Message passing networks

Devavrat Shah
Laboratory for Information and Decision Systems
Department of EECS
Massachusetts Institute of Technology
Message passing

- They are
  - Iterative, distributed algorithms
  - Using minimal computation and little memory

Iterative algorithm: e.g. Newton Raphson

\[
x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}
\]
Message passing

- They are
  - Iterative, distributed algorithms
  - Using minimal computation and little memory

- Iterative, distributed algorithm: computing average
  - Problem: network connectivity graph \( G = (V, E) \) of \( n \) nodes
    - nodes have values \( x_1, \ldots, x_n \)
    - wish to compute average \( x_{\text{ave}} = \frac{1}{n} \left( \sum_{i=1}^{n} x_i \right) \)

```
Avg = 4
```
Message passing

- They are
  - Iterative, distributed algorithms
  - Using minimal computation and little memory
- Iterative, distributed algorithm: computing average
  - Problem: network connectivity graph $G = (V, E)$ of $n$ nodes
    - nodes have values $x_1, \ldots, x_n$
    - wish to compute average $x_{ave} = \frac{1}{n} \left( \sum_{i=1}^{n} x_i \right)$
  - A naive message passing: iterative, ‘pair-wise’ averaging
Message passing

- Message passing for averaging (and variants)
  - Engineered systems
    - consensus among unmanned vehicles, e.g. Tsitsiklis 84
    - load balancing, e.g. Rabani-Sinclair-Wanka 98
  - Information processing and estimation
    - linear estimation (Gossip), e.g. Boyd-Ghosh-Prabk’r-S 04
    - information spreading, e.g. Kempe-Dobra-Gehrke 03
  - Natural systems
    - modeling learning in a society, e.g. Golub-Jackson 09
    - modeling behavior birds, e.g. Vicsek et al 95; Chazelle 09
Message passing

- Message passing paradigm is useful for
  - Engineered systems
    - algorithmic building blocks for scalable architecture
  - Statistical networks, information processing
    - algorithm that can cope with scale of data
  - Natural systems
    - modeling behavior of ‘agents’
Message passing

- Basic challenge
  - What global problems can be solved using message passing?
Message passing

• Basic challenge
  ○ What global problems can be solved using message passing?
  ○ Engineering system, statistical inference
    • to achieve high performance
  ○ Natural system
    • to understand or predict overall behavior
Message passing

• Basic challenge
  ◦ What global problems can be solved using message passing?
  ◦ Engineering system, statistical inference
    • to achieve high performance
  ◦ Natural system
    • to understand or predict overall behavior

• I will address this challenge in the context of
  ◦ Wireless communication network
    • medium access
  ◦ Inference in graphical model
    • belief propagation
Wireless Medium Access

with Jinwoo Shin

Applied Math, MIT
Contestation resolution

- Examples
  - (Old) Ethernet, wireless network, large software systems, parallel computation, distributed database system,...
Contention resolution

• Key challenge: efficient algorithm design under stringent constraint
  ○ Minimal co-operation to reduce ‘protocol overhead’, e.g.
    • nodes know if resource is BUSY or FREE
    • or, their attempt to access was SUCCESS or FAILURE
Medium access

- Let’s play a game
  - Reward $20
Medium access

• Let’s play a game
  ○ Reward $20

• Rules
  ○ Respond, when asked, within 20ms
  ○ No reward if
    • none, or more than one simultaneous responses
  ○ Else, unique responder wins

Fact: reaction time to auditory stimulus is 14
Medium access

Let’s play a game

- Reward $20

Rules

- Respond, when asked, within 20ms
- No reward if
  - None, or more than one simultaneous responses
  - Else, unique responder wins

Fact: reaction time to auditory stimulus is 140-160ms

Model

- Constraints
  - Interfering nodes cannot transmit simultaneously
Model

- Constraints
  - Interfering nodes cannot transmit simultaneously
Model

Constraints

- Interfering nodes cannot transmit simultaneously
- Nodes have only local information
  - Contending simultaneous transmissions
Model

Network Interference Graph

- Medium access
  - When to transmit subject to inference constraints
    - using local information
    - with an aim to maximize utilization of wireless medium
Model

