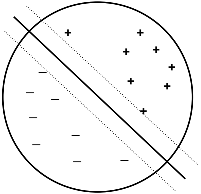
Algorithms and hardness results for parallel large-margin learning

Phil Long (Google) and Rocco Servedio (Columbia)

Setting:



- Target: γ-separated halfspace
- Domain: unit ball in $\mathbf{R}^n$
- Computational model: PRAM (parallel RAM)
- Question: can output an ϵ-accurate hypothesis using
  - $\mathrm{poly}\left(\log n, \log\frac{1}{\gamma}, \log\frac{1}{\epsilon}\right)$ time
  - $\mathrm{poly}\left(n, \frac{1}{\gamma}, \frac{1}{\epsilon}\right)$ processors?

Poster T068

Positive result

Dependence on $1/\epsilon$ already handled by Boost-by-majority [Fre95]. Revised goal:

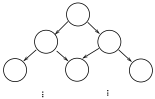
- $\mathrm{poly}\left(\log n, \log \frac{1}{\gamma}\right)$ time
- $\mathrm{poly}\left(n, \frac{1}{\gamma}\right)$ processors.

Algorithm	**Number of processors**	**Running time**
Perceptron	$\mathrm{poly}(n, 1/\gamma)$	$\tilde{O}(1/\gamma^2)(\log n)$
SmoothBoost	$\mathrm{poly}(n, 1/\gamma)$	$\tilde{O}(1/\gamma^2)(\log n)$
LP	1	$\mathrm{poly}(n, \log(1/\gamma))$
This paper	$\mathrm{poly}(n, 1/\gamma)$	$\tilde{O}(1/\gamma) + O(\log n)$

Algorithm

- Parallel boost-by-majority to handle ϵ-dependence [Fre95]
- Weak learner:
  - Random projection [JL84,AV99]
  - Interior point method [Ren88]
    - Invert Hessian using parallel matrix inversion [Rei95]
    - Round intermediate solutions (preserve margin)

Negative result



- Some boosters [KM95,MM00,KS02,LS05,LS08] use decision trees or branching programs.
- Calls to weak learners from the same "layer" parallelizable.
- Q: Can save iterations?
- A: No (so $\Omega(1/\gamma^2)$ iterations needed)
- Proof sketch:
  - variables conditionally independent given label
  - in stage *i*, give variable *i* to all weak learners.