Scaling Bayesian Network Parameter Learning with Expectation Maximization using MapReduce

Erik Reed and Ole Mengshoel

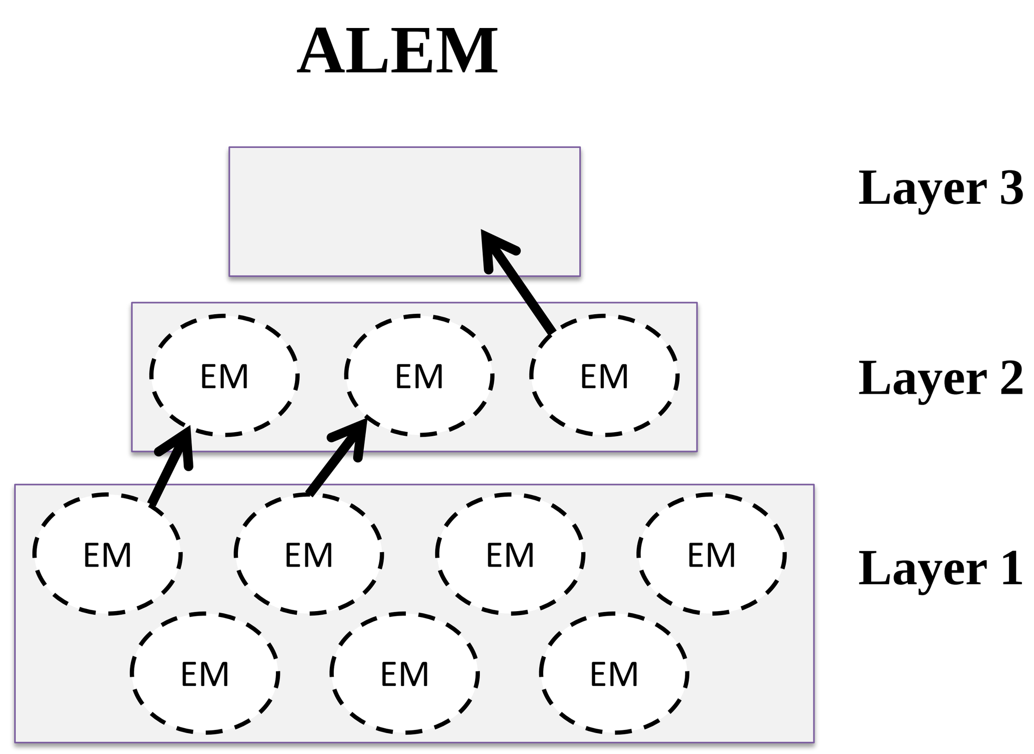
Carnegie Mellon University Silicon Valley

