Prophet Inequality

Zhengyu Jin

Department of Computer Science and Technology
Zhejiang University

2025 年 12 月 11 日

Overview

1. Basic Prophet Inequality

问题设定

盒子序列问题

有 n 个盒子 $1, 2, \ldots, n$。每个盒子 i 包含一个不可见的数字 $v_i \geq 0$，从独立的分布 F_i 中抽取。盒子按顺序 $1, 2, \ldots, n$ 到达。

决策规则

- 当盒子 i 到达时，我们观察到 v_i 并需要决定是否接受该盒子
- 如果接受盒子 i，游戏结束，我们的奖励是 v_i
- 如果让盒子 i 通过，我们无法再回去取回它

算法任务

目标

算法任务是设计一个算法，给定分布 $F_1, \cdots, F_n$，最大化期望奖励。

后向归纳算法

- 如果等到最后一个盒子，我们将获得的期望奖励是 $\mathbb{E}[v_n]$
- 给定这一点，当盒子 $n-1$ 到来时，我们应该只接受任何大于 $\mathbb{E}[v_n]$ 的值
- 这允许我们计算等待最后两个盒子时获得的期望值
- 以此类推，后向归纳使我们能够知道盒子到达的顺序

阈值算法

当每个盒子 i 到达时，算法持有一个阈值 θ_i，使得我们接受盒子 i 当且仅当 $v_i \geq \theta_i$。不难看出 θ_i 随着 i 的增加而减小。

Prophet Inequality 问题

与先知的比较

然而，先知不等式问题提出了一个更关注信息价值的问题。它关注的是将在线算法的性能与先验基准进行比较：

- 如果先知事先知道盒子中的所有值，那么先知的期望表现是 $\mathbb{E}[\max_i v_i]$
- 一个只知道分布的在线算法与这个基准相比表现如何?

主要定理

定理 (prophet inequality)

存在一个阈值 θ，使得接受第一个值至少为 θ 的盒子可以获得期望值至少为 $\frac{1}{2}\mathbb{E}[\max_i v_i]$。

对于任何 $\epsilon > 0$，不存在在线算法的性能保证至少为 $(0.5 + \epsilon)\mathbb{E}[\max_i v_i]$。

这个定理表明，即使在完全不确定的情况下，一个简单的阈值策略也能获得先知期望值的至少一半。这个 $\frac{1}{2}$ 的比率是紧的。

1.2 Quantile Approach

分位数方法

Quantile Approach

令随机变量 $v^* = \max_i v_i$。那么 v^* 的累积分布函数是 $F = \prod_i F_i$。令 θ 为 $F^{-1}(\frac{1}{2})$。

假设 $\Pr[\exists i, v_i = \theta] = 0$；这在所有分布都是原子分布时是成立的。

目标

我们证明接受第一个值至少为 θ 的盒子可以获得期望值至少为 $\frac{1}{2}\mathbb{E}[v^*]$。

先知值的上界

期望值分解

首先给出先知值的上界:

$$
\begin{aligned}
\mathbb{E}[v^*] &= \frac{1}{2}\mathbb{E}[v^* \mid v^* < \theta] + \frac{1}{2}\mathbb{E}[v^* \mid v^* \geq \theta] \\
&\leq \frac{1}{2}\theta + \frac{1}{2}\mathbb{E}[\theta + (v^* - \theta) \mid v^* \geq \theta] \\
&= \theta + \frac{1}{2}\mathbb{E}[(v^* - \theta) \mid v^* \geq \theta] \\
&= \theta + \mathbb{E}[(v^* - \theta)_+]
\end{aligned}
$$

