Distributed Iterative Statistical Computing Framework and The Distributed Expectation Maximization Algorithm

Background
- Modern distributed computing architectures facilitate storage of complex and large data (distributed data) on multiple machines.
- Distributed data pose nontrivial problems for comprehensive data analysis and the design of computational environments (Cleveland et al., 2012).
- Similar to the scalable bootstraps of Kleiner et al. (2012), classical statistical algorithms and procedures need to adapt to the distributed data setting, where they can access only a subset of the distributed data, while retaining their generic applicability and theoretical guarantees.
- We present a framework and software package in R (R Development Core Team, 2012), Distributed Iterative Statistical Computing (disc), that is tuned to address the challenges posed by distributed data in their analysis, their visualization, and especially in performing iterative computations with them.

Components of the disc framework:
- User implements the iterative algorithm in R using the disc framework.
- Master cluster machine that manages the tasks and serves as channel of communication between user and cluster.
- Slave nodes on the clusters, manage local R sessions and computations with local version of data.
- Graphical device to facilitate data analysis and visualization in the distributed setting (optional component).

MapReduce Framework and Hadoop For Embarrassingly Parallel Computation
- MapReduce framework of Dean and Ghemawat (2008) and Hadoop are generic programming tools for embarrassingly parallel computations with distributed data, which can be applied to a wide variety of problems.
- Bootstrap has been adapted to scale to massive data by Klein et al. (2012) and it fits perfectly in the embarrassingly parallel computations of MapReduce.
- Smola and Naranarathur (2010) modify the computer architecture to scale iterative computations.

Distributed Iterative Statistical Computing (disc) Framework And Package
- Assumptions of the disc framework:
  - Inference and parameter estimates for the whole distributed data are desirable.
  - Iterative algorithms can access only a subset of data that are stored across different machines.
  - Inference and parameter estimates for the whole distributed data are desirable.
  - Iterative algorithms can access only a subset of data that are stored across different machines.
  - Optimization method to estimate parameters.

Features of the disc framework and package:
- User-friendly and generic framework for implementing iterative algorithms for computing with distributed data.
- Communication between machines is expensive and should be minimized for computational efficiency.
- Iterative algorithms can access only a subset of data that are stored across different machines.

Distributed EM Algorithm Example: Mixture of Three Normals

Model: $\pi_1N(\mu_1, \sigma_1^2) + \pi_2N(\mu_2, \sigma_2^2) + \pi_3N(\mu_3, \sigma_3^2)$

- D-E Step:
  Compute complete data sufficient statistics from the local version of the data on the slaves:
  $$S_2^i = \sum_{j=1}^{n_i} \left[z_{ij} \left(\frac{x_{ij} - \mu_2}{\sigma_2}\right)^2 + \left(1 - z_{ij}\right) \left(\frac{x_{ij} - \mu_1}{\sigma_1}\right)^2\right]$$
  $$S_3^i = \sum_{j=1}^{n_i} z_{ij} x_{ij}^2$$
  $$S_4^i = \sum_{j=1}^{n_i} z_{ij} x_{ij}^4$$

- D-M Step:
  Update the global parameter estimates using the local sufficient statistics when a large percentage of sufficient statistics have been returned to the master.

Distributed Monte Carlo EM Algorithm Example: Survival Analysis
- Often E-Step is analytically intractable in EM algorithm, so we replace the E-Step by Monte Carlo E-Step, which is computationally expensive.
- Example: Lifetimes of parts are assumed to have Weibull distribution and are observed with censoring.
  $$f(x) = abx^{b-1} \exp(-ax^n)$$
  $$\lambda = \sum_{i=1}^{n} \pi_i$$

- Data: Observe $N$ lifetimes of which $N_1$ are censored that are observed by $\{z_i\}$.

- D-E Step:
  $$\pi^{(k)} \sim \text{Uniform}(0, 1)$$
  $$S_2^{(k)} = E[z_i \hat{a}^{(k)}] = \left(\sum_{i=1}^{N} n_i \pi_i \hat{a}^{(k)} \right)^{-1}$$

- D-M Step:
  $$\hat{a} = \sum_{i=1}^{N} n_i \pi_i \hat{a}^{(k)} - \sum_{i=1}^{N} n_i \pi_i \hat{a}^{(k+1)}$$
  $$\log \text{likelihood}$$
  $$\text{Optimization method to estimate } \hat{b}.$$

- Ran 10,000 iterations with 6 slaves (20 nodes) with half the observations are censored.

The D-MCEM algorithm is exponentially faster than the vanilla MCEM algorithm on the exponential time scale with exponentially increasing data size.

Future Work
- Derive the convergence rate of the D-EM algorithm.
- Expand ideas to other iterative algorithms, such as MCMC.

References
- Dlugosz, M., Smola, A. J., and Srivastava, S. (2012). Distributed Iterative Statistical Computing (disc), that is tuned to address the challenges posed by distributed data in their analysis, their visualization, and especially in performing iterative computations with them.