Knapsack Problem 1c: Proof

Knapsack Combined Proof by construction

Given the proofs below for 0-1 knapsack and fractional knapsack the proof of combined knapsack is as follows.

Fractional Knapsack produces an optimal set of fractionable items for given weight w O_f: [f_0, f_1,... f_n]

0-1 Knapsack produces an optimal set of indivisible items for given weight w $O_{0\text{-}1}$. [x_0, x_1, … x_n]

For computing the optimal combined knapsack for a given weight w: we divide the weight into m and n w = m + n. Where O for w = $O_{0\text{-}1}$ for m and O_f for n.

We use the 0-1 knapsack table to find O' where weight is divided into m' and n'.

The weight capacity w is divided into i + w-i. We find $O_{0\text{-}1}$ for each i and O_f for each i-w. We add $O_{0\text{-}1}$ and O_f for each element in the table. And get max of the elements for getting the optimal profit p', for any set P(O) = p', then the set O' is the optimal set.

Knapsack 0-1 Proof

2.3 Correctness Proof for Knapsack

We will do proof by induction here:

We say pair $(i, j) < (i', j')$ if $i < i'$ or ($i = i'$ and $j < j'$). For example

$$(0,0) < (0,1) < (0,2) < ... < (1,0) < (1,1) < ...$$

Induction Hypothesis: algorithm is correct for all values of $a[i, j]$ where $(i, j) < (i', j')$. Or in other words, all previous elements in table are correct.

Base Case: $a[i, 0] = a[0, j] = 0$ for all i, j

Induction Step: When computing $a[i', j']$, by induction hypothesis, we have $a[i' - 1, j']$, $a[i' - 1, j' - w_{i'}]$ are already computed **correctly**. Then algorithm considers the optimal value for item i' in knapsack as $a[i' - 1, j' - w_{i'}] + v_{i'}$ and for item i' not in knapsack as $a[i' - 1, j']$. Therefore,

the value at $a[i', j']$ is correct.

Citation: https://www2.cs.duke.edu/courses/fall17/compsci330/lecture5note.pdf

Fractional Knapsack Proof

Proof of Correctness: Assume towards contradiction that there is an instance of fractional knapsack such that the solution of this algorithm (ALG) is not optimal. Let OPT denote the optimal solution. Without loss of generality, assume that items are sorted in decreasing order by $\frac{v_i}{w_i}$, and that no two items that have

the same $\frac{v_i}{w_i}$. Let $ALG = \{p_1, p_2, \ldots, p_n\}$ denote the sequence of decision (fractions taken) made by our greedy algorithm and $OPT = \{q_1, q_2, \ldots, q_n\}$ denote that in OPT.

Therefore, by assumption we have $\sum_{i=1}^{n} p_i v_i < \sum_{i=1}^{n} q_i v_i$. Let i be there first index at which $p_i \neq q_i$. By the design of our algorithm, it must be that $p_i > q_i$. By the optimality of OPT, there must exist an item $j > i$ such that $p_j < q_j$. Consider a new solution $q' = \{q'_1, q'_2, \ldots, q'_n\}$ where $q'_k = q_k$ for all $k \neq i, j$. q' will take a little more of item i and a little less of item j compared to OPT: $q'_i = q_i + \epsilon$, $q'_j = q_j - \epsilon \frac{w_i}{w_j}$. The total weight does not change: $\sum_{i=1}^{n} q'_i w_i = \sum_{i=1}^{n} q_i w_i$. Yet the total value strictly increases: $\sum_{i=1}^{n} q'_i v_i = \sum_{i=1}^{n} q_i v_i + \epsilon v_i - \epsilon \frac{w_i}{w_j} v_j > \sum_{i=1}^{n} q_i v_i$.

That q' is a valid and better solution than OPT is a contradiction. Hence, our the solution from our algorithm is optimal. QED

Citation: https://www2.cs.duke.edu/courses/spring19/compsci330/lecture8scribe.pdf

Complexity

The complexity of the problem is dependent on the complexity of the subproblems, The complexity of 0-1 knapsack is O(n*V) where n is the number of indivisible items items and V is the weight capacity. For fractional it is O(m) where m is the number of divisible items. We compute fractional for weights 1-V so we have complexity V*O(m): O(V*m). The complexity of the algorithm is O(n*V + m*V) = O((m+n)*V) = O(K*V), where K = m+n.