Online Facility Location with Predictions

Shaofeng Jiang
Peking University

Joint work with Erzhi Liu, You Lyu, Zhihao Tang, Yubo Zhang

Algorithm Design with Predictions

Traditional algorithm design: focus on worst-case

- Strong guarantee, but often too pessimistic to be useful in practice

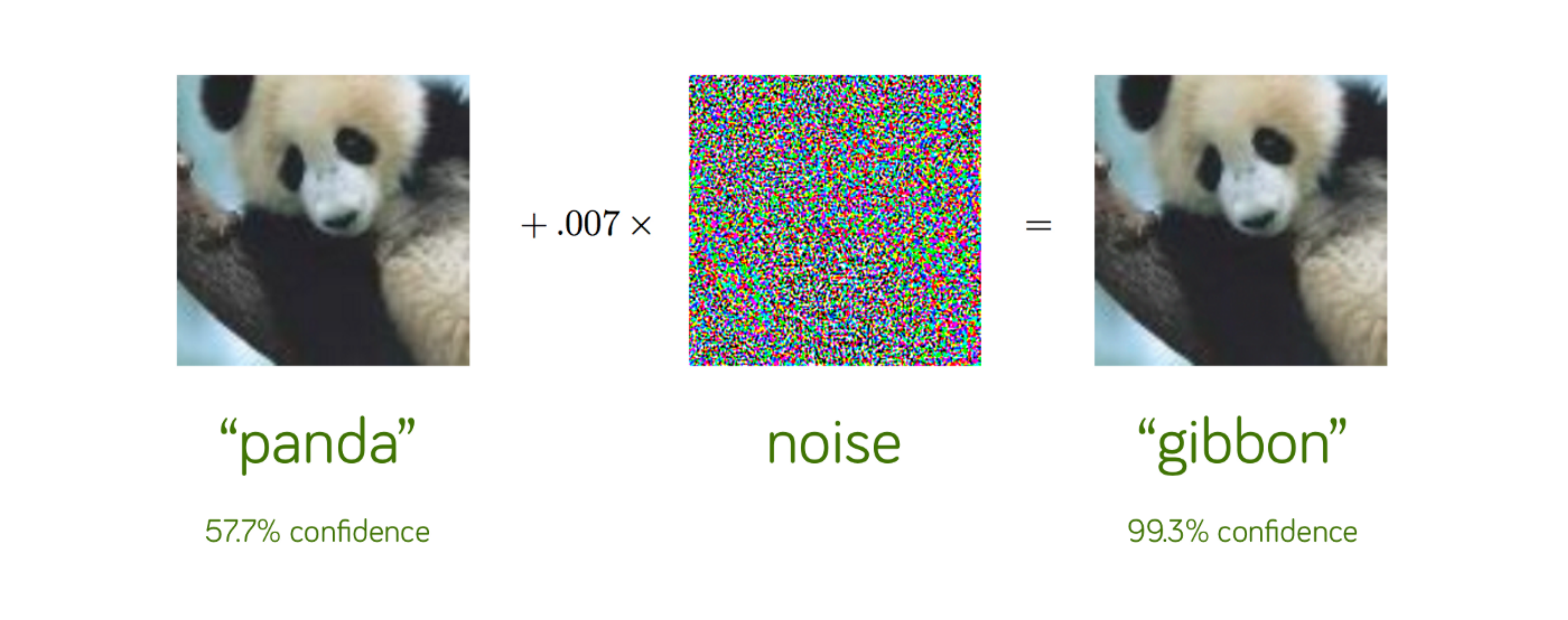
ML techniques:

- Data-driven, can leverage the structure of data, performs well in practice

Goal: design algorithms with a learned predictor to go beyond worst-case

Can We Trust The Predictor?

Adversarial example attack: small but structured noise



Unfortunately, perfectly-robust ML predictor is unlikely to exist

Utilizing Untrusted Predictions: LV Framework

(Lykouris-Vassilvitskii, JACM' 21)

Premises:

Algorithm does NOT know η in advance!

