Belief propagation decoding of quantum channels by passing quantum messages

Joseph M. Renes

arXiv:1607.04833

QIP 2017
To do research in quantum information theory, pick a favorite text on classical information theory, open to a chapter, and translate the contents into quantum-mechanical language.

Spatially Coupled Ensembles Universally Achieve Capacity Under Belief Propagation
Shrinivas Kudekar, Tom Richardson, Fellow, IEEE, and Rüdiger L. Urbanke

Abstract—We investigate spatially coupled code ensembles. For transmission over the binary erasure channel, it was recently shown that spatial coupling increases the belief propagation threshold of the ensemble to essentially the maximum \textit{a priori} threshold of the underlying component ensemble. This explains why convolutional LDPC ensembles, originally introduced by Felstern and Zigangirov, perform so well over this channel. We show that the equivalent result holds true for transmission over general binary-input memoryless output-symmetric channels. More precisely, given a desired error probability and a gap to capacity, we can construct a spatially coupled ensemble that builds these constraints universally on this class of channels under belief-propagation decoding. In fact, most codes in this ensemble have this property. The quantifier universal refers to the single ensemble/code that is good for all channels but we assume that the channel is known at the receiver. The key technical result is a proof that, under belief-propagation decoding, spatially coupled...
Belief propagation: *message passing* algorithm for performing *inference* in a *graphical model*

decoder infers channel input from output; code can be described by a graphical model

eyear precursor: Bethe-Peierls approximation in statistical physics

many applications in statistics and machine learning besides coding:

inference, optimization, constraint satisfaction
Quantum belief propagation decoding?

- Can use classical algorithm for usual stabilizer decoding
- Optimal for: \( W \) \( \rightarrow \) \( M \)
- But not, e.g. amplitude damping \( \rightarrow \)

Let’s consider CQ channels for simplicity

Need to infer channel input from quantum output
(not trying to compute marginals of quantum states)
Results

- BP decoder for pure state channel, tree codes

- Also works for polar codes:
  Efficient, capacity-achieving decoder for BPSK over lossy Bosonic channel

- And for quantum communication, part of conjugate basis decoder:
  Efficient, capacity-achieving decoder for amplitude damping

arXiv:1607.04833
1. Coding setup
2. Factor graphs
3. Classical BP
4. Quantum BP
5. Many open questions
Coding setup

Shannon scenario: stochastic iid noise. not adversarial; fault-free decoding

Linear code

bitwise decoding of random codeword

classically, marginalize joint distribution

\[ P_{X_1^ny_1^n = y_1^n} \rightarrow P_{X_1y_1^n = y_1^n} \]

quantumly, perform Helstrom measurement for each bit

\[ \rho_{X_1^nB_1^n} \rightarrow \rho_{X_1B_1^n} \]
Factor graphs

factorizeable joint probability

\[ P(x_1, x_2, x_3, x_4) = \frac{1}{Z} f(x_1, x_2) g(x_2) h(x_2, x_3, x_4) \]

easier to compute marginals:

\[ P(x_1) = \sum_{x_2, x_3, x_4} P(x_1, x_2, x_3, x_4) = \frac{1}{Z} \sum_{x_2} f(x_1, x_2) g(x_2) \left( \sum_{x_3, x_4} h(x_2, x_3, x_4) \right) \]

channel input & output distribution:

\[ P_{X^nY^n}(x_1^n, y_1^n) = \frac{1}{|C|} \mathbb{1}[x_1^n \in C] \prod_{j=1}^{n} W(y_j|x_j). \]
Classical BP decoding

symmetric binary input channel: only care about likelihood ratio

\[ \ell(y) = \frac{W(y|1)}{W(y|0)} \]

\( x_i \rightarrow y_1^n \) is a channel; want \( \ell(y_1^n) \)

assume tree factor graph

for each node, associate a channel to all its leaves

BP recursively computes \( \ell(y_1^n) \)

starting from the leaves
for two channels

algorithm can be seen as successively simplifying the channel from the root to the leaves by rules for combining messages can then be interpreted as rules for combining channels, and the ratio is a su

effective channel from that node to its descendants. This is sensible as the likelihood coefficient statistic for estimating the (binary) input from the channel output. The our quantum generalization. At every step the message can be interpreted as the log-likelihood correct value. This is borne out in practice for turbo codes and LDPC codes.

nevertheless hope that the result is a good approximation and that the decoder outputs the edges. For graphs that contain loops, the algorithm is not guaranteed to converge, but one can

inward, combining all relevant information as it goes. Simplifying the general BP rules (see \[ \text{FIG. 1: Factor graph for the joint probability distribution of a four-bit code with two parity checks} \]

