$x_{ij} = \begin{cases} 1 & \text{if } i \to j \\ 0 & \text{o.w.} \end{cases}$

Lagrangian techniques for k-median*

Austin Buchanan

September 9, 2019

cost of assigning $i \to j$

Recall the k-median problem, which can expressed as the following IP.

pick k-medians

assignment constraints

penalize violation

$$
\begin{aligned}
z = \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} & & \text{(1a)}\\
\sum_{j \in V} x_{jj} = k & & \text{(1b)}\\
\sum_{j \in V} x_{ij} = 1 & \quad \forall i \in V & \text{(1c)}\\
x_{ij} \leq x_{jj} & \quad \forall i, j \in V,\ i \neq j & \text{(1d)}\\
x_{ij} \in \{0,1\} & \quad \forall i, j \in V. & \text{(1e)}
\end{aligned}
$$

Letting $n = |V|$, how large is this formulation?

\# vars $= n^2$
\# constraints $= \Theta(n^2)$
\# Nonzeros $= \Theta(n^2)$

What size instance do you think a MIP solver can handle? What about for the LP relaxation?

depends on RAM available
- Hamid has solved LP for $n = 8{,}057$

5 days and 60 GB

$n = 1000$ is plausible for LP & IP

Remark 1. *The integrality constraints $x_{ij} \in \{0,1\}$ for $i \neq j$ can be relaxed to $x_{ij} \geq 0$.*

If one of the constraint sets (1b), (1c), (1d) were removed, is the problem still NP-hard?

*Supporting code available at `https://github.com/AustinLBuchanan/kmedian`

A relaxed problem

To make the problem easier, we can relax the assignment constraints (1c) but penalize their violation in the objective function with multipliers α_i. This yields a Lagrangian relaxation model.

$$\underset{\substack{\uparrow \\ \text{depends on choice of alpha}}}{\mathcal{L}(\alpha)} = \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \sum_{i \in V} \alpha_i \underbrace{\left(1 - \sum_{j \in V} x_{ij}\right)}_{=0 \text{ if satisfied}}$$

$$\sum_{j \in V} x_{jj} = k$$

$$\cancel{\sum_{j \in V} x_{ij} = 1 \qquad \forall i \in V}$$

$$x_{ij} \le x_{jj} \qquad \forall i, j \in V,\ i \ne j$$

$$x_{jj} \in \{0, 1\} \qquad \forall j \in V.$$

Lemma 1. *For every $\alpha \in \mathbb{R}^n$, $\mathcal{L}(\alpha)$ lower bounds the k-median objective z, i.e., $z \ge \mathcal{L}(\alpha)$.*

Pf. Let x^* be optimal for k-median (model (1)). Then x^* is feasible for the Lagrangian relaxation, and

$$z = \sum_{i \in V} \sum_{j \in V} c_{ij} x^*_{ij} = \sum_{i \in V} \sum_{j \in V} c_{ij} x^*_{ij} + \sum_{i \in V} \alpha_i \underbrace{\left(1 - \sum_{j \in V} x^*_{ij}\right)}_{=0} \ge \mathcal{L}(\alpha).$$

def of z

by def, $\mathcal{L}(\alpha)$ is the minimum objective value and x^* is just feasible for Lagrangian.

Corollary 1. *The Lagrangian dual $\mathcal{L} := \sup\{\mathcal{L}(\alpha) \mid \alpha \in \mathbb{R}^n\}$ is a lower bound on z, i.e., $z \ge \mathcal{L}$.*

The max exists, so we can replace sup by max.

Fact 1. *The Lagrangian dual $\mathcal{L}$ is* equal *to the objective value of the LP relaxation of* (1).

Are these bounds $\mathcal{L}(\alpha)$ and $\mathcal{L}$ useful? How would we compute them?