- Access to an untrusted predictor with error η (under certain measure)

Consistency:

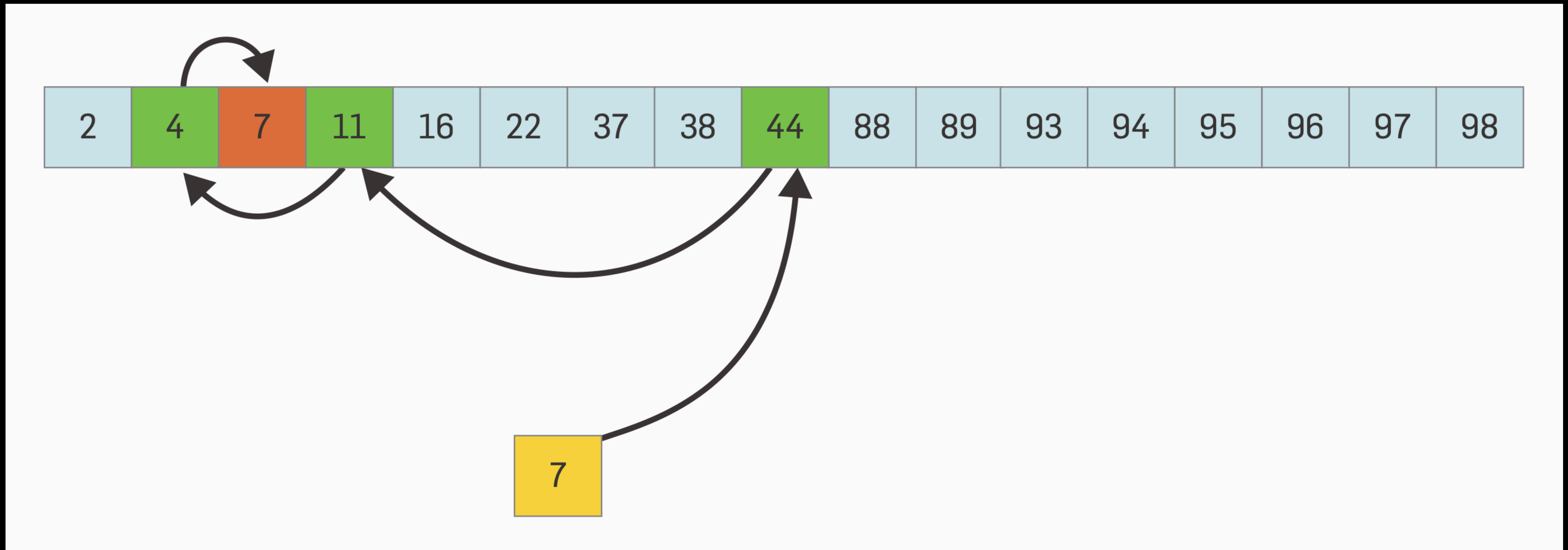
This requirement may bypass certain lower bounds

- if $\eta \to 0$ then algorithm (nearly) achieves optimal

Robustness:

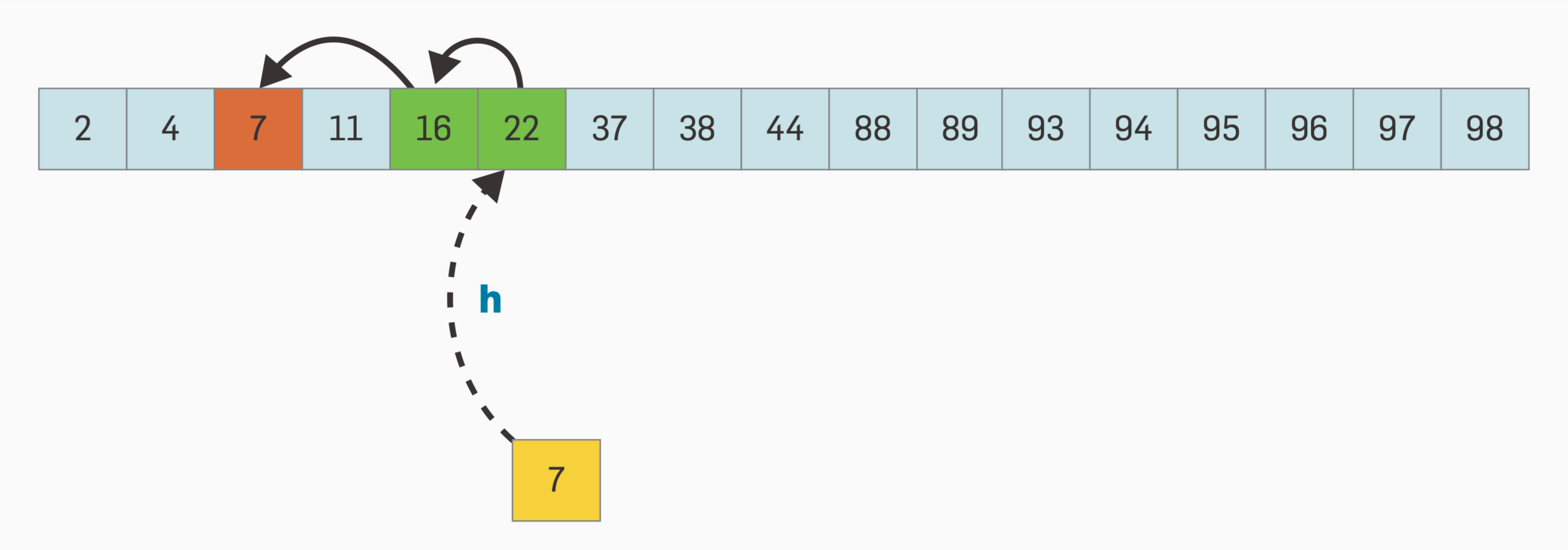
- if $\eta \to \infty$ then algorithm still has worst-case guarantee

