Belief propagation and turbo code: Information geometrical view

Shiro Ikeda[†,†††], *Toshiyuki Tanaka*[††], *and Shun-ichi Amari*[†††]

Email:shiro@brain.kyutech.ac.jp, tanaka@eei.metro-u.ac.jp, amari@brain.riken.go.jp

† Kyushu Institute of Technology, & PRESTO, JST, Kitakyushu, Fukuoka 808–0196, Japan
†† Tokyo Metro. Univ., Fac. of Eng., 1–1 Minami Oosawa, Hachioji, Tokyo, 192–0397 Japan
††† RIKEN, BSI, Hirosawa 2–1, Wako, Saitama 351–0198, Japan

ABSTRACT

In this article, we describe the information geometrical understanding of the belief propagation decoder, especially of the turbo decoding. The turbo decoding was proposed by Berrou *et al.* early in 90's, and many studies have been appeared on this practical and powerful error correcting code. Even though many experimental results support the potential of the turbo decoding, there is not sufficient theoretical analysis for the decoding method. We investigate the problem from information geometrical viewpoint. From the new viewpoint, we establish a new framework for analyzing the turbo code, and reveal basic properties.

KEYWORDS: turbo decoding, belief propagation, information geometry

1. Introduction

Since the turbo code[4] was proposed, its high performance of error correction has been investigated mainly through experiments. The experimental results strongly support the potential of this iterative decoding method, however theoretical analysis has not revealed the mystery of the turbo decoding. Further theoretical understanding was sought in similar iterative methods. McEliece et al.[7] have shown the turbo decoding is equivalent to the Pearl's BP algorithm[8] applied to a special BN. Although there is a beautiful theoretical result for the BP, the proof is only available for BNs without loops, while the BN for the turbo decoding is loopy. BP for a loopy BN gives only an approximation, and the approximation ability is not clearly understood.

This article gives an information geometrical understanding of the turbo decoding. We have developed a mathematical framework for analyzing the turbo code based on the information geometry. Based on the framework, we revealed some basic properties such as local stability condition, convergence properties, cost function of the algorithm, and approximation accuracy. In this article, we show the analysis of the turbo decoding since the structure of its BN is rather simple, but the result is general.

2. General problem

Let us define a distribution of $\boldsymbol{x} = (x_1, \cdots, x_N)^T$ as follows

$$q(\boldsymbol{x}) = C \exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})). \quad (1)$$

Here, $c_0(\boldsymbol{x})$ is the linear function of $\{x_i\}$, and $c_r(\boldsymbol{x})$ $r = 1, \cdots, K$ consists of the higher order correlations of $\{x_i\}$. The ultimate goal of the turbo decoding is the MPM (maximization of the posterior marginals) decoding, and it can be generalized as the marginalization of a probability distribution $q(\boldsymbol{x})$. The practical form of $c_r(\boldsymbol{x})$, $r = 0, \cdots, K$ is given in the following subsection. Let Π denote the operator of marginalization as, $\Pi \circ q(\boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{N} q(x_i)$. The marginalization is equivalent to take the expectation of $\boldsymbol{x}$ as

$$\boldsymbol{\eta} \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}} \boldsymbol{x} p(\boldsymbol{x}), \quad \boldsymbol{\eta} = (\eta_1, \cdots, \eta_N)^T, \quad (2)$$

where the sign of each η_i is the result of the MPM decoding. In the case of the turbo decoding, the marginalization of eq.(1) is not tractable, but the marginalization of the following distribution is tractable.

$$p_r(\boldsymbol{x}; \boldsymbol{\xi}) = \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\xi} \cdot \boldsymbol{x} - \varphi_r(\boldsymbol{\xi})) \quad r = 1, \cdots, K, \quad \boldsymbol{\xi} \in \mathcal{R}^N. \quad (3)$$

Each $p_r(\boldsymbol{x}; \boldsymbol{\xi})$ includes a part of $\{c_r(\boldsymbol{x})\}$ in eq.(1), and additional parameter $\boldsymbol{\xi}$ is used to adjust the linear part of $\boldsymbol{x}$. The turbo decoding is exchanging information through $\boldsymbol{\xi}$ for each p_r, and finally approximates $\Pi \circ p(\boldsymbol{x})$.

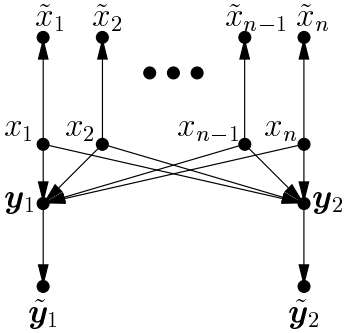

Figure 1: Belief network for turbo decoding

The BP algorithm was proposed by Pearl. The "belief" of x_i is equivalent to η_i in eq.(2). When a BN is loop free, the BP gives an exact marginalization, but if the BN is loopy, BP gives only an approximation. BN for the turbo decoding is shown in Fig.1. This BN has loops, and if the BP is applied to the node as, $\boldsymbol{x}$, $\boldsymbol{y}_1$, $\boldsymbol{x}$, $\boldsymbol{y}_2$, $\boldsymbol{x}$, $\boldsymbol{y}_1$, $\cdots$, it is equivalent to the turbo decoding.