NIPS

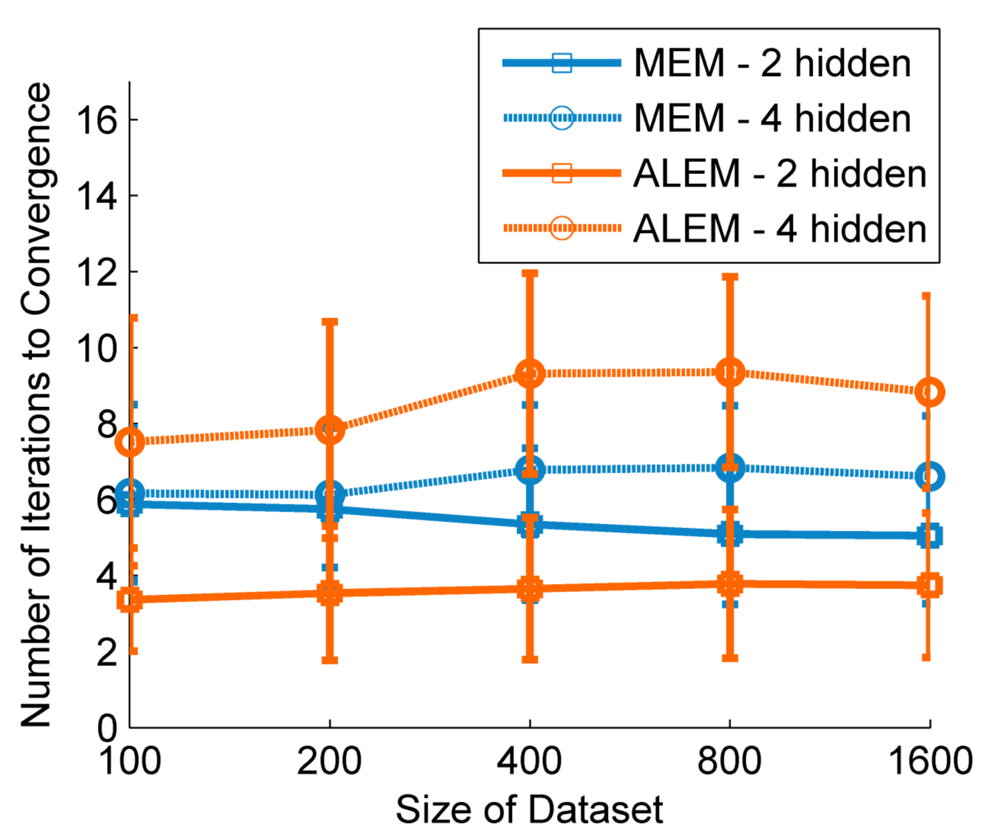
Introduction and Motivation

In order to speed up Bayesian network (BN) parameter learning from incomplete data, we use Expectation Maximization (EM) and MapReduce (MR). We investigate two EM variants: Multiple Expectation Maximization (MEM) and Age-Layered Expectation Maximization (ALEM). Both algorithms create a population of EM runs, distributing both the dataset and EM runs to a cluster of machines, achieving good solution quality and computation time.

EM converges to local optima and often requires running many EM trials to attain a high-likelihood solution. ALEM (to the right) uses a competitive population structure to restart low-likelihood EM runs.



