Prophet Secretary Against the Online Optimal

Paul Dütting[1], Evangelia Gergatsouli[2], Rojin Rezvan[3], Yifeng Teng[1], Alexandros Tsigonias-Dimitriadis[4]

[1]Google Research, [2]UW-Madison, [3]UT Austin, [4]Universidad de Chile

Background

→ The secretary problem and the prophet inequality are two of the most important optimal stopping problems with many applications.

→ We have to decide immediately and irrevocably whether to choose the current item.
Secretary problem: random arrival order, adversarial values for the items.
Prophet inequality: adversarial order, values drawn from known distributions.

→ We study the prophet secretary problem [1], a well-studied variant that arises as a natural combination of the two. Typically, we compare our algorithm to the **prophet**, who knows the values and always picks the best.
Closing the gap is still open! Current best strategy guarantees 0.669 [2] and UB is 0.7254 [3].

→ We turn to a natural, but much less-studied benchmark: the **online optimal algorithm** [4]. This benchmark has the same info as we do at every step, but infinite computational power.

→ This way, we measure the potential loss that arises due to computational limitations, rather than due to the fact that we have to make decisions online.

A new benchmark: The online optimal

Optimal strategy can be found via dynamic programming/backward induction → it is in fact a series of **decreasing thresholds**. But DP size is exponential…

Idea: Group variables together into g types without hopefully losing much in the DP value.

$$\text{OPT}(X, k_1, \ldots, k_g) = \mathbb{E}\left[\max\left(X, \sum_{i\in[g]} \frac{k_i}{k_1+\cdots+k_g}\text{OPT}(X_i, k_1, \ldots, k_{i-1}, k_i\text{-}1, k_{i+1}\ldots, k_g)\right)\right]$$

Running time is $O(n^g)$ → $O(1)$ types needed for PTAS.

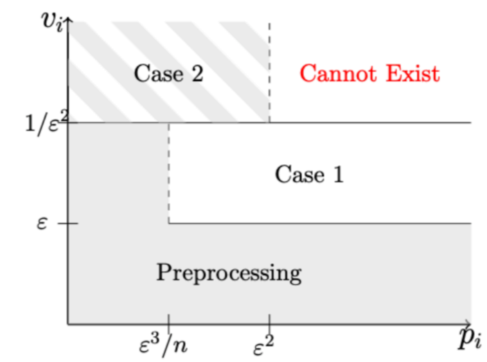
Obsv 1: If we can calculate thresholds that are always within $(1-\varepsilon)$ of the DP ones, then the strategy that uses these thresholds is a $(1-\varepsilon)$-approximation to the online optimal.

Obsv 2: Perturbing the support values and the probs of each X_i by a bit does not change the DP solution by much.

[1] Esfandiari H., HajiAghayi M., Liaghat V., Monemizadeh M. Prophet Secretary, ESA 2015.
[2] Correa J.R., Saona R., Ziliotto B. Prophet Secretary Through Blind Strategies, SODA 2019, Math. Program. 2022.
[3] Bubna A., Chiplunkar A. Prophet Inequality: Order selection beats random order, EC 2023.
[4] Papadimitriou C., Pollner T., Saberi A., Wajc D. Online Stochastic Max-Weight Bipartite Matching: Beyond Prophet Inequalities, EC 2021.

Warmup: QPTAS

→ To simplify, think of 2-point distr. with one mass at 0. WLOG, normalize OPT $\in [0.5, 1]$.



Preprocessing: Omit and lose $O(\varepsilon)$.

Case 1: Discretize values and probs to powers of $(1+\varepsilon)$.

Case 2: Need to compress values/probs to $poly\log(n)$ range.

Expectation-preserving transformation
For each X_i, set $v_i = v_{\max}$ and rescale p_i s.t. $\mathbb{E}[X_i'] = \mathbb{E}[X_i]$.
This loses only $O(\varepsilon^2)$. Next, discretize as in Case 1.

Counting the types: $O(1)$ values, $O(poly\log(n))$ probs → $O\left(n^{poly\log n}\right)$ running time.

→ For general distr. we need more because each X_i can have multiple big support values.

Fact: OPT (the optimal DP) will not set a threshold of more than OPT at any step.

$$\text{Bundling: } X_i' = \begin{cases} x & \text{when } x \le 1, \text{ w.p. } \Pr\left[X_i = x\right] \\ \mathbb{E}\left[X_i | X_i > 1\right] & \text{w.p. } \Pr\left[X_i > 1\right]. \end{cases}$$

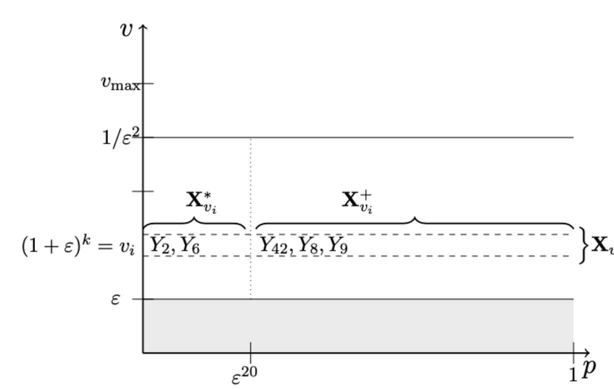
The bundling does not change the value of the optimal DP solution!

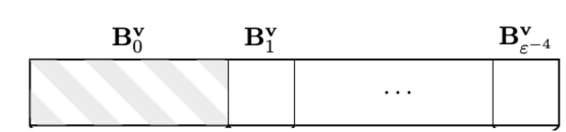
Main result: PTAS

→ Problem: Each support value can still have $\Omega(\log n)$ different realization probabilities.

→ Frontloading: For a sequence of r.v.'s, if their total realization prob. is "small", we can make a decision for the sequence as a whole: even if we have seen the realization of all such r.v.'s, we do not gain much compared to making a decision upon arrival of each!

Start from where we left off with the QPTAS and think again of 2-point distr.





→ "Move" the uncertainty to the beginning of each block (outside option).

→ DP needs to track # remaining (v, p)-r.v.'s and value of the outside option of current block of v.