Arrival process with rate $\lambda_i$

- Network interference graph $G = (V, E)$ with $n$ queues
  - $E = \{(i, j) : i$ and $j$ can't tx simultaneously$\}$
  - Packets arrive at rate $\lambda_i$ for queue $i$
- Medium access: at each time instance
  - Selects non-interfering queues (to tx), i.e. independent set of $G$
Model

- Network interference graph $G = (V, E)$ with $n$ queues
  - $E = \{(i, j) : i$ and $j$ can't tx simultaneously$\}$
  - Packets arrive at rate $\lambda_i$ for queue $i$
- Medium access: at each time instance
  - Selects non-interfering queues (to tx), i.e. independent set of $G$
Model

- Network interference graph $G = (V, E)$ with $n$ queues
  - $E = \{(i, j) : i$ and $j$ can't tx simultaneously\}$
  - Packets arrive at rate $\lambda_i$ for queue $i$
- Medium access: at each time instance
  - Selects non-interfering queues (to tx), i.e. independent set of $G$
Model

- Network interference graph $G = (V, E)$ with $n$ queues
  - $E = \{(i, j) : i \text{ and } j \text{ can't tx simultaneously}\}$
  - Packets arrive at rate $\lambda_i$ for queue $i$
- Medium access: at each time instance
  - Selects non-interfering queues (to tx), i.e. independent set of $G$
Model

• Let $\mathcal{I}(G)$ be set of independent sets of $G$
  
  ○ That is, $\mathcal{I}(G) = \{ \sigma \in \{0, 1\}^n : \sigma_i + \sigma_j \leq 1 \text{ for all } (i, j) \in E \}$

• Effective service rate vector $\mu = [\mu_i]$ is s.t.

  ○ $\mu = \sum_{\sigma \in \mathcal{I}(G)} \alpha_\sigma \sigma$, with $\alpha_\sigma \geq 0$

  • $\sum_\sigma \alpha_\sigma \leq 1$

• Therefore, effective resource or ‘capacity region’

  ○ Convex hull of $\mathcal{I}(G)$, say $\text{conv}(\mathcal{I}(G))$
Performance metric

• Throughput optimal medium access

  ○ Queues remain finite for any $\lambda \in \text{conv}(\mathcal{I}(G))^o$
### Performance metric

- **Notations**
  - \( Q(t) = [Q_i(t)] \in \mathbb{R}_+^n \) be the queue-sizes at time \( t \)
  - \( A(s, t) = [A_i(s, t)] \) cumulative arrivals to queue \( i \) in time \( [s, t] \)
    - arrival rate vector \( \lambda = [\lambda_i] \)
    - and \( \mathbb{E}[A_i(s, t)] = \lambda_i(t - s) \)
  - \( \sigma(t) = [\sigma_i(t)] \in \mathcal{I}(G) \subset \{0, 1\}^n \) be the schedule at time \( t \)
    - \( \sigma_i(t) = 1 \) means the queue \( i \) is transmitting at time \( t \).

- **Dynamics:** for each \( i \)

\[
Q_i(t) = Q_i(s) + A_i(s, t) - \int_s^t \sigma_i(y) \cdot 1_{Q_i(y) > 0} \, dy
\]
Prior work

- Two classes of medium access algorithms (since early 70s)
  - Practical random access algorithm
  - Performance optimal queue-based algorithm
Random access algorithm

• Each queue \( i \) checks medium ‘regularly’
  ◦ Whether any ‘neighboring’ node is txing or not
  ◦ If medium is free, attempts transmission with prob. \( p_i \)
    • upon being successful, tx for time duration \( W_i \)
  ◦ Else
    • do nothing

• Popular back-off protocols are instance of this, e.g.
  ◦ ALOHA [Abramson-Kuo 73], [Metcalfe-Bogg 76]
Random access algorithm

• Properties
  ◦ Naive message passing and easy to implement
  ◦ But, as is poor in performance or hard to analyze

• Various positive, negative results
  ◦ [Kelly-McPhee 85, 87], [Aldous 87], [Tsitsiklis 87], [Tsybakov-Likhanov 87], [Mosley-Humblet 85], ...

