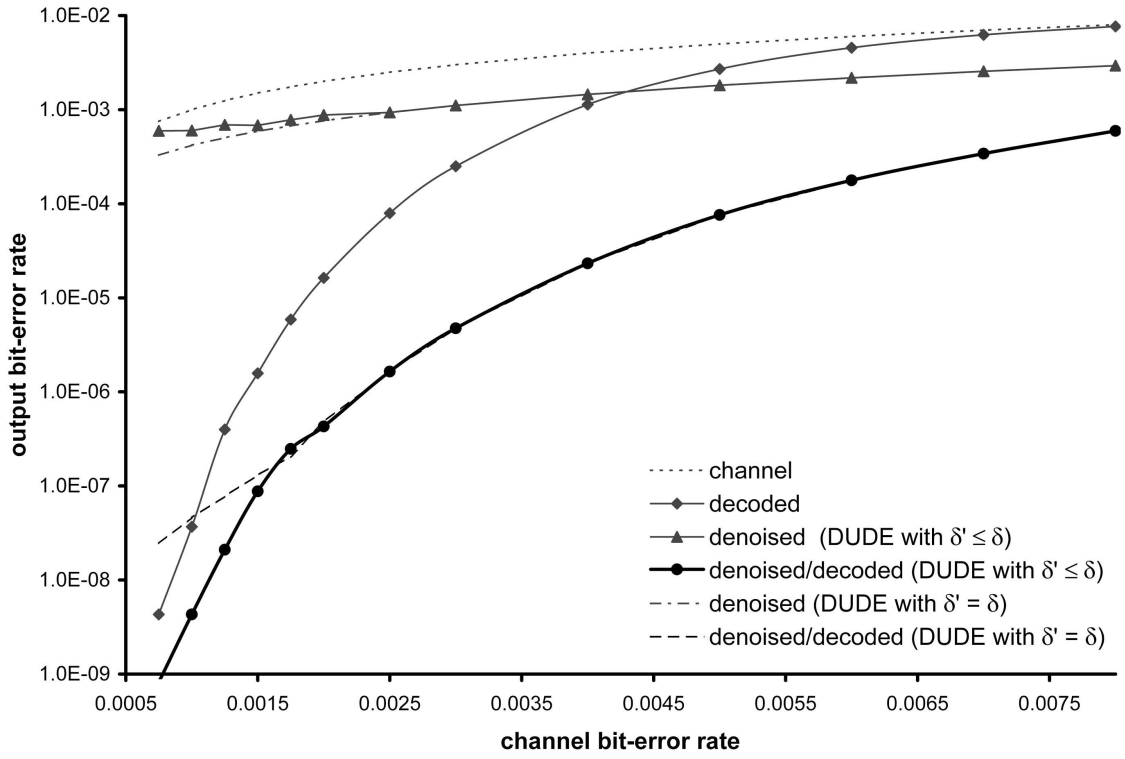

Figure 6: Output BER vs. channel BER for Approach A with a $(255, 235)$ RS code on Data Set 3, with DUDE tuned for a weighted loss function. Results for the standard DUDE are shown for comparison.

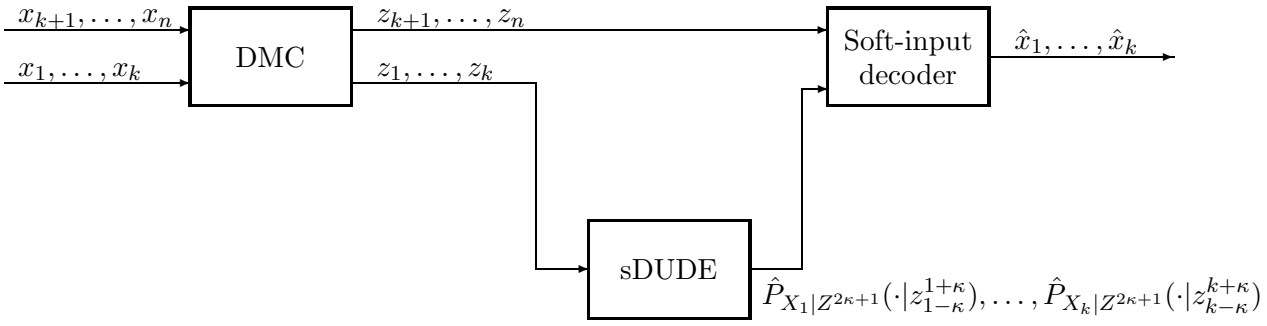

Figure 7: Approach B

IV Approach B: sDUDE-enhanced decoding

In Approach A the noisy information symbols were denoised by the DUDE and the denoised information symbols were input to a decoder along with the noisy parity check symbols. In Approach B, instead of performing a hard denoising using the DUDE, the noisy information symbols are input to the sDUDE, described in Section II, which outputs soft information about each of the noiseless information symbols. This information along with the noisy parity check symbols are sent to a decoder that takes soft-information as input and outputs estimates of the clean symbols.

Approach B is depicted in Figure 7, with notation similar to that of Figure 2 and Section III-1. The sDUDE block takes the noisy systematic symbols z_1^k as input, and outputs *a posteriori* probability estimates $\{\hat{P}_{X_i|Z^{2\kappa+1}}(\cdot|z_{i-\kappa}^{i+\kappa})\}_{i=1}^k$ of the clean symbols $\{X_i\}$ conditioned on a subset of noisy symbols. The soft-input decoder takes these estimates and z_{k+1}^n as input and outputs a sequence of estimates $\hat{x}_1^k$ for the noiseless information symbols.

In general, a soft-input decoder can be viewed as a mapping $g : \mathbb{R}^{|\mathcal{A}|n} \to \mathcal{A}^k$ that takes soft information about the transmitted symbols as input and outputs an estimate of the information symbols. We observe that while Approach B can be easily modified to apply to any soft-input decoder defined as above, the approach is particularly well suited for belief-propagation based decoders [16, 17]. We therefore illustrate the approach with such decoders.

Here we describe a belief propagation decoder that incorporates the additional soft information about the information symbols generated by the sDUDE block. The input to the soft decoder is a sequence $\{\mathcal{L}_i\}_{i=1}^n$ of non-negative $|\mathcal{A}|$-dimensional vectors. Let the information variable nodes of the Tanner graph underlying the belief propagation decoder be denoted by the integers $\{1, 2, \ldots, k\}$ and the parity check variable nodes by $\{k+1, k+2, \ldots, n\}$. Then for $1 \leq i \leq k$

$$\mathcal{L}_i = \hat{P}_{X_i|Z^{2\kappa+1}}(\cdot|z_{i-\kappa}^{i+\kappa}) \tag{9}$$

and for $k+1 \leq i \leq n$

$$\mathcal{L}_i = u_{|\mathcal{A}|} \odot \boldsymbol{\pi}_{z_i} \tag{10}$$

where

$$u_{|\mathcal{A}|} = \left(\frac{1}{|\mathcal{A}|}, \frac{1}{|\mathcal{A}|}, \ldots, \frac{1}{|\mathcal{A}|}\right)$$

is the uniform distribution over $\mathcal{A}$ and $\odot$ represents coordinatewise multiplication. The soft decoder is initialized with these probabilities and a predetermined number of iterations of the belief-propagation decoding algorithm is executed on the Tanner graph of the linear code. For completeness, we describe the decoding.

Denoting the check nodes by the integers $\{1, 2 \ldots n-k\}$, let $\nu_{i\to j}^t \in \mathbb{R}^{|\mathcal{A}|}$ denote the message from the variable node i to the check node j in the t_{th} iteration and let $\mu_{j\to i}^t \in \mathbb{R}^{|\mathcal{A}|}$ denote the message from check node j to variable node i in the t_{th} iteration. We index the message vectors by elements of $\mathcal{A}$. Moreover for any node i, variable or check, let $\mathcal{N}_i$ denote the set of neighbors of i. The rules for updating the messages at the variable and check nodes are given below. At the variable node i, for all $x \in \mathcal{A}$ and $t \geq 0$

$$\nu_{i\to j}^{(t)}[x] = \mathcal{L}_i[x] \prod_{j' \in \mathcal{N}_i \setminus \{j\}} \mu_{j'\to i}^{(t-1)}[x], \tag{11}$$

where $\mu_{j\to i}^{(-1)}[x] = 1$ for all i, j, and x. At the check node j, for $t \geq 0$

$$\mu_{j\to i}^{(t)}[x] = \sum_{\substack{x^{\mathcal{N}_j} \in \mathcal{A}^{|\mathcal{N}_j|}: \\ x_i = x}} 1_j(x^{\mathcal{N}_j}) \prod_{i' \in \mathcal{N}_j \setminus \{i\}} \nu_{i'\to j}^{(t)}[x_{i'}] \tag{12}$$

where $x^{\mathcal{N}_j}$ denotes the tuple of values $\{x_{i'} \in \mathcal{A} : i' \in \mathcal{N}_j\}$ indexed by nodes in $\mathcal{N}_j$, and $1_j(x^{\mathcal{N}_j})$ is equal to 1 for those tuples $x^{\mathcal{N}_j}$ satisfying the parity check for check node j and is equal to 0 for remaining tuples. After a fixed number, say ℓ, of iterations the decisions at the information variable nodes are made according to the function

$$\hat{x}_i = \arg\max_{x \in \mathcal{A}} \mathcal{L}_i[x] \prod_{j' \in \mathcal{N}_i} \mu_{j'\to i}^{(\ell-1)}[x].$$

To demonstrate the improved effectiveness of Approach B over A we describe the results of experiments conducted on the data sets described in Section III-2. In each case, the rows of the image are concatenated and divided into blocks of length 4000. Each block is encoded using a rate-1/4 regular repeat-accumulate (RA) code selected at random from an ensemble. The encoded blocks are transmitted over a binary symmetric channel of known crossover probability and the noisy output from the channel is denoiser-enhanced decoded using the algorithms described in Sections III and IV, with belief propagation comprising the channel decoding block. For the DUDE and sDUDE blocks we use a simple one-dimensional context scheme with context length $K = 12$ for Data Set 1, and the 2D context scheme described in Section III-2 for Data Set 2.

	Output bit error rate			
Channel	Denoised	Decoded	A	B
0.13000	0.05596	0.00001	0.00000	0.00000
0.14000	0.06263	0.00002	0.00000	0.00000
0.15000	0.06992	0.00024	0.00000	0.00000
0.16000	0.07715	0.00455	0.00002	0.00000
0.17000	0.08537	0.02357	0.00038	0.00000
0.18000	0.09375	0.05182	0.00468	0.00002
0.19000	0.10251	0.07925	0.02006	0.00053
0.20000	0.11185	0.10335	0.04278	0.00478
0.21000	0.12075	0.12655	0.06509	0.01756
0.22000	0.13045	0.14800	0.08563	0.03831

Table 4: Results for Approaches A and B with a $(16000, 4000)$ RA code and Data Set 1 (Figure 3).

