Multiple Choice Knapsack Problem

congyu

December 1, 2022

1. MCKP
2. Generalized MCKP

1. MCKP
2. Generalized MCKP

Multiple Choice Knapsack Problem

The multiple-choice knapsack problem (MCKP) is a generalization of the ordinary knapsack problem, where the set of items is partitioned into classes. The binary choice of taking an item is replaced by the selection of exactly one item out of each class of items.

Linear relaxation of the IP:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} x_{ij} \\
s.t. \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij} \leq B \\
& \sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1, \ldots, n, \\
& x_{ij} \geq 0 \qquad i = 1, \ldots, n, j = 1, \ldots, m
\end{aligned}
$$

Why do we consider the lp?

- can this lp get a good approximation ratio?
- can this lp be solved fast?

The solution obtained from lp is at most 1 group away from the optimum and we can solve the lp in **linear time**(MCKP can be solved in $O(Bnm)$ through dynamic programming).

Theorem 1

Sinha and Zoltners proved that a basic solution to the linear program has at most two fractional variables and in an optimal solution two fractional variables are from the same group.

The basic solutions to the lp has $n+1$ variables since there are $n+1$ constraints, then there are at most $n+1$ non-zero variables. Consider the constraint $\sum_{j=1}^{m} x_{ij} = 1$, there will be at most 2 fractional variables and two fractional variables are from the same group.

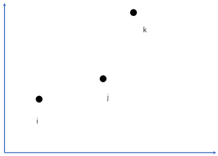
Theorem 2

If two items i and j in the same group satisfy $c_i \geq c_j$ and $v_i \leq v_j$, then lp optimal solution with $x_i = 0$ exists.
If three items i, j, k in the same group with $v_i < v_j < v_k$ and $c_i < c_j < c_k$ satisfy

$$\frac{v_k - v_j}{c_k - c_j} \geq \frac{v_j - v_i}{c_j - c_i}$$

then an optimal solution to the lp with $x_j = 0$ exists.

The first case is obvious.
The second case can be observed on the $v - c$ plane.



Items with $x \neq 0$ in the optimal solution are on the upper-left convex hull.

algorithm for lp

Greedy: initially we select item with the smallest weight in every group(this must be a feasible solution), then we iteratively move to the "next" item on the LP-extreme of every group(select item in some group with larger weight and larger value and abandon the previously selected one, but we have to select the "next" item with maximal efficiency $\frac{\Delta v}{\Delta c}$. The efficiency is monotonously decreasing because the LP-extreme items are one the upper left convex hull), then finally we found that the weight limit is exceeded when moving to the next item on LP-extreme, then 2 fractional variables are generated and the final slope is the current efficiency.

algorithm for lp

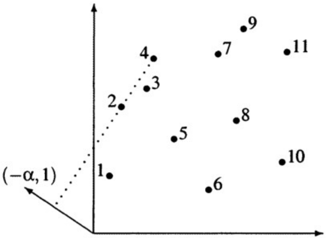
$O(nm \log m + nm \log n)$: find convex hull for each group, greedily pick items until $\sum x_{ij} c_{ij} = B$. (fractional x appears while selecting some item k but current cost $B' < B - c_k$)

Can we skip computing the convex hull?

Optimal slope is the incremental efficiency of the last item added in the greedy algorithm.

Theorem 3

If the optimal slope $\alpha*$ is known, then the corresponding optimal solution to the lp can be determined in $O(n)$ time



Now the problem is how to find the optimal slope in linear time.
For each group we determine $M_i(\alpha) = \arg\max_j \{v_{ij} - c_{ij}\alpha\}$, $M_i(\alpha)$ may contains more than one items, we define $a_i = \arg\min_{j \in M_i(\alpha)}\{c_{ij}\}$, $b_i = \arg\max_{j \in M_i(\alpha)}\{c_{ij}\}$, if α is the optimal slope, we have

$$\sum_{i=1}^{n} c_{ia_i} \leq B \leq \sum_{i=1}^{n} c_{ib_i}$$

then we can guess a slope α and test whether it is the optimal slope.

