Research Memorandum No. 890 September 4, 2003

Stochastic reasoning, free energy and information geometry

Shiro Ikeda
Toshiyuki Tanaka
and
Shun-ichi Amari

Stochastic Reasoning, Free Energy and Information Geometry

Shiro Ikeda
shiro@ism.ac.jp
Institute of Statistical Mathematics
Minato-ku, Tokyo, 106-8569 Japan*

Toshiyuki Tanaka
tanaka@eei.metro-u.ac.jp
Tokyo Metropolitan University
Hachioji, Tokyo, 192-0397 Japan

Shun-ichi Amari
amari@brain.riken.go.jp
RIKEN Brain Science Institute
Wako-shi, Saitama, 351-0198 Japan

Abstract

Belief propagation is a universal method of stochastic reasoning. It gives a good approximate solution, when it is applied to a stochastic model with loopy interactions. AI, statistical physical, and information geometrical methods have so far been used to analyze its performance separately. The present paper gives a unified framework to understand the relation underlying these concepts. In particular, the free energy and its relation to BP and CCCP is elucidated from the point of view of information geometry. We then propose a family of new algorithms. The stability of the algorithms is also analyzed, and methods of accelerating these algorithms are proposed.

key words: Belief propagation; Information geometry; Concave-Convex Procedure; Natural gradient

1 Introduction

Stochastic reasoning is a technique used in wide areas of AI, statistics, neural networks, and others, to estimate the values of random variables based on partially observed variables [Pearl, 1988]. Here, a large number of mutually interacting random variables are represented in the form of joint probability. However, these interactions often have specific structure such that some variables are independent of others when a set of variables are fixed. In other words, they

*This work is done during his long term visit in Gatsby Computational Neuroscience Unit, UCL

are conditionally independent, and their interactions take place only through these conditioning variables. When such structure is represented by a graph, it is called a graphical model [Jordan, 1999, Lauritzen and Spiegelhalter, 1988]. The problem is to infer the values of unobserved variables based on observed ones, by reducing the conditional joint probability distribution to the marginal probability distributions.

When the random variables are binary, their marginal probabilities are determined by the (conditional) expectation. Hence, the problem reduces to calculation of the conditional expectation. However, when the number of the random variables is large, it is computationally intractable to carry out calculations from the definition. Statistical physics has used the mean field approximation and its modifications [Opper and Saad, 2001]. When the underlying causal graphical structure does not include loops, belief propagation (BP) [Pearl, 1988] proposed in AI, gives a correct answer. It is also applied to loopy graphical models, giving good approximate solutions.

These techniques are used in the decoding of the turbo codes, low-density parity-check (LDPC) codes as well as spin glass and Boltzmann machines. There are a number of theoretical approaches to analyze their performances. The statistical physical framework uses the Bethé free energy [Yedidia et al., 2001a] or the like [Kabashima and Saad, 1999, Kabashima and Saad, 2001] to analyze these methods. A geometrical theory was initiated by [Richardson, 2000] to elucidate the turbo decoding. Information geometry [Amari and Nagaoka, 2000] gives a framework to elucidate the method of belief propagation [Ikeda et al., 2003, Ikeda et al., 2002], which is also used in the studying the mean field approximation [Tanaka, 2000, Tanaka, 2001, Amari et al., 2001]. The TRP (tree reparameterization) [Wainwright et al., 2002] also uses information geometry. Recently an approach named CCCP (convex concave computational procedure) was proposed [Yuille and Rangarajan, 2002] and applied to solve the minimization problem of the Bethé free energy, thereby providing an alternative to belief propagation, the CCCP-Bethé (which, in the following, we will simply call the CCCP algorithm).

The problem is interdisciplinary, where various concepts and frameworks originate from statistics, AI, statistical physics and information geometry. The present paper gives a unified framework, based on information geometry, to understand the role of the free energy, belief propagation, CCCP and their variants. To this end, we propose a new function of the free-energy type, to which the Bethé free energy [Yedidia et al., 2001a] and that of [Kabashima and Saad, 2001] are closely related. By constraining the search space in proper ways, we obtain a family of algorithms, including BP, CCCP or CCCP without double loops. We also give their stability analysis. The error analysis was given in another paper [Ikeda et al., 2003].

The paper is organized as follows. In section 2, the problem is stated compactly followed by preliminary of information geometry. Section 3 introduces information geometrical view of BP, the characteristics of equilibrium, and related algorithms, TRP and CCCP. We discuss the free energy which is related to BP in section 4, and a new algorithm is proposed with stability analysis of the algorithms in section 5. Section 6 gives some extensions of BP from information geometrical viewpoint, and finally section 7 concludes the paper.

2 Problem and Geometrical Framework

2.1 Basic Problem and Strategy

Let $\boldsymbol{x} = (x_1, \cdots, x_n)^T$ be hidden and $\boldsymbol{y} = (y_1, \cdots, y_m)^T$ be observed random variables. We start with the case where each x_i is binary i.e., $x_i \in \{-1, +1\}$ for simplicity. An extension to wider class of distributions will be given in section 6.1.

The conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ is written as $q(\boldsymbol{x}|\boldsymbol{y})$, and our task is to give a good inference of $\boldsymbol{x}$ from the observations. We hereafter simply write $q(\boldsymbol{x})$ for $q(\boldsymbol{x}|\boldsymbol{y})$ and omit $\boldsymbol{y}$.

One natural inference of $\boldsymbol{x}$ is the MAP (maximum a posteriori), that is

$$\hat{\boldsymbol{x}}_{map} \stackrel{\text{def}}{=} \operatorname*{argmax}_{\boldsymbol{x}} q(\boldsymbol{x}).$$

This minimizes the error probability that $\hat{\boldsymbol{x}}_{map}$ does not coincide with the true one. However, this calculation is not tractable when n is large because the number of candidates of $\boldsymbol{x}$ increases exponentially with respect to n. The MPM (maximization of the posterior marginals) is another inference that minimizes the number of component errors. If each marginal distribution $q(x_i)$, $i = 1, \cdots, n$, is known, the MPM inference decides $\hat{x}_i = +1$ when $q(x_i = +1) \geq q(x_i = -1)$ and $\hat{x}_i = -1$ otherwise. Let η_i be the expectation of x_i with respect to $q(\boldsymbol{x})$, that is

$$\eta_i \stackrel{\text{def}}{=} E_q[x_i] = \sum_{x_i} x_i q(x_i).$$

The MPM inference is equivalent to $\hat{x}_i = \text{sgn}\ \eta_i$, which is directly calculated if we know the marginal distributions $q(x_i)$, or the expectation

$$\overline{\boldsymbol{x}} = E_q[\boldsymbol{x}].$$

The present paper focuses on the method to obtain a good approximation to $\overline{\boldsymbol{x}}$, which is equivalent to the inference of $\prod_{i=1}^n q(x_i)$.

For any $q(\boldsymbol{x})$, $\ln q(\boldsymbol{x})$ can be expanded as a polynomial of $\boldsymbol{x}$ up to degree n, because every x_i is binary. However, in many problems, mutual interactions of random variables exist only in specific manners. We represent $\ln q(\boldsymbol{x})$ in the form

$$\ln q(\boldsymbol{x}) = \boldsymbol{h} \cdot \boldsymbol{x} + \sum_{r=1}^{L} c_r(\boldsymbol{x}) - \psi_q,$$

where $\boldsymbol{h} \cdot \boldsymbol{x} = \sum_i h_i x_i$ is the linear term, $c_r(\boldsymbol{x}), r = 1, \cdots, L$, is a simple polynomial representing the r-th clique among related variables, and ψ_q is a normalizing factor which is called the (Helmholtz) free energy,

$$\psi_q = \ln \sum_{\boldsymbol{x}} \exp\Big(\boldsymbol{h} \cdot \boldsymbol{x} + \sum_r c_r(\boldsymbol{x})\Big).$$

In the case of a Boltzmann machine (Fig. 1) or of a spin glass model, $c_r(\boldsymbol{x})$ is a quadratic function of x_i, that is,

$$c_r(\boldsymbol{x}) = w_{ij}^r x_i x_j,$$

where r is the index of the mutual coupling between x_i and x_j. In an undirected graphical model, it corresponds to an edge of the graph.

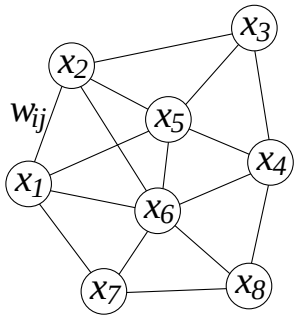

Figure 1: The Boltzmann Machine.