The results for Data Sets 1 and 2 are presented, respectively, in Tables 4 and 5, and plotted in Figures 8 and 9. Fixing a bit error rate of 10^{-4}, we see that for Data Set 1, Approach A results in a coding gain of 0.7dB over a plain decoder, and Approach B results in a coding gain of 0.7dB over Approach A. For Data Set 2, Approach A results in a coding gain of 1.4dB over a plain decoder, and Approach B results in a coding gain of 0.8dB over Approach A.

V Approaches C–F: Iterative denoising/decoding

In this section, we show how to carry out additional iterations of denoising and decoding beyond Approach B, by combining error correction decoding algorithms that also output *a posteriori* reliability information concerning the information symbols with several alternatives to the denoising stage that can incorporate the decoder-generated soft information. Each subsection below describes an alternative denoising stage and how it interacts with the error correction decoder in the iterative process. While the various approaches are compatible with any soft-input-soft-output decoding algorithm, the descriptions below are tailored to belief propagation decoding in the setting of LDPC codes. All of the approaches start off with an iteration of Approach B (sDUDE), followed by the respective iterative stages. The experimental results for the various approaches and data sets, involving the RA codes and belief propagation decoding of the previous section, are presented in Section V-5, along with comparisons to Approaches A and B for this setting.

V-1 Approach C

The key new component in this approach is the ssDUDE, a DUDE-like denoising stage which can process the decoder generated reliability information along with the noisy information symbols

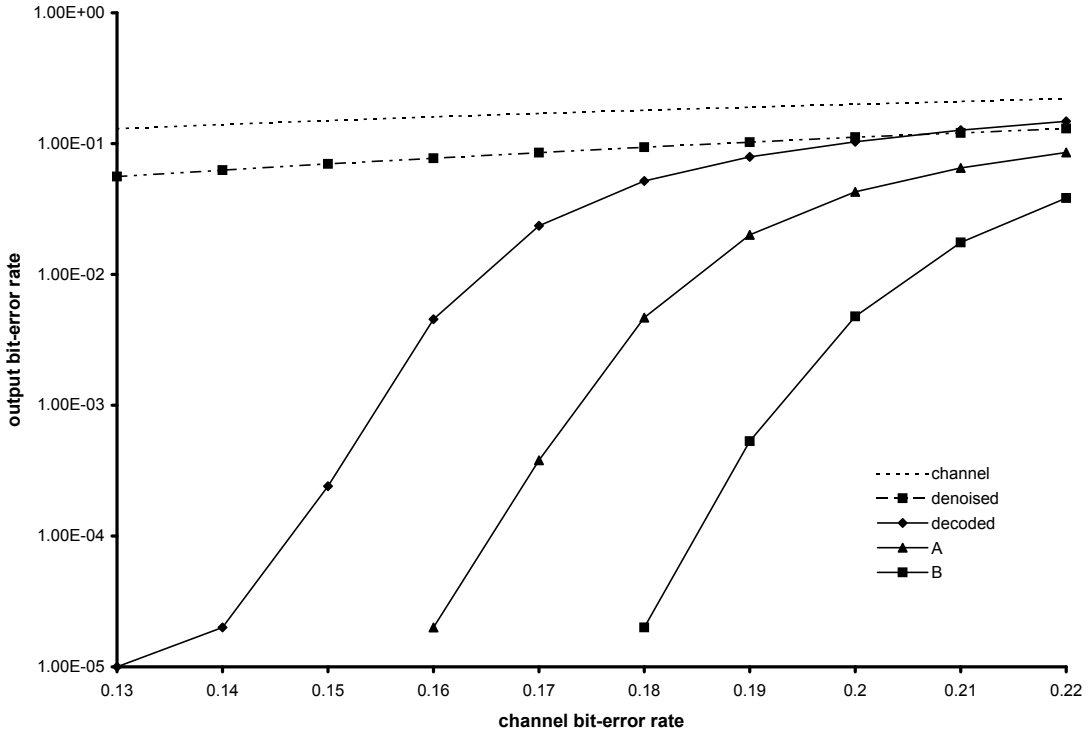

Figure 8: Bit-error rate of Approach A and Approach B applied to Data Set 1 (Figure 3).

	Output bit error rate			
Channel	Denoised	Decoded	A	B
0.13000	0.01310	0.00000	0.00000	0.00000
0.14000	0.01458	0.00002	0.00000	0.00000
0.15000	0.01630	0.00030	0.00000	0.00000
0.16000	0.01818	0.00464	0.00000	0.00000
0.17000	0.02003	0.02375	0.00001	0.00000
0.18000	0.02221	0.05221	0.00002	0.00000
0.19000	0.02453	0.07974	0.00134	0.00000
0.20000	0.02704	0.10464	0.00564	0.00003
0.21000	0.02965	0.12742	0.01441	0.00027
0.22000	0.03248	0.14902	0.02655	0.00116
0.23000	0.03558	0.16954	0.03942	0.00326
0.24000	0.03879	0.18908	0.05125	0.00654
0.25000	0.04215	0.20752	0.06170	0.01098
0.26000	0.04555	0.22504	0.07061	0.01614

Table 5: Results for Approaches A and B with a $(16000, 4000)$ RA code and Data Set 2 (Figure 4).

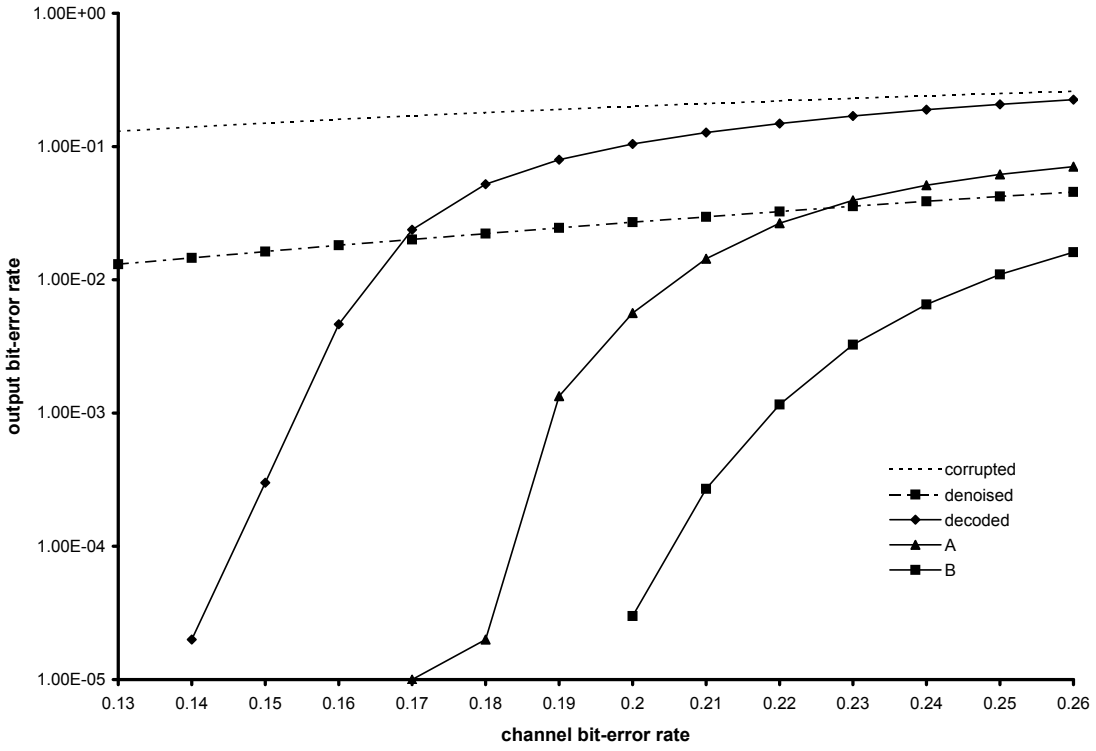

Figure 9: Bit-error rate of Approach A and Approach B applied to Data Set 2 (Figure 4).

to further refine reliability information for use in yet another error correction decoding stage. Additional iterations between the decoder and ssDUDE can be carried out, though we note that in practice just one iteration can yield significant gains over Approach B.

As above, let x^n and z^n, respectively denote the clean and noisy code symbols with indices $1, \ldots, k$ corresponding to information symbols and indices $k+1, \ldots, n$ corresponding to parity check symbols. Let $\mathcal{P}_X$ denote the k-fold product space of marginal probability distributions on the set of information symbols and let $\mathcal{V}$ denote the n-fold product space of M–dimensional non-negative vectors (recall that M denotes the input/output alphabet size). We may view each iteration of a soft-input-soft-output decoding operation as a mapping g from $\mathcal{P}_X \times \mathcal{V} \times \mathcal{S}$ to $\mathcal{P}_X \times \mathcal{S}$ where $\mathcal{S}$ denotes the internal state space of the decoding algorithm, e.g. the last round of edge messages in belief propagation. Letting $\{P^{(0)}_{X_i}(\cdot)\}$ denote the set of marginals on the information symbols generated by the sDUDE algorithm, with $P^{(0)}_{X_i}(x) = P_{X_i|Z^{2\kappa+1}}(x|z_{i-\kappa}^{i+\kappa})$, the output of the soft–output decoder after Approach B for received codeword symbols z^n can be expressed as

$$[\{Q^{(1)}_{X_i}(\cdot)\}, s_1] = g(\{P^{(0)}_{X_i}(\cdot)\}, \{\boldsymbol{\pi}_{z_i}\}, s_0),$$

where s_0 denotes an initial decoder state. The distribution $Q^{(1)}_{X_i}(\cdot)$ computed by the decoder for

the i–th symbol can be interpreted as a conditional probability distribution on the values that this symbol might take on given the entire (or significant portions of the) received, channel corrupted codeword. We next describe how ssDUDE operates on the distributions $\{Q^{(1)}_{X_i}(\cdot)\}$ and z^k to generate a refined set of information symbol marginals $\{P^{(1)}_{X_i}(\cdot)\}$ for input to a second decoding stage.

Common to all ssDUDE stages after the initial sDUDE stage is an estimate $\hat{P}_{X^{2\kappa+1}}(\cdot)$ of the probability distribution on values assumed by tuples of source (equivalently information) symbols. The estimate is based on the following property of the joint distribution of clean and noise corrupted random variables.