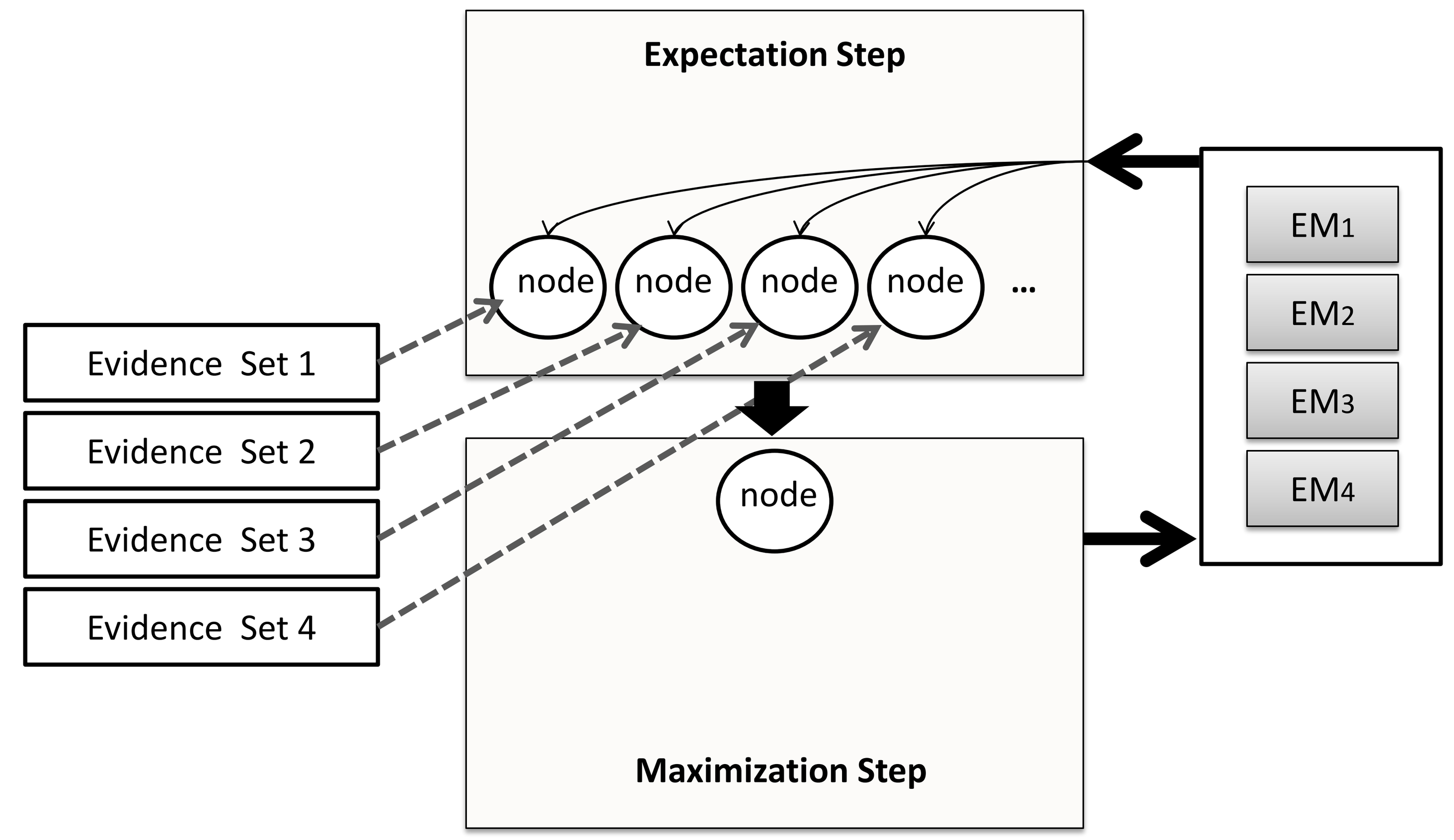
The computation time of EM is also linear with respect to size of the dataset. Reducing the number of EM iterations while maintaining a similar log-likelihood is a desirable trait of ALEM (to the right).



The E-step of the EM algorithm can be computed for each data point in parallel and reduced in the M-step. This is ideal for a multi-core and distributed MapReduce environment.