Algorithm 1: Algorithm Dyer-Zemel

Input: n groups, each group has m items (c_{ij}, v_{ij})
Output: optimal slope $\lambda*$.

```
while true do
    for i=1,...,n do
        Pair the items in group i two by two as (ij, ik);
        Order each pair such that c_ij ≤ c_ik breaking ties such that v_ij ≥ v_ik;
        if item c_ij ≤ c_ik and v_ij ≥ v_ik:
        Delete item k and pair j with another item.
    end
    for i=1,...,n do
        if the group i has only one item j left:
        Decrease the budget B = B − c_ij;
    end
    for all pairs (ij, ik) do
        Derive slope λ = (v_ik − v_ij)/(c_ik − c_ij);
        λ* = the median of the slopes;
    end
    if λ* is the optimal slope then
        return λ*;
    end
    if Σ_{i=1}^{m} c_{ia_i} ≥ B then
        for all pair (ij, ik) with λ ≤ λ* delete item k;
    end
    if Σ_{i=1}^{m} c_{ib_i} < B then
        for all pair (ij, ik) with λ ≥ λ* delete item j;
    end
end
```

time complexity

Proof. Assume that all items and all classes are represented as lists, such that deletions can be made in $O(1)$. At any stage, n_i refers to the current number of items in class N_i and m is the current number of classes. We will use the terminology (ij, ik) to denote a pair of items $j, k \in N_i$. Notice that each iteration of Steps 1-7 can be performed in time linear in the current number of items.

There are $\sum_{i=1}^{m} \lfloor n_i/2 \rfloor$ pairs of items (ij, ik). Since α is the median of $\{\alpha_{ijk}\}$, half of the pairs will satisfy the criteria in Steps 6 or 7, and thus one item from these pairs will be deleted, i.e. at least $\frac{1}{2}\sum_{i=1}^{m} \lfloor n_i/2 \rfloor$ items are deleted out of $n = \sum_{i=1}^{m} n_i \leq \sum_{i=1}^{m}(2\lfloor n_i/2 \rfloor + 1)$. Since $\lfloor n_i/2 \rfloor \geq 1$, each iteration deletes at least

$$\frac{\frac{1}{2}\sum_{i=1}^{m} \lfloor n_i/2 \rfloor}{\sum_{i=1}^{m}(2\lfloor n_i/2 \rfloor + 1)} \geq \frac{\sum_{i=1}^{m} \lfloor n_i/2 \rfloor}{2m + 4\sum_{i=1}^{m} \lfloor n_i/2 \rfloor} \geq \frac{1}{6}, \tag{11.12}$$

of the items. The running time now becomes

$$O\left(n + \frac{5}{6}n + \left(\frac{5}{6}\right)^2 n + \left(\frac{5}{6}\right)^3 n + \dots\right) = O\left(\frac{n}{1 - \frac{5}{6}}\right) = O(n), \tag{11.13}$$

which shows the claimed. □

1. MCKP
2. Generalized MCKP

Generalized MCKP

The difference between generalized MCKP and MCKP is that we do not only demand a single item to be chosen from each group, but any strict cardinality constraint.
lp:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} v_{ij}x_{ij} \\
s.t. \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \leq B \\
& \sum_{j=1}^{m} x_{ij} = \mathbf{p} \qquad i = 1,\ldots,n, \\
& x_{ij} \geq 0 \qquad i = 1,\ldots,n, j = 1,\ldots,m \\
& x_{ij} \leq 1 \qquad i = 1,\ldots,n, j = 1,\ldots,m
\end{aligned}
$$

a more linear programming perspective

The greedy algorithm is confusing for $p > 1$ case, instead we consider the lagrangian dual of the LP:

$$\min_{\lambda} \left(\max_{x} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} x_{ij} + \lambda (B - \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}) \right)$$

$$\begin{aligned} s.t. \sum_{j=1}^{m} x_{ij} &= p & i &= 1, \ldots, n, \\ x_{ij} &\leq 1 & i &= 1, \ldots, n, j = 1, \ldots, m \\ x_{ij} &\geq 0 & i &= 1, \ldots, n, j = 1, \ldots, m \\ \lambda &\geq 0 \end{aligned}$$

This is a minimax parametric optimization problem.

lagrangian dual

Lagrangian dual is an upperbound for the primal lp and has the same optimum as primal lp if and only if complementary slackness condition($\lambda(B - \sum\sum c_{ij}x_{ij}) = 0$) is satisfied.