• A notable, positive result by Hastad-Leighton-Rogoff 96
  ◦ Polynomial backoff is throughput optimal
  ◦ But, only for complete interreference graph

• History (till 00) maintained by L. Goldberg
  ◦ http://www.csc.liv.ac.uk/~leslie/contention.html
Maximum weight algorithm

- At each time instance $t$
  - Choose $\sigma(t) \in \mathcal{I}(G)$ so that
    $$\sigma(t) = \arg\max_{\sigma \in \mathcal{I}(G)} \sum_{i} \sigma_i Q_i(t)$$
Maximum weight algorithm

\[ t = 0 \]

\[ Q_1 = 30 \quad \sigma_1 = 1 \]

\[ Q_2 = 30 \quad \sigma_2 = 1 \]

\[ Q_3 = 45 \quad \sigma_3 = 0 \]

\[ Q_4 = 10 \quad \sigma_4 = 0 \]

\[ Q_5 = 5 \quad \sigma_5 = 0 \]

• At each time instance \( t \)
  
  ○ Choose \( \sigma(t) \in \mathcal{I}(G) \) so that

  \[ \sigma(t) = \arg\max_{\sigma \in \mathcal{I}(G)} \sum_i \sigma_i Q_i(t) \]
Maximum weight algorithm

\[ t = 20 \]

\[ Q_1 = 25 \quad \sigma_1 = 0 \]

\[ Q_2 = 25 \quad \sigma_2 = 0 \]

\[ Q_3 = 50 \quad \sigma_3 = 1 \]

\[ Q_4 = 12 \quad \sigma_4 = 1 \]

\[ Q_5 = 8 \quad \sigma_5 = 0 \]

- At each time instance \( t \)
  - Choose \( \sigma(t) \in \mathcal{I}(G) \) so that
    \[ \sigma(t) = \arg \max_{\sigma \in \mathcal{I}(G)} \sum_{i} \sigma_i Q_i(t) \]
Generalized maximum weight algorithm

- At each time instance $t$
  - Choose $\sigma(t) \in I(G)$ so that
    $$\sigma(t) = \arg \max_{\sigma \in I(G)} \sum_i \sigma_i f(Q_i(t))$$
  - For increasing function $f$ with $f(0) = 0$, $\lim_{x \to \infty} f(x) = \infty$

- Properties
  - Throughput optimal [Tassiulas-Ephremides 92]
  - Appropriate $f$ provides optimal queue-size [S-Wischik 06, 09, 10]
  - But
    - as is, centralized and
    - requires solving computationally hard problem each time
• Our algorithm
  ◦ Adaptive random access based on queue-size
  ◦ ‘Simulates’ maximum weight algorithm

Performance

Ease of Implementation

Status quo

Ideal
Random Access
Max. Weight
Our algorithm

- Adaptive random access based on queue-size
- ‘Simulates’ maximum weight algorithm
Our algorithm

• Each queue $i$ checks medium ‘regularly’
  ◦ Whether any ‘neighboring’ node is txing or not
  ◦ If medium is free, attempts transmission with prob. $p_i$
    • upon being successful, tx for time duration $W_i$
  ◦ Else
    • do nothing

• Our choice
  ◦ $p_i = 1$ and $\mathbb{E}[W_i] = f(Q_i)$
    • choice of $f$ determines performance crucially
    • a reasonable choice of $f$ is $\log$
Our algorithm: continuous time