Simple Example: Binary Search



Worst-case binary search

$O(\log n)$



Initial guess: h

Iterative-doubling from h

$O(\log |h - i^*|)$

Error $\eta = |h - i^*|$

Facility Location

Fundamental problem in OR and CS

Input: metric space (V, d), demand points $X = (x_1, \ldots, x_n) \subseteq V$

Classical setting:

- Find a set of open facilities $F \subseteq V$ (each with opening cost $w(f)$) s.t.

$$\sum_{f \in F} w(f) + \sum_{x_i \in X} d(x_i, f_i)$$

Opening cost

Connection cost

where f_i is the facility assigned to x_i

Online Setting

Input: metric space (V, d), demand points $X = (x_1, \ldots, x_n) \subseteq V$

Online: when x_i arrives, algorithm must irrevocably assign x_i to an open facility

- The next x_{i+1} is only revealed after x_i is assigned

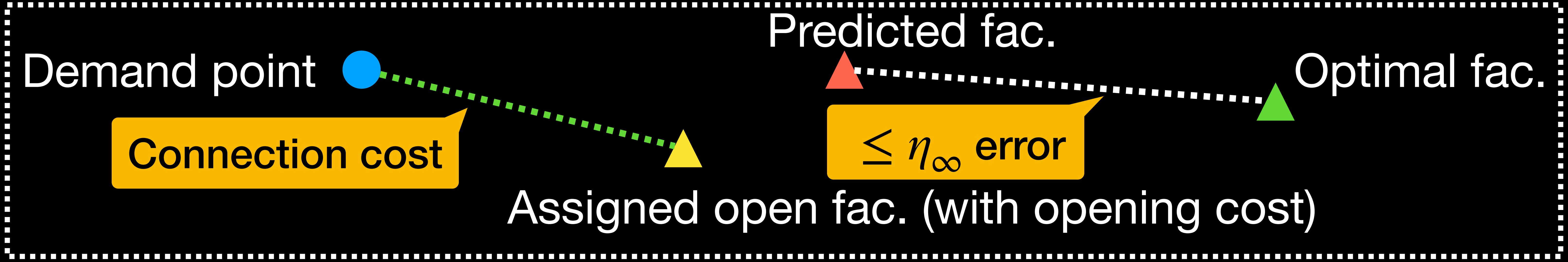
Competitive ratio: $\max_X \frac{\mathbb{E}[\mathrm{ALG}(X)]}{\mathrm{OPT}(X)} \geq 1$

Worst-case relative performance

The Prediction Model

Predictor: returns a (supposedly optimal) facility f_i^{pred} for each x_i

Error measure: $$\eta_\infty := \max_{1 \le i \le n} d(f_i^{\mathrm{pred}}, f_i^{\mathrm{opt}})$$



Results: Nearly-tight Bounds

$O(\log n)$ even when η_∞ is large;
Matches an UB by Meyerson (FOCS' 01)

Recall $\eta_\infty := \max_{1 \le i \le n} d(f_i^{\text{pred}}, f_i^{\text{opt}})$

Upper bound: There is an $O\left(\min\left\{\log n, \log(n\eta_\infty/\text{OPT})\right\}\right)$-competitive alg.