Computing $\mathcal{L}(\alpha)$

First, we can simplify the definition of $\mathcal{L}(\alpha)$ by combining like terms. For this, define:

$$\bar{c}_{ij} := c_{ij} - \alpha_i$$

This substitution allows us to write the Lagrangian's objective function as:

$$\mathcal{L}(\alpha) = \min \sum_i \sum_j \bar{c}_{ij} x_{ij} + \underbrace{\sum_{i \in V} \alpha_i}_{\text{a constant}}$$

Since we have relaxed the assignment constraints, i can be assigned to as many j's as it wants. The only thing stopping it is the coupling constraints (1d). So, let us consider the case where facility j is opened. By the objective function, it would then be optimal to assign i to j if and only if $\bar{c}_{ij} \leq 0$. Thus, if facility j were opened, its contribution to the objective would be:

$$C_j := \bar{c}_{jj} + \sum_{i \in V \setminus \{j\}} \min\{\bar{c}_{ij}, 0\}$$

By this observation, the problem reduces to the following.

Solved by sorting

$$\mathcal{L}(\alpha) = \min \sum_{j \in V} C_j x_{jj} + \sum_{i \in V} \alpha_i$$

$$\text{s.t.} \quad \sum_{j \in V} x_{jj} = k$$

$$x_{jj} \in \{0, 1\} \quad \forall j \in V.$$

Computing $\mathcal{L}$

To solve for $\mathcal{L} := \max\{\mathcal{L}(\alpha) \mid \alpha \in \mathbb{R}^n\}$, use subgradient methods. This is beyond our scope today.

Experiments for relaxations

Table 1: Comparing the lower bounds provided by the LP relaxation and Lagrangian relaxation on random k-median instances. The LP was solved with Gurobi 8.1 (concurrent method). The Lagrangian was solved using Lykhovyd's implementation of Shor's r-algorithm. The rightmost columns give results for Shor's r-algorithm when it is limited to 100 iterations. The lower bounds are rounded to the nearest integer, and times are given in seconds (rounded to two decimal places).

		LP relaxation		Lagrangian		Lagr. (100 iter)	
n	k	LB	time	LB	time	LB	time
100	5	2070	0.23	2070	0.00	2070	0.00
100	10	1356	0.21	1356	0.03	1355	0.01
100	15	1003	0.10	1003	0.01	1003	0.01
100	20	818	0.11	818	0.02	817	0.01
100	25	674	0.10	674	0.01	674	0.01
500	5	54637	18.62	54637	0.10	54637	0.15
500	10	37342	26.52	37342	13.04	37321	0.18
500	15	29220	16.32	29220	0.20	29220	0.16
500	20	24698	19.02	24698	6.84	24665	0.17
500	25	21391	12.14	21391	0.92	21385	0.17
1000	5	223143	205.82	223143	92.71	223062	1.76
1000	10	150530	138.87	150530	4.00	150505	1.66
1000	15	120713	260.64	120711	112.56	120106	1.40
1000	20	101737	203.69	101737	106.07	101387	1.42
1000	25	89511	211.73	89511	49.10	89109	1.30
1500	5	505023	867.59	505021	400.53	504760	5.15
1500	10	344672	336.04	344672	6.02	344671	4.81
1500	15	275959	1061.19	275950	385.50	275257	4.47
1500	20	234558	923.52	234553	381.51	233656	4.45
1500	25	205570	491.39	205570	208.79	205176	4.38
2000	5	887424	892.95	887424	262.48	887285	9.31
2000	10	615275	2110.18	615214	750.18	613121	8.69
2000	15	490876	3036.08	490834	716.11	489938	8.49
2000	20	420594	2567.41	420541	743.20	418948	8.43
2000	25	371435	1957.55	371381	743.38	369271	8.33

would cause Gurobi crash

Some observations:

- When $n = 2,000$, Gurobi used about 3.5 GB RAM. Shor's r-algorithm didn't use this much RAM until $n = 12,000$.
- When $n = 20,000$ Shor's algorithm uses 7.5 GB RAM. When $n = 25,000$, 12 GB RAM.
- The Lagrangian should give the same bound as the LP relaxation if α is truly optimal. However, numerical issues and the termination criteria used by Shor's r-algorithm mean that this is not the case. (I trust that the LP bound given by Gurobi is correct.)
- The (empirical) Lagrangian bound differs from the LP bound by at most 0.015%.
- The Lagrangian bound (100 iterations) differs from the LP bound by at most 0.58%.

