Online Facility Location

and its versatile applications

Harry Lang

Geometrica, INRIA Saclay

October 21, 2015

Outline

1. Facility Location
   - Offline Version
   - Online Version
2. k-Median Clustering
   - Offline Version
   - Streaming Version
3. More Accurate Techniques

Facility Location

Metric Space : $(\mathcal{X}, d)$
Facility cost : f

Offline Problem : given a set of points, select a set to open as facilities.
For each facility, pay f in cost.
For each non-facility x, pay $d(x, c)$ where c is the nearest facility.

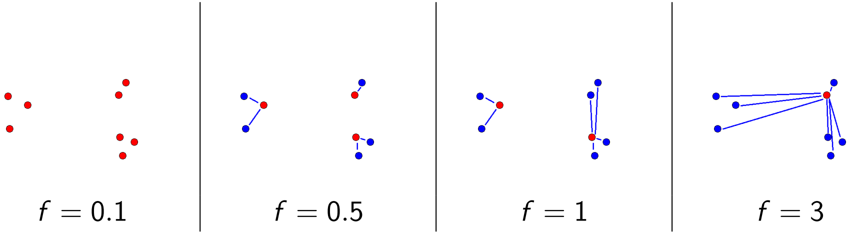
Goal : Find a facility set that minimizes cost.

Facility Location

What does an optimum solution look like? It depends on f.

For small enough f, optimal solutions have a facility at every location.

For large enough f, optimal solutions have only one facility.

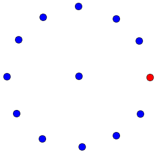


$f = 0.1$ | $f = 0.5$ | $f = 1$ | $f = 3$

Online Facility Location

Online Problem : receive points sequentially. Before reading the next point, decide whether to open as a facility or connect to an existing facility.

Since we have to make the decisions online, the optimal offline solution may not be possible. For example:



At the end our connections may not be to the nearest facility. Also, we cannot decide later to open a connected point as a facility.

Online Facility Location

Let Φ denote the current set of facilities. Let $d(x, \Phi)$ denote the minimum of $d(x, y)$ for all $y \in \Phi$.

Algorithm of Meyerson (2001)

For each point x

  With probability $d(x, \Phi)/f$, open x as a facility

  Otherwise, connect x to the nearest facility

The expected[1] cost of Meyerson's algorithm is less than $8 \cdot OPT$ (where OPT is the optimal solution for the offline problem).

[1] over randomness in the algorithm and ordering of the data

Online Facility Location

This simple algorithm provides a rich structure that can be a building block for many different problems.

Online Facility Location

This simple algorithm provides a rich structure that can be a building block for many different problems.

1. f gives a deterministic max-distance between points and facilities.

Online Facility Location

This simple algorithm provides a rich structure that can be a building block for many different problems.

1. f gives a deterministic max-distance between points and facilities.
2. satellite data can be stored with the facilities (i.e. the number of connected points; the distance the points have moved; etc).

Online Facility Location

This simple algorithm provides a rich structure that can be a building block for many different problems.

1. f gives a deterministic max-distance between points and facilities.
2. satellite data can be stored with the facilities (i.e. the number of connected points; the distance the points have moved; etc).
3. the algorithm generalizes simply to hold for a wide variety of related problems (approximate triangle inequality, weighted points, etc).

Online Facility Location

This simple algorithm provides a rich structure that can be a building block for many different problems.

1. f gives a deterministic max-distance between points and facilities.
2. satellite data can be stored with the facilities (i.e. the number of connected points; the distance the points have moved; etc).
3. the algorithm generalizes simply to hold for a wide variety of related problems (approximate triangle inequality, weighted points, etc).

Our example application: Streaming k-Median Clustering

Offline k-Median

Given a set P of n points, select a subset $C \subset P$ of size k to designate as centers.

$Cost(P, C) := \sum_{x \in P} d(x, C)$

Goal : Find C of size k to minimize $Cost(P, C)$

This problem is Max-SNP hard, but offline 2.66-approximations exist (Byrka 2015). All streaming solutions work by maintaining a summary, and then running an offline algorithm on the summary.

Streaming k-Median

Insertion-Only Streams : receive the data points sequentially, and after each point maintain an approximate solution for the stream so far.

Quantities of Interest :

Space : try to keep in $O(k \log n)$ space

Update Time : time required to process a single point

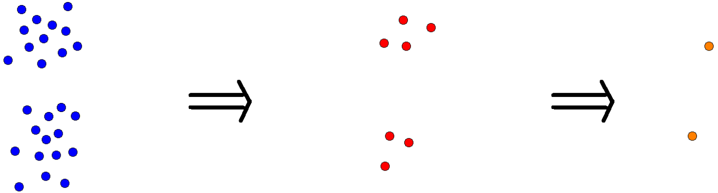
Streaming k-Median

What if we just run Online Facility Location by somehow choosing f so that we end up with $O(k)$ facilities?

Streaming k-Median

What if we just run Online Facility Location by somehow choosing f so that we end up with $O(k)$ facilities?

That doesn't work, but in fact we can choose f so that we maintain O(k log n) facilities. Then $|Cost(P, C) - Cost(\Phi, C)| \leq Connect(P, \Phi)$ for any set C, so we can cluster Φ to approximate a clustering for P.



Original Stream
n points

Summary
$O(k \log n)$ facilities

Output
k centers

Step 1 : Assume OPT is known

The connection between OFL and Streaming Clustering was first observed by Charikar et al (2003) and then improved by Braverman et al (2011).

OPT refers to the optimal cost of a k-median clustering on P.

Theorem of Braverman/Ostrovsky (2011)

Let $f \sim OPT/k \log n$. With probability $1 - \frac{1}{n}$, running OFL on P results in $O(k \log n)$ facilities with connection cost at most $4 \cdot OPT$.

The key is selecting f to be large-enough to upper-bound the facilities and low-enough to upper-bound the connections.

Step 2 : Full Solution

OPT is unknown, but it is monotonically increasing and we know it approximately. Instead of keeping f constant, also increase it as the stream progresses.

The result : We maintain a facility set Φ of size O(k log n) such that $Connect(P, \Phi) \leq (3 + \epsilon)OPT$. We then run an offline approximation on Φ.

Theorem

Let Q be such that $Connect(P, Q) \leq \alpha OPT$. Then any γ-approximation for Q is a $\alpha + 2\gamma(1 + \alpha)$-approximation for P.

More Accurate Techniques

Coreset (Definition)

A (k, ϵ)-coreset for a set P is a set Z such that for every set $C \in P^k$,

$$|Cost(P, C) - Cost(Z, C)| \leq \epsilon Cost(P, C)$$

In Euclidean space, coresets are well-studied, and they exist in size $\tilde{O}(\epsilon^{-2}k)$ even on streams. For general metric spaces, offline constructions exist in size $O(\epsilon^{-2}k \log n)$ (Feldman et al 2014).