Lemma 1 *Suppose the random variables $X^m \stackrel{\triangle}{=} X_1, X_2, \ldots, X_m$ taking values in $\mathcal{A}$ are transmitted over a DMC with transition probability matrix $\mathbf{\Pi}$ to obtain $Z^m \stackrel{\triangle}{=} Z_1, Z_2, \ldots, Z_m$. Then for any $j = 1, \ldots, m$, and any $(\xi^m, \zeta^m) \in \mathcal{A}^m \times \mathcal{A}^m$*

$$\left[\begin{array}{c} P_{X^{j-1},Z_j^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},1,\zeta_{j+1},\ldots,\zeta_m) \\ P_{X^{j-1},Z_j^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},2,\zeta_{j+1},\ldots,\zeta_m) \\ P_{X^{j-1},Z_j^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},3,\zeta_{j+1},\ldots,\zeta_m) \\ \vdots \\ P_{X^{j-1},Z_j^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},M,\zeta_{j+1},\ldots,\zeta_m) \end{array}\right] =$$

$$\mathbf{\Pi}^T \left[\begin{array}{c} P_{X^{j},Z_{j+1}^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},1,\zeta_{j+1},\ldots,\zeta_m) \\ P_{X^{j},Z_{j+1}^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},2,\zeta_{j+1},\ldots,\zeta_m) \\ P_{X^{j},Z_{j+1}^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},3,\zeta_{j+1},\ldots,\zeta_m) \\ \vdots \\ P_{X^{j},Z_{j+1}^m}(\xi_1,\xi_2,\ldots,\xi_{j-1},M,\zeta_{j+1},\ldots,\zeta_m) \end{array}\right]. \quad (13)$$

The proof of Lemma 1 is completely analogous to the derivation of (8) in [26] and is omitted.

Assuming an invertible $\mathbf{\Pi}$, multiplying both sides of (13) by $\mathbf{\Pi}^{-T}$ gives a way to obtain the vector of probabilities on the right side, corresponding to one more X_j and one less Z_j, in terms of the vector of probabilities on the left side. Several iterations of this step are the basis for computing the estimate $\hat{P}_{X^{2\kappa+1}}(\cdot)$, the details of which follow.

Let

$$\hat{P}_{Z^{2\kappa+1}}(\zeta^{2\kappa+1}) = \frac{\mathbf{m}(z^k, \zeta^\kappa, \zeta_{\kappa+2}^{2\kappa+1})[\zeta_{\kappa+1}] + c}{N} \quad (14)$$

where $N = \sum_{\tilde{\zeta}^{2\kappa+1}} (\mathbf{m}(z^k, \tilde{\zeta}^\kappa, \tilde{\zeta}_{\kappa+2}^{2\kappa+1})[\tilde{\zeta}_{\kappa+1}] + c)$ is a normalization constant, $\mathbf{m}(\cdot)$ is given by (1), and c is a smoothing constant, set to $c = 1$ in the simulations. Thus, $\hat{P}_{Z^{2\kappa+1}}(\cdot)$ should be interpreted as a probability distribution on $(2\kappa+1)$-tuples of consecutive noisy symbols. Up to the constant c, it corresponds to the empirical distribution of such tuples in the noisy information symbols z^k. The estimate $\hat{P}_{X^{2\kappa+1}}(\cdot)$ of the empirical distribution of $(2\kappa+1)$-tuples of consecutive clean information

symbols is derived from $\hat{P}_{Z^{2\kappa+1}}(\cdot)$ in $(2\kappa+1)$-steps, each comprised of a collection of DUDE-like computations involving $\mathbf{\Pi}^{-1}$, as justified by Lemma 1. The first step operates on $\hat{P}_{Z^{2\kappa+1}}(\cdot)$ to compute an estimate $\hat{P}_{X_1,Z_2^{2\kappa+1}}(\cdot)$ of the empirical distribution of $(2\kappa+1)$-tuples of consecutive symbols in which the first symbol in each tuple is taken from the clean signal and the remaining symbols are taken from the noisy signal. In general, the j-th step outputs an estimate $\hat{P}_{X^j,Z_{j+1}^{2\kappa+1}}(\cdot)$ of the empirical distribution of $(2\kappa+1)$-tuples of consecutive symbols in which the first j symbols in each tuple are taken from the clean signal and the remaining symbols are taken from the noisy signal. The j-th step computes its estimate from the output of the $j-1$-th step $\hat{P}_{X^{j-1},Z_j^{2\kappa+1}}(\cdot)$ as follows

$$\begin{bmatrix} \hat{P}_{X^j,Z_{j+1}^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},1,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \\ \hat{P}_{X^j,Z_{j+1}^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},2,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \\ \hat{P}_{X^j,Z_{j+1}^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},3,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \\ \vdots \\ \hat{P}_{X^j,Z_{j+1}^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},M,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \end{bmatrix} =$$

$$\varphi\left(\mathbf{\Pi}^{-T}\begin{bmatrix} \hat{P}_{X^{j-1},Z_j^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},1,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \\ \hat{P}_{X^{j-1},Z_j^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},2,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \\ \hat{P}_{X^{j-1},Z_j^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},3,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \\ \vdots \\ \hat{P}_{X^{j-1},Z_j^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},M,\zeta_{j+1},\ldots,\zeta_{2\kappa+1}) \end{bmatrix}\right). \quad (15)$$

The smoothing operation $\varphi(\cdot)$ in (15) ensures that all of the components of $\hat{P}_{X^j,Z_{j+1}^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},\cdot,\zeta_{j+1},\ldots,\zeta_{2\kappa+1})$ are non-negative and have the same sum as the components of $\mathbf{\Pi}^{-T}[\hat{P}_{X^{j-1},Z_j^{2\kappa+1}}(\xi_1,\xi_2,\ldots,\xi_{j-1},\cdot,\zeta_{j+1},\ldots,\zeta_{2\kappa+1})]$.

For the simulation results, on binary sources/channels ($M=2$), the following smoothing function was used (cf. (7))

$$\varphi([x,y]^T) = \begin{cases} [x,y]^T & \text{if } x \geq c/N \text{ and } y \geq c/N \\ [c/N,(x+y-c/N)]^T & \text{if } x < c/N \\ [(x+y-c/N),c/N]^T & \text{if } y < c/N \end{cases} \quad (16)$$

where c and N are, respectively, the offset and normalization constants appearing in (14). The fact that $\mathbf{\Pi}$ (and hence $\mathbf{\Pi}^{-1}$) is a stochastic matrix imply that $\varphi(\cdot)$ will always operate on vectors $[x,y]^T$ satisfying $x+y \geq 2c/N$. The simulation results for Approach C in Section V-5 were generated using the smoothing function (16), and $c=1$ in both the smoothing function and in (14).

We next describe how the ssDUDE stage uses $\hat{P}_{X^{2\kappa+1}}(\cdot)$ and the output of the decoding stage

$\{Q_{X_i}^{(1)}(\cdot)\}$ to generate the next stage *a priori* reliability information $P_{X_i}^{(1)}(x)$. Define $\tilde{Q}_{X_i}^{(1)}(\cdot)$ by

$$\tilde{Q}_{X_i}^{(1)}(x) = \frac{Q_{X_i}^{(1)}(x)\boldsymbol{\pi}_{z_i}[x]}{P_{X_i}^{(0)}(x)}. \tag{17}$$

This step removes the "intrinsic" information from the decoder generated posteriors. Then, for each information symbol index i, and each value x, let

$$\tilde{P}_{X_i}^{(1)}(x) = \sum_{\xi^{2\kappa+1}:\xi_{\kappa+1}=x} \hat{P}_{X^{2\kappa+1}}(\xi^{2\kappa+1})\boldsymbol{\pi}_{z_i}[x] \prod_{\substack{-\kappa\le j\le\kappa \\ j\neq 0}} \tilde{Q}_{X_{i+j}}^{(1)}(\xi_{\kappa+j+1}), \tag{18}$$

and set

$$P_{X_i}^{(1)}(x) = \frac{\tilde{P}_{X_i}^{(1)}(x)}{\sum_x \tilde{P}_{X_i}^{(1)}(x)}. \tag{19}$$

As explained below in the context of LDPC codes, these operations can be interpreted as being part of a belief propagation-like procedure on an augmented factor graph representing both the channel code constraints and the estimated clean source distribution.

The distributions $\{P_{X_i}^{(1)}(\cdot)\}$ are then fed into a second decoding stage whose operation can be expressed as

$$[\{Q_{X_i}^{(2)}(\cdot)\}, s_2] = g(\{P_{X_i}^{(1)}(\cdot)\}, \{\boldsymbol{\pi}_{z_i}\}, s_1).$$

Following this, additional iterations of the above stages can be carried out, and after the t-th decoding stage the distributions $\{Q_{X_i}^{(t)}(\cdot)\}$ can be the basis for MAP decisions to yield an Approach C decoded signal. Note that $\hat{P}_{X^{2\kappa+1}}(\cdot)$, the estimate of counts of tuples of clean source symbols, is constant for all ssDUDE iterations.

