INFORMATION GEOMETRY OF TURBO AND LDPC CODES

Shiro Ikeda

Institute of Statistical Mathematics
4-6-7 Minami-Azabu, Minato-ku, Tokyo, 106-8569, Japan

1. INTRODUCTION

Turbo[2] and LDPC (Low-Density Parity Check) codes[3, 6] are simple and new type of error correction codes which give a powerful and practical performance of error correction. Although experimental results show their efficacy, further theoretical analysis is necessary, which is not straightforward. We have built unified framework of turbo and LDPC codes based on information geometry[1]. The framework helps our intuitive understanding of the codes and opens a new prospect of further analysis. We have revealed some properties of these codes in the proposed framework[4]. This paper summarizes the results.

2. INFERENCE PROBLEM AND INFORMATION GEOMETRICAL FRAMEWORK

2.1. Unified View

First, we give the unified view of turbo and LDPC codes. Let $\boldsymbol{x} \in \{-1, +1\}^N$, and the ultimate goal of both of the codes are to compute $\boldsymbol{\eta}_q = \sum_{\boldsymbol{x}} q(\boldsymbol{x})$ where $q(\boldsymbol{x})$ is the distribution of $\boldsymbol{x} = (x_1, \cdots, x_N)^T$, defined as

$$q(\boldsymbol{x}) = C \exp[c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_L(\boldsymbol{x})],$$

here, $c_0(\boldsymbol{x})$ consists of linear term of $\{x_i\}$, and $c_r(\boldsymbol{x})$ includes higher order terms of $\{x_i\}$. We restrict ourselves to the case x_i is binary, but generalization to multiple symbols is easy[5]. The forms of $c_r(\boldsymbol{x})$ are different in each code. The direct computation of $\boldsymbol{\eta}_q$ is not tractable, and both of the codes utilizes the following distributions

$$p_0(\boldsymbol{x}; \boldsymbol{\theta}^*) = \exp[c_0(\boldsymbol{x}) + \boldsymbol{\xi}_1^* \cdot \boldsymbol{x} + \cdots + \boldsymbol{\xi}_L^* \cdot \boldsymbol{x} - \psi_0(\boldsymbol{\theta}^*)]$$
$$p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1^*) = \exp[c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + \boldsymbol{\xi}_L^* \cdot \boldsymbol{x} - \psi_1(\boldsymbol{\zeta}_1^*)]$$
$$\vdots$$
$$p_L(\boldsymbol{x}; \boldsymbol{\zeta}_r^*) = \exp[c_0(\boldsymbol{x}) + \boldsymbol{\xi}_1^* \cdot \boldsymbol{x} + \cdots + c_L(\boldsymbol{x}) - \psi_L(\boldsymbol{\zeta}_L^*)].$$

p_0 is a factorizable distributions, and if we can choose p_0 to be equal to $\prod_i q(x_i)$, then $\boldsymbol{\eta}_0 = \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \boldsymbol{\eta}_q$, and

the exact decoding is possible. But generally, both codes only gives an approximation. In the following two subsections, we show that turbo and LDPC codes are formulated in this framework.
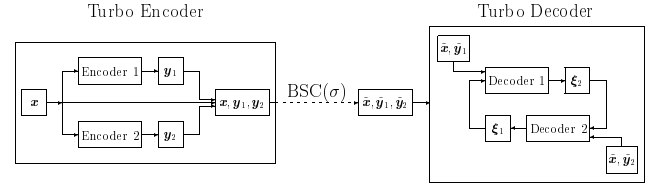
2.2. Turbo Codes

Fig. 1. Turbo codes

Let $\boldsymbol{x} \in \{-1, +1\}^N$ be the information bits, from which the turbo encoder generates two sets of parity bits, $\boldsymbol{y}_1 = (y_{11}, \cdots, y_{1L})^T$, and $\boldsymbol{y}_2 = (y_{21}, \cdots, y_{2L})^T$, $y_{1j}, y_{2j} \in \{-1, +1\}$ (Fig.1). Each parity bit is expressed in the form $\prod_{i \in \mathcal{L}_{rj}} x_i$, $(r = 1, 2)$, where $\mathcal{L}_{rj} \subset \{1, \cdots, N\}$. The codeword $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2)$ is transmitted over a noisy channel, which we assume a BSC (binary symmetric channel) with flipping probability $\sigma < 1/2$, and $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$, $\tilde{x}_i, \tilde{y}_{1j}, \tilde{y}_{2j} \in \{-1, +1\}$ are received.

