Lossy source coding with sparse graph codes: A variational formulation of soft decimation

Nima Noorshams[1] and Martin J. Wainwright[1,2]
Department of statistics[2] and
Electrical Engineering & Computer Science[1],
University of California Berkeley,
{nshams, wainwrig}@eecs.berkeley.edu

***Abstract*—Various authors have obtained state-of-the-art results in lossy source coding by applying algorithms based on a combination of message-passing and decimation to low-density generator matrix codes, but to date, theoretical understanding of these procedures has been limited. We show that certain forms of soft decimation can be understood as iterative procedures for attempting to maximize a cost function of the node biases. This variational characterization allows us to exhibit appropriate choices of stepsize that ensure convergence to a fixed point, and to provide guarantees on the distortion of the encoding obtained from the fixed point for the case of symmetric Bernoulli sources. Our analysis applies to both an oracle form of soft decimation, in which exact marginals can be computed, and a practical form based on the (reweighted) sum-product algorithm.**

Keywords: Lossy source coding; low-density generator matrix (LDGM) codes; message-passing; decimation schemes; variational methods.

I. Introduction

Sparse graph codes and message-passing algorithms are widely used in coding theory, with a rich literature on both theory and practice for channel coding (e.g., see the book [12] and references therein). A line of more recent work has focused on their use for lossy source coding (e.g., [3], [4], [7], [15]), as well as for related problems involving compression or quantization as sub-routines (e.g., [2], [18], [16]). Much of this work has been based on low-density generator matrix (LDGM) codes, obtained by taking the dual of low-density parity check codes. A desirable consequence of the resulting sparse graph structure of LDGM codes is that they are well-suited to message-passing algorithms such as the sum-product and max-product algorithms.

However, it turns out the standard sum-product algorithm is not sufficient (on its own) to obtain good results in lossy source encoding using LDGM codes. Indeed, there are typically a large number of codewords that achieve roughly the same distortion, so that the marginals from running sum-product tend not to exhibit strong bias towards any particular codeword. Fortunately, message-passing algorithms *in conjunction with decimation* turn out to be very effective for lossy data compression using LDGM and related sparse graph codes. In a message-passing/decimation scheme, the approximate marginals obtained from one run of message-passing are used to fix the values of certain bits (known as hard decimation), or to adjust the biases of nodes (soft decimation). These schemes were first introduced in the context of survey propagation and satisfiability problems [9], and later studied and extended by various researchers (e.g., [2], [18], [3], [4], [10], [11], [15]). Although practical performance of decimation schemes is excellent, obtaining theoretical results has proven more challenging, in part because the act of decimation introduces statistical dependencies among the nodes. This dependency means that the assumptions required for validity of techniques such as density evolution [12] are no longer directly applicable.

In this paper, we provide a variational formulation of certain soft decimation schemes for lossy source encoding using LDGM codes. Many classical message-passing algorithms can be understood from a variational perspective, meaning that they can be cast as methods for solving an underlying optimization problem [14]. In the case of the sum-product algorithm, the seminal work of Yedidia et al. [17] showed that it is attempting to solve a (non-convex) optimization problem known as the Bethe variational problem. These types of variational characterizations provide a lens through which the underlying algorithm can be studied, and can be useful in establishing convergence and characterizing the fixed points. Indeed, as we show in this paper, our variational characterization allows us to establish convergence of soft decimation, and to provide guarantees on the distortion obtained from the resulting fixed point. Although this paper focuses exclusively on symmetric Bernoulli sources, many of the ideas are more generally applicable to other memoryless sources.

The remainder of this paper is organized as follows. In Section II, we provide background on the problem of lossy compression, and the use of LDGM codes in this context. In sectionIII, we state and prove our the main results on the algorithms, and illustrate the performance of the sum-product version.

II. Background

We begin with background on the problem of lossy data compression for binary sources, and how low-density generator matrix (LDGM) codes are used to solve the problem.

a) Lossy compression of binary sources: The goal of lossy source coding is to compress a given sequence up to some distortion. More precisely, for the case of symmetric Bernoulli sources, we are given a bit string $y \in \{0,1\}^n$, in which each $y_i \sim \mathrm{Ber}(\frac{1}{2})$. Consider a binary code $\mathbb{C}$ of rate $R \in (0,1)$, which consists of a collection of 2^{nR} binary sequences contained within the hypercube $\{0,1\}^n$. For any such code, the *minimum distance (MD) source encoding* problem (under Hamming distortion) is to find the codeword $\widehat{y} \in \mathbb{C}$ with minimal Hamming distance to the source sequence y, which amounts to solving the combinatorial optimization problem $\widehat{y} \in \arg\min_{\tilde{y} \in \mathbb{C}} \|\tilde{y} - y\|_1$. The resulting optimal distortion is given by $D(y) = \frac{\|\widehat{y}-y\|_1}{n}$, and our goal is to achieve a low average distortion $D := \mathbb{E}[D(y)]$, where the expectation is taken over the randomness in the source sequence. A brute force approach to the source encoding problem entails a search over the exponentially large codebook (2^{nR} codewords); moreover, in the absence of additional structure, the source encoding problem is known to be computationally intractable. In classical information theory, this computational complexity is not the primary concern, and random codebooks with exponential search are studied. In contrast, the goal of sparse graph codes and message-passing is to find computationally efficient procedures.

b) Low-density generator matrix codes: Low-density parity check (LDPC) codes widely used for channel coding problems [12]. The duals of such codes are known as low-density generator matrix (LDGM) codes, and have proven very effective for problems of lossy data compression. An LDGM code of blocklength n and rate $R = \frac{m}{n}$ can be represented by a sparse generator matrix $G \in \{0,1\}^{n\times m}$, such that each codeword $y \in \mathbb{C}$ is in the range space of G, so can be written in the form $y = Gz$, for some sequence $z \in \{0,1\}^m$ of information bits. For a binary code, all operations are performed in modulo two arithmetic. Assuming that G is full rank, the resulting code has $2^m = 2^{nR}$ codewords.