The simulation results presented in Section V-5 are for an Approach C implementation involving systematic LDPC codes (RA codes, specifically) and a soft-input-soft-output version of belief propagation decoding. For concreteness, we specify the function

$$[\{Q_{X_i}(\cdot)\}, s_{out}] = g(\{P_{X_i}(\cdot)\}, \{\boldsymbol{\pi}_i\}, s_{in}) \tag{20}$$

corresponding to this decoder, with reference to the description of belief propagation in Section IV (equations (9)–(12)). The generic input parameters $\{P_{X_i}(\cdot)\}$, $\{\boldsymbol{\pi}_i\}$, and s_{in} in (20) respectively denote distributions on information symbols, channel matrix columns corresponding to noisy parity check symbols, and an initial decoder state, while the generic output parameters $\{Q_{X_i}\}$ and s_{out} denote refined distributions on information symbols and a final decoder state. The function $g(\cdot)$ then implements the belief propagation computations of (9) through (12) in Section IV, and incorporates the above generic parameters as follows. The $\mathcal{L}_i$ are set as in (9) and (10), with the generic $g(\cdot)$ input parameter $P_{X_i}(\cdot)$ replacing the corresponding $\hat{P}_{X_i|Z^{2\kappa+1}}(\cdot|z_{i-\kappa}^{i+\kappa})$ in (9), and the generic $\boldsymbol{\pi}_i$

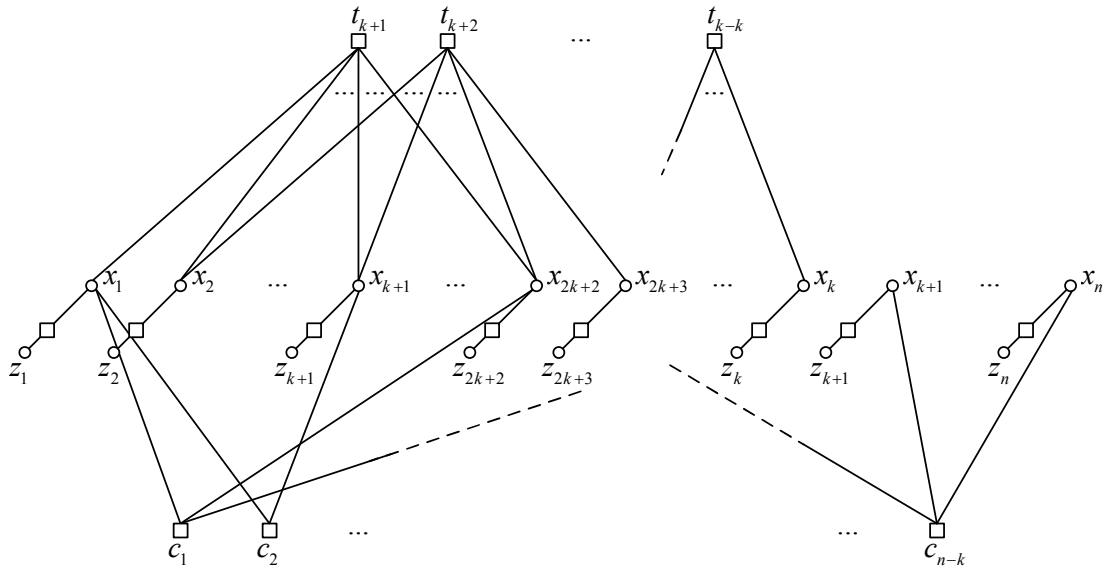

Figure 10: Factor graph for ssDUDE enhanced decoding.

parameter replacing $\boldsymbol{\pi}_{z_i}$ in (10). The generic input parameter s_{in}, corresponding to the decoder state, is assumed to specify $\mu_{j\to i}^{(-1)}[x]$ for all i, j, and x in (11) for the first iteration. The state portion of the output of $g(\cdot)$, assuming ℓ decoder iterations, is set to $\{\mu_{j\to i}^{(\ell-1)}\}$, while the probabilistic portion corresponding to index i is set to

$$Q_{X_i}(x) = \frac{\mathcal{L}_i[x]\prod_{j'\in\mathcal{N}_i}\mu_{j'\to i}^{(\ell-1)}[x]}{\sum_{x\in\mathcal{A}}\mathcal{L}_i[x]\prod_{j'\in\mathcal{N}_i}\mu_{j'\to i}^{(\ell-1)}[x]}.$$

Note, that the decoding stage of each Approach C denoising/decoding iteration in general consists of numerous (sub)iterations of belief propagation on the LDPC code graph.

The ssDUDE enhanced decoder for LDPC codes, as just described, can also be interpreted as a slightly modified form of belief propagation (sum-product algorithm) applied to the factor graph (see [39]) shown in Figure 10, representing both the channel code constraints and the estimated clean symbol probabilities. The factors in Figure 10 are denoted by squares. The pairwise factors involving the clean and noisy codeword variables $\{x_j\}$ and $\{z_j\}$ correspond to the channel transition probabilities. The factors $\{c_j\}$ correspond to the parity check constraints, while the factors $\{t_j\}$ correspond to $\hat{P}_{X^{2\kappa+1}}(\cdot)$ applied to the 2κ+1-tuples of information symbols $x_{j-\kappa}^{j+\kappa}$, $j = \kappa+1, \ldots, k-\kappa$ connected to them in the figure.[2] Note that these latter factors do not include variables $x_{k+1}, \ldots, x_n$

[2]For simplicity, we omit factors corresponding to the boundary indices $j < \kappa+1, j > k-\kappa$. A good choice for such factors would be marginal distributions on fewer random variables derived from $\hat{P}_{X^{2\kappa+1}}(\cdot)$.

which correspond to parity symbols. The ssDUDE and decoding stages of each Approach C iteration described above correspond, respectively, to the updating of messages passing between $\{t_j\}$ and $\{x_j\}$ and to the updating of messages passing between $\{x_j\}$ and $\{c_j\}$ in a belief propagation-like algorithm. The overall goal is to approximate the conditional distribution of each x_j given all the observed $\{y_j\}$. As we explain below, the main difference between Approach C and standard belief propagation on this factor graph has to do with the messages passing back and forth between the factors $\{t_j\}$ and the variable nodes $\{x_j\}$. The factor to variable node edge messages are initialized as usual, with the exception of the edges $(t_j \to x_j)$ which are initialized to $\hat{P}_{X_j|Z^{2\kappa+1}}(x|z_{j-\kappa}^{j+\kappa})/\mathbf{\Pi}(x, z_j)$, $x \in \mathcal{A}$, where $\hat{P}_{X_j|Z^{2\kappa+1}}(x|z_{j-\kappa}^{j+\kappa})$ is the output of sDUDE. We divide by $\mathbf{\Pi}(x, z_j)$ for consistency with the factor graph of Figure 10, which multiplies this term back in through the pairwise channel factors connecting x_j and z_j. In future iterations, messages passing on the edges $(x_j \to t_j)$ and $(t_j \to x_j)$ are updated as usual, with the updates of the former corresponding roughly[3] to equations (18) and (19). In a departure from standard belief propagation, however, messages passing on $(t_j \to x_i)$ for $i \neq j$ are always fixed to the all-ones vector $[1, 1, \ldots, 1]$ and are never updated, and the messages passing on $(x_i \to t_j)$ for $i \neq j$ are set equal to the messages on $(x_i \to t_i)$, the latter updated in the usual way, as already noted. These modifications stem from the fact that we do not want products of factors t_j with common variables to appear in the propagating messages. An additional difference relative to standard belief propagation is in the schedule of message updates. In particular, decoding proceeds with multiple updates of the messages passing between $\{x_j\}$ and $\{c_j\}$, while the messages between $\{t_j\}$ and $\{x_j\}$ are held fixed. This corresponds to the multiple channel decoder iterations per denoising iteration mentioned above. After a certain number of such iterations, the messages passing back and forth between $\{x_j\}$ and $\{t_j\}$ are updated in the usual way, with the aforementioned exceptions for $(t_j \to x_i)$ and $(x_i \to t_j)$, $i \neq j$.

V-2 Approach D

Approach C has the drawback that the error correction decoding stage is not being leveraged to refine $\hat{P}_{X^{2\kappa+1}}(\cdot)$, the estimate of the counts of clean source symbols. Approach D (along with Approaches E and F of the next subsections) is an attempt to patch this hole via a different iterative process.

Approach D also consists of a sequence of iterations of a denoising stage followed by a decoding stage. Like Approach C of the previous section, the first iteration consists of a run of the sDUDE algorithm followed by the corresponding enhanced error correction decoding stage, as detailed in Section V-1. Subsequent iterations differ from the iterations of Approach C only in how the

[3]The actual update in the modified belief propagation formulation involves a similar sum-of-products, with the $\boldsymbol{\pi}_{z_i}[x]$ factor omitted and the $\tilde{Q}^{(1)}_{X_{i+j}}(\xi_{\kappa+j+1})$ factors derived from the $(x_j \to t_j)$ messages.

denoising stage processes the aggregate reliability information $\{Q_{X_i}^{(t)}(\cdot)\}$ to generate the new a–priori reliability information $\{P_{X_i}^{(t)}(\cdot)\}$. We now describe the corresponding Approach D processing.

1. Obtain a hard decision decoded sequence of source symbols by decoding each symbol i to the value x maximizing $Q_{X_i}^{(t)}(x)$. Denote this sequence of symbols by $\hat{\mathbf{z}}^{(t)} = \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_k$.

2. Determine the fraction of occurrences of each context of symbols (context in the DUDE sense) in the decoded sequence $\hat{\mathbf{z}}^{(t)}$ for which the "center" symbol takes on each possible value x. This is equivalent to the normalized $\mathbf{m}(\hat{\mathbf{z}}^{(t)}, \boldsymbol{c})$ vector from the DUDE description.

3. For each i, let $P_{\hat{Z}_i}(z)$ denote $\mathbf{m}(\hat{\mathbf{z}}^{(t)}, \boldsymbol{\eta}_i)[z] / \sum_z \mathbf{m}(\hat{\mathbf{z}}^{(t)}, \boldsymbol{\eta}_i)[z]$ where $\boldsymbol{\eta}_i$ represents the value of the context of the i-th symbol, and let $\hat{Z}_i$ denote the corresponding random variable.

4. For each i, derive a prior distribution $Pr(X_i = x)$ on the clean symbol X_i that is consistent with the following postulated approximations about the joint distribution of X_i and $\hat{Z}_i$:

   (a) $Pr(\hat{Z}_i = z) = P_{\hat{Z}_i}(z)$ derived above, for all z.

   (b) $Pr(X_i = x | \hat{Z}_i = \hat{z}_i) = Q_{X_i}^{(t)}(x)$, the output of the decoding stage of the previous iteration.

   (c) $Pr(\hat{Z}_i = z | X_i = x)$ for all z and x correspond to an M–ary symmetric channel.

   Under these approximations, a system of M equations in the M unknowns $Pr(X_i = x)$, $x \neq \hat{z}_i$ can be derived. One equation is the trivial $\sum_x Pr(X_i = x) = 1$. For $x \neq \hat{z}_i$ we have

$$\begin{aligned} Pr(X_i = x) &= Pr(X_i = x, \hat{Z}_i = \hat{z}_i) + \sum_{z \neq \hat{z}_i} Pr(X_i = x, \hat{Z}_i = z) \\ &= Pr(X_i = x, \hat{Z}_i = \hat{z}_i) + \sum_{z \neq \hat{z}_i} Pr(\hat{Z}_i = z | X_i = x) Pr(X_i = x) \\ &= Pr(X_i = x, \hat{Z}_i = \hat{z}_i) + \\ &\qquad [(M-2) Pr(\hat{Z}_i = \hat{z}_i | X_i = x) + Pr(\hat{Z}_i = \hat{z}_i | X_i = \hat{z}_i)] Pr(X_i = x) \qquad (21) \\ &= (M-1) Pr(X_i = x, \hat{Z}_i = \hat{z}_i) + Pr(X_i = \hat{z}_i | \hat{Z}_i = \hat{z}_i) Pr(X_i = x) \frac{Pr(\hat{Z}_i = \hat{z}_i)}{Pr(X_i = \hat{z}_i)} \qquad (22) \end{aligned}$$

   where (21) follows from the M–ary symmetric channel approximation. Letting $Pr(\hat{Z}_i = \hat{z}_i) = P_{\hat{Z}_i}(\hat{z}_i)$ and $Pr(X_i = x | \hat{Z}_i = \hat{z}_i) = Q_{X_i}^{(t)}(x)$, according to the other approximations, we have the following $M - 1$ equations,

$$Pr(X_i = x) = (M-1) Q_{X_i}^{(t)}(x) P_{\hat{Z}_i}(\hat{z}_i) + Q_{X_i}^{(t)}(\hat{z}_i) P_{\hat{Z}_i}(\hat{z}_i) \frac{Pr(X_i = x)}{Pr(X_i = \hat{z}_i)}, \qquad (23)$$

   with one such equation for each of $x \neq \hat{z}_i$.

