Conquering the Divide: Continuous Clustering of Distributed Data Streams

Graham Cormode
graham@research.att.com
S. Muthukrishnan
muthu@cs.rutgers.edu
Wei Zhuang
weiz@cs.rutgers.edu
Outline

• Motivation in Data Streams
• Continuous Distributed Model
• Our CD Clustering Algorithms
• Experiments
• Extensions and conclusion
Data Streams

• Modern data acquisition generates large distributed data coming as distributed streams
  – Environmental sensors
  – Web/blog crawlers
  – Soldier locations on battlefield
  – Different mail servers collate their mails over time
  – Repositories on book, audio, video evolve on their own

• Need support not provided in traditional DBMS
Monitoring Challenges

• Need real-time monitoring of data properties
  – To help identify any problem of the data (collection): network attack detection, performance debugging
  – To help understand the nature of the data distribution: classification, distinct count, etc.

• Clustering is a foundational monitoring problem
  – Web crawlers cluster pages into groups to monitor traffic load
  – Underwater fish tracking monitors the clusters of schools to quickly deploy attracting or dispelling devices
Outline

• Motivation
• Continuous Distributed Model
• Our CD Clustering Algorithms
• Experiments
• Extensions and conclusion
Continuous Distributed (CD) Model

Coordinator site

Union of streams

$S^t = S_1^t U \ldots U S_m^t$

Continuous Query $Q(S^t)$

Immediate answer to $Q(S^t)$ with accuracy guarantee

Save Communication

Update message

Sharing message

Site 1

$S_1^t$

Update stream

Site m

$S_m^t$

Update stream
Outline

• Motivation
• Continuous Distributed Model
• Our CD Clustering Algorithms
  – Local algorithms
  – Global algorithms
• Experiments
• Extensions and conclusion
Our Contribution

• We provide several k-center clustering algorithms that
  – Monitor distributed dynamic streams
  – Answer continuous query
  – Achieve constant factor approximation as good as centralized algorithm
  – Minimize the communication between sites
K-center clustering

Furthest Point

Parallel Guessing

Local - FP

Local - PG

Global - FP

Global - PG

our algorithms
K-center clustering

• Given
  – a set $P$ of $n$ points
  – a distance function $d(p,q)$ between pairs of points
• Find a set $C \subseteq P$ of $k<n$ points as centers
• Minimize the radius $r = \max_{p \in P} d(p, C(p))$
  – $C(p) = \arg\min_{c \in C} d(c,p)$, distance of $p$ to closest center
“Furthest Point” (FP) Algorithm

• Furthest Point (FP) (Gonzalez ’85)
  – Pick an arbitrary point as the first center, $C_1 = \{c_1\}$
  – Given $i$ centers in $C_i$, find the point $p$ to maximize $d(p, C_i(p))$, set $C_{i+1} = C_i \cup \{p\}$.
  – Stop after $k$ iterations with $k$ centers $C_k$.

• Analysis:
  – Factor 2 approximation to the optimal clustering
  – $k$ passes, $O(kn)$ distance computations
An example

- A 4-center clustering using FP algorithm
“Parallel Guessing” (PG) Algorithm

• Assume $R$ is a lower bound on the optimal radius
• Pick an arbitrary first center, $C = \{c_1\}$
• For each point $p$, if $d(p, C(p)) > R$, then $C = C \cup \{p\}$
• In parallel, guess $R = (1+\varepsilon/2), (1+\varepsilon/2)^2, (1+\varepsilon/2)^3, \ldots$
• Use result for smallest $R$ that generates $k$ or fewer centers
Analysis of PG Algorithm

- Makes one pass over the input
- Gives \((2 + \varepsilon)\) approximation
- If the ratio \(\max_{p,q} d(p,q) / \min_{p,q} d(p,q) = \Delta\), the number of parallel guesses is \(O(1/\varepsilon \log \Delta)\)
- Stores \(O(k/\varepsilon \log \Delta)\) points, and each update needs \(O(k/\varepsilon \log \Delta)\) distance computations
An example

- A 4-center clustering using PG algorithm
Outline

• Motivation
• Continuous Distributed Model
• Our CD Clustering Algorithms
  – Local algorithms
  – Global algorithms
• Experiments
• Extensions and conclusion
Merging Local Clusterings

- **Local-FP**: each site runs FP on its points, send k-centers to coordinator on recluster.

- **Local-PG**: each site runs PG on its points, shares centers for current good guess R with coordinator.

- Both have variants which share more or less data:
  - Coordinator does not share cluster information
  - Coordinator broadcasts the global cluster radius
  - Coordinator lazily shares radius (when contacted by site)

- All versions guarantee a \((4+\varepsilon)\) approximation
Space and Communication Cost — Local Algorithms

- **Local-FP**: each local reclustering sends $O(k)$ points
  - No tight bounds on worst case number of reclusterings.