2.2 Important Family of Distributions

Let us consider the set of probability distributions

$$p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{x} + \boldsymbol{v} \cdot \boldsymbol{c}(\boldsymbol{x}) - \psi(\boldsymbol{\theta}, \boldsymbol{v})) \tag{1}$$

parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{v}$, where $\boldsymbol{v} = (v_1, \cdots, v_L)^T$, $\boldsymbol{c}(\boldsymbol{x}) = (c_1(\boldsymbol{x}), \cdots, c_L(\boldsymbol{x}))^T$ and $\boldsymbol{v} \cdot \boldsymbol{c}(\boldsymbol{x}) = \sum_{r=1}^{L} v_r c_r(\boldsymbol{x})$. We name the family of the probability distributions S, which is an exponential family

$$S = \Big\{ p(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{v}) \;\Big|\; \boldsymbol{\theta} \in \mathcal{R}^n, \boldsymbol{v} \in \mathcal{R}^L \Big\}, \tag{2}$$

where its canonical coordinate system is $(\boldsymbol{\theta}, \boldsymbol{v})$. The joint distribution $q(\boldsymbol{x})$ is included in S, which is easily proved by setting $\boldsymbol{\theta} = \boldsymbol{h}$ and $\boldsymbol{v} = \mathbf{1}_L = (1, \cdots, 1)^T$,

$$q(\boldsymbol{x}) = p(\boldsymbol{x}; \boldsymbol{h}, \mathbf{1}_L).$$

The submanifold specified by $\boldsymbol{v} = \mathbf{o}$,

$$M_0 = \left\{ p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \exp(\boldsymbol{h} \cdot \boldsymbol{x} + \boldsymbol{\theta} \cdot \boldsymbol{x} - \psi_0(\boldsymbol{\theta})) \mid \boldsymbol{\theta} \in \mathcal{R}^n \right\},$$

consists of all the independent distributions which include no mutual interactions between x_i and x_j $(i \neq j)$, and its coordinate system is $\boldsymbol{\theta}$. Apparently $M_0 \subset S$, and the product of marginal distributions of $q(\boldsymbol{x})$, that is, $\prod_{i=1}^n q(x_i) \in M_0$ is included in M_0. The ultimate goal is to derive $\prod_{i=1}^n q(x_i)$ or corresponding coordinate $\boldsymbol{\theta}$ of M_0.

2.3 Preliminary of Information Geometry

We give preliminary of information geometry [Amari and Nagaoka, 2000, Amari, 2001] in this subsection. First we define e–flat and m–flat submanifolds of S.

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

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

where $c(t)$ is the normalization factor. Obviously, $\{r(\boldsymbol{x}; t) | t \in [0,1]\}$ is an exponential family connecting two distributions, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. When an e–flat submanifold is a one–dimensional curve, it is called an e–geodesic. In terms of the e–affine coordinates, $\boldsymbol{\theta}$, a submanifold M is e–flat when it is linear in $\boldsymbol{\theta}$.

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

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

When an m–flat submanifold is a one–dimensional curve, it is called an m–geodesic. Hence, the above mixture family is the m–geodesic connecting them.

From the definition, any exponential family is an e–flat manifold. Therefore S and M_0 are e–flat. Next we define the m–projection [Amari and Nagaoka, 2000].

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

$$\mathit{\Pi}_M \circ q(\boldsymbol{x}) = \operatorname*{argmin}_{p(\boldsymbol{x}) \in M} D[q(\boldsymbol{x}); p(\boldsymbol{x})]$$

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

Here, $D[\cdot;\cdot]$ is the KL (Kullback-Leibler)-divergence defined as

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

The KL–divergence satisfies $D[q(\boldsymbol{x}); p(\boldsymbol{x})] \geq 0$, and $D[q(\boldsymbol{x}); p(\boldsymbol{x})] = 0$ when and only when $q(\boldsymbol{x}) = p(\boldsymbol{x})$ holds for every $\boldsymbol{x}$. Although symmetry $D[q; p] = D[p; q]$ does not hold generally, it is regarded as an asymmetric squared distance. Finally, the m–projection theorem follows.

Theorem 1. *Let M be an e–flat submanifold in S, and let $q(\boldsymbol{x}) \in S$. The m–projection of $q(\boldsymbol{x})$ to M is unique and given by a point in M such that the m–geodesic connecting $q(\boldsymbol{x})$ and $\Pi_M \circ q$ is orthogonal to M at this point in the sense of the Riemannian metric due to the Fisher information matrix.*

2.4 MPM Inference

We show the idea of MPM inference is equivalent to the m–projection from $q(\boldsymbol{x})$ to M_0. From the definition, the m–projection of $q(\boldsymbol{x})$ to M_0 is characterized by $\boldsymbol{\theta}^*$, that satisfies

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

We hereafter denote the m–projection to M_0 as,

$$\boldsymbol{\theta} = \Pi_{M_0} \circ q(\boldsymbol{x}) \stackrel{\text{def}}{=} \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{R}^n} D[q(\boldsymbol{x}); p_0(\boldsymbol{x}; \boldsymbol{\theta})]$$

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

$$\sum_{\boldsymbol{x}} q(\boldsymbol{x}) \boldsymbol{x} - \partial_{\boldsymbol{\theta}} \psi_0(\boldsymbol{\theta}^*) = \mathbf{o}, \tag{3}$$

where $\partial_{\boldsymbol{\theta}}$ shows the derivative with respect to $\boldsymbol{\theta}$. From the definition of exponential family,

$$\partial_{\boldsymbol{\theta}} \psi_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} \ln \sum_{\boldsymbol{x}} \exp(\boldsymbol{h} \cdot \boldsymbol{x} + \boldsymbol{\theta} \cdot \boldsymbol{x}) = \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta}). \tag{4}$$

We define the new parameter $\boldsymbol{\eta}_0(\boldsymbol{\theta})$ in M_0 as

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}} \psi_0(\boldsymbol{\theta}). \tag{5}$$

This is called the expectation parameter [Amari and Nagaoka, 2000]. From eqs. (3), (4), and (5), the m–projection is equivalent to the marginalization of $q(\boldsymbol{x})$.

We have shown that the ultimate goal of the problem is the m–projection of $q(\boldsymbol{x})$ to M_0. In the next section, we show how the belief propagation algorithm approximates the m–projection to M_0.

3 BP and Variants: Information Geometrical View

3.1 Belief Propagation Algorithm

Information Geometrical View of BP algorithm

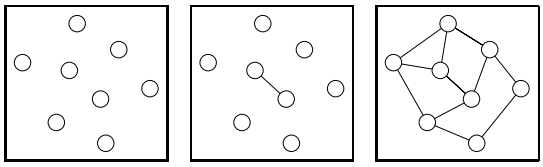

Figure 2: A: Belief graph, B: Graph with a single edge, C: Graph with all the edges.

In this subsection, we give the information geometrical view of the BP algorithm. The well-known definition of the BP algorithm is found in [Pearl, 1988, Lauritzen and Spiegelhalter, 1988, Weiss, 2000], and the detail is not given in this paper. We note that our derivation is based on the BP algorithm for undirected graphs. It is well-known that, for loopy graphs, the BP algorithm does not necessarily converge, and even if it converges, the result is not exactly equal to the true MPM inference.

Figure 2 shows three important graphs for the belief propagation algorithm. The belief graph in Fig. 2.A corresponds to $p_0(\boldsymbol{x};\boldsymbol{\theta})$, and that in Fig. 2.C corresponds to the true distribution $q(\boldsymbol{x})$. Figure 2.B shows an important distribution, which includes only a single edge. This distribution is defined as $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, where

$$p_r(\boldsymbol{x};\boldsymbol{\zeta}_r) = \exp\Big(\boldsymbol{h}\cdot\boldsymbol{x} + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r\cdot\boldsymbol{x} - \psi_r(\boldsymbol{\zeta}_r)\Big), \quad r = 1,\cdots,L.$$

This can be generalized, without any change, to the case when $c_r(\boldsymbol{x})$ is a polynomial. Here, r is the index of edges or polynomials, and L is the number of the edges in the graph or such higher interactions. The set of the distributions $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$ parameterized by $\boldsymbol{\zeta}_r$ is an e–flat manifold defined as

$$M_r = \Big\{p_r(\boldsymbol{x};\boldsymbol{\zeta}_r) \;\Big|\; \boldsymbol{\zeta}_r \in \mathcal{R}^n\Big\}, \quad r = 1,\cdots,L.$$

Its coordinates are $\boldsymbol{\zeta}_r$. We also define the expectation parameter $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)$ of M_r,

$$\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) \stackrel{\text{def}}{=} \partial_{\boldsymbol{\zeta}_r}\psi_r(\boldsymbol{\zeta}_r) = \sum_{\boldsymbol{x}} \boldsymbol{x} p_r(\boldsymbol{x};\boldsymbol{\zeta}_r).$$

In M_r, only the r-th edge is taken into account but all the other edges are replaced by a linear term $\boldsymbol{\zeta}_r\cdot\boldsymbol{x}$, and $p_0(\boldsymbol{x};\boldsymbol{\theta}) \in M_0$ is used to integrate all the information from $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r), r = 1,\cdots,L$, giving $\boldsymbol{\theta}$, which is the parameter of $p_0(\boldsymbol{x};\boldsymbol{\theta})$ to infer $\prod_i q(x_i)$. The BP algorithm iteratively

modifies $\{\boldsymbol{\zeta}_r\}$ of $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r), r = 1, \cdots, L$, by using the integrated information $\boldsymbol{\theta}$, which in turn is renewed by and integrating local information $\{\boldsymbol{\zeta}_r\}$. Information geometry has elucidated its geometrical meaning for special graphs for error correcting codes([Ikeda et al., 2003], see also [Richardson, 2000]), and we give the framework for general graphs in the following.