Using Lagrangian techniques in branch-and-bound

Computing bounds at B&B nodes. In a B&B algorithm, we need to be able to compute a lower bound when some variables are fixed to zero or one. Denote by F_0 and F_1 as the vertices j for which x_{jj} is fixed to zero and one, respectively. A B&B node can then be identified as (F_0, F_1).

The associated inner problem is as follows.

$$
\begin{aligned}
\mathcal{L}(F_0, F_1, \alpha) = \min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} + \sum_{i \in V} \alpha_i \left(1 - \sum_{j \in V} x_{ij}\right) & \\
\sum_{j \in V} x_{jj} = k & \\
\cancel{\sum_{j \in V} x_{ij} = 1} & \quad \cancel{\forall i \in V} \\
x_{ij} \leq x_{jj} & \quad \forall i, j \in V,\ i \neq j \\
x_{jj} \in \{0, 1\} & \quad \forall j \in V \\
x_{jj} = 0 & \quad \forall j \in F_0 \\
x_{jj} = 1 & \quad \forall j \in F_1.
\end{aligned}
$$

By similar arguments as before, this problem reduces to the following easy problem:

$$
\begin{aligned}
\mathcal{L}(F_0, F_1, \alpha) = \min \sum_{j \in V} C_j x_{jj} + \sum_{i \in V} \alpha_i & \\
\left.\begin{aligned}
\sum_{j \in V} x_{jj} = k & \\
x_{jj} \in \{0, 1\} & \quad \forall j \in V \\
x_{jj} = 0 & \quad \forall j \in F_0 \\
x_{jj} = 1 & \quad \forall j \in F_1.
\end{aligned}\right\} & \text{ basically a sorting problem}
\end{aligned}
$$

The Lagrangian dual at this B&B node (F_0, F_1) is then:

$$\mathcal{L}(F_0, F_1) = \max \{ \mathcal{L}(F_0, F_1, \alpha) \mid \alpha \in \mathbb{R}^n \}.$$

Heuristic. When solving the inner problem $\mathcal{L}(F_0, F_1, \alpha)$, we identified a set of k vertices $S \subset V$: $|F_1|$ that are fixed to one, and $k - |F_1|$ vertices that belong to $V \setminus (F_0 \cup F_1)$ and whose C_j values are minimum. This implies a feasible solution to the k-median problem: open facilities at S and assign vertices from $V \setminus S$ to their nearest open facility, giving a solution with objective:

$$\text{obj}(S) := \sum_{j \in S} c_{jj} + \sum_{i \in V \setminus S} \min\{c_{ij} \mid j \in S\}.$$

It is common to improve the solution S by performing local search.

B&B algorithm. We are now ready to give a Lagrangian-based B&B algorithm.

collection of open B&B nodes

root node

incumbent

1. initialize $\mathcal{B} \leftarrow \{(\emptyset, \emptyset)\}$ and $S^* \leftarrow \emptyset$;

Lagrangian bound

2. while $\mathcal{B} \neq \emptyset$ do
   - pick and remove a node (F_0, F_1) from $\mathcal{B}$;
   - compute $\mathcal{L}(F_0, F_1)$ and let S and α be the associated medians and multipliers;
   - if $\text{obj}(S) < \text{obj}(S^*)$ then
     - $S^* \leftarrow S$;

     update incumbent
   - if $\mathcal{L}(F_0, F_1) < \text{obj}(S^*)$ then else prune
     - pick an unfixed vertex, perhaps $v \in \arg\min\{C_j \mid j \in S \setminus F_1\}$;
     - add $(F_0 \cup \{v\}, F_1)$ and $(F_0, F_1 \cup \{v\})$ to $\mathcal{B}$;
3. return S^*

