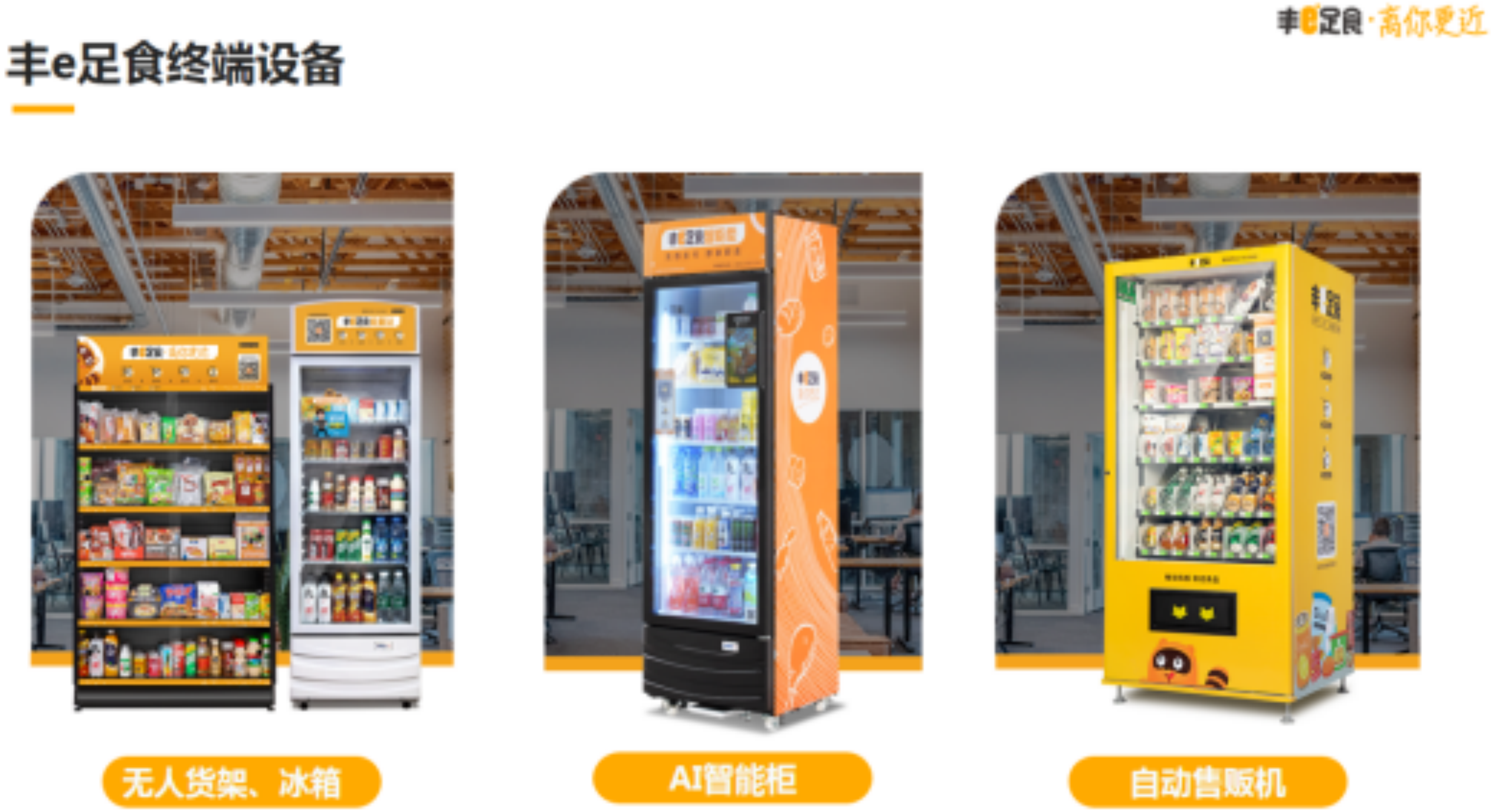
Distributionally Robust Newsvendor Problem

AIE1901 - AI Exploration - LLM for Optimization

Jie Wang, 2025/11/18

Smart Vending Shelves Operations: Fengyi



- Assortment
- Inventory Management
  - Forecasting project
- Pricing
- Logistics/routing

- Hibernated by SF Express in 2017
- Series A round ¥300M in 2021
- 30 cities in China + 100,000 shelves
- Adding thousands of shelves per month

A Day in the Life of a Newsvendor

- **The Scenario**:
  - You run a newspaper stand
  - Each evening, you must decide how many newspapers y to order for the next day
  - You do not know the exact customer demand D for tomorrow
- **The Dilemma**:
  - Order too many? You have leftovers and lose money
  - Order too few? You miss sales and disappoint customers
- **Question**: How do we find the "sweet spot"?

