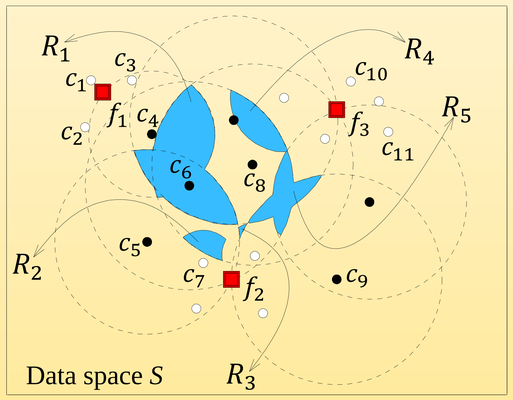
Location Selection for Utility Maximization with Capacity Constraints

Yu Sun, Jin Huang, Yueguo Chen, Rui Zhang, Xiaoyong Du
Renmin University of China, University of Melbourne

Problem Definition

- In a 2-dimensional data space S,
  - let F denote the facility set containing locations of existing facilities f,
  - C denote the client set containing locations of clients c,
  - $v(f)$ denote the capacity of a facility f,
  - and $SE(C,F)$ denote the number of served clients.
- Given the capacity $v(f_n)$ of a new facility f_n, the **location selection query for utility maximization** returns a set M of regions that contains all regions such that if f_n is set up in any of them, the number of newly served clients (utility of the new facility), denoted as $u(R)$ is maximized.
  - $u(R) = SE(C, F \cup \{f_n\}) - SE(C,F)$

Query Example



- In the above example, Rectangles f_1, f_2, f_3 represent the distribution centers.
- $v(f_1) = 3$, $v(f_2) = 4$ and $v(f_3) = 5$, which indicate the number of clients they can serve, respectively.
- Client c_4 is unserved because its nearest distribution center f_1 has a capacity of 3, which is used up by c_1, c_2, c_3, since they are nearer to f_1 than c_4.
- Assume that we want to build f_n with $v(f_n) = 4$.
- Setting it up in any region in the set $\{R_1, R_2, R_3, R_4, R_5\}$ will serve 4 more clients.
  - If f_n is set up in R_1 or R_5, it directly serves 4 unserved clients.
  - If f_n is located in R_2, c_7 is taken over by f_n, then f_2 is able to serve unserved c_9. And f_n directly serves 3 unserved clients.
- This is the highest utility can be achieved. So $\{R_1, R_2, R_3, R_4, R_5\}$ is the answer to this query.
- Considering only unserved clients is not enough in solving our problem.

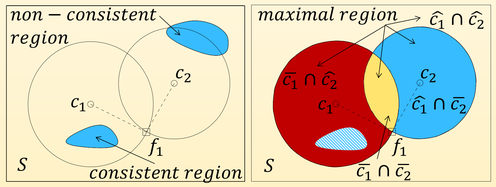
Main Contributions

- A new type of query, named ***location selection query for utility maximization***, is proposed;
- **Arc** based methods are introduced to describe and validate regions;
- An efficient algorithm to answer the query is obtained via applying three pruning rules to a straightforward solution;
- Extensive experiments are conducted. The results confirm the efficiency of the proposed algorithm.

Maximal Regions and BRNN Sets

Given a client c, the nearest facility circle (NFC) of c, denoted as $n(c)$, is a circle that centers at c and has a radius of $dist(c,f)$, where f is the nearest facility of c and $dist(c,f)$ is the Euclidean distance between c and f. We denote the set of all the points within $n(c)$ as $\bar{c}$ and the set of all the points outside as $\hat{c}$.

- Definition:
  - **Consistent region**: a consistent region R is a set of points satisfying that: 1) $\forall\, c \in C, \nexists\, p_i, p_j \in \mathrm{R}$ such that $p_i \in \bar{c}$ and $p_j \in \hat{c}$; 2) $\forall\, c \in C, n(c) \cap R = \emptyset$.



  - **Maximal region**: a consistent region R is a maximal region if $\forall\, R' \cap R \neq \emptyset$, then $R' \subseteq R$.
- Lemma:
  - $\forall\, p_i, p_j$ in a region R, we have $B(p_i) = B(p_j)$.
  - $\forall\, p_i, p_j$ if $B(p_i) = B(p_j)$, then p_i, p_j must be in the same maximal region.

