O problema da k-mediana

Atílio Gomes Luiz

MO418 - Algoritmos de Aproximação

Instituto de Computação, Universidade Estadual de Campinas

26 de Maio de 2012

Definição do Problema

Problema da k-mediana

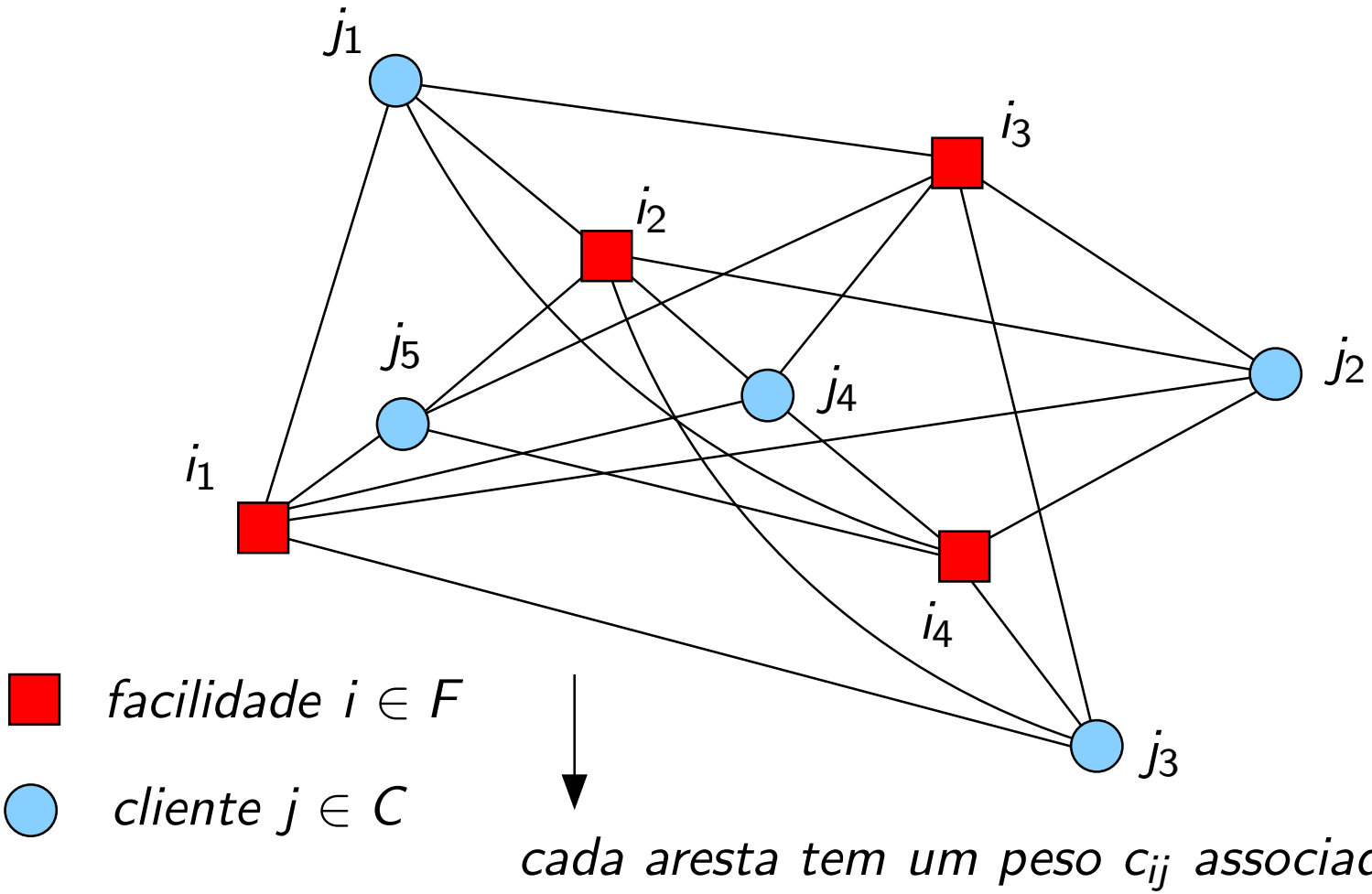
Seja:

- F = conjunto de facilidades
- C = conjunto de clientes
- $k \geq 0$ = quantidade máxima de facilidades a serem abertas.
- c_{ij} = distância entre a facilidade i e o cliente j. (satisfaz desigualdade triangular).