3. Turbo code

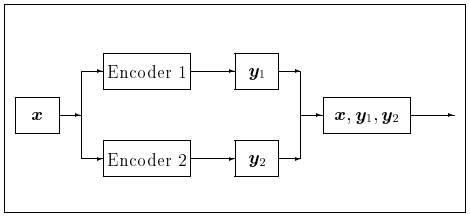

Figure 2: Turbo encoding

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

The goal of the turbo decoding is the MPM decoding based on $p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. From the assumption of BSC,

$$
\begin{aligned}
&p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x}) \\
&= \exp(\beta\tilde{\boldsymbol{x}}\cdot\boldsymbol{x} + \beta\tilde{\boldsymbol{y}}_1\cdot\boldsymbol{y}_1 + \beta\tilde{\boldsymbol{y}}_2\cdot\boldsymbol{y}_2 - (N+2K)\psi(\beta)) \\
&\beta > 0, \;\; \sigma = \frac{1}{2}(1 - \tanh\beta), \;\; \psi(\beta) \stackrel{\text{def}}{=} \ln(e^{\beta} + e^{-\beta}).
\end{aligned}
$$

By assuming the uniform prior on $\boldsymbol{x}$, the posterior distribution of $\boldsymbol{x}$ becomes,

$$
\begin{aligned}
p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) &= \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})} \\
&= C\exp\left(\beta\tilde{\boldsymbol{x}}\cdot\boldsymbol{x} + \beta\tilde{\boldsymbol{y}}_1\cdot\boldsymbol{y}_1 + \beta\tilde{\boldsymbol{y}}_2\cdot\boldsymbol{y}_2\right) \\
&= C\exp\left(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})\right).
\end{aligned} \tag{4}
$$

Here C is the normalizing factor, and $c_0(\boldsymbol{x}) = \beta\tilde{\boldsymbol{x}}\cdot\boldsymbol{x}$, $c_r(\boldsymbol{x}) = \beta\tilde{\boldsymbol{y}}_r\cdot\boldsymbol{y}_r$ $r = 1, 2$. Equation(4) is equivalent to $q(\boldsymbol{x})$ in eq.(1), where $K = 2$. We describe $p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$ with $q(\boldsymbol{x})$ for the following of the article. When N is large, marginalization of $q(\boldsymbol{x})$ is intractable since it needs summation over 2^N terms. The turbo code utilizes two decoders which solve the marginalization of $p_r(\boldsymbol{x}; \boldsymbol{\xi})$, $r = 1, 2$ in eq.(3). The distribution is derived from $p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})$ and the prior of $\boldsymbol{x}$ which is defined as,

$$
\omega(\boldsymbol{x}; \boldsymbol{\xi}) = \exp(\boldsymbol{\xi}\cdot\boldsymbol{x} - \psi(\boldsymbol{\xi})).
$$

The marginalization of $p_r(\boldsymbol{x}; \boldsymbol{\xi})$ is tractable for $\boldsymbol{\xi} \in \mathcal{R}^N$.

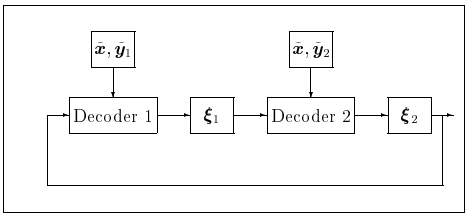

Figure 3: Turbo decoding

We give the original definition of the turbo decoding(Fig.3). Let us define the following variables based on the conditional probabilities $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$, and $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, $r = 1, 2$, of the received signals

$$
\begin{aligned}
lx_i &\stackrel{\text{def}}{=} \ln\frac{\sum_{\{\boldsymbol{x}:x_i=+1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}:x_i=-1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}, \\
ly_{rj} &\stackrel{\text{def}}{=} \ln\frac{\sum_{\{\boldsymbol{x}:y_{rj}=+1\}} p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}:y_{rj}=-1\}} p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}, \\
F(l\boldsymbol{x}, l\boldsymbol{y}_r) &\stackrel{\text{def}}{=} \left\{ \ln\frac{\sum_{\{\boldsymbol{x}:x_i=+1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}:x_i=-1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})} \right\}.
\end{aligned}
$$

The turbo decoding makes use of the two slack variables, $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in R^N$. These variables are used for exchanging information between the decoders. The turbo decoding consists of the following iterative procedures.

1. Let $\boldsymbol{\xi}_1^0 = 0$ and put $t = 1$.
2. Update $\boldsymbol{\xi}_2^t$ as,
$$
\boldsymbol{\xi}_2^t = F((l\boldsymbol{x} + \boldsymbol{\xi}_1^{t-1}), l\boldsymbol{y}_1) - (l\boldsymbol{x} + \boldsymbol{\xi}_1^{t-1}).
$$
3. Update $\boldsymbol{\xi}_1^t$ as,
$$
\boldsymbol{\xi}_1^t = F((l\boldsymbol{x} + \boldsymbol{\xi}_2^t), l\boldsymbol{y}_2) - (l\boldsymbol{x} + \boldsymbol{\xi}_2^t).
$$
4. Iterate 2 and 3 until $F((l\boldsymbol{x} + \boldsymbol{\xi}_1^{t-1}), l\boldsymbol{y}_1) = F((l\boldsymbol{x} + \boldsymbol{\xi}_2^t), l\boldsymbol{y}_2)$.