Key Parameters

- **Selling Price** ($p = 20$): The revenue for each newspaper you sell
- **Buying Cost** ($c_v = 4$): The cost for each newspaper you order
- **Holding Cost** ($h = 1$): The loss for each *leftover* newspaper (waste, storage)
- **Backorder Cost** ($b = 25$): The penalty for each unsatisfied customer (lost goodwill, lost profit)

The Profit Function

How much money do you make for a given demand D and order quantity y?

$$\text{Profit} = \text{Revenue} - \text{TotalCost}$$

- Revenue: $p \cdot \min(D, y)$ (You can only sell as many as you have or as customers want)
- Total Cost:
  - **Ordering Cost**: $c_v \cdot y$
  - **Holding Cost (Cost for leftovers):** $h \cdot \max(y - D, 0)$
  - **Backorder Cost (Penalty for stockout):** $b \cdot \max(D - y, 0)$

The Profit Function

How much money do you make for a given demand D and order quantity y?

$$\text{Profit} = \text{Revenue} - \text{TotalCost}$$

$$g(D, y) = p \cdot \min(D, y) - c_v \cdot y - h \cdot \max(y - D, 0) - b \cdot \max(D - y, 0)$$

Solving Newsvendor Problem with Data

- We have a problem: We don't know future demand D
- **The idea**: Use past data to predict the future!
- Suppose we have historical demand for the last n days:

$$\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n$$

- **Sample Average Approximation (SAA):**
  - Assume the future will behave exactly like the past
  - Approximate the real, unknown customer demand with our data samples

$$\mathbf{Pr}(D = \hat{d}_1) = \frac{1}{n}, \quad \mathbf{Pr}(D = \hat{d}_2) = \frac{1}{n}, \quad \ldots, \mathbf{Pr}(D = \hat{d}_n) = \frac{1}{n}$$

The SAA Optimization Problem

- **Goal**: Find the order quantity y that gives the highest average profit.

$$\max_{y} \ \frac{1}{n} \sum_{i=1}^{n} g(\hat{d}_i, y)$$

$$\text{s.t. } y \in \{0,1,\ldots,100\}$$

- We calculate the profit for each historical demand $\hat{d}_i$
- We average these profits
- We try each possible y among $\{0,1,\ldots,100\}$ and pick the one with the highest average profit

A Moment of Doubt

- What if our past data were just lucky (or unlucky)?
- What if the true underlying demand pattern is different?
- Our SAA model is **overconfident** in the historical data.

We need a strategy that is more … robust!

Distributionally Robust Optimization

- **Idea**: Instead of trusting one specific distribution (like SAA), we consider **all possible distributions** that are "reasonably similar" to our data.
- What is “reasonably similar”?
  1. The mean demand (μ)
  2. The variance of demand (σ^2)
- We look at all demand distributions that have the same mean and variance

A "Max-Min" Problem

- We want to choose y that performs well even in the worst-case scenario:

$$\max_{y \in \{0,1,\ldots,100\}} \left\{ \min_{p_0,\ldots,p_{100}} \sum_{d=0}^{100} p_d \cdot g(d,y) \right\}$$

Subject to the probabilities $p_0, p_1, \ldots, p_{100}$ being plausible:

- Probabilities non-negative, sum to 1
- The mean matches μ
- The variance matches σ^2

A "Max-Min" Problem

- We want to choose y that performs well even in the worst-case scenario:

$$\max_{y \in \{0,1,\ldots,100\}} \left\{ \begin{array}{ll} \min\limits_{p_0,\ldots,p_{100}} & \sum_{d=0}^{100} p_d \cdot g(d,y) \\ \text{s.t.} & p_0, \ldots, p_{100} \geq 0, \sum_{d=0}^{100} p_d = 1, \\ & \sum_{d=0}^{100} d p_d = \mu, \\ & \sum_{d=0}^{100} (d-\mu)^2 p_d = \sigma^2 \end{array} \right\}$$

A "Max-Min" Problem

- We want to choose y that performs well even in the worst-case scenario:

$$\max_{y \in \{0,1,\ldots,100\}} \left\{ \begin{array}{ll} \displaystyle\min_{p_0,\ldots,p_{100}} & \displaystyle\sum_{d=0}^{100} p_d \cdot g(d, y) \\ \text{s.t.} & p_0, \ldots, p_{100} \geq 0, \displaystyle\sum_{d=0}^{100} p_d = 1, \\ & \displaystyle\sum_{d=0}^{100} d p_d = \mu, \\ & \displaystyle\sum_{d=0}^{100} (d-\mu)^2 p_d = \sigma^2 \end{array} \right\}$$

How to solve?

1. For each possible y, use solver to find the **worst-case probabilities** that resulting in **minimum profit**
2. Choose y with the **highest** minimum profit.

Interpretation of Robust Solution

- Shape of the worst-case distribution:

- Calculate its profit on the **training** and **testing** data

SAA versus Robust: What is the Difference?

- **SAA** is **optimistic**. It bets on the historical pattern repeating. It can get a higher reward but is riskier.
- **Robust** is **pessimistic**. It protects you against bad scenarios. It often leads to a more conservative order quantity and a more reliable, if slightly lower, profit.
- If the testing profit for Robust is higher, it means the SAA model was "overfitted" to the noisy training data.

Summary

- The **Newsvendor Problem** is a fundamental model for decision-making under uncertainty.
- **SAA** is simple and effective if you have high-quality data.
- **Distributionally Robust Optimization** is a powerful tool for when you are uncertain about the true probabilities and want a safer, more reliable plan.
- There is always a **trade-off** between aiming for the highest profit (optimization) and protecting against the worst case (robustness).