Binary quantization using Belief Propagation with decimation over factor graphs of LDGM codes

Tomáš Filler, Jessica Fridrich

***Abstract*—We propose a new algorithm for binary quantization based on the Belief Propagation algorithm with decimation over factor graphs of Low Density Generator Matrix (LDGM) codes. This algorithm, which we call Bias Propagation (BiP), can be considered as a special case of the Survey Propagation algorithm proposed for binary quantization by Wainwright et al. [1]. It achieves the same near-optimal rate-distortion performance with a substantially simpler framework and 10–100 times faster implementation. An important advantage of our reformulation using Belief Propagation is that the algorithm can be analyzed using standard tools previously developed for Density Evolution. We derive a necessary condition that the node degree distributions of the associated factor graphs must satisfy to obtain good BiP quantizers. Finally, we give examples of suitably irregular LDGM codes that fulfill the necessary condition and show their performance.**

I. Introduction

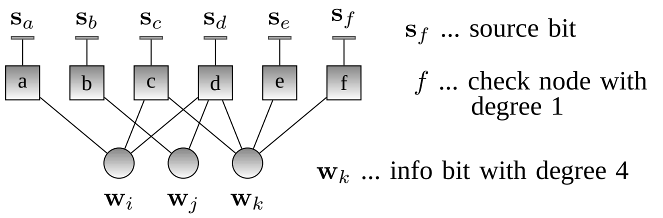
BINARY quantization is an important problem for various fields (information hiding, lossy compression). The recent work of Wainwright et al. [1] shows that Low Density Generator Matrix (LDGM) codes combined with Survey Propagation based message-passing algorithm can be used to achieve near-optimal binary quantization in practice.

We propose a much simpler approach based on pure Belief Propagation for quantizing random Bernoulli source with $p = \frac{1}{2}$. This algorithm, which we call Bias Propagation (BiP), achieves the same near-optimal rate-distortion performance and is amenable to theoretical analysis similar to density evolution.

In particular, we are quantizing a random n-bit source sequence $\mathbf{s}$ to the nearest codeword $\mathbf{c_s}$ from an LDGM code $\mathcal{C}$ with rate $R = \frac{m}{n}$. The rate-distortion function for our case is in the form $R(D) = 1 - H(D)$ for $D \in [0, 0.5]$ and 0 otherwise, where H is the binary entropy and $D = \mathbb{E}\big[\frac{1}{n}\sum_{i=1}^{n}|\mathbf{s}_i - (\mathbf{c_s})_i|\big]$ is the average distortion per bit.

II. LDGM code representation

The generator matrix $\mathbf{G} \in \{0,1\}^{n\times m}$ is obtained randomly using degree distributions from edge perspective (ρ,λ), $\rho(x) = \sum_{i=1}^{d_R}\rho_i x^{i-1}$ and $\lambda(x) = \sum_{i=1}^{d_L}\lambda_i x^{i-1}$, where ρ_i and λ_i denote the portion of all edges connected to check nodes and info bits with degree i, respectively.



We use C and V to denote the set of all check nodes and the set of all info bits in the factor graph, respectively. Finally, we define the sets $C(i) = \{a \in C \,|\, a \text{ is connected to } i\}$, $V(a) = \{i \in V \,|\, i \text{ is connected to } a\}$, and $\overline{V}(a) = V(a) \cup \{\mathbf{s}_a\}$.

III. Bitwise MAP estimation over weighted codewords

Finding $\mathbf{c_s}$ ($\mathbf{c_s} = \mathbf{G}\mathbf{w_s}$, $\mathbf{w_s} \in \{0,1\}^m$) is equivalent to finding MAP assignment in the following probability distribution

$$P(\mathbf{w}|\mathbf{s};\gamma) = \frac{1}{Z}e^{-2\gamma d_H(\mathbf{G}\mathbf{w},\mathbf{s})},$$

where Z is a normalization constant and $d_H(\mathbf{G}\mathbf{w},\mathbf{s})$ is the Hamming distance between $\mathbf{G}\mathbf{w}$ and $\mathbf{s}$. We find $\mathbf{w_s}$ using bitwise MAP, where $\big(\mathbf{w_s}\big)_i = \arg\max_{\mathbf{w}_i\in\{0,1\}} P(\mathbf{w}_i|\mathbf{s};\gamma)$. The marginal probabilities $P(\mathbf{w}_i|\mathbf{s};\gamma)$ can be calculated efficiently using Belief Propagation algorithm.