The equations can be solved for the $Pr(X_i = x)$ as follows. Summing the equations over $x \neq \hat{z}_i$ (along with the first trivial equation) leads to a quadratic equation in $Pr(X_i = \hat{z}_i)$ as the only unknown. The corresponding solution for $Pr(X_i = \hat{z}_i)$ can then be inserted into each of the above $M-1$ equations which become linear in the corresponding unknowns $Pr(X_i = x)$ and hence are easily solved. Then, for each x, set $P_{X_i}^{(t)}(x)$ equal to $Pr(X_i = x)$, so obtained, as the output of the t-th iteration Approach D denoising stage.

We remark that in the binary case $M = 2$, there is essentially only one equation to be solved in step 4, namely the quadratic one. In general, there will be two real solutions to the quadratic equation, both in the binary and non–binary cases. Clearly that solution which is strictly between 0 and 1 should be chosen. If both solutions satisfy this condition then the natural choice is the one maximizing the induced

$$Pr(\hat{Z}_i = \hat{z}_i | X_i = \hat{z}_i) = \frac{Q_{X_i}^{(t)}(x) P_{\hat{Z}_i}(\hat{z}_i)}{Pr(X_i = \hat{z}_i)},$$

which translates into choosing the smaller of the two solutions for $Pr(X_i = \hat{z}_i)$. This is the selection rule applied in obtaining the simulation results in Section V-5. If the solution to the quadratic equation is imaginary a good heuristic is to simply set $P_{X_i}^{(t)}(\cdot)$ equal to $P_{\hat{Z}_i}(\cdot)$.

V-3 Approach E

Approach E also consists of a sequence of iterations of a denoising stage followed by a decoding stage. Like Approach C of the previous section, the first iteration consists of a run of the sDUDE algorithm followed by the corresponding enhanced error correction decoding stage, as detailed in Section V-1. Subsequent iterations differ from the iterations of Approach C only in how the denoising stage processes the aggregate reliability information $\{Q_{X_i}^{(t)}(\cdot)\}$ to generate the new a–priori reliability information $\{P_{X_i}^{(t)}(\cdot)\}$. We now describe the corresponding Approach E processing.

1. Obtain a hard decision decoded sequence of source symbols by decoding each symbol i to the value x maximizing $Q_{X_i}^{(t)}(x)$. Denote this sequence of symbols by $\hat{\mathbf{z}}^{(t)} = \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_k$.

2. Determine the fraction of occurrences of each context of symbols (context in the DUDE sense) in the decoded sequence $\hat{\mathbf{z}}^{(t)}$ for which the "center" symbol takes on each possible value x. This is equivalent to the normalized $\mathbf{m}(\hat{\mathbf{z}}^{(t)}, \boldsymbol{c})$ vector from the DUDE description.

3. For each i, set

$$P_{X_i}^{(t)}(x) = \frac{\boldsymbol{\pi}_{z_i}[x] \mathbf{m}(\hat{\mathbf{z}}^{(t)}, \boldsymbol{\eta}_i)[x]}{\sum_z \boldsymbol{\pi}_{z_i}[z] \mathbf{m}(\hat{\mathbf{z}}^{(t)}, \boldsymbol{\eta}_i)[z]} \tag{24}$$

where $\boldsymbol{\eta}_i$ represents the value of the context of the i-th symbol in the hard decision output $\hat{\mathbf{z}}^{(t)}$.

Note that the alternative denoising stage requires only hard decisions from the decoder and otherwise ignores the soft output information. Approach E can also be interpreted as a natural approximation of Approach F described next.

V-4 Approach F

Approach F is also an iterative denoising/decoding procedure nearly identical to Approach C except that it attempts to improve the estimate of the distribution of tuples of clean symbols $\hat{P}_{X^{2\kappa+1}}(\cdot)$ in each iteration, rather than leave it unchanged. Let $\hat{P}^{(t)}_{X^{2\kappa+1}}(\cdot)$ denote the alternative estimate for the t–th iteration, as used in (18). It is computed as follows.

1. Like in Approach E, obtain a hard decision decoded sequence of source symbols by decoding each symbol i to the value x maximizing $Q^{(t)}_{X_i}(x)$. Denote this sequence of symbols by $\hat{\mathbf{z}}^{(t)} = \hat{z}_1, \hat{z}_2, \ldots, \hat{z}_k$.

2. Determine $\hat{P}^{(t)}_{X^{2\kappa+1}}(\cdot)$ as

$$\hat{P}^{(t)}_{X^{2\kappa+1}}(\xi^{2\kappa+1}) = \frac{\mathbf{m}(\hat{\mathbf{z}}^{(t)}, \xi^{\kappa}, \xi_{\kappa+2}^{2\kappa+1})[\xi_{\kappa+1}] + c}{N}$$

   where $N = \sum_{\tilde{\xi}^{2\kappa+1}} (\mathbf{m}(\hat{\mathbf{z}}^{(t)}, \tilde{\xi}^{\kappa}, \tilde{\xi}_{\kappa+2}^{2\kappa+1})[\tilde{\xi}_{\kappa+1}] + c)$ is a normalization constant, $\mathbf{m}(\cdot)$ is given by (1), and c is a smoothing constant set to $c = 1$ in the simulations.

Approach F is then identical to Approach C with $\hat{P}^{(t)}_{X^{2\kappa+1}}(\cdot)$, as defined above, replacing $\hat{P}_{X^{2\kappa+1}}(\cdot)$, derived according to (15), in the t-th iteration update (18). In the factor graph/belief propagation interpretation of Approach C (see Figure 10), the change introduced by Approach F is to modify the factors $\{t_j\}$ to $\hat{P}^{(t)}_{X^{2\kappa+1}}(\cdot)$ before the t-th update of the messages $(t_j \to x_j)$.

We next show how Approach E can be viewed as an approximation of Approach F. Approach F updates according to

$$\tilde{P}^{(t)}_{X_i}(x) = \sum_{\xi^{2\kappa+1}:\xi_{\kappa+1}=x} \hat{P}^{(t)}_{X^{2\kappa+1}}(\xi^{2\kappa+1}) \boldsymbol{\pi}_{z_i}[x] \prod_{\substack{-\kappa \le j \le \kappa \\ j \neq 0}} \tilde{Q}^{(t)}_{X_{i+j}}(\xi_{\kappa+j+1}), \tag{25}$$

where $\tilde{Q}^{(t)}_{X_j}(x)$ is the "intrinsic" information removed soft output from the previous decoder stage, as computed according to (17). Approach E can then be interpreted as replacing $\tilde{Q}^{(t)}_{X_j}(x)$ in the expression (25) with $Q^{(t)}_{X_j}(x)$, the unmodified decoder soft output, and restricting the summation in (18) to one term, namely,

$$\begin{aligned} \xi^{2\kappa+1} &= \arg \max_{\tilde{\xi}^{2\kappa+1}:\tilde{\xi}_{\kappa+1}=x} \prod_{\substack{-\kappa \le j \le \kappa \\ j \neq 0}} Q^{(t)}_{X_{i+j}}(\tilde{\xi}_{\kappa+j+1}) \\ &= \hat{z}_{i-\kappa}^{i-1}, x, \hat{z}_{i+1}^{i+\kappa}. \end{aligned}$$

	Output bit error rate						
Channel	Denoised	Decoded	A	B	C	E	F
0.18000	0.09375	0.05182	0.00468	0.00002	0.00000	0.00000	0.00000
0.19000	0.10251	0.07925	0.02006	0.00053	0.00000	0.00000	0.00000
0.20000	0.11185	0.10335	0.04278	0.00478	0.00000	0.00000	0.00000
0.21000	0.12075	0.12655	0.06509	0.01756	0.00001	0.00000	0.00000
0.22000	0.13045	0.14800	0.08563	0.03831	0.00005	0.00002	0.00001
0.23000	0.14119	0.16917	0.10573	0.05993	0.00060	0.00050	0.00075
0.24000	0.15198	0.18857	0.12405	0.08078	0.00436	0.00510	0.00880
0.25000	0.16284	0.20749	0.14177	0.10093	0.01597	0.02000	0.03099
0.26000	0.17418	0.22449	0.15941	0.12032	0.03712	0.04164	0.05952
0.27000	0.18580	0.24094	0.17529	0.13967	0.06302	0.06461	0.08862

Table 6: Results for Approaches A to F with a $(16000, 4000)$ RA code and Data Set 1 (Figure 3).

Up to normalization, the summation in update (25) over this one term becomes

$$\tilde{P}_{X_i}^{(1)}(x) = \hat{P}_{X^{2\kappa+1}}(\hat{z}_{i-\kappa}^{i-1}, x, \hat{z}_{i+1}^{i+\kappa})\boldsymbol{\pi}_{z_i}[x]$$

which corresponds exactly to the update of Approach E.

A range of algorithms between Approaches E and F can be obtained through alternative restrictions of the summation in (25), to, for example, allowing for only one error in the context (as opposed to zero above), or two, etc, relative to the hard decision sequence.

V-5 Results

In this section, we present the results of experiments carried out on the data sets of Section III-2 with Approaches C, E, and F. Results for Approaches A and B are also included, for comparison purposes. Approach D was found to uniformly and significantly underperform Approaches C, E, and F in our experiments and we omit it from the comparisons below to simplify the presentation. The channel coding setup (RA codes, etc.) is identical to that described at the end of Section IV. For Data Sets 1 and 2, the noisy output from the channel is decoded with each of the algorithms discussed in Section III-1 and Sections IV to V-4. The hard decision channel decoding block for Approach A is a standard belief propagation decoder. The context sizes for Data Sets 1 and 2 are identical to those chosen for the experiments detailed at the end of Section IV.

