Project ideas

John Langford, Microsoft Resarch, NYC

Large Scale Learning Class, March 25, 2012
(Post Presentation version)

Parallelize Regularized Greedy Forests

Rie Johnson http://riejohnson.com/index.html implemented http://stat.rutgers.edu/home/tzhang/software/rgf/ regularized greedy forests http://arxiv.org/abs/1109.0887 which are plausibly better than boosted decision trees in some ways.

Idea: Make them work in parallel using allreduce.

Code: 15876 lines of c++ with a GPL license.

Support: John Langford (jl@hunch.net) can consult on algorithmic parallelization. This is difficult, but it could be very effective if done well.

Parallel Matrix Factorization in VW

Vowpal Wabbit http://hunch.net/~vw has matrix factorization.
VW has reductions.
VW has parallel linear learning.

Idea: Make the matrix factorization work by reduction to linear learning (and hence in parallel).

Code: 305 lines of c++ for MF with a BSD license. All of VW is 16907 lines of C++ w/ BSD license.

Support: Jake Hofman (jmh@microsoft.com) and John Langford (jl@hunch.net) will consult.

Optimized Allreduce

The allreduce in VW is a baseline implementation.

1. An avoidable barrier between "reduce" and "broadcast".
2. Tree should respect IP boundaries.
3. Tree should be multirooted to optimize bandwidth usage.

Optimize it.

Benefit: Synchronization is a critical primitive for ML, so this should benefit many.

Code: 689 lines of C++ with a BSD license.

Support: Alekh Agarwal (alekha@microsoft.com) and John Langford (jl@hunch.net) will consult.

Propensity Scoring for Contextual Bandits

When learning with exploration, recorded probabilities are often wrong or features accidentally contain information about the random choice taken. The black box solution is to directly estimate the probability of an action.

Benefit: More robust contextual bandit learning.

Code: 728 lines of C++ in Vowpal Wabbit with a BSD license. Must also read ahead for the class.

Support: Miro Dudik (mdudik@microsoft.com) and John Langford (jl@hunch.net) will consult.

Conditional Probability Tree

The conditional probability tree is an algorithm for logarithmic time prediction of class probabilities. Implement it, and then improve it.

Benefit: Who can argue with an exponential reduction in computational complexity?
A: Anyone for whom it does not work.

Code: Vowpal Wabbit (BSD license with C++). This would be a new reduction, for which there are many examples now. Must also read ahead for the class.

Support: Alina Beygelzimer (beygel@gmail.com) and John Langford (jl@hunch.net) will consult.