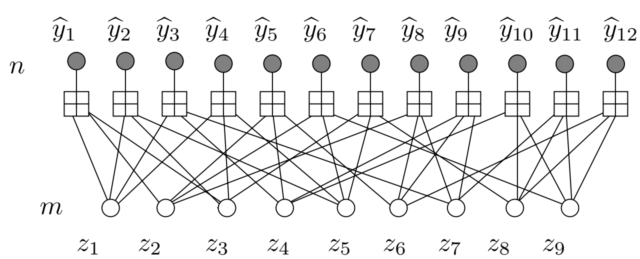


Fig. 1. Factor graph representation of low-density generator matrix (LDGM) code. In the top row, a series of n checks (represented by square nodes) are each connected to a source bit (represented by a gray circular node). The checks also connect to a set of m information bits (represented by white circles) in the bottom row. The illustrated code has $n = 12$ and $m = 9$ for design rate $R = 0.75$, and regular check degree $c = 3$ and bit degree $d = 4$.

The structure of a given generator matrix G can be captured by its factor graph [6], $\mathcal{G} = (V, F, E)$, a bipartite graph where $V = \{1, 2, \cdots, m\}$ and $F = \{1, 2, \cdots, n\}$ are the set of variable and check nodes respectively and $(i, a) \in E$ if and only if $G_{ai} = 1$. For instance, Figure 1 shows a binary linear code of blocklength $n = 12$ (square nodes) and $m = 9$ information bits (circular nodes), represented in factor graph form by its generator matrix $G \in \{0,1\}^{12\times 9}$, with an overall rate of $R = 3/4$. The degrees of the check and variable nodes in the factor graph are $c = 3$ and $d = 4$ respectively, so that the associated generator matrix G has 3 ones in each row, and 4 ones in each column. Throughout this paper, we use letters $a, b, c, \ldots$ to represent factors or checks F, and numbers $1, 2, 3, \ldots$ to represent variable nodes V.

c) Lossy compression using LDGM codes: We now describe how LDGM codes can be used to perform compression of a binary source. Given a $\{0,1\}$-valued source sequence $\{y_a, a = 1, \ldots, n\}$, we assign to each check $a \in F$ the weight $\theta_a = 1 - 2y_a$. For theoretical convenience, we describe the optimization in terms of the spin vector $x \in \{-1,+1\}^m$, obtained from the original sequence $z \in \{0,1\}^m$ via the mapping $z_i \mapsto x_i = 1 - 2z_i$. Given the spin vector $x \in \{-1,+1\}^m$ and the weight vector $\theta \in \{-1,+1\}^n$, we define the cost function

$$J(x;\theta) := \sum_{a=1}^{n} \theta_a \prod_{i\in\mathcal{N}(a)} x_i, \tag{1}$$

where $\mathcal{N}(a)$ is the set of neighbors of check a. In order to solve the MD source encoding problem, we consider the integer program

$$x^* \in \arg\max_{x\in\{-1,1\}^m} J(x;\theta). \tag{2}$$

Any such x^* defines a quantized sequence $\widehat{y} \in \{0,1\}^n$ with elements $\widehat{y}_a = \frac{1-\left(\prod_{i\in\mathcal{N}(a)} x_i^*\right)}{2}$, and the associated Hamming distortion is given by

$$D^* = D(x^*) := \frac{1}{2} - \frac{J(x^*;\theta)}{2n}. \tag{3}$$

III. Soft decimation algorithms and their properties

In this section, we describe some soft decimation algorithms, and state some theoretical results on their properties for lossy source coding. Our first algorithm is based on an "oracle" that can compute exact marginals, whereas our second algorithm computes an approximation to the marginals using the re-weighted sum product algorithm.

A. Weighted distributions

Both of our algorithms are based on a certain weighted distribution over all binary vectors, which we now define. Introducing a linear term which for obvious reasons will be referred to as bias, consider the following cost function

$$K(x;\theta,\gamma) := \sum_{a=1}^{n} \theta_a \prod_{i\in\mathcal{N}(a)} x_i + \sum_{i=1}^{m} \gamma_i x_i. \tag{4}$$

Note that this is a natural generalization of the cost function $J(x;\theta)$ previously defined (1), to which it reduces when

$\gamma = 0$. For a given parameter $\beta > 0$, we then consider a Gibbs distribution of the form

$$\mathbb{P}_{\gamma,\theta}(x) = \exp\big\{\beta\, K(x;\gamma,\theta) - \Phi(\gamma,\theta)\big\}, \tag{5}$$

and we use $\mathbb{E}_{\gamma,\theta}$ to denote expectation under the distribution $\mathbb{P}_{\gamma,\theta}$. In this definition, the quantity

$$\Phi(\gamma,\theta) := \log\bigg[\sum_{x\in\{-1,+1\}^m} \exp\big\{\beta K(x;\gamma,\theta)\big\}\bigg] \tag{6}$$

corresponds to the log normalization constant, and plays an important role in the sequel. It is well-known (e.g., [14]) that the function Φ is strictly convex, and moreover that its derivative has the form

$$\nabla_\gamma \Phi(\gamma;\theta) = \mathbb{E}_{\gamma,\theta}[X] \;\in\; [-1,+1]^m. \tag{7}$$

These properties play an important role in the sequel.

B. Oracle algorithm

We begin by describing an "oracle" version of soft decimation, based on a procedure that returns exact marginal distributions of the Markov random field (5). Of course, this procedure is not practical for large graphs with cycles (due to the intractability of exact marginalization), but it serves to build useful intuition for the practical algorithm that we analyze in the sequel.

