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

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

[1] MCT Inc. at NASA Ames Research Center, MS 269/3, Moffett Field, CA-94035
[2] MS 269/3, NASA Ames Research Center, Moffett Field, CA-94035

Summary

- Motivation
  - Peer-to-peer data mining growing area of research for analyzing data content, modeling user behavior and computing network statistics
  - Expectation maximization useful for data clustering, anomaly detection, target tracking, and density estimation
- Contribution
  - Provably correct *local* distributed asynchronous algorithm for monitoring gaussian mixture model parameters in peer-to-peer networks using expectation maximization
  - Shows how second order statistics can be directly monitored in a peer-to-peer network
- Application
  - Clustering and fault isolation in (1) large scale sensor networks such as embedded aircraft sensors on systems and subsystems, (2) national air space for identifying anomalous aircrafts

Motivation

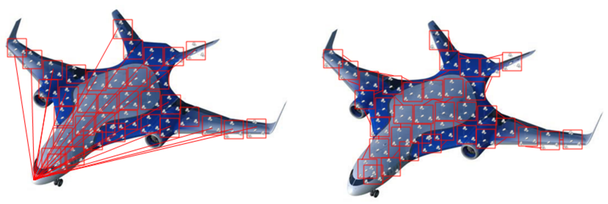

Figure: Centralized vs. in-network computation

Overview of approach

- Monitoring phase
  - Each peer maintains estimate of prior probability, mean, covariance of the global (all peers') data
  - When data changes, peers jointly track this change
  - Triggers alarm if model is outdated with respect to global data
- Computation phase

  Convergecast: Sample data from network, build new model

  Broadcast: Send new model to all peers

Background

- Input
  - Data stream at each peer $S_i = [\overrightarrow{x_{i,1}}, \overrightarrow{x_{i,2}}, \ldots, \overrightarrow{x_{i,m_i}}]$
  - k gaussians
  - Global input $\mathcal{G} = \bigcup_{i=1,\ldots,p} S_i$
  - $X_{i,j}$: messages sent by P_i to P_j
- Set statistics
  - Knowledge of P_i: $\mathcal{K}_i = S_i \bigcup_{P_j \in \Gamma_i} X_{j,i}$
  - Agreement of P_i, P_j: $\mathcal{A}_{i,j} = X_{i,j} \cup X_{j,i}$
  - Withheld knowledge of P_i, P_j: $\mathcal{W}_{i,j} = \mathcal{K}_i \setminus \mathcal{A}_{i,j}$
  - Message: $X_{i,j} = \mathcal{K}_i \setminus X_{j,i}$

Convex stopping rule

- For each P_i and for every $P_j \in \Gamma_i$, if
  - $\overline{\mathcal{K}_i} \in R$
  - $\overline{\mathcal{A}_{i,j}} \in R$
  - $\overline{\mathcal{W}_{i,j}} \in R$ or $\mathcal{W}_{i,j} = \emptyset$

then $\overline{\mathcal{G}} \in R$

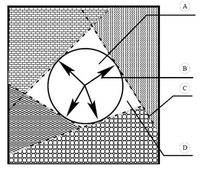

Figure: Convex regions

Expectation maximization

Log-likelihood:

$$\overline{\mathcal{L}}(\Theta|\mathcal{G}) = \frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} \log\left(\sum_{s=1}^{k} \pi_s \mathcal{N}(\overrightarrow{x_{i,a}}; \overrightarrow{\mu_s}, \mathbf{C}_s)\right)}{\sum_{i=1}^{p} m_i}$$

E-step:

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

M-step:

$$\pi_s = \frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}}{\sum_{i=1}^{p} m_i}$$

$$\overrightarrow{\mu_s} = \frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}\overrightarrow{x_{i,a}}}{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}}$$

$$\mathbf{C}_s = \frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}(\overrightarrow{x_{i,a}} - \overrightarrow{\mu}_s)(\overrightarrow{x_{i,a}} - \overrightarrow{\mu}_s)^{\mathrm{T}}}{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}}$$

Algorithm details

- $\overline{\mathcal{L}}(\widehat{\Theta}|\mathcal{G}) = \frac{\sum_{i=1}^{p}\mathcal{L}_i(\widehat{\Theta}|S_i)}{\sum_{i=1}^{p} m_i} < \epsilon$
- $Err(\pi_s) = |\pi_s - \widehat{\pi}_s| < \epsilon_1$
- $Err(\overrightarrow{\mu_s}) = \left|\left|\overrightarrow{\mu_s} - \widehat{\overrightarrow{\mu}}_s\right|\right| = \left|\left|\frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}\left[\overrightarrow{x_{i,a}} - \widehat{\overrightarrow{\mu}}_s\right]}{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}}\right|\right| < \epsilon_2$
- $Err(C_s^n) = \sum_{k=1}^{d}\left\{\frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}x_{i,a,k}^2}{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}} - \left(\frac{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}x_{i,a,k}}{\sum_{i=1}^{p}\sum_{a=1}^{m_i} q_{i,s,a}}\right)^2\right\} - \widehat{C}_s < \epsilon_3$

Example: For for monitoring $\overrightarrow{\mu_s}$ input: $S_i = \left\{q_{i,s,a}\left(\overrightarrow{x_{i,a}} - \widehat{\overrightarrow{\mu}}_s\right)\right\}$

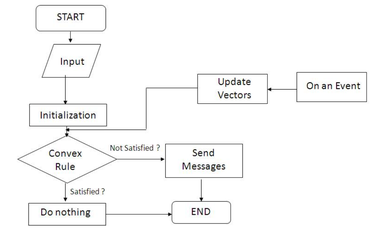

Figure: Flowchart of algorithm

Synthetic data experiments

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

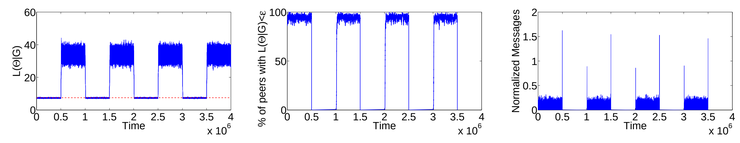

Figure: Experimental results in monitoring mode

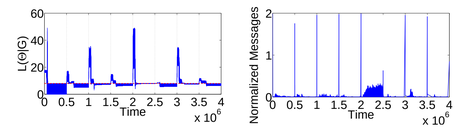

Figure: Experimental results in closed loop mode

Conclusions

- First algorithm for monitoring gaussian mixture model parameters in a local completely decentralized fashion
- Extensive experimental results show low communication cost and correctness of results