The above two lemmas show the one-to-one relationship between a maximal region and a BRNN set. Formally, given a BRNN set $B(p)$, the corresponding maximal region is:

$$R = \left(\cap_{c_i \in B(p)} \bar{c_i}\right) \cap \left(\cap_{c_j \in \overline{B(p)}} \hat{c_j}\right)$$

So it is possible to get all maximal regions through all combinations of clients.

Validating An Enumeration

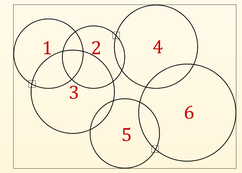
For $|C|$ clients, there are totally $2^{|c|}$ possible BRNN sets. Some may have no corresponding real maximal regions. Thus we adopt an incremental way of checking the validation of an enumerated BRNN set:

- A temporary maximal region R is incrementally maintained.

- When a new client c is involved for checking, R is updated according to the 12 cases of the location relationship between $n(c)$ and the enclosing arcs of R.
- If it is $\bar{c}$, then we keep the inside part of the arc, else the outside part is kept.
- New enclosing arcs are added by similarly checking their middle points.
- Checking one client after another, if R becomes $\emptyset$, then the enumeration is invalid. Otherwise, R is the corresponding maximal region of the enumerated set
- In the example, we need validate $\bar{c_1}\hat{c_2}\hat{c_3}$. Now, the temoprary maximal region $R = \bar{c_1}\hat{c_2}$, and it is enclosed by arc a_1 and a_2. When we check c_3, a_1 and a_2 are both updated in case 12, leading to 4 smaller arcs left. Two new arcs a_3 and a_4 from $n(c_3)$ are added. Then the two smaller blue regions form the maximal region corresponding to $\bar{c_1}\hat{c_2}\hat{c_3}$.

Search Algorithms

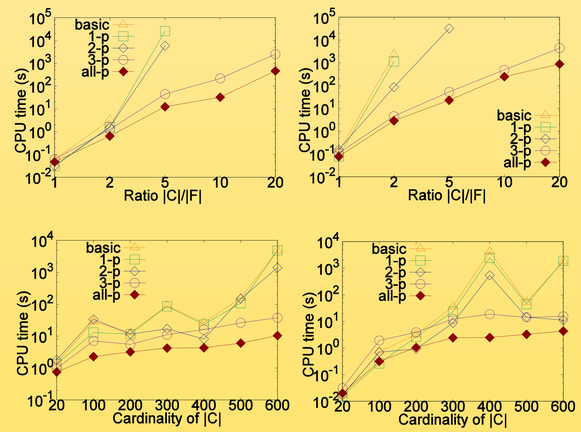
- Baseline Algorithm:
  - A naive solution is to enumerate all combinations of all clients, validate each enumeration, get the increment of the BRNN set and calculate the utility.
  - We only need consider those NFCs which it intersects with. In order to avoid repetition we assign each NFC an unique ID so they can be scanned in order.



  - The above example has six NFCs. We start from the one with the smallest ID 1. Since 1 only intersects with 2 and 3. Then, all BRNN sets we enumerate for 1 are $123\hat{4}\hat{5}\hat{6}$, $12\hat{3}\hat{4}\bar{5}\hat{6}$, $1\hat{2}\hat{3}\hat{4}\bar{5}\hat{6}$ and $1\hat{2}\hat{3}\hat{4}\bar{5}\bar{6}$.
  - Then, we turn to check 2, it intersects with 1,3,4, since 1 is smaller than 2, thus we would only enumerate $\hat{1}23\hat{4}\bar{5}\hat{6}$, $\hat{1}\hat{2}\hat{3}\hat{4}\bar{5}\hat{6}$, $\hat{1}\hat{2}\hat{3}\bar{4}\bar{5}\hat{6}$ and $\hat{1}\hat{2}\hat{3}\hat{4}\bar{5}\bar{6}$. In this way, the maximal region that lies in both 1 and 2 will not be repeated.
- We propose three pruning rules to speed up the search process and get the prune algorithm.
  - (I) Expected Max Benefit Pruning
  - (II) Max Workload Sharing Pruning
  - (III) Early Abandoning

Experiments

- We study the performance of the baseline (basic), baseline with first (1-p), second (2-p), third (3-p) pruning rule and the pruning algorithm (all-p) which features all three pruning rules.
- Both real and synthetic data sets are used.



- For details, please refer to our paper in the proceedings.