left branch $x_{vv} = 0$

right branch $x_{vv} = 1$

predicted to improve bound the most in child node

Useful trick: variable fixing. Using Lagrangian info, we can fix some variables x_{jj} to zero or one. For example, consider a vertex j that does not belong to the current Lagrangian solution S at B&B node (F_0, F_1). Every feasible solution x^* with $x^*_{jj} = 1$ has objective at least

$$\underbrace{\mathcal{L}(F_0, F_1)}_{\text{prev bound}} + \underbrace{C_j}_{\text{swap in } j} - \underbrace{\max\{C_v \mid v \in S \setminus F_1\}}_{\text{swap out worst median (unfixed)}}$$

If this value is greater than $\text{obj}(S^*)$, then we can safely fix $x_{jj} = 0$.

takes $\Theta(n)$ time! almost free!

Experiments with B&B

Table 2: Comparing the running times of LP-based B&B (Gurobi 8.1) and Lagrangian-based B&B on random k-median instances. The Lagrangian-based B&B uses a best-bound strategy.

		Gurobi		Lagrangian B&B		
n	k	opt	time	opt	time	Faster
100	5	2070	0.25	2070	0.00	✓
100	10	1357	0.44	1357	0.25	✓
100	15	1004	0.22	1004	0.26	
100	20	820	0.53	820	1.79	
100	25	674	0.14	674	0.45	
500	5	54637	21.02	54637	0.10	✓
500	10	37469	121.61	37469	77.23	✓
500	15	29220	18.13	29220	0.20	✓
500	20	24788	140.04	24788	6184.27	
500	25	21401	26.31	21401	106.20	
1000	5	223424	386.51	223424	92.27	✓
1000	10	150530	144.67	150530	11.74	✓
1000	15	120946	1025.90	120946	9410.60	
1000	20	101974	787.05	101974	13418.80	
1000	25	89604	508.85	89604	60165.90	
10,000	5	MEM CRASH		2.1×10^7	4709.77	✓

takes ~87.5 GB RAM for solving the LP.

Some observations and remarks:

- Gurobi is running on 8 threads simultaneously, while Lagrangian B&B currently runs on 1. Still, Lagrangian B&B seems faster when $k \in \{5, 10\}$.
- Lagrangian B&B can handle much larger instances ($n \approx 20,000$) due to less memory use.
- Lagrangian B&B could be improved with: better heuristics, different Lagrangian termination criteria, adding variable fixing capabilities, better branching choices, better node selection.

Further reading. Research papers using Lagrangian relaxation for k-median include [1, 2, 3, 7, 8]. See also the location textbooks [5] (particularly chapter 6), the IP textbook [4] (particularly section 8.1), and the primer on Lagrangian relaxation by [6].

References

talk based mostly on these

[1] John E Beasley. A note on solving large p-median problems. *European Journal of Operational Research*, 21(2):270–273, 1985.

[2] John E Beasley. Lagrangean heuristics for location problems. *European Journal of Operational Research*, 65(3):383–399, 1993.

[3] Nicos Christofides and John E Beasley. A tree search algorithm for the p-median problem. *European Journal of Operational Research*, 10(2):196–204, 1982.

[4] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*, volume 271. Springer, 2014.

[5] Mark S Daskin. *Network and discrete location: models, algorithms, and applications.* John Wiley & Sons, 2011.

[6] Arthur M Geoffrion. Lagrangian relaxation for integer programming. In *50 Years of Integer Programming 1958-2008*, pages 243–281. Springer, 2010.

[7] Kamal Jain and Vijay V Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001.

[8] Subhash C Narula, Ugonnaya I Ogbu, and Haakon M Samuelsson. An algorithm for the p-median problem. *Operations Research*, pages 709–713, 1977.