Online Scheduling via Learned Weights

Silvio Lattanzi, **Thomas Lavastida**,
Benjamin Moseley, Sergei Vassilvitskii

January 7, 2020

Carnegie Mellon University
Tepper School of Business

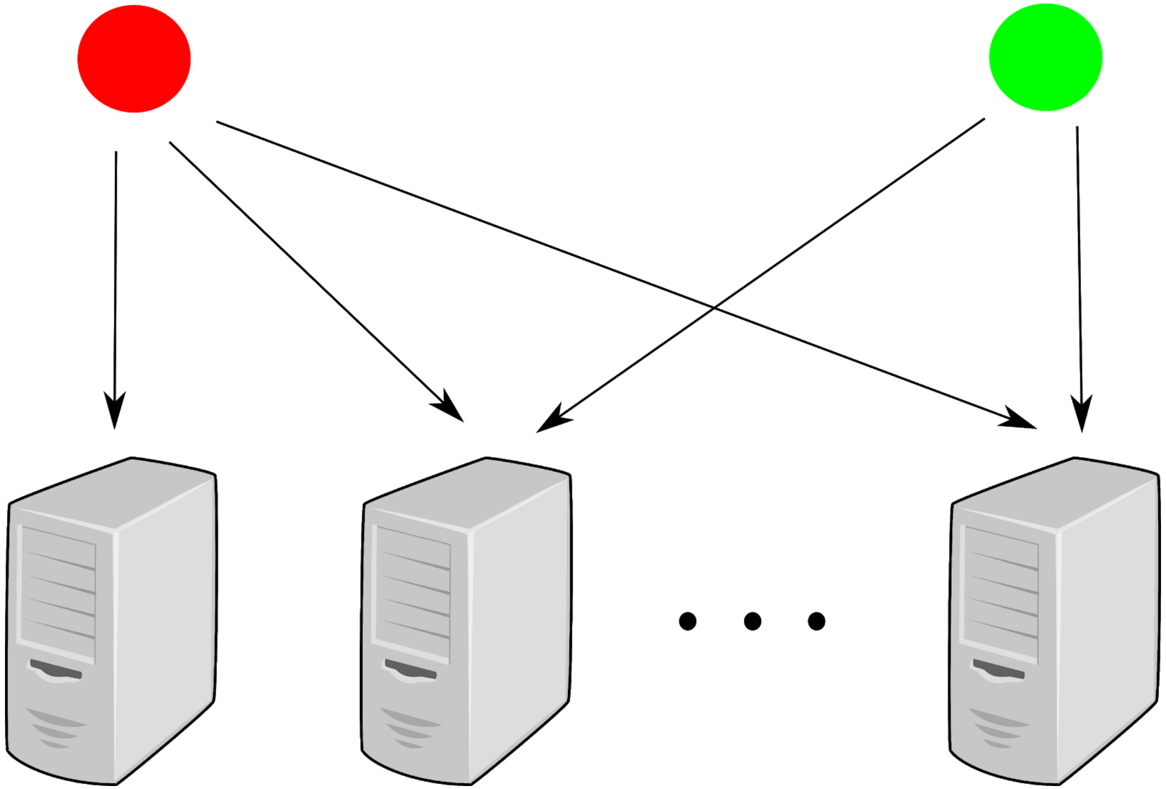
Google

Machine Learning + Algorithms

- Massive success in recent years
  - Image recognition, natural language processing, clustering, etc.
- Analysis and improvements of learning algorithms
- Can ML help to improve algorithms for classic problems?
  - Many interesting questions
  - Focus on online scheduling problem

Online Load Balancing

- m machines
- n jobs arrive in online list
  - Restricted assignments
  - $N(j) =$ subset of feasible machines for job j
  - $p_j =$ size of job j
- Machine load: total size of jobs assigned to a machine
- Goal: minimize makespan



Worst Case Analysis

- Online algorithm c-competitive if for all inputs

$$ALG \leq c \cdot OPT$$

- Every algorithm $\Omega(\log m)$-competitive
- Greedy algorithm $O(\log m)$-competitive
  - [Azar, Naor, Rom 1995]

Learning Augmented Algorithms

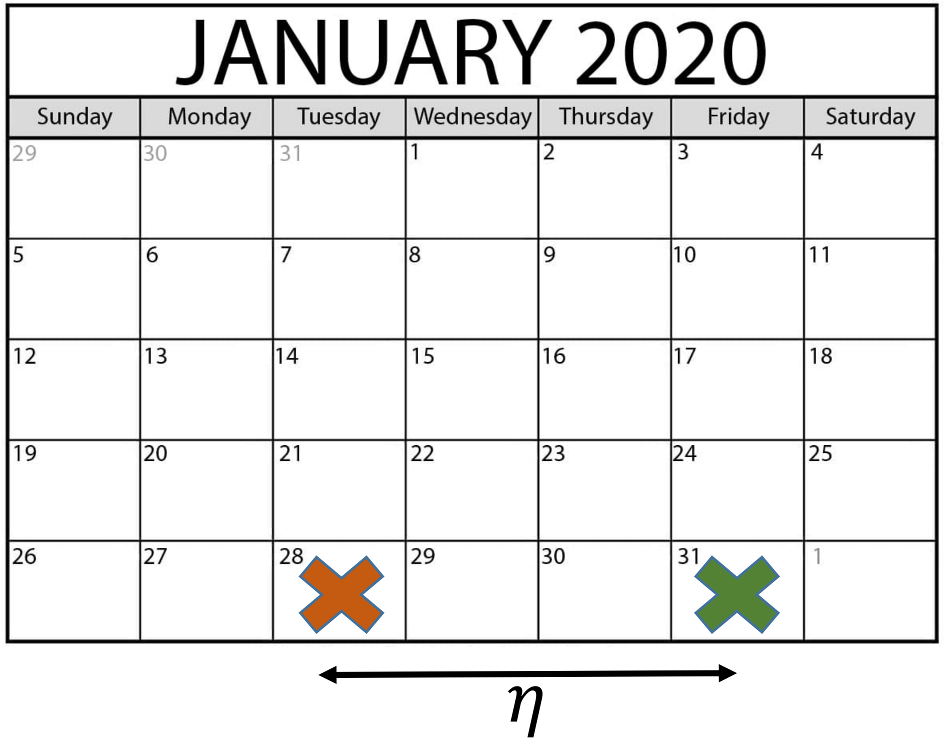
- Access to many traces of past jobs
- Learnable patterns may occur in practice
- Can ML be used to augment the design of online algorithms?
- Prediction about online instance
  - What to predict?
  - Handle errors?

