Principles of Mathematical Economics applied to a Physical-Stores Retail Business

Ashwin Rao

ICME, Stanford University

March 12, 2021
Mathematical Economics is a vast and diverse area, consisting of different flavors of Optimization and Prediction problems.

We focus on a subset of these problems which are pertinent to running a real-world Retail Business.

In particular, many Retail problems involve Stochastic Optimization...

... that can be viewed from the abstract lens of Mathematical Econ.

We can model these as Markov Decision Processes, eg:

- How to *Supply* optimally given random demand and cost structures
- How to *Price* optimally given random demand and supply

Retail also involves forecasting problems, eg: Demand Forecasting

... as well as optimization problems on strategy/planning/scheduling
A fundamental problem in Retail is Inventory Control

How to move inventory optimally from suppliers to shoppers

Let us view this from the lens of Mathematical Economics

Abstracting to a Supply $\Rightarrow$ Demand optimization problem

Nirvana is when Supply appears “just in time” to satisfy demand

We start with an exposition of two classical simple problems:

- Economic Order Quantity (EOQ) problem
- Newsvendor problem
Assumption: Demand for an item is at a constant rate of $\mu$ units/year
Note the big assumption of deterministic demand
A new order is delivered in full when inventory reaches 0
Fixed cost $K$ for each order of non-zero units
Holding cost $h$/unit/year for storage in store
What is the optimal number of units to order?
To minimize the annual cost of ordering + storage
Note: Deterministic Demand is often an unreasonable assumption
But EOQ is a useful foundation to build intuition
Many extensions to EOQ (eg: EOQ for Perishables)
EOQ concept goes beyond Retail (foundation in Mathematical Econ.)
Solving EOQ

- Assume $Q$ is the order quantity ($Q^*$ is optimal order quantity)
- Then we order at annual frequency $\frac{\mu}{Q}$ (Period $\frac{Q}{\mu}$)
- Annual Ordering Cost is $\frac{\mu K}{Q}$
- Annual Holding Cost is $\frac{hQ}{2}$ (note: average inventory during year is $\frac{Q}{2}$)
- Annual Total Cost is:
  \[ \frac{\mu K}{Q} + \frac{hQ}{2} \]
- Taking derivative w.r.t. $Q$ and setting it to 0 yields:
  \[ Q^* = \sqrt{\frac{2\mu K}{h}} \]
Daily demand for newspapers is a random variable \( x \)

The newsvendor has an estimate of the PDF \( f(x) \) of daily demand

For each newspaper that stays unsold, we suffer a *Holding Cost* \( h \)

Think of \( h \) as the purchase price minus salvage price

For each newspaper we’re short on, we suffer a *Stockout Cost* \( p \)

Think of \( p \) as the missed profits (sale price minus purchase price)

But \( p \) should also include potential loss of future customers

What is the optimum \( \# \) of newspapers to bring in the morning?

To minimize the expected cost (function of \( f \), \( h \) and \( p \))
Solution to the Newsvendor problem

- For tractability, we assume newspapers are a continuous variable $x$.
- Then, we need to solve for the optimal supply $S$ that maximizes

$$g(S) = h \int_0^S (S - x) \cdot f(x) \cdot dx + p \int_S^\infty (x - S) \cdot f(x) \cdot dx$$

- Setting $g'(S) = 0$, we get:

$$\text{Optimal Supply } S^* = F^{-1}\left(\frac{p}{p+h}\right)$$

where $F(y) = \int_0^y f(x) dx$ is the CDF of daily demand

- $\frac{p}{p+h}$ is known as the critical fractile.
- It is the fraction of days when the newsvendor goes “out-of-stock”.
- Assuming the newsvendor always brings this optimal supply $S^*$
- Solution details and connections with Financial Options Pricing [here](#)
The store experiences random daily demand given by PDF $f(x)$
The store can order daily from a supplier carrying infinite inventory
There’s a cost associated with ordering, and order arrives in $L$ days
Like newsvendor, there’s a Holding Cost $h$ and Stockout Cost $p$
This is an MDP where $State$ is current Inventory Level at the store
$State$ also includes current in-transit inventory (from supplier)
$Action$ is quantity to order in any given $State$
$Reward$ function has $h$, $p$ (just like newsvendor), and ordering cost
Transition probabilities are governed by demand distribution $f(x)$
This has a closed-form solution, similar to newsvendor formula
Optimal Ordering Policy

Order-Up-To Level

Order Point

Inventory
The Core of Textbook Problem has this Pictorial Intuition

Probability and Cost against Demand

- Assume it's CDF is $F(\cdot)$
- Stockout Cost (Slope $\rho$)
- Holding Cost (Slope $h$)
- Start-of-Day Optimal Inventory $F^{-1}\left(\frac{\rho}{\rho + h}\right)$
Costs viewed against End-of-Day Inventory

Cost against End-of-Day Inventory

- Stockout cost (Slope \( p \))
- Holding Cost (Slope \( h \))
Under/Over-Capacity Cost against Start-of-Day Inventory
Retail Mantra: “Stack it high and watch it fly”
Customers like to see shelves well stocked
Visual emptiness is known to be a sales deterrent
So, full-looking shelves are part of presentation strategy
At a certain level of emptiness, the deterrent rises sharply
Hence the convex nature of this cost curve
Note that this curve varies from item to item
It also varies from regular season to end of season
Modeling/calibrating this is tricky!
However, getting a basic model in place is vital
Retail store backrooms have limited capacity
Typically tens of thousands of items compete for this space
Retailers like to have clean and organized backrooms
A perfect model is when all your inventory is on store shelves
With backroom used purely as a hub for home deliveries
Practically, some overflow from shelves is unavoidable
Hence, the convex nature of this curve
Modeling this is hard because it’s a multi-item cost/constraint
Again, getting a basic model in place is vital
What other costs are involved?