IV. Bias Propagation (BiP) algorithm

The BiP is an iterative message-passing algorithm that performs bitwise MAP estimation. One round of the BiP algorithm consists of `max_iter` message-passing iterations followed by a decimation step (selected info bits are fixed and removed from graph). In ℓ-th iteration, bias messages $B_{i\to a}^{(\ell)}$ and constant source messages $B_{\mathbf{s}_a\to a}^{(\ell)}$ are sent from info bits and source bits to connected check nodes. Check nodes are sending satisfaction messages $S_{a\to i}^{(\ell)}$ to their connected info bits. Final bias B_i expresses the difference of marginal probabilities $P(\mathbf{w}_i = 0|\mathbf{s};\gamma) - P(\mathbf{w}_i = 1|\mathbf{s};\gamma)$. We use damping to avoid short cycles in the factor graph. See Figure 1 for the complete pseudo-code and update rules of the BiP algorithm. More information will be available in [2].

V. Convergence analysis

We say that BiP message-updates converge in r-th round if $\max_{i\in V}|B_{i\to a}^{(\hat{\ell})}| > \mathtt{t}$, where $\mathtt{t}$ is a constant parameter and $\hat{\ell} = \mathtt{max\_iter}$ (see Figure 1). To describe the set of suitably irregular degree distributions for the BiP algorithm, we derived the following necessary condition.

Theorem 1 (Convergence condition): If the BiP algorithm converge in its 1-st round, then the degree distributions have to fulfil the following necessary condition:

$$\mathfrak{C}(b_{\mathbf{s}},\rho,\lambda) = \mathfrak{D}_2(b_{\mathbf{s}})\rho'(0)\frac{\lambda'\big(1-\mathfrak{D}_2(b_{\mathbf{s}})\rho(0)\big)}{\lambda\big(1-\mathfrak{D}_2(b_{\mathbf{s}})\rho(0)\big)} \geq 1, \qquad (1)$$

where $\mathfrak{D}_2(b_{\mathbf{s}}) = \tanh^2(\gamma)$ is the variance of source messages $B_{\mathbf{s}_a\to a}^{(\ell)}$ (variance of their D-density).

This convergence condition has to be fulfiled in each round. To be able to calculate the value $\mathfrak{C}(b_{\mathbf{s}},\rho,\lambda)$, we have to know the degree distribution in each round. The decimation process modifies only the check node distribution. After removing $(1-\tau)$ fraction of all info bits, we can write the i-th term of the degree distribution $\rho^{(1-\tau)}$ as $\rho_i^{(1-\tau)} = \frac{r_i(\tau)}{\tau}$. The function $r_i(\tau)$ is given by the following system of equations

$$\frac{d}{d\tau}r_i(\tau) = \big(r_i(\tau) - r_{i+1}(\tau)\big)\frac{i}{\tau}, \qquad i < d_R \qquad (2)$$

$$\frac{d}{d\tau}r_{d_R}(\tau) = r_{d_R}(\tau)\frac{d_R}{\tau}, \qquad (3)$$

where $r_i(1) = \rho_i^{(0)} = \rho_i$ is the i-th term of the initial $\rho(x)$.

Bias Propagation Algorithm (BiP)

(a) pseudo-code

```
procedure w = BiP(G, s)
  G.B_saa = calc_src_msg(s, gamma) /* (BiP-1) */
  G.S_ai = calc_ai(1, G.B_saa)      /* (BiP-4) */
  while not all_bits_fixed(w)
    bias = BiP_iter(G, s)
    bias = sort(bias)
    if max(|bias|)>t
      num = min(num_max, num_of_bits(|bias|>t))
    else
      num = num_min
    [G,s,w] = dec_most_biased_bits(G,s,w,num)
  end
end
```

```
procedure bias = BiP_iter(G, s)
  G.B_saa = calc_src_msg(s, gamma)            /* (BiP-1) */
  while iter<max_iter
    G.B_ia_old = G.B_ia
    G.B_ia = calc_ia(G.S_ai)                  /* (BiP-2) */
    if iter>start_damp then
      G.B_ia = damping(G.B_ia, G.B_ia_old)    /* (BiP-3) */
    end
    G.S_ai = calc_ai(G.B_ia, G.B_saa))        /* (BiP-4) */
    iter = iter+1
  end
  bias = calc_bias(G.S_ai)                    /* (BiP-5) */
end
```

(b) message-passing update rules

Source message initialization:

$$B^{(\ell)}_{\mathbf{s}_a\to a} = (-1)^{\mathbf{s}_a}\tanh(\gamma) \qquad \text{(BiP-1)}$$

Bias update rule:

$$\tilde{B}_{i\to a} = \frac{\prod_{b\in C(i)\setminus\{a\}}\left(1+S^{(\ell-1)}_{b\to i}\right) - \prod_{b\in C(i)\setminus\{a\}}\left(1-S^{(\ell-1)}_{b\to i}\right)}{\prod_{b\in C(i)\setminus\{a\}}\left(1+S^{(\ell-1)}_{b\to i}\right) + \prod_{b\in C(i)\setminus\{a\}}\left(1-S^{(\ell-1)}_{b\to i}\right)}. \qquad \text{(BiP-2)}$$

