Approximation Algorithms for Combinatorial Optimization with Predictions

ICLR'25

Antonios Antoniadis, Marek Eliáš, Adam Polak, Moritz Venzin

Stanford CS/MS&E 331

Motivation: Algorithms with Predictions

For many problems (vertex cover, knapsack, …),
classic fast algorithms give constant-factor **approximations**:

- For minimization, (algorithm's output cost) $\leq \rho \cdot$ OPT, for $\rho \geq 1$
- For minimization, (algorithm's output value) $\geq \frac{1}{\rho} \cdot$ OPT, for $\rho \geq 1$

Improvements usually requires much slower algorithms

Key insight: Many applications have **rich historical data**

- Goal: Use this data to **predict** structure of **near-optimal** solutions
- But predictions may be infeasible, lead to costly mistakes, …

Outline

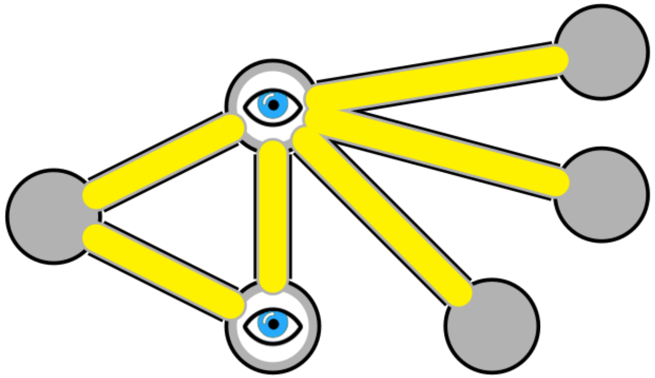
1. Motivation
2. **Background: Approximation algorithm example**
3. Paper setup
4. Main result: Algorithm with predictions

Background: Apx alg for vertex cover

- **Input:** graph $G = (V, E)$
- **Goal:** find $C \subseteq V$ s.t. every edge has at least one endpoint in C
  - Objective: minimize $|C|$

Algorithm:

1. Initialize cover $C \leftarrow \emptyset$, $\boxed{\text{matching } M \leftarrow \emptyset}$ ← Just for analysis
2. While there's an uncovered edge $(u, v) \in E$:
   i. Add both endpoints: $C \leftarrow C \cup \{u, v\}$
   ii. Add (u, v) to matching: $M \leftarrow M \cup \{(u, v)\}$
   iii. Delete all edges in E incident to u or v
3. Output C



Background: Apx alg for vertex cover

2. While there's an uncovered edge $(u, v) \in E$:
   i. Add both endpoints: $C \leftarrow C \cup \{u, v\}$
   ii. Add (u, v) to matching: $M \leftarrow M \cup \{(u, v)\}$
   iii. Delete all edges in E incident to u or v
3. Output C

Thm: 2-approximation algorithm $|C| \leq 2 \cdot \mathrm{OPT}$

- The edges in M are **disjoint** (no shared endpoints)
  - Any vertex cover must have **at least one endpoint** per edge in M
  - $\Rightarrow |M| \leq \mathrm{OPT}$
- Algorithm selects **both endpoints** of every edge in $|M|$:
  - $\Rightarrow |C| = 2|M| \leq 2 \cdot \mathrm{OPT}$

Outline

1. Motivation
2. Background: Approximation algorithm example
3. **Paper setup**
4. Main result: Algorithm with predictions

Setup: Selection problems

Universe of items: $[n] = \{1, \ldots, n\}$ each with weight $w(i) \geq 0$

Feasible solutions: subsets $X \subseteq [n]$, feasible set $\mathcal{X}$

Objective:

- *Minimization:* pick $X \in \mathcal{X}$ minimizing $w(X) \coloneqq \sum_{i \in X} w(i)$
- *Maximization:* pick $X \in \mathcal{X}$ maximizing $w(X) \coloneqq \sum_{i \in X} w(i)$

Many classical NP-hard problems fit this template:

- Set cover, TSP, Steiner tree, Knapsack, …

Predictions and error model

Prediction is simply a subset of items, $\hat{X} \subseteq [n]$

- Need not be feasible

To measure **prediction quality**, compare $\hat{X}$ to optimum X^*

- **False positives:** items predicted but not truly in opt $\eta^+ = w\big(\hat{X} \setminus X^*\big)$
- **False negatives:** $\eta^- = w\big(X^* \setminus \hat{X}\big)$

Predictions may come from data:

- E.g., ERM, probabilistic neural model, …

Outline

1. Motivation
2. Background: Approximation algorithm example
3. Paper setup
4. **Main result: Algorithm with predictions**