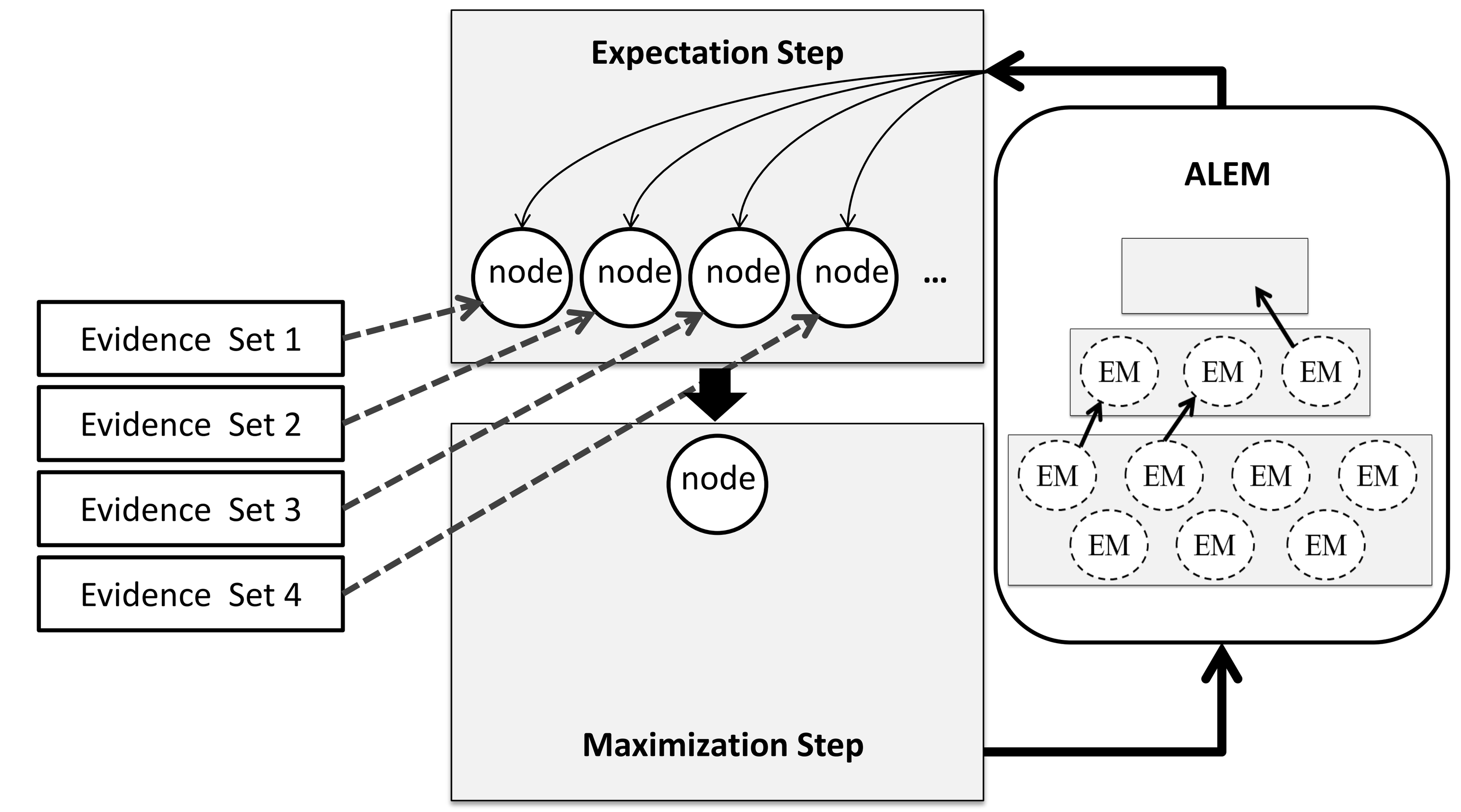
Supported, in part, by NSF awards CCF0937044 and ECCS0931978