Let us consider the form of the distribution $p(\boldsymbol{x} | \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. From the assumption of the memoryless channel,

$$p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2 | \boldsymbol{x}) = C \exp[\beta \tilde{\boldsymbol{x}} \cdot \boldsymbol{x} + \beta \tilde{\boldsymbol{y}}_1 \cdot \boldsymbol{y}_1 + \beta \tilde{\boldsymbol{y}}_2 \cdot \boldsymbol{y}_2]$$
$$\sigma = \frac{1}{2}(1 - \tanh \beta),$$

where '$\cdot$' denotes the inner-product. By assuming the uniform prior on $\boldsymbol{x}$, the posterior becomes

$$p(\boldsymbol{x} | \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) = \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2 | \boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2 | \boldsymbol{x})} \tag{1}$$
$$= C \exp[\beta \tilde{\boldsymbol{x}} \cdot \boldsymbol{x} + \beta \tilde{\boldsymbol{y}}_1 \cdot \boldsymbol{y}_1 + \beta \tilde{\boldsymbol{y}}_2 \cdot \boldsymbol{y}_2]$$
$$= C \exp[c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})].$$

C is the normalizing factor, and $c_0(\boldsymbol{x}) = \beta \tilde{\boldsymbol{x}} \cdot \boldsymbol{x}$, $c_r(\boldsymbol{x}) = \beta \tilde{\boldsymbol{y}}_r \cdot \boldsymbol{y}_r$, $(r = 1, 2)$. The MPM decoding is to compute the

signs of $\eta_i\ (i = 1, \cdots, N)$ defined as

$$\boldsymbol{\eta} = (\eta_1, \cdots, \eta_N)^T = \textstyle\sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2).$$

The direct computation of $\boldsymbol{\eta}$ is intractable. Turbo codes utilize two decoders. Each of them gives the soft decoding based on one of the two sets of the parity bits. For the soft decoding, the following $p_r(\boldsymbol{x}; \boldsymbol{\xi})\ (r = 1, 2)$ is used.

$$p_r(\boldsymbol{x}; \boldsymbol{\xi}_r) = \exp[c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\xi}_r \cdot \boldsymbol{x} - \psi_r(\boldsymbol{\xi}_r)], \quad (2)$$
$$\boldsymbol{\xi}_r \in \Re^N, \ \ \psi_r(\boldsymbol{\xi}_r) = \ln \textstyle\sum_{\boldsymbol{x}} \exp[c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\xi}_r \cdot \boldsymbol{x}].$$

This distribution is derived from $p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r | \boldsymbol{x})$ and the prior of $\boldsymbol{x}$ which has the form of
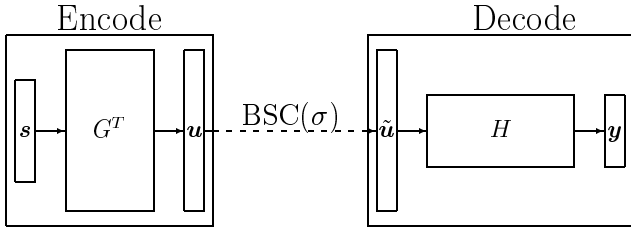
$$\omega(\boldsymbol{x}; \boldsymbol{\xi}) = \exp[\boldsymbol{\xi} \cdot \boldsymbol{x} - \psi(\boldsymbol{\xi})], \quad \psi(\boldsymbol{\xi}) = \textstyle\sum_i \psi(\xi_i).$$

The final decoding is obtained as $\mathrm{sgn}(\boldsymbol{\eta}_0)$, where

$$\boldsymbol{\eta}_0 = \textstyle\sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta})$$
$$p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \exp[c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} - \psi_0(\boldsymbol{\theta})], \quad \boldsymbol{\theta} \in \Re^N.$$

2.3. LDPC Codes

Fig. 2. Structure of LDPC codes.