Apart from the vector $\theta \in \mathbb{R}^m$ and the positive parameters (β,λ), the algorithm also involves a positive weight and a stepsize parameter $\alpha \in (0,1)$. It generates a sequence of iterates $\{\gamma(t)\}_{t=0}^{\infty}$ contained with $\mathbb{R}^m$ according to steps shown in Figure 2.

Algorithm 1 (Oracle version):

1) At time $t = 0$, initialize at a non-zero vector of biases $\gamma(0) \in [-\lambda,\lambda]^m$.
2) For iterations $t = 0,1,2,\cdots$
   - (i) Given parameters $(\theta,\gamma(t)) \in \mathbb{R}^n \times \mathbb{R}^m$, compute the vector $\mu(t) \in \mathbb{R}^m$ with elements $\mu_i(t) = \mathbb{E}_{\gamma(t),\theta}[X_i]$ for $i = 1,2,\ldots,m$.
   - (ii) Update the biases according
   $$\gamma(t+1) = \gamma(t) + \alpha\big\{\mu(t) - \frac{1}{\lambda}\gamma(t)\big\}. \tag{8}$$
   - (iii) Compute the estimate
   $$x(t) = \operatorname{sgn}(\gamma(t)) \in \{-1,+1\}^m.$$

Fig. 2. Form of a soft-decimation algorithm based on an oracle that computes exact mean parameters.

Of interest to us are the following questions:

- Under what conditions does the sequence $\{\gamma(t)\}_{t=0}^{\infty}$ converge to some fixed point $\widehat{\gamma}$?
- When such convergence does occur, what are the properties of the final estimate $\widehat{x} := \operatorname{sgn}(\widehat{\gamma})$?

The following theorem sheds some light on the behavior of this algorithm. In particular, it is based on linking it to the function

$$\mathcal{F}_\Phi(\gamma;\theta,\lambda) := \Phi(\gamma,\theta) - \frac{1}{2\lambda}\|\gamma\|_2^2. \tag{9}$$

We frequently adopt the shorthand $\mathcal{F}_\Phi(\gamma)$ when the fixed parameters (θ,λ) are clear from the context. The following result shows that Algorithm 1 is a technique for maximizing this function, and will converge to a stationary point under mild conditions. It also provides guarantees on the distortion attained by the final output $\widehat{x}$.

Theorem 1: Given a sequence $\{\gamma(t)\}_{t=0}^{\infty}$ generated by Algorithm 1:

(a) There exist numbers $0 < \epsilon_\ell < \epsilon_u$ such that for all $\alpha \in [\epsilon_\ell,\epsilon_u]$, the sequence $\gamma(t)$ converges to some $\widehat{\gamma} \in [-\lambda,\lambda]^m$ for which $\nabla\mathcal{F}_\Phi(\gamma)\big|_{\gamma=\widehat{\gamma}} = 0$.

(b) Suppose that $|\widehat{\gamma}_i| \geq \lambda\,(1-\frac{1}{m})$ for all $i = 1,2,\ldots,m$. Then for any global maximum $\widehat{\gamma}$ of $\mathcal{F}_\Phi$ and $\widehat{x} = \operatorname{sgn}(\widehat{\gamma})$, we have

$$D(\widehat{x}) \leq D^* + \frac{R}{2\beta}\big(\log 2 + \frac{\lambda}{2}\big). \tag{10}$$

Moreover, for any fixed point $\widehat{\gamma}$, we have

$$D(\widehat{x}) \leq (D^*+\delta) + \frac{\lambda\,R}{2}\min_{x^*\in\mathcal{X}^*(\delta)}\frac{\|\widehat{x}-x^*\|_1}{m}, \tag{11}$$

where $\mathcal{X}^*(\delta)$ is the set of δ-distortion achieving configurations.

We provide the proof of this result in the following section. To interpret its claims, note that for appropriate choices of (λ,β), the bound (10) implies that the gap between $D(\widehat{x})$ and D^* is arbitrarily small. Similarly, if we consider the lower bound (11), note that in the worst-case, we have $\frac{\|\widehat{x}-x^*\|_1}{m} \leq 1$, which yields the worst-case guarantee $D(\widehat{x}) \leq D^* + \frac{\lambda R}{2}$. For any fixed rate $R \in (0,1]$, this term can be made arbitrarily small by our choice of λ.

Before going into the technical details, we pause to provide some intuition as to why maxima of the function $\mathcal{F}_\Phi$ should lead to a configuration $\widehat{x}$ with low distortion, as guaranteed by the bounds (10) and (11). Recall that the source encoding problem can be formulated as maximizing the function J (see equation (2)). We now observe that

$$\begin{aligned}\max_{\gamma\in[-\lambda,\lambda]^m}\max_{x\in\{-1,1\}^m} K(x;\theta,\gamma) &= \max_{x\in\{-1,1\}^m}\max_{\gamma\in[-\lambda,\lambda]^m} K(x;\gamma)\\ &= \max_{x\in\{-1,1\}^m} J(x;\theta) + m\lambda,\end{aligned}$$

since the linear term is maximized when $\gamma = \lambda\operatorname{sgn}(x)$. Therefore, the desired source coding problem (2) is equivalent to solving

$$\gamma^* \in \arg\max_{\gamma\in[-\lambda,\lambda]^m}\max_{x\in\{-1,1\}^m} K(x;\gamma,\theta), \tag{12}$$

since if we set $x^* = \operatorname{sgn}(\gamma^*)$, then we are guaranteed that x^* is optimal for the integer program (2). As a function of

γ, the quantity $\max_{x\in\{-1,1\}^m} K(x;\gamma,\theta)$ is piecewise linear. The log partition function $\Phi(\gamma,\theta)$ from equation (6) is a smooth approximation to this piecewise linear function, one that becomes increasingly accurate for large $\beta > 0$.

C. Proof of Theorem 1

With this intuition in place, we now turn to the technical details.