- Each queue has an independent Exponential clock of rate $1/2$

- When clock of queue $i$ ticks, say at time $t$
  - If $\sigma_i(t^-) = 1$,
    \[
    \sigma_i(t) = \begin{cases} 
    0 & \text{with probability } \frac{1}{f(Q_i([t]))} \\
    1 & \text{otherwise}
    \end{cases}
    \]
  - Else, $i$ check if medium is free at time $t^-$ and if so,
    \[
    \sigma_i(t) = \begin{cases} 
    1 & \text{with probability } 1 \\
    0 & \text{otherwise}
    \end{cases}
    \]
Our algorithm: example (cont time)

\[ Q_1 = 40 \]
\[ \sigma_1 = 1 \]

\[ Q_2 = 10 \]
\[ \sigma_2 = 0 \]

\[ Q_3 = 5 \]
\[ \sigma_3 = 0 \]
Our algorithm: example (cont time)

\[ Q_1 = 40, \quad \sigma_1 = 1 \]

\[ Q_2 = 10, \quad \sigma_2 = 0 \]

\[ Q_3 = 5, \quad \sigma_3 = 0 \]
Our algorithm: example (cont time)

\[ Q_1 = 40, \quad \sigma_1 = 0 \]

\[ Q_2 = 10, \quad \sigma_2 = 0 \]

\[ Q_3 = 5, \quad \sigma_3 = 0 \]
Our algorithm: example (cont time)

$Q_1 = 40,$
$\sigma_1 = 1$

$Q_3 = 5$
$\sigma_3 = 0$

$Q_2 = 10$
$\sigma_2 = 0$

w.p. $\frac{1}{f(40)}$

$Q_1 = 40,$
$\sigma_1 = 0$

$Q_3 = 5$
$\sigma_3 = 0$

$Q_2 = 10$
$\sigma_2 = 0$
Our algorithm: example (cont time)

\[ Q_1 = 40 \]
\[ \sigma_1 = 1 \]
\[ Q_2 = 10 \]
\[ \sigma_2 = 0 \]
\[ Q_3 = 5 \]
\[ \sigma_3 = 0 \]

\[ Q_1 = 40 \]
\[ \sigma_1 = 0 \]
\[ Q_2 = 10 \]
\[ \sigma_2 = 0 \]
\[ Q_3 = 5 \]
\[ \sigma_3 = 0 \]

\[ w.p. \frac{1}{f(40)} \]
Our algorithm: example (cont time)

\[
Q_1 = 40, \quad \sigma_1 = 1
\]
\[
Q_2 = 10, \quad \sigma_2 = 0
\]
\[
Q_3 = 5, \quad \sigma_3 = 0
\]

w.p. \( \frac{1}{f(40)} \)

\[
Q_1 = 40, \quad \sigma_1 = 0
\]
\[
Q_2 = 10, \quad \sigma_2 = 0
\]
\[
Q_3 = 5, \quad \sigma_3 = 0
\]

w.p. 1
Our algorithm: example (cont time)
Our algorithm: discrete time

- Each queue has an independent Bernoulli clock of rate $1/2$

- If clock of queue $i$ ticks at time $t$, then
  - If $\sigma_i(t-1) = 1$,
    \[
    \sigma_i(t) = \begin{cases} 
    0 & \text{with probability } \frac{1}{f(Q_i(t))} \\
    1 & \text{otherwise}
    \end{cases}
    \]
  - Else, $i$ check if medium free at time $t-1$
    - if so, it attempts to transmit with probability 1
      \[
      \sigma_i(t) = \begin{cases} 
    1 & \text{if no collision} \\
    0 & \text{otherwise}
    \end{cases}
    \]
Our algorithm: throughput optimality

- **Theorem.** [S-Shin 09, 10] The algorithm is throughput optimal.
  - For both continuous and discrete time
    - ACM Sigmetrics 09 paper award

- Specifically, we establish that
  - The network Markov process is positive (Harris) recurrent

- Based on insights from
  - Reversible dynamics and product-form distributions
  - Variational characterization
  - Mixing time theory of Markov chains
Summary, thus far

Model

Policy design and Performance analysis

Implementable algorithm

Kelly et al 98
Harrison 00
S-Wischik 08
S-Moallemi 10

Tassiulas-Ephremides 92
Dai 95, Stolyar-Rybko 92
Bramson 98, Williams 98
Kelly-Williams 04
S-Wischik 06, 08, 09
S-Tse-Tsitsiklis 10

McKeown-Ananthram-Walrand 96
Giaccone-Prabhakar-S 03
S-Shin 09, 10
Summary, thus far