Objetivo: Encontrar subconjunto de facilidades $S \subseteq F$, $|S| \leq k$ e uma função ϕ: $C \to S$ de modo que o custo total de conexão seja minimizado.

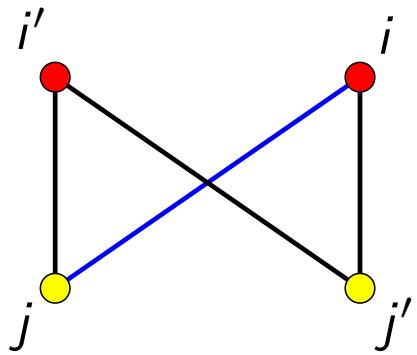
k-mediana - Ilustração

clientes e facilidades no problema da $k-$ *mediana*



k-mediana - Observações sobre o problema

- Utilizaremos o problema da k-**mediana métrica**. (facilidades e clientes são pontos no espaço métrico e os custos satisfazem a desigualdade triangular).



$$c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j}$$

- k-**mediana** é **semelhante** ao problema de **Localização de Facilidades**. No entanto possui duas diferenças: não existem custos para abrir facilidades e existe um limite superior k no número de facilidades que podem ser abertas.

k-mediana - Observações sobre a solução

Queremos usar o esquema primal-dual para desenvolver uma solução para o problema da k-mediana.

- **Entrave:** O esquema primal-dual funciona através de melhoramentos locais e não é adequado para garantir uma restrição global, tal como a restrição de que no máximo k facilidades sejam abertas.
- **Ideia:** Para contornar essa dificuldade, usaremos a técnica de **Relaxação Lagrangeana** para reduzir o problema da k-mediana ao problema de Localização de Facilidades.

Vamos ver uma $2(3+\epsilon)$-aproximação usando técnica primal-dual e relaxação lagrangeana.

Formulação para o problema da k-mediana

Formulação em Programação Inteira

$$
\begin{aligned}
\text{minimizar} \quad & \sum_{i \in F,\ j \in C} c_{ij} x_{ij} \\
\text{sujeito a} \quad & \sum_{i \in F} x_{ij} = 1, && j \in C \\
& y_i - x_{ij} \geq 0, && i \in F,\ j \in C \\
& \sum_{i \in F} y_i \leq k \\
& x_{ij} \in \{0,1\}, && i \in F,\ j \in C \\
& y_i \in \{0,1\}, && i \in F
\end{aligned}
$$

x_{ij} := indica que o cliente j está conectado a facilidade i.
y_i := indica se a facilidade i é aberta ou não.

Formulação para o problema da k-mediana

Relaxação Linear

$$
\begin{aligned}
\text{minimizar} \quad & \sum_{i \in F,\ j \in C} c_{ij} x_{ij} & & \\
\text{sujeito a} \quad & \sum_{i \in F} x_{ij} = 1, & & j \in C \\
& y_i - x_{ij} \geq 0, & & i \in F,\ j \in C \\
& \sum_{i \in F} y_i \leq k & & \\
& x_{ij} \geq 0, & & i \in F,\ j \in C \\
& y_i \geq 0, & & i \in F
\end{aligned}
$$

Técnica de Relaxação Lagrangeana

Relaxação Lagrangeana: consiste em remover algumas das restrições da formulação original, mas tentando embutir essas desigualdades na função objetivo. A ideia é penalizar a função objetivo quando as restrições removidas forem violadas.

Técnica de Relaxação Lagrangeana

Ideia: Retirar a restrição no número máximo de facilidades a serem abertas.

$$
\begin{aligned}
\text{minimizar} \quad & \sum_{i \in F,\ j \in C} c_{ij} x_{ij} \\
\text{sujeito a} \quad & \sum_{i \in F} x_{ij} = 1, && j \in C \\
& y_i - x_{ij} \geq 0, && i \in F,\ j \in C \\
& \sum_{i \in F} y_i \leq k \\
& x_{ij} \geq 0, && i \in F,\ j \in C \\
& y_i \geq 0, && i \in F
\end{aligned}
$$

Como: Aplicando a relaxação lagrangeana.

