

My Flow Chart for Learning Optimization

1. Choose **efficient effective** algorithm
2. Use **compact binary** representations
3. **goto 1**

How to Parallelize

1. If (Computationally Constrained)
2. then GPU/FPGA/ASIC (Problem: server nonstandard)
3. else
 - (a) If low pass algorithm
 - (b) then Map-Reduce
 - (c) else Research Problem.

Feature Split?—Singh

1. Supports **nonlinear/nonconvex** optimization.
2. **Minimizes delay** (= minimizes regret).
3. Only **byte/example/node**.
4. No model consistency issues.

Example Split?—Slav

1. **Less bandwidth** (maybe).
2. At least with **convex functions**, averaging is often sane.
3. Programming is easy—**reuse sequential** algorithm directly.

Communication Complexity?

1. $O(\text{feature})$ — Xiao?
2. $O(\text{example})$ — Ye
3. $O(\text{parameter})$ — Petrov

Consensus Question

How do we best subvert Map-Reduce for Machine Learning?

How should a cluster node be constructed?

1. Local Storage

2. GPU?

Huge/Large/Fast/Very X

1. "This dataset is huge/larger than anything I could previously solve."
2. "It works much faster than my algorithm used to work."

Not much information... Maybe Input complexity / time?

What's a **cloud** (for ML)?

1. A commercial-access cluster?
2. Cheaper on-demand cluster?