Lossy Source Compression of Non-Uniform Binary Source via Reinforced Belief Propagation over GQ-LDGM Codes

Jianping Zheng, Baoming Bai, and Ying Li

In this letter, we consider the lossy coding of a non-uniform binary source based on GF(q)-quantized low-density generator matrix (LDGM) codes with check degree d_c=2. By quantizing the GF(q) LDGM codeword, a non-uniform binary codeword can be obtained, which is suitable for direct quantization of the non-uniform binary source. Encoding is performed by reinforced belief propagation, a variant of belief propagation. Simulation results show that the performance of our method is quite close to the theoretic rate-distortion bounds. For example, when the GF(16)-LDGM code with a rate of 0.4 and block-length of 1,500 is used to compress the non-uniform binary source with probability of 1 being 0.23, the distortion is 0.091, which is very close to the optimal theoretical value of 0.074.

Keywords: Source coding, lossy compression, LDGM codes, belief propagation.

I. Introduction

Inspired by the success of the low-density parity check (LDPC) codes and belief propagation algorithm in approaching the Shannon capacity, similar techniques have been proposed for lossy source coding. In particular, low-density generator matrix (LDGM) codes in conjunction with variants of message-passing algorithms, for example, survey propagation [1], have shown the potential to approach the rate-distortion bound [2]-[7]. However, almost all the existing results exclusively focused on uniformly distributed sources. Also, the extension to sources with general distributions is not straightforward.

To address the problem of lossy compression of non-uniform binary sources, the GF(q)-quantized LDGM (GQ-LDGM) codes and multi-level coding scheme have recently been proposed in [8] and [9], respectively. In both methods, by employing quantization, a non-uniform binary codeword can be obtained, which is then suitable for direct quantization of the non-uniform binary source. The message-propagation/decimation algorithm is utilized to perform compression, which has a complexity of $\mathcal{O}(nK)$ with n and K being the source block-length and the times of decimation, respectively.

In this work, we present a novel method to compress the non-uniform binary source based on GQ-LDGM codes with check degree d_c=2. Unlike the method proposed in [8], we employ reinforced belief propagation (RBP), a variant of belief propagation, to perform the compression over GQ-LDGM codes. According to the reformulation of the Thouless-Anderson-Palmer method in [7], RBP appeared first in [3] and was defined formally in [10]. In RBP, the priors are updated using the current message after a few message-propagation iterations, and the message-propagation is then started with the new priors. The complexity of the RBP algorithm is linear with the source block-length n, that is, $\mathcal{O}(n)$, since no decimation step is included in this algorithm. On the other hand, it is well known that the cyclic GF(q) LDPC code, dual of the LDGM codes employed herein, has better error rate results with larger q [11]. Therefore, to achieve good performance with RBP, we increase the size of the finite field q from 2, 3, and 5 in [8] to 8

Manuscript submitted June 11, 2010; revised July 7, 2010; accepted July 21, 2010.

This work was supported by the National Natural Science Foundation of China (No. U0635003, 60972046, 60903199), the Fundamental Research Funds for the Central Universities (No. GY10000901028), and the National S&T Major Special Project (No. 2009ZX03003-011-05, 2010ZX03003-003).

Jianping Zheng (phone: +86 029 88201015, email: jpzheng@xidian.edu.cn), Baoming Bai (email: bmbai@mail.xidian.edu.cn), and Ying Li (email: yli@mail.xidian.edu.cn) are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China.

doi:10.4218/etrij.10.0210.0204

and 16 in our method. Simulation results show that the performance of our method is an improvement over that in [8] and is quite close to the theoretic rate-distortion bound.

It should be noted that a related work is reported in [12], where both the RBP and the GF(q) sparse-graph code are also utilized to perform the lossy data compression. However, unlike in our work, the sparse-graph code used in [12] is the LDPC code, and the source considered in [12] is only the binary uniformly distributed source.