The structure of LDPC codes is shown in Fig.2. $\boldsymbol{s} = (s_1, \cdots, s_M)^T$, $s_i \in \{0, 1\}$, is the information bits. The parity check matrix is

$$H = \{h_{ij}\} = (C_1\ C_2), \ \ H \in \{0, 1\}^{K \times N}.$$

$C_1 \in \{0, 1\}^{K \times M}$ and $C_2 \in \{0, 1\}^{K \times K}$ are sparse and C_2 is invertible in mod 2. The code $\boldsymbol{u} = (u_1, \cdots, u_N)^T$ is generated with $G^T \in \{0, 1\}^{N \times M}$ as

$$\boldsymbol{u} = G^T \boldsymbol{s}, \ \ G^T = \begin{pmatrix} E_M \\ C_2^{-1} C_1 \end{pmatrix} \mod 2,$$

where E_M is an identity matrix of size M. The first M bits of $\boldsymbol{u}$ are identical to $\boldsymbol{s}$, and $\boldsymbol{u}$ is sent through a channel. We assume a BSC with bit-error rate σ. Codeword $\boldsymbol{u}$ is disturbed and received as $\tilde{\boldsymbol{u}}$. Let $\boldsymbol{x} = (x_1, \cdots, x_N)^T$, $x_i \in \{0, 1\}$ be the noise vector, and received codeword $\tilde{\boldsymbol{u}}$ is

$$\tilde{\boldsymbol{u}} = \boldsymbol{u} + \boldsymbol{x} \mod 2.$$

In the decoding, noise vector $\boldsymbol{x}$ is estimated, which yields an estimate of $\boldsymbol{s}$. In the decoding process, H is used. Syndrome vector $\boldsymbol{y} = (y_1, \cdots, y_K)^T$ is defined as

$$\tilde{\boldsymbol{y}} = H\tilde{\boldsymbol{u}} = HG^T \boldsymbol{s} + H\boldsymbol{x} = H\boldsymbol{x} = \boldsymbol{y}(\boldsymbol{x}) \mod 2.$$

The decoding is to infer $\boldsymbol{x}$ that satisfies $\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x})$.

In the followings, we treat $\boldsymbol{s}, \boldsymbol{u}, \tilde{\boldsymbol{u}}, \tilde{\boldsymbol{y}}$, and $\boldsymbol{x}$ in the bipolar ($\{-1, +1\}$) form while G^T and H are still in the binary form. Each y_r of $\boldsymbol{y}(\boldsymbol{x})$ is written as a monomial in $\boldsymbol{x}$:

$$y_r(\boldsymbol{x}) = \textstyle\prod_{j \in \mathcal{L}_r} x_j, \ \ \mathcal{L}_r = \{j \mid h_{jr} = 1\}.$$

We consider the distribution of $\tilde{\boldsymbol{y}}$ conditioned on $\boldsymbol{x}$

$$\begin{aligned} p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) &= \exp[\rho \tilde{\boldsymbol{y}} \cdot \boldsymbol{y}(\boldsymbol{x}) - K\psi(\rho)] \\ &= \exp[c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x}) - K\psi(\rho)], \quad (3) \\ c_r(\boldsymbol{x}) &= \rho \tilde{y}_r y_r(\boldsymbol{x}), \ \ \rho \in \Re, \ \ \rho > 0, \end{aligned}$$

where a positive number ρ is introduced. We discuss the "soft constraint" which infers $\boldsymbol{x}$ based on $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$. The LDPC decoding algorithm generally uses the "hard constraint" $\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x})$. But as ρ becomes larger, both becomes equivalent.

The noise $\boldsymbol{x}$ is bitwise independent, and its error rate is $\sigma = (1/2)(1 - \tanh \beta)$. Let $\omega_0(\boldsymbol{x})$ be defined as follows

$$\omega_0(\boldsymbol{x}) = \exp[\beta \mathbf{1}_N \cdot \boldsymbol{x} - N\psi(\beta)] = \exp[c_0(\boldsymbol{x}) - N\psi(\beta)]$$
$$c_0(\boldsymbol{x}) = \beta \mathbf{1}_N \cdot \boldsymbol{x}.$$

As a result, the posterior distribution becomes

$$p(\boldsymbol{x}|\tilde{\boldsymbol{y}}) = C \exp[c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})].$$

The goal of the LDPC codes is also the MPM decoding, that is to compute $\boldsymbol{\eta} = \sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}|\tilde{\boldsymbol{y}})$.

