Optimization

Linear Programming Topics

- Sensitivity Analysis
- Duality
- Integer Programming

Linear Programming Assumptions:

1. Proportionality
2. Additivity
3. Divisibility
4. Certainty

Linear Programming function:

Decision x**:** $x \varepsilon R^{+}$,

Objective $\int(.)$**:** $c_1 x_1 + c_2 x_2$,

Constraints g(.): $a_{1x} x_1 + a_2 x_2 < b$

• Linear programming applies when all decisions are continuous

Basic Price Optimization Function :

Mathematical Model

Maximize

$$\int (x, \theta)$$

Subject to

$$g\ (x,\ \theta)$$

$\int(.)$: Objective Function

$g(.)$: Constraints

x: Decision variables

θ: Parameters

Price Optimization

Decisions:

Prices and/or allocation of different products and services to different customers through different channels

Objective:

Usually to maximize expected net contributions

Constraints:

One or more

- Physical limitation in capacity or inventory
- Business or marketing rules
- Legal limitation., etc.

Primal vs. Dual

Maximize $C^t x$

Subject to $X <= 0$

Minimize $b^t y$

Subject to $A^t y \geq c$

$y \geq 0$

Duality Theorems

Weak Duality

$$C^T x \leq y^t b$$

Strong Duality

$$C^T x = y^t b$$

Duality Theorem

If we transform the dual into an equivalent minimization problem and then form its dual, we obtain a problem equivalent to the original problem.

Integer programming applies when all decisions are discrete.

Decision x**:** $x = \{0, 1\}$, Objective$\int(.)$: $c_1 x_1 + c_2 x_2$, Constraints: $a_1 x_1 + a_2 x_2 < b$

Integer programs are optimization problems that require some or all of the variables to take integer values

$$min_x\, C^T x$$
$$Ax \geq b$$
$$x \geq 0 \text{ and integral,}$$

where A ∈ IRm×n, b ∈ IRm, c ∈ IRn are given, and x ∈ INn is the variable vector to be determined.

An important case occurs when the variables xj represent binary decision variables, that is, x ∈ {0,1}n. The problem is then called a 0–1 linear program.

When there are both continuous variables and integer constrained variables, the problem is called a **mixed integer linear program (MILP):**

$$min_x\, C^T x$$
$$Ax \geq b$$
$$x \geq 0$$
$$x_j \in IN\, for\, j = 1,,,,,p.$$

where A,b,c are given data and the integer p (with 1 ≤ p < n) is also part of the input.

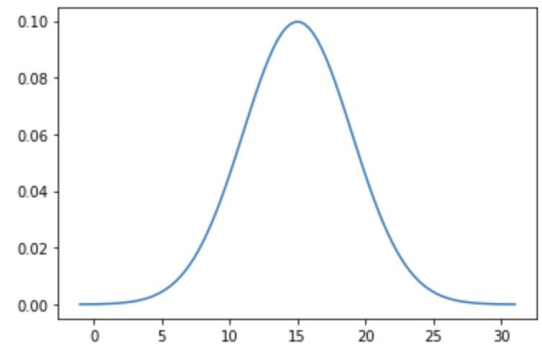
Willingness-to-Pay (WTP), Demand Curves and Profit Functions

Basic Idea:

Use Willingness-to-Pay curve to develop demand function and use the demand function to develop profit function. After getting the profit function, using optimization techniques to get the maximum profit.

For example, if the **willingness to pay** is Normally distributed:

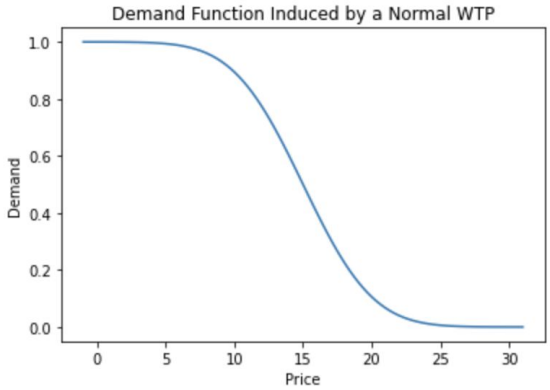
$$w(p) \sim N\left(\mu, \sigma^2\right)$$



The x is the price for the product and y is the probability that someone is going to pay for the product when the price is x. If the price is 15, all the people that have the WTP price higher than 15 are going to pay for the product. The number of people that would buy the product is the cumulative proportion （or the Integral) on the right of 15.