(a) We first show by induction that $\gamma(t) \in [-\lambda,\lambda]^m$ for all iterations $t = 0, 1, 2, \ldots$. By definition of the algorithm, the claim holds for $t = 0$. Assuming that it holds at iteration t, we then observe that

$$\begin{aligned}\|\gamma(t+1)\|_\infty &\leq (1-\frac{\alpha}{\lambda})\|\gamma(t)\|_\infty + \alpha\|\mu(t)\|_\infty \\ &\leq \big(1-\frac{\alpha}{\lambda}\big)\lambda + \alpha = \lambda,\end{aligned}$$

where we have used the fact that $\|\mu(t)\|_\infty \leq 1$. We now show that for an interval of stepsize choices, the sequence in fact converges to some $\widehat{\gamma} \in [-\lambda,\lambda]^m$ satisfying $\nabla\mathcal{F}_\Phi(\widehat{\gamma}) = 0$. By definition, the function Φ is the cumulant generating function associated with the exponential family (5); therefore, by standard properties of exponential families [14], it is differentiable with $\nabla_\gamma \Phi(\gamma,\theta) = \mathbb{E}_{\theta,\gamma}[X]$. Consequently, we have $\nabla\mathcal{F}_\Phi(\gamma) = \mathbb{E}_{\theta,\gamma}[X] - \frac{\gamma}{\lambda}$, and the update (8) is equivalent to

$$\gamma(t+1) = \gamma(t) + \alpha\nabla\mathcal{F}_\Phi(\gamma(t)),$$

so that it is performing gradient ascent method on the function $\mathcal{F}_\Phi$. Moreover, from the continuity of $\nabla\mathcal{F}_\Phi$, there is a constant $L > 0$ such that

$$\|\nabla\mathcal{F}_\Phi(\gamma) - \nabla\mathcal{F}_\Phi(\gamma')\|_2 \leq L\,\|\gamma-\gamma'\|_2 \quad \forall \gamma,\gamma' \in [-\lambda,\lambda]^m.$$

Therefore, by standard results on gradient ascent for functions with Lipschitz derivatives [1], for a suitably small but fixed $\epsilon \in (0,1)$, the method converges to a stationary point for all $\alpha \in [\epsilon, \min\{1,\ \frac{2-\epsilon}{L}\}]$.

(b) Recall that $x^* \in \arg\max_{x\in\{-1,+1\}^m} J(x;\theta)$, and that $D^* = \frac{1}{2} - \frac{J(x^*;\theta)}{2n}$. For any $\gamma \in [-\lambda,\lambda]^m$, we have

$$\begin{aligned}\mathcal{F}_\Phi(\gamma;\theta,\lambda) \;&\geq\; \beta K(x^*;\theta,\gamma) - \frac{1}{2\lambda}\|\gamma\|_2^2 \\ &\geq\; \beta K(x^*;\theta,\gamma) - \frac{\lambda m}{2}.\end{aligned}$$

Maximizing both sides over $\gamma \in [-\lambda,\lambda]^m$ yields

$$\begin{aligned}\max_{\gamma\in[-\lambda,\lambda]^m} \mathcal{F}_\Phi(\gamma;\theta,\lambda) \;&\geq\; \beta \max_{\gamma\in[-\lambda,\lambda]^m} K(x^*;\theta,\gamma) - \frac{\lambda m}{2} \\ &=\; \beta\big\{J(x^*;\theta) + \lambda m\big\} - \frac{\lambda m}{2},\end{aligned}$$

thereby establishing the lower bound

$$\max_{\gamma\in\mathbb{R}^m} \mathcal{F}_\Phi(\gamma;\theta) \geq \beta\big\{J(x^*;\theta) + \lambda m\big\} - \frac{\lambda m}{2}. \tag{13}$$

We now establish an upper bound on this same quantity in terms of $K(\widehat{x};\theta)$. In order to do so, we require the following auxiliary lemma.

Lemma 1: Suppose that $|\widehat{\gamma}_i| \geq \lambda(1-\frac{1}{m})$ for all $i = 1, 2, \ldots, m$. Then $\widehat{x} := \mathrm{sgn}(\widehat{\gamma})$ is an element of $\arg\max_{x\in\{-1,+1\}^m} K(x;\theta,\widehat{\gamma})$.

See Appendix A for the proof of this claim, which we now use in order to prove the bound (10). First observe that whenever $\widehat{\gamma}$ achieves the global maximum, we have

$$\begin{aligned}\max_{\gamma\in\mathbb{R}^m} \mathcal{F}_\Phi(\gamma;\theta) \;&=\; \mathcal{F}_\Phi(\widehat{\gamma};\theta) \;\leq\; \Phi(\widehat{\gamma};\theta) \\ &\leq\; \beta \max_{x\in\{-1,+1\}^m} K(x;\theta,\widehat{\gamma}) + m\log 2.\end{aligned}$$

By Lemma 1, we then have

$$\begin{aligned}\max_{\gamma\in\mathbb{R}^m} \mathcal{F}_\Phi(\gamma;\theta) &\leq \beta K(\widehat{x};\theta,\widehat{\gamma}) + m\log 2 \\ &\overset{(i)}{=}\; \beta\big\{J(\widehat{x};\theta) + \lambda m\big\} + m\log 2,\end{aligned} \tag{14}$$

where equality (i) uses the fact that $\widehat{x} = \mathrm{sgn}(\widehat{\gamma})$. Combining the lower bound (13) with the upper bound (14), we obtain

$$\beta\big\{J(\widehat{x};\theta) + \lambda m\big\} + m\log 2 \geq \beta\big\{J(x^*;\theta) + \lambda m\big\} - \frac{\lambda m}{2}.$$

Dividing by $2\beta n$ and recalling that $R = m/n$, we conclude that

$$\begin{aligned}D(\widehat{x}) \;&=\; \frac{1}{2} - \frac{J(\widehat{x};\theta)}{2n} \leq \frac{1}{2} - \frac{J(x^*;\theta)}{2n} + \frac{R\log 2}{2\beta} + \frac{R\lambda}{4\beta} \\ &=\; D^* + \frac{R}{2\beta}\big(\log 2 + \frac{\lambda}{2}\big).\end{aligned}$$

which establishes the upper bound (10).