Note: O(1)-competitive when $\eta \to 0$

Related error measure: $\eta_1 := \sum_{1 \le i \le n} d(f_i^{\text{pred}}, f_i^{\text{opt}})$

Does it make sense to consider the ℓ_1 error measure?

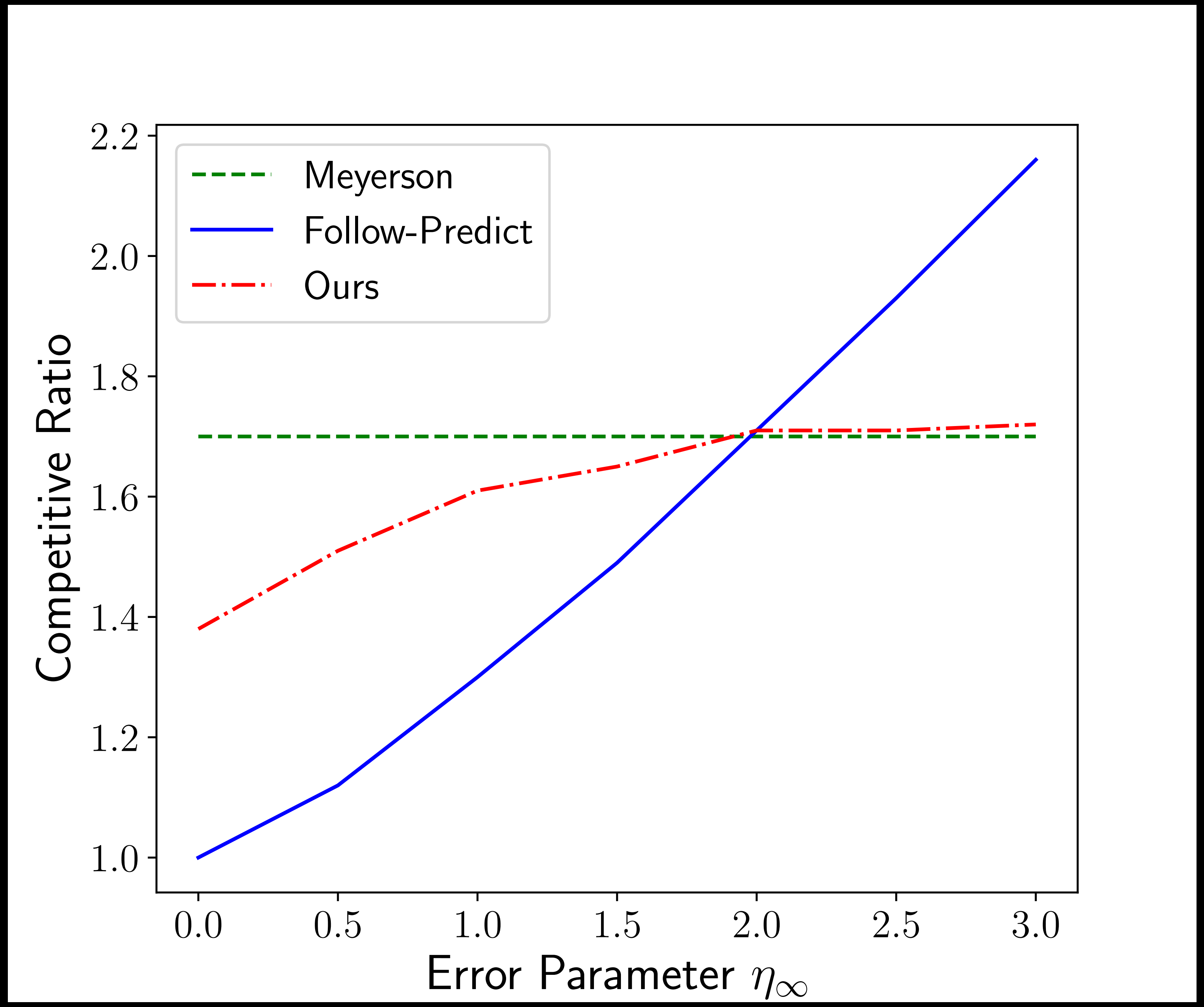
Lower bound: For every $\eta_\infty \in (0,1]$, any randomized online algorithm is $\tilde{\Omega}(\log(n\eta_\infty/\text{OPT}))$-competitive (with $\text{OPT} = O(1)$), even when $\eta_1 = O(1)$.

