12 Feb 2018. The Knapsack Problem

Given budget W and n items where item i has
value v_i } $\in$ integers
weight w_i

Select set of items with combined weight $\leq W$,
and maximize combined value under this constraint.

Today. Algorithm with running time $O(nW)$.

"Pseudo-polynomial time"

If input were written in binary, W would be represented using $\lceil \log(W) \rceil$ bits, so $O(nW)$ is potentially exponential in input size.

"Pseudo-polynomial" means if all the numbers were represented in unary (e.g. representing 8 as ||||||||) then the running time is polynomial in the size of that input representation.

Dynamic Program for Knapsack.

Question 1. What's the last decision you would need to make when assembling a solution?

Ans. Include the n-th item or leave it out?

Question 2. What information would you need to have on hand to make that decision optimally?

Ans. [a.] If the opt. solution leaves out n^{th} item, then it is an opt knapsack solution for budget W and items $1, \ldots, n-1$.

[b.] If the opt. solution includes the n^{th} item then it is an opt. knapsack solution for budget $W - w_n$ and items $1, \ldots, n-1$.

Question 3. What information should the dynamic programming table store, to ensure this information is always on hand?

Ans. $T[i,j]$ should store maximum value knapsack solution using items $1, \ldots, i$ and with total weight $\leq j$.

$T[i,j]$ should store maximum value knapsack solution using items $1,\dots,i$ and with total weight $\le j$.

Algorithm for Knapsack

Initialize $T[0,j] = 0$ for $j = 0,\dots,W$.
for $i = 1,\dots,n$
    for $j = 0,\dots,W$
        if $j < w_i$ then $T[i,j] = T[i-1,j]$
        else $T[i,j] = \max\{T[i-1,j],\ v_i + T[i-1, j-w_i]\}$
    end for
end for

→ Output $T[n,W]$.

Initialize $S = \emptyset$. Initialize $B = W$.
for $i = n, n-1, \dots, 1$
    if $(w_i \le B)$ && $(T[i,B] = v_i + T[i-1, B-w_i])$
        $S \leftarrow S \cup \{i\}$
        $B \leftarrow B - w_i$
    end if
end for

→ Output S.

Proving Correctness. For the sake of brevity, this proof will only show that $T[n,W]$ equals the value of the opt knapsack solution.

Induction hypothesis:

$T[i,j]$ should store maximum value knapsack solution using items $1,\dots,i$ and with total weight $\le j$.

Induction over pairs (i,j) in the order that the alg fills in the corresponding table entries.

Base Case: $i=0$, items $1,\dots,i$ denotes an empty set, so max value $= 0$.

Induction Step: The opt solution with items $1,\dots,i$ and budget j either includes i or it doesn't. If it includes i, it must be the case that $j \ge w_i$, and the remainder of the opt knapsack solution uses items $1,\dots,i-1$ and budget $j-w_i$. Since induction hypothesis implies $T[i-1,j-w_i]$ is the best we can get from those items with that budget, the opt value is $T[i-1, j-w_i] + v_i$ in this case.

In the other case, the opt knapsack solution excludes item i and, again using induct hypoth, has value $T[i-1, j]$.

Hence the opt solution in all cases has value

$$\begin{cases} T[i-1, j] & \text{if } w_i > j \\ \max\{T[i-1,j],\ v_i + T[i-1, j-w_i]\} & \text{if } w_i \le j. \end{cases}$$

matching the formula in the pseudocode.