BP recursively computes \( \ell(y) = \frac{W(y|1)}{W(y|0)} \) starting from the leaves

\( x_i \rightarrow y_1^n \) is a channel; want \( \ell(y_1^n) \)

assume tree factor graph

for each node, associate a channel to all its leaves

BP recursively computes \( \ell(y_1^n) \) starting from the leaves
Classical BP decoding

\[
\ell(y) = \frac{W(y|1)}{W(y|0)}
\]

\(x_i \rightarrow y_1^n\) is a channel; want \(\ell(y_1^n)\)

assume tree factor graph

for each node, associate a channel to all its leaves

BP recursively computes \(\ell(y_1^n)\) starting from the leaves
Classical BP decoding

symmetric binary input channel: only care about likelihood ratio

\[ \ell(y) = \frac{W(y|1)}{W(y|0)} \]

\( x_i \rightarrow y_1^n \) is a channel; want \( \ell(y_1^n) \)

assume tree factor graph

for each node, associate a channel to all its leaves

BP recursively computes \( \ell(y_1^n) \) starting from the leaves
Classical BP decoding

BP rules specify two kinds of channel convolution:

Variable node:

\[
W \oplus W' = \begin{cases} 
W_0 \otimes W_0' & \\
W_1 \otimes W_1' & 
\end{cases}
\]

Check node:

\[
W \otimes W' = \frac{1}{2}(W_0 \otimes W_0' + W_1 \otimes W_1')
\]

- BP finds exact marginals on trees
- Can run algorithm for all codeword bits concurrently
- Also works on loopy LDPC factor graphs
Quantum BP decoding

Now the outputs are quantum, no likelihood function

Want to perform Helstrom measurement

Can we combine Helstrom measurements along the branches?

Can we “repackage” the outputs as with likelihood?

Try the simplest case: pure state outputs

\[
\begin{align*}
&0 &\quad \text{W} &\quad |+\theta\rangle \\
&1 &\quad &\quad |-\theta\rangle \\
\end{align*}
\]

\[\langle +\theta | -\theta \rangle = \cos \theta\]

optimal measurement is \(\sigma_x\)
Quantum BP decoding

\[ \langle +\theta| -\theta \rangle = \cos \theta \]

\[ W \otimes W' \]

\[ \begin{pmatrix} W_0 \otimes W_0' \\ W_1 \otimes W_1' \end{pmatrix} \]

\[ \frac{1}{2} (W_0 \otimes W_0' + W_1 \otimes W_1') \]

\[ \frac{1}{2} (W_0 \otimes W_1' + W_1 \otimes W_0') \]

\[ \otimes \text{ convolution yields pure states:} \]

\[ |+\theta\rangle \otimes |+\theta'\rangle \quad |-\theta\rangle \otimes |-\theta'\rangle \quad \rightarrow \quad |\pm\theta^\otimes\rangle \quad \cos \theta^\otimes = \cos \theta \cos \theta' \]

\[ \bigstar \text{ convolution gives a \textit{heralded} mixture of pure states!} \]

\[ \text{unitary } U_{\otimes} \text{ gives} \quad \sum_{j \in \{0,1\}} p_j \ |j\rangle \langle j| \otimes |\pm \theta_j^\otimes\rangle \otimes |\pm \theta_j^\otimes\rangle \]

\[ p_0 = \frac{1}{2} (1 + \cos \theta \cos \theta') \quad \cos \theta_0^\otimes = \frac{\cos \theta + \cos \theta'}{1 + \cos \theta \cos \theta'} \quad \cos \theta_1^\otimes = \frac{\cos \theta - \cos \theta'}{1 - \cos \theta \cos \theta'} \]
Quantum BP decoding

- pass qubits and some classical bits: “sufficient statistic”
- decode all codeword bits sequentially, unwinding each time
- $O(n^2)$ implementation of all Helstrom measurements

\[
\sum_{j \in \{0,1\}} p_j |\pm \theta_j^\oplus \rangle \otimes |\pm \theta_j^\oplus \rangle \otimes |j\rangle \langle j|
\]
Quantum BP & polar codes

Variable and check convolutions = “better” and “worse” synth channels

Polar decoder has tree structure  (for message not codeword bits, tho)

Quantum polar decoder uses polar decoder for classical amplitude and phase info

amplitude = classical Z channel, phase = pure state channel

O(n^2) decoder for capacity-achieving quantum polar code for amplitude damping
Open

Questions
Open

other channels? e.g. classical coding for amplitude damping?

loops? spatially-coupled LDPC codes?

use density matrix BP? need sufficient statistics, local tree to global

relation to tensor networks?

junction-tree algorithm?

Viterbi decoder (blockwise decoder)?

fully quantum version?

other tasks besides decoding?

Questions