- Message passing algorithms are widely applicable
  - e.g., Internet routers, optical core network
  - ‘Mixing time’ of scheduling Markov chain $\approx$ Queue size

Model

Policy design and Performance analysis

Implementable algorithm

- Kelly et al 98
- Harrison 00
- S-Wischik 08
- S-Moallemi 10

- Tassiulas-Ephremides 92
- Dai 95, Stolyar-Rybko 92
- Bramson 98, Williams 98
- Kelly-Williams 04
- S-Wischik 06, 08, 09
- S-Tse-Tsitsiklis 10

- McKeown-Ananthram-Walrand 96
- Giaccone-Prabhakar-S 03
- S-Shin 09, 10
Belief Propagation for Inference in Graphical Model

with Various Collaborators
Graphical model and belief propagation

- Probabilistic graphical models
  - Succinct representation for joint distribution
  - Have been quite useful in variety of applications
    - e.g. coding, language processing, vision, bioinformatics, etc.
  - Two inference problems of interest
    - computing marginal distribution (counting)
    - mode of distribution (optimization)
  - In general, inference is computationally hard
Graphical model and belief propagation

- Probabilistic graphical models
  - Succinct representation for joint distribution
  - Have been quite useful in variety of applications
    - e.g. coding, language processing, vision, bioinformatics, etc.
  - Two inference problems of interest
    - computing marginal distribution (counting)
    - mode of distribution (optimization)
  - In general, inference is computationally hard

- Belief propagation: an umbrella heuristic for both problems
  - Approximate computation by Bethe and Peierls (1934)
  - Decoding algorithm by Gallager (1963)
  - Inference heuristic by Pearl (1980s)
Belief propagation: mode or optimization

- Let’s start with an example
  - Graph $G = (V, E), V = \{1, \ldots, n\}$
  - Edge weights $w_{ij} \geq 0$ for $(i, j) \in E$
  - Goal:
    - find maximum weight matching in $G$
    - that is,
      $$\text{maximize} \sum_{(i,j) \in E} w_{ij}x_{ij} \quad \text{over} \quad x_{ij} \in \{0, 1\},$$
      $$\text{subject to} \sum_{j \in \mathcal{N}(i)} x_{ij} \leq 1, \quad \forall \ i.$$  
  - Equivalently, find an optimal assignment of each node $i$
    - either to one of its neighbors or none
Maximum weight matching: example
BP is an iterative approximation of dynamic programming

- Obtained from tree structured graph
Recursive evaluation of “messages” on edges:

\[ B_1 = w_1 - \max(B_3, B_4, 0). \]
Maximum weight matching: BP iteration

- Iterative “message update” on edges under BP: at iteration $t + 1$,
  \[ B_{0}^{t+1} = w_0 - \max_i \left( \max_i B_i^t, 0 \right) . \]

- Well-define for any graph $G$ (even if its not a tree)
Belief propagation

• In summary
  ◦ BP is a ‘tree-based’ approximation of dynamic programming
  ◦ Hence, it is reasonable to expect that
    • BP is good approximation for ‘tree-like’ graphs

• Empirically
  ◦ It seems to do well even on ‘loopy’ structures
  ◦ And this deserves some explanation

• Finally, BP is not a solution to all problems
  ◦ That is, need to understand its limitations
Belief propagation: known properties

- BP is an iterative procedure
  - Does it have a fixed point or fixed points?
  - Are they any good?
  - Does algorithm converge to them?

- In example we considered

\[ B_{t+1}^0 = w_0 - \max_i (\max B_t^i, 0). \]

  - Inductively, it follows that

\[ -\max_i w_i \leq B_{t+1}^0 \leq \max_i w_i. \]

  - 'Iteration' is a cont. func. from convex set to convex set
    \[ \rightarrow \text{fixed point exists by Brouwer's fixed point theorem} \]
Brief history of BP

• Yedidia-Freeman-Weiss ’01 & Weiss-Freeman ’01
  ◦ Fixed points
    • exist, and
    • have certain local optimality properties

• Next, I’ll discuss scenarios when BP works
Max weight matching: bipartite graph