Novo programa linear relaxado

Desta forma, temos o novo programa linear relaxado:

$$
\begin{aligned}
\text{minimizar} \quad & \sum_{i \in F,\, j \in C} c_{ij} x_{ij} + \sum_{i \in F} \lambda y_i \; - \lambda k \\
\text{sujeito a} \quad & \sum_{i \in F} x_{ij} = 1, && j \in C \\
& y_i - x_{ij} \geq 0, && i \in F,\ j \in C \\
& x_{ij} \geq 0, && i \in F,\ j \in C \\
& y_i \geq 0, && i \in F
\end{aligned}
$$

Observação: Este programa linear nos dá um **limite inferior** no custo de uma solução ótima para o problema da k-mediana.

Formulação do Dual

Formulação do dual do PL anterior

$$
\begin{aligned}
\text{maximizar} \quad & \sum_{j \in C} v_j - \lambda k \\
\text{sujeito a} \quad & \sum_{j \in C} w_{ij} \leq \lambda, \quad \forall i \in F \\
& v_j - w_{ij} \leq c_{ij}, \quad i \in F,\ j \in C \\
& w_{ij} \geq 0, \qquad\quad i \in F, j \in C
\end{aligned}
$$

Observação: Este dual é praticamente o mesmo que o dual para o problema da Localização de Facilidades.

Em direção à solução

- Denotaremos o custo ótimo de uma solução para o problema da k-mediana como OPT_k.
- Sabemos que qualquer solução viável para o problema dual da relaxação lagrangeana fornece um limite inferior para o custo de uma solução ótima do problema da k-mediana. Desta forma, temos que:

$$\sum_{j \in C} v_j - \lambda k \ \leq \ OPT_k \tag{1}$$

Em direção à solução

- Gostaríamos de usar o algoritmo primal-dual do problema de Localização de Facilidades.
- Todos os custos f_i seriam iguais a λ, para algum $\lambda \geq 0$.

Teorema 1 (Jain, Vazirani'99)

O algoritmo primal-dual Facility-JV abre um conjunto de facilidades $S \subseteq F$, atribui $j \in C$ a $i \in S$ e cria uma solução viável (v, w) tal que:

$$\sum_{i \in S,\ j \in C} c_{ij} + 3 \sum_{i \in S} f_i \leq 3 \sum_{j \in C} v_j$$

Em direção à solução

Resultado anterior:

$$\sum_{i \in S,\ j \in C} c_{ij} + 3 \sum_{i \in S} f_i \leq 3 \sum_{j \in C} v_j$$

- Substituindo $f_i = \lambda$ e manipulando a desigualdade, temos que

$$c(S) \leq 3 \left(\sum_{j \in C} v_j - \lambda |S| \right) \tag{2}$$

  onde: $c(S) = \sum_{i \in S,\ j \in C} c_{ij}$.

- O resultado (2) fornece uma ideia para a solução:

Em direção à solução - Ideia 1

- Dada uma instância do problema da k-mediana, a partir dela criaremos uma instância de Facility Location com $f_i = \lambda, \forall i \in F$. Aplicamos o algoritmo Facility-JV nesta instância.
- Do resultado (2), temos que

$$c(S) \leq 3 \left(\sum_{j \in C} v_j - \lambda |S| \right)$$

- Se $|S| = k$, o algoritmo produz uma solução viável para o problema da k-mediana e então, temos:

$$c(S) \leq 3 \left(\sum_{j \in C} v_j - \lambda k \right) \leq 3OPT_k$$

onde a segunda desigualdade é válida pelo resultado (1).

Em direção à solução - Ideia 1

Problema: Infelizmente, se $|S| < k$, então:

$$c(S) \not\leq 3\left(\sum_{j \in C} v_j - \lambda k\right) \leq 3OPT_k$$

Se $|S| > k$ também é ruim, pois o resultado não é uma solução viável.

No entanto, podemos contornar este problema com uma segunda ideia:

Em direção à solução - Ideia 2

Ideia 2: Tentar encontrar valor de λ tal que o algoritmo Facility-JV abre um conjunto de Facilidades S com $|S| = k$. Para isso, usaremos uma **busca binária** em λ.

A ideia é encontrar dois valores λ_1 e λ_2 tais que:

- $|S_1| > k > |S_2|$

Então, usando uma combinação convexa de λ_1 e λ_2, obteremos uma solução aproximada para S e λ.

Para começar a busca precisamos de dois valores iniciais para λ:

- $\lambda_1 = 0 \rightarrow |S| = |F|$: todas as facilidades serão abertas.
- $\lambda_2 = \sum_{i \in F, j \in C} c_{ij} \rightarrow |S| = 1$: somente uma facilidade será aberta.

