The Belief Propagation Decoding of LDPC Codes

Chaonian Guo

CIS Lab, Coding Group

Jan 9, 2009

OUTLINE

1. ABOUT LDPC CODES
   - Research aspects
   - Basic Conceptions
   - Coding & Decoding Process
2. THE BP DECODING OF LDPC CODES
   - Posteriori Probability
   - Belief Propagation
   - LLR BP
3. SOME IMPROVED RESULTS
   - Fluctuations
   - Adaptive Erasure
   - Simulation Results
4. THE END: LITERATURE REVIEW

Outline

1. About LDPC Codes
   - Research aspects
   - Basic Conceptions
   - Coding & Decoding Process
2. The BP Decoding of LDPC Codes
   - Posteriori Probability
   - Belief Propagation
   - LLR BP
3. Some Improved Results
   - Fluctuations
   - Adaptive Erasure
   - Simulation Results
4. The End: Literature Review

Outline

Research aspects

- Coding: H.
- Decoding: simplification&accuracy of decoding.
- Density Evolution: improvements.
- Design of Irregular LDPC Codes: degree distribution.
- Distance&Performance: analysis.
- Implementation&Application: communication.

Outline

LDPC Codes

A binary Low-Density Parity-Check code, specified by a parity check matrix $H_{(N-K)\times N}$ in *GF(2)*: the 0's are far more than the 1's.

- N: the linear block length of a codeword c.
- K: the length of the source s.
- M: the number of check bits ($M = N - K$).
- R: code rate = $\frac{K}{N}$.
- $G_{K\times N}$: generator matrix specified by $G^T H = 0$.

Example 1

The parity check matrix of a trivial LDPC code may be

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

which means:

$$\begin{cases} c_1 + c_2 + c_4 = 0 \\ c_2 + c_3 + c_6 = 0 \\ c_1 + c_3 + c_5 = 0 \end{cases}$$

Regular & Irregular LDPC Codes

- (d_v, d_c)-regular codes
  - All the column weights are d_v.
  - All the row weights are d_c.
- $(\lambda(x), \rho(x))$-irregular codes
  - $\lambda(x) = \sum \lambda_i x^{i-1}$, $\rho(x) = \sum \rho_i x^{i-1}$.
  - λ_i: the fraction of columns of weight i in H.
  - ρ_i: the fraction of rows of weight i in H.

Example 2

- (2,4)-regular LDPC code:

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

- $(\lambda(x), \rho(x))$-irregular LDPC code, $\lambda(x) = 0.4x + 0.6x^2$, $\rho(x) = 0.2x^2 + 0.8x^3$:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

OUTLINE

1. **ABOUT LDPC CODES**
   - Research aspects
   - Basic Conceptions
   - Coding & Decoding Process
2. **THE BP DECODING OF LDPC CODES**
   - Posteriori Probability
   - Belief Propagation
   - LLR BP
3. **SOME IMPROVED RESULTS**
   - Fluctuations
   - Adaptive Erasure
   - Simulation Results
4. **THE END: LITERATURE REVIEW**

Coding & Decoding Process

- Source sequence $s = \{s_1, s_2, ..., s_K\}$.
- Code by $s \cdot G \rightarrow$ codeword $c = \{c_1, c_2, ..., c_N\}$.
- Modulate codeword $c \rightarrow x$.
- Transmit x.
- Receive x and demodulate $x \rightarrow y$.
- Decode $y \rightarrow$ codeword $\hat{c}$.

Decoding

Giving y, how to determine $\hat{c}$?

- Hard decision decoding: Bit-Flip.
- Soft decision decoding: Belief Propagation.

Outline

Outline

Posteriori Probability

The *Posteriori Probability* of the codeword c is computed based on the received value y .

The d^{th} bit c_d:

- $\Pr(c_d = 0 | y_d, S)$, $\Pr(c_d = 1 | y_d, S)$.
- S: bit c_d satisfies all the check equations.

Lemma

Consider a sequence of m independent binary digits in which the l^{th} digit is a 1 with probability P_l. Then the probability that an even number of digits are 1 is $\frac{1+\prod_{l=1}^{m}(1-2P_l)}{2}$.

