Material Didático sobre Algoritmos Gulosos

Victor de Oliveira Colombo

Instituto de Matemática e Estatística da Universidade de São Paulo

victor.colombo@usp.br

26 de Novembro de 2018

Objetivos

- Resolver problemas que utilizem algoritmos gulosos aliados a outros tópicos.
- Desenvolver o raciocínio e a intuição por trás das técnicas gulosas
- Criar uma ferramenta sistemática para resolver problemas dessa classe.
- Apresentar algoritmos gulosos de uma maneira pragmática, a partir de problemas de competições de programação.
- Destoar dos problemas clássicos que são frequentemente abordados nos livros-texto.

Problema da Mochila

- São dados n itens e uma mochila de tamanho S.
- Cada item tem um peso w_i e um valor v_i.
- Escolher subconjunto de itens $I \subseteq [1, n]$ tal que

$$\sum_{i \in I} w_i \leq S$$

- Maximizar

$$\sum_{i \in I} v_i$$

Problema da Mochila - Programação Dinâmica

- Seja $f(i,s)$ o valor máximo para o subproblema considerando os itens $i, i+1, ..., n$ e uma mochila de tamanho s.

$$f(i,s) = \begin{cases} 0 & \text{se } i = n+1 \\ max(f(i+1,s), v_i + f(i+1, s-w_i)) & \text{se } w_i \leq s \\ f(i+1,s) & \text{c.c.} \end{cases} \qquad (1)$$

- Resposta é $f(1,S)$
- Programação Dinâmica: complexidade de tempo $O(nS)$

Problema da Mochila modificado

- São dados n itens e uma mochila de tamanho S.
- Cada item tem um peso w_i **após** de ser colocado na mochila, um peso c_i **antes** de ser colocado na mochila e um valor v_i.
- Para todo item, $c_i \geq w_i$.
- Escolher subconjunto de itens $I \subseteq [1, n]$ e uma permutação r de I tal que, para todo j,

$$c_{r_j} + \sum_{i=1}^{j-1} w_{r_i} \leq S$$

- Maximizar

$$V(I) = \sum_{i \in I} v_i$$

Problema da Mochila modificado - Exemplo 1

- $S = 5,\ n = 4,\ v = \{1, 2, 3, 1\},\ c = \{3, 2, 3, 4\},\ w = \{1, 2, 2, 4\}$
- A escolha $I = \{1, 2, 3\}$ e $r = \{1, 2, 3\}$ não é válida, já que:
  - $c_1 = 3 < S$ ✓
  - $c_2 + w_1 = 2 + 1 = 3 < S$ ✓
  - $c_3 + w_2 + w_1 = 3 + 2 + 1 = 6 > S$ ⚠

Problema da Mochila modificado - Exemplo 2

- $S = 5,\ n = 4,\ v = \{1, 2, 3, 1\},\ c = \{3, 2, 3, 4\},\ w = \{1, 2, 2, 4\}$
- A escolha $I = \{1, 2, 3\}$ e $r = \{3, 1, 2\}$ é válida e ótima, já que:
  - $c_3 = 3 < S$ ✓
  - $c_1 + w_3 = 3 + 2 = 5 = S$ ✓
  - $c_2 + w_1 + w_3 = 2 + 1 + 2 = 5 = S$ ✓
  - $V(I) = 6$ é máximo

Problema da Mochila modificado - Programação Dinâmica?

$$f'(i,s) = \begin{cases} 0 & \text{se } i = n+1 \\ max(f'(i+1,s), v_i + f'(i+1,s-w_i)) & \text{se } \boldsymbol{c_i \leq s} \\ f'(i+1,s) & \text{c.c.} \end{cases} \tag{2}$$

- A recorrência assume ordem fixa!
- Escolhe elementos de maneira sequencial...
- Aplicar recorrência para toda permutação de itens? Podemos fazer melhor!

Ordenação gulosa

- Podemos “desconfiar” de uma ordenação gulosa.
- Ideia 1: Inserir elementos em ordem decrescente de c
    - Falso!
- Ideia 2: Inserir elementos em ordem decrescente de $c - w$
    - Verdadeiro?

Teorema

Theorem

Se $I \subseteq [1, n]$ é um subconjunto ótimo de itens e r é uma permutação de I tal que $c_{r_i} - w_{r_i} \geq c_{r_j} - w_{r_j}$ para todo $i \leq j$, então r é uma permutação válida.

Definiremos uma função $F(p)$ para uma permutação p como o número de pares de índices (i, j), $i \leq j$, tal que $c_{p_i} - w_{p_i} < c_{p_j} - w_{p_j}$.

Por construção, $F(r) = 0$.

Suponha que r não seja uma permutação válida.

Demonstração

Seja r^* uma permutação válida de I com menor valor de $F(r^*) > 0$. Seja $t \in [1, |I| - 1]$ o primeiro índice tal que $c_{r^*_t} - w_{r^*_t} < c_{r^*_{t+1}} - w_{r^*_{t+1}}$.

Seja $S^* = S - \sum_{i=1}^{t-1} w_{r^*_i}$. Para r^* ser uma permutação válida, temos as condições:

$$c_{r^*_t} \leq S^*$$

$$c_{r^*_{t+1}} \leq S^* - w_{r^*_t}$$

Demonstração - cont.

- Ora mas como t é uma "inversão", vale que:

$$c_{r_t^*} - w_{r_t^*} < c_{r_{t+1}^*} - w_{r_{t+1}^*}$$

- Juntando com as equaçãoes anteriores, temos:

$$c_{r_t^*} + w_{r_{t+1}^*} < c_{r_{t+1}^*} + w_{r_t^*} \leq S^*$$

- A sequência $\widetilde{r} = \{r_1, ..., r_{t-1}, r_t, r_{t+1}^*, r_{t+2}^*, ..., r_k^*\}$ é válida e remove pelo menos um par de índices que viola a ordenação gulosa. Assim, temos que $F(\widetilde{r}) < F(r^*)$, uma contradição na hipótese que $F(r^*)$ é mínimo.

Algortimo

- Para todo subconjunto ótimo, podemos reordená-lo de acordo com o critério guloso.
- Agora podemos ordenar para remover a restrição da ordem e aplicar a recorrência $f'(i,s)$ para escolher o subconjunto.

Pseudocodigo

Algoritmo 5.10 Solução para o Problema 5

função MOCHILA$(i, s, v, w, c, n, memo)$
    se $i = n + 1$ **então**
        devolve 0
    se $memo[i][s] = -1$ **então**
        $memo[i][s] \leftarrow Mochila(i + 1, s, v, w, c, n)$
        se $s \geq c[i]$ **então**
            $memo[i][s] \leftarrow max(memo[i][s], v[i] + Mochila(i + 1, s - w[i], v, w, c, n))$
    devolve $memo[i][s]$
função RESOLVE(v, w, c, S, n)
    $Ordene(v, w, c)$ (decrescente em $c - w$)
    $memo \leftarrow \{\{-1\}^S\}^n$
    devolve MOCHILA$(1, S, v, w, c, n, memo)$

- Complexidade de tempo $O(nS + n \lg n)$ e espaço $O(nS)$.
- Possível reduzir o espaço para $O(S)$ numa implementação iterativa.

- Perguntas?
- Mais informações em: https://linux.ime.usp.br/~colombo/mac0499/