The BP algorithm is stated as follows: Let $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^t)$ be the approximation to $q(\boldsymbol{x})$ at time t, which each M_r, $r = 1, \cdots, L$, specifies.

1. Set $t = 0$, $\boldsymbol{\zeta}_r^t = \mathbf{o}$, $r = 1, \cdots, L$.

2. Increment t by one and set $\boldsymbol{\xi}_r^t$, $r = 1, \cdots, L$ as follows,

$$\boldsymbol{\xi}_r^t = \Pi_{M_0} \circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^t) - \boldsymbol{\zeta}_r^t.$$

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

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

4. Repeat steps 2 and 3 until convergence.

The BP algorithm is summarized as follows: Calculate iteratively

$$\boldsymbol{\theta}^{t+1} = \sum_r (\Pi_{M_0} \circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^t) - \boldsymbol{\zeta}_r^t),$$
$$\boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\theta}^{t+1} - (\Pi_{M_0} \circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^t) - \boldsymbol{\zeta}_r^t) \quad r = 1, \cdots, L.$$

We have introduced two sets of parameters $\{\boldsymbol{\xi}_r\}$ and $\{\boldsymbol{\zeta}_r\}$. Let the converged point of the BP algorithm be $\{\boldsymbol{\xi}_r^*\}$, $\{\boldsymbol{\zeta}_r^*\}$ and $\boldsymbol{\theta}^*$, where $\boldsymbol{\theta}^* = \sum_r \boldsymbol{\xi}_r^* = \sum_r \boldsymbol{\zeta}_r^*/(L-1)$, and $\boldsymbol{\theta}^* = \boldsymbol{\xi}_r^* + \boldsymbol{\zeta}_r^*$. The probability distribution of $q(\boldsymbol{x})$, and its final approximation $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ in M_0, and approximation $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^*)$ in M_r are described as

$$\begin{aligned} q(\boldsymbol{x}) \quad &= \exp(\boldsymbol{h}\cdot\boldsymbol{x} + c_1(\boldsymbol{x}) + \cdots + c_L(\boldsymbol{x}) - \quad \psi_q) \\ p_0(\boldsymbol{x};\boldsymbol{\theta}^*) &= \exp(\boldsymbol{h}\cdot\boldsymbol{x} + \boldsymbol{\xi}_1^*\cdot\boldsymbol{x} + \cdots + \boldsymbol{\xi}_L^*\cdot\boldsymbol{x} - \psi_0(\boldsymbol{\theta}^*)) \\ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^*) &= \exp(\boldsymbol{h}\cdot\boldsymbol{x} + \boldsymbol{\xi}_1^*\cdot\boldsymbol{x} + \cdots + c_r(\boldsymbol{x}) + \quad \cdots \quad + \boldsymbol{\xi}_L^*\cdot\boldsymbol{x} - \psi_r(\boldsymbol{\zeta}_r^*)). \end{aligned}$$

The idea of BP is to approximate $c_r(\boldsymbol{x})$ by $\boldsymbol{\xi}_r^*\cdot\boldsymbol{x}$ in M_r, taking the information from $M_{r'}$ $(r' \neq r)$ into account. The independent distribution $p_0(\boldsymbol{x};\boldsymbol{\theta})$ integrate all the information.

Equilibrium of BP Algorithm

The following theorem proved in [Ikeda et al., 2003] characterizes the equilibrium point of the BP algorithm.

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

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

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

In order to have an information geometrical view, we define two submanifolds M^* and E^* as follows,

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

$$E^* = \left\{ p(\boldsymbol{x}) = C p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)^{t_0} \prod_{r=1}^{L} p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)^{t_r} \;\middle|\; \sum_{r=0}^{L} t_r = 1 \right\},\ C : \text{normalization factor.}$$

Note that M^* and E^* are an m–flat and an e–flat submanifold, respectively.

The geometrical implications of these conditions are as follows:

m–**condition:** The m–flat submanifold M^* which includes $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, $r = 1, \cdots, L$, and $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ is orthogonal to M_r, $r = 1, \cdots, L$ and M_0, that is, they are the m–projections to each other.

e–**condition:** The e–flat submanifold M^* which includes $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, $r = 1, \cdots, L$, and $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ also includes $q(\boldsymbol{x})$.

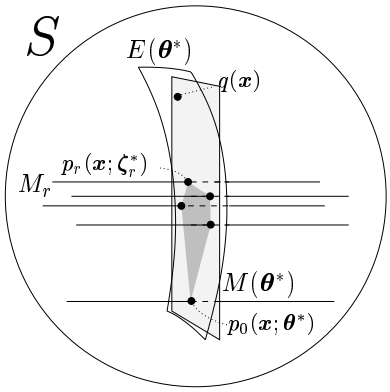

Figure 3: Structure of Equilibrium

The m–condition not only implies that the m–projection of $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$ to M_0 is $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$, but also that the m–projection of $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ to M_r is $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, that is,

$$\boldsymbol{\zeta}_r^* = \Pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^*), \quad r = 1, \cdots, L. \tag{6}$$

where Π_{M_r} denotes the m–projection to M_r. In other words, the m–condition is described as $\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) = \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r^*)$ $r = 1, \cdots, L$. If $q(\boldsymbol{x}) \in M^*$ holds, $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$ is the true marginalization, and $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*) = \prod_{i=1}^n q(x_i)$ holds, but this does not happen generally because there is a discrepancy between M^* and E^*, which is shown schematically in Fig. 3.

It is well-known that in the graphs with tree structures, the BP algorithm gives the true marginalization, that is, $q(\boldsymbol{x}) \in M^*$ holds. In this case, we have the following relation

$$q(\boldsymbol{x}) = \frac{\prod_{r=1}^L p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)}{p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)^{L-1}}.$$

This relationship gives the following proposition.

Proposition 1. *When $q(\boldsymbol{x})$ is represented in a tree graph, $q(\boldsymbol{x})$, $p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$, and $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$, $r = 1, \cdots, L$ are included in M^* and E^* simultaneously.*

3.2 TRP

There have been proposed some variants of the BP algorithm, and information geometry gives a general framework to understand them. We begin with TRP [Wainwright et al., 2002]. TRP selects the set of trees $\{\mathcal{T}_i\}$, where each tree $\mathcal{T}_i$ consists of a set of edges, and renew related parameters in the process of inference. Let the set of edges be $\mathcal{L}$ and $\mathcal{T}_i \subset \mathcal{L}$, $i = 1, \cdots, K$ be its subsets where each graph with the edges $\mathcal{T}_i$ does not have any loop. The choice of the sets $\{\mathcal{T}_i\}$ is arbitrary, but every edge must be included in at least in one of the trees.

In order to give the information geometrical view, we use the parameters $\boldsymbol{\zeta}_r$, $\boldsymbol{\theta}_r$, $r = 1, \cdots, L$, and $\boldsymbol{\theta}$. The information geometrical view of TRP is given as follows,

1. Set $\boldsymbol{\zeta}_r^0 = \boldsymbol{\theta}_r^0 = \mathbf{o}$, and $\boldsymbol{\theta}^0 = \mathbf{o}$.

2. For a tree $\mathcal{T}_i$, construct a tree distribution $p_{\mathcal{T}_i}^t(\boldsymbol{x})$ as follows

$$\begin{aligned} p_{\mathcal{T}_i}^t(\boldsymbol{x}) &= C p_0(\boldsymbol{x}; \boldsymbol{\theta}^t) \prod_{r \in \mathcal{T}_i} \frac{p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t)}{p_0(\boldsymbol{x}; \boldsymbol{\theta}_r^t)} \\ &= C' \exp\Big(\boldsymbol{h} \cdot \boldsymbol{x} + \sum_{r \in \mathcal{T}_i} c_r(\boldsymbol{x}) + \big(\sum_{r \in \mathcal{T}_i} (\boldsymbol{\zeta}_r^t - \boldsymbol{\theta}_r^t) + \boldsymbol{\theta}^t\big) \cdot \boldsymbol{x} \Big). \end{aligned}$$

   Let the m–projection of $p_{\mathcal{T}_i}^t(\boldsymbol{x})$ to M_0 be $\boldsymbol{\theta}^{t+1}$, the m–projection of $p_{\mathcal{T}_i}^t(\boldsymbol{x})$ to M_r be $\boldsymbol{\zeta}_r^{t+1}$, and $\boldsymbol{\theta}_r^{t+1} = \boldsymbol{\theta}^{t+1}$ where $r \in \mathcal{T}_i$.

$$\begin{aligned} \boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}_r^{t+1} &= \Pi_{M_0} \circ p_{\mathcal{T}_i}^t(\boldsymbol{x}) \\ \boldsymbol{\zeta}_r^{t+1} &= \Pi_{M_r} \circ p_{\mathcal{T}_i}^t(\boldsymbol{x}). \end{aligned}$$

   For $r \notin \mathcal{T}_i$, $\boldsymbol{\theta}_r^{t+1} = \boldsymbol{\theta}_r^t$ and $\boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\zeta}_r^t$.

