収束保証つき確率推論アルゴリズム

西山　悠　新機軸創発センター　統計的機械学習NOE　特任研究員

私たちの周りには不確実なものであふれています．不確実なものを制御し，計算することは大切です．不確定要素を表す確率変数をいくつか集めたとき，それらが従う同時分布から周辺分布を計算することを考えます．これは基本計算であることから，様々な応用を持ちます．

期待値計算，事後確率推定, 因果推論, コンピュータビジョン, たんぱく質のフォールディング, 遺伝子解析, etc

同時分布から周辺分布を計算する問題は，確率変数がたくさんある高次元のときには，一般に計算が困難です．確率変数の個数について指数オーダとなります．

同時分布:

$$p(\mathbf{x};\theta) = \exp\left\{\sum_{i\in\mathcal{V}}\theta_i(x_i) + \sum_{ij\in\mathcal{E}}\theta_{ij}(x_i,x_j) - \Phi(\theta)\right\},$$

周辺分布:

$$p_\alpha(\mathbf{x}_\alpha) = \sum_{\mathbf{x}\backslash\mathbf{x}_\alpha} p(\mathbf{x};\theta).$$

周辺分布の計算のしやすさは，同時分布の持つ位相的な性質により決まります．複数の確率変数の間の確率的関係が，鎖，木構造のように，サイクルを持たないグラフで与えられるとき，効率的なアルゴリズムである確率伝搬法(Belief Propagation; BP)があります．これは確率変数の個数Nについて線形オーダ$O(N)$です．確率伝搬法は，グラフのエッジに沿って，各確率変数の情報(メッセージ)を伝搬，伝達することで，周辺分布を計算するメッセージパシングアルゴリズムです．

それでは，確率的関係がサイクルを含むグラフで表されるとき，どのように同時分布から周辺分布を計算するのでしょうか? 効率的に周辺分布を"近似"計算する推論アルゴリズムがあります．

Loopy BP (Pearl, 1988)
Generalized BP (Yedidia, 2001)
Fractional BP (Wiegerinck, 2003)
Tree reweighted BP (Wainwright, 2005)
etc

これらのアルゴリズムは，収束が保証されない問題点があります．同時分布によっては，収束しません．それでは，収束が数学的に保証された高信頼と言えるアルゴリズムはどのように構成されるのでしょうか？解の候補である周辺分布の集合の上に目的関数を設定し，目的関数を最小化する制約付き最適化問題を解くことで周辺分布を計算します．

【目的関数】:
ベーテ自由エネルギー

$$F_{\text{Bethe}}(\mathbf{b}) = -\sum_{\{ij\}\in\mathcal{E}} S_{ij}(b_{ij}) - \sum_{i\in\mathcal{V}}(1-|\mathcal{N}_i|)S_i(b_i) + \langle E(\mathbf{x})\rangle_{\mathbf{b}}$$

菊池自由エネルギー

$$F_{\text{Kikuchi}}(\mathbf{b}) = -\sum_{\alpha\in\mathcal{R}} c_\alpha S_\alpha(b_\alpha) + \langle E(\mathbf{x})\rangle_{\mathbf{b}}$$

TRW自由エネルギー

$$F_{\text{TRW}}(\mathbf{b}) = -\sum_{i\in\mathcal{V}} S_i(b_i) + \sum_{ij\in\mathcal{E}} \rho_{ij} I_{ij}(b_{ij}) + \langle E(\mathbf{x})\rangle_{\mathbf{b}}$$

目的関数によって，計算される周辺分布の近似解，関数の凸性は異なります．たとえば，TRW自由エネルギーは，周辺分布の解の候補上で凸性が保証されています．これより，TRW自由エネルギー最小化では，初期値によらず，常に同じ近似周辺分布を計算することになります．ベーテ/菊池自由エネルギーでは，一般に，複数の局所解を持つ多峰性の関数です．これらの目的関数は統計物理学とつながりがあります．

目的関数の性質に応じて，最小化アルゴリズムが選ばれます(凸最適化, Concave Convex Procedure (Yuille 2002), Heskes' minimizing algorithm (Heskes 2006), etc).