It remains to prove the upper bound (11). Since $\widehat{x} = \mathrm{sgn}(\widehat{\gamma})$ and Lemma 1 implies that $\widehat{x}$ is optimal, so for $x^* \in \mathcal{X}^*(\delta)$ we have

$$\begin{aligned}J(\widehat{x};\theta) + \sum_{i=1}^m |\widehat{\gamma}_i| &= K(\widehat{x};\theta,\widehat{\gamma}) \geq K(x^*;\theta,\widehat{\gamma}) \\ &\geq J(x^*;\theta) + \sum_{i=1}^m |\widehat{\gamma}_i| - \sum_{i=1}^m |\widehat{\gamma}_i|\,|x_i^* - \widehat{x}_i|.\end{aligned}$$

Since $\widehat{\gamma} \in [-\lambda,\lambda]^m$, we conclude that

$$J(\widehat{x};\theta) \geq J(x^*;\theta) - \lambda\|x^* - \widehat{x}\|_1.$$

Dividing both sides by $2n$ and recalling the definition of the distortion (3), complete the proof of the bound (11).

D. Algorithm based on (reweighted) sum-product

We now describe a practical algorithm based on a form of message-passing. In particular, the sum-product algorithm is a message-passing algorithm for computing approximations to the marginals of a Markov random field [6], [17]. In the current context, it can be used to compute approximations to the mean parameters $\mu_i(t) = \mathbb{E}_{\theta,\gamma(t)}[X_i]$ in step (i) of Algorithm 1.

Here we state and prove a theoretical result for a reweighted version [13] of the sum-product algorithm. Like the sum-product algorithm, it is a message-passing algorithm but it involves weights on the edges of the graph. Let $M_{j\to i}$ denote the message passed from node j to node i in the graph. The reweighted sum-product algorithm involves a collection of weights $\{\rho_{ij}, (i,j) \in E\}$ associated with the edges of the graph.

$$M_{j\to i}(x_i) \leftarrow \kappa \sum_{x_j} \psi_j(x_j)\psi_{ij}(x_i,x_j) \frac{\prod_{k\in\mathcal{N}(j)\backslash i} [M_{k\to j}(x_j)]^{\rho_{kj}}}{[M_{i\to j}(x_j)]^{1-\rho_{ij}}}, \tag{15}$$

In this update, the quantity $\kappa > 0$ denotes a normalization constant, whose value may change from line to line, typically chosen so that

In the tree-reweighted version of the algorithm, the edge weights $\rho_{ij} \in [0,1]$ are derived from a probability distribution over spanning trees of the graph [13]; for this choice, we say that the edge weights are valid. Note that the ordinary sum-product algorithm corresponds to the setting $\rho_{ij} = 1$ for all edges, but for a graph with cycles, this choice is invalid. On the other hand, for any valid choice of edge weights, it can be shown that

- the updates (15) have a unique fixed point [13], and
- the updates will converge to this unique fixed point with appropriate scheduling of the message updates [5], [8].

Upon convergence, the fixed point M^* of the messages can be used to compute approximations $\widetilde{\mu}$ to the single node marginals using the equation

$$\tau_i(x_i) = \kappa_i \ \psi_i(x_i) \prod_{j\in\mathcal{N}(i)} [M^*_{j\to i}(x_i)]^{\rho_{ji}}, \tag{16}$$

where the normalization constant $\kappa_i > 0$ is chosen to ensure that $\sum_{x_i} \tau_i(x_i) = 1$. A final useful fact is that the algorithm is actually computing the value of a function Ψ that provides an upper bound on the cumulant generating function—viz.

$$\Phi(\gamma,\theta) \leq \Psi(\gamma,\theta). \tag{17}$$

We now describe how this algorithm can be used to implement a practical version of Algorithm 1.

Algorithm 2 (Message-passing version):

1) At time $t = 0$, initialize at a non-zero vector of biases $\gamma(0) \in [-\lambda,\lambda]^m$.
2) For iterations $t = 0, 1, 2, \cdots$
   (i) Given parameters $(\theta, \gamma(t)) \in \mathbb{R}^n \times \mathbb{R}^m$, run the reweighted sum-product algorithm on the Markov random field $\mathbb{P}_{\theta,\gamma(t)}$ to compute approximate mean parameters $\widetilde{\mu}(t) \in \mathbb{R}^m$ with elements
   $$\widetilde{\mu}_i(t) = \tau_i(+1) - \tau_i(-1). \tag{18}$$
   (ii) Update the biases according
   $$\gamma(t+1) = \gamma(t) + \alpha\big\{\widetilde{\mu}(t) - \frac{1}{\lambda}\gamma(t)\big\}. \tag{19}$$
   (iii) Compute the estimate
   $$x(t) = \mathrm{sgn}(\gamma(t)) \in \{-1,+1\}^m.$$

Fig. 3. Form of a soft-decimation algorithm based on an oracle that computes approximate mean parameters using (reweighted) sum-product.

As with Algorithm 1, this algorithm turns out to have a variational interpretation in terms of the function

$$\mathcal{F}_\Psi(\gamma,\theta) := \Psi(\gamma,\theta) - \frac{1}{2\lambda}\|\gamma\|_2^2. \tag{20}$$

Note that this function is the same as $\mathcal{F}_\Phi$, but with the surrogate function Ψ replacing the true cumulant generating function Φ.