Theorem

Let P_d be the probability that c_d is a 1 conditional on the received digit y_d, and let P_{il} be same probability for the l^{th} bit in the i^{th} check equation. Let the digits be statistically independent of each other. Then

$$\frac{Pr(c_d=0|y_d,S)}{Pr(c_d=1|y_d,S)} = \frac{1-P_d}{P_d} \prod_{i=1}^{d_v} \frac{1+\prod_{l=1}^{d_c-1}(1-2P_{il})}{1-\prod_{l=1}^{d_c-1}(1-2P_{il})}$$

Proof

$$\frac{Pr(c_d=0|y_d,S)}{Pr(c_d=1|y_d,S)}$$
$$=\frac{Pr(c_d=0,y_d,S)/Pr(y_d,S)}{Pr(c_d=1,y_d,S)/Pr(y_d,S)}$$
$$=\frac{Pr(c_d=0,y_d,S)}{Pr(c_d=1,y_d,S)}$$
$$=\frac{Pr(y_d)Pr(c_d=0|y_d)Pr(S|c_d=0,y_d)}{Pr(y_d)Pr(c_d=1|y_d)Pr(S|c_d=1,y_d)}$$
$$=\frac{1-P_d}{P_d}\prod_{i=1}^{d_v}\frac{(1+\prod_{l=1}^{dc-1}(1-2P_{il}))/2}{(1-\prod_{l=1}^{dc-1}(1-2P_{il}))/2}$$
$$=\frac{1-P_d}{P_d}\prod_{i=1}^{d_v}\frac{1+\prod_{l=1}^{dc-1}(1-2P_{il})}{1-\prod_{l=1}^{dc-1}(1-2P_{il})} \qquad \square$$

Outline

Notations

- N(m): $\{n | H_{mn}=1, 1 \leq n \leq N\}$.
- M(n): $\{m | H_{mn}=1, 1 \leq m \leq M\}$.
- $r_{mn}(0)$: the probability of check *m* being satisfied when bit *n* is 0.
- $r_{mn}(1)$: ...
- $q_{mn}(0)$: the probability that bit *n* has the value 0, given the information obtained by the checks other than check *m*.
- $q_{mn}(1)$: ...
- $q_n(0)$: the probability that bit *n* has the value 0, given the information obtained by all the checks.
- $q_n(1)$: ...

Decoding Process(1)

- Initialization:

$$q_{mn}^{(0)}(0) = P_i(0), q_{mn}^{(0)}(1) = P_i(1), t = 1$$

- Updating check node messages:

$$r_{mn}^{(t)}(0) = \frac{1}{2} + \frac{1}{2} \prod_{n' \in N(m) \setminus n} (1 - 2q_{mn'}^{(t-1)}(1))$$

$$r_{mn}^{(t)}(1) = \frac{1}{2} - \frac{1}{2} \prod_{n' \in N(m) \setminus n} (1 - 2q_{mn'}^{(t-1)}(1))$$

DECODING PROCESS(2)

- Updating variable node messages:

$$q_{mn}^{(t)}(0) = P_n(0) \prod_{m' \in M(n) \backslash m} r_{m'n}^{(t)}(0)$$

$$q_{mn}^{(t)}(1) = P_n(1) \prod_{m' \in M(n) \backslash m} r_{m'n}^{(t)}(1)$$

Decoding Process(3)

- Decoding:

$$q_n^{(t)}(0) = P_n(0) \prod_{m \in M(n)} r_{mn}^{(t)}(0)$$

$$q_n^{(t)}(1) = P_n(1) \prod_{m \in M(n)} r_{mn}^{(t)}(1)$$

  - if $q_n^{(t)}(0) > q_n^{(t)}(1)$, then $\hat{c}_n = 0$;
  - else $\hat{c}_n = 1$.

Decoding Process(4)

- Stopping criterion test:
  - if $H\hat{c} = 0$, then the decoding process ends;
  - if t exceeds some maximum number, and $\hat{c}$ is considered as the final codeword, then the process ends;
  - otherwise, continue the iteration.

Outline

Log-likelihood Belief Propagation

Identical Equation

$\tanh(\frac{1}{2}\ln\frac{p_0}{p_1}) = p_0 - p_1 = 1 - 2p_1.$

$(p_0 + p_1 = 1, \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}})$

LLR

$r_{mn}^{(t)}(1) = \frac{1}{2} - \frac{1}{2}\prod_{n' \in N(m)\backslash n}(1 - 2q_{mn'}^{(t-1)}(1))$

$\Rightarrow$ 1-2$r_{mn}^{(t)}(1) = \prod_{n' \in N(m)\backslash n}(1 - 2q_{mn'}^{(t-1)}(1))$