This generalizes an $\tilde{\Omega}(\log(n))$ worst-case lower bound by Fotakis (Algorithmica, 2008)

Results: Experiments

Baselines: Follow-Prediction; Meyerson is an $O(\log n)$-competitive worst-case algo.

Simulated predictor



Error (η_∞) vs ratio for Twitter dataset

Greedy predictor

- Use 30% dataset as the training set, and compute OPT from it
- When online demand arrive, generate prediction from current OPT
- Update OPT, as OPT on the dataset union the new request

dataset	Meyerson	Follow-Predict	Ours
Twitter	1.70	1.69	1.57
Adult	1.55	1.57	1.49
US-PG	1.47	1.47	1.43
Non-Uni	5.66	5.7	2.93

Performance when using the greedy predictor

Strategy of Algorithm Design

$\eta = 0$ and $\eta \to \infty$ are two extremes

- Corresponding algorithms: always-trust-predictor vs worst-case algorithm

Strategy: start with worst-case algorithm, then extend it to use the prediction

Worst-case algorithm: $O(\log n)$-competitive by Meyerson (FOCS' 01)

Meyerson's Algorithm

For simplicity, consider the uniform opening cost $w(f) = w, \forall f \in V$

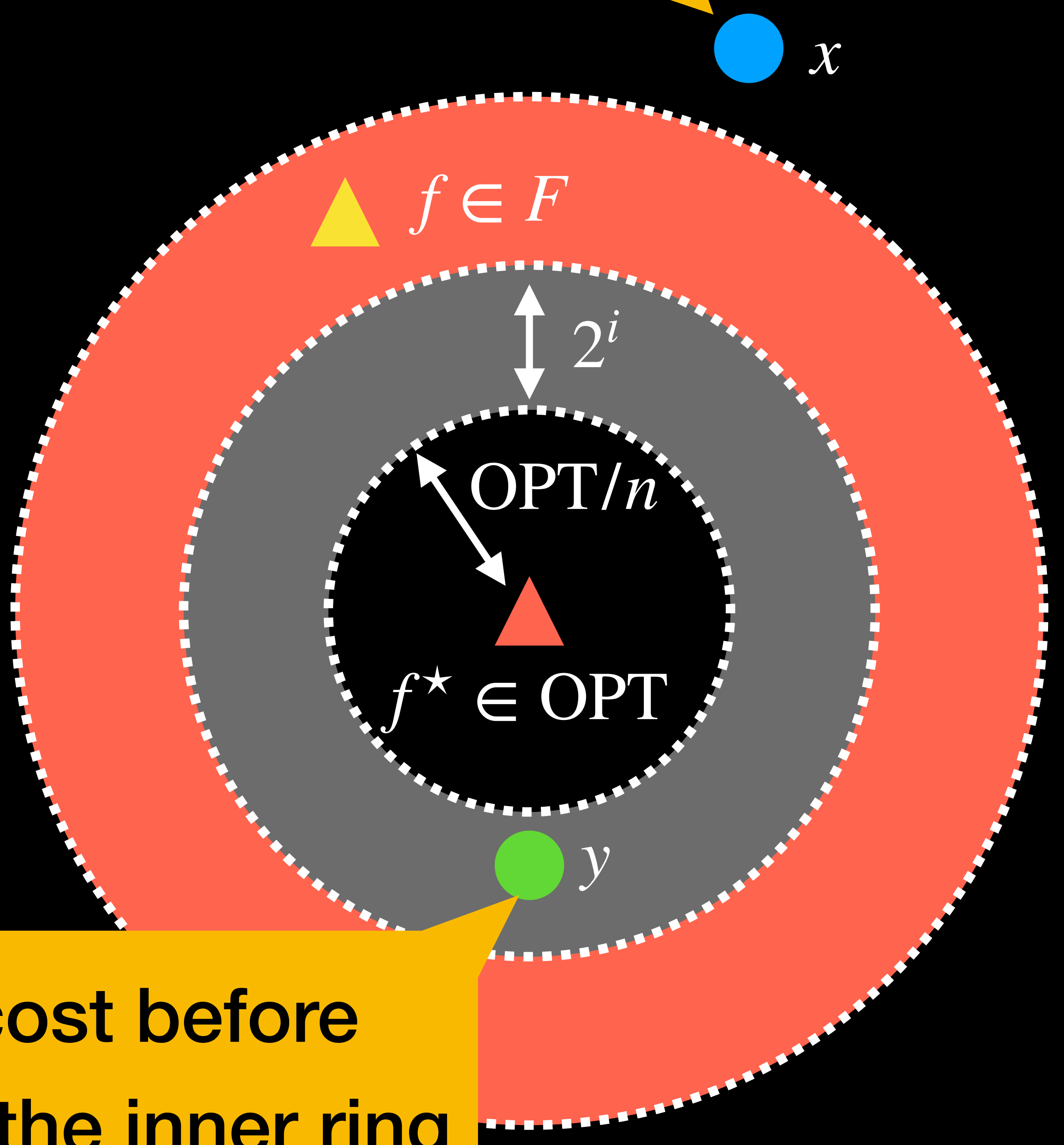
Initialize open facilities $F := \emptyset$