In the LDPC decoding, $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)\ (r = 1, \cdots, K)$, which is composed of $p(\tilde{y}_r|\boldsymbol{x})$ and prior $\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ is used. We have

$$\begin{aligned} p(\tilde{y}_r|\boldsymbol{x}) &= \exp[c_r(\boldsymbol{x}) - \psi(\beta)], \\ \omega(\boldsymbol{x}; \boldsymbol{\zeta}_r) &= \exp[(\beta \mathbf{1}_N + \boldsymbol{\zeta}_r) \cdot \boldsymbol{x} - \psi(\beta \mathbf{1}_N + \boldsymbol{\zeta}_r)], \\ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) &= \exp[c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \psi_r(\boldsymbol{\zeta}_r)], \\ \psi_r(\boldsymbol{\zeta}_r) &= \ln \textstyle\sum_{\boldsymbol{x}} \exp[c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x}], \boldsymbol{\zeta}_r \in \Re^N. \end{aligned}$$

Now, we have shown that both of the codes are summarized with the unified view.

3. INFORMATION GEOMETRICAL VIEW

3.1. Preliminaries of Information Geometry

Let us first define the family of distributions S.

$$\begin{aligned} S = \Big\{ &p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) = \exp[c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} + \boldsymbol{v} \cdot \boldsymbol{c}(\boldsymbol{x}) - \psi(\boldsymbol{\theta}, \boldsymbol{v})], \\ &\boldsymbol{\theta} \in \Re^n, \boldsymbol{v} \in \Re^L \Big\}, \quad (4) \end{aligned}$$

here $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_n)^T$, $\boldsymbol{v} = (v_1, \cdots, v_L)^T$, and $\boldsymbol{c}(\boldsymbol{x}) = (c_1(\boldsymbol{x}), \cdots, c_L(\boldsymbol{x}))^T$. S is an exponential family, where $(\boldsymbol{\theta}, \boldsymbol{v})$ is the natural parameter. Next, let us define a submanifold $M_0 \subset S$ as

$$M_0 = \left\{ p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \exp[c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x} - \psi_0(\boldsymbol{\theta})] \mid \boldsymbol{\theta} \in \Re^n \right\}.$$

Each component is independent for the distributions of M_0 and $\prod_{i=1}^n q(x_i) \in M_0$. There exists a $\boldsymbol{\theta}$, s.t., $p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \prod_{i=1}^n q(x_i)$. We define the e– and m–flat submanifolds.

e–flat submanifold: Submanifold $M \subset S$ is e–flat, if $r(\boldsymbol{x}; t)$ belongs to M for all $t \in [0, 1]$, $q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$.

$$\ln r(\boldsymbol{x}; t) = (1 - t)\ln q(\boldsymbol{x}) + t \ln p(\boldsymbol{x}) + c(t),$$

m–flat submanifold: Submanifold $M \subset S$ is m–flat if $r(\boldsymbol{x}; t)$ belongs to M for all $t \in [0, 1]$, $q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$.

$$r(\boldsymbol{x}; t) = (1 - t)q(\boldsymbol{x}) + tp(\boldsymbol{x}).$$

From its definition, M_0 is e–flat. Next, we define m–projection to an e–flat submanifold.

Definition 1. *Let M be an e–flat submanifold in S, and let $q(\boldsymbol{x}) \in S$. The distribution in M that minimizes the KL–divergence from $q(\boldsymbol{x})$ to M is denoted by*

$$\Pi_M \circ q(\boldsymbol{x}) = \operatorname*{argmin}_{p(\boldsymbol{x}) \in M} D[q(\boldsymbol{x}); p(\boldsymbol{x})], \qquad (5)$$

and is called the m–projection of $q(\boldsymbol{x})$ to M.

$D[q; p]$ is the Kullback-Leibler divergence defined as

$$D[q(\boldsymbol{x}); p(\boldsymbol{x})] = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \ln \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}.$$

Finally, the m–projection theorem follows.