3. Repeat step 2 for trees $\mathcal{T}_j \in \{\mathcal{T}_i\}$.

4. Repeat steps 2 and 3 until $\boldsymbol{\theta}_r^{t+1} = \boldsymbol{\theta}^{t+1}$, holds for every r, and $\{\boldsymbol{\zeta}_r^{t+1}\}$ converges.

Let us show that the e– and m–conditions are satisfied at the equilibrium of TRP. From the fact that $p_{\mathcal{T}_i}^t(\boldsymbol{x})$ is a tree graph, Proposition 1 gives the following relation,

$$\sum_{r\in\mathcal{T}_i}(\boldsymbol{\zeta}_r^{t+1} - \boldsymbol{\theta}_r^{t+1}) + \boldsymbol{\theta}^{t+1} = \sum_{r\in\mathcal{T}_i}(\boldsymbol{\zeta}_r^{t} - \boldsymbol{\theta}_r^{t}) + \boldsymbol{\theta}^{t}.$$

Since $\sum_{r\notin\mathcal{T}_i}(\boldsymbol{\zeta}_r^t - \boldsymbol{\theta}_r^t)$ does not change through step 2, we have the following relation, which shows the e–condition holds for the convergent point of TRP,

$$\sum_r \boldsymbol{\zeta}_r^* - (L-1)\boldsymbol{\theta}^* = \sum_r(\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}_r^*) + \boldsymbol{\theta}^* = \sum_r(\boldsymbol{\zeta}_r^t - \boldsymbol{\theta}_r^t) + \boldsymbol{\theta}^t = \mathbf{o}.$$

Moreover, from (6) we observe

$$\boldsymbol{\zeta}_r^* = \mathit{\Pi}_{M_r}\circ p_0(\boldsymbol{x};\boldsymbol{\theta}_r^*), \quad r = 1,\cdots,L,$$

which implies the m–condition. From the results, both of the e– and m–conditions are satisfied at the convergent point.

3.3 CCCP

CCCP (Concave-Convex Procedure) is an iterative procedure to obtain the minimum of a function, represented by the difference of two convex functions [Yuille and Rangarajan, 2003]. The algorithm is applied to solve the inference problem of loopy graphs [Yuille, 2002]. It consists of the inner loop and outer loop:

inner loop: Given $\boldsymbol{\theta}^t$, calculate $\{\boldsymbol{\zeta}_r^{t+1}\}$ by solving

$$\mathit{\Pi}_{M_0}\circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^{t+1}) = L\boldsymbol{\theta}^t - \sum_r \boldsymbol{\zeta}_r^{t+1} \quad r = 1,\cdots,L. \tag{7}$$

outer loop: Given a set of $\{\boldsymbol{\zeta}_r^{t+1}\}$ as the result of the inner loop, calculate

$$\boldsymbol{\theta}^{t+1} = L\boldsymbol{\theta}^t - \sum_r \boldsymbol{\zeta}_r^{t+1}. \tag{8}$$

Here, the inner loop solves a simultaneous equation with respect to $\{\boldsymbol{\zeta}_r^{t+1}\}$ and an iterative procedure is given for solving it.

From eqs. (7) and (8), one obtains

$$\boldsymbol{\theta}^{t+1} = \mathit{\Pi}_{M_0}\circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^{t+1}), \quad r = 1,\cdots,L,$$

which means that the CCCP algorithm enforces the m–condition at each iteration. On the other hand, the e–condition is satisfied only at the convergent point, which can be easily verified by letting $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t = \boldsymbol{\theta}^*$ in eq. (8) to yield the e–condition $(L-1)\boldsymbol{\theta}^* = \sum_r \boldsymbol{\zeta}_r^*$. One can therefore regard that the inner and outer loops of the CCCP algorithm solve the m–condition and the e–condition, respectively.

4 Free Energy Function

4.1 Bethé Free Energy

We have described the information geometrical view of BP and related algorithms. It gives the characteristics of the equilibrium point, but it is not enough to describe the approximation accuracy, and the dynamics of the algorithm.

An energy function helps us to clarify them, and there are some functions proposed for this purpose. Most popular function is the Bethé free energy. The Bethé free energy itself has been well known in the literature of statistical mechanics, being used in formulating the so-called Bethé approximation [Itzykson and Drouffe, 1989]. As far as we know, Kabashima and Saad [Kabashima and Saad, 2001] were the first to point out that BP is derived by considering a variational extremization of a free energy. It was Yedidia et al.[Yedidia et al., 2001b] who introduced to the machine-learning community the formulation of BP based on the Bethé free energy. Following [Yedidia et al., 2001b] and using their terminology, the definition of the free energy is given as follows,

$$\mathcal{F}_\beta = \sum_{r\in\mathcal{L}} \sum_{\boldsymbol{x}_r} b_r(\boldsymbol{x}_r) \ln \frac{b_r(\boldsymbol{x}_r)}{\exp(h_i x_i + h_j x_j + c_r(\boldsymbol{x}))} - \sum_i (l_i - 1) \sum_{x_i} b_i(x_i) \ln \frac{b_i(x_i)}{\exp(h_i x_i)}.$$

Here, $\boldsymbol{x}_r$ denotes the pair of vertexes which is included in the edge r, $b_i(x_i)$ and $b_r(\boldsymbol{x}_r)$ are a belief and a pairwise belief respectively, and l_i is the number of neighbors of vertex i. From its definition, $\sum_{x_i} b_i(x_i) = 1$, and $\sum_{\boldsymbol{x}_r} b_r(\boldsymbol{x}_r) = 1$ is satisfied. In our formulation, they are given by

$$b_r(\boldsymbol{x}_r) = p_r(\boldsymbol{x}_r; \boldsymbol{\zeta}_r) = \frac{p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)}{p_0(\boldsymbol{x}; \boldsymbol{\theta}_r)} p_0(\boldsymbol{x}_r; \boldsymbol{\theta}_r), \quad b_i(x_i) = p_0(x_i; \boldsymbol{\theta})$$

where $\boldsymbol{\theta}_r = \Pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ is satisfied.

In [Yedidia et al., 2001a, Yedidia et al., 2001b], the following marginalization conditions (also called the reducibility conditions) are further imposed,

$$b_i(x_i) = \sum_{x_j} b_{ij}(x_i, x_j), \quad b_j(x_j) = \sum_{x_i} b_{ij}(x_i, x_j). \tag{9}$$

These conditions are equivalent to the m–condition in our definition, that is, $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) = \boldsymbol{\eta}_0(\boldsymbol{\theta})$ $(r \in \mathcal{L})$ holds, so that every $\boldsymbol{\zeta}_r$ is no more an independent variable but is dependent on $\boldsymbol{\theta}$. Since

$\boldsymbol{\eta}_0(\boldsymbol{\theta}_r) = \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)$ holds, also $\boldsymbol{\theta}_r = \boldsymbol{\theta}$ $(r \in \mathcal{L})$ holds. With these constraints, the Bethé free energy is simplified as follows,

$$\mathcal{F}_\beta(\boldsymbol{\theta}) = (L-1)\varphi_0(\boldsymbol{\theta}) - \sum_r \varphi_r(\boldsymbol{\zeta}_r(\boldsymbol{\theta})) + \Big(\sum_r \boldsymbol{\zeta}_r(\boldsymbol{\theta}) - (L-1)\boldsymbol{\theta}\Big) \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}). \tag{10}$$

We have to note that at each step of the BP algorithm, eq. (9) is not satisfied, but the e–condition is satisfied. Therefore assuming (9) for original BP immediately gives the equilibrium, and no free parameters are left, which does not allow us to give any further analysis in terms of the Bethé free energy. An important lesson here is that one has to specify, in any analysis based on the free energy, what are the independent variables and what are not, in order for a proper argument.

4.2 A New View on Free Energy

Instead of assuming eq. (9), let us start from the free energy defined in eq. (10) without any constraint on the parameters, that is, all of $\boldsymbol{\theta}, \boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_L$ are the free parameters,

$$\mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_L) = (L-1)\varphi_0(\boldsymbol{\theta}) - \sum_r \varphi_r(\boldsymbol{\zeta}_r) + \Big(\sum_r \boldsymbol{\zeta}_r - (L-1)\boldsymbol{\theta}\Big) \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}).$$

The above function is rewritten in terms of the KL–divergence as,

$$\mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_L) = D[p_0(\boldsymbol{x};\boldsymbol{\theta}); q(\boldsymbol{x})] - \sum_{r=1}^{L} D[p_0(\boldsymbol{x};\boldsymbol{\theta}); p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)] + C$$

where C is a constant. The following theorem is easily derived.