其中 $(x)_+ = \max(x, 0)$。

阈值算法的分析

算法保证

以概率 $\frac{1}{2}$，阈值为 θ 的算法接受一个盒子，我们至少可以获得收益 $\frac{1}{2}\theta$。

在此之上，如果可能接受的盒子价值严格大于 θ，则它会贡献额外的期望收益：

$$
\begin{aligned}
&\sum_i \mathbb{E}[(v_i-\theta)_+]\cdot \Pr[\text{box } i \text{ is looked at}] \\
&\geq \sum_i \mathbb{E}[(v_i-\theta)_+]\cdot \Pr[\text{no box is taken in the end}] \\
&= \frac{1}{2}\sum_i \mathbb{E}[(v_i-\theta)_+] \geq \frac{1}{2}\mathbb{E}[(v^*-\theta)_+]
\end{aligned}
$$

因此，算法总共获得的期望值至少为 $\frac{1}{2}(\theta+\mathbb{E}[(v^*-\theta)_+])$。

重要说明

Remark

在分析 $(v_i - \theta)_+$ 部分时，我们规定了：$\Pr[\text{no box is taken in the end}] \geq \frac{1}{2}$；

而在分析 θ 部分时，我们规定了 $\Pr[\text{a box is taken in the end}] \geq \frac{1}{2}$ 。

因此原子性假设很重要。如果条件不成立，需要非常小心地打破平局。(如何做?)

1.3 An Economic Interpretation and Balanced Price Approach

经济学解释

拍卖视角

如果我们考虑向一系列竞标者出售单个物品，其中每个竞标者 i 的估值 v_i 独立地从分布 F_i 中抽取，那么阈值算法可以被视为最简单的销售策略：

- 发布价格 θ
- 将其出售给第一个愿意以该价格购买的买家 i，即 $v_i \geq \theta$

问题目标

问题要求制定一个销售策略，使得期望中购买该物品的买家具有较高的估值。这在经济学中被称为**社会福利最大化**（social welfare maximization）。

收益与效用

经济学视角下的解释

在这个视角下：

- $\theta \cdot \Pr[\text{某个买家购买}]$ 是卖家的**收益**（revenue）（或者说，卖家的效用）
- $\sum_i (v_i - \theta)_+ \cdot \Pr[i\ \text{被考虑}]$ 是买家的**效用**（utilities）之和

关键观察

因此，我们上面执行的计算分别约束了收益和卖家的效用，而交易的福利就是卖家收益和买家效用之和！

Balanced Price Approach

收益与效用

考虑 $\theta = \frac{1}{2}\mathbb{E}[\max_i v_i]$。在这个发布价格 θ 下，令 p 表示任何买家购买的概率。

那么卖家的收益是 θp。买家的效用是:

$$\sum_i \mathbb{E}[(v_i - \theta)_+] \cdot \Pr[\text{buyer } i \text{ has an opportunity to purchase}]$$

$$\geq \sum_i \mathbb{E}[(v_i - \theta)_+](1-p) \geq (1-p)\mathbb{E}\left[\max_i v_i - \theta\right] = (1-p)\theta.$$

结论

因此福利至少为 θ。

1.4 The Lower Bound

下界构造

紧性证明

考虑两个盒子:

- v_1 确定性地为 1
- v_2 以概率 $1/h$ 为 h，以概率 $1-1/h$ 为 0，其中 h 是任意大的数

性能比较

- 先知在这个实例上的表现是 $2-1/h$
- 任何在线算法，无论是接受还是不接受第一个盒子，都不能获得超过 1 的值

结论

因此 $\frac{1}{2}$ 是紧的。

2. Matroid Prophet Inequalities

问题设定

Matroid 约束下的盒子选择

如前所述，令 $[n]$ 为一组盒子，每个盒子 i 包含独立从分布 F_i 中抽取的值 v_i。令 $\mathcal{M}$ 为 $[n]$ 上的一个 matroid，$\mathcal{I}$ 为独立集的集合。

规则

- 盒子按对抗性顺序到达，该顺序在算法之前是已知的
- 不失一般性，假设盒子按顺序 $1, 2, \ldots, n$ 到达
- 算法在看到盒子 i 的值 v_i 时必须决定是否接受该盒子，并且不能稍后撤回已接受的盒子
- 在任何时候，算法接受的盒子集合必须是 $\mathcal{M}$ 的独立集

