Complexity of Approximating a Bethe Equilibrium

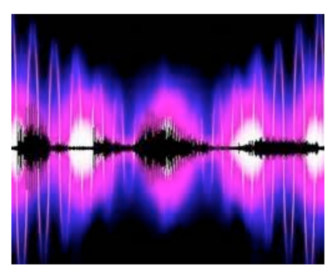
JINWOO SHIN
IBM T. J. Watson Research

INTRODUCTION -- GRAPHICAL MODELS

- Graphical models have been studied as powerful formalisms modeling inference problems
  - They are also known as Markov Random Fields (MRFs)
  - Applications include

Face detection

Error correcting codes



Speech separation

Image denoising

- Formal setup of pairwise binary MRFs
  - Graph of n vertices : $G=(V,E)$
  - Functions on vertices and edges : $\psi_v:\{0,1\}\to\mathbb{R}_+, \forall v\in V$ and $\psi_{u,v}:\{0,1\}^2\to\mathbb{R}_+, \forall (u,v)\in E$
  - Joint distribution of binary random variables $\mathbf{x}=\{x_v \mid v\in V\}$ is given as

$$p(\mathbf{x})=\frac{1}{Z}\prod_{v\in V}\psi_v(x_v)\prod_{(u,v)\in E}\psi_{u,v}(x_u,x_v) \qquad \text{for } \mathbf{x}\in\{0,1\}^n.$$

  - The normalizing factor $Z=\sum_{\mathbf{x}\in\{0,1\}^n}\prod_{v\in V}\psi_v(x_v)\prod_{(u,v)\in E}\psi_{u,v}(x_u,x_v)$ is called `partition function'
  - Our result is easily extendable to general pairwise MRFs
  - Any (even non-pairwise) MRF can be expressed by a pairwise MRF

COMPUTATIONAL PROBLEMS FOR GRAPHICAL MODELS AND BELIEF PROPAGATION

- How to compute **marginal probabilities** in the joint distribution ?
- How to compute the **partition function** ?
- How to compute the **MAP** configuration, i.e., $x^*=\arg\max_x\ p(x)$?
  - All are NP-hard (even to approximate) in general
- BP (Belief Propagation) is a popular heuristic algorithm for these problems
  - Each vertex iteratively exchanges messages (called beliefs) with its neighbors (until they converge)
  - Empirically successful in many problems, e.g., error-correcting codes [Gallager 1960], compress sensing [Donoho et al, 2008], image processing [Sudderth et al. 2008], etc.

OUR GOAL -- FIXING CONVERGENCE OF BP -- PROVABLY

- BP often does not converge
  - Then, it does not provide any answer
  - However, a fixed point message of BP always exists due to Brouwer's theorem
- **Goal : Design a BP-like algorithm always converging to a fixed point message of BP in polynomial time**
  - Then, it becomes a better alternative to BP (i.e. fixing its convergence issue)
  - In general, the fixed point computation is PPAD-hard
- Known algorithms converging a fixed-point message of BP
  - e.g. [Teh and Welling 2001], [Yuille 2002]
  - Not guaranteed to converge in **polynomial time** (provably, in any sense)

OUR ALGORITHM

1. Algorithm parameters:

$$\mathbf{y}(t)=[\,y_v(t)\in(0,1):v\in V\,] \quad \text{at the } t\text{-th iteration.}$$

2. $\mathbf{y}(t)$ is updated as:

$$y_v(t+1) = \left[y_v(t)+\frac{1}{\sqrt{t}}\left(\psi^{(v)}+\ln\frac{1-y_v(t)}{y_v(t)} + \sum_{u\in\mathcal{N}(v)}\ln\left(\frac{1-y_v(t)-y_u(t)+y_{u,v}(t)}{1-y_v(t)}\cdot\frac{y_v(t)}{y_v(t)-y_{u,v}(t)}\right)\right)\right]_*,$$

where the projection $[\cdot]_*$ at the t-th iteration is defined as

$$[x]_*=\begin{cases} x & \text{if } \frac{1}{t^{1/4}}\le x\le 1-\frac{1}{t^{1/4}} \\ \frac{1}{t^{1/4}} & \text{if } x<\frac{1}{t^{1/4}} \\ 1-\frac{1}{t^{1/4}} & \text{if } x>1-\frac{1}{t^{1/4}} \end{cases},$$

and $y_{u,v}(t)>0$ is computed as the unique solution satisfying

$$e^{\psi^{(u,v)}}\cdot\frac{y_u(t)-y_{u,v}(t)}{1-y_u(t)-y_v(t)+y_{u,v}(t)}\cdot\frac{y_v(t)-y_{u,v}(t)}{y_{u,v}(t)}=1 \quad \text{and} \quad y_{u,v}(t)<\min\{y_v(t),y_u(t)\}.$$

3. Compute messages $\{m_{u\to v},m_{v\to u}\}$ as

$$m_{u\to v} = \frac{\psi_{u,v}(0,1)}{\psi_{u,v}(0,0)}\cdot\frac{1-y_v(t)-y_u(t)+y_{u,v}(t)}{1-y_v(t)}\cdot\frac{y_v(t)}{y_v(t)-y_{u,v}(t)}.$$

MAIN THEOREM

- The algorithm outputs an approximate BP-fixed point message
  - In $2^{O(\Delta)}n^2/\varepsilon^5$ iterations
  - ε : (multiplicative) approximation parameter
  - Δ : maximum degree of underlying graph G

QUESTION I -- WHY HARD TO FIND A FIXED POINT OF BP ?

- It is known [Yedidia, Freeman and Weiss 2004] that

$$\text{Fixed points of BP} \overset{1-1}{=} \text{Zero gradient points of } F_{bethe}$$

  - F_{bethe} is called `Bethe free energy function' and provide an approximation of $\log Z$
- However, finding a zero gradient is hard since F_{bethe} is not convex
  - In general, the zero gradient point (i.e. local minimum) computation is PLS-hard

QUESTION II -- DO GRADIENT ALGORITHMS WORK ?

- One can hope the gradient algorithm finds a zero gradient point

$$x(t+1)=x(t)+\alpha\nabla F_{bethe}(x(t))$$

  - α is called the step-size
- When does it work ?
  - Sufficient conditions : **Bounded** derivatives and **Unbounded** underlying domain
  - Then, one can choose the step-size for converging with provable convergence rate
- However, the Bethe free energy function has two issues
  - The underlying domain D of F_{bethe} is **bounded** (hence, a projection may require)
  - Derivatives of F_{bethe} are **unbounded** close to the boundary of D

MAIN LEMMA -- HOW TO FIX THE ISSUES ?

- We prove that It is possible to choose the step-size so that the gradient algorithm for the Bethe free energy function F_{bethe} always keeps ε-far from the boundary of its domain D
  - Hence, derivatives computed by the gradient algorithm is bounded
  - Furthermore, a projection is not necessary
  - We study the behavior of gradient ∇F_{bethe} near the boundary of domain D

SIMULATION RESULTS