MEM with MapReduce



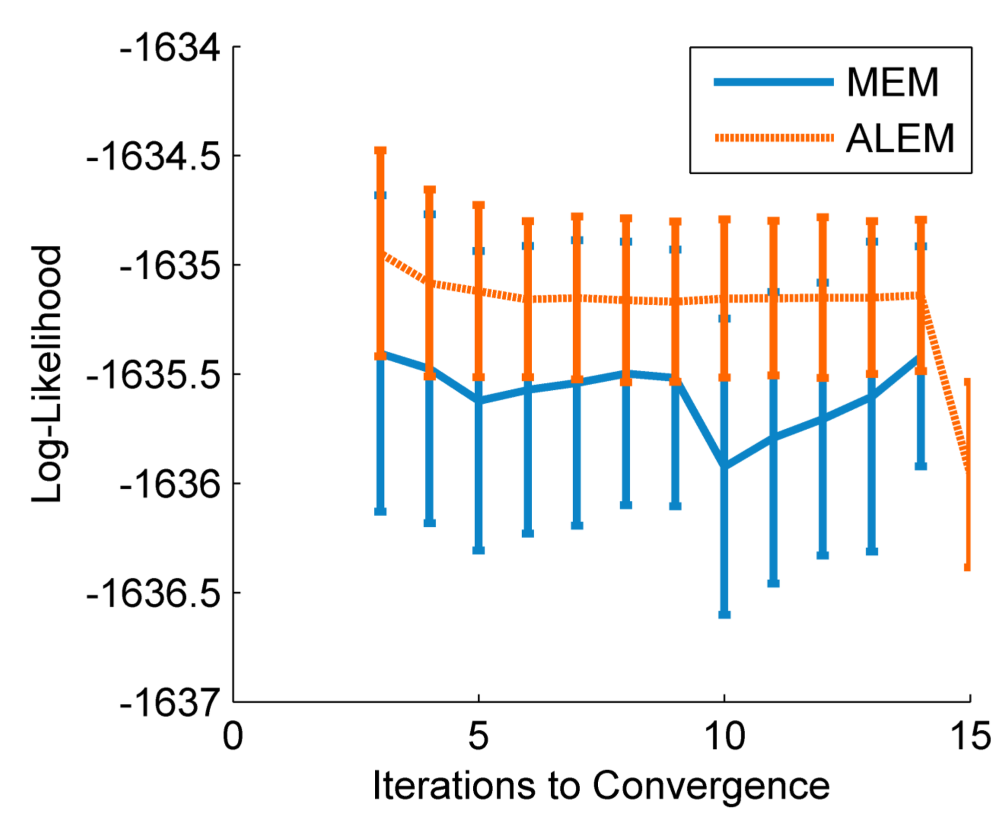
MEM with MR initializes a set of *n* EM runs which are propagated across *k* mappers. The dataset is also split into *k* chunks and distributed. The mappers compute the E-step and send the results to a single reducer, which performs maximum likelihood estimation.

ALEM with MapReduce

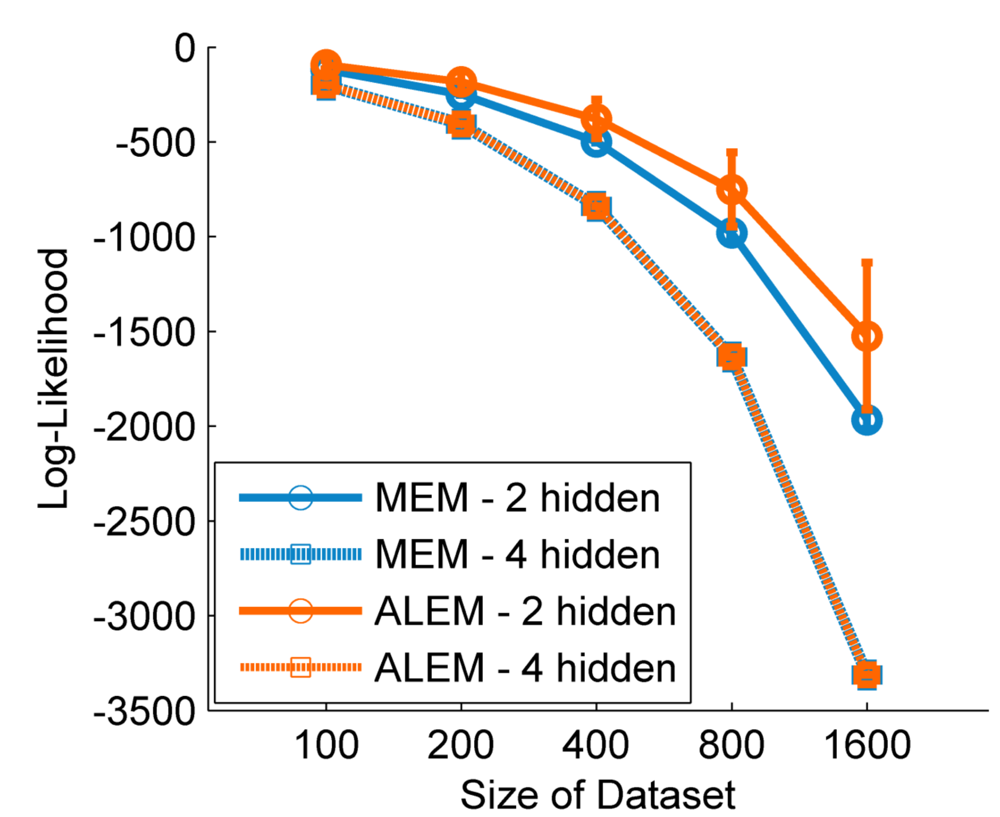


ALEM with MR initializes a population of EM runs, but the number of runs can change as EM runs are terminated or restarted. The reducer additionally performs the population management and likelihood checking of all the EM runs.

