CSE599 Project Proposal

David Leen, Brian Walker

February 4, 2013

Project Title: Strategies for Parallelizing Sequential Algorithms
Teammates: David Leen and Brian Walker
Data set: Potential data sets include:

- Click Prediction dataset from HW1 & KDD cup datasets
- Predicting risk from financial reports with regression (Shotgun paper)
- Large scale learning challenge datasets: `http://largescale.ml.tu-berlin.de/instructions/`
- Various other datasets used in the Hogwild and Shotgun papers

Project Idea:

We will compare parallel implementations of several algorithms which are generally considered sequential. Specifically, we will look at the Hogwild!, Shotgun and distributed averaging algorithms and how parallelization with these algorithms improves performance. We will compare the Hogwild!, Shotgun and distributed averaging algorithms against one another on similar data sets to compare the speedup and runtimes. We will use a sequential implementation as a baseline against which to compare the parallel versions.

Not all of these algorithms parallelize in the traditional sense. The Hogwild! algorithm takes advantage of sparsity to parallelize stochastic gradient descent. The idea is simple. With sparse data, individual SGD steps only update a small part of the weights. In Hogwild! each processor is allowed to modify the weights in parallel without locking, relying on the sparsity of the updates to prevent processors overwriting one another. Even when overwriting occurs the error introduced is minor. The Shotgun algorithm uses coordinate descent and parallelizes over the features rather than over the samples. This is a nice contrast to the Hogwild! algorithm. Shotgun makes p coordinate updates in parallel. Finally, distributed averaging uses each processor to optimize a subset of the data and averages to combine the result. The algorithms will be compared on computers with 2, 4, and 8 cores to see how the speed up scales with computer resources.

Software:

The software will primarily be written in Java. A total of four algorithms will need to be written. Three of these algorithms will be the parallelized versions discussed in the papers and the fourth will be an iterative algorithm with which we can compare against the parallelized versions. We will also need to write up a framework to run the tests for each of the algorithms and the code to keep track of the statistics and runtimes for the algorithm runtime analysis.

Papers:

1. Bradley, Joseph et al. "Parallel Coordinate Descent for L1-Regularized Loss Minimization" , International Conference on Machine Learning (2011), `http://select.cs.cmu.edu/publications/scripts/papers.cgi?Bradley+al:icml11parlasso`
2. Boyd, Stephen et al. "Fast Linear Iterations for Distributed Averaging", Systems and Control Letters:53:65-78 (2004) `http://www.stanford.edu/~boyd/papers/pdf/fastavg.pdf`
3. Niu, Feng, et al. "HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent." arXiv:1106.5730v2 (2011), `http://pages.cs.wisc.edu/~brecht/papers/hogwildTR.pdf`
4. Zhang, Yuchen, et al. "Communication-Efficient Algorithms for Statistical Optimization." arXiv:1209.4129v1 (2012), `http://arxiv.org/pdf/1209.4129.pdf`

Milestone:

For the milestone we will implement the Hogwild! algorithm and demonstrate an approximately linear speedup over the sequential algorithm. We will initially run Hogwild! for the Click Prediction logistic regression problem from homework one before moving onto different classification algorithms. We will report the results of running the algorithms on machines with a different number of processor cores to compare how the running times are affected by processor resources.