Theorem 3. *The equilibrium $(\boldsymbol{\theta}^*, \boldsymbol{\zeta}_r^*)$ of BP is a critical point of $\mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_r)$.*

Proof. By calculating

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{\zeta}_r} = \mathbf{o},$$

we easily have

$$\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) = \boldsymbol{\eta}_0(\boldsymbol{\theta})$$

which is the m–condition. By calculating

$$\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}} = \mathbf{o}, \tag{11}$$

we are led to the e–condition $(L-1)\boldsymbol{\theta} = \sum_r \boldsymbol{\zeta}_r$. □

The theorem shows that eq. (10) works as the free energy function without giving any constraint. Finally, we compare it with the free energy proposed by Kabashima and Saad [Kabashima and Saad, 2001]. It is a function of $(\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_L)$ and $(\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_L)$, given by

$$\mathcal{F}_{KS}(\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_L; \boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_L) = \mathcal{F}(\boldsymbol{\theta}, \boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_L) + \sum_r D[p_0(\boldsymbol{x};\boldsymbol{\theta}); p_0(\boldsymbol{x};\boldsymbol{\zeta}_r + \boldsymbol{\xi}_r)],$$

where $\boldsymbol{\theta} = \sum_r \boldsymbol{\xi}_r$. When $\boldsymbol{\xi}_r + \boldsymbol{\zeta}_r = \boldsymbol{\theta}$ is satisfied for all r, $\mathcal{F}_{KS}$ is equivalent to our $\mathcal{F}$.

4.3 Property of Fixed Points

Let us study the stability of the fixed point of the BP algorithm with respect to $\mathcal{F}(\boldsymbol{\theta})$. We consider the derivative of $\mathcal{F}$ in (10) with respect to $\boldsymbol{\theta}$. The first derivative is eq. (11) which yields the e–condition, and the second derivative gives the property around the stationary point, that is

$$\frac{\partial^2 \mathcal{F}}{\partial \boldsymbol{\theta}^2} = I_0(\boldsymbol{\theta}) + I_0(\boldsymbol{\theta}) \sum_r \Big(I_r(\boldsymbol{\zeta}_r)^{-1} - I_0(\boldsymbol{\theta})^{-1} \Big) I_0(\boldsymbol{\theta}) + \Delta. \tag{12}$$

Here, $I_0(\boldsymbol{\theta})$ and $I_r(\boldsymbol{\zeta}_r)$ are the Fisher information matrices of $p_0(\boldsymbol{x};\boldsymbol{\theta})$ and $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, respectively, and Δ is the term related to the derivative of the Fisher information matrix, which vanishes when the e–condition is satisfied.

If eq. (12) is positive definite at the stationary point, the Bethé free energy is at least locally minimized at the equilibrium. But it is not always positive definite. Therefore, the conventional gradient descent method of $\mathcal{F}$ may fail.

5 Algorithms and Their Convergence

5.1 e–constraint Algorithm

Since the equilibrium of BP is characterized with the e– and m–conditions, there are two possible versions of algorithms for finding the equilibrium. One is to constrain the parameters always to satisfy the e–condition, and search for the parameters which satisfy the m–condition (e–constraint algorithm), the other is to constrain the parameters to satisfy the m–condition, and search for the parameters which satisfy the e–condition (m–constraint algorithm).

In this section, we discuss e–constraint algorithms. The BP algorithm is an e–constraint algorithm since the e–condition is satisfied at each step, but its convergence is not necessarily guaranteed. We give an alternate of the e–constraint algorithm which has a better convergence property. Let us begin with proposing a new cost function as

$$F(\{\boldsymbol{\zeta}_r\}) = \sum_{r \in \mathcal{L}} ||\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)||^2,$$

under the e–constraint $\boldsymbol{\theta} = \sum_{r\in\mathcal{L}} \boldsymbol{\zeta}_r/(L-1)$. If the cost function is minimized to 0, the m–condition is satisfied, and it is the equilibrium. A naive method to minimize F is the gradient descent algorithm. The gradient is

$$\frac{\partial F}{\partial \boldsymbol{\zeta}_r} = -2 I_r(\boldsymbol{\zeta}_r)(\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)) + \frac{2}{L-1} I_0(\boldsymbol{\theta}) \sum_r (\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)). \tag{13}$$

If the derivative is available, $\boldsymbol{\zeta}_r$ and $\boldsymbol{\theta}$ are updated as,

$$\boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\zeta}_r^t - \delta \frac{\partial F}{\partial \boldsymbol{\zeta}_r^t}, \;\; \boldsymbol{\theta}^{t+1} = \frac{1}{L} \sum_{r\in\mathcal{L}} \boldsymbol{\zeta}_r^{t+1}.$$

where δ is a small positive learning rate. Since $p_0(\boldsymbol{x};\boldsymbol{\theta})$ is a factorisable distribution, it is easy to calculate $\boldsymbol{\eta}_0(\boldsymbol{\theta})$ and $I_0(\boldsymbol{\theta})$. With the BP algorithm, $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)$ is calculated. Since $\boldsymbol{\eta}_0(\boldsymbol{\theta})$, $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)$, and $I_0(\boldsymbol{\theta})$ are tractable, the rest of the problem is to calculate the first term of eq. (13). Fortunately, we have the relation,

$$I_r(\boldsymbol{\zeta}_r)\boldsymbol{h} = \lim_{\alpha\to 0} \frac{\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r + \alpha\boldsymbol{h}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)}{\alpha}.$$

If $(\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r))$ is substituted for $\boldsymbol{h}$, this becomes the first term of eq. (13). Now, we propose a new algorithm.

New algorithm

1. Set $t=0$, $\boldsymbol{\theta}^t = \mathbf{o}$, $\boldsymbol{\zeta}_r^t = \mathbf{o}$, $r \in \mathcal{L}$.
2. Calculate $\boldsymbol{\eta}_0(\boldsymbol{\theta}^t)$, $I_0(\boldsymbol{\theta}^t)$, and $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r^t)$, $r \in \mathcal{L}$ with BP.
3. Let $\boldsymbol{h}_r = \boldsymbol{\eta}_0(\boldsymbol{\theta}^t) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r^t)$ and calculate $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r^t + \alpha\boldsymbol{h}_r)$ for $r \in \mathcal{L}$, where $\alpha > 0$ is small. Then calculate
$$\boldsymbol{g}_r = \frac{\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r^t + \alpha\boldsymbol{h}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r^t)}{\alpha}.$$
4. For $t = 1, 2, \cdots$, update $\boldsymbol{\zeta}_r^{t+1}$ as follows,
$$\boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\zeta}_r^t - \delta\big(-2\boldsymbol{g}_r + \frac{2}{L-1} I_0(\boldsymbol{\theta}^t) \sum_r \boldsymbol{h}_r\big)$$
$$\boldsymbol{\theta}^{t+1} = \sum_{r\in\mathcal{L}} \boldsymbol{\zeta}_r^{t+1}/(L-1).$$
5. If $F(\{\boldsymbol{\zeta}_r\}) = \sum_{r\in\mathcal{L}} ||\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)||^2 > \epsilon$ (ϵ is a threshold) holds, $t+1 \to t$ and go to 2.

This algorithm does not include double loops, which is different from CCCP.

We used here the square norm to define $(\{\boldsymbol{\zeta}_r\})$. However, it is natural to use the Riemannian metric to define the norm. However, this is computationally not easy. We can use the local metric to modify the cost function as

$$\mathcal{F}_{r_0}(\{\boldsymbol{\zeta}_r\}) = \sum_{r\in\mathcal{L}} \left\{\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r\left(\boldsymbol{\zeta}_r\right)\right\}^T I_0(\boldsymbol{\theta}_0)^{-1} \left\{\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r\left(\boldsymbol{\zeta}_r\right)\right\},$$

where $\boldsymbol{\theta}_0$ is the convergent point. The gradient can be calculated similarly by fixing $\boldsymbol{\theta}_0$, which is unknown. Hence, we replace it by $\boldsymbol{\theta}^t$. The calculation of $\boldsymbol{g}_r$ should also be modified to

$$\tilde{\boldsymbol{g}}_r = \frac{\boldsymbol{\eta}_r\left(\boldsymbol{\zeta}_r^t + \alpha I_0(\boldsymbol{\theta}^t)^{-1} \sum \boldsymbol{h}_r\right)}{\alpha}$$

from the point of view of the natural gradient method [Amari, 1998]. We finally have

$$\boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\zeta}_r^t - 2\delta I_0(\boldsymbol{\theta}^t)^{-1}\left[-\tilde{\boldsymbol{g}}_r + \frac{1}{L-1}\sum \boldsymbol{h}_r\right].$$

It should be noted that $I_0(\boldsymbol{\theta})$ is a diagonal matrix having a simple form, so that computation is simple as well.