Equation for damping update in ℓ-th iteration:

$$B^{(\ell)}_{i\to a} = \frac{\sqrt{(1+\tilde{B}_{i\to a})(1+B^{(\ell-1)}_{i\to a})} - \sqrt{(1-\tilde{B}_{i\to a})(1-B^{(\ell-1)}_{i\to a})}}{\sqrt{(1+\tilde{B}_{i\to a})(1+B^{(\ell-1)}_{i\to a})} + \sqrt{(1-\tilde{B}_{i\to a})(1-B^{(\ell-1)}_{i\to a})}}. \qquad \text{(BiP-3)}$$

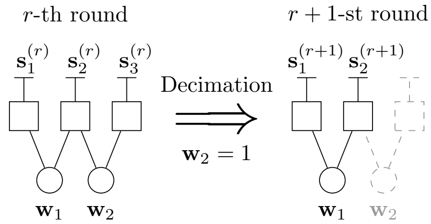
Equation for calculating final bias B_i in $\hat{\ell}$-th iteration:

$$B_i = \frac{\prod_{b\in C(i)}\left(1+S^{(\hat{\ell})}_{b\to i}\right) - \prod_{b\in C(i)}\left(1-S^{(\hat{\ell})}_{b\to i}\right)}{\prod_{b\in C(i)}\left(1+S^{(\ell)}_{b\to i}\right) + \prod_{b\in C(i)}\left(1-S^{(\ell)}_{b\to i}\right)}. \qquad \text{(BiP-5)}$$

Satisfaction update rule:

$$S^{(\ell)}_{a\to i} = \prod_{j\in\overline{V}(a)\setminus\{i\}} B^{(\ell)}_{j\to a} \qquad \text{(BiP-4)}$$

Graph decimation:



$$\mathbf{s}_1^{(r+1)} = \mathbf{s}_1^{(r)} \qquad \mathbf{s}_2^{(r+1)} = XOR(\mathbf{s}_2^{(r)}, \mathbf{w}_2)$$

Constant parameters:
`num_max`... maximum info bits to decimate
`num_min`... minimum info bits to decimate
`t`... decimation threshold
`gamma` ... check node satisfaction strength
`max_iter` ... max. # of iteration in round
`start_damp` ... # iterations without damping

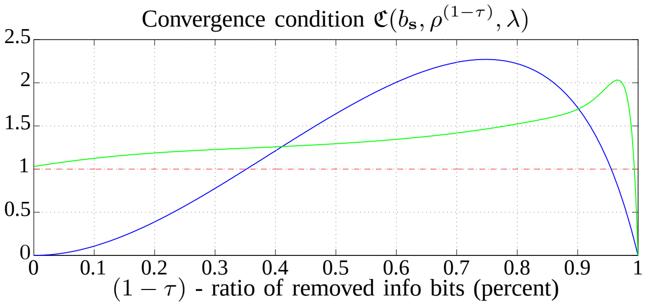
Fig. 1. Summary of Bias Propagation algorithm.

VI. Results: Convergence Analysis

The BiP algorithm does not converge well with regular codes. It converges in every round with the following "suitably irregular" degree distribution with rate $R = 0.5$:

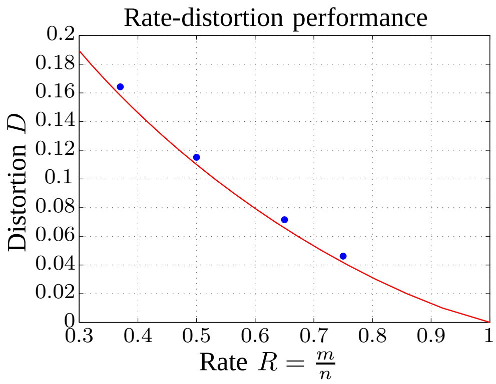
$$\rho(x) = 0.1787x + 0.1762x^2 + 0.1028x^5 + 0.1147x^6 + \\ + 0.0122x^{12} + 0.0479x^{13} + 0.1159x^{14} + 0.2516x^{39}$$
$$\lambda(x) = 0.9988x^9 + 0.0012x^{10}.$$

The following graph shows the convergence condition evaluated for $(4, 8)$ regular code ($\gamma = 1$) and for code based on the irregular degree distribution above ($\gamma = 1.1$).



VII. Results: Binary Quantization using BiP

Each • is an average over 100 random trials. Code length $n = 10^4$. Average throughput (number of source bits quantized per second) was 11kb/sec.



References

[1] M. J. Wainwright and E. Maneva. Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes. In *Proceedings of the International Symposium on Information Theory, Adelaide, Australia*, September 2005.

[2] T. Filler and J. Fridrich. Binary quantization using belief propagation with decimation over factor graphs of LDGM codes. In preparation.