Theorem 1. *Let M be an e–flat submanifold in S and let $q(\boldsymbol{x}) \in S$. $\Pi_M \circ q(\boldsymbol{x})$ is unique.*

Now, we show that marginalization of $q(\boldsymbol{x})$ is equivalent to m–projection from $q(\boldsymbol{x})$ to M_0. Let $\boldsymbol{\theta}^*$ be the natural parameter of M_0 which corresponds to the m–projection of $q(\boldsymbol{x})$ to M_0. From eq. (5),

$$p_0(\boldsymbol{x}; \boldsymbol{\theta}^*) = \Pi_{M_0} \circ q(\boldsymbol{x}),$$

$$\boldsymbol{\theta}^* \stackrel{\text{def}}{=} \pi_{M_0} \circ q(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\theta}} D[q(\boldsymbol{x}); p_0(\boldsymbol{x}; \boldsymbol{\theta})].$$

Take the derivative of $D[q(\boldsymbol{x}); p_0(\boldsymbol{x}; \boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$, and we have

$$\sum_{\boldsymbol{x}} \boldsymbol{x} q(\boldsymbol{x}) - \partial_{\boldsymbol{\theta}} \psi_0(\boldsymbol{\theta}^*) = \boldsymbol{o}. \qquad (6)$$

We have the following relation

$$\partial_{\boldsymbol{\theta}} \psi_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} \ln \sum_{\boldsymbol{x}} \exp[c_0(\boldsymbol{x}) + \boldsymbol{\theta} \cdot \boldsymbol{x}] = \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta}).$$

Now, we define $\boldsymbol{\eta}_0(\boldsymbol{\theta})$ as the expectation parameter of M_0,

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} \psi_0(\boldsymbol{\theta}).$$

Since $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_0$ has one-to-one relation, the m–projection from $q(\boldsymbol{x})$ to M_0 is equivalent to marginalization of $q(\boldsymbol{x})$.

3.2. Decoding Algorithms

Let us define the following submanifolds

$$M_r = \left\{ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) \mid \boldsymbol{\zeta}_r \in \Re^n \right\},$$

$$p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) = \exp[c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \psi_r(\boldsymbol{\zeta}_r)],$$

$$\boldsymbol{\zeta}_r \in \Re^n, \quad r = 1, \cdots, L.$$

$p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ is an exponential family which includes $c_r(\boldsymbol{x})$. Its natural and expectation parameters are $\boldsymbol{\zeta}_r$ and $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)$, respectively, and defined as follows

$$\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) = \partial_{\boldsymbol{\zeta}_r} \psi_r(\boldsymbol{\zeta}_r) = \sum_{\boldsymbol{x}} \boldsymbol{x} p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r), \quad r = 1, \cdots, L.$$

Computation of $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)$ and $\pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ is possible for every $\boldsymbol{\zeta}_r \in \Re^n$. Using $p_0(\boldsymbol{x}; \boldsymbol{\theta}^t)$ and $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t)$ $r = 1, \cdots, L$, we can rewrite the decoding algorithms as follows[4],

Decoding algorithms

1. Set $\boldsymbol{\xi}_r^t = \boldsymbol{o}$, $\boldsymbol{\zeta}_r^t = \boldsymbol{o}$, $r = 1, \cdots, L$.

2. Increase t by 1 and update $\boldsymbol{\xi}_r^{t+1}$ as follows

$$\boldsymbol{\xi}_r^{t+1} = \pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t) - \boldsymbol{\zeta}_r^t. \qquad (7)$$

3. Update $\boldsymbol{\theta}^{t+1}$ and $\boldsymbol{\zeta}_r^{t+1}$ as follows

$$\boldsymbol{\zeta}_r^{t+1} = \sum_{r' \neq r} \boldsymbol{\xi}_{r'}^{t+1}, \, \boldsymbol{\theta}^{t+1} = \sum_r \boldsymbol{\xi}_r^{t+1} = \frac{1}{L-1} \sum_r \boldsymbol{\zeta}_r^{t+1}.$$

4. Repeat 2 and 3 until the convergence.

In turbo codes, $L = 2$ and step 2 updates one of $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\}$ iteratively, while in LDPC codes, L is large all of $\{\boldsymbol{\xi}_r\}$ are updated simultaneously. In the following, we denote the parameters at the convergent as $\boldsymbol{\theta}^*$, $\{\boldsymbol{\zeta}_r^*\}$, and $\{\boldsymbol{\xi}_r^*\}$. Note that $\boldsymbol{\theta}^t = \boldsymbol{\zeta}_r^t + \boldsymbol{\xi}_r^t$ holds throughout the algorithm.