The BER results for Data Sets 1 and 2 are presented, respectively, in Tables 6 and 7 and plotted in Figures 11 and 12. From the figures it can be inferred that, for an output bit error rate of 10^{-4}, iterative denoising and decoding schemes yield coding gains over Approach B of 1.4-1.5dB for Data Set 1 and 1.4-1.8dB for Data Set 2.

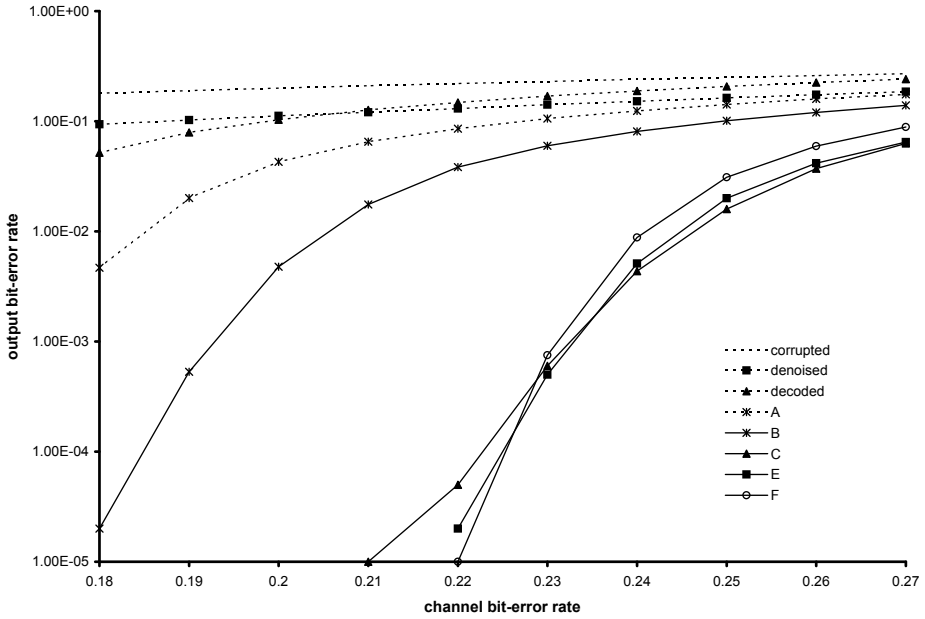

Figure 11: Bit-error rate of Approaches A–F applied to Data Set 1 (Figure 3).

From Figures 11 and 12 we observe that the iterative decoding and denoising schemes C, E, and F substantially outperform the non-iterative schemes A and B. Among the iterative schemes, Approach F seems to give the best performance as the channel noise decreases, though no single scheme completely dominates the others for all noise levels.

We perform a similar experiment for Data Set 3 (HTML data), with the same channel coding set-up. We restrict the experiment to Approach E, which competed well with the best iterative algorithms for the image data, and include the results for Approaches A and B for reference. As mentioned in Section III-2, we use a "character-aware" scheme for denoising, where the context size K is varied from 16 to 23 based on the noise levels.

The experiments for Data Set 3 are presented in Table 8 and plotted in Figure 13. From the figure it can be inferred that, for a bit error rate of 10^{-4}, Approach A yields a coding gain of about 0.3dB over a plain decoder, Approach B, in turn, yields a coding gain of approximately 0.5 dB over Approach A, and Approach E yields a coding gain of about 0.9dB over Approach B. Thus, we observe continued significant performance gains in decoding for yet a third type of data (with Data Sets 1 and 2 corresponding to two different types of image data), providing empirical evidence of the universality of our schemes. In the next section, we study the universality properties more directly by evaluating the proposed DUDE-enhanced decoding structures along with corresponding source-distribution-aware (and hence non-universal) enhanced decoding structures on synthetic

	Output bit error rate						
Channel	Denoised	Decoded	A	B	C	E	F
0.20000	0.02704	0.10464	0.00564	0.00003	0.00000	0.00000	0.00000
0.21000	0.02965	0.12742	0.01441	0.00027	0.00000	0.00000	0.00000
0.22000	0.03248	0.14902	0.02655	0.00116	0.00000	0.00000	0.00000
0.23000	0.03558	0.16954	0.03942	0.00326	0.00001	0.00000	0.00000
0.24000	0.03879	0.18908	0.05125	0.00654	0.00006	0.00002	0.00001
0.25000	0.04215	0.20752	0.06170	0.01098	0.00028	0.00017	0.00005
0.26000	0.04555	0.22504	0.07061	0.01614	0.00089	0.00093	0.00041
0.27000	0.04922	0.24149	0.07817	0.02155	0.00244	0.00274	0.00155
0.28000	0.05305	0.25744	0.08464	0.02739	0.00512	0.00615	0.00400
0.29000	0.05717	0.27237	0.08979	0.03367	0.00929	0.01120	0.00817
0.30000	0.06130	0.28658	0.09370	0.04027	0.01472	0.01765	0.01380

Table 7: Results for Approaches A to F with a $(16000, 4000)$ RA code and Data Set 2 (Figure 4).

data generated by Markov sources.

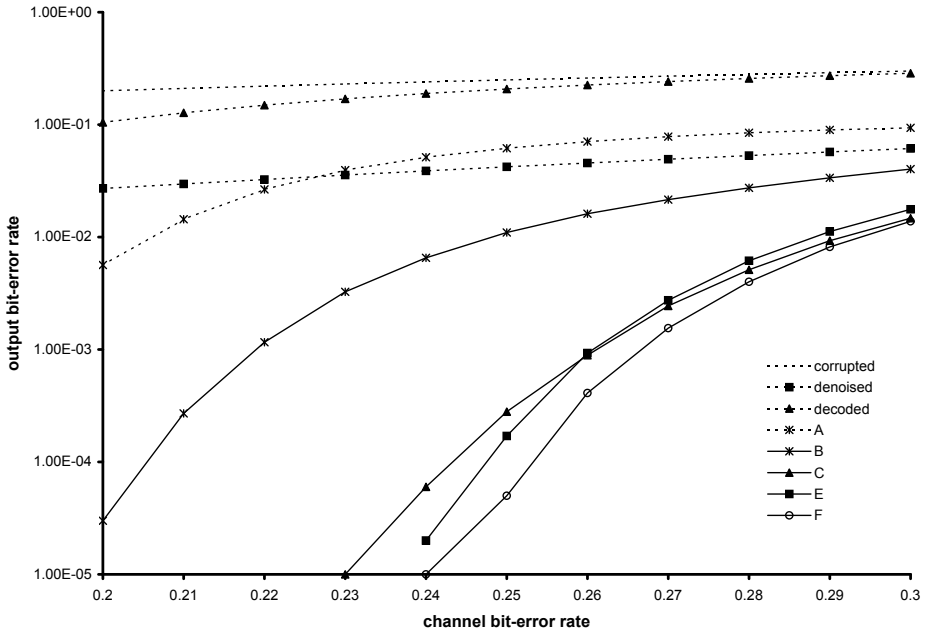

Figure 12: Bit-error rate of Approaches A–F applied to Data Set 2 (Figure 4).

	Output bit error rate				
Channel	Denoised	Decoded	A	B	E
0.12000	0.08368	0.00000	0.00000	0.00000	0.00000
0.13000	0.09239	0.00001	0.00000	0.00000	0.00000
0.14000	0.10092	0.00002	0.00001	0.00000	0.00000
0.15000	0.10950	0.00027	0.00003	0.00001	0.00000
0.16000	0.11787	0.00460	0.00039	0.00002	0.00000
0.17000	0.12626	0.02369	0.00481	0.00013	0.00001
0.18000	0.13447	0.05209	0.02135	0.00170	0.00001
0.19000	0.14262	0.07943	0.04570	0.01024	0.00004
0.20000	0.15076	0.10437	0.06948	0.02874	0.00038
0.21000	0.15889	0.12748	0.09117	0.05074	0.00430
0.22000	0.16703	0.14912	0.11113	0.07186	0.02255

Table 8: Results for Approaches A, B, and E with a $(16000, 4000)$ RA code and Data Set 3.

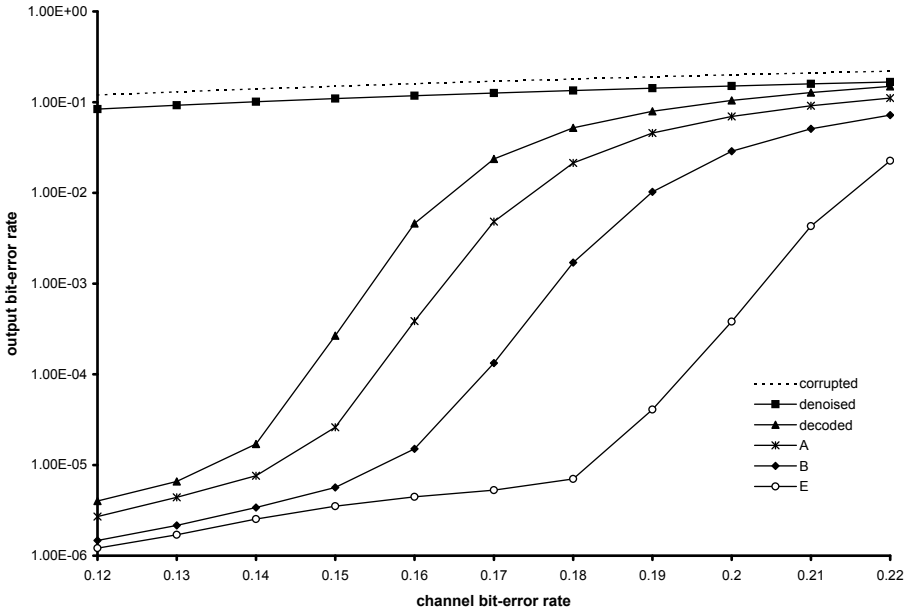

Figure 13: Bit-error rate of Approaches A, B, and E applied to Data Set 3

VI Markov sources

In this section, we evaluate the performance of the proposed schemes on channel-encoded binary Markov sources. The channel code and baseline channel decoder are again, respectively, the RA code and belief propagation decoder of Sections IV and V-5, while the Markov sources are first-order with symmetric transition probabilities. We compare the performance of the proposed schemes to analogous non-universal enhanced decoders, denoted as Hard BCJR+BP, Soft BCJR+BP, and Iterative BCJR+BP, that exploit knowledge of the source statistics. Hard BCJR+BP corresponds to Approach A with a BCJR denoiser using the actual Markov chain parameters replacing the DUDE block in Figure 2. Similarly, Soft BCJR+BP is Approach B but with the sDUDE block in Figure 7 replaced by a version of BCJR that emits the computed clean symbol posteriors. These posteriors then serve as inputs to the belief propagation decoder, as in Approach B. Iterative BCJR+BP carries out belief propagation on the augmented factor graph of Figure 14, similarly to what is done in [7] and in Approaches C and F above. The factor graph of Figure 14 is similar to that of Figure 10. In this case, the factors t_j correspond to the Markov transition probabilities,

$$t_j(x_j, x_{j+1}) = p_{X_{j+1}|X_j}(x_{j+1}|x_j) = \begin{cases} \pi & \text{if } x_{j+1} \neq x_j \\ 1-\pi & \text{if } x_{j+1} = x_j \end{cases}$$

Unlike in Approaches C and F, Iterative BCJR+BP involves standard belief propagation on the augmented factor graph. The schedule of message updates, however, is similar to C and F and entails multiple outer iterations of an inner subiterative process comprised, in turn, of numerous iterations back and forth between $\{t_j\}$ and $\{x_j\}$, sufficiently many, in fact, to implement the full sequence BCJR, followed by numerous iterations between $\{x_j\}$ and $\{c_j\}$, while holding the messages from $\{t_j\}$ fixed. After at least two outer iterations of this subiterative process, hard decisions are taken at the variable nodes. As in Approaches C and F, negligible improvement in performance is seen beyond two iterations.

