A Local Scalable Distributed Expectation Maximization Algorithm for Large Peer-to-Peer Networks

Kanishka Bhaduri[1] Ashok N. Srivastava[2]

[1]MCT Inc., Intelligent Data Understanding
NASA Ames Research Center, Moffett Field CA-94035

[2]Intelligent Data Understanding
NASA Ames Research Center, Moffett Field CA-94035

IEEE ICDM 2009

Roadmap

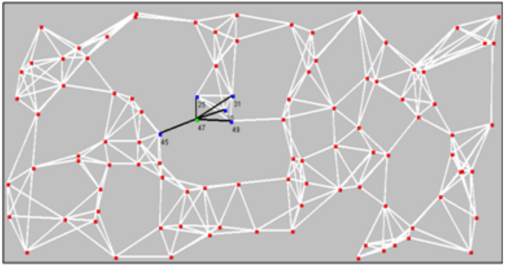


Figure: A P2P network

- Highlights:
  - Highly scalable
  - Asynchronous
  - Completely decentralized
  - Ad-hoc connections

Introduction

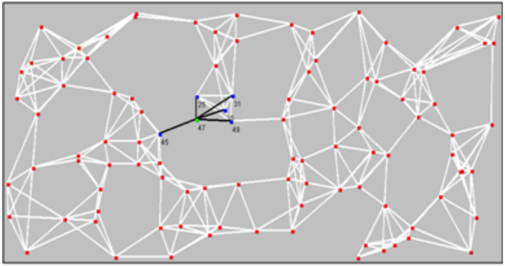


Figure: A P2P network

- Highlights:
  - Highly scalable
  - Asynchronous
  - Completely decentralized
  - Ad-hoc connections

Data mining in P2P networks?

P2P setup

- Millions of peers (Skype $\sim$ 50 million)
- Dynamic topology and data — peers can join/leave at any time
- No global clock — completely asynchronous
- Same features across all peers
- Communication — reliable, bandwidth-limited, asynchronous, asymmetric
- Impracticalities / impossibilities
  - global communication
  - global synchronization

An example

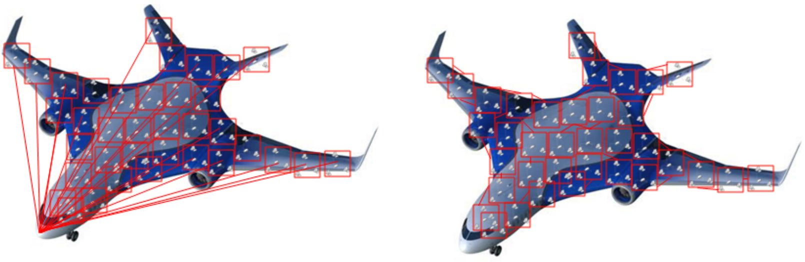


Figure: Centralized vs. in-network computation

Motivation

- EM very useful for variety of data mining tasks
- Can be deployed in P2P networks for
  - clustering
  - anomaly detection
  - target tracking
  - inferencing
- Centralizing data expensive/impractical; collaborative computing *e.g.* cloud computing can harness power of multiple processors/storage

Motivation

- EM very useful for variety of data mining tasks
- Can be deployed in P2P networks for
  - clustering
  - anomaly detection
  - target tracking
  - inferencing
- Centralizing data expensive/impractical; collaborative computing *e.g.* cloud computing can harness power of multiple processors/storage

Can we develop an EM algorithm for P2P networks?

Related work

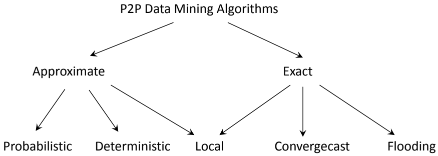


Figure: Distributed data mining algorithms

- S. Datta, K. Bhaduri, C. Giannella, R. Wolff, H. Kargupta. Distributed Data Mining in Peer-to-Peer Networks. IEEE Internet Computing Vol. 10(4), 2006.
- S. Datta, H. Kargupta. Uniform Data Sampling from a Peer-to-Peer Network. ICDCS 2007.
- S. Mukherjee, H. Kargupta. Distributed Probabilistic Inferencing in Sensor Networks using Variational Approximation. JPDC Vol. 68(1), 2008.
- K. Bhaduri, R. Wolff, C. Giannella, H. Kargupta. Distributed Decision Tree Induction in Peer-to-Peer Systems. Statistical Analysis and Data Mining. Vol. 1(2) 2008.