3.3. Equilibrium

Theorem 2. [4] *The equilibrium $(\boldsymbol{\theta}^*, \{\boldsymbol{\zeta}_r^*\})$ satisfies*

m–condition: $\boldsymbol{\theta}^* = \pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*).$

e–condition: $\boldsymbol{\theta}^* = \dfrac{1}{L-1} \sum_{r=1}^L \boldsymbol{\zeta}_r^*.$

From eq. (7) and $\boldsymbol{\theta}^* = \boldsymbol{\zeta}_r^* + \boldsymbol{\xi}_r^*$, it is clear that m–condition is satisfied, and e–condition is clear from step 3. Next, we define two submanifolds M^* and E^* in S.

$$M^* = \left\{ p(\boldsymbol{x}) \,\Big|\, \sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}) = \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta}^*) = \boldsymbol{\eta}_0(\boldsymbol{\theta}^*) \right\},$$

$$E^* = \left\{ p(\boldsymbol{x}) = Z(\boldsymbol{t}) p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)^{t_0} \prod_{r=1}^L p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)^{t_r} \,\Big|\, \sum_{r=0}^L t_r = 1 \right\},$$

M^* and E^* are an m–flat and an e–flat submanifolds, respectively. From the definition, M^* includes $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$, and E^* includes $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ and $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, $r = 1, \cdots, L$. Now, 2 conditions are redefined as follows,

m–**condition:** M^* includes $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, $r = 1, \cdots, L$.

e–**condition:** E^* includes $q(\boldsymbol{x})$.

The m–condition is easily checked and e–condition is also checked by setting $t_0 = -(L-1), t_1 = \cdots = t_L = 1$.

At the equilibrium, e– and m–conditions are satisfied. If $q(\boldsymbol{x})$ is included in M^*, the decoding result is exact, but $q(\boldsymbol{x})$ is only included in E^*, and M^* differs from E^*.

3.4. Perturbation Analysis

We have analyzed the accuracy of these decoding algorithms based on the perturbation analysis[4]. In the following discussion, we use $p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v})$ in eq. (4). From the definition, $p_0(\boldsymbol{x}; \boldsymbol{\theta})$, $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$, and $q(\boldsymbol{x}) \in S$. The partial expectation parameter is

$$\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = \partial_{\boldsymbol{\theta}} \psi(\boldsymbol{\theta}, \boldsymbol{v}) = \sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}).$$

We now restrict $p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v})$ to belong to M^*. In other words, we set the parameters to satisfy the following equation

$$\boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*, \mathbf{o}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

Under the constraint, $\boldsymbol{\theta}$ is a function of $\boldsymbol{v}$, as $\boldsymbol{\theta}(\boldsymbol{v})$. We analyze how $\boldsymbol{\theta}$ changes from $\boldsymbol{\theta}^*$, as $\boldsymbol{v}$ moves from $\mathbf{o}$ to $\mathbf{1}_L$.

In the perturbation analysis based on the second order expansion, we used the following two equations.

$$0 = \frac{d}{d\boldsymbol{v}} \boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v}) = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{v}}.$$

$$0 = \frac{d^2}{d\boldsymbol{v} d\boldsymbol{v}} \boldsymbol{\eta}(\boldsymbol{\theta}, \boldsymbol{v})$$
$$= 2 \frac{\partial^2 \boldsymbol{\eta}}{\partial \boldsymbol{\theta} \partial \boldsymbol{v}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} + \frac{\partial^2 \boldsymbol{\eta}}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}} + \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\theta}} \frac{\partial^2 \boldsymbol{\theta}}{\partial \boldsymbol{v} \partial \boldsymbol{v}} + \frac{\partial^2 \boldsymbol{\eta}}{\partial \boldsymbol{v} \partial \boldsymbol{v}},$$

which yields

$$\boldsymbol{\theta}(\boldsymbol{v}) = \boldsymbol{\theta}^* + \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{v}}\Big|_{\boldsymbol{v}=\mathbf{o}} \boldsymbol{v} + \frac{1}{2} \boldsymbol{v}^T \frac{\partial^2 \boldsymbol{\theta}}{\partial \boldsymbol{v} \partial \boldsymbol{v}}\Big|_{\boldsymbol{v}=\mathbf{o}} \boldsymbol{v} + o(\|\boldsymbol{v}\|^3)$$
$$\simeq \boldsymbol{\theta}^* - \tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}^*)\boldsymbol{v} - \frac{1}{2} \boldsymbol{v}^T I_0^{-1}(\boldsymbol{\theta}^*)(B^2 \boldsymbol{\eta}(\boldsymbol{\theta}^*))\boldsymbol{v}. \quad (8)$$