II. Lossy Coding and GQ-LDGM Codes

Consider a Ber(p) source where the probabilities of 0 and 1 are 1–p and p, respectively. Any particular independent and identically distributed (i.i.d.) realization $\boldsymbol{y} \in \{0,1\}^n$ is referred to as a source sequence. The goal is to compress source sequence $\boldsymbol{y}$ by mapping it to shorter vector $\boldsymbol{z} \in \{0,1,\ldots,q-1\}^m$ with code rate $R = \log_2 q \cdot m/n < 1$. The source decoder then maps the compressed sequence $\boldsymbol{z}$ to a reconstructed source sequence $\boldsymbol{x}$. The quality of the source compression is then measured by the average Hamming distortion $D := \mathrm{E}\left[d_H(\boldsymbol{x},\boldsymbol{y})\right]$ with $d_H(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{n} \left|x_i - y_i\right|/n$. For the Ber($p$) source, the rate distortion function is well known to take the form $H(D)=H(p)-R$ with $H(\cdot)$ being the binary entropy function.

In the GQ-LDGM approach to source coding, the encoding phase amounts to mapping a given source sequence $\boldsymbol{y}$ to an information vector $\boldsymbol{z}$ according to

$$\boldsymbol{z} = \arg \min_{z' \in \{0,1,\ldots,q-1\}^m} d_H\left(\boldsymbol{x} = Q(\boldsymbol{t}), \boldsymbol{y}\right) \tag{1}$$

with $\boldsymbol{t} = \boldsymbol{G}\boldsymbol{z}'$. Here $\boldsymbol{G}$ is the $n \times m$ generator matrix of the GF(q) LDGM code, $Q(\cdot)$ is the quantizer based on a threshold Q_t. In practice, $Q(t)=0$ for all $t<Q_t$, while $Q(t)=1$ otherwise. Hence, the employed GQ-LDGM codes have elements assuming the value of 1 with probability $r = (q - Q_t)/q$. Decoding is straightforward: we simply form $\boldsymbol{x} = Q(\boldsymbol{t} = \boldsymbol{G}\boldsymbol{z})$.

It is convenient to represent a given (n, m) LDGM code over GF(q)={0, 1,…, q–1}, specified by generator matrix $\boldsymbol{G}$, as a factor graph made by n checks a and n constrained variables t_a, $a \in C=\{0, 1,\ldots, n-1\}$, plus m free variables z_i, $i \in V=\{0, 1,\ldots, m-1\}$. As illustrated in Fig. 1, each check a is connected to t_a on one side and to $z_{V(a)}$ on the other side, where $V(a)=\{i \in V | g_{ai} \neq 0\}$ and $|V(a)|=2$ since $d_c=2$. Each such connection is labeled with a weight $g_{ai} \in \mathrm{GF}(q)\backslash\{0\}$. Each variable z_i is then connected to checks in $C(i)=\{a \in C | g_{ai} \neq 0\}$. Variables take values on GF(q), that is, $z_i \in \mathrm{GF}(q)$, $i \in V=\{0, 1,\ldots, m-1\}$. Checks are satisfied if

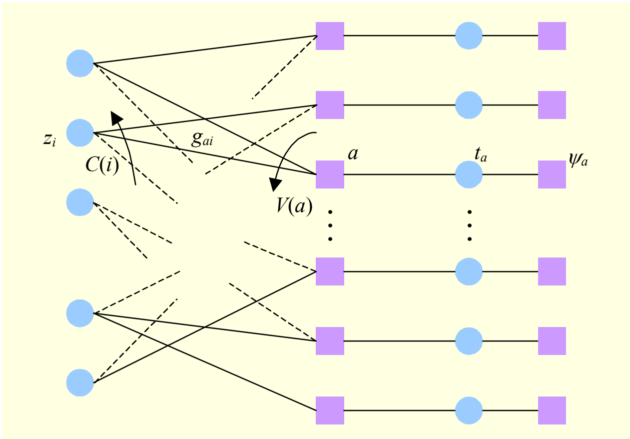

Fig. 1. Factor graph of original LDGM code, completed with compatibility function of constrained codeword symbol.

$t_a = \sum_{i \in V(a)} g_{ai} z_i$, and the z and t such that all checks are satisfied from the information and codeword symbols, respectively, of the GF(q)-LDGM code.

III. Binary Source Compression Using RBP

In this section, we describe the RBP for identifying the information bits of a GQ-LDGM code.

