动态规划算法
解背包问题

背包问题 (Knapsack Problem)

一个旅行者随身携带一个背包. 可以放入背包的物品有n 种, 每种物品的重量和价值分别为 w_i, v_i. 如果背包的最大重量限制是 b, 每种物品可以放多个. 怎样选择放入背包的物品以使得背包的价值最大？ 不妨设上述 w_i, v_i, b 都是正整数.

实例： $n = 4$, $b = 10$

$v_1 = 1$, $v_2 = 3$, $v_3 = 5$, $v_4 = 9$,

$w_1 = 2$, $w_2 = 3$, $w_3 = 4$, $w_4 = 7$,

建模

解是< $x_1, x_2,\dots, x_n$ >，其中 x_i 是装入背包的第 i 种物品个数

目标函数 $\max \sum_{i=1}^{n} v_i x_i$

约束条件 $\sum_{i=1}^{n} w_i x_i \le b, \quad x_i \in \mathrm{N}$

线性规划问题：由线性条件约束的线性函数取最大或最小的问题

整数规划问题：线性规划问题的变量 x_i 都是非负整数

子问题界定和计算顺序

子问题界定：由参数 k 和 y 界定

k：考虑对物品1, 2, … , k 的选择

y：背包总重量不超过 y

原始输入：$k = n$, $y = b$

子问题计算顺序：

$k = 1, 2, \dots, n$

对于给定的 k，$y = 1, 2, \dots, b$

优化函数的递推方程

$F_k(y)$：装前 k 种物品，总重不超过 y，背包达到的最大价值

$$\boxed{F_k(y)=\max\{F_{k-1}(y),F_k(y-w_k)+v_k\}}$$

$$F_0(y)=0,\quad 0\le y\le b,\quad F_k(0)=0,\quad 0\le k\le n$$

$$F_1(y)=\left\lfloor \frac{y}{w_1}\right\rfloor v_1,\quad F_k(y)=-\infty\quad y<0$$

类似于投资问题,如何写递推方程

这两种写法有什么区别

$$F_k(y)=\max_{0\le x_k\le\lfloor y/w_k\rfloor}\{F_{k-1}(y-x_kw_k)+x_kv_k\}$$

标记函数

$i_k(y)$：装前 k 种物品，总重不超 y, 背包达到最大价值时装入物品的最大标号

$$i_k(y)=\begin{cases} i_{k-1}(y) & F_{k-1}(y)>F_k(y-w_k)+v_k \\ k & F_{k-1}(y)\le F_k(y-w_k)+v_k \end{cases}$$

$$i_1(y)=\begin{cases} 0 & y<w_1 \\ 1 & y\ge w_1 \end{cases}$$

实例

输入： $v_1=1,\ v_2=3,\ v_3=5,\ v_4=9,$

$w_1=2,\ w_2=3,\ w_3=4,\ w_4=7,$

$b=10$

$F_k(y)$ 的计算表如下：

$k \backslash y$	1	2	3	4	5	6	7	8	9	10
1	0	1	1	2	2	3	3	4	4	5
2	0	1	3	3	4	6	6	7	9	9
3	0	1	3	5	5	6	8	10	10	11
4	0	1	3	5	5	6	9	10	10	12

追踪解

$k \backslash y$	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	1	1	1	1	1
2	0	1	2	2	2	2	2	2	2	2
3	0	1	2	3	3	3	3	3	3	3
4	0	1	2	3	3	3	4	3	4	4

$i_4(10)=4 \Rightarrow x_4 \geq 1$

$i_4(10-w_4)=i_4(3)=2 \Rightarrow x_2 \geq 1,\ x_4=1,\ x_3=0$

$i_2(3-w_2)=i_2(0)=0 \Rightarrow x_2=1,\ x_1=0$

解 $x_1=0,\ x_2=1,\ x_3=0,\ x_4=1$，价值12

追踪算法

算法 Track Solution

输入: $i_k(y)$表, k=1,2,…,n, y=1,2,…,b

输出: x_1, x_2,…,x_n， n种物品的装入量

1. for $k \leftarrow 1$ to n do $x_k \leftarrow 0$
2. $y \leftarrow b$, $k \leftarrow n$
3. $j \leftarrow i_k(y)$
4. $x_k \leftarrow 1$
5. $y \leftarrow y - w_k$
6. while $i_k(y)=k$ do
7. $\qquad y \leftarrow y - w_k$
8. $\qquad x_k \leftarrow x_k + 1$
9. if $i_k(y) \neq 0$ then goto 4

初始追踪位置

继续放k种物品

继续追下一种

时间复杂度 $O(nb)$

根据公式

$$F_k(y) = \max\{F_{k-1}(y), F_k(y-w_k)+v_k\}$$

备忘录需计算 nb 项，每项常数时间，计算时间为 $O(nb)$.

伪多项式时间算法：时间为参数 b和 n的多项式，不是输入规模的多项式.

参数 b 是整数，表达 b 需要 $\log b$ 位，输入规模是 $\log b$.

背包问题的推广

物品数受限背包：第 i 种物品最多用 n_i 个

0-1背包问题： $x_i = 0, 1$， $i = 1, 2, \ldots, n$

多背包问题：m个背包，背包 j 装入最大重量 B_j, $j = 1, 2, \ldots, m$. 在满足所有背包重量约束条件下使装入物品价值最大.

二维背包问题：每件物品有重量 w_i 和体积 t_i, $i = 1, 2, \ldots, n$，背包总重不超过 b，体积不超过V, 如何选择物品以得到最大价值.

小结

- 划分子问题，确定子问题边界，将问题求解转变成多步判断的过程.
- 定义优化函数,以该函数极大(或极小)值作为依据,确定是否满足优化原则.
- 列优化函数的递推方程和边界条件
- 自底向上计算，设计备忘录 (表格)
- 考虑是否需要设立标记函数
- 时间复杂度估计