Here, $\tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}} = I_0^{-1}(\boldsymbol{\theta}) G_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta})$ and $B = \frac{\partial}{\partial \boldsymbol{v}} - \tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}^*)\frac{\partial}{\partial \boldsymbol{\theta}}$. For simplicity, we denote the (r, s) component of B^2 as $B_{rs} = B_r B_s$. Here, even if $(d^2/d\boldsymbol{v} d\boldsymbol{v}) \boldsymbol{\eta}(\boldsymbol{\theta}^*) = \mathbf{o}$ holds, generally $B^2 \boldsymbol{\eta}(\boldsymbol{\theta}^*) \neq \mathbf{o}$.

So far, we have only considered the m–condition. Now, we move to the e–condition. The condition is

$$\boldsymbol{\theta}^* = -\sum_{r=1}^{L} (\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^*).$$

Since $\boldsymbol{\zeta}_r^* = \boldsymbol{\theta}(\boldsymbol{e}_r)$, eq. (8) gives

$$\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^* \approx -\tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}^*)\boldsymbol{e}_r - \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) B_{rr} \boldsymbol{\eta}(\boldsymbol{\theta}^*)$$

$$\boldsymbol{\theta}^* \approx \tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}^*)\mathbf{1}_L + \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) \sum_r B_{rr} \boldsymbol{\eta}(\boldsymbol{\theta}^*). \quad (9)$$

Now we have the following approximation

$$\boldsymbol{\theta}(\mathbf{1}_L) - \boldsymbol{\theta}^* \approx -\tilde{G}_{\boldsymbol{\theta}\boldsymbol{v}}(\boldsymbol{\theta}^*)\mathbf{1}_L - \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) \sum_{rs} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

$$\boldsymbol{\theta}(\mathbf{1}_L) \approx -\frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) \sum_{r \neq s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*). \quad (10)$$

This yields the following theorem

Theorem 3. *Let the expectation of $\boldsymbol{x}$ w.r.t $q(\boldsymbol{x})$ be $\boldsymbol{\eta}_q = \boldsymbol{\eta}(\mathbf{o}, \mathbf{1}_L)$. And we have the following approximation,*

$$\boldsymbol{\eta}_q \approx \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2} \sum_{r \neq s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

When every $c_r(\boldsymbol{x})$ is a monomial of $\{x_i\}$, we further have the following result.

Theorem 4. *[7] When any pair $c_r(\boldsymbol{x})$ and $c_{r'}(\boldsymbol{x})$, does not have more than one common x_i, $\sum_{r \neq s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*)/2 = 0$.*

4. CONCLUSION

We have shown our information geometrical framework of turbo and LDPC codes. Base on the framework, we have shown the perturbation analysis result. We also have studied the local convergence properties and analysis of related algorithms in [4, 5].

5. REFERENCES

[1] S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS and Oxford University Press, 2000.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *Proc. IEEE Int. Conf. on Commun.*, pages 1064–1070, 1993.

[3] R. G. Gallager. Low density parity check codes. *IRE Trans. Inform. Theory*, IT-8:21–28, 1962.

[4] S. Ikeda, T. Tanaka, and S. Amari. Information geometry of turbo and low-density parity-check codes. *IEEE Trans. Inform. Theory*, 50(6):1097–1114, 2004.

[5] S. Ikeda, T. Tanaka, and S. Amari. Stochastic reasoning, free energy, and information geometry. *Neural Computation*, 16(9):1779–1810, 2004.

[6] D. J. C. MacKay. Good error-correcting codes based on very sparse matrices. *IEEE Trans. Inform. Theory*, 45(2):399–431, March 1999.

[7] T. Tanaka, S. Ikeda, and S. Amari. Information-geometrical significance of sparsity in Gallager codes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS 14*, pp. 527–534. MIT Press, 2002.