Faster Jobs in Distributed Data Processing using MTL

Neeraja J. Yadwadkar, et al.

Problem

→ Stragglers:

1. extended job completion times
2. inefficient use of resources
3. increased costs

Two Approaches

Reactive

→ e.g. MapReduce

- wait-and-speculate if task is executing slower
- spawn multiple copies

Proactive

→ e.g. Wrangler

1. model builder
   a. data points: current resource usage counters
2. model-informed scheduler
   a. predict whether given task on given node would cause a straggler

Problems with Wrangler

1. need to build for each node

2. retrained for each workload

3. how to handle sparse data

AIM:

“Adapt MTL for learning a generalized predictive model with better prediction accuracy and improved job completion times.”

Multi-Task Learning

→ $w_0 + v_t = w_t$

$\min_{w0,vt,b} = \lambda_0 \| w_o \|^2 + \lambda_1 / T\ \Sigma_t \| v_t \|^2 + \Sigma_i \Sigma_t \xi_{it}$

subject to:

$y_{it}\ ((w_0 + v_t)^T x_{it} + b) \geq 1 - \xi_{it} : \forall_{i,t}$

$\xi_{it} \geq 0 : \forall_{i,t}$

MTL: Formulation

N nodes, **L** workloads, **NL** tasks:

→ general group: w_o

→ one group for each node: w_n

→ one group for each workload: w_l

→ one group for each task: v_t

$w = w_0 + w_n + w_l + v_t$

Multi-Task Learning

→ $w = w_0 + w_n + w_l + v_t$

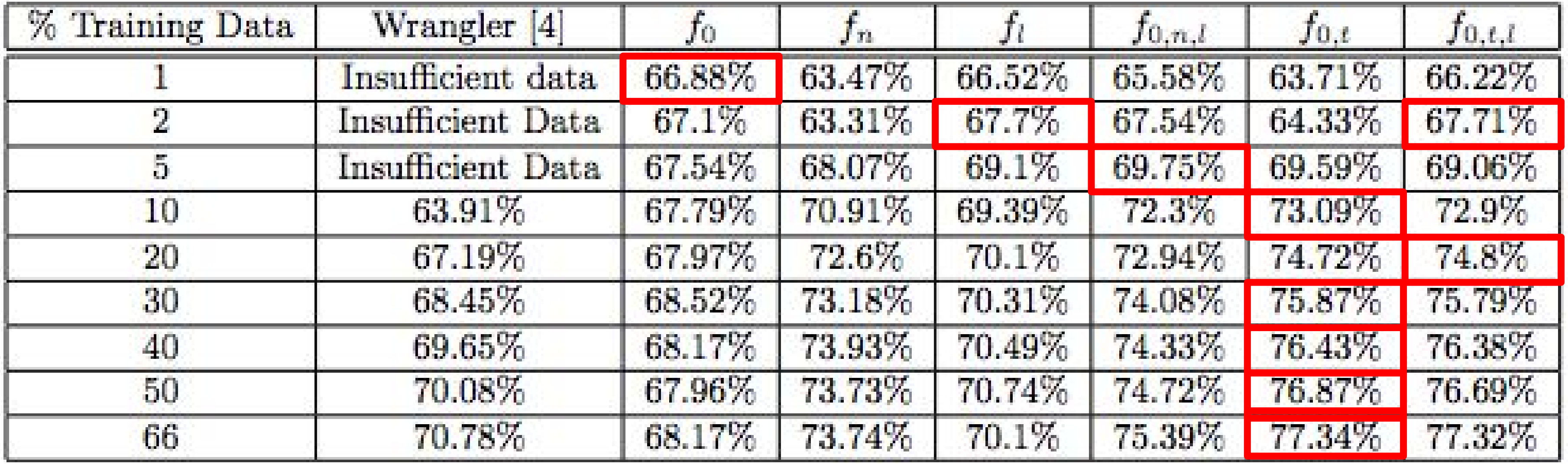
$\min_{w0,vt,b} = \lambda_0 \| w_o \|^2 + \nu / N \ \Sigma_t \| w_n \|^2 + \omega / L \ \Sigma_t \| w_l \|^2 + \tau / T \ \Sigma_t \| v_t \|^2 + \Sigma_i \Sigma_t \xi_{it}$

subject to:

