Stochastic Simulation.

1

HW 2 :-

↳

→ weights.

→ create portfolio

→ over 100 stocks

↳ 100 stocks

250 days

Q.1 → 2019 → maximizes return.

constraint → No more than 100 days have a negative return.

Objective = Maximize the expected return.

decision_variables → ◦ weights of stocks,

Big M will be used ← Breach_boolean for all 250 days.

Ex → 3 stocks

Day-1. → $w_0 s_0 + w_1 s_1 + w_2 s_2 + d\_day1 \times M \geq 0$

var_p → 0, 1

M → ∞

var_c is dependent on var_p.

↳ 0 var_c = 0

↳ 1 var_c = कुछ

var_c ≤ ∞ × var_p.

var_c ≤ M × var_p.

(2) Here parent variable var_p is Indicator variable for day-i

And child variable var_c is return on day-i → wi x return day-i

vector

~~But~~ var_c = 0 if var_p

But var_c < 0 if var_p = 1.

$var\_c > -M \times Var\_p$. (-1000

संवदा रहे

↳ But

* - तभी हो

जब Var_p

1 है।

o $var\_c + M \times var\_p \geq 0$

day-1.

⇒ $w_0 s_0 + w_1 s_1 + w_2 s_2 + M \times I\_day1 \geq 0$

$\Sigma I\_day \leq 100$.

↳ Only 100 negatives allowed

o. Big M Trigger value ε → Find var_p or var_c

→ Build their relation

→ Constrict Σvar_p_i

Newsvendor Problem

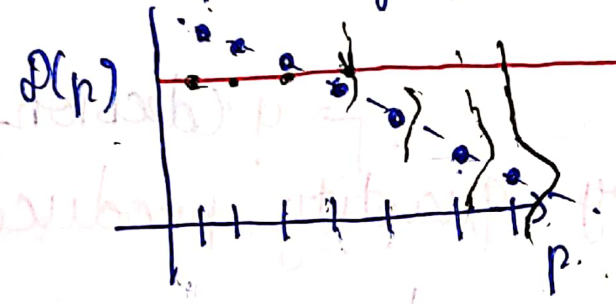
↳ Stochastic Optimization

↳ capacity + Demand based.

$D(p)$ -ve. &

↳ $A + B \times p + \varepsilon$. ↗ $N(0, \sigma)$.

↳ Capped by a bound of capacity.



→ K = Capacity = Quantity to be printed by News Vendor

↳ Set = q.

$D(p)$ is non-linear ↗ if based.

Then we have price (p)

Revenue = $D(p) \times p$. → Can be integrated over all values of price

OR.

Be simulated ↗ by generating ε.

\+

$D(p)$ → using max or if procedure

Cost → Fixed

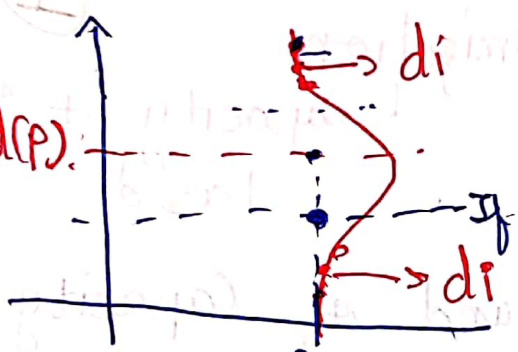
↳ Dependent on quantity to be print

↳ $c \times q$

(Bottomline → Maximum profit is capacity or quantity-printed dependent)

↳ price × capacity.

For single price



q (decision variable)

Profit is still capped by quantity produced

$c \times q \rightarrow$ cost

Revenue $= p \times q$ or $p \times d_i$

↳ dependent on whether the the demand is lower than capacity.

→ hi

Profit → $p \times q - c \times q$; $p \times d_i - c \times q$.

$h_i \leq$ the two profits

↳ Theory of capacity procedure can be used

h_i is derived from $d(p) \rightarrow$ capped by capacity

(3) Basically d_i's are self-simulate points

↳ Else would have been generated using $N(0, \alpha^2)$ ε.

↳ d_i → d_i (capped by q_i) ↳ capacity / order printed

Will be used to compute profit_i

Take average of these profits ⟸

↳ optimize

In data generation we due it using if or max procedure

Here we resolve / solve it using constraints. ←

↳ Instead of $d_i \leftarrow (d_i \text{ or } q_i)$

profit ← (profit by d_i, profit by q_i)