収束保証つき確率推論アルゴリズムの例:

Bethe/Kikuchi-CCCP (Yuille, 2002, Nishiyama, 2008), TRW-CCCP (Nishiyama, 2010), HAK algorithms (Heskes, 2006), TRW-GP (Globerson, 2007), sum-TRW-S (Meltzer, 2009)

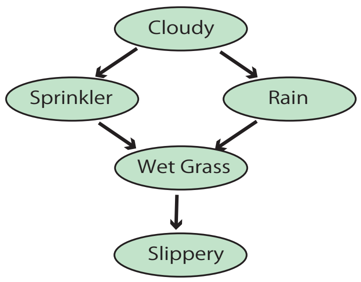
【最小化アルゴリズム】:

Concave-Convex Procedure (CCCP)
目的関数が$F(\mathbf{b}) = f_{\text{vex}}(\mathbf{b}) - g_{\text{vex}}(\mathbf{b})$ (f_{vex}, g_{vex}は共に凸関数)の形で与えられたとき，変数$\mathbf{b}$を初期値$\mathbf{b}^0$から始めて，$\partial f_{\text{vex}}(\mathbf{b}^{t+1}) = \partial g_{\text{vex}}(\mathbf{b}^t)$を満たすように$\mathbf{b}^t$から$\mathbf{b}^{t+1}$へと更新していきます．2つの凸関数$f_{\text{vex}}$, g_{vex}の間で，勾配が等しくなるように$\mathbf{b}$を更新する更新式です．このとき目的関数の単調減少性が数学的に保証されます．

CCCPが誘導するアルゴリズム集合
目的関数$F(\mathbf{b})$が与えられたとき，2つの凸関数f_{vex}, g_{vex}による差表示は無限個（関数自由度）あります．これによりCCCPは，与えられた目的関数$F(\mathbf{b})$について，$F(\mathbf{b})$の単調減少アルゴリズムの集合を導きます．

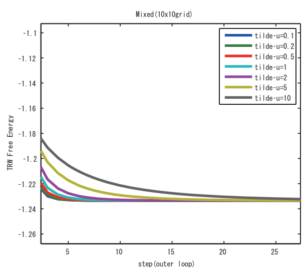
このアルゴリズム集合を積極的に取り入れることによる，ベーテ/菊池/TRW自由エネルギー最小化を行う収束保証つき確率推論アルゴリズムを開発しました．選ばれるアルゴリズム集合内の点に依存して，収束の速さが異なります．

この研究は，東京工業大学大学院博士後期課程での研究内容であり，東京工業大学 渡辺澄夫教授, UCLA大学 Alan Yuille教授, Xingyao Yeとの共同研究に基づいています．



Algorithm 1 TRW-CCCP; A family of CCCP to minimize the TRW free energy.

1. **Initialization:** $b_\alpha^{(1)}(\mathbf{x}_\alpha) = b_\alpha^{(0)}(\mathbf{x}_\alpha) \propto \exp\left[\frac{\theta_\alpha(\mathbf{x}_\alpha)}{c_\alpha}\right]$　　$\alpha\in\mathcal{V}\cup\mathcal{E}$
2. Choose a free vector $\mathbf{u}$ in the set $\mathcal{U}_{\text{CCCP}}$.
3. **Inner loop:**
   $$b_i(x_i) \propto \left[\frac{b_{ij}(x_i)}{b_i(x_i)}\right]^{\frac{\tilde{u}_{ij}}{\tilde{u}_i+\tilde{u}_{ij}}} b_i(x_i)$$
   $$b_{ij}(x_i,x_j) \propto \left[\frac{b_{ij}(x_i)}{b_i(x_i)}\right]^{-\frac{\tilde{u}_i}{\tilde{u}_i+\tilde{u}_{ij}}} b_{ij}(x_i,x_j) \qquad ij\in\mathcal{E}$$
4. **Outer loop:**
   $$b_\alpha^{(t+2)}(\mathbf{x}_\alpha) \propto \left[\frac{b_\alpha^{(t+1)}(\mathbf{x}_\alpha)}{b_\alpha^{(t)}(\mathbf{x}_\alpha)}\right]^{\frac{u_\alpha}{\tilde{u}_\alpha}} b_\alpha^{(t+1)}(\mathbf{x}_\alpha) \qquad \alpha\in\mathcal{V}\cup\mathcal{E}$$
5. **Output:** A set of approximate marginals $\mathbf{b}^*(\simeq\mathbf{p})$ when outer loop converges. Otherwise, set beliefs to $\mathbf{b}^{(t)} = \mathbf{b}^{(t+1)}$, $\mathbf{b}^{(t+1)} = \mathbf{b}^{(t+2)}$ and go to **2.**



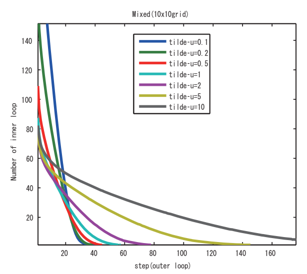


Figure 1: （左図）ベイジアンネットワーク（中央図）TRW-CCCPアルゴリズム（右図）TRW自由エネルギーが単調減少をしている様子