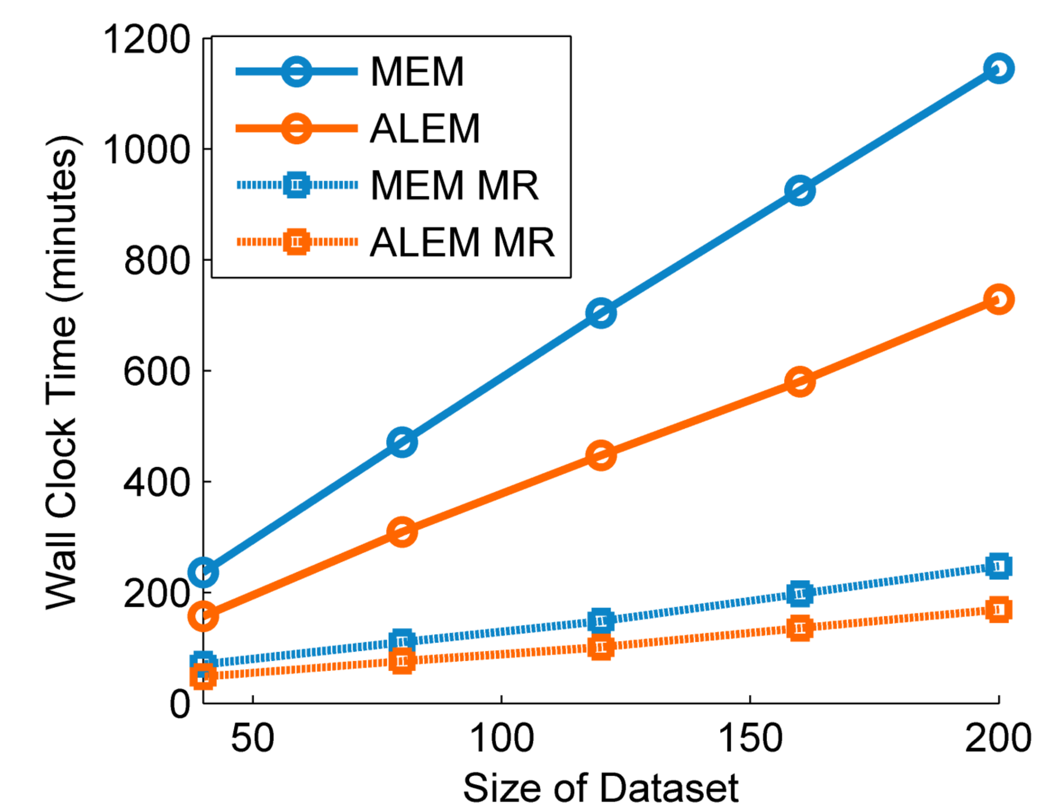
Experiments and Results



Comparison of likelihood performance of MEM/ALEM on MR for the BN *Asia* for 10k trials. In this case, ALEM achieves a higher mean likelihood regardless of iterations to convergence.



We vary dataset size and perform 10k MEM/ALEM trials were performed. For two hidden nodes, ALEM/MEM attain similar likelihoods, while four hidden nodes results in higher ALEM likelihood.



Wall clock times are compared for MR and non-MR trials using 16 Amazon EC2 small instance machines. In both cases, ALEM showed speed improvements, amplified further by employing MR.