k-Center Problems

Joey Durham

Graphs, Combinatorics and Convex Optimization Reading Group
Summer 2008
Outline

- General problem definition
- Several specific examples
  - k-Center, k-Means, k-Mediod
- Approximation methods
- Other methods
  - Lloyd algorithm
  - Annealing
- Summary of properties
General k-Center Problem

- Given:
  - $n$ in points in a vector space or a complete graph
  - Distance function satisfying the triangle inequality
- Find $k$ “centroids” to minimize some measure of cluster size
- NP-hard

Image from www.graph-magics.com
Applications

- Data clustering
- Statistical analysis
- Deployment
- Task allocation
- Image classification
- Facility location

Image from www.spatialanalysisonline.com
Variations on k-Center

- **Centroids**
  - Member of data set
  - Any point in vector space
- **Cluster measures**
  - Maximum distance => minimize worst case
  - Sum of distances => minimize expected distance
  - Sum of square distances => minimize variance
- **Vertex weights**
- **Added centroid cost**
  - Facility location problem
k-Means Clustering

- Vector space, Euclidean distances
- Minimize intra-cluster variance
- Centroids NOT in data set
  - k-medoids: centroids in set
- The most famous: 21,000+ hits on Google Scholar
- Often used in data clustering/statistics

Resources:
- MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations"
- [http://www.autonlab.org/tutorials/kmeans.html](http://www.autonlab.org/tutorials/kmeans.html)
Standard k-Center

- Complete graph, edge costs satisfy tri. ineq.
- Minimize worst case distance of vertex to centroid
- Centroid in data set
  - Resources: Vazirani (2003), *Approximation Algorithms*
2-Approximation Algorithm

1) Order all edges $e_i$ by cost
2) Construct graphs $G_i$ containing all edges up to $e_i$
3) Construct square graphs $G_i^2$
4) Compute maximal independent set $M_i$ of $G_i^2$
5) Find smallest $i$ s.t. $|M_i| \leq k$, say $j$
6) Return $M_j$

- Best possible polynomial time approximation: 2
- At least $O(n^3)$
- Resources: Vazirani (2003), Approximation Algorithms
2-Approximation Algorithm

1) Order all edges $e_i$ by cost

2) Construct graphs $G_i$ containing all edges up to $e_i$

3) Construct square graphs $G_i^2$

4) Compute maximal independent set $M_i$ of $G_i^2$

5) Find smallest $i$ s.t. $|M_i| \leq k$, say $j$

6) Return $M_j$

- Square graph contains a one-hop connection wherever base graph had a one- or two-hop connection
2-Approximation Algorithm

1) Order all edges $e_i$ by cost

2) Construct graphs $G_i$ containing all edges up to $e_i$

3) Construct square graphs $G_i^2$

4) Compute maximal independent set $M_i$ of $G_i^2$

5) Find smallest $i$ s.t. $|M_i| \leq k$, say $j$

6) Return $M_j$

- Maximal independent set
  - A set $S$ such that every edge of the graph has at least one endpoint not in $S$ and every vertex not in $S$ has at least one neighbor in $S$
  - aka independent dominating set

Image from en.wikipedia.org
Lloyd algorithm

1) Pick initial centroids
2) Given centroids, compute clusters
3) Given clusters, compute new centroids
4) Repeat 2 & 3 until “convergence” (centroids don’t move very much)

- Most commonly used heuristic solver
  - Nearly synonymous with k-means
  - aka Voronoi iteration
  - Over 2,500 hits on G scholar
- Converges quickly to a good approximation in practice
  - Num iterations often $<< n$
- Many applications
- Poor theoretical bounds
Lloyd algorithm

1) Pick initial centroids
2) Given centroids, compute clusters
3) Given clusters, compute new centroids
4) Repeat 2 & 3 until “convergence” (centroids don’t move very much)

- **Bad bounds**
  - Time: super-polynomial in $n$
  - Approximation: can get stuck in local minimum

- **“Seeding” initial centroids very important**
  - Many complex methods for picking initial centroids

- **Resources:**
  - Arthur & Vassilvitskii (2006), "How Slow is the k-means Method?"
  - Arthur & Vassilvitskii (2007), "k-means++ The Advantages of Careful Seeding"
Simulated Annealing

- Lloyd algorithm with added randomness
  - “Temperature” $T$ controls level of randomness
  - At high temperature, bypasses local minima
- $T$ is decreased on a schedule
  - Schedule affects result
  - Ideal cooling rate cannot be pre-computed
- Resources:
Deterministic Annealing

- Not stochastic!
  - Fractional ownership of vertices based on “temperature” $T$
- $T$ controls centroid greed
  - At $T = \infty$, every centroid claims every vertex equally
  - At $T = 0$, like Lloyd
- Resources:

High T solution

Low T solution
Deterministic Annealing

- Like S.A., at high $T$ D.A. bypasses local minima
  - Without randomness
- Still requires a temperature schedule
  - Again, determining an ideal schedule is complex
  - Depends on topography
## Summary: k-Center Variations

<table>
<thead>
<tr>
<th></th>
<th>k-center</th>
<th>k-means</th>
<th>k-medoids</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Datapoints in:</strong></td>
<td>Graph</td>
<td>Cont. space</td>
<td>Cont. space</td>
</tr>
<tr>
<td><strong>Centroids</strong></td>
<td>In set</td>
<td>Not in set</td>
<td>In set</td>
</tr>
<tr>
<td><strong>Distance norms</strong></td>
<td>Max or 1</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
## Summary: Solvers

<table>
<thead>
<tr>
<th></th>
<th>Approx. alg.</th>
<th>Lloyd alg.</th>
<th>Simulated Annealing</th>
<th>Deterministic Annealing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approx. factor</td>
<td>2</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Running time</td>
<td>Long</td>
<td>Short to very long</td>
<td>(# iter)*(lloyd)</td>
<td>(# iter)*(lloyd)</td>
</tr>
<tr>
<td>Stuck in local min</td>
<td>NA</td>
<td>Yes</td>
<td>No with good T schedule</td>
<td>No with good T schedule</td>
</tr>
<tr>
<td>Seeding importance</td>
<td>NA</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>