Theorem 2: Given a sequence $\{\gamma(t)\}_{t=0}^\infty$ generated by Algorithm 2:

(a) There exist numbers $0 < \epsilon_\ell < \epsilon_u$ such that for all $\alpha \in [\epsilon_\ell, \epsilon_u]$, the sequence $\gamma(t)$ converges to some $\widehat{\gamma} \in [-\lambda,\lambda]^m$ for which $\nabla\mathcal{F}_\Psi(\gamma)\big|_{\gamma=\widehat{\gamma}} = 0$.

(b) Suppose that $|\widehat{\gamma}_i| \geq \lambda\,(1-\frac{1}{m})$ for all $i = 1,2,\ldots,m$. Then for any global maximum $\widehat{\gamma}$ of $\mathcal{F}_\Psi$ and $\widehat{x} = \mathrm{sgn}(\widehat{\gamma})$, we have
$$D(\widehat{x}) \leq D^* + \frac{R}{2\beta}\big(\log 2 + \frac{\lambda}{2}\big). \tag{21}$$
Moreover, for any fixed point $\widehat{\gamma}$, we have
$$D(\widehat{x}) \leq (D^*+\delta) + \frac{\lambda\,R}{2}\min_{x^*\in\mathcal{X}^*(\delta)} \frac{\|\widehat{x}-x^*\|_1}{m}, \tag{22}$$
where $\mathcal{X}^*(\delta)$ is the set of δ-distortion achieving configurations.

Proof: (a) This claim follows by an argument entirely analogous to the proof of Theorem 1(a). This same argument also shows that $\widehat{\gamma} \in [-\lambda,\lambda]^m$, as before.

(b) Recall that $x^* \in \arg\max_{x\in\{-1,+1\}^m} J(x;\theta)$, and that $D^* = \frac{1}{2} - \frac{J(x^*;\theta)}{2n}$. For any $\gamma \in [-\lambda,\lambda]^m$, we have $\mathcal{F}_\Psi(\gamma;\theta) \geq \mathcal{F}_\Phi(\gamma;\theta)$ by construction. Therefore, following the same steps as in the proof of Theorem 1(b) leads to the lower bound

$$\max_{\gamma\in\mathbb{R}^m} \mathcal{F}_\Psi(\gamma;\theta) \geq \beta\big\{J(x^*;\theta) + \lambda m\big\} - \frac{\lambda m}{2}. \tag{23}$$

We now need to establish an analogue of Lemma 1, in particular showing that $\widehat{x} = \mathrm{sgn}(\widehat{\gamma})$ achieves the maximum, and that this maximum can be used to upper bound $\mathcal{F}_\Psi$. This step requires more work, but the key results are summarized in the following:

Lemma 2: Given a fixed point $\widehat{\gamma}$ of Algorithm 2, suppose that $|\widehat{\gamma}_i| \geq \lambda\,(1-\frac{1}{m})$ for all $i = 1,2,\ldots,m$. Then $\widehat{x} := \mathrm{sgn}(\widehat{\gamma})$ is an element of $\arg\max_{x\in\{-1,+1\}^m} K(x;\theta,\widehat{\gamma})$, and moreover if $\widehat{\gamma}$ is the global max of $\mathcal{F}_\Psi(\gamma;\theta)$, then we have

$$\mathcal{F}_\Psi(\widehat{\gamma};\theta) \leq \beta\,\big\{J(\widehat{x};\theta) + \lambda m\big\} + m\log 2. \tag{24}$$

Proof: Some further background on the reweighted sum-product algorithm [13] is required for this proof. Given an undirected graph $\mathcal{G}$, consider a collection $\{T \in \mathfrak{T}\}$ of its spanning trees, and let $\{\rho(T), T \in \mathfrak{T}\}$ be a probability distribution over this trees. For a given vector of parameters $\begin{bmatrix}\widehat{\gamma} & \theta\end{bmatrix} \in \mathbb{R}^m \times \mathbb{R}^n$, suppose that we can find a set of tree-structured parameters $\{(\gamma(T), \theta(T)), T \in \mathfrak{T}\}$ such that

$$\sum_{T \in \mathfrak{T}} \rho(T) \begin{bmatrix}\gamma(T) & \theta(T)\end{bmatrix} = \begin{bmatrix}\widehat{\gamma} & \theta.\end{bmatrix} \tag{25}$$

In this case, by convexity of the cumulant generating function [14] and Jensen's inequality, we have the upper bound

$$\Phi(\widehat{\gamma}, \theta) \leq \sum_{T \in \mathfrak{T}} \rho(T) \Phi(\gamma(T), \theta(T)).$$

The reweighted sum-product minimizes this upper bound over the collection of all possible parameters satisfying equation (25). It can be shown [13] that at the optimum, the algorithm finds a set of parameters $\{(\gamma(T), \theta(T)), T \in \mathfrak{T}\}$ satisfying equation (25) such that

$$\mathbb{E}_{\gamma(T),\theta(T)}[X] = \underbrace{\nabla\Psi(\widehat{\gamma}, \theta)}_{\widetilde{\mu}} \qquad \text{for all } T \in \mathfrak{T}, \tag{26}$$

where $\mathbb{E}_{(\gamma(T),\theta(T))}$ denotes expectation under the tree-structured distribution on T. The vector $\widetilde{\mu} \in [-1, +1]^m$ corresponds to the single-node marginals returned by the algorithm. Moreover, by construction of the algorithm, the value $\Psi(\widehat{\gamma}, \theta)$ is given by

$$\Psi(\widehat{\gamma}, \theta) = \sum_{T \in \mathfrak{T}} \rho(T) \Phi(\gamma(T), \theta(T)). \tag{27}$$

We now turn to the proof of Lemma 2. As in the proof of Lemma 1, we may assume without loss of generality that $\widehat{\gamma}_i > 0$ for all $i = 1, 2, \ldots, m$, so that $\widehat{x} = \operatorname{sgn}(\widehat{\gamma}) = \mathbb{1}$. Since the function $(\gamma, \theta) \mapsto \max_{x \in \{-1,+1\}^m} K(x; \gamma, \theta)$ is convex, using equation (25) and Jensen's inequality, we have

$$\max_{x \in \{-1,+1\}^m} K(x; \widehat{\gamma}, \theta) \leq \sum_{T \in \mathfrak{T}} \rho(T) \max_x K(x; \gamma(T), \theta(T)),$$

with equality holding if there is a configuration $\widehat{x}$ that achieves a common maximum. Therefore, our next step is to show that $\widehat{x} \in \arg\max_{x \in \{-1,+1\}^m} K(x; \gamma(T), \theta(T))$ for each tree $T \in \mathfrak{T}$.

