Distributed k-Means and k-Median Clustering on General Topologies

Maria-Florina Balcan, Steven Ehrlich, and Yingyu Liang

Problem Setup

- **k-Clustering**: Given a set \( P \) of \( N \) points in \( \mathbb{R}^d \), find centers \( x = \{x_1, \ldots, x_k\} \) to minimize \( \sum_{p \in P} \text{cost}(p,x) \).
- Widely studied cost functions in ML & TCS
  - \( k \)-median: \( \text{cost}(p,x) = \min_{x \in x} d(p,x) \)
  - \( k \)-means: \( \text{cost}(p,x) = \min_{x \in x} \|p-x\|^2 \)
- **Modern Challenge**: data distributed over different sites, e.g., distributed databases, images and videos over networks,

Distributed Clustering:

- Communication graph: undirected graph \( G \) on \( n \) nodes with \( m \) edges, where an edge indicates that the two nodes can communicate.
- Global data: \( P \) is divided into local data sets \( P_1, \ldots, P_n \).
- Goal: efficient distributed algorithm with low communication

Our Results

- Efficient algorithm that
  - outputs \((1+\epsilon)\alpha\)-approx, given any non-distributed \( \alpha\)-approx algo
  - has low communication independent of #points in global data set: \( O(kd+nk) \) points
  - has good experimental performance
- Two stages of our distributed algorithm:
  1. Each node constructs a local portion of a global summary
  2. Communicate the local portions, and compute approximation solution on the summary

Coreset

- **Coreset**: [Kar-Peled-Mazorod, STOC'14]: short summaries capturing relevant info w.r.t. all clusterings

Definition. An \( r \)-coreset for \( P \) is a set of points \( D \) and weights \( w \) on \( D \) s.t.

\[
\forall x, (1-\epsilon)\text{cost}(P,x) \leq \sum_{p \in D} w_p \text{cost}(q,x) \leq (1+\epsilon)\text{cost}(P,x).
\]

- Non-distributed coreset construction [Feldman-Langberg, STOC'11]
  1. Compute a constant approximation solution \( A \)
  2. Sample points \( S \) with probability proportional to \( \text{cost}(p,A) \);
  \( |S| = O(kd) \) for constant \( \epsilon \)

Distributed Coreset Construction

**Algorithm** (two rounds, interactive)

1. Compute a constant approximation solution \( A_i \) for \( P_i \).
   Communicate the costs \( \text{cost}(P_i,A_i) \)
   \[
   \text{cost}(P_i,A_i) \quad \quad \text{cost}(P_i,A_i)
   \]

2. Sample points from \( P_i \) according to the multimonial distribution given by \( \text{cost}(P_i,A_i) \);
   \# sampled points = \( O(kd) \) for constant \( \epsilon \)

Analysis

- **Uniform sampling for metric balls**: \( \forall B(x,r) = \{ p : d(p,x) \leq r \} \), \( |B(x,r)| = \frac{|B(x,r)|}{n} \pm \epsilon \) when \( |S| = O(\log|\text{distinct } B(x,r) \cap P_i|^2) \)

Distributed Clustering

**Algorithm**

1. Distributed coreset construction
2. Communicate the local portions of the coreset
3. Compute approximation solution on the coreset

**Theorem.** Given any non-distributed \( \alpha\)-approx algo as a subroutine, our algo computes a \((1+\epsilon)\alpha\)-approx solution. The total communication cost is \( O(m(kd+nk)) \) points for constant \( \epsilon \)

**Total Communication on Different Networks** (for constant \( \epsilon \)):

1. Star graph: \( O(kd+nk) \) points
   - by sending the local portions of the coreset to the coordinator
2. Rooted Tree: \( O(h(kd+nk)) \) points
   - by sending the local portions of the coreset to the root
3. General Topologies: \( O(m(kd+nk)) \) points
   - Message Passing: on each node do
     - Communicate its local message to all its neighbors
     - When the node receives new message, communicate to all its neighbors

Experiments

- **Data set**: ColorHistogram (\( \approx 68k \) points in \( \mathbb{R}^{22}, k = 10, n = 25 \))
- **YearPredictionMSD** (\( \approx 0.5m \) points in \( \mathbb{R}^{45}, k = 50, n = 100 \))

**Results on ColorHistogram:**

- star, exponential
- grid, weighted
- preferential, degree-based

**Results on YearPredictionMSD:**

- star, exponential
- grid, weighted
- preferential, degree-based