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

T. Filler J. Fridrich

Department of ECE, Binghamton University, Binghamton, NY

Allerton Conference 2007

Binary quantization problem

$\mathcal{C}$ ... binary linear $[n, m]$ code

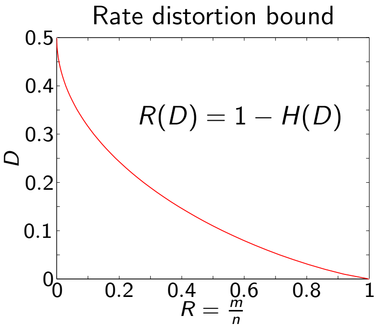
$\mathbf{s} \in \{0,1\}^n$ ... source sequence i.i.d. $P(\mathbf{s}_i = 0) = P(\mathbf{s}_i = 1) = \frac{1}{2}$

$$\mathbf{c}_\mathbf{s} = \arg\min_{\mathbf{c} \in \mathcal{C}} d(\mathbf{s}, \mathbf{c}) = \arg\min_{\mathbf{c} \in \mathcal{C}} \frac{1}{n} \sum_{i=1}^{n} |\mathbf{s}_i - \mathbf{c}_i|$$

$d(\mathbf{s}, \mathbf{c})$ ... relative Hamming distance between $\mathbf{s}$ and $\mathbf{c}$

Distortion: $D = E[d(\mathbf{s}, \mathbf{c}_\mathbf{s})]$
Rate: $R = \frac{m}{n}$

We assume that $\mathcal{C}$ is Low Density Generator Matrix (LDGM) code.

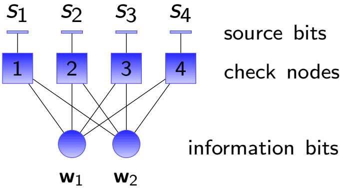


LDGM code representation

Graphical representation

Each codeword $\mathbf{c} \in \mathcal{C}$ can be obtained as

$$\mathbf{c} = \mathbf{G}\mathbf{w}, \ \mathbf{w} \in \{0,1\}^m.$$



$C(i) = \{$checks connected to infobit $i\}$
$V(a) = \{$infobits connected to check $a\}$

$$\overline{V}(a) = V(a) \cup \{\mathbf{s}_a\}$$

Degree distribution

Generator matrix $\mathbf{G} \in \{0,1\}^{n \times m}$ is obtained randomly according to given degree distribution $(\rho, \lambda) = (\sum_{i=1}^{dR} \rho_i x^{i-1}, \sum_{i=1}^{dL} \lambda_i x^{i-1})$.

ρ_i ... portion of all edges connected to check nodes with degree i
λ_i ... portion of all edges connected to infobits with degree i

Recent results

List of recent results sorted by distortion performance:

- S. Ciliberti, M. Mezard and R. Zecchina: nonlinear nodes.
  [Lossy data compression with random gates, Physical Review Letters, 2005].
- T. Murayama: regular LDGM codes.
  [Thouless-Anderson-Palmer approach for lossy compression, Physical Review E, 2004].
- M. J. Wainwright and E. Maneva: near-optimal performance.
  [Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes, IEEE ISIT, 2005].

Recent results

List of recent results sorted by distortion performance:

- S. Ciliberti, M. Mezard and R. Zecchina: nonlinear nodes.
  [Lossy data compression with random gates, Physical Review Letters, 2005].
- T. Murayama: regular LDGM codes.
  [Thouless-Anderson-Palmer approach for lossy compression, Physical Review E, 2004].
- M. J. Wainwright and E. Maneva: near-optimal performance.
  [Lossy source encoding via message-passing and decimation over generalized codewords of LDGM codes, IEEE ISIT, 2005].

Does it mean that we need Survey Propagation (SP) algorithm to achieve near-optimal distortion?

Probability distribution over LDGM codewords

For a given constant γ and source sequence $\mathbf{s}$, we define

$$P(\mathbf{w}|\mathbf{s};\gamma) = \frac{1}{Z}\exp\left[-2\gamma\sum_{i=1}^{n}|(\mathbf{G}\mathbf{w})_i - \mathbf{s}_i|\right],$$

where Z is a normalization constant.

- Finding closest codeword $\mathbf{c}_{\mathbf{s}}$ is equivalent to MAP estimation.
- Perform bitwise MAP estimation in rounds, set

$$(\mathbf{w}_{\mathbf{s}})_i = \arg\max_{\mathbf{w}_i\in\{0,1\}} P(\mathbf{w}_i|\mathbf{s};\gamma).$$

Probability distribution over LDGM codewords

For a given constant γ and source sequence $\mathbf{s}$, we define

$$P(\mathbf{w}|\mathbf{s};\gamma) = \frac{1}{Z}\exp\left[-2\gamma\sum_{i=1}^{n}|(\mathbf{Gw})_i - \mathbf{s}_i|\right],$$

where Z is a normalization constant.

