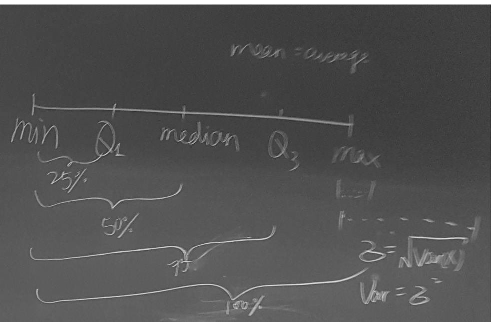

Knapsack Problem

Optimization:

maximise $p_1 x_1 + p_2 x_2 + \cdots + p_n x_n$

subject to $w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \leq W$

$x_1, x_2, \ldots, x_n \geq 0$ and are integer

Assumptions that

(i) $1 \leq w_j \leq W$ for all j

(ii) $p_j > 0$ for all j

(I) Scout

W = volume of knapsack

p_j = value of one of item j

w_j = volume of " " " "

(II) Cargo Plane

W = weight limit

p_j = profit item j

w_j - weight of item j

(III) W = money to invest

p_j = expected profit per unit of stock j

w_j = cost per unit of stock j

Look at a dynamic programming formulation

Knapsack Problem

Optimization:

maximize $p_1 x_1 + p_2 x_2 + \cdots + p_n x_n$

subject to $w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \le W$

$x_1, x_2, \ldots, x_n \ge 0$ and an integer

(i) $1 \le w_j \le W$ for all j

(ii) $p_j > 0$ for all j

(i) Scout

W = volume of knapsack

p_j = value of one of item j

w_j = volume of " " " "

(ii) Cargo plane

W = weight limit

p_j = profit item j

w_j = weight of item j

(iii) W = money to invest

p_j = expected profit per unit of stock j

w_j = cost per unit of stock j

Look at a dynamic programming formulation

$f(r, w)$ = maximum of the variable, replacing n by r and W by w.

$$= \max_{0 \le x_r \le \left\lfloor \frac{w}{w_r} \right\rfloor} \left[p_r x_r + f(r-1, w - w_r x_r) \right]$$

#steps to compute $f(n, W) = O(nW^2)$

Reduce time to $O(nW)$

Instead of deciding how big is x_r, decide whether $x_r = 0$ or ≥ 1

$$f(r, w) = \max \begin{cases} f(r-1, w) & : x_r = 0 \\ f(r, w - w_r) + p_r & : x_r \ge 1 \end{cases}$$

#steps $= O(nW)$

$f(r, w)$ = maximum obtainable, replacing W by w.

$$= \max_{0 \le x_r \le \left\lfloor \frac{w}{w_r} \right\rfloor} \left[p_r x_r + f(r-1, w - w_r x_r) \right]$$

#steps to compute $f(n, W) = O(nW^2)$

Instead of deciding how big is x_r, decide whether $x_3 = 0$ or ≥ 1

$$f(r, w) = \max \begin{cases} f(r-1, w) : & x_r = 0 \\ f(r, w - w_r) + p_r : & x_r \ge 1 \end{cases}$$

#steps $= O(nW)$

Maximise $x_1 + 3x_2 + 6x_3 + 13x_4$

st $w_1 + 2w_2 + 3w_3 + 4x_4 \le 10$

	r=4		r=3		r=2		r=1	
w	f	b	f	b	f	b	f	b
0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	1	1
2	3	0	3	0	3	1	2	1
3	6	0	6	1	4	1	3	1
4	13	1	7	1	6	1	4	1
5	14	1	9	1	7	1	5	1
6	16	1	12	1	9	1	6	1
7	19	1	13	1	10	1	7	1
8	26	1	15	1	12	1	8	1
9	27	1	18	1	13	1	9	1
10	29	1	19	1	15	1	10	1

Does not mean exactly one item it means one more.

Max = 29

x_1	x_2	x_3	x_4
0	1	0	1