Applying facility location to clustering a large dataset

Moses Charikar
Princeton University

Joint work with
Liadan O’Callaghan
and
Rina Panigrahy
Sources of Massive Data Sets

- World Wide Web
- Traffic on the internet
- Telephone records
- Multimedia data
- Customer transactions
- Astronomical data
New limitations and paradigms

- Data too large to fit in main memory
- Linear or near linear time algorithms
- Random access to data is infeasible

**Sketching model**
- Process compact sketches instead of original data

**Streaming model**
- One or more passes over data using small storage space
Streaming model

- Algorithm must process data by making one or more passes over it
- Size of data is massive compared to memory size
- Random access not feasible
- What problems can be solved?
- Can we get approximate answers to interesting questions?
Clustering

- **Given**: very large collection of objects
  - Objects could be web pages, news stories, images, customer profiles, etc

- **Objective**: cluster the objects
  - Disjoint partition into clusters
  - Similar/related objects in the same cluster
  - Dissimilar objects in different clusters
Clustering objective functions

- Typically, associate each cluster with cluster center (representative)
- **Goal**: partition into $k$ clusters
- Equivalently, find $k$ centers and assign points to centers
- Clustering is good if points are close to cluster centers
- Common clustering objectives measure distances of points to cluster centers
Clustering objective functions

- **Maximum cluster radius** (*k-center*)
- **Sum of distances of points to cluster centers** (*k-median*)
- **Sum of cluster radii** (*k-sumradii*)
Offline vs. Streaming

- **Offline model:**
  - Find good clustering solution in polynomial time
  - Arbitrary access to data

- **Streaming model:**
  - Produce implicit description of clusters (i.e. cluster centers + additional info) in one pass, using small amount of space.
Input representation

- Measure space requirement in terms of number of objects stored
- What if objects themselves are large?
  - Schemes to represent objects compactly
  - Distance of objects can be estimated from their compact representations
Talk outline

- Streaming algorithms for clustering
- K-center
- K-median
- Clustering formulations with outliers
**K-center**

- Given collection of points
- Pick $k$ cluster centers
- Assign each point to closest center
- Minimize maximum point-center distance

**Offline:** 2-approximation

[Hochbaum, Shmoys] [Dyer, Frieze] [Gonzalez]
Offline algorithm

- Suppose optimal radius is $OPT$
- Process points sequentially
- Maintain set of centers $S$
  (Initially $S = \{\text{first point}\}$)
- Consider next point $p$
  - If $p$ is within distance $2OPT$ of some center in $S$, add to corresponding cluster
  - Else, add $p$ as new center in $S$
Analysis

Assuming we know $OPT$

**Guarantee on solution cost**

- Radius of each cluster is at most $2OPT$

**Guarantee on number of centers**

- Distance between points in $S$ is $>2OPT$
- Every point in $S$ must be in a distinct cluster in optimal solution
- $S$ can have at most $k$ points
Streaming algorithm

- Start with very low guess on OPT
- Run \textit{offline} algorithm
- If we get \( k \) centers, guess was too low
- Increase guess, merge clusters

- Algorithm runs in phases
- \( r_i : \) guess used in phase \( i \)
- \( r_{i+1} = 2 \, r_i \)
Phase transitions

- **End of phase** $i$
  - $k+1$ points with pairwise distance $> 2r_i$
  - Each cluster of radius $< 4r_i$
- **Beginning of phase** $i+1$
  - $r_{i+1} = 2r_i$
  - Pick arbitrary center $c$, merge clusters whose center within $2r_{i+1}$ from $c$ (repeat)
- **New point** $p$
  - Add to cluster if within $2r_{i+1}$ from center
  - Else, add $p$ to set of centers (create new cluster)
Radius of new clusters $\leq 2r_{i+1} + 4r_i = 4r_{i+1}$
Approximation guarantee

- Clusters in phase $i+1$ have radius $< 4r_{i+1}$
- $\text{OPT} > r_i$
- Approximation ratio $= 4r_{i+1}/r_i = 8$
- Note: storage required is $k$

- Ratio can be improved
  - More sophisticated algorithm
  - Randomization

- [C, Chekuri, Feder, Motwani]
Given collection of points
Pick \( k \) cluster centers
Assign each point to closest center
Minimize sum of point-center distances