- Finding closest codeword $\mathbf{c}_\mathbf{s}$ is equivalent to MAP estimation.
- Perform bitwise MAP estimation in rounds, set

$$(\mathbf{w}_\mathbf{s})_i = \arg\max_{\mathbf{w}_i\in\{0,1\}} P(\mathbf{w}_i|\mathbf{s};\gamma).$$

  In r-th round we need to:

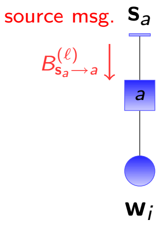
  1. Calculate bias:
     $B_i = P(\mathbf{w}_i = 0) - P(\mathbf{w}_i = 1)$ for all infobits i.
  2. Decimate the factor graph:
     Fix 1 information bit with maximal bias magnitude $|B_i|$.

Bias Propagation algorithm (BiP)

Calculating marginal probabilities

Sum-product algorithm can be used for approximating marginal probabilities.

1. Source message initialization
2. Message-passing iterations
3. Calculate final bias after $\hat{\ell}$ iterations



Source messages are constant within one round.

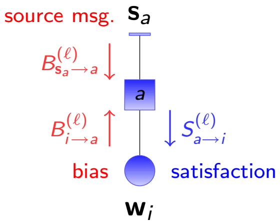
$$B^{(\ell)}_{\mathbf{s}_a \to a} = (-1)^{\mathbf{s}_a} \tanh(\gamma) \qquad \text{(BiP-1)}$$

Bias Propagation algorithm (BiP)

Calculating marginal probabilities

Sum-product algorithm can be used for approximating marginal probabilities.

1. Source message initialization
2. Message-passing iterations
3. Calculate final bias after $\hat{\ell}$ iterations



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

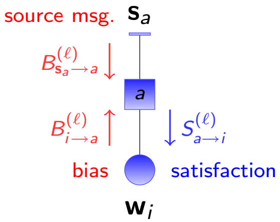
$$S^{(\ell)}_{a \to i} = \prod_{j \in \overline{V}(a) \setminus \{i\}} B^{(\ell)}_{j \to a} \qquad \text{(BiP-4)}$$

Bias Propagation algorithm (BiP)

Calculating marginal probabilities

Sum-product algorithm can be used for approximating marginal probabilities.

1. Source message initialization
2. Message-passing iterations
3. Calculate final bias after $\hat{\ell}$ iterations

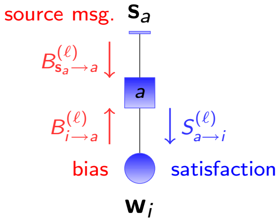


$$B_i = \frac{\prod_{b \in C(i)}\left(1 + S^{(\hat{\ell})}_{b \to i}\right) - \prod_{b \in C(i)}\left(1 - S^{(\hat{\ell})}_{b \to i}\right)}{\prod_{b \in C(i)}\left(1 + S^{(\ell)}_{b \to i}\right) + \prod_{b \in C(i)}\left(1 - S^{(\ell)}_{b \to i}\right)} \qquad \text{(BiP-5)}$$

Bias Propagation algorithm (BiP)

Dealing with cycles in factor graph

The messages tends to oscilate due to the presence of cycles in the factor graph. We suppress these oscilations using damping procedure.

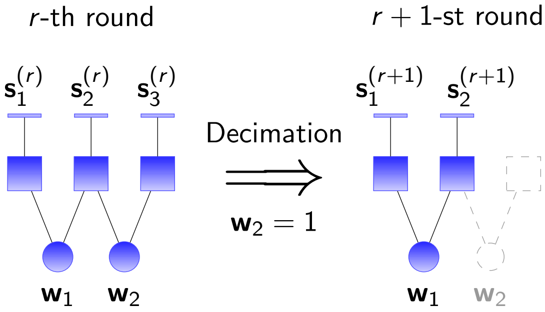


$$\tilde{B}_{i\to a} = \frac{\prod_{b\in C(i)\setminus\{a\}}\left(1+S^{(\ell-1)}_{b\to i}\right) - \prod_{b\in C(i)\setminus\{a\}}\left(1-S^{(\ell-1)}_{b\to i}\right)}{\prod_{b\in C(i)\setminus\{a\}}\left(1+S^{(\ell-1)}_{b\to i}\right) + \prod_{b\in C(i)\setminus\{a\}}\left(1-S^{(\ell-1)}_{b\to i}\right)}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