算法目标

优化目标

算法的目标是最大化接受盒子中的总价值。

基准定义

自然的先知基准是 $\mathbb{E}[\max_{S\in\mathcal{I}}\sum_{i\in S} v_i]$。接下来我们定义 ex ante 基准：令 $\mathcal{P}_{\mathcal{M}}$ 为与 matroid M 相关联的多面体，*ex ante* 最优定义为：

$$\max_{x\in\mathcal{P}_{\mathcal{M}}}\sum_i \mathbb{E}\left[v_i x_i \mid v_i \text{ is in the top } x_i \text{ quantile of } F_i\right]. \tag{1}$$

注意

容易看出 ex ante 最优不少于先知基准。Ex ante 基准通常严格大于先知，尽管差距由 $\frac{e}{e-1}$ 的因子界定，这被称为相关性差距（correlation gap）（Agrawal et al., 2012)。

阈值算法

阈值算法

阈值算法在每个盒子 i 到达时计算一个阈值 θ_i，使得盒子 i 被接受当且仅当 $v_i \geq \theta_i$。需要注意两点：

- 阈值 θ_i 的计算仅使用盒子 i 到达之前的信息，包括观察到的值 $v_1, \ldots, v_{i-1}$ 和盒子 $i-1$ 之后接受的盒子集合，我们记为 A_{i-1}；
- 如果 $A_{i-1} \cup \{i\} \notin \mathcal{I}$，那么 θ_i 应该设为 ∞。

定理

存在一个阈值算法，其期望收集的总价值至少为 *ex ante* 最优的一半。

Ex ante 的具体表示

定义

我们将在 $\boldsymbol{v}' = (v'_1, \ldots, v'_n)$ 上定义一个相关分布。这些随机变量独立于 $\boldsymbol{v}$ 但具有相同的边际分布。定义：

$$\boldsymbol{x}^* := \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{P}_{\mathcal{M}}} \sum_i \mathbb{E}\left[v_i x_i \mid v_i \text{ is in the top } x_i^* \text{ quantile of } F_i\right].$$

由于 $\boldsymbol{x}^* \in \mathcal{P}_{\mathcal{M}}$，$\boldsymbol{x}^*$ 可以表示为凸组合 $\boldsymbol{x}^* = \sum_{S \in \mathcal{I}} \alpha_S 1_S$，其中 1_S 是集合 S 的指示变量，$\sum_S \alpha_S = 1$，$\alpha_S \geq 0, \forall S$。$\boldsymbol{v}'$ 由 $\boldsymbol{x}^*$ 定义：首先以概率 α_S 抽取 $\boldsymbol{x} \in \{0,1\}^n$，使得 $\boldsymbol{x} = 1_S$；然后对每个 $i \in [n]$，如果 $x_i = 1$，v'_i 从 F_i 的顶部 x_i^* 分位数中抽取；否则 v'_i 从 F_i 的底部 $1 - x_i^*$ 分位数中抽取。

v'_i 有 x_i^* 的概率来自 F_i 的顶部 x_i^* 分位数，且 $1 - x_i^*$ 的概率来自底部 $1 - x_i^*$ 分位数。因此这样的构造下，v'_i 的边际分布与 v_i 相同，不同的是 v'_i 之间存在相关性。

Ex Ante 最优的上界

命题 1

Ex ante 最优至多为 $\mathbb{E}[\max_{S\in\mathcal{I}}\sum_{i\in S} v_i']$。

证明.