The results are shown in Figure 15 for a Markov source with transition probability $\pi = .05$, and as mentioned, the rate 1/4 regular RA code also used in Sections IV and V-5. To simplify the plot, we have omitted the results for iterative universal Approaches C and E, which are similar to those of Approach F, with Approach E being slightly inferior to C and F for noisier channels. For $\pi = .05$, the universal and (corresponding) non-universal error rates are found to track each other closely in all cases. Though we omit the plots, we find, as π decreases from .05, that the universal and non-universal hard decisions schemes remain fairly close in performance, while there is some divergence in performance for the corresponding soft decision schemes. This is most likely due to slower convergence of the sDUDE estimated clean symbol posteriors to the actual posteriors used by the BCJR based schemes. Evidence of this slower convergence can already be seen in Table I in [26] for the case of BSC crossover $\delta = .20$, Markov transition $\pi = .01$, and a sequence length of 10^6,

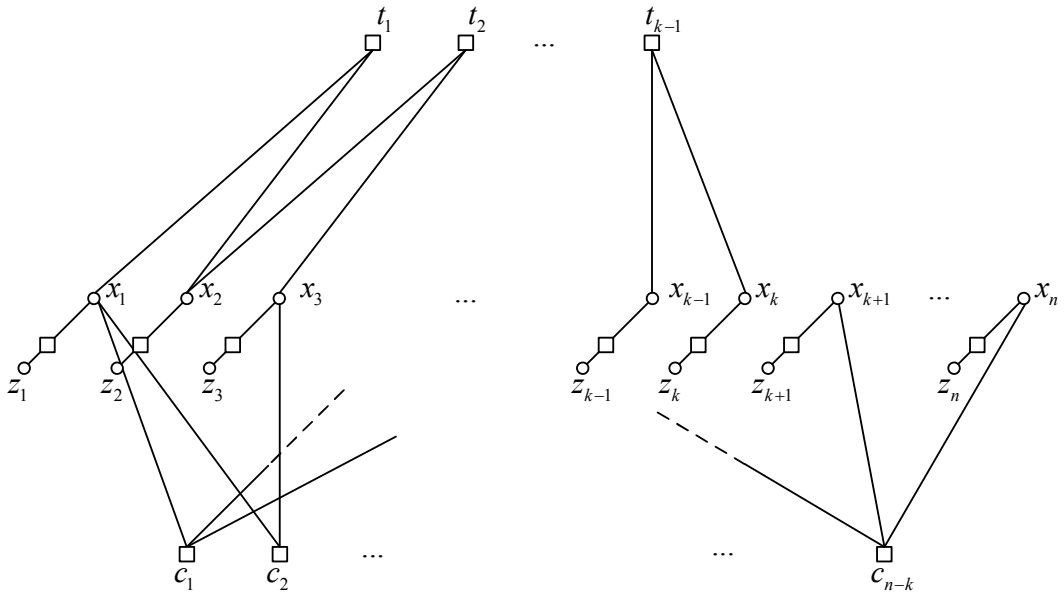

Figure 14: Factor graph for Iterative BCJR+BP enhanced decoding.

where Bayes optimal denoising (e.g. BCJR) significantly outperforms the DUDE. These empirical observations are consistent with the theoretical analysis in [26], which shows a convergence penalty that increases with δ.

As additional reference points, we can derive the theoretical maximum values of the BSC crossover probability δ that allow, respectively, reliable *uncompressed* and *compressed* channel encoding of the Markov source at the coding rate of 1/4 source symbols per channel symbol used in the experiments of Figure 15, with possibly source dependent encoding and decoding. In uncompressed channel encoding, which is the paradigm assumed throughout the paper, the source symbols are transmitted in the clear as part of the channel codeword, while compressed channel encoding is the classical Shannon paradigm of source-channel coding for which separate data compression and channel encoding is asymptotically optimal. Let $\mathcal{H}(\mathbf{X}|\mathbf{Z})$ denote the conditional entropy rate of the Markov source $\mathbf{X}$ given its noisy version $\mathbf{Z}$, which is a function of δ and π. Let $h(\delta)$ denote the binary entropy function evaluated at probability δ, so that $1 - h(\delta)$ is the capacity of a BSC with crossover δ, and let R denote the rate of source symbols per channel symbols of the uncompressed encoding. According to Theorem 1 of [40], uncompressed channel encoding at rate R with

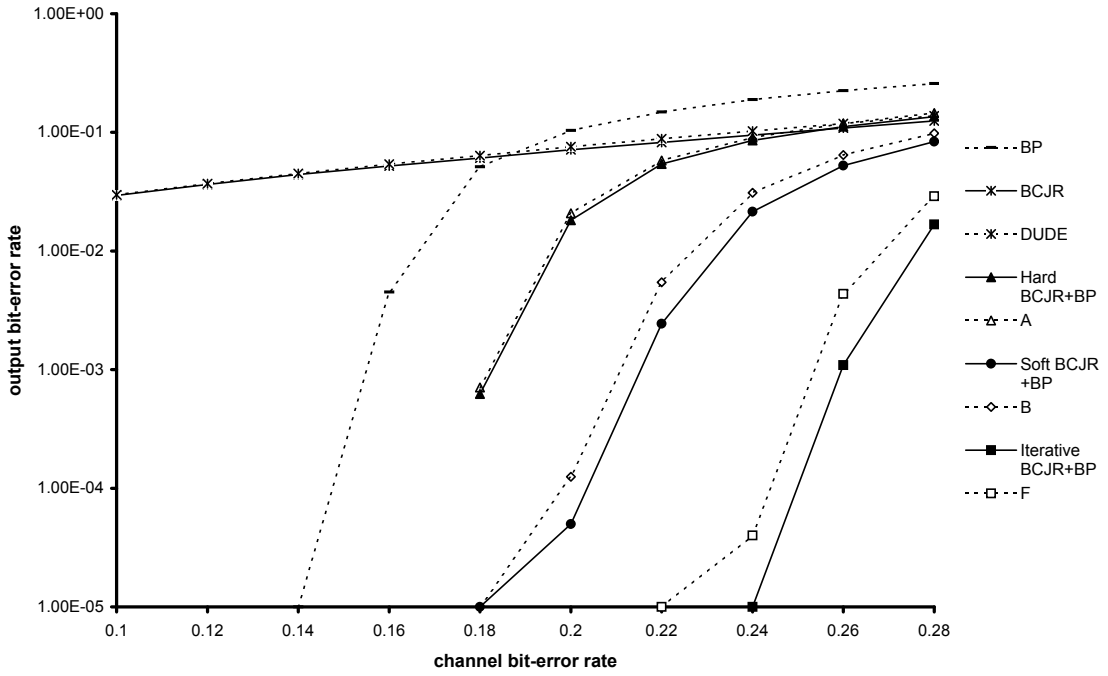

Figure 15: Performance of universal Approaches A, B, and F, and non-universal analogues Hard BCJR+BP, Soft BCJR+BP, and Iterative BCJR+BP for $\pi = .05$. Data length is 10^6 bits.

asymptotically vanishing error probability is possible if and only if

$$\frac{\mathcal{H}(\mathbf{X}|\mathbf{Z})}{1-h(\delta)} \leq \frac{1-R}{R}. \tag{26}$$

Correspondingly, the classical source channel separation theorem implies that compressed channel encoding at rate R source symbols per channel symbol, with asymptotically vanishing error probability, is possible if and only if

$$\frac{\mathcal{H}(\mathbf{X})}{1-h(\delta)} \leq \frac{1}{R}, \tag{27}$$

where $\mathcal{H}(\mathbf{X}) = h(\pi)$ is the entropy rate of the symmetric Markov source with transition probability π.

We can numerically solve for the largest δ for which (26) and (27) are satisfied for the values of R and π used in the above simulations. To carry out this computation for (26), we express $\mathcal{H}(\mathbf{X}|\mathbf{Z})$ as

$$\begin{aligned}\mathcal{H}(\mathbf{X}|\mathbf{Z}) &= h(\delta) + \mathcal{H}(\mathbf{X}) - \mathcal{H}(\mathbf{Z}) \\ &= h(\delta) + h(\pi) - \mathcal{H}(\mathbf{Z}),\end{aligned}$$

where $\mathcal{H}(\mathbf{Z})$ is the entropy rate of the hidden Markov source, which has no closed form expression, but, following [41, 42, 43], can be estimated easily for different values of π and δ using a Monte Carlo method. This technique is based on the asymptotic equipartition property (AEP) for stationary ergodic sources, which states that $(1/n) \log P(Z^n)$ converges to $\mathcal{H}(\mathbf{Z})$ with probability one. The technique thus involves simulating a long realization of Z^n, and computing its log-probability using the standard dynamic programming technique (forward recursions from the hidden Markov process literature). A binary search, coupled with this Monte Carlo entropy rate estimator, can be used to estimate the maximal $\delta = \delta^*$, satisfying (26).