Main result: Minimization

- Suppose we have a ρ-approximation algorithm A
- Algorithm with prediction $\hat{X}$:
  1. Discount predicted items $\overline{w}(i) = \begin{cases} 0, & i \in \hat{X} \\ w(i), & \text{else} \end{cases}$
  2. Return $X = A(\overline{w})$
- Guarantee:

$$\frac{w(X)}{w(X^*)} \leq \min\left\{\rho, 1 + \frac{\eta^+ + (\rho - 1)\eta^-}{w(X^*)}\right\}$$

  - Perfect prediction ⇒ optimal solution
  - Bad prediction ⇒ still fall back to the ρ-approximation

Proof of $\frac{w(X)}{w(X^*)} \leq 1 + \frac{\eta^+ + (\rho - 1)\eta^-}{w(X^*)}$

By construction, $w\left(X \setminus \hat{X}\right) = \overline{w}(X)$

$$\begin{aligned} &\leq \rho \cdot \mathrm{OPT}_{\overline{\boldsymbol{w}}} \\ &\leq \rho \cdot \overline{w}(X^*) \\ &= \rho \cdot w\left(X^* \setminus \hat{X}\right) \end{aligned}$$

As a result, $w(X)$

$$\begin{aligned} &= w\left(X \cap \hat{X}\right) + w\left(X \setminus \hat{X}\right) \\ &\leq w\left(X \cap \hat{X}\right) + \rho \cdot w\left(X^* \setminus \hat{X}\right) \\ &= w\left(X \cap \hat{X}\right) + w\left(X^* \setminus \hat{X}\right) + (\rho - 1) \cdot w\left(X^* \setminus \hat{X}\right) \\ &\leq w\left(\hat{X}\right) + w\left(X^* \setminus \hat{X}\right) + (\rho - 1) \cdot w\left(X^* \setminus \hat{X}\right) \end{aligned}$$

Proof of $\frac{w(X)}{w(X^*)} \leq 1 + \frac{\eta^+ + (\rho-1)\eta^-}{w(X^*)}$

As a result, $w(X) \leq \underbrace{w(\hat{X}) + w(X^* \setminus \hat{X})}_{\boxed{= w(\hat{X} \cup X^*) = w(X^*) + w(\hat{X} \setminus X^*)}} + (\rho - 1) \cdot w(X^* \setminus \hat{X})$

Therefore, $w(X) \leq w(X^*) + \underbrace{w(\hat{X} \setminus X^*)}_{\eta^+} + (\rho - 1) \cdot \underbrace{w(X^* \setminus \hat{X})}_{\eta^-}$

Guarantee: $\frac{w(X)}{w(X^*)} \leq \min\left\{\rho, 1 + \frac{\eta^+ + (\rho-1)\eta^-}{w(X^*)}\right\}$

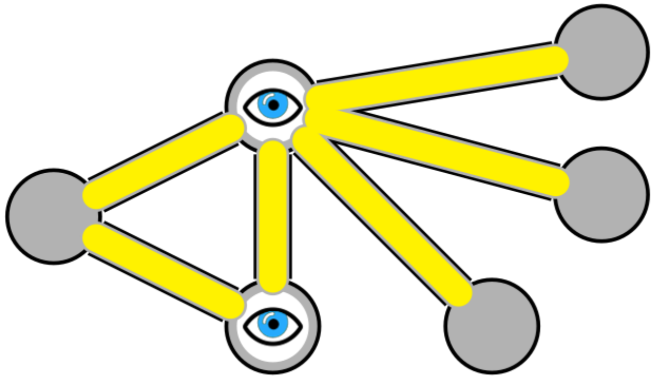
Run this algorithm and ρ-approximation algorithm $A(\mathbf{w})$ in parallel, output better solution

Implications for vertex cover

Hardness: Under Unique Games Conjecture,
no $(2-\epsilon)$-approximation is possible

Learning-augmented algorithm: approximation ratio

$$1+\frac{\eta^{+}+\eta^{-}}{\mathrm{OPT}}$$



Additional results

Additional **minimization** problems:

- Min-weight Steiner tree
- Min-weight perfect matching
  - Poly-time with $O(|V| \cdot |E|)$ runtime
  - Linear-time 2-approximation algorithm

Similar results for **maximization** problems

- Max-weight clique
- Max-weight independent set
- Knapsack

Overview

Key idea: Adapt fast classic algorithms

- Turn any ρ-approximation into a **prediction-aware** algorithm

Smooth improvement with prediction quality:

- Approximation ratio improves as $\eta^+, \eta^- \to 0$, yet never worse than ρ

Broad applicability:

- Vertex cover, Steiner tree, matching, independent set, knapsack, …