Ideally, step 2 and 3 should be iterated until convergence, but practically, the number of iteration is fixed to a few to ten times.

4. Information geometrical view

4.1. Preliminaries

In this subsection, we give the preliminaries of information geometry. We consider the family of all probability distributions over $\boldsymbol{x}$, which we call S. The e–flat and m–flat submanifolds are defined as follows[1, 3].

e–flat manifold: A submanifold $M \in S$ is e–flat, when the following $r(\boldsymbol{x}; t)$ belongs to M for all $q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$,

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

where c is the normalization factor.

m–flat manifold: A submanifold $M \in S$ is m–flat, when the following $r(\boldsymbol{x};t)$ belongs to M for all $q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$,

$$r(\boldsymbol{x};t) = (1-t)q(\boldsymbol{x}) + tp(\boldsymbol{x}), \qquad t \in [0,1].$$

Now, we consider a submanifold M_D, where every joint distribution can be decomposed as,

$$p(\boldsymbol{x}) = \prod_{i=1}^{N} p_i(x_i), \qquad p(\boldsymbol{x}) \in M_D. \tag{5}$$

Each bit of $\boldsymbol{x}$ is independent for $p(\boldsymbol{x}) \in M_D$. Since each bit is binary, $p(\boldsymbol{x})$ belongs to an exponential family,

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \prod_{i=1}^{N} p_i(x_i;\theta^i) = \exp\left(\boldsymbol{\theta}\cdot\boldsymbol{x} - \psi(\boldsymbol{\theta})\right). \tag{6}$$

The natural parameter $\boldsymbol{\theta} \in \mathcal{R}^N$ gives a coordinate system of M_D. The submanifold M_D is written as,

$$M_D = \left\{p(\boldsymbol{x};\boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}\cdot\boldsymbol{x} - \psi(\boldsymbol{\theta})\right)\right\}.$$

M_D is an e–flat submanifold[3].

We now define three e–flat submanifolds which depend on the observed data $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{y}}_1$, and $\tilde{\boldsymbol{y}}_2$. The first one is the submanifold of $p_0(\boldsymbol{x};\boldsymbol{\theta})$ defined by

$$M_0 = \left\{p_0(\boldsymbol{x};\boldsymbol{\theta}) = \exp\left(c_0(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi_0(\boldsymbol{\theta})\right)\right\}. \tag{7}$$

M_0 is identical to M_D of the independent or decomposable distributions, since $c_0(\boldsymbol{x}) = \beta\tilde{\boldsymbol{x}}\cdot\boldsymbol{x}$. Here, the coordinate $\boldsymbol{\theta}$ is shifted by $\beta\tilde{\boldsymbol{x}}$ compared to the coordinate of M_D. We use the new coordinates $\boldsymbol{\theta}$ of M_0, in which integration of informations from the component encoders takes place.

Next, we consider the submanifolds M_r, $r = 1, 2$ which includes a part of the information of $q(\boldsymbol{x})$,

$$M_r = \left\{p_r(\boldsymbol{x};\boldsymbol{\xi}) = \exp\left(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\xi}\cdot\boldsymbol{x} - \varphi_r(\boldsymbol{\xi})\right)\right\}.$$

Here $\boldsymbol{\xi}$ is the coordinate system of M_r. It is easy to check that M_r is also an e–flat submanifold. But $M_r \neq M_0$, because $c_r(\boldsymbol{x})$ includes higher order cross terms of $\boldsymbol{x}$, and $M_1 \neq M_2$ holds, since $c_1(\boldsymbol{x}) \neq c_2(\boldsymbol{x})$ in general.

We also define the expectation parameters as follows with φ_0 in eq.(7) and φ_r in eq.(3)

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x};\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}),$$

$$\boldsymbol{\eta}_r(\boldsymbol{\xi}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}} \boldsymbol{x} p_r(\boldsymbol{x};\boldsymbol{\xi}) = \partial_{\boldsymbol{\xi}}\varphi_r(\boldsymbol{\xi}) \quad r = 1, 2.$$

Next, we show that the marginalization corresponds to the m–projection[3] to M_0.

The m–projection of $q(\boldsymbol{x})$ to M_0 is defined by,

$$\Pi_{M_0}\circ q(\boldsymbol{x}) = \operatorname*{argmin}_{p_0(\boldsymbol{x};\boldsymbol{\theta}) \in M_0} D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})].$$

Here, $D[\cdot;\cdot]$ is the Kullback-Leibler divergence,

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

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

$$\partial_{\boldsymbol{\theta}} D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})] = \boldsymbol{\eta}_0 - \sum_{\boldsymbol{x}} \boldsymbol{x} q(\boldsymbol{x}).$$

This derivative vanishes at the projected point. Hence the η–coordinates of the projected point $\boldsymbol{\eta}_0^*$ is given by $\boldsymbol{\eta}_0^* = \sum_{\boldsymbol{x}} \boldsymbol{x} q(\boldsymbol{x})$,

$$\eta_{0,i}^* = \sum_{\boldsymbol{x}} x_i q(\boldsymbol{x}) = \sum_{x_i} x_i q(x_i).$$

The m–projection of $q(\boldsymbol{x})$ does not change the expectation of $\boldsymbol{x}$, and the m–projection of $q(\boldsymbol{x})$ to M_0 results in marginalization of $q(\boldsymbol{x})$.