By Theorem 2(a), for any fixed point of Algorithm 2, we have $\nabla_\gamma \Psi(\widehat{\gamma}, \theta) = \frac{\widehat{\gamma}}{\lambda}$. Consequently, using equation (26), the condition $|\widehat{\gamma}_i| \geq \lambda\,(1 - \frac{1}{m})$ implies that

$$\mathbb{E}_{(\gamma(T),\theta(T))}[X] \geq 1 - \frac{1}{m}, \qquad \text{for every tree } T \in \mathfrak{T}.$$

Following the proof of Lemma 1, we conclude that $\widehat{x} \in \arg\max_{x \in \{-1,+1\}^m} K(x; \gamma(T), \theta(T))$ for each tree $T \in \mathfrak{T}$, and hence that $\widehat{x} \in \arg\max_{x \in \{-1,+1\}^m} K(x; \widehat{\gamma}, \theta)$.

It remains to establish the bound (24). We have

$$\begin{aligned}\max_{\gamma \in \mathbb{R}^m} \mathcal{F}_\Psi(\gamma; \theta) \;&=\; \mathcal{F}_\Psi(\widehat{\gamma}; \theta) \;\leq\; \Psi(\widehat{\gamma}, \theta) \\ &=\; \sum_{T \in \mathfrak{T}} \rho(T)\, \Phi(\gamma(T), \theta(T)),\end{aligned}$$

using equation (27). Now

$$\begin{aligned}\Phi(\gamma(T), \theta(T)) &\leq \beta \max_{x \in \{-1,+1\}^m} K(x; \gamma(T), \theta(T)) + m \log 2 \\ &= \beta K(\widehat{x}; \gamma(T), \theta(T)) + m \log 2,\end{aligned}$$

since $\widehat{x}$ achieves the maximum for each problem $(\gamma(T), \theta(T))$. Consequently, taking the weighted sum over trees, we conclude that

$$\begin{aligned}\mathcal{F}_\Psi(\widehat{\gamma}; \theta) &\leq \beta \sum_{T \in \mathfrak{T}} \rho(T) K(\widehat{x}; \gamma(T), \theta(T)) + m \log 2 \\ &= \beta K(\widehat{x}; \widehat{\gamma}, \theta) + m \log 2,\end{aligned}$$

where the final equality uses the linearity of K in $(\widehat{\gamma}, \theta)$ and the relation (25). Finally, since $\widehat{x} = \operatorname{sgn}(\widehat{\gamma})$ and $\|\widehat{\gamma}\|_\infty \leq \lambda$, we have

$$\beta K(\widehat{x}; \widehat{\gamma}, \theta) + m \log 2 \leq \beta\big\{J(\widehat{x}; \theta) + \lambda m\big\} + m \log 2,$$

as claimed. ∎

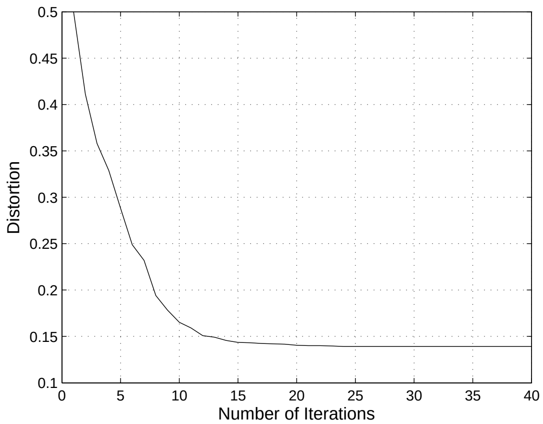
Returning to the main thread of the proof, combining the upper bound (24) with the lower bound (23) yields the claim (21). Finally, we need to prove the lower bound (22). Since $\widehat{x} = \operatorname{sgn}(\widehat{\gamma})$ and Lemma 2 implies that $\widehat{x}$ is optimal, this bound follows by the same argument as in Theorem 1. ∎

In order to illustrate performance of the algorithms, we performed some numerical simulations. (See the papers [4], [15] for more comprehensive sets of simulations using similar algorithms.) Figure 4 illustrates the average distortion verses the number of iterations in both cases of ordinary sum-product (edge weights $\rho_{ij} = 1$) as well as reweighted sum-product (edge weights $\rho_{ij} = .9$) for the case of LDGM-(d,2). We performed simulations using codes with blocklength $m = 500$, rate $R = 0.5$, and we set the message-passing parameters as $\beta = \log(m) \approx 6.2$, $\lambda = 1$ and $\alpha = 0.1$.

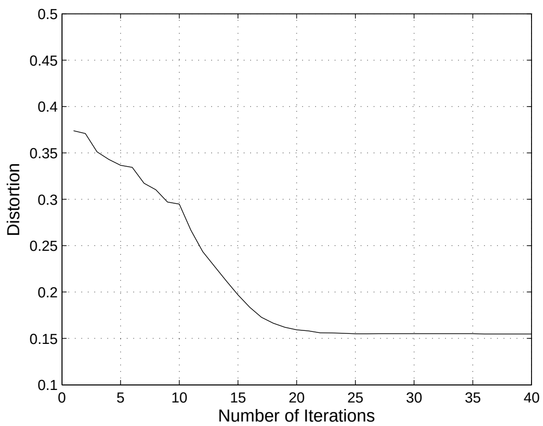
It is interesting to notice that since the distortion (D) only depends on the sign of the limit point, it converges really fast–within 15—20 iterations—even though the bias (γ) may converge somewhat more slowly This makes the algorithm practically very efficient. In our experience to date, we have that reweighted BP is slower to converge than ordinary BP, and yields slightly higher distortion.