Problem statement

- Consider large P2P network
  - each node has local data which change over time
  - each node can exchange messages with immediate neighbors

Goal

Fit and monitor a gaussian mixture model (gmm) via EM to global data

- Constraints:
  - communication-efficient and scalable
  - asynchronous
  - able to handle dynamic data and network
  - *provably* correct result compared to centralized computation

Contribution

- Algorithm for monitoring gmm parameters using EM in large P2P networks
  - *local* and highly scalable
  - asynchronous
  - *provably* correct
  - seamlessly handles changes in the data and network

What is locality?

- Every node communicates with only fixed number of other nodes
- Bounded total query size
- Advantages:
  - Scalable
  - Fault-tolerant
  - Robust

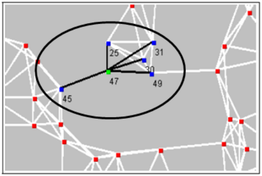


Figure: Locality of distributed algorithms

What is locality?

- Every node communicates with only fixed number of other nodes
- Bounded total query size
- Advantages:
  - Scalable
  - Fault-tolerant
  - Robust

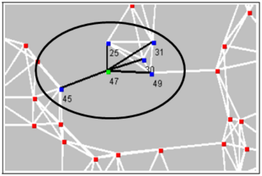


Figure: Locality of distributed algorithms

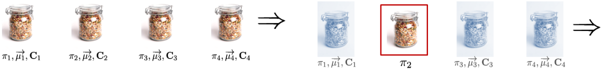
Local algorithms

For data dependent algorithms, there exist problem instances whose resource consumption is constant, independent of network size

Expectation maximization

$\pi_1, \vec{\mu_1}, \mathbf{C}_1$ $\pi_2, \vec{\mu_2}, \mathbf{C}_2$ $\pi_3, \vec{\mu_3}, \mathbf{C}_3$ $\pi_4, \vec{\mu_4}, \mathbf{C}_4$ $\Rightarrow$

Expectation maximization



Expectation maximization

Figure: Expectation Maximization

Expectation maximization

Figure: Expectation Maximization

- Given $\mathbf{X} = \{\overrightarrow{x_1}, \overrightarrow{x_2}, \ldots, \overrightarrow{x_n}\}$, where $\overrightarrow{x_i} = \mathcal{N}(\overrightarrow{\mu}, \mathbf{C}_s)$
- Goal: estimate parameters $\Theta = \{\overrightarrow{\mu_1}, \ldots, \overrightarrow{\mu_k}, \mathbf{C}_1, \ldots, \mathbf{C}_k, \pi_1, \ldots, \pi_k\}$
- Approach: maximize log-likelihood of parameters given $\mathbf{X}$

Expectation maximization

Update equations

E-step (estimate the contribution of each point towards each gaussian):

$$q_{s,a} = \frac{\pi_s \mathcal{N}(\vec{x_a}; \vec{\mu_s}, \mathbf{C}_s)}{\sum_{r=1}^{k} \pi_r \mathcal{N}(\vec{x_a}; \vec{\mu_r}, \mathbf{C}_r)}$$

no communication

M-step (recompute the parameters of each gaussian):

$$\pi_s = \frac{\sum_{a=1}^{n} q_{s,a}}{n}$$

$$\vec{\mu_s} = \frac{\sum_{a=1}^{n} q_{s,a} \vec{x_a}}{\sum_{a=1}^{n} q_{s,a}}$$

communication

$$\mathbf{C}_s = \frac{\sum_{a=1}^{n} q_{s,a} (\vec{x_a} - \vec{\mu}_s)(\vec{x_a} - \vec{\mu}_s)^{\mathrm{T}}}{\sum_{a=1}^{n} q_{s,a}}$$

Notations

- $P_1, \ldots, P_p$ — a set of peers
- Data stream at P_i

$$S_i = [\overrightarrow{x_{i,1}}, \overrightarrow{x_{i,2}}, \ldots, \overrightarrow{x_{i,m_i}}]$$