Offline: \( 3+\varepsilon \) approximation \([\text{Arya, etal}]\)
LP rounding, primal dual, local search
Previous streaming algorithm

- [Guha, Mishra, Motwani, O’Callaghan]
- Storage: \( n^\epsilon \), approximation ratio \( 2^{O(1/\epsilon)} \)
- Apply offline algorithm to cluster blocks of \( n^\epsilon \) points
- Clustering proceeds in levels
- Centers for level \( i \) form input for level \( i+1 \)
New approach

- [C, O’Callaghan, Panigrahy]
- Idea: mimic k-center approach
- Suppose we knew OPT
- Can we maintain solution with k centers and cost $O(OPT)$ in streaming fashion?
Facility location

- Given collection of points, facility cost $f$
- Find subset $S$ of centers
- Assign each point to closest center
- Cost = sum of point-cluster distances $+ f |S|$

- Contrast with $k$-median
- (sort of) Lagrangian relaxation
Using facility location for k-median

- Given $k$-median instance with optimal value $OPT$
  - Produce facility location instance by setting facility cost $f = OPT/k$
  - Optimal for facility location $\leq 2 \cdot OPT$
- Given $\beta$ approx algorithm for fac locn
  - Fac locn solution of cost $\leq 2\beta \cdot OPT$
- Interpret as $k$-median solution
  - Cost $\leq 2\beta \cdot OPT$, #centers $\leq 2\beta \cdot k$
Online algorithm for facility location

[Meyerson]

- $f$ = facility cost
- For each point $p$
- $\delta = \text{distance of } p \text{ to closest center}$
- Open center at $p$ with probability $\delta/f$

**Theorem:** Expected cost of solution $= O(\log n) \text{ OPT}$
Using the online algorithm

- Suppose we have lower bound \( L \) on \( \text{OPT} \)
- We set \( f = \frac{L}{k(1+\log n)} \)
- Run online facility location algorithm (Online-Fac-Locn)

**Lemma:**
- Expected number of centers produced \( \leq k(1+\log n)(1+4\text{OPT}/L) \)
- Expected cost \( \leq L+4\text{OPT} \)
- Procedure to check if \( \text{OPT} \) much larger than \( L \)
Updating the lower bound

- With probability at least $\frac{1}{2}$, Online-Fac-Locn produces solution with
  - Cost $\leq 4(L+4OPT)$
  - #centers $\leq 4k(1+\log n)(1+4OPT/L)$

- Run $O(\log n)$ invocations of this in parallel
- Invocation fails if cost exceeds bound, or number of centers exceed bound $O(k \log n)$
- If all invocations fail, update lower bound $L$
Changing phases

- Increase lower bound to $\beta \cdot L$
- Pick solution produced by invocation that finished last
- Feed (weighted) centers as input to next phase
- Finally, $O(k \log n)$ centers with cost $O(OPT)$
- Run offline algorithm on weighted centers to get $k$ centers with cost $O(OPT)$
- **Note:** storage = $O(k \log^2 n)$ points
Many little Details

- Algorithm succeeds with high probability
  - When a phase ends, $OPT > \beta \cdot L$ w.h.p
  - During a phase, solution cost $< \gamma \cdot L$ w.h.p.
  - $\beta$ and $\gamma$ chosen appropriately to maintain invariants
  - avoid multiplicative increase in approx ratio

- At phase change, need good lower bound on $OPT$
  - solve offline $k$-median on weighted medians and one new point.
Clustering with outliers

- Can exclude $\epsilon$ fraction of the points
- Find solution to optimize clustering objective on remaining $(1- \epsilon)$ fraction of point set

Offline: [C, Khuller, Mount, Narasimhan]
Streaming: [C, O’Callaghan, Panigrahy]
Outliers analysis ideas

- **Algorithm**: Sample data set and apply offline clustering algorithm to sample

- **Analysis**: show that sample is representative of data set, i.e.
  - If particular solution excludes $\epsilon$ fraction of points in the sample
  - Solution scaled up to entire data set does not exclude much more than $\epsilon$ fraction of points