$\Rightarrow \tanh(\frac{1}{2}\ln\frac{r_{mn}^{(t)}(0)}{r_{mn}^{(t)}(1)}) = \prod_{n' \in N(m)\backslash n}\tanh(\frac{1}{2}\ln\frac{q_{mn'}^{(t-1)}(0)}{q_{mn'}^{(t-1)}(1)})$

LLR BP

$$ln\frac{r_{mn}^{(t)}(0)}{r_{mn}^{(t)}(1)} = 2tanh^{-1}\left(\prod_{n'\in N(m)\backslash n} tanh\left(\frac{1}{2}ln\frac{q_{mn'}^{(t-1)}(0)}{q_{mn'}^{(t-1)}(1)}\right)\right); \quad (1)$$

$$ln\frac{q_{mn}^{(t)}(0)}{q_{mn}^{(t)}(1)} = ln\frac{P_n(0)}{P_n(1)} + \sum_{m'\in M(n)\backslash m} ln\frac{r_{m'n}^{(t)}(0)}{r_{m'n}^{(t)}(1)}. \quad (2)$$

Outline

Outline

The variable node LLR

The variable node LLR fluctuates continously during the iterative decoding:

- ...
- $\ln \frac{q_{mn}^{(t-1)}(0)}{q_{mn}^{(t-1)}(1)} > 0 \ (< 0)$;
- $\ln \frac{q_{mn}^{(t)}(0)}{q_{mn}^{(t)}(1)} < 0 \ (> 0)$;
- $\ln \frac{q_{mn}^{(t+1)}(0)}{q_{mn}^{(t+1)}(1)} > 0 \ (< 0)$;
- ...

Outline

CFT

Introduce a sequence of counters to record the Continuous Fluctuant Times (CFT) of the variable node LLRs:

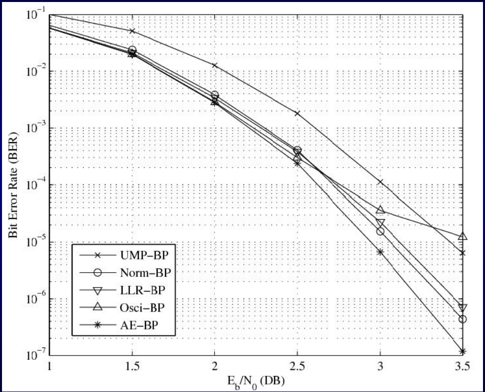
$$CFT_{mn}^{(t)} = \begin{cases} 0; & t = 0; \\ CFT_{mn}^{(t-1)} + 1; & ln\frac{q_{mn}^{(t-1)}(0)}{q_{mn}^{(t-1)}(1)} \cdot ln\frac{q_{mn}^{(t)}(0)}{q_{mn}^{(t)}(1)} < 0; \\ 0; & otherwise. \end{cases}$$

Erase the LLRs

$$ln\frac{q_{mn}^{(t)}(0)}{q_{mn}^{(t)}(1)} = \begin{cases} 0; & CFT_{mn}^{(t)} \geq 2; \\ ln\frac{q_{mn}^{(t-1)}(0)}{q_{mn}^{(t-1)}(1)} + ln\frac{q_{mn}^{(t)}(0)}{q_{mn}^{(t)}(1)}; & CFT_{mn}^{(t)} = 1; \\ ln\frac{q_{mn}^{(t)}(0)}{q_{mn}^{(t)}(1)}; & otherwise. \end{cases}$$

Outline

Simulation Results



Error performances for iterative decoding,
(3,6)-regular LDPC code with N=504 and R=1/2.

Outline

Main Progress

- Gallager, 1963: first proposed LDPC codes.
- Tanner, 1981: modeled the decoding process by Tanner Graph.
- Mackay&Neal, 1996: rediscovered LDPC codes.
- Luby, 1997: proposed irregular LDPC codes.
- Richardson&Urbanke, 2001: Density Evolution and code threshold.
- S.-Y Chung, 2001: within 0.0045dB of the Shannon Limit by Gaussian Approximation.
- M. Ardakani, 2004: semi-Gaussian Approximation.
- Now: Over *GF(q)* & Quasi-cyclic LDPC Codes.

LDPC Codes Decoding

BP

- BP, LLR-BP (SPA).
- UMP-BP: Fossorier 1999.
- Normalized/Offset BP: J. Chen 2002.
- BP based on Oscillation: S. Gounai 2006.

Bit-Flip

- ...

Next

Storage $\leftrightarrow$ Error Correcting Code

Thank you
& happy Niu year!