5.2 m–constraint Algorithm

The other possibility is to constrain the parameters always to satisfy the m–condition, and modify the parameters to satisfy the e–condition. Since the m–condition is satisfied, $\{\boldsymbol{\zeta}_r\}$ are determined dependent on $\boldsymbol{\theta}$.

A naive idea is to repeat the following two steps,

1. For every $r \in \mathcal{L}$,

$$\boldsymbol{\zeta}_r^t = \mathit{\Pi}_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^t) \tag{14}$$

2. Update the parameters as

$$\boldsymbol{\theta}^{t+1} = L\boldsymbol{\theta}^t - \sum_r \boldsymbol{\zeta}_r^t. \tag{15}$$

Starting from $\boldsymbol{\theta}^t$, the algorithm find $\boldsymbol{\zeta}_r^{t+1}$ that satisfy the m–condition by (14), and $\boldsymbol{\theta}^{t+1}$ is adjusted to satisfy the e–condition.

This is a simple recursive algorithm without double loops. We call it the naive m–constraint algorithm. One may use an advanced iteration method that uses, instead of $\boldsymbol{\zeta}_r^t$, new $\boldsymbol{\zeta}_r^{t+1}$. In this case, the algorithm is

$$\boldsymbol{\zeta}_r^{t+1} = \mathit{\Pi}_r \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^{t+1}), \quad \text{where} \quad \boldsymbol{\theta}^{t+1} = L\boldsymbol{\theta}^t - \sum_r \boldsymbol{\zeta}_r^{t+1}. \tag{16}$$

In this algorithm, starting from $\boldsymbol{\theta}^t$, one should solve a non-linear equation in $\boldsymbol{\theta}^{t+1}$, because $\{\boldsymbol{\zeta}_r^{t+1}\}$ are functions of $\boldsymbol{\theta}^{t+1}$. This algorithm therefore uses double loops, the inner loop and the outer loop. This is the idea of CCCP algorithm, and it is also an m–constraint algorithm.

Stability of the Algorithms

Although the naive m–constraint algorithm and the CCCP algorithm share the same equilibrium $\boldsymbol{\theta}^*, \{\boldsymbol{\zeta}_r^*\}$, their local stabilities at the equilibrium are different. It is reported that CCCP has superior properties in this respect. The local stability of the BP algorithm was analyzed by [Richardson, 2000] and also by [Ikeda et al., 2003] in geometrical terms. We give the local stability of the other algorithms.

If we eliminate the intermediate variables $\{\boldsymbol{\zeta}_r\}$ in the inner loop, the naive m–constraint algorithm is

$$\boldsymbol{\theta}^{t+1} = L\boldsymbol{\theta}^t - \sum_r \Pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^t), \tag{17}$$

and the CCCP algorithm is represented as

$$\boldsymbol{\theta}^{t+1} = L\boldsymbol{\theta}^t - \sum_r \Pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^{t+1}), \tag{18}$$

In order to derive the variational equation at the equilibrium, we note that, for the m–projection

$$\boldsymbol{\zeta}_r = \Pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}),$$

a small perturbation $\delta\boldsymbol{\theta}$ in $\boldsymbol{\theta}$ is updated as

$$\delta\boldsymbol{\zeta}_r = I_r(\boldsymbol{\zeta}_r)^{-1} I_0(\boldsymbol{\theta})\delta\boldsymbol{\theta}.$$

The proof is given as follows,

$$\begin{aligned}\boldsymbol{\eta}_r(\boldsymbol{\zeta_r}) + I_r(\boldsymbol{\zeta}_r)\delta\boldsymbol{\zeta}_r \simeq \boldsymbol{\eta}_r(\boldsymbol{\zeta_r} + \delta\boldsymbol{\zeta}_r) = \boldsymbol{\eta}_0(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \simeq \boldsymbol{\eta}_0(\boldsymbol{\theta}) + I_0(\boldsymbol{\theta})\delta\boldsymbol{\theta} \\ \delta\boldsymbol{\zeta}_r = I_r(\boldsymbol{\zeta}_r)^{-1} I_0(\boldsymbol{\theta})\delta\boldsymbol{\theta}.\end{aligned}$$

The variational equations are hence for eq. (17),

$$\delta\boldsymbol{\theta}^{t+1} = \Big(LE - \sum_r I_r(\boldsymbol{\zeta}_r)^{-1} I_0(\boldsymbol{\theta})\Big)\delta\boldsymbol{\theta}^t$$

where E is the identity matrix, and for eq. (18),

$$\delta\boldsymbol{\theta}^{t+1} = L\Big(E + \sum_r I_r(\boldsymbol{\zeta}_r)^{-1} I_0(\boldsymbol{\theta})\Big)^{-1}\delta\boldsymbol{\theta}^t,$$

respectively. Let K be a matrix defined by

$$K = \frac{1}{L}\sum_r \sqrt{I_0(\boldsymbol{\theta})} I_r(\boldsymbol{\zeta}_r)^{-1} \sqrt{I_0(\boldsymbol{\theta})},$$

and $\delta\tilde{\boldsymbol{\theta}}^t$ be a new variable defined as

$$\delta\tilde{\boldsymbol{\theta}}^t = \sqrt{I_0(\boldsymbol{\theta})}\delta\boldsymbol{\theta}^t.$$

The variational equations for eqs. (17) and (18) are then

$$\begin{aligned}\delta\tilde{\boldsymbol{\theta}}^{t+1} &= L(E - LK)\delta\tilde{\boldsymbol{\theta}}^t, \\ \delta\tilde{\boldsymbol{\theta}}^{t+1} &= L(E + LK)^{-1}\delta\tilde{\boldsymbol{\theta}}^t,\end{aligned}$$

respectively.

The equilibrium is stable when the absolute values of the eigenvalues of the respective coefficient matrices are smaller than 1. Let $\lambda_1, \cdots, \lambda_n$ be the eigenvalues of K. They are all real and positive, since K is a symmetric positive-definite matrix. We note that λ_i are close to 1, when $I_r(\boldsymbol{\zeta}_r) \approx I_0(\boldsymbol{\theta})$, or M_r is close to M_0.

Theorem 4. *The equilibrium of the naive m–constraint algorithm* (17) *is stable when*

$$1 + \frac{1}{L} > \lambda_i > 1 - \frac{1}{L}, \quad i = 1, \cdots, n.$$

The equilibrium of the CCCP is stable when the eigen values of K satisfies

$$\lambda_i > 1 - \frac{1}{L}, \quad i = 1, \cdots, n. \tag{19}$$

The theorem shows CCCP has a good convergent property.

Under the m–constraint, the Hessian of $\mathcal{F}(\boldsymbol{\theta})$ at an equilibrium point is equal to (cf. eq. (12))

$$\sqrt{I_0(\boldsymbol{\theta})}\big[LK - (L-1)E\big]\sqrt{I_0(\boldsymbol{\theta})},$$

so that the stability condition eq. (19) for the CCCP is equivalent to the condition that the equilibrium is a local minimum of $\mathcal{F}(\boldsymbol{\theta})$ under the m–constraint. The theorem therefore states that CCCP is locally stable around an equilibrium if and only if the equilibrium is a local minimum of $\mathcal{F}(\boldsymbol{\theta})$, whereas the naive m–constraint algorithm is not necessarily stable even if the equilibrium is a local minimum.

It should be noted that the above local stability result for the CCCP does not follow from the global convergence result given by Yuille [Yuille, 2002]. The latter only states that the CCCP converges to an extremal point of $\mathcal{F}(\boldsymbol{\theta})$, which is not necessarily a local minimum.

Natural Gradient and Discretization

We can consider a gradient rule for updating $\boldsymbol{\theta}$ to find a minimum of $\mathcal{F}(\boldsymbol{\theta})$.

$$\dot{\boldsymbol{\theta}} = -\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}}$$

A natural gradient [Amari, 1998] version of the update rule is

$$\dot{\boldsymbol{\theta}} = -\frac{\partial \mathcal{F}}{\partial \boldsymbol{\eta}_0} = -I_0^{-1}(\boldsymbol{\theta})\frac{\partial \mathcal{F}}{\partial \boldsymbol{\theta}} = (L-1)\boldsymbol{\theta} - \sum_r \Pi_{M_r} \circ p_0(\boldsymbol{x};\boldsymbol{\theta}). \tag{20}$$

For the implementation, it is necessary to discretize the continuous-time update rule. The "fully explicit" scheme of discretization (Euler's method) reads

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \Delta t\Big[(L-1)\boldsymbol{\theta}^t - \sum_r \Pi_{M_r} \circ p_0(\boldsymbol{x};\boldsymbol{\theta}^t)\Big]. \tag{21}$$

When $\Delta t = 1$, this equation is equivalent to the naive m–constraint algorithm (eq. (17)). However, we do not necessarily have to let $\Delta t = 1$: Instead, we may use arbitrary positive value for Δt. We will show how the convergence rate will be affected by the change of Δt in the next section.