First, define the compatibility function, shown in Fig. 1, for the constrained variable as

$$\psi_a(t_a) = \begin{cases} \mathrm{e}^{\beta}, & \text{if } Q(t_a) = y_a, \\ \mathrm{e}^{-\beta}, & \text{otherwise,} \end{cases} \tag{2}$$

where y_a is the source bit to be quantized by the constrained variable t_a.

Next, let $\boldsymbol{\mu}^l_{z\to a}$ denote the message vector from free variable z to check node a at the l-th iteration. For each symbol $k \in \mathrm{GF}(q)$, the k-th component of $\boldsymbol{\mu}^l_{z\to a}$ is the probability that variable z takes the value k and is denoted by $\boldsymbol{\mu}^l_{z\to a}(k)$. Similarly, $\boldsymbol{\mu}^l_{a\to z}$ denotes the message vector from check node a to free variable node z at the iteration l, and $\boldsymbol{\mu}^l_{a\to z}(k)$ is its k-th component. Also, let $\boldsymbol{\lambda}^l_z$ denote the marginal function of free variable z at the iteration l, and $\boldsymbol{\lambda}^l_z(k)$ is its k-th component.

Then, the RBP update rule can be summarized as

Initialization:

$$\boldsymbol{\mu}^1_{z\to a}(k) = 1/q \pm \text{dither}. \tag{3}$$

Check node update:

$$\boldsymbol{\mu}^l_{a\to z}(k) = \sum_{g_{az'} k_1 + k_2 = -g_{za} k} \boldsymbol{\mu}^l_{z' \in V(a)\backslash\{z\} \to a}(k_1) \cdot \psi_a(k_2). \tag{4}$$

Marginal function update:

$$\lambda^l(k) \propto \prod_{a \in C(z)} \boldsymbol{\mu}^l_{a\to z}(k). \tag{5}$$

Variable node update:

$$\boldsymbol{\mu}_{z\to a}^{l+1}(k) \propto \varphi_z^l(k) \prod_{a' \in C(z)\setminus\{a\}} \boldsymbol{\mu}_{a'\to z}^{l}(k). \quad (6)$$

The major character of the RBP is that the *a priori* probability of free variable node z varies during iterations, as shown in (6). In practice, at the l-th iteration, the *a priori* probability of $z=k\in\mathrm{GF}(q)$ is updated by

$$\varphi_z^l(k) = \alpha \cdot \frac{1}{q} + (1-\alpha)\lambda^l(k). \quad (7)$$

Note that the complexity of the RBP update is dominated by the complicated convolutional operation at the check node update (4). Fortunately, by employing Fourier transform, an efficient implementation of the check node update can be obtained. Also, the total number of computations per RBP iteration is $\mathcal{O}(nq\log_2 q)$ [10]. Therefore, noting that the complexity of the proposed method in [8] is $\mathcal{O}(nKq\log_2 q)$ with $K\in[10,100]$, our method has a lower complexity than the method in [8] since the q in our method is medium, that is, q=8 and 16 in our simulations.

The RBP algorithm stops when the algorithm achieves a predetermined maximum iteration number L or the algorithm convergences, that is, for all variables z and all checks a, $\boldsymbol{\mu}_{a\to z}^{l} = \boldsymbol{\mu}_{a\to z}^{l-1}$.

IV. Simulation Results

In these simulations, we derive the GQ-LDGM code from PEG-construction [13] GF(q)-LDGM code with regular degree distribution. We fix the check degree d_c=2 and derive, for each size q of employed finite field, a suitable variable degree d_v for obtaining a rate of R bits/sample, as shown in Table 1.

In each simulation, we generate a random i.i.d. source in GF(2) with a probability of 1 equal to the values of p shown in Table 1. In order to make GQ-LDGM codeword statistics match the optimal distribution, the threshold parameters Q_t are also chosen according to Table 1.

In all simulations, the parameter β in the compatibility function (2) is set to be

$$\beta = \frac{1}{2}\ln\frac{1-D(R,p)}{D(R,p)} \quad (8)$$

as suggested in [8], and its value is shown in Table 1. Further, we fix the parameter at α=0.9 in the update of the *a priori* probability (7), and the maximum iteration number L=300.