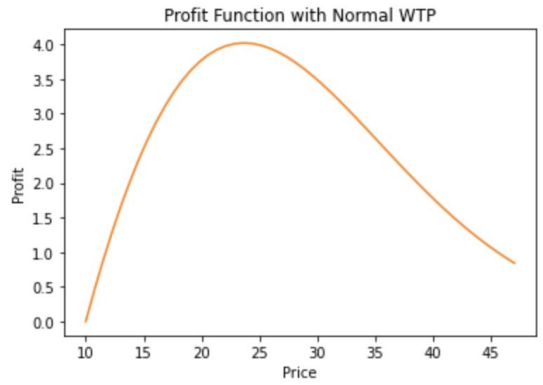
So the **demand curve** would be:

$$d(p) = M \times \left[1 - \Phi\left(\frac{p-\mu}{\sigma}\right)\right]$$



This would decide the **profit function**, which is:

$$d(p) \times (p-c) = M \times \left[1 - \Phi\left(\frac{p-\mu}{\sigma}\right)\right] \times (p-c)$$



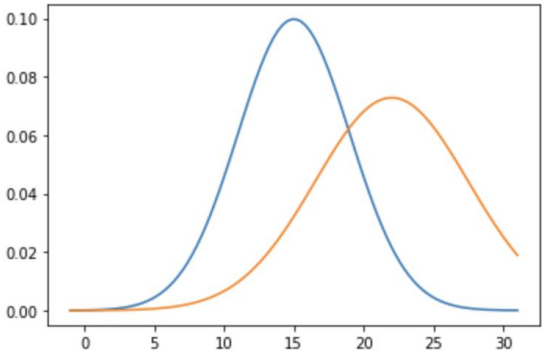
And the profit function is the function that needs to be maximized.

This is an example when the WTP follows the normal distribution. The procedure will be the same when the WTP curve is Uniform, Exponential etc..

Two Segment Case

In real life, there may be more than one segment of customers who have a different distribution of WTP.

For example, if there are two segments of customer following two different normal distribution:



There will be two demand functions.

If the company decided to charge **two segments two different prices**, the solution would be to do the one segment profit optimization twice. Each optimization is the same as above.

If the company decided to charge only **one price for both segments**, the solution is to use the same price for both profit functions, and combine the two profit functions into one profit function and maximize the combined function.

Pricing Analytic

Basic Idea:

Make a model to imitate profit function with different variables. Maximize the profit function with different price/discount.

Sample procedure for B2B pricing:

1. Decide on the model for whether the customer will buy the product or not. For example, the model might be a logistic regression containing the type of product it sells or the discount.
2. Get the parameter of the model and put it in a profit function.
3. Maximize the profit function with discount as the number that needs to be adjusted.

Dynamic Programming in Optimization

Intuition:

Procedures:

a. Characterize the structure of an optimal solution.b. Recursively define the value of an optimal solution. c. Compute the value of an optimal solution in a bottom-up fashion.d. Construct an optimal solution from computed information

Elements in Dynamic Programming

- Stages(S)
- state (st)
- Decision (d)
- Optimal Policy (p)

When to use? - Applicable Scenarios

- Stock Market: Buy/Sell gains
- Knapsack1:

$$v(i) = \max_{d:y_d \le i} \{v(i - y_d) + p_d\}.$$

Note that $\mathcal{S}(i) = \emptyset$ and $v(i) = 0$ for $i = 0, 1,$ and 2 in our example.

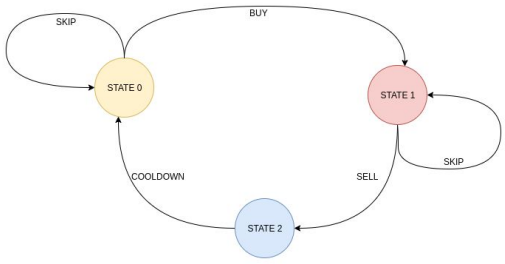
- (1 of each kind of item), (2 OPT vars, double decision), 2D memoize
- Knapsack2 w. infinite qty of each item (limit + multi-way(qty) decision), 2D memoize Weighted Interval Scheduling
- Finding Longest Shortest Path / Segment in a Directed acyclic graph(DAG)

Optimization
Bag trapping problem
Summary

Formulas:

- Algorithm
- Elements in Dynamic programming

Case1: Stock Market



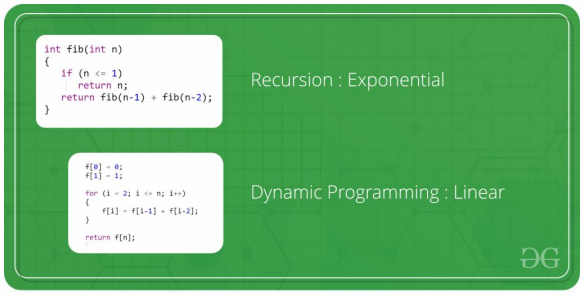
Case2: Knapsack

Intuition: a problem in combinatorial optimization. Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit while **maximizing total value.**

Case3: Shortest & Longest path on a DAG

Dynamic Programming in Coding

Intuition: A recursion with memory



Advantage of Dynamic Programming:

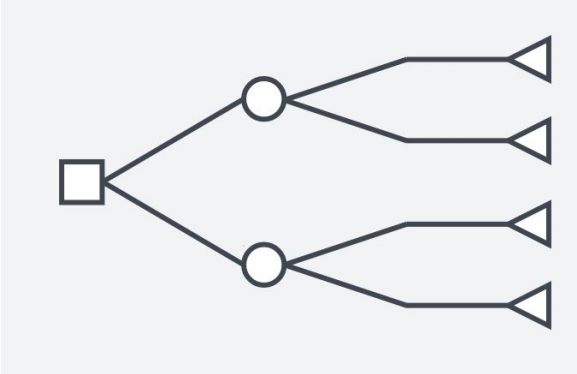
Procedures to Structure a Dynamic Programming solution

(1) find a recursive formula

(2) identify the base cases (initial conditions)

(3) solve the problems such that no problem is solved twice.

Decision Analysis



Decision Tree:

- A diagram of a decision is called a Decision Tree. It is read from left to right. It consists of a root node, chance nodes and an endpoint.

Pros & Cons of using a Decision Tree

- Advantages:
  - applicability
  - problem framing
  - strategic thinking
  - focus on terminals
  - evaluation on information
- Limitations:
  - discrete representation of decisions and uncertainties
  - multiplication of branches

Expected Value:

- The Expected Value for an alternative is calculated by multiplying each possible outcome of the alternative by its probability, and summing the results.
- EV of Imperfect Information = EV with costless Imperfect Information - best EV without the information

EV of Discrete Variable:

$$\mathbb{E}[x] = p_1 x_1 + p_2 x_2 + \cdots + p_n x_n$$

$$\mathbb{E}[x] = \sum_{i=1}^{n} p_i x_i$$

EV of Continuous Variable:

$$\mathbb{E}[x] = \int x f(x) dx$$

Algorithm of Decision Tree:

- Start at the last stage
- For each chance node, calculate the EV
- For each decision node, select the branch with the best EV
- Roll back to preceding decision stage, reiterate step 2 and step 3

Value of Information:

- Value of Information = the EV from having information - the EV from not having it

Thinking Strategically

Game Theory:

- the analysis of strategies for dealing with competitive situations where the outcome of a participant's choice of action depends critically on the actions of other participants
- Applications of game theory: 1) prisoner's dilemma 2) price wars 3) auctions
- Frameworks: Non-cooperative game theory; cooperative theory

- Key concepts: best response function; dominant strategy; Nash Equilibrium v. Stackelberg Equilibrium

Non-cooperative game theory:

	Timing of Actions	
	Simultaneous	Sequential
One period	Static Nash Game	Stackelberg Game
Multiple periods	Dynamic Nash	Dynamic Stackelberg

Nash Equilibrium:

- a stable state of a system involving the interaction of different participants, in which no participant can gain by a unilateral change of strategy if the strategies of the others remain unchanged.

$$\forall\, i, x_i \in S_i : f_i(x_i^*, x_{-i}^*) \geq f_i(x_i, x_{-i}^*)$$

x_i^* is the dominant strategy

Strategic Pricing:

- Demand pricing

$$q_i = a_i - b_i p_i + \gamma_i p_{-i}$$

- Profit functions

$$\Pi_i = q_i(p_i, p_{-j}) \times (p_i - c_i) - F_i$$

- Question

Assume $i = \{1,2\}$. Derive $p_1^*(p_2)$ and $p_2^*(p_1)$

Cooperative game theory:

- Key concept: Shapley value, is a solution concept of fairly distributing both gains and costs to several actors working in coalition. The Shapley value applies primarily in situations when the contributions of each actor are unequal, but they work in cooperation with each other to obtain the payoff
- Use Case: digital attribution model