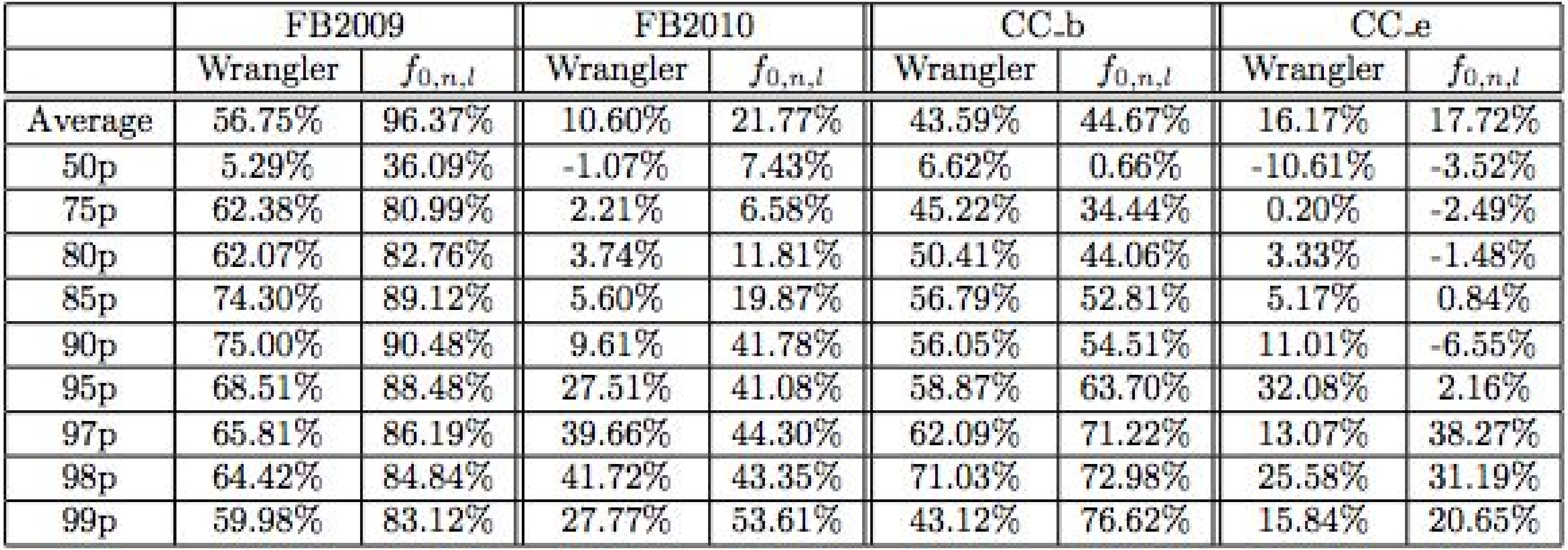
$y_{it} \ ((w_0 + w_n + w_l + v_t)^T x_{it} + b) \geq 1 - \xi_{it} : \forall_{i,t}$

$\xi_{it} \geq 0 : \forall_{i,t}$

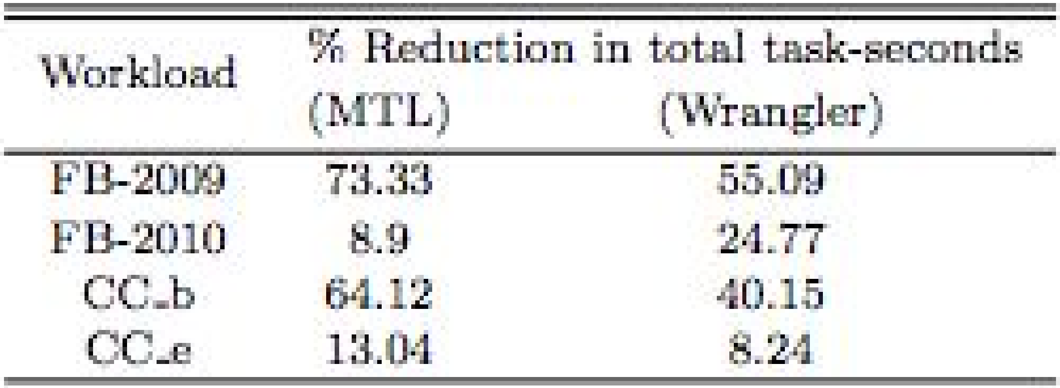
Evaluation & Experiments



% Training Data	Wrangler [4]	f_0	f_n	f_l	$f_{0,n,l}$	$f_{0,t}$	$f_{0,t,l}$
1	Insufficient data	66.88%	63.47%	66.52%	65.58%	63.71%	66.22%
2	Insufficient Data	67.1%	63.31%	67.7%	67.54%	64.33%	67.71%
5	Insufficient Data	67.54%	68.07%	69.1%	69.75%	69.59%	69.06%
10	63.91%	67.79%	70.91%	69.39%	72.3%	73.09%	72.9%
20	67.19%	67.97%	72.6%	70.1%	72.94%	74.72%	74.8%
30	68.45%	68.52%	73.18%	70.31%	74.08%	75.87%	75.79%
40	69.65%	68.17%	73.93%	70.49%	74.33%	76.43%	76.38%
50	70.08%	67.96%	73.73%	70.74%	74.72%	76.87%	76.69%
66	70.78%	68.17%	73.74%	70.1%	75.39%	77.34%	77.32%



	FB2009		FB2010		CC_b		CC_e	
	Wrangler	$f_{0,n,l}$	Wrangler	$f_{0,n,l}$	Wrangler	$f_{0,n,l}$	Wrangler	$f_{0,n,l}$
Average	56.75%	96.37%	10.60%	21.77%	43.59%	44.67%	16.17%	17.72%
50p	5.29%	36.09%	-1.07%	7.43%	6.62%	0.66%	-10.61%	-3.52%
75p	62.38%	80.99%	2.21%	6.58%	45.22%	34.44%	0.20%	-2.49%
80p	62.07%	82.76%	3.74%	11.81%	50.41%	44.06%	3.33%	-1.48%
85p	74.30%	89.12%	5.60%	19.87%	56.79%	52.81%	5.17%	0.84%
90p	75.00%	90.48%	9.61%	41.78%	56.05%	54.51%	11.01%	-6.55%
95p	68.51%	88.48%	27.51%	41.08%	58.87%	63.70%	32.08%	2.16%
97p	65.81%	86.19%	39.66%	44.30%	62.09%	71.22%	13.07%	38.27%
98p	64.42%	84.84%	41.72%	43.35%	71.03%	72.98%	25.58%	31.19%
99p	59.98%	83.12%	27.77%	53.61%	43.12%	76.62%	15.84%	20.65%



Workload	% Reduction in total task-seconds	
	(MTL)	(Wrangler)
FB-2009	73.33	55.09
FB-2010	8.9	24.77
CC_b	64.12	40.15
CC_e	13.04	8.24

Evaluation & Take Away

Pros:

→ Improve over speculative approach

→ Can handle stragglers

Cons:

→ Unclear evaluation