- **Local-PG**: one pass over the input, each site sends up to $k$ points for each guess.
  - Total communication is at most $O(km/\varepsilon \log \Delta)$.
  - Space required at each site is $O(k/\varepsilon \log \Delta)$.
Cluster Merging

- Clusters can be merged by clustering the cluster centers
- Given point set $P_i$, $i=1,\ldots,m$, let $C_i \subseteq P_i$ be the $k$ centers of $\alpha$-approximate $k$-center clustering of $P_i$
- Run a $\beta$-approximate $k$-center clustering on union of $C_i$'s
- **Result**: the resulting $k$ centers $C$ form an $(\alpha+\beta)$ clustering of the point set $P = P_1 \cup P_2 \cup \ldots \cup P_m$
Outline

• Motivation
• Continuous Distributed Model
• Our CD Clustering Algorithms
  – Local algorithms
  – Global algorithms
• Experiments
• Extensions and conclusion
Global-FP Clustering

• Distributed collaboration to find k global centers using FP:
  – In round i, each site sends its furthest point from current centers $C_i$
  – Coordinator picks furthest point as a center and sends it to all sites
• Each remote client monitors its local points, recluster if some point is more than $(1 + \varepsilon/2)R$ from the global centers
• Alternate collaborating and monitoring phrases
• Each reclustering sends $O(mk)$ points
Global-PG Clustering

- Coordinator maintains a global PG clustering, shares the current good guess and centers with all clients
  - Client sends new point $p$ if $\exists i, d(p, C_i(p)) > R_i$
  - Coordinator shares new centers with all clients
- Total communication bounded by $O(km/\varepsilon \log \Delta)$
- Coordinator always has a $(2+\varepsilon)$-approximation
Comparison

- A summary of the algorithms:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Guarantee</th>
<th>Space</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local-FP</td>
<td>$4 + \varepsilon$</td>
<td>$O(n)$</td>
<td>—</td>
</tr>
<tr>
<td>Local-PG</td>
<td>$4 + \varepsilon$</td>
<td>$O(k/\varepsilon \log \Delta)$</td>
<td>$O(km/\varepsilon \log \Delta)$</td>
</tr>
<tr>
<td>Global-FP</td>
<td>$2 + \varepsilon$</td>
<td>$O(n)$</td>
<td>—</td>
</tr>
<tr>
<td>Global-PG</td>
<td>$2 + \varepsilon$</td>
<td>$O(k/\varepsilon \log \Delta)$</td>
<td>$O(km/\varepsilon \log \Delta)$</td>
</tr>
</tbody>
</table>
Outline

• Motivation
• Continuous Distributed Model
• Our CD Clustering Algorithms
• Experiments
• Extensions and conclusion
Experimental Data Sets

• **US Census 2005 (TIGER) data**
  - \( N = 600k \) 2-D points
  - State of New Jersey, \( m = 21 \)

• **Stock price series**
  - Price series of a stock with \( n = 330k \) values
  - Simulate 100-D vectors

• **Measure communication cost as ratio compared to sending complete data**
Experimental Results — Communication cost

- Fix sites $m$ and clusters $k$, vary $\varepsilon$: compare communication

- The local algorithms are more cost effective than naive global implementations of distributed algorithms
Experimental Results — Cluster Quality

- Fix $m$, $k$, vary $\varepsilon$, compare radius: smaller is better
- Not much to choose between algorithms in terms of accuracy
- Choice of $\varepsilon$ seems not very sensitive – can choose a large value and not suffer
Experimental Results — Other Dependence

- Varying number of clusters $k$ and sites $m$

- Analysis of cost seems a good predictor: all algorithms very close to linear on $m$, quadratic on $k$ for FP algorithms
Outline

• Motivation
• Continuous Distributed Model
• Related Work
• Centralized K-center Clustering Algorithms
• Our CD Clustering Algorithms
• Experiments
• Extensions and conclusion
Extensions

- **Deletions**: run same algorithms as before. Monitor the \((k+1)\) points which are ‘proof’ of current clustering and recluster if any get deleted.
- **Moving Points**: can treat as deletion of old point, reinsertion at new location if possible.
- **Sliding window**: treat the point that is no longer in the window as a deletion of point.
Variable number of clusters

- Although we built a k-center clustering, we can get any $k'$-clustering for $k' \leq k$
  - For FP algorithms, pick the first $k'$ centers
  - For PG algorithms, the $k'$ clustering is the smallest guess that generates at most $k'$ centers
- 1-center problem is a special case, and we can get a stronger $(1+\varepsilon)$-approximation
Conclusion

• Introduced **distributed continuous clustering**
• Shown local and global k-center algorithms
  – Minimize communication, guarantee clustering quality
• Experimental results show
  – local and parallel guessing uses least communication
  – clustering quality is similar to centralized algorithm
  – space needed independent of the input size
• Open to study other popular clustering algorithms, and other settings
• Thank you!