Demand points x on or outside the f ring: $d(x, f) = O(1) \cdot d(x, f^*)$, so O(1) to OPT

When x_i arrives:

Expected cost of x_i is $\leq \delta/w \cdot w + \delta \leq 2d(x_i, F)$

- Let $\delta := d(x_i, F)$ be the min-dist to the open facilities F
- With prob. δ/w, open a facility at x_i $(F := F \cup \{x_i\})$
- Assign x_i to the nearest facility in F



On average, $O(1) \cdot \mathrm{OPT}$ cost before opening facility at some y in the inner ring

Conclusion: ratio = O(# of rings)

Key Property

Suppose the initial open facilities F satisfies $d(F, \text{OPT}) \leq \eta$ then Meyerson's algorithm is $O(\log(n\eta/\text{OPT}))$-competitive, where

$$d(F, \text{OPT}) := \min_{f \in F, f' \in \text{OPT}} d(f, f')$$

In other words, every facility opened in OPT has $f \in F$ within dist η

(Follows from last slide: # of rings = $\log(\eta/(\text{OPT}/n)) = \log(n\eta/\text{OPT})$)

Simple Algorithm for Uniform Case

Algorithm: Run Meyerson's, and whenever Meyerson's decide to open a facility at some x_i, also open a facility at x_i's prediction f_i^{pred}

In the worst-case: only $O(1)$ more costly than Meyerson's, which implies $O(\log n)$ worst-case ratio

Why it works?

- Let c_i^* be the (offline) optimal facility that x_i is assigned to
- Prediction error guarantee: $d(f_i^{\text{pred}}, f_i^{\text{opt}}) \leq \eta_\infty$
- Hence, the very first facilities F we open satisfies $d(F, \text{OPT}) \leq \eta_\infty$

The cost is $O(1) \cdot \text{OPT}$ before this F is open

Implies the main bound: $O(\log(n\eta_\infty)/\text{OPT})$

Difficulties in Non-uniform Case

Non-uniform case: $w(f)$ can be arbitrary

- Meyerson's can handle the non-uniform case (with slight modifications)

"Whenever Meyerson's opens facility x_i, also open facility at f_i^{pred}"

- Doesn't work: $w(f_i^{\text{pred}})$ can be very large (and $w(f_i^{\text{opt}})$ can even be 0!)

Challenge: η_∞ measures connection cost, but say nothing on the opening cost

New Steps for Non-uniform Case

If one knows $w(f_i^{\text{opt}})$, then the nearest facility f' to f_i^{pred} with $w(f') \leq w(f_i^{\text{opt}})$ satisfies $d(f', f_i^{\text{opt}}) \leq \eta_\infty$

Hence, we need to "guess" $w(f_i^{\text{opt}})$

Don't be too aggressive — always bounded by Meyerson

- Set budget b, open f' closest to f_i^{pred} s.t. $w(f') \leq b$ w.p. $\text{cost}^{\text{Mey}}(x_i)/w(f')$
- Double budget b every time a new facility is opened

Use cost $O(1) \cdot \text{OPT}$ to open f' such that $w(f') \leq O(w(f_i^{\text{opt}}))$

Many Results, Many to Be Done

Online algorithms (competitive ratio)

- Caching, scheduling, online learning, online primal-dual

Data structures (space/time)

- ℓ_p-sampling, heavy hitters, bloom filter

Efficient algorithms for data analysis (running time)

- Clustering, nearest neighbor, low-rank approximation

… Many to be done

Thanks!