Learning Augmented Algorithms

- Caching Problem [Lykouris and Vassilvitskii 2018]
- Ski Rental + Non-Preemptive Sched. [Purohit et al. 2018]
- Heavy Hitters Sketches [Hsu et al 2019]
- Improved Bloom Filters [Mitzenmacher 2018]
- Learned Index Structures [Kraska et al 2018]

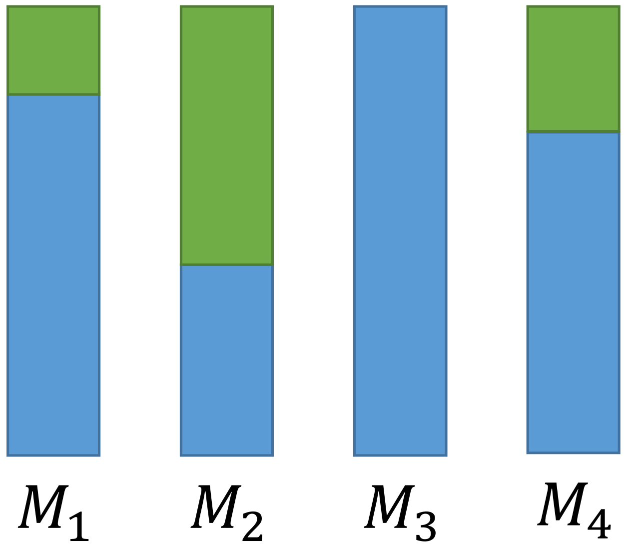
Online Algorithms + Predictions

- Ski Rental problem
  - Predict length of trip
- η := prediction error in hindsight
- Competitive ratio $= f(\eta)$
- Beat worst case for small η?
- Retain worst case for large η



Predictions for Load Balancing?

- Load of machines in OPT?
  - Pad the instance
- Dual variables?
  - Too sensitive to small errors
- Distribution over job subsets?
  - Potentially too many!
- Our approach:
  - Use predictions to get fractional solutions
  - Round online to get assignment



Results

Theorem 1 – Machine Weights

For any $\epsilon > 0$ and any instance there exists weights $w \in R_+^m$ and a fractional assignment $x(w)$ with fractional makespan at most $(1+\epsilon)OPT$

Given predictions w' there is an online algorithm yielding fractional assignments with fractional makespan at most $O(\log(\eta)\,OPT)$, $\eta := \max_i \frac{w_i'}{w_i}$ is relative error

Machine Weights

- Associate weight w_i to each machine
- Fractional Assignment:

$$x_{ij}(w) = \frac{w_i}{\sum_{i' \in N(j)} w_{i'}}$$

- Weights should satisfy

$$\sum_j p_j x_{ij}(w) \leq (1+\epsilon)OPT$$

- Builds off of [Agrawal et al. 2018]

Using Weights Online

- Say given predicted weights w'
- Operate in phases
- Assign using weights
- Update $w_i' \leftarrow \frac{w_i'}{2}$ if machine i gets load $10 \cdot OPT$
- Reset load and start a new phase
- If $\eta = \max_i \frac{w_i'}{w_i}$ then get $O(\log \eta)$-competitive assignment

Online Rounding Problem

- Receive j's size, neighborhood, fractional assignment online

$$\{x_{ij}\}_{i \in N(j)} \; s.t. \sum_{i \in N(j)} x_{ij} = 1$$

- Use x_{ij}'s to compute integral assignment online
- Rounding algorithm c-competitive if

$$ALG \leq c \cdot T$$

- $T \coloneqq \max\{\max_i \sum_j p_j x_{ij}, \max_j p_j\}$

Results

Theorem 2 – Online Rounding

There exists a $O((\log\log m)^3)$-competitive randomized online rounding algorithm for restricted assignment and succeeds with high probability.

Theorem 3 – Lower Bounds

Every deterministic online rounding algorithms is $\Omega(\frac{\log m}{\log\log m})$-competitive and every randomized online rounding algorithm is $\Omega(\frac{\log\log m}{\log\log\log m})$-competitve

Conclusions

- Theorems 1 and 2 imply $O((\log\log m)^3 \log \eta)$-competitive algorithm with predictions
- Moderately accurate predictions go beyond worst case
- Can retain $O(\log m)$-competitiveness when η large
- Connect prediction error to competitiveness

Questions?