The "fully implicit" scheme yields

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \Delta t\big[(L-1)\boldsymbol{\theta}^{t+1} - \sum_r \Pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^{t+1})\big], \tag{22}$$

which, after rearrangement of terms, becomes

$$\big[1 - \Delta t(L-1)\big]\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \Delta t \sum_r \Pi_{M_r} \circ p_0(\boldsymbol{x}; \boldsymbol{\theta}^{t+1}).$$

When $\Delta t = 1/L$, this equation is equivalent to the CCCP algorithm (eq. (18)). Again, we do not have to be bound to the choice $\Delta t = 1/L$. We will also show the relation between Δt and the convergence rate in the next section.

We have just shown that the naive m–constraint algorithm and the CCCP algorithm can be viewed as first-order methods of discretization applied to the continuous-time natural gradient system (20). The local stability result for the CCCP proved in Theorem 4 can also be understood as an example of the well-known absolute stability property of the fully-implicit scheme applied to linear systems. It should also be noted that other, more sophisticated methods for solving ordinary differential equations, such as Runge-Kutta methods (possibly with adaptive stepsize control), the Bulirsch-Stoer method, and so on [Press et al., 1992], are applicable to formulate m–constraint algorithms with better properties, for example, better stability. In this paper, however, we do not discuss possible extension along this line any further.

Acceleration of m–constraint Algorithms

We give the analysis of eqs. (21) and (22) in this section.

The variational equation for eq. (21) is

$$\delta\tilde{\boldsymbol{\theta}}^{t+1} = \big\{E - [LK - (L-1)E]\Delta t\big\}\delta\tilde{\boldsymbol{\theta}}^t.$$

Let

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \tag{23}$$

be the eigenvalues of K. Then, the convergence rate is improved by choosing an adequate Δt. The convergence rate is governed by the largest absolute values of the eigenvalues of $E - \big[LK - (L-1)E\big]\Delta t$, which are given by

$$\mu_i = 1 - \big[L\lambda_i - (L-1)\big]\Delta t.$$

From eq. (23) we have $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$. The stability condition is $|\mu_i| < 1$ for all i. At a locally stable equilibrium point, $\mu_1 < 1$ always holds, so that the algorithm is stable if

$\mu_n > -1$ holds. The convergence to a locally stable equilibrium point is most accelerated when $\mu_1 + \mu_n = 0$, which holds by taking

$$\Delta t_{\text{opt}} = \frac{1}{L\frac{\lambda_1+\lambda_n}{2} - (L-1)}.$$

The variational equation for eq. (22) is

$$\delta\tilde{\boldsymbol{\theta}}^{t+1} = \left\{E + [LK - (L-1)E]\Delta t\right\}^{-1}\delta\tilde{\boldsymbol{\theta}}^{t},$$

and the convergence rate is governed by the largest of the absolute values of the eigenvalues of $\{E + [LK - (L-1)E]\Delta t\}^{-1}$, which should be smaller than 1 for convergence. The eigenvalues are

$$\mu_i = \frac{1}{1 + \left[L\lambda_i - (L-1)\right]\Delta t}.$$

We again have $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$. At a locally stable equilibrium point, $0 < \mu_n$ and $\mu_1 < 1$ always hold, so that the algorithm is always stable. In principle, the smaller μ_1 becomes, the faster the algorithm converges, so that taking $\Delta t \to +\infty$ yields the fastest convergence: However, the algorithm in this limit reduces to the direct evaluation of the e–condition under the m–constraint, which is usually infeasible.

6 Extension

6.1 Extend the Framework to Wider Class of Distributions

In this section, two important extensions of BP is given in the information geometrical framework. First, we extend the model to the case where the marginal distribution of each vertex is an exponential family. A similar extension is given in [Wainwright et al., 2003].

Let $\boldsymbol{t}_i$ be the sufficient statistics of x_i, where $p(x_i;\boldsymbol{\theta}_i)$ is expressed as

$$p(x_i;\boldsymbol{\theta}_i) = \exp(\boldsymbol{\theta}_i \cdot \boldsymbol{t}_i - \phi_i(\boldsymbol{\theta}_i)).$$

This includes many important distributions. We give two examples bellow.

Multinominal distribution: Let x be a discrete stochastic variable, $x \in \{0, \cdots, m\}$.

$$p(x=i) = \exp\Big(\sum_{j=1}^{m}\theta_j t_j(x) - \varphi(\boldsymbol{\theta})\Big)$$

$$t_j(x) = \delta_x(j), \;\; \theta_j = \ln\frac{p(x=j)}{p(y=0)} \;\text{ for }\; j = 1, \cdots, m$$

$$\varphi(\boldsymbol{\theta}) = -\ln p(x=0) = \ln\Big(1 + \sum_{j=1}^{m}\exp\theta_j\Big)$$

Normal distribution: When x follows normal distribution,

$$p(x;\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\Big(-\frac{(x-\mu)^2}{2\sigma^2}\Big) = \exp(\boldsymbol{\theta}\cdot\boldsymbol{t} - \varphi(\boldsymbol{\theta}))$$
$$t_1 = x, \quad t_2 = x^2, \quad \theta_1 = \frac{\mu}{\sigma^2}, \quad \theta_2 = -\frac{1}{2\sigma^2}, \quad \varphi(\boldsymbol{\theta}) = -\frac{\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-\frac{\theta_2}{\pi})$$

Let us define $\boldsymbol{t} = (\boldsymbol{t}_1^T, \cdots, \boldsymbol{t}_n^T)^T$, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \cdots, \boldsymbol{\theta}_n^T)^T$, and let the true distribution be

$$q(\boldsymbol{x}) = \exp(\boldsymbol{h}\cdot\boldsymbol{t} + \boldsymbol{c}(\boldsymbol{x}) - \psi(\boldsymbol{\theta}, \boldsymbol{h})).$$

We can now redefine eq. (1) as follows,

$$p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v}) = \exp(\boldsymbol{\theta}\cdot\boldsymbol{t} + \boldsymbol{v}\cdot\boldsymbol{c}(\boldsymbol{x}) - \psi(\boldsymbol{\theta},\boldsymbol{v}))$$

and S in eq. (2) as

$$S = \Big\{p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v}) \;\Big|\; \boldsymbol{\theta}\in\Theta, \boldsymbol{v}\in\mathcal{V}\Big\},$$

When the problem is to infer the marginal distribution $q(x_i)$ of $q(\boldsymbol{x})$, we can redefine the BP algorithm in this new S, by redefining M_0 and M_r. This extension based on the new definition is simple, and we do not give further details in this article.

6.2 Generalized Belief Propagation

In this section, we show the information geometrical framework for the general belief propagation (GBP) [Yedidia et al., 2001b], which is an important extension of BP.

A naive explanation of GBP shows that the links are reformulated by the subsets of the old set of links. This brings us a new implementation of the algorithm and different inference. In our formulation, we define a new set of variables $c'_s(\boldsymbol{x})$, $s = 1, \cdots, L'$, which summarizes the effect of the links in $\mathcal{L}_s$

$$c'_s(\boldsymbol{x}) = \sum_{r\in\mathcal{L}_s} c_r(\boldsymbol{x}),$$

where $\mathcal{L}_s$ is a subset of $\mathcal{L}$. Those $\mathcal{L}_s$ may have overlap, but $\mathcal{L}_s$ must be chosen to satisfy $\cup_s\mathcal{L}_s = \mathcal{L}$.

GBP is a general framework, which includes a lot of possible cases. We categorize the problem into three important classes, and give an information geometrical framework for them

Case 1

In the simplest case, each $\mathcal{L}_s$ does not have any loop. This is equivalent to TRP. As we have seen in section 3.2, the algorithm is explained in our framework.

Case 2

In the next case, each $\mathcal{L}_s$ can have loops, but there is no overlap, that is, $\mathcal{L}_s \cap \mathcal{L}_{s'} = \emptyset$. The extension to this case is simple. We can apply information geometry by redefining M_r as M_s, where its definition is given as follow

$$M_s \stackrel{\text{def}}{=} \left\{ p_s(\boldsymbol{x}; \boldsymbol{\zeta}_s) = \exp(\boldsymbol{h} \cdot \boldsymbol{x} + c'_s(\boldsymbol{x}) + \boldsymbol{\zeta}_s \cdot \boldsymbol{x} - \varphi_s(\boldsymbol{\zeta}_s)) \;\middle|\; \boldsymbol{\zeta}_s \in \mathcal{R}^n \right\}.$$

Since some loops are treated in a different way, the result might be different from the original BP.

Case 3

Finally, we describe the case where each $\mathcal{L}_s$ can have loops and overlaps with the other sets. In this case we have to extend our framework. Suppose $\mathcal{L}_s$ and $\mathcal{L}_{s'}$ have overlap, and both have loops. We explain the case with an example in Fig.4.

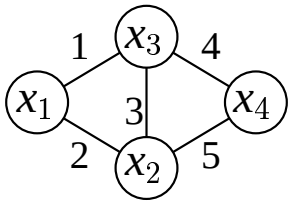

Figure 4: Case 3.