Assume that λ is known(we solve this optimization problem by enumerating λ), then the remaining problem is:

$$\max_x \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_{ij}$$

$$s.t. \sum_{j=1}^{m} x_{ij} = p \qquad i = 1, \dots, n,$$

$$x_{ij} \leq 1 \quad i = 1, \dots, n, j = 1, \dots, m$$

$$x_{ij} \geq 0 \quad i = 1, \dots, n, j = 1, \dots, m$$

where $a_{ij} = v_{ij} - \lambda c_{ij}$

p-level

The remaining problem seems independent for each group, the optimal solution can be obtained by sorting a_{ij} in each group and choose top p items in each group. Clearly $x_{ij} = 1$ for top p items in each group and the rest $x_{ij} = 0$ there is no fractional variable.

for each item in some group, $v_{ij} - \lambda c_{ij}$ is a line. For each λ, we want to find the top p lines. This is the k-level problem, can be solved in $O(m \log m + mp^{1/3})$.

$\sum v_{ij} - \lambda c_{ij}$ is a piece-wise linear convex function, the number of breakpoints is $O(mp^{1/3})$.

After running p-level algorithm for each group, our problem becomes:

Given n piece-wise linear convex functions $f_1(\lambda), \ldots, f_n(\lambda)$, find $\arg\max_\lambda B\lambda + \sum_{i=1}^n f_i(\lambda)$.

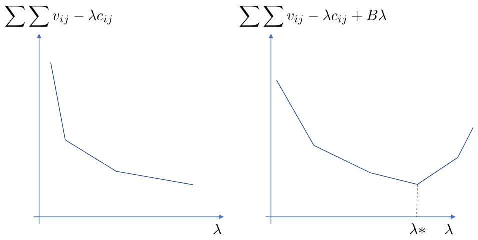
Since $B\lambda + \sum_{i=1}^n f_i(\lambda)$ is also a piece-wise linear convex function, we can easily solve this problem.

p-level algorithm has time complexity $O(m \log m + mp^{1/3})$ and we use p-level for each group, so the preprocessing has time complexity $O(nm \log m + nmp^{1/3})$; the merge process uses a priority queue of at most n elements and the number of breakpoints for p-level has an upperbound $O(mp^{1/3})$, the total time complexity is $O(nm \log m + mnp^{1/3} \log n)$.

Theorem 4

when $p > 1$ a basic solution to the linear program has at most two fractional variables and in an optimal solution two fractional variables are from the same group.

Lagrangian dual has the same optimum as primal lp if and only if complementary slackness condition($\lambda(B - \sum\sum c_{ij}x_{ij}) = 0$) is satisfied. Since $\lambda > 0$ in our algorithm, we need to make sure $\sum\sum c_{ij}x_{ij} = B$, this situation is very similar to $\sum_{i=1}^{n} c_{ia_i} \leq B \leq \sum_{i=1}^{n} c_{ib_i}$ in $p = 1$ case.



The objective function can be rewritten as $\min \sum_{i\in[n]} \sum_{j:\text{top } p} v_{ij} + \lambda(B - \sum_{i\in[n]} \sum_{j:\text{top } p} c_{ij})$. The left derivative at $\lambda*$ is negative and right derivative is positive, that means $B - \sum_{i\in[n]} \sum_{j:\text{top } p} c_{ij} \leq 0$ holds for the left line segment of $\lambda*$ and $B - \sum_{i\in[n]} \sum_{j:\text{top } p} c_{ij} \geq 0$ holds for the right line segment. So obviously a linear combination of x_{ij} of those two items because of which this breakpoint $\lambda*$ is generated can fully fill the budget B.