Algoritmo

k-MEDIAN(F,C,c,k)

1. $\lambda_1 \leftarrow 0$
2. $\lambda_2 \leftarrow \sum_{i \in F,\ j \in C} c_{ij}$
3. **do**
4. $\quad \lambda \leftarrow \frac{1}{2}(\lambda_1 + \lambda_2)$
5. $\quad S \leftarrow$ FACILITY-JV(F, C, λ)
6. $\quad$ **If** $|S| = k$
7. $\quad\quad$ **then return** S $\quad$ *// solução com* $c(S) \leq 3OPT_k$
8. $\quad$ **elseif** $|S| > k$ **then** $\lambda_1 \leftarrow \lambda$; $S_1 \leftarrow S$
9. $\quad$ **elseif** $|S| < k$ **then** $\lambda_2 \leftarrow \lambda$; $S_2 \leftarrow S$
10. **while** $(\lambda_2 - \lambda_1 \leq \epsilon \frac{c_{min}}{|F|})$

11. **If** $(\lambda_2 - \lambda_1 \leq \epsilon \frac{c_{min}}{|F|})$
12. $\quad$ **then** $S \leftarrow$ COMPOSED-SOLUTION$(S_1, S_2, \lambda_1, \lambda_2, k)$

13. **return** S $\quad$ *// solução com* $c(S) \leq 2(3 + \epsilon)OPT_k$

Algoritmo

COMPOSED-SOLUTION$(S_1, S_2, \lambda_1, \lambda_2, k)$

1. $$\alpha_2 \leftarrow \frac{|S_1| - k}{|S_1| - |S_2|}$$
2. **If** $\alpha_2 \geq \frac{1}{2}$
3. **then return** S_2 *// solução com* $c(S_2) \leq 2(3+\epsilon)OPT_k$
4. **else**
5. **For each** facility $i \in S_2$, open the closest facility $h \in S_1$.
6. **If** this doesn't open $|S_2|$ facilities of S_1
7. **then** open arbitrary facilitites in S_1 so that exactly $|S_2|$ are opened.
8. **return** the resulting set of facilities opened.

Algoritmo - Complexidade

Complexidade: A busca binária possui complexidade de tempo polinomial, pois ela faz $O(log \frac{|F| \sum c_{ij}}{\epsilon c_{min}})$ chamadas ao algoritmo *FACILITY-JV*, que também executa em tempo polinomial.

O procedimento COMPOSED-SOLUTION usa duas soluções S_1 e S_2 para obter uma solução S em tempo polinomial tal que $|S| = k$ e $c(S) \leq 2(3 + \epsilon)OPT_k$.

Questão: Como COMPOSED-SOLUTION constrói tal solução?

COMPOSED-SOLUTION

Se ao final da busca binária não tivermos uma solução com exatamente k facilidades, o algoritmo finalizou com soluções S_1 e S_2 e soluções duais correspondentes (v^1, w^1) e (v^2, w^2) tais que $|S_1| > k > |S_2|$ e

$$C(S_x) \leq 3 \left(\sum_{j \in C} v_j^x - \lambda_x |S_x| \right), \ x = 1, 2.$$

Pela condicão de término, também temos $\lambda_2 - \lambda_2 \leq \epsilon c_{min}/|F|$.
S.p.g vamos assumir que $0 < c_{min} < OPT_k$, pois caso contrário teríamos $OPT_k = 0$.

Desta forma podemos encontrar λ_1, λ_2 obedecendo a todas as condições acima em tempo polinomial.

COMPOSED-SOLUTION

Sejam α_1, α_2 tais que $\alpha_1|S_1| + \alpha_2|S_2| = k$, $\alpha_1 + \alpha_2 = 1$, com $\alpha_1, \alpha_2 \geq 0$. Isto implica em

$$\alpha_1 = \frac{k - |S_2|}{|S_1| - |S_2|} \qquad \text{e} \qquad \alpha_2 = \frac{|S_1| - k}{|S_1| - |S_2|}.$$

Logo, podemos obter uma solução viável ($\tilde{v}$, $\tilde{w}$) para o dual do programa linear da k-mediana tal que

$$\tilde{v} = \alpha_1 v^1 + \alpha_2 v^2 \quad \text{e} \quad \tilde{w} = \alpha_1 w^1 + \alpha_2 w^2$$

Temos que ($\tilde{v}$,$\tilde{w}$,λ_2) é uma solução viável para o problema da k-mediana, pois é uma combinação convexa de duas soluções duais viáveis.