For $R = 1/4$ and a Markov source with $\pi = .05$, the case corresponding to Figure 15, we obtain $\delta^* \approx .339$, which is well above the values of δ achieving low error rates in the figure, e.g., $\delta \approx 0.24$ for a BER between 10^{-5} and 10^{-4} for either scheme F or the non-universal BCJR-based schemes. The largest δ satisfying the compressed channel-encoding condition (27) is easily found using numerical methods to be $\delta^{**} \approx .344$ for these same parameter values. Interestingly, in this case, the theoretical penalty for not compressing the source is fairly small, though it would grow with increasing encoding rate.

The limits on uncompressed encoding have the most relevance to the proposed schemes, and illustrate what may be possible with no additional restrictions on encoding and decoding (such as complexity). Attaining them, however, has not been the objective of the paper, which rather has focused on broadly applicable enhancements to the decoding of existing, practical families of channel codes. Indeed, the gap from the limits observed in the experimental results is due not only to the suboptimality of the decoders and their lack of knowledge of the source statistics, but also substantially to the fact that we are using simple off-the-shelf channel codes which do not operate close to capacity.

VII Conclusion

The proposed methods are practical approaches that harness the dual redundancy present at the channel inputs (due to uncompressed data and error control coding) capitalizing on the knowledge of the error control code at the decoder but without the requirement that the statistics of the data be known at the encoder/decoder. An existing decoder system design, subject to standard protocols and error-correcting codes, can be replaced by the algorithms presented in this paper, without requiring any change in the corresponding encoder system.

A progression of schemes was described, offering a complexity-performance trade-off. The simplest schemes (Approach A) feed the systematic (information) part of the encoded data to a hard-decision denoiser, which in turn feeds a hard-input decoder. This simple configuration already shows significant performance improvement over decoding alone. Further improvements in perfor-

mance, at the cost of a moderate increase in complexity, are obtained by letting the DUDE scheme output soft information, fed to a soft-input decoder (Approach B). The strongest, but also most complex schemes (Approaches C–F) are iterative, feeding back *a posteriori* reliability information on the decoded information symbols to variants of ssDUDE, an enhanced version of DUDE that incorporates such information. These universal decoders show orders of magnitude improvement over state-of-the-art decoders that do not take into account the source redundancy.

While this paper has focused on memoryless channels, analogous denoiser-enhanced decoders can be constructed for channels with memory. The corresponding building blocks for such enhanced decoders would be extensions of the DUDE to channels with memory [32, 33] combined with codes and decoders targeting the relevant channels. The literature on the latter is extensive. Of particular relevance might be recent work on extensions of LDPC/Turbo codes and iterative decoding to channels with memory, as in e.g. [44, 45, 46, 47]. We leave the study of such enhanced decoders for future work.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, 623–656, Jul.-Oct. 1948.

[2] S. Vembu, S. Verdú, and Y. Steinberg, "The source-channel separation theorem revisited," *IEEE Trans. Inform. Theory*, vol. 41, pp. 44–54, Jan. 1995.

[3] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Communications*, vol. 43, pp. 2449–2457, Sep. 1995.

[4] M. G. Luby, "LT codes," *Proc. 43rd IEEE Symp. Foundations of Computer Science*, pp. 271–280, 2002.

[5] D. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.

[6] F. Alajaji, N. Phamdo, and T. Fuja, "Channel codes that exploit the residual redundancy in CELP-encoded speech," *IEEE Trans. Speech Audio Processing*, pp. 325–336, Sept. 1996.

[7] J. Garcia-Frias and J. D. Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Communication Letters*, vol. 1, pp. 111–113, July 1997.

[8] R. E. Van Dyck and D. Miller, "Transport of wireless video using separate, concatenated, and joint source-channel coding," *Proc. of the IEEE*, pp. 1734–1750, Oct. 1999.

[9] T. Hindelang, J. Hagenauer, and S. Heinen, "Source-controlled channel decoding: Estimation of correlated parameters," *3rd ITG Conference Source and Channel Coding*, pp. 259–266, Jan. 2000.

[10] T. Fingscheidt, T. Hindelang, N. Seshadri, and R. Cox, "Combined source/channel decoding: Can a priori information be used twice?," *Proc 2000 IEEE Int. Commun. Conf.*, vol. 3, 2000.

[11] J. Garcia-Frias and J. D. Villasenor, "Joint turbo decoding and estimation of hidden Markov models," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1671–1679, Sep 2001.

[12] M. Bystrom, S. Kaiser, and A. Kopansky, "Soft source decoding with applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, pp. 1108–1120, Oct. 2001.

[13] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy coded sources," *IEEE J. Selected Areas in Communications*, vol. 19, pp. 1680–1696, Sep. 2001.

[14] N. Goertz, "On the iterative approximation of optimal joint source-channel decoding," *IEEE Journal on Sel. Areas in Communication*, vol. 14, no. 9, pp. 1662–1670, Sept. 2001.

[15] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information theory*, vol. IT-20, pp. 284–287, Mar 1974.

[16] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon-limit error-correcting coding and decoding: Turbo codes," in *Proc. 1993 IEEE Int. Conf. Communications, Geneva, Switzerland*, 1993.

[17] R. Gallager, *Low-density parity check codes*. MIT Press, 1963.

[18] N. Wiberg, H. Loeliger, and R. Koetter, "Codes and iterative decoding on general graphs," *Eur. Trans. Telecomm.*, vol. 6, pp. 513–525, Sep/Oct 1995.

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

[20] J. Hagenauer, "Source-controlled channel decoding using the Krichevsky-Tromifov (KT) estimator," *Elektro- und Informationstechnik*, vol. 6, pp. 217 – 220, June 01, 2005.

[21] G. Shamir and K. Xie, "Universal lossless source controlled channel decoding for iid sequences," *IEEE Communications Letters*, vol. 9, pp. 450–452, May 2005.

[22] E. Ordentlich, G. Seroussi, S. Verdú, K. Viswanathan, M. Weinberger, and T. Weissman, "Channel decoding of systematically encoded unknown redundant sources," *2004 Proc. IEEE Symposium on Information Theory*, p. 165, 2004. Chicago, IL.

[23] G. Caire, S. Shamai, and S. Verdú, "Noiseless data compression with low-density parity-check codes," in *Advances in Network Information Theory* (P. Gupta, G. Kramer, and A. J. van Wijngaarden, eds.), vol. 66 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pp. 263–284, American Mathematical Society, 2004.

[24] G. I. Shamir and L. Wang, "Context decoding of low density parity check codes," *2005 Conf. on Information Sciences and Systems*, March 16-18, 2005. The Johns Hopkins University, Baltimore, Md.

[25] K. Xie and G. Shamir, "Context and denoising based decoding of non-systematic turbo codes for redundant data," *2005 Proc. IEEE Symposium on Information Theory*, Sept. 2005. Adelaide, Australia.

[26] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 5–28, Jan. 2005.

[27] E. Ordentlich, G. Seroussi, S. Verdú, M. J. Weinberger, and T. Weissman, "A discrete universal denoiser and its application to binary images," in *Proc. of IEEE International Conference on Image Processing (ICIP'03)*, (Barcelona, Catalonia, Spain), Sep. 2003.

[28] G. Motta, E. Ordentlich, I. Ramírez, G. Seroussi, and M. J. Weinberger, "The DUDE framework for continuous tone image denoising," in *Proc. of IEEE International Conference on Image Processing (ICIP'05)*, (Genoa, Italy), Sep. 2005.

[29] K. Sivaramakrishnan and T. Weissman, "Universal denoising of discrete-time continuous-amplitude signals," *IEEE Int. Symp. on Information Theory*, July 2006. Seattle, WA.

[30] A. Dembo and T. Weissman, "Universal denoising for the finite-input-general-output channel," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1507–1517, April 2005.

[31] G. Gemelos, S. Sigurjonsson, and T. Weissman, "Universal minimax discrete denoising under channel uncertainty," *IEEE Transactions on Information Theory*, vol. IT-52, pp. 3476–3497, Aug. 2006.

[32] C. D. Giurcaneanu and B. Yu, "Efficient algorithms for discrete universal denoising for channels with memory," *IEEE Int. Symp. Information Theory*, Sep. 2005. Adelaide, Australia.

[33] R. Zhang and T. Weissman, "Discrete denoising for channels with memory," *Communications in Information and Systems*, no. 2, pp. 257–288, 2005.

[34] E. Ordentlich, M. J. Weinberger, and T. Weissman, "Efficient pruning of bidirectional context trees with applications to universal denoising and compression," *2004 IEEE Information Theory Workshop*, Oct. 2004. San Antonio, TX.

[35] J. Yu and S. Verdú, "Schemes for bi-directional modeling of discrete stationary sources," *IEEE Transactions on Information Theory*, 2006. To appear.

[36] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," *Proc. International Symposium on Turbo Codes and Related Topics*, p. 111, Sept. 1997. Brest, France.

[37] `http://mobile.yahoo.com`, Aug. 2006.

[38] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam: North-Holland Publishing Co., 1983.

[39] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information theory*, vol. IT-47, pp. 498–519, Feb 2001.

[40] S. Shamai and S. Verdú, "Capacity of channels with side information," *European Transactions on Telecommunications*, vol. 6, pp. 587–600, Sep./Oct. 1995.

[41] D. Arnold and H.-A. Loeliger, "On the information rate of binary-input channels with memory," in *Proc. 2001 IEEE Int. Conf. on Communications*, (Helsinki, Finland), pp. 2692–2695, June 2001.

[42] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite-state ISI channels," in *Proc. 2001 IEEE Globecom*, (San Antonio, TX), pp. 2992–2996, Nov. 2001.

[43] V. Sharma and S. K. Singh, "Entropy and channel capacity in the regenerative setup with applications to Markov channels," in *Proc. 2001 IEEE Int. Symp. Inform. Theory*, (Washington, D.C.), p. 283, June 2001.

[44] A. Kavčić, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Transactions on Information theory*, vol. IT-49, pp. 1636–1652, Jul 2003.

[45] J. Garcia-Frias, "Decoding of low density parity-check codes over finite-state binary Markov channels," *IEEE Trans. Commun.*, vol. 52, pp. 1840–1843, Nov. 2004.

[46] A. W. Eckford, F. R. Kschischang, and S. Pasupathy, "Analysis of low-density parity-check codes for the Gilbert-Elliott channel," *IEEE Trans. Inform. Theory*, vol. 51, Nov. 2005.

[47] J. B. Soriaga, H. D. Pfister, and P. H. Siegel, "Determining and approaching achievable rates of binary intersymbol interference channels using multistage decoding," *IEEE Transactions on Information Theory*, vol. 53, pp. 1416–1429, Apr 2007.