Let us first define the following distributions,

$$q(\boldsymbol{x}) = C \exp\Big(\boldsymbol{h} \cdot \boldsymbol{x} + \sum_{i=1}^{5} c_i(\boldsymbol{x})\Big), \quad p_0(\boldsymbol{x}; \boldsymbol{\theta}) = \exp(\boldsymbol{h} \cdot \boldsymbol{x} + \boldsymbol{\theta} \cdot \boldsymbol{x} - \varphi_0(\boldsymbol{\theta}))$$
$$p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1) = \exp\Big(\boldsymbol{h} \cdot \boldsymbol{x} + \sum_{i=1}^{3} c_i(\boldsymbol{x}) + \boldsymbol{\zeta}_1 \cdot \boldsymbol{x} - \varphi_1(\boldsymbol{\zeta}_1)\Big) \tag{24}$$
$$p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2) = \exp\Big(\boldsymbol{h} \cdot \boldsymbol{x} + \sum_{i=3}^{5} c_i(\boldsymbol{x}) + \boldsymbol{\zeta}_2 \cdot \boldsymbol{x} - \varphi_2(\boldsymbol{\zeta}_2)\Big).$$

Even if $\boldsymbol{\zeta}_1$, $\boldsymbol{\zeta}_2$, and $\boldsymbol{\theta}$ satisfies the e–condition as $\boldsymbol{\theta} = \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2$, this does not imply

$$C \frac{p_1(\boldsymbol{x}; \boldsymbol{\zeta}_1) p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2)}{p_0(\boldsymbol{x}; \boldsymbol{\theta})}$$

is equivalent to $q(\boldsymbol{x})$, since $c_3(\boldsymbol{x})$ is counted twice. Therefore, we introduce another model $p_3(\boldsymbol{x}; \boldsymbol{\zeta}_3)$, which has the following form.

$$p_3(\boldsymbol{x}; \boldsymbol{\zeta}_3) = \exp\Big(\boldsymbol{h} \cdot \boldsymbol{x} + c_3(\boldsymbol{x}) + \boldsymbol{\zeta}_3 \cdot \boldsymbol{x} - \varphi_3(\boldsymbol{\zeta}_3)\Big). \tag{25}$$

Now,

$$C\frac{p_1(\boldsymbol{x};\boldsymbol{\zeta}_1)p_2(\boldsymbol{x};\boldsymbol{\zeta}_2)}{p_3(\boldsymbol{x};\boldsymbol{\zeta}_3)}$$

becomes equal to $q(\boldsymbol{x})$ where $\boldsymbol{\zeta}_3 = \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2$ is the e–condition.

Next we look at the m–condition. The original form of the m–condition is

$$\sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \boldsymbol{x} p_s(\boldsymbol{x};\boldsymbol{\zeta}_s)$$

but, in this case, this form is not enough. We need a further condition, that is,

$$p_s(x_2, x_3;\boldsymbol{\zeta}_s) = \sum_{x_1,x_4} p_s(\boldsymbol{x};\boldsymbol{\zeta}_s)$$

should be the same for $s = \{1, 2, 3\}$. The models in eqs. (24) and (25) are not sufficient, since we do not have enough parameters to specify a joint distribution of (x_2, x_3), and the model must be extended. In the binary case, we can extend the models by adding one variable as follows,

$$\begin{aligned}
p_1(\boldsymbol{x};\boldsymbol{\zeta}_1, v_1) &= \exp\Big(\boldsymbol{h}\cdot\boldsymbol{x} + \sum_{i=1}^{3} c_i(\boldsymbol{x}) + \boldsymbol{\zeta}_1\cdot\boldsymbol{x} + v_1 x_2 x_3 - \varphi_1(\boldsymbol{\zeta}_1, v_1)\Big)\\
p_2(\boldsymbol{x};\boldsymbol{\zeta}_2, v_2) &= \exp\Big(\boldsymbol{h}\cdot\boldsymbol{x} + \sum_{i=3}^{5} c_i(\boldsymbol{x}) + \boldsymbol{\zeta}_2\cdot\boldsymbol{x} + v_2 x_2 x_3 - \varphi_2(\boldsymbol{\zeta}_2, v_2)\Big)\\
p_3(\boldsymbol{x};\boldsymbol{\zeta}_3, v_3) &= \exp(\boldsymbol{h}\cdot\boldsymbol{x} + c_3(\boldsymbol{x}) + \boldsymbol{\zeta}_3\cdot\boldsymbol{x} + v_3 x_2 x_3 - \varphi_3(\boldsymbol{\zeta}_2, v_3)),
\end{aligned}$$

and the m–projection becomes,

$$\begin{aligned}
\sum_{\boldsymbol{x}} \boldsymbol{x} p_0(\boldsymbol{x};\boldsymbol{\theta}) &= \sum_{\boldsymbol{x}} \boldsymbol{x} p_s(\boldsymbol{x};\boldsymbol{\zeta}_s), \quad s = 1, 2, 3\\
\sum_{\boldsymbol{x}} x_2 x_3 p_1(\boldsymbol{x};\boldsymbol{\zeta}_1, v_1) &= \sum_{\boldsymbol{x}} x_2 x_3 p_2(\boldsymbol{x};\boldsymbol{\zeta}_2, v_2) = \sum_{\boldsymbol{x}} x_2 x_3 p_3(\boldsymbol{x};\boldsymbol{\zeta}_3, v_3).
\end{aligned}$$

We revisit the e–condition, which is now extended as,

$$\boldsymbol{\zeta}_3 = \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2, \quad v_3 = v_1 + v_2.$$

This is a simple example, but we can describe any GBP problem in the information geometrical framework in a similar way.

7 Conclusion

Stochastic reasoning is a useful method to analyze the behavior of graphical models including stochastic neural networks. Belief propagation is a useful method, and in order to analyze its behavior and to give a theoretical foundation, a variety of methods are proposed from statistical

physics, information theory and information geometry. The present paper elucidates various interdisciplinary concepts from the point of view of information geometry, and proposes a free-energy-like function to derive a family of new algorithms. Their local stability and convergence rates are analyzed explicitly.

References

[Amari, 1998] Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.

[Amari, 2001] Amari, S. (2001). Information geometry on hierarchy of probability distributions. *IEEE Transactions on Information Theory*, 47(5):1701–1711.

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

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

[Ikeda et al., 2002] Ikeda, S., Tanaka, T., and Amari, S. (2002). Information geometrical framework for analyzing belief propagation decoder. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 407–414. The MIT Press, Cambridge, MA.

[Ikeda et al., 2003] Ikeda, S., Tanaka, T., and Amari, S. (2003). Information geometry of turbo codes and low-density parity-check codes. submitted to IEEE transaction on Information Theory.

[Itzykson and Drouffe, 1989] Itzykson, C. and Drouffe, J.-M. (1989). *Statistical Field Theory*, volume 1. Cambridge Univ. Press, NY.

[Jordan, 1999] Jordan, M. I. (1999). *Learning in Graphical Models*. The MIT Press.

[Kabashima and Saad, 1999] Kabashima, Y. and Saad, D. (1999). Statistical mechanics of error-correcting codes. *Europhysics Letters*, 45(1):97–103.

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

[Lauritzen and Spiegelhalter, 1988] Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:157–224.

[Opper and Saad, 2001] Opper, M. and Saad, D., editors (2001). *Advanced Mean Field Methods – Theory and Practice*. The MIT Press.

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

[Press et al., 1992] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C*. Cambridge University Press, second edition.

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

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

[Tanaka, 2001] Tanaka, T. (2001). Information geometry of mean-field approximation. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods – Theory and Practice*, chapter 17, pages 259–273. The MIT Press.

[Wainwright et al., 2002] Wainwright, M., Jaakkola, T., and Willsky, A. (2002). Tree-based reparameterization for approximate inference on loopy graphs. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 1001–1008. The MIT Press, Cambridge, MA.

[Wainwright et al., 2003] Wainwright, M., Jaakkola, T., and Willsky, A. (2003). Tree-reweighted belief propagation algorithms and approximate ML estimate by pseudo-moment matching. In Bishop, C. M. and Frey, B. J., editors, *Proceeding of Ninth International Workshop on Artificial Intelligence and Statistics*.

[Weiss, 2000] Weiss, Y. (2000). Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12(1):1–41.

[Yedidia et al., 2001a] Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001a). Bethe free energy, Kikuchi approximations, and belief propagation algorithms. Technical Report TR2001–16, Mitsubishi Electric Research Laboratories.

[Yedidia et al., 2001b] Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001b). Generalized belief propagation. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems 13*, pages 689–695. MIT Press.

[Yuille, 2002] Yuille, A. L. (2002). CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722.

[Yuille and Rangarajan, 2002] Yuille, A. L. and Rangarajan, A. (2002). The concave-convex procedure (CCCP). In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, pages 1033–1040. The MIT Press, Cambridge, MA.

[Yuille and Rangarajan, 2003] Yuille, A. L. and Rangarajan, A. (2003). The concave-convex procedure. *Neural Computation*, 15(4):915–936.