- Global input $\mathcal{G} = \bigcup_{i=1,\ldots,p} S_i$
- $X_{i,j}$: messages sent by P_i to P_j

Notations

- $P_1, \ldots, P_p$ — a set of peers
- Data stream at P_i

$$S_i = [\overrightarrow{x_{i,1}}, \overrightarrow{x_{i,2}}, \ldots, \overrightarrow{x_{i,m_i}}]$$

- Global input $\mathcal{G} = \bigcup_{i=1,\ldots,p} S_i$
- $X_{i,j}$: messages sent by P_i to P_j

Goal

Build and monitor gmm model on $\mathcal{G}$ without collecting $\mathcal{G}$

Thresholding problem

- Problem 1: Compute gmm parameters
  - O(n) communication for exact computation $\times$
- Problem 2: Given pre-computed parameters, monitoring them vs. $\mathcal{G}$
  - Less than O(n) communication...very efficient $\checkmark$

Thresholding problem

- Problem 1: Compute gmm parameters
  - O(n) communication for exact computation $\times$
- Problem 2: Given pre-computed parameters, monitoring them vs. $\mathcal{G}$
  - Less than O(n) communication...very efficient ✓
- Sufficient statistics
  - Knowledge: $\mathcal{K}_i = S_i \bigcup_{P_j \in \Gamma_i} X_{j,i}$
  - Agreement: $\mathcal{A}_{i,j} = X_{i,j} \cup X_{j,i}$
  - Withheld: $\mathcal{W}_{i,j} = \mathcal{K}_i \setminus \mathcal{A}_{i,j}$

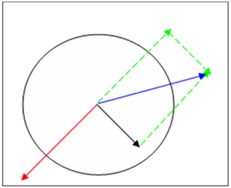


Figure: Set statistics

Geometric interpretation

Conflicting objectives:

- For correct computation, $\mathcal{K}_i = \mathcal{G}$
- For communication efficient solution, $\mathcal{K}_i \neq \mathcal{G}$

Geometric interpretation

Conflicting objectives:

- For correct computation, $\mathcal{K}_i = \mathcal{G}$
- For communication efficient solution, $\mathcal{K}_i \neq \mathcal{G}$

Solution

- Decompose domain into several non-overlapping convex regions such that any function computed on $\mathcal{G}$ remains invariant inside each convex region

- Even if $\mathcal{K}_i \neq \mathcal{G}$, $\mathcal{F}(\mathcal{K}_i) = \mathcal{F}(\mathcal{G})$ inside any such region
  - Example: Is $||\mathcal{G}|| < \epsilon$?
- Still nobody knows $\mathcal{G}$...

Local criterion

- Need conditions on local set statistics to infer about $\mathcal{G}$

Theorem

For each peer and each of its neighbors, if all its set statistics $\mathcal{K}_i$, $\mathcal{A}_{i,j}$, $\mathcal{W}_{i,j}$ are in **same** convex region, **then so is** $\mathcal{G}$

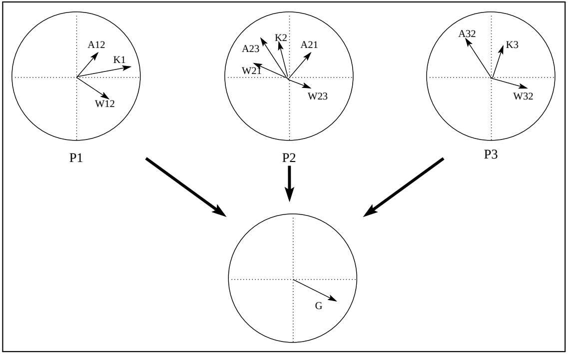


Figure: An example

Local criterion

- Allows a peer to terminate computation and communication whenever stopping condition is satisfied irrespective of other conditions
- Still guarantees eventual correctness
- Remarkably efficient in pruning messages
- Allows a peer to sit idle until an event occurs:
    - send or receive message
    - change in local data
    - change in immediate neighborhood

Back to EM

Monitoring algorithm:

1. Input: local dataset, precomputed parameters, error threshold ϵ
2. Goal: monitor $\mathcal{L}(\Theta)$, π, $\overrightarrow{\mu}$, $\mathbf{C}$
3. Initialization
   - $S_i = \left\{ q_{i,s,a} \left(\overrightarrow{x_{i,a}} - \widehat{\overrightarrow{\mu}}_s \right) \right\}$
   - Compute sufficient statistics vectors
   - Define convex regions