$\{\alpha_S\}_S$ 定义在 x^* 的实现下，独立集上的分布。F_S 表示给定 S 时 v' 的条件分布。那么：

$$\begin{aligned}\mathbb{E}_{v'}\left[\max_{S\in\mathcal{I}}\sum_{i\in S} v_i'\right] &\geq \mathbb{E}_{S\sim\alpha}\left[\mathbb{E}_{v'\sim F_S}\left[\sum_{i\in S} v_i'\right]\right] \\ &= \mathbb{E}_{S\sim\alpha}\left[\sum_{i\in S}\mathbb{E}[v_i \mid v_i \text{ is in the top } x_i^* \text{ quantile of } F_i]\right]\end{aligned}$$

事实上这个上界也是可以取等的，因为 $\mathbb{E}[\max_{S\in\mathcal{I}}\sum_{i\in S} v_i']$ 所枚举的不同 v'，令 $\max_{S\in\mathcal{I}}\sum_{i\in S} v_i'$ 达到最大值的 S 恰好是 v_i' is in the top x_i^* quantile of F_i 的集合。 □

记号说明

Notations

令 A_i 表示阈值算法在第 i 个盒子之后接受的盒子集合，$A = A_n$ 为最终选择；对于任何 v_i' 的实现，令 $B = \text{argmax}_{S \in \mathcal{I}} \sum_{n \in B} v_i'$.

定义 B 的一个划分 $B = R \cup C, R \cap C = \emptyset$，其中 $A \cup R \in \mathcal{I}, |C| = |A|$；令 $R(A)$ 和 $C(A)$ 为所有合法划分中 $\sum_{i \in R(A)} v_i'$ 最大的那个划分。注意 A 仅依赖于 v，而 $C(A)$ 和 $R(A)$ 依赖于 v 和 v'。

α-平衡定义

定义

如果对于任何实现的 v，对于任何 $V \subseteq [n]$ 使得 $V \cup A \in \mathcal{I}$，算法使用的阈值满足：

$$\sum_{i \in A} \theta_i \geq \frac{1}{\alpha} \mathbb{E}\left[\sum_{i \in C(A)} v'_i\right], \tag{2}$$

$$\sum_{i \in V} \theta_i \leq \left(1 - \frac{1}{\alpha}\right) \mathbb{E}\left[\sum_{i \in R(A)} v'_i\right]. \tag{3}$$

那么阈值算法被称为是 α-平衡的。

α-平衡算法的性能

引理 (Lemma 3)

α-平衡阈值算法获得的期望值至少为 *ex ante* 最优的 $\frac{1}{\alpha}$ 倍。

证明

根据 v' 的定义，$R(A)$ 和 $C(A)$，ex ante 最优为 $\mathbb{E}[\sum_{i\in C(A)} v'_i + \sum_{i\in R(A)} v'_i]$。
由 (2)，

$$\mathbb{E}_v\left[\sum_{i\in A}\theta_i\right] \geq \frac{1}{\alpha}\mathbb{E}_v\left[\mathbb{E}_{v'}\left[\sum_{i\in C(A)} v'_i\right]\right]. \tag{4}$$

证明（续）

福利分解

通过经济学解释（见 Section 1.2），这部分价值可以看作收益，剩余部分是效用，我们将其下界如下：

$$\mathbb{E}_{\mathbf{v}}\left[\sum_{i\in A}(v_i-\theta_i)_+\right]=\mathbb{E}_{\mathbf{v}}\left[\sum_{i\in[n]}(v_i-\theta_i)_+\right]=\mathbb{E}_{\mathbf{v},\mathbf{v}'}\left[\sum_{i\in[n]}(v'_i-\theta_i)_+\right]$$
$$\geq\mathbb{E}_{\mathbf{v},\mathbf{v}'}\left[\sum_{i\in R(A)}(v'_i-\theta_i)_+\right].$$

第一个等式来自阈值算法的定义。第二个等式来自以下事实：

(a) θ_i 仅依赖于 $v_1,\ldots,v_{i-1}$ 但不依赖于 v_i，也不依赖于 v'；

(b) v 独立于 v'，因此特别地 v'_i 独立于 $v_1,\ldots,v_{i-1},\theta_i$；

(c) v'_i 与 v_i 有相同的边际分布。

证明（续）

应用不等式 (3)