IV. Discussion

In this paper, we have shown that soft decimation algorithms can be understood from a variational perspective, namely as methods for trying to maximize a certain cost function. This variational perspective allowed us to specify stepsize choices that ensure convergence, and to prove bounds on the sub-optimality of the final solutions. There remain a number of questions left open in this paper. It remains to determine the minimal setting of the parameter β that ensures that the resulting biases satisfy the conditions of our theorem. In practice, we have observed that the choice $\beta \approx \log m$ is sufficient. We also plan to study the trade-offs between the settings of β and λ. Whereas the theoretical bounds benefit from small λ, convergence is aided by larger



(a)



(b)

Fig. 4. Average distortion vs number of iterations, ordinary sum-product (a), reweighted sum-product (b). Both sets of simulations were performed on the same rate $R = 0.5$ LDGM code with blocklength $m = 500$, and using message-passing parameters $\beta = \log(m) \approx 6.2$, $\lambda = 1$, and $\alpha = 0.1$.

values. We suspect that there are connections between the performance of soft decimation, and the clustering structure of the δ-optimal solutions, which should be further explored. Finally, it would be interesting to see if variants of these ideas can be used to improve the performance of channel coding algorithms.

APPENDIX

Without loss of generality, we may assume that $\widehat{\gamma}_i > 0$ for all $i = 1, 2, \ldots, m$, so that $\widehat{x} = \text{sgn}(\widehat{\gamma}) = \mathbb{1}$. Letting $\widehat{\mu}_i = \mathbb{E}_{\widehat{\gamma}}[X_i]$, we observe that the relation $\nabla \mathcal{F}_\Phi(\widehat{\gamma}) = 0$ and the assumption $\widehat{\gamma}_i \geq \lambda\,(1 - \frac{1}{m})$ imply that $\widehat{\mu}_i \geq 1 - \frac{1}{m}$. Define $\widehat{p}_i = \mathbb{P}_{\widehat{\gamma}}[X_i = -1]$, and observe that $\widehat{p}_i = \frac{1-\widehat{\mu}_i}{2}$, so that the condition $1 - \widehat{\mu}_i \leq \frac{1}{m}$ implies that $\widehat{p}_i \leq \frac{1}{2m}$. Consequently, we have

$$\sum_{x \neq \widehat{x}} \mathbb{P}_{\widehat{\gamma}}(x) \leq \sum_{i=1}^{m} \sum_{\{x \neq \widehat{x}, x_i = -1\}} \mathbb{P}_{\widehat{\gamma}}(x) = \sum_{i=1}^{m} \widehat{p}_i \leq \frac{1}{2},$$

which implies that $\mathbb{P}_{\widehat{\gamma}}(\widehat{x}) \geq \frac{1}{2}$, so that $\widehat{x} \in \arg\max_{x \in \{-1,+1\}^m} K(x; \theta, \widehat{\gamma})$ as claimed.

a) Acknowledgements:: This work was partially supported by NSF grants CAREER CCF-0545862 and CCF-0635372, and a grant from the Marvell Corporation.

REFERENCES

[1] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, Belmont, MA, 1995.

[2] A. Braunstein, F. Kayhan, G. Montorsi, and R. Zecchina. Encoding for the Blackwell channel with reinforced belief propagation. In *Proc. IEEE International Symposium on Information Theory*, Nice, France, 2007.

[3] S. Ciliberti, M. Mézard, and R. Zecchina. Message-passing algorithms for non-linear nodes and data compression. *Complexus*, 3:58, 2006.

[4] T. Filler and J. Fridrich. Binary quantization using belief propagation with decimation over factor graphs of LDGM codes. In *Proc. Allerton Conf. Comm. Control and Computing*, September 2007.

[5] V. Kolmogorov. Convergent tree-reweighted message-passing for energy minimization. *IEEE Trans. PAMI*, 28(10):1568–1583, October 2006.

[6] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Info. Theory*, 47(2):498–519, February 2001.

[7] E. Martinian and J.S. Yedidia. Iterative quantization using codes on graphs. In *Allerton Conference on Control, Computing, and Communication*, October 2003.

[8] T. Meltzer, A. Globerson, and Y. Weiss. Convergent message-passing algorithms: A unifying view. In *Uncertainty in Artificial Intelligence*, 2009.

[9] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297, 812, 2002.

[10] A. Montanari, F. R. Tersenghi, and G. Semerjian. Solving constraint satisfaction problems through belief propagation-guided decimation. In *Proc. Allerton Conf. on Comm., Control, and Computing*, 2007.

[11] T. Murayama. Thouless-Anderson-Palmer approach for lossy compression. *Physical Review E*, 69:035105(1)–035105(4), 2004.

[12] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.

[13] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *IEEE Trans. Info. Theory*, 51(7):2313–2335, July 2005.

[14] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1—305, December 2008.

[15] M. J. Wainwright, E. Maneva, and E. Martinian. Lossy source compression using low-density generator matrix codes: Analysis and algorithms. *IEEE Trans. Information Theory*, 56(3):1351–1368, March 2010.

[16] M. J. Wainwright and E. Martinian. Low-density codes that are optimal for binning and coding with side information. *IEEE Trans. Info. Theory*, 55(3):1061–1079, March 2009.

[17] J.S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Trans. Info. Theory*, 51(7):2282–2312, July 2005.

[18] W. Yu and M. Aleksic. Coding for the Blackwell channel: A survey propagation approach. In *International Symposium on Information Theory*, pages 1583–1587, Adelaide, Australia, September 2005.