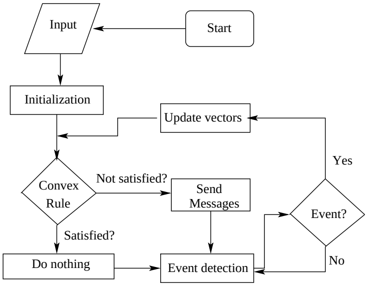


Figure: Flowchart of algorithm

Computing EM models

- Monitoring algorithm raises an alarm on correct detection
- For closed-loop solution, sample data, rebuild model
- Non-local solution — correctness of monitoring algorithm minimizes false dismissals and false alarms

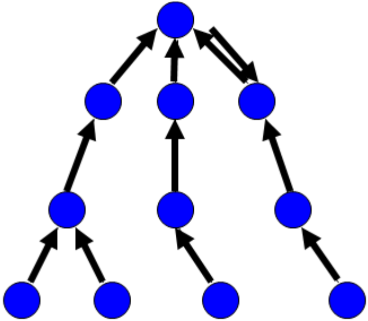


Figure: Convergecast

Monitoring results

- Simulated data consists of multivariate correlated gaussians with arbitrary parameters
- Parameters changed at fixed simulator intervals

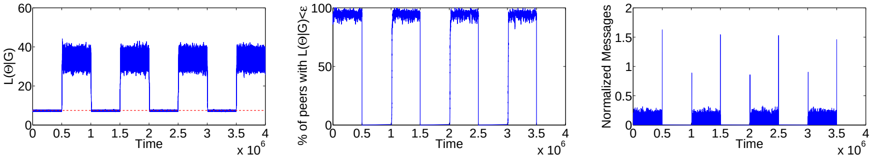


Figure: Experimental results in monitoring mode

Closed loop results

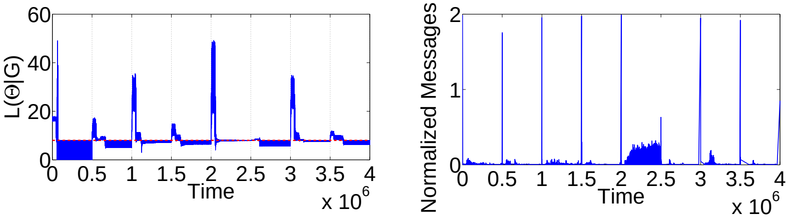


Figure: Experimental results in closed loop mode

Scalability results

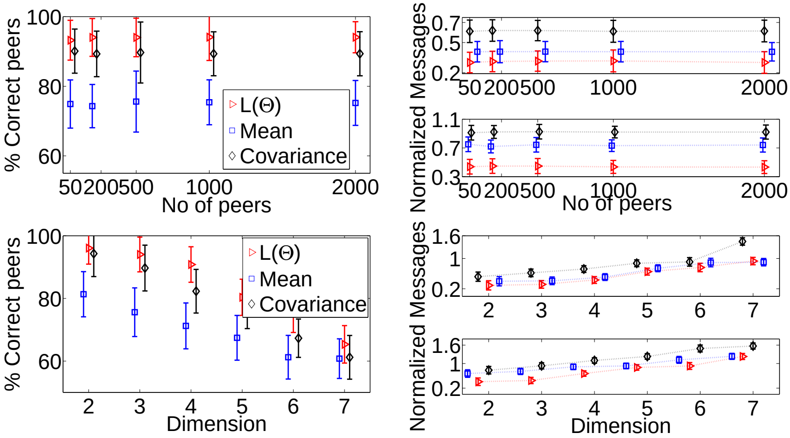


Figure: Scalability results

Conclusion

- First work on developing a local algorithm for gmm monitoring
- Algorithm provably correct, communication efficient, highly scalable, in-network and asynchronous
- Extensive experimental results show low communication cost and correctness of results

Resources:

- `http://ti.arc.nasa.gov/profile/kbhaduri/`
- Distributed Data Mining Bibliography: `http://www.csee.umbc.edu/~hillol/DDMBIB/`