The results of our method are reported in Fig. 2 with R=0.4, q=16, and n=300, 1,500, and 12,000. For comparison, we also plot the curves of the rate-distortion bound $H(D)=H(p)-R$ and the time-sharing bound. As seen in Fig. 2, our method's performance is quite close to that of the rate-distortion bound, and it is always better than the time-sharing bound. In fact, for p=0.23, 0.365, 0.42, and 0.5, the distortions are 0.091, 0.147, 0,157, and 0.162, respectively, which are very close to the optimal theoretical values (the corresponding theoretical values are 0.074, 0.126, 0.139, and 0.147). Further, note that the performance of our method is nearly independent from the source block-length. Similar conclusions can be drawn from Fig. 3 regarding R=0.6 and q=8.

The comparison of our method and the method proposed in [8] is given in Fig. 4 with R=0.5 and n=1,000. Our method has

Table 1. Experiment set in simulations.

Experiment set		1	2	3	4
p		0.23	0.365	0.42	0.50
R=0.4	Q_t	12	10	9	8
q=16	d_v	20	20	20	20
	r	0.25	0.375	0.4375	0.5
	β	1.27	0.97	0.92	0.88
R=0.6	Q_t	6	5	5	4
q=8	d_v	10	10	10	10
	r	0.25	0.375	0.375	0.5
	β	1.80	1.33	1.26	1.22
R=0.5	Q_t	12	10	9	8
q=16	d_v	16	16	16	16
	r	0.25	0.375	0.4375	0.5
	β	1.53	1.19	1.10	1.05
Method in [8]	q	5	3	5	2
R=0.5	Q_t	4	2	3	1
	d_v	9	6	9	4
	r	0.20	0.333	0.4	0.5
	β	1.53	1.19	1.10	1.05

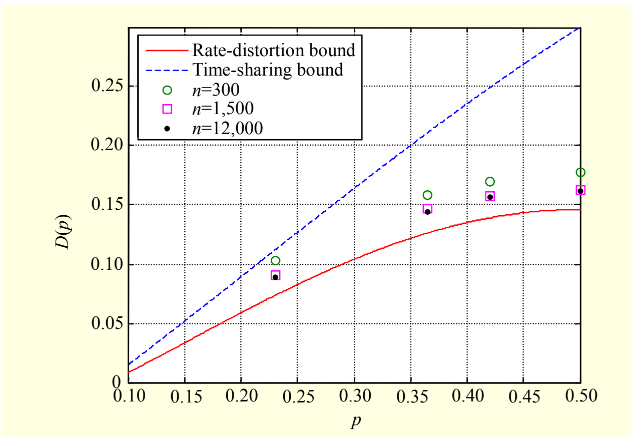

Fig. 2. Distortion curve of our method with R=0.4, q=16, and n=300, 1,500, and 12,000. For comparison, curves of theoretical rate-distortion and time-sharing bounds for same rate are also plotted.

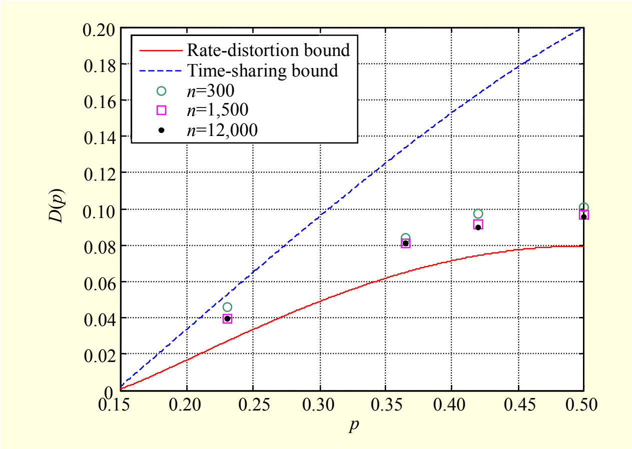

Fig. 3. Distortion curve of our method with R=0.6, q=8, and n=300, 1,500, and 12,000. For comparison, curves of theoretical rate-distortion and time-sharing bounds for same rate are also plotted.