我们现在可以应用 (3) 并得到：

$$\mathbb{E}_{v'}\left[\sum_{i\in R(A)}(v'_i-\theta_i)_+\right]\geq\mathbb{E}_{v'}\left[\sum_{i\in R(A)}v'_i\right]-\mathbb{E}_{v'}\left[\sum_{i\in R(A)}\theta_i\right]$$
$$\geq\mathbb{E}_{v'}\left[\sum_{i\in R(A)}v'_i-\left(1-\frac{1}{\alpha}\right)\mathbb{E}_{v'}\left[\sum_{i\in R(A)}v'_i\right]\right]=\frac{1}{\alpha}\mathbb{E}_{v'}\left[\sum_{i\in R(A)}v'_i\right].\qquad(5)$$

引理通过对 (4) 和 (5) 求和得出。 □

2-平衡阈值算法

下一步我们将通过引理 3 构造一个 2-平衡阈值算法。

引理 (Lemma 4)

具有以下阈值的阈值算法是 2-平衡的：对于每个盒子 i，如果 $A_{i-1} \cup \{i\} \notin \mathcal{I}$，令 θ 为 ∞；否则 $\theta_i := \frac{1}{2}(f(A_{i-1}) - f(A_{i-1} \cup \{i\}))$。

其中定义 $f : \mathcal{I} \to \mathbb{R}$ 为 $f(S) := \mathbb{E}[\sum_{i \in R(S)} v_i]$。

子模性的定义

定义 (子模函数)

对于一个集合函数 $f\colon 2^{[n]} \to \mathbb{R}$（或在本文中定义在独立集族 $\mathcal{I}$ 上），如果对任意集合 $S, T \subseteq [n]$（且属于定义域），有：

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T),$$

则称 f 是子模的。

等价定义（边际递减性质）

对于任意 $X \subseteq Y \subseteq [n]$ 和元素 $e \notin Y$，有：

$$f(X \cup \{e\}) - f(X) \geq f(Y \cup \{e\}) - f(Y).$$

即：增加一个元素带来的边际增益，随着集合的扩大而减小。

2-平衡阈值算法

证明

我们检验 2-平衡性的两个性质。对于任何 v，将 A 中的元素编号为 $a_1, \ldots, a_k$，令 a_0 为 0，那么：

$$
\begin{aligned}
\sum_{i \in A} \theta_i &= \frac{1}{2} \sum_{j=1}^{k} f(A_{a_{j-1}}) - f(A_{a_j}) \\
&= \frac{1}{2} \left[f(A_0) - f(A_{a_k}) \right] = \frac{1}{2} \left[f(\emptyset) - f(A) \right] \\
&= \frac{1}{2} \mathbb{E} \left[\sum_{i \in C(A)} v_i \right].
\end{aligned}
$$

2-平衡阈值算法

证明（续）

容易证明 $f(\cdot)$ 是子模的。对于任何 $V \subseteq [n]$ 使得 $A \cup V \in \mathcal{I}$，令 V 中的元素为 $a_1, \ldots, a_k$，令 W_j 为 $A \cup \{a_1, \ldots, a_j\}$，对 $j = 1, \cdots, k$，且 $W_0 = A$，那么：

$$\begin{aligned}\sum_{i \in V} \theta_i &\leq \frac{1}{2} \sum_{i \in V} (f(A) - f(A \cup \{i\})) \\ &\leq \frac{1}{2} \sum_{j=1}^{k} (f(W_{j-1}) - f(W_j)) = \frac{1}{2} f(A).\end{aligned}$$

□

References

Fu, H. (2021). Basic Prophet Inequality, Pandora Box Problem and Online Contention Resolution Schemes. Lecture Notes. Retrieved from https://fuhuthu.com/notes/prophet.pdf

Agrawal, S., Ding, Y., Saberi, A., & Ye, Y. (2012). Price of correlations in stochastic optimization. Operations Research, 60(1), 150-162.

Feldman, M., Svensson, O., & Zenklusen, R. (2016). Online contention resolution schemes. In Krauthgamer, R., editor, Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016, pages 1014-1033. SIAM.

Kleinberg, R., & Weinberg, S. M. (2019). Matroid prophet inequalities and applications to multi-dimensional mechanism design. Games and Economic Behavior, 113, 97-115.

The End