- Bipartite graph $G = (V_1 \times V_2, E)$:
  - $V_1 = \{\alpha_1, \ldots, \alpha_n\}$ and $V_2 = \{\beta_1, \ldots, \beta_n\}$
  - $E = V_1 \times V_2$, $w_{ij}$ be weight of $(\alpha_i, \beta_j) \in E$
  - Goal: compute Max Wt Matching (MWM) in $G$

- An example with $n = 2$

![Diagram of a bipartite graph with weights and an MWM solution]
BP for MWM: correctness and convergence

• Notation
  ◦ Let \( \epsilon \) be difference between weight of MWM and second MWM
    \[ \rightarrow \text{If MWM not unique, then } \epsilon = 0 \]
  ◦ Let \( w^* = \max_{ij} w_{ij} \)

• Theorem. [Bayati-S-Sharma 05] BP estimate converges to the correct MWM in \( \frac{2n w^*}{\epsilon} \) number of iterations.

• Implication: for fixed \( w^* \) and \( \epsilon \)
  ◦ Number of iterations scale as \( O(n) \)
  ◦ Per-node computation in each iteration \( O(n) \)
  ◦ Thus, total computation cost \( O(n^3) \)
BP for MWM: correctness and convergence

• Notation
  ◦ Let $\varepsilon$ be difference between weight of MWM and second MWM
    → If MWM not unique, then $\varepsilon = 0$
  ◦ Let $w^* = \max_{ij} w_{ij}$

• **Theorem.** [Bayati-S-Sharma 05] BP estimate converges to the correct MWM in $\frac{2nw^*}{\varepsilon}$ number of iterations.

• Implication: for fixed $w^*$ and $\varepsilon$
  ◦ Number of iterations scale as $O(n)$
  ◦ Per-node computation in each iteration $O(n)$
  ◦ Thus, total computation cost $O(n^3)$

• **Theorem.** [Salez-S 09] Under ‘random’ weights, BP converges in ‘essentially’ $O(1)$ iterations.
Correctness of BP

- An important question
  - Why does BP solve MWM in a loopy graph?

- BP and Auction algorithm (cf. Bertsekas 80)
  - BP is essentially ‘parallel’ version of Auction
  - Messages of BP are like ‘dual’ variables
  - And matching is solvable by linear programing (LP)
    [Birkhoff-Von Neumann 40s]

- In general, this suggests a relation between BP and LP
BP versus LP

- **Theorem.** [Sanghavi-S-Willsky 08, 09] If BP ‘works’ then (edge-based) LP relaxation must find correct solution for
  - matching and independent set
BP versus LP

- **Theorem.** [Sanghavi-S-Willsky 08, 09] If BP ‘works’ then (edge-based) LP relaxation must find correct solution for
  - matching and independent set

![Diagram showing the relationship between BP and LP, with layers indicating different aspects such as matching and network flow, and references to Bayati-S-Sharma 05, Sanghavi-S-Willsky 09, and Gamarnik-S-Wei 10.]
Understanding BP

• Other works in combinatorial, continuous optimization
  ○ Huang-Jebara 08, Sanghavi-Malioutov-Willsky 08
  ○ Bayati-Borgs-Chayes-Zecchina 08, Bayati-Braunstein-Zecchina 08
  ○ Gamarnik-Nowicki-Swirczcs 05, Malioutov-Johnson-Willsky 06
  ○ Moallemi-Van Roy 06, 08, ...

• My other results
  ○ Random combinatorial problems
    • Sanghavi-S 09 – independent set
    • Salez-S 09 – matching
  ○ Compressed sensing
    • Chandar-S-Wornell 10
    • Related work by Donoho-Maleki-Montanari 10
• Message passing algorithms
  ○ Random access for wireless network
    • provably throughput optimal
    • algo. complexity (mixing time) $\approx$ queue-size scaling
  ○ Belief propagation for optimization or mode estimation
    • solves an important class of linear programming problems
    • suggests why it may be working well empirically

• Message passing algorithms will play an important role
  ○ They’ll be useful in designing and modeling networked systems
    • including engineering, statistical and natural systems
  ○ We have started understanding specific instances of them