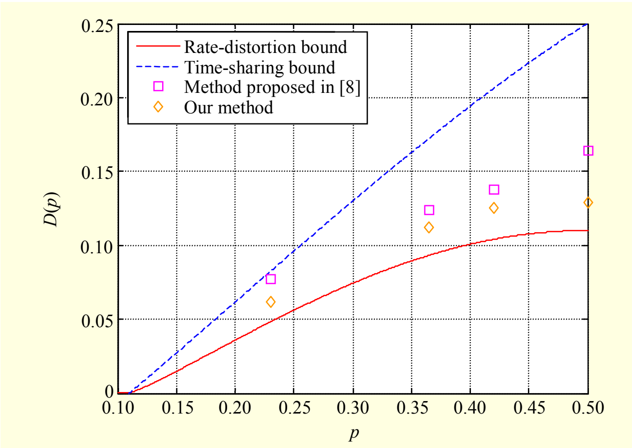

Fig. 4. Performance comparison of our method and method proposed in [8] with R=0.5 and n=1,000. Curves of theoretical rate-distortion and time-sharing bounds for same rate are also plotted.

a better rate-distortion performance than that in [8], which means the cyclic GF(q) LDPC code with larger q has better error rate results. The dual GF(q) LDGM code with check degree d_c=2 is likely to have better rate-distortion performance with larger q.

V. Conclusion

In this letter, we presented a method to compress a non-uniform binary source. The method, based on GQ-LDGM codes, permits generation of codewords with general distribution, and hence is suitable to directly compress sources with non-uniform distribution. Furthermore, the low-complexity RBP is employed to perform the compression. The performance of our method is shown to be quite close to the theoretic rate-distortion bound by simulation validations.

References

[1] A. Braunstein, M. Mézard, and R. Zecchina, "Survey Propagation: An Algorithm for Satisfiability," *Random Structures and Algorithms*, vol. 27, no. 2, Sept. 2005, pp. 201-226.

[2] E. Martinian and J. Yedidia, "Iterative Quantization Using Codes on Graphs," *Proc. Allerton Conf. Commun., Control, Computing*, Monticello, IL, USA, 2003, pp. 110-122.

[3] T. Murayama, "Thouless-Anderson-Palmer Approach for Lossy Compression," *J. Phys. Rev. E: Stat. Nonlin. Soft Matter Phys.*, vol. 69, 2004, p. R035105.

[4] M. J. Wainwright, E. Maneva, and E. Martinian, "Lossy Source Compression Using Low-Density Generator Matrix Codes: Analysis and Algorithms," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, June 2010, pp. 1351-1368.

[5] S. Ciliberti, M. Mézard, and R. Zecchina, "Lossy Data Compression with Random Gates," *Phys. Rev. Lett.*, vol. 95, 2005, p. 038701.

[6] T. Filler and J. Fridrich, "Binary Quantization Using Belief Propagation with Decimation over Factor Graphs of LDGM Codes," *Proc. Allerton Conf. Commun., Control, Computing*, Monticello, IL, USA, 2007, pp. 495-501.

[7] P.A. Regalia, "A Modified Belief Propagation Algorithm for Code Word Quantization," *IEEE Trans. Commun.*, vol. 57, no. 12, Dec. 2009, pp. 3513-3517.

[8] L. Cappellari, "Lossy Source Compression of Non-Uniform Binary Source Using GQ-LDGM Codes," *ITW*, Dublin, 2010, available: arXiv:1004.2628.

[9] Z. Sun et al., "Lossy Source Compression for Non-Uniform Sources Using LDGM Codes," *24th Queen's Biennial Symp. Commun.*, Kingston, Ontario, Canada, June 2008.

[10] A. Braunstein et al., "Encoding for the Blackwell Channel with Reinforced Belief Propagation," *Proc. IEEE Int. Symp. Inform. Theory*, Nice, 2007, pp. 1891-1895.

[11] X.-Y. Hu, *Low-Density Low-Complexity Error-Correcting Codes on Sparse Graphs*, Doctoral dissertation, Swiss Federal Institute of Technology Lausanne (EPFL), 2002.

[12] A. Braunstein, F. Kayhan, and R. Zecchina, "Efficient LDPC Codes over GF(q) for Lossy Data Compression," *Proc. IEEE Int. Symp. Inform. Theory*, Seoul, Korea, 2009, pp. 1978-1982.

[13] X.-Y. Hu, E. Eleftheriou, and D.M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, Jan. 2005, pp. 386-398.