Distributed k-median/k-means Clustering on General Topologies

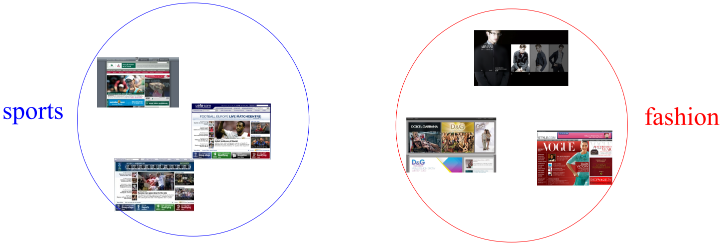
Yingyu Liang

Joint work with Maria Florina Balcan, Steven Ehrlich
Georgia Institute of Technology

To appear in NIPS 2013

k-median/k-means Clustering

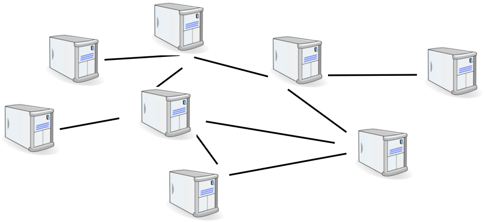
- A set P of N objects, represented as points in $\mathbf{R}^d$



- Find centers $\mathbf{x} = \{x_1, \ldots, x_k\}$ to minimize $\sum_{p \in P} \mathrm{cost}(p, \mathbf{x})$
- Widely used cost functions
  - k-median: $\mathrm{cost}(p, \mathbf{x}) = \min_{x \in \mathbf{x}} d(p, x)$
  - k-means: $\mathrm{cost}(p, \mathbf{x}) = \min_{x \in \mathbf{x}} d^2(p, x)$

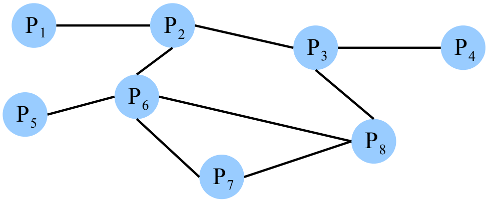
Modern Challenge: Distributed Data

- Distributed databases
- Images and videos on the Internet
- Sensor networks
- ...



Distributed Clustering

- Communication graph G with n nodes and m edges: 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 for k-median/k-means
with guarantees for clustering cost and communication cost

Related Work

1. Direct adaptation of non-distributed algorithms, e.g. Lloyd's method [Forman et al., 2000; Datta et al., 2005]
   - no consideration on the communication cost
2. Transmitting summaries of local data to central coordinator [Januzaj et al., 2003; Kargupta et al., 2001]
   - no guarantee on clustering cost
   - not for general communication topologies

Our Results

A distributed algorithm for k-median/k-means that

1. produces $(1+\epsilon)\alpha$-approximation, using any α-approximation non-distributed algorithm as a subroutine
2. with total communication cost
   - independent of #points N
   - linear in #clusters k and the dimension d
   - linear in #nodes n and #edges m

Our Results

Two stages of our distributed algorithm

1. Constructs a global summary of the data
   - each node constructs a local portion of the summary
2. Compute approximation solution on the summary
   - each node broadcasts its local portion

Outline

Coreset

Weighted points whose cost approximates that of the original data

Coreset [Har-Peled and Mazumdar, 2004]

An ϵ-coreset for a set of points P with respect to a cost objective function is a set of points D and a set of weights $w\colon D \to \mathbf{R}$ such that for any set of centers $\mathbf{x}$,

$$(1-\epsilon)\mathrm{cost}(P,\mathbf{x}) \leq \sum_{p \in D} w_p \mathrm{cost}(p,\mathbf{x}) \leq (1+\epsilon)\mathrm{cost}(P,\mathbf{x}).$$

Coreset Construction in the Non-distributed Setting

Coreset construction [Feldman and Langberg, 2011]

1. Compute a constant approximation solution A
2. Sample points S with probability proportional to $\mathrm{cost}(p, A)$
3. Let the coreset $D = S \cup A$ (with weights specified later)

Naïve Adaptation in Distributed Setting

COMBINE

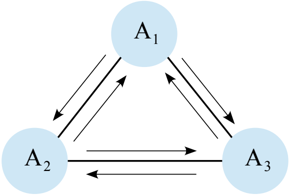
1. Compute a coreset for each local data set
2. Combine these local coresets to get a global coreset

- Need to transmit n coresets
- Can we do with 1 coreset?

Distributed Coreset Construction

Algorithm 1: Distributed coreset construction

1. Compute a constant approximation solution A_i for P_i
2. Broadcast the costs $\mathrm{cost}(P_i, A_i)$
3. Let $\frac{|S_i|}{\sum_j |S_j|} = \frac{\mathrm{cost}(P_i, A_i)}{\sum_j \mathrm{cost}(P_j, A_j)}$;
   Sample S_i from P_i with probability proportional to $\mathrm{cost}(p, A_i)$
4. Let the coreset $D = \bigcup_i (S_i \cup A_i)$ (with weights specified later)

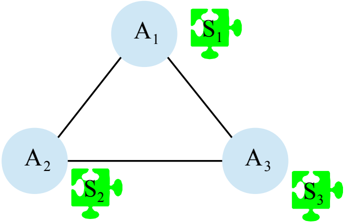


Distributed Coreset Construction

Algorithm 1: Distributed coreset construction

1. Compute a constant approximation solution A_i for P_i
2. Broadcast the costs $\text{cost}(P_i, A_i)$
3. Let $\frac{|S_i|}{\sum_j |S_j|} = \frac{\text{cost}(P_i,A_i)}{\sum_j \text{cost}(P_j,A_j)}$;
   Sample S_i from P_i with probability proportional to $\text{cost}(p, A_i)$
4. Let the coreset $D = \bigcup_i (S_i \cup A_i)$ (with weights specified later)

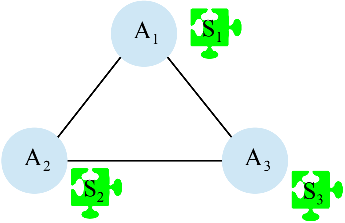
the size of the local sample is proportional to the local cost



Distributed Coreset Construction

Algorithm 1: Distributed coreset construction

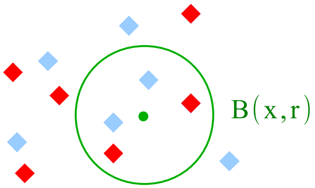
1. Compute a constant approximation solution A_i for