4.2. Information geometrical view of turbo decoding

Let $\pi_{M_0}\circ q(\boldsymbol{x})$ denote the parameter in M_0 of the m–projected distribution,

$$\pi_{M_0}\circ q(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{R}^N} D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})].$$

The information geometrical definition of the turbo decoding can be written as follows.

Turbo decoding

1. Let $\boldsymbol{\xi}_1^t = 0$ for $t = 0$, and $t = 1$.
2. Project $p_2(\boldsymbol{x};\boldsymbol{\xi}_1^t)$ onto M_0 as $\pi_{M_0}\circ p_2(\boldsymbol{x};\boldsymbol{\xi}_1^t)$ and calculate $\boldsymbol{\xi}_2^{t+1}$ as,
$$\boldsymbol{\xi}_2^{t+1} = \pi_{M_0}\circ p_2(\boldsymbol{x};\boldsymbol{\xi}_1^t) - \boldsymbol{\xi}_1^t. \tag{8}$$
3. Project $p_1(\boldsymbol{x};\boldsymbol{\xi}_2^{t+1})$ onto M_0 as $\pi_{M_0}\circ p_1(\boldsymbol{x};\boldsymbol{\xi}_2^{t+1})$ and calculate $\boldsymbol{\xi}_1^{t+1}$ as,
$$\boldsymbol{\xi}_1^{t+1} = \pi_{M_0}\circ p_1(\boldsymbol{x};\boldsymbol{\xi}_2^{t+1}) - \boldsymbol{\xi}_2^{t+1}. \tag{9}$$
4. If $\pi_{M_0}\circ p_1(\boldsymbol{x};\boldsymbol{\xi}_2^{t+1})$ and $\pi_{M_0}\circ p_2(\boldsymbol{x};\boldsymbol{\xi}_1^{t+1})$ does not converge, go to step 2.

Finally, the turbo decoding approximates the estimated parameter $\boldsymbol{\theta}^*$, the projection of $q(\boldsymbol{x})$ onto M_0, as

$$\boldsymbol{\theta}^* = \boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^*, \tag{10}$$

where, the estimated distribution is written as,

$$p_0(\boldsymbol{x};\boldsymbol{\theta}^*) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}^* \cdot \boldsymbol{x} - \varphi_0(\boldsymbol{\theta}^*)) \\ = \exp\left(c_0(\boldsymbol{x}) + (\boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^*)\cdot\boldsymbol{x} - \varphi_0(\boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^*)\right). \quad (11)$$

The intuitive understanding of the turbo decoding is as follows. In step 2, $(\boldsymbol{\xi}_2 \cdot \boldsymbol{x})$ in eq.(11) is replaced with $c_2(\boldsymbol{x})$. The distribution becomes $p_2(\boldsymbol{x};\boldsymbol{\xi}_1)$, and $\boldsymbol{\xi}_2$ is estimated by projecting it onto M_D. In step 3, $(\boldsymbol{\xi}_1 \cdot \boldsymbol{x})$ in eq.(11) is replaced with $c_1(\boldsymbol{x})$, and $\boldsymbol{\xi}_1$ is estimated by $m-$projection of $p_1(\boldsymbol{x};\boldsymbol{\xi}_2)$.

4.3. Equilibrium

Let us denote the convergent state of the turbo decoding as $p_1(\boldsymbol{x};\boldsymbol{\xi}_1^*)$, $p_2(\boldsymbol{x};\boldsymbol{\xi}_2^*)$, and $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$. They satisfy the following two conditions:

1. $\Pi \circ p_1(\boldsymbol{x};\boldsymbol{\xi}_2^*) = \Pi \circ p_2(\boldsymbol{x};\boldsymbol{\xi}_1^*) = p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ (12)
2. $\boldsymbol{\theta}^* = \boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^*$ (13)

For the following discussion, we define two submanifolds. One is the submanifold $M(\boldsymbol{\theta})$ defined as

$$M(\boldsymbol{\theta}) = \left\{ p(\boldsymbol{x}) \middle| \sum_{\boldsymbol{x}} p(\boldsymbol{x})\boldsymbol{x} = \sum_{\boldsymbol{x}} p_0(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{x} \right\}.$$

From its definition, the expectation of $\boldsymbol{x}$ is the same for any $p(\boldsymbol{x}) \in M(\boldsymbol{\theta})$. This is an m–flat submanifold, and for any $p(\boldsymbol{x}) \in M(\boldsymbol{\theta})$, its m–projection to M_0 coincides to $p_0(\boldsymbol{x};\boldsymbol{\theta})$. Since the m–projection to M_0 is the marginalization of $p(\boldsymbol{x})$, we call $M(\boldsymbol{\theta})$ the equimarginal submanifold.