Lema 1 - COMPOSED-SOLUTION

O seguinte lema afirma que a combinação convexa dos custos de S_1 e S_2 nos dá um valor próximo de uma solução ótima:

Lema 1

$\alpha_1 c(S_1) + \alpha_2 c(S_2) \leq (3 + \epsilon) OPT_k$

COMPOSED-SOLUTION

O procedimento COMPOSED-SOLUTION se divide em dois casos:

- caso 1: $\alpha_2 \geq \frac{1}{2}$.

Se $\alpha_2 \geq \frac{1}{2}$, nós retornamos S_2 como uma solução. Como $|S_2| < k$, ela é uma solução viável.

Usando o Lema 1 e o fato de que $\alpha_2 \geq \frac{1}{2}$, nós obtemos:

$$c(S_2) \leq 2\alpha_2 c(S_2) \leq 2(\alpha_1 c(S_1) + \alpha_2 c(S_2)) \leq 2(3+\epsilon)OPT_k.$$

como queríamos.

- Agora só nos resta o caso $\alpha_2 < \frac{1}{2}$.

COMPOSED-SOLUTION

- caso 2: $\alpha_2 < \frac{1}{2}$.

1) Para cada facilidade $i \in S_2$, o algoritmo abre a facilidade mais próxima $h \in S_1$.

2) Se isso não abrir exatamente $|S_2|$ facilidades em S_1, o algoritmo escolhe abrir aleatoriamente algumas facilidades de S_1 de modo que no final tenhamos exatamente $|S_2|$ facilidades abertas em S_1.

O algoritmo escolhe um subconjunto de $k - |S_2|$ facilidades das $|S_1| - |S_2|$ facilidades restantes de S_1 e abre essas facilidades.

3) O conjunto de facilidades abertas deste modo é uma solução de custo $\leq 2(3 + \epsilon)OPT_k$.

Lema 2 - COMPOSED-SOLUTION

Seja S o conjunto de facilidades abertas por COMPOSED-SOLUTION no caso em que $\alpha_2 < \frac{1}{2}$. Temos o seguinte lema:

Lema 2

Se $\alpha_2 < \frac{1}{2}$, então a abertura de novas facilidades por COMPOSED-SOLUTION tem custo

$$E[c(S)] \leq 2(3+\epsilon)OPT_k.$$

Este lema conclui o algoritmo. Portanto mostramos uma $2(3+\epsilon)$-aproximação para o problema da k-mediana.

Considerações Finais

- Existem algoritmos melhores para o problema da k- mediana do que o que foi apresentado aqui.
  - Algoritmos de Busca Local e Algoritmos Gulosos
- Seção 9.4 apresenta um algoritmo guloso para o problema da Localização de Facilidades que abre um conjunto de facilidades tal que

$$c(S) + 2\sum_{i \in S} f_i \leq 2\sum_{j \in C} v_j$$

desta forma, podemos usar a mesma lógica utilizada no algoritmo apresentado aqui para obter uma $2(2+\epsilon)$-aproximação para o problema da k-mediana.

Considerações Finais

- É crucial para a análise que tenhamos um algoritmo para o problema da Localização de Facilidades que retorne uma solução S tal que

$$c(S) + \alpha \sum_{i \in S} f_i \leq \alpha \sum_{j \in C} v_j$$

Se tivermos tal algoritmo, podemos configurar $f_i = \lambda$ e obteremos:

$$c(S) \leq \alpha \left(\sum_{j \in C} v_j - \lambda |S| \right)$$

o que nos permite usar a função objetivo do dual da relaxação lagrangeana como um limite inferior.

Considerações Finais

Resultado negativo:

Teorema

Não existe nenhum algoritmo de aproximação para o problema da k-mediana com constante $\alpha < 1 + \frac{2}{e} \approx 1.736$ a menos que todo problema em NP tenha um algoritmo de complexidade $O(n^{O(loglogn)})$.

Referências

David P. Williamson and David B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2010. Pages 184-190.

Vijay V. Vazirani. *Approximation algorithms*, Springer-Verlag, Berlin, 2001. Capítulo 25.