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

Bias Propagation algorithm (BiP)

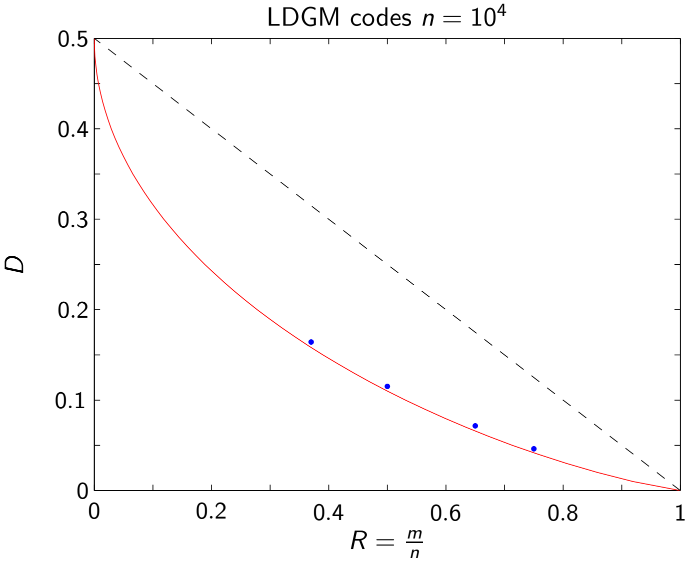
Decimation step



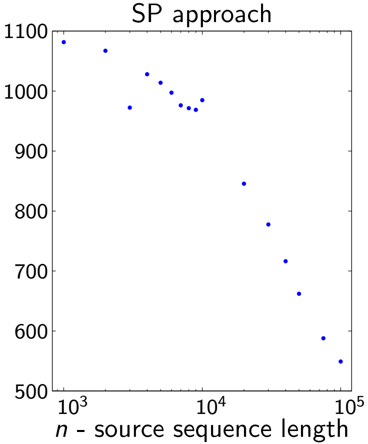
$$\mathbf{s}_1^{(r+1)} = \mathbf{s}_1^{(r)} \qquad \mathbf{s}_2^{(r+1)} = XOR\left(\mathbf{s}_2^{(r)}, \mathbf{w}_2\right)$$

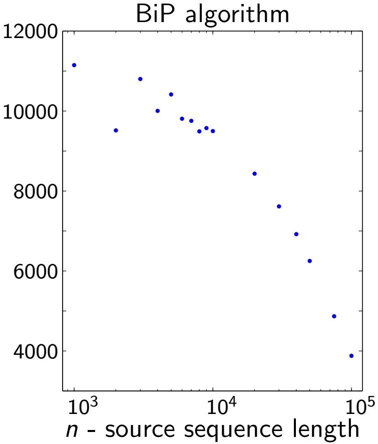
The decimation step preserves the values of messages associated with each edge.

Rate-distortion performance



BiP algorithm vs. Survey Propagation approach ($R = 1/2$)

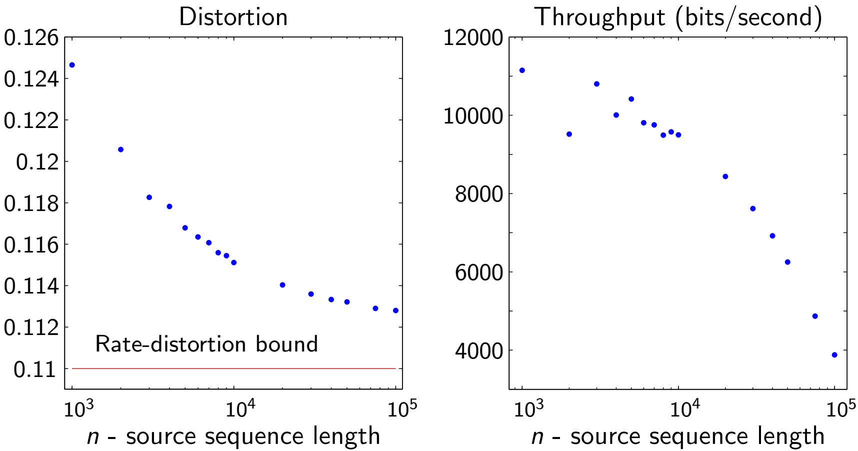




Throughput = number of source bits quantized per second.

Both algorithms achieve the same distortion.

Throughput performance ($R = 1/2$)



Throughput = number of source bits quantized per second.

Summary

Bias Propagation (BiP) algorithm

- new algorithm for binary quantization based on BP
- same near-optimal rate-distortion performance as SP
  We do not need to use Survey Propagation approach.
- much simpler framework
- $10\times$ faster implementation
- more amenable to analysis