Let us define an e–flat submanifold $E(\boldsymbol{\theta})$ connecting $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\boldsymbol{\xi}_2^*)$, and $p_2(\boldsymbol{x};\boldsymbol{\xi}_1^*)$.

$$E(\boldsymbol{\theta}^*) = \left\{ p(\boldsymbol{x}) = C p_0(\boldsymbol{x};\boldsymbol{\theta}^*)^{t_0} p_1(\boldsymbol{x};\boldsymbol{\xi}_2^*)^{t_1} p_2(\boldsymbol{x};\boldsymbol{\xi}_1^*)^{t_2} \middle| \sum_{r=0}^{2} t_r = 1 \right\}.$$

At the equilibrium, eq.(13) is satisfied, and $q(\boldsymbol{x})$ is included in $E(\boldsymbol{\theta}^*)$. It is proved by taking $t_0 = -1$ and $t_1 = t_2 = 1$

$$C \frac{p_1(\boldsymbol{x};\boldsymbol{\xi}_2^*) p_2(\boldsymbol{x};\boldsymbol{\xi}_1^*)}{p_0(\boldsymbol{x};\boldsymbol{\theta}^*)} \\ = C \exp\left(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})\right) = q(\boldsymbol{x}).$$

That is, eq.(13) implies that the conditional distribution is included in the e–flat submanifold M. This is summarized in the following Theorem.

Theorem 1. *When the turbo decoding converges, the convergent probability distributions $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\boldsymbol{\xi}_2^*)$, and $p_2(\boldsymbol{x};\boldsymbol{\xi}_1^*)$ belong to $M(\boldsymbol{\theta}^*)$ while $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\boldsymbol{\xi}_2^*)$, $p_2(\boldsymbol{x};\boldsymbol{\xi}_1^*)$, and $q(\boldsymbol{x})$ belong to $E(\boldsymbol{\theta}^*)$.*

If $M(\boldsymbol{\theta}^*)$ includes $q(\boldsymbol{x})$, $p(\boldsymbol{x};\boldsymbol{\theta}^*)$ is the true marginalization of $q(\boldsymbol{x})$. Instead of $M(\boldsymbol{\theta}^*)$, its e–flat version $E(\boldsymbol{\theta}^*)$ includes $q(\boldsymbol{x})$. Generally, there

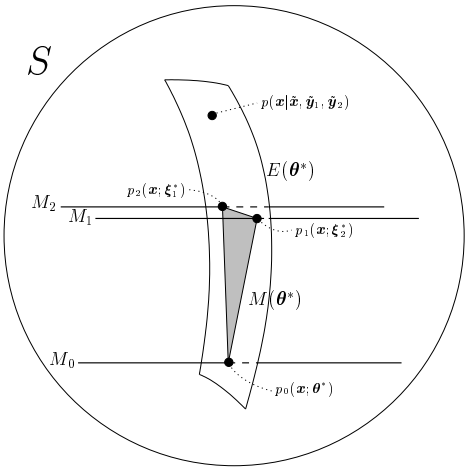

Figure 4: $M(\boldsymbol{\theta}^*)$ and $E(\boldsymbol{\theta}^*)$ of turbo decoding

is a discrepancy between $M(\boldsymbol{\theta}^*)$ and $E(\boldsymbol{\theta}^*)$. Therefore $q(\boldsymbol{x})$ is not necessarily included in $M(\boldsymbol{\theta}^*)$. The similar structure exists in some problems in statistical physics[2, 6, 10]

5. Information geometrical analysis

5.1. Local stability analysis

We show the condition for the equilibrium point to be stable. Equation (12) is rewritten as follows with $\boldsymbol{\eta}$,

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) = \boldsymbol{\eta}_1(\boldsymbol{\xi}_2^*) = \boldsymbol{\eta}_2(\boldsymbol{\xi}_1^*).$$

For the following discussion, we define the Fisher information matrices of the models. $G_0(\boldsymbol{\theta})$ is the Fisher information matrix of $p_0(\boldsymbol{x};\boldsymbol{\theta})$, and $G_r(\boldsymbol{\xi})$ is that of $p_r(\boldsymbol{x};\boldsymbol{\xi})$, $r = 1, 2$. Since the distributions are exponential family, we have the following relations,

$$G_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}\boldsymbol{\theta}'} \varphi_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} \boldsymbol{\eta}_0(\boldsymbol{\theta}), \\ G_r(\boldsymbol{\xi}) = \partial_{\boldsymbol{\xi}\boldsymbol{\xi}'} \varphi_r(\boldsymbol{\xi}) = \partial_{\boldsymbol{\xi}} \boldsymbol{\eta}_r(\boldsymbol{\xi}), \quad r = 1, 2.$$

Note that $G_0(\boldsymbol{\theta})$ is a diagonal matrix. In order to discuss the local stability property, we give a sufficiently small perturbation $\boldsymbol{\delta}$ to $\boldsymbol{\xi}_1^*$ and apply one step of the turbo decoding algorithm. Let $\boldsymbol{\xi}_1 = \boldsymbol{\xi}_1^* + \boldsymbol{\delta}$. Let $\boldsymbol{\theta} = \pi_{M_0} \circ p_2(\boldsymbol{x};\boldsymbol{\xi}_1^* + \boldsymbol{\delta})$, and the following relation is derived

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}) = \boldsymbol{\eta}_2(\boldsymbol{\xi}_1) \\ \boldsymbol{\eta}_0(\boldsymbol{\theta}^*) + G_0(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*) = \boldsymbol{\eta}_1(\boldsymbol{\xi}_2^*) + G_2(\boldsymbol{\xi}_1^*)\boldsymbol{\delta} \\ \boldsymbol{\theta} = \boldsymbol{\theta}^* + G_0(\boldsymbol{\theta}^*)^{-1} G_2(\boldsymbol{\xi}_1^*)\boldsymbol{\delta}.$$