- **Holding Cost:** Interest on Inventory, Superficial Damage, Maintenance
- **Stockout Cost:** Lost Sales, sometimes Lost Customers
- **Labor Cost:** Replenishment involves movement from truck to shelf
- **Spoilage Cost:** Food & Beverages can have acute perishability
- **End-of-Season/Obsolescence Cost:** Intersects with Clearance Pricing
The store experiences random daily demand
The store can place a replenishment order in casepack multiples
This is an MDP where State is current Inventory Level at the store
State also includes current in-transit inventory (from warehouse)
Action is the multiple of casepack to order (or not order)
Reward function involves all of the costs we went over earlier
State transitions governed by demand probability distribution
Solve: Dynamic Programming or Reinforcement Learning Algorithms
In practice, Inventory flows through a network of warehouses
From source (suppliers) to destination (stores or homes)
So, we have to solve a multi-“node” Inventory Control problem
State is joint inventory across all nodes (and between nodes)
Action is recommended movements of inventory between nodes
Reward is the aggregate of daily costs across the network
Space and Throughput constraints are multi-item costs/constraints
So, real-world problem is multi-node and multi-item (giant MDP)
Clearance Pricing

- You are a few weeks away from end-of-season (e.g., Christmas Trees)
- Assume you have too much inventory in your store
- What is the optimal sequence of price markdowns?
- Under (uncertain) demand responding to markdowns
- So as to maximize your total profit (sales revenue minus costs)
- Note: There is a non-trivial cost of performing a markdown
- If price markdowns are small, we end up with surplus at season-end
- Surplus often needs to be disposed at poor salvage price
- If price reductions are large, we run out of Christmas trees early
- “Stockout” cost is considered to be large during holiday season
MDP for Clearance Pricing

- **State** is [Days Left, Current Inventory, Current Price, Market Info]
- **Action** is Price Markdown
- **Reward** includes Sales revenue, markdown cost, stockout cost, salvage
- **Reward & State**-transitions governed by *Price Elasticity of Demand*
- Real-world *Model* can be quite complex (eg: competitor pricing)
- Ambitious Idea: Blend Inventory and Price Control into one MDP
I always start with a simple version of problem to develop intuition

My first line of attack is DP customized to the problem structure

RL Algorithms that are my personal favorites (links to lectures):
- Deep Q-Network (DQN): Experience Replay, 2nd Target Network
- Least Squares Policy Iteration (LSPI) - Batch Linear System
- Exact Gradient Temporal-Difference (GTD)
- Policy Gradient (esp. Natural Gradient, TRPO)

Separate Model Estimation from Policy Optimization

So we could customize RL algorithms to take advantage of:
- Knowledge of transition probabilities
- Knowledge of reward function
- Any problem-specific structure that simplifies the algorithm

Feature Engineering based on known closed-form approximations

Many real-world, large-scale problems ultimately come down to suitable choices of DNN architectures and hyperparameter tuning
The core forecasting problem is Customer Demand Forecasting.

Demand Forecasting is a supervised learning problem (DNNs!)

Assume Inventory+Price Control jointly outputs following Policy:

\[ f : (Time, Inventory, Price) \rightarrow (InventoryMoves, PriceChanges) \]

Assume Demand Forecasting outputs following function:

\[ g : (Time, Price) \rightarrow \text{Demand PDF} \]

We can write a simple simulator that takes as input functions \( f \) and \( g \) and produces as output future projections of Expected Demand.

This time-series of Demand can be aggregated by items/location/time.

Another simulator might produce future projections of Sales/Profits.

Or future projections of Inventory/Labor/Trucks/Boxes etc.

Note: Function \( g \) is an input to Inventory+Price Control MDP.
Inputs to Inventory Control MDPs (other than the costs)

- Demand Forecast function (function $g$ on previous slide)
- Shelf Capacity
- Casepack size
- Lead Time (time from replenishment order to arrival on shelf)
Where do these inputs come from?

- We’ve already discussed Demand Forecast function
- The other inputs come from outputs of Planning problems
- Planning problems are Optimization problems
- Some Planning Problems:
  - Assortment Selection
  - Shelf-size Planning
  - Casepack Sizing
  - Network Planning (for Lead Time)
  - Labor Planning
- Some planning problems need as input solution to Inventory Control
- How do we resolve this Chicken-and-egg situation?
The *Fixed-Point* of Planning and Control

- Planning problems are optimizations over parameter choices $p$
- For example, a set of Shelf-size choices or casepack choices
- Denote the Inventory Control MDP as $D_p$ ($p$ is input to MDP)
- Denote the Solution (Optimal Policy) to $D_p$ as $\pi_p^*$
- Solve the planning problems (optimization) with input $\pi_p^*$
- The solution is the optimal parameter set $p^*$
- Feed $p^*$ back into the MDP to solve for policy $\pi_{p^*}$
- Iterate this until we get stable $p^*$ and $\pi_{p^*}$
- Very important to design the interfaces consistently
- Clean software framework for overall system design is vital to success