Therefore, $\boldsymbol{\xi}_2$ in step 2 will be,

$$\boldsymbol{\xi}_2 = \boldsymbol{\xi}_2^* + \left(G_0(\boldsymbol{\theta}^*)^{-1} G_2(\boldsymbol{\xi}_1^*) - I_N \right) \boldsymbol{\delta}.$$

Here, I_N is an identity matrix of size N. Following the same calculation for step 3,

$$\boldsymbol{\xi}_1' = \boldsymbol{\xi}_1^* + \mathcal{T}\boldsymbol{\delta} \\ \mathcal{T} = (G_0(\boldsymbol{\theta}^*)^{-1} G_1(\boldsymbol{\xi}_2^*) - I_N)(G_0(\boldsymbol{\theta}^*)^{-1} G_2(\boldsymbol{\xi}_1^*) - I_N).$$

Original perturbation $\boldsymbol{\delta}$ is updated to $\mathcal{T}\boldsymbol{\delta}$, and the following theorem is derived. This result coincides with the result of Richardson[9].

Theorem 2. *Let λ_i, $i = 1, \ldots, N$ be the eigen values of the matrix $\mathcal{T}$. If $|\lambda_i| < 1$ holds for all i, the equilibrium point is stable.*

5.2. Cost function

In this subsection, we discuss a cost function which plays an important role for the turbo code.

Let $\boldsymbol{\theta} = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2$. We define the following function,

$$D[p_0(\boldsymbol{x};\boldsymbol{\theta}); q(\boldsymbol{x})] - D[p_0(\boldsymbol{x};\boldsymbol{\theta}); p_1(\boldsymbol{x};\boldsymbol{\xi}_2)] - D[p_0(\boldsymbol{x};\boldsymbol{\theta}); p_2(\boldsymbol{x};\boldsymbol{\xi}_1)]. \tag{14}$$

From simple calculation, we can show that eq.(14) is equivalent to $\mathcal{F}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$ by neglecting a constant value.

$$\mathcal{F}(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) = \varphi_0(\boldsymbol{\theta}) - (\varphi_1(\boldsymbol{\xi}_2) + \varphi_2(\boldsymbol{\xi}_1)). \tag{15}$$

This function coincides with the result of [5].

Theorem 3. *The equilibrium state $\boldsymbol{\xi}_1^*, \boldsymbol{\xi}_2^*$ is the critical point of $\mathcal{F}$.*

Proof. The derivative of $\mathcal{F}$ with respect to $\boldsymbol{\xi}_r$ $r = 1, 2$ is,

$$\partial_{\boldsymbol{\xi}_1}\mathcal{F} = \partial\varphi_0(\boldsymbol{\theta}) - \partial\varphi_2(\boldsymbol{\xi}_1) = \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_2(\boldsymbol{\xi}_1)$$
$$\partial_{\boldsymbol{\xi}_2}\mathcal{F} = \partial\varphi_0(\boldsymbol{\theta}) - \partial\varphi_1(\boldsymbol{\xi}_2) = \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_1(\boldsymbol{\xi}_2).$$

For the equilibrium, $\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) = \boldsymbol{\eta}_1(\boldsymbol{\xi}_2^*) = \boldsymbol{\eta}_2(\boldsymbol{\xi}_1^*)$ holds, and the proof is completed. □

Let us rewrite the result in the matrix form.

$$\begin{pmatrix}\partial_{\boldsymbol{\xi}_1}\mathcal{F}\\ \partial_{\boldsymbol{\xi}_2}\mathcal{F}\end{pmatrix} = \begin{pmatrix}\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_2(\boldsymbol{\xi}_1)\\ \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_1(\boldsymbol{\xi}_2)\end{pmatrix}$$

And we can understand the turbo code as follows.

- Step 2 makes $\partial_{\boldsymbol{\xi}_1}\mathcal{F} = 0$, by adjusting $\boldsymbol{\xi}_2$.
- Step 3 makes $\partial_{\boldsymbol{\xi}_2}\mathcal{F} = 0$, by adjusting $\boldsymbol{\xi}_1$.

When turbo code converges, following equation holds.

$$\partial_{\boldsymbol{\xi}_1}\mathcal{F} = \partial_{\boldsymbol{\xi}_2}\mathcal{F} = 0, \tag{16}$$

Equation (16) does not necessarily mean this is the minimum nor maximum of $\mathcal{F}$. Let us consider the convergence property. Suppose $(\boldsymbol{\xi}_2^{t+1} - \boldsymbol{\xi}_2^t)$ is small. Then, in step 2,

$$\boldsymbol{\eta}_0(\boldsymbol{\theta} + \delta\boldsymbol{\xi}_2) - \boldsymbol{\eta}_2(\boldsymbol{\xi}_1) = 0$$
$$\partial^2\varphi_0(\boldsymbol{\theta})\delta\boldsymbol{\xi}_2 \sim -(\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_2(\boldsymbol{\xi}_1))$$
$$\delta\boldsymbol{\xi}_2 \sim -G_0(\boldsymbol{\theta})^{-1}\partial_{\boldsymbol{\xi}_1}\mathcal{F}$$

And we have,

$$\begin{pmatrix}\delta\boldsymbol{\xi}_1\\ \delta\boldsymbol{\xi}_2\end{pmatrix} = \begin{pmatrix}O & G_0(\boldsymbol{\theta})^{-1}\\ G_0(\boldsymbol{\theta})^{-1} & O\end{pmatrix}\begin{pmatrix}\partial_{\boldsymbol{\xi}_1}\mathcal{F}\\ \partial_{\boldsymbol{\xi}_2}\mathcal{F}\end{pmatrix}.$$

This shows how the algorithm works, but it does not give the characteristics of the equilibrium point. The Hessian of $\mathcal{F}$ is,

$$\mathcal{H} = \begin{pmatrix}\partial_{\boldsymbol{\xi}_1\boldsymbol{\xi}_1}\mathcal{F} & \partial_{\boldsymbol{\xi}_1\boldsymbol{\xi}_2}\mathcal{F}\\ \partial_{\boldsymbol{\xi}_2\boldsymbol{\xi}_1}\mathcal{F} & \partial_{\boldsymbol{\xi}_2\boldsymbol{\xi}_2}\mathcal{F}\end{pmatrix} = \begin{pmatrix}G_0 - G_1 & G_0\\ G_0 & G_0 - G_2\end{pmatrix},$$

and let us transform the variables as,

$$\boldsymbol{\theta} = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2$$
$$\boldsymbol{\nu} = \boldsymbol{\xi}_1 - \boldsymbol{\xi}_2.$$

Then,

$$\begin{pmatrix}\partial_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{F} & \partial_{\boldsymbol{\theta}\boldsymbol{\nu}}\mathcal{F}\\ \partial_{\boldsymbol{\nu}\boldsymbol{\theta}}\mathcal{F} & \partial_{\boldsymbol{\nu}\boldsymbol{\nu}}\mathcal{F}\end{pmatrix} = \frac{1}{4}\begin{pmatrix}4G_0(\boldsymbol{\theta}) - (G_1 + G_2) & (G_1 - G_2)\\ (G_1 - G_2) & -(G_1 + G_2)\end{pmatrix}.$$

Most probably, $\partial_{\boldsymbol{\theta}\boldsymbol{\theta}}\mathcal{F}$ is positive definite but $\partial_{\boldsymbol{\nu}\boldsymbol{\nu}}\mathcal{F}$ is always negative, and $\mathcal{F}$ is generally saddle at the converged point.

5.3. Perturbation analysis

We have shown the information geometrical understanding of the decoding methods, and the condition of the equilibrium point to be stable. Now, we show some analysis of the approximation ability. For the following discussion, we define a distribution $p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})$ as

$$p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v}) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} + \boldsymbol{v}\cdot\boldsymbol{c}(\boldsymbol{x}) - \varphi(\boldsymbol{\theta},\boldsymbol{v}))$$
$$\varphi(\boldsymbol{\theta},\boldsymbol{v}) = \log\sum_{\boldsymbol{x}}\exp\left(c_0(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} + \boldsymbol{v}\cdot\boldsymbol{c}(\boldsymbol{x})\right),$$
$$\boldsymbol{c}(\boldsymbol{x}) \stackrel{\text{def}}{=} (c_1(\boldsymbol{x}), c_2(\boldsymbol{x}))^T.$$

Here $\boldsymbol{\theta} = (\theta^1, \cdots, \theta^N)^T \in \mathcal{R}^N$ and $\boldsymbol{v} = (v^1, v^2)^T \in \mathcal{R}^2$. This distribution includes $p_0(\boldsymbol{x};\boldsymbol{\theta})$ ($\boldsymbol{v} = \mathbf{o}$), $q(\boldsymbol{x})$ ($\boldsymbol{\theta} = \mathbf{o}$, $\boldsymbol{v} = \mathbf{1}$), and $p_r(\boldsymbol{x};\boldsymbol{\xi})$ ($\boldsymbol{\theta} = \boldsymbol{\xi}$, $\boldsymbol{v} = \boldsymbol{e}_r$), where $\mathbf{1} = (1,1)^T$, $\boldsymbol{e}_1 = (1,0)^T$, and $\boldsymbol{e}_2 = (0,1)^T$. The expectation parameter $\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v})$ is defined as,

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

Let us consider $M(\boldsymbol{\theta}^*)$, where every distribution $p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v}) \in M(\boldsymbol{\theta}^*)$ has the same expectation parameter, that is, $\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v}) = \boldsymbol{\eta}^*$ holds. Here, we define, $\boldsymbol{\eta}^* = \boldsymbol{\eta}(\boldsymbol{\theta}^*, \mathbf{o})$. From the Taylor expansion,

we have,

$$\eta_i(\boldsymbol{\theta},\boldsymbol{v}) = \eta_i^* + \sum_j \partial_j \eta_i^* \Delta\theta^j + \sum_r \partial_r \eta_i^* v^r + \frac{1}{2}\sum_{r,s} \partial_r \partial_s \eta_i^* v^r v^s + \sum_{j,r} \partial_r \partial_j \eta_i^* v^r \Delta\theta^j + \frac{1}{2}\sum_{k,l} \partial_k \partial_l \eta_i^* \Delta\theta^k \Delta\theta^l + O(\|\boldsymbol{v}\|^3) + O(\|\Delta\boldsymbol{\theta}\|^3). \tag{17}$$

The indexes $\{i,j,k,l\}$ are for $\boldsymbol{\theta}$, $\{r,s\}$ are for $\boldsymbol{v}$, and $\Delta\boldsymbol{\theta} \stackrel{\text{def}}{=} \boldsymbol{\theta} - \boldsymbol{\theta}^*$. After adding some definitions, that is, $\eta_i(\boldsymbol{\theta},\boldsymbol{v}) = \eta_i^*$, and $\partial_j \eta_i^* = g_{ij}(\boldsymbol{\theta}^*)$, where $\{g_{ij}\}$ is the Fisher information matrix of $p(\boldsymbol{x};\boldsymbol{\theta}^*,\mathbf{o})$ which is a diagonal matrix, we substitute $\Delta\theta^i$ with function of v^r up to its 2nd order, and neglect the higher orders of v^r. And we have,

$$\Delta\theta^i \simeq -g^{ii}\sum_r A_r^i v^r - \frac{g^{ii}}{2} \times \sum_{r,s}\Big(\partial_r - \sum_k g^{kk} A_r^k \partial_k\Big)\Big(\partial_s - \sum_j g^{jj} A_s^j \partial_j\Big)\eta_i^* v^r v^s, \tag{18}$$

where, $g^{ii} = 1/g_{ii}$, and $A_r^i = \partial_r \eta_i^*$. Let us consider, $p(\boldsymbol{x};\boldsymbol{\theta}^*,\mathbf{o})$, $p(\boldsymbol{x};\boldsymbol{\xi}_1,\delta\boldsymbol{e}_2)$, $p(\boldsymbol{x};\boldsymbol{\xi}_2,\delta\boldsymbol{e}_1)$, and $p(\boldsymbol{x};\mathbf{o},\delta\mathbf{1})$. Let $p(\boldsymbol{x};\boldsymbol{\theta}^*,\mathbf{o})$, $p(\boldsymbol{x};\boldsymbol{\xi}_1,\delta\boldsymbol{e}_2)$, and $p(\boldsymbol{x};\boldsymbol{\xi}_2,\delta\boldsymbol{e}_1)$, be included in $M(\boldsymbol{\theta}^*)$, and the following relation be satisfied

$$\boldsymbol{\theta}^* = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2.$$

By putting $\delta = 1$, $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ converge to the equilibrium point of the turbo decoding. From the result of eq.(17) and eq.(18), we have the following theorem.

Theorem 4. *The true expectation of $\boldsymbol{x}$, which is equivalent to $\boldsymbol{\eta}(\mathbf{o},\mathbf{1})$, is approximated as,*

$$\boldsymbol{\eta}(\mathbf{o},\mathbf{1}) \simeq \boldsymbol{\eta}^* + \frac{1}{2}\sum_{r\neq s}\Big(\partial_r - \sum_k g^{kk} A_r^k \partial_k\Big)\Big(\partial_s - \sum_j g^{jj} A_s^j \partial_j\Big)\boldsymbol{\eta}^*. \tag{19}$$

Where $\boldsymbol{\eta}^*$ is the solution of the turbo decoding.

This theorem gives the decoding error of the turbo code. Equation (19) is related to the m–embedded–curvature of $E(\boldsymbol{\theta}^*)$ (Fig.4).

6. Discussion

We have studied the mechanism of the turbo decoding from information geometrical viewpoint. It gives the intuitive understanding of the algorithm, and the framework for analysis. The structure of the equilibrium of the turbo decoding is summarized in Theorem 1. A set of m–flat and e–flat submanifolds defined on the same set of probability distributions, plays an important role, and the discrepancy between the two submanifolds gives the decoding error, which is shown in Theorem 4. The cost function in eq.(15) is also a new result. It revealed the dynamics of the algorithm, and showed that the equilibrium is generally a suddle point.

We have shown these results for the case of the turbo decoding, but the results are general. Therefore, they are valid for other iterative methods including the Gallager code and BP for loopy BN.

Finally, we note that this article gives the first step to the information geometrical understanding of the belief propagation decoders. We believe further study in this direction will lead us to better understanding of these methods.

References

[1] S. Amari. *Differential-Geometrical Methods in Statistics*, volume 28 of *Lecture Notes in Statistics*. Springer-Verlag, Berlin, 1985.

[2] S. Amari, S. Ikeda, and H. Shimokawa. Information geometry and mean field approximation: The α-projection approach. In M. Opper and D. Saad, editors, *Advanced Mean Field Methods – Theory and Practice*, chapter 16, pages 241–257. MIT Press, 2001.

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

[4] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: Turbo-codes. *IEEE Transactions on Communications*, 44(10):1261–1271, October 1996.

[5] Y. Kabashima and D. Saad. The TAP approach to intensive and extensive connectivity systems. In M. Opper and D. Saad, editors, *Adavanced Mean Field Theory – Theory and Practice*, chapter 6, pages 65–84. MIT Press, 2001.

[6] H. J. Kappen and W. J. Wiegerinck. Mean field theory for graphical models. In M. Opper and D. Saad, editors, *Adavanced Mean Field Theory – Theory and Practice*, chapter 4, pages 37–49. MIT Press, 2001.

[7] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng. Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE Journal on Selected Areas in Communications*, 16(2):140–152, February 1998.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo. CA: Morgan Kaufmann, 1988.

[9] T. Richardson. The geometry of turbo-decoding dynamics. *IEEE Transactions on Information Theory*, 46(1):9–23, January 2000.

[10] T. Tanaka. Information geometry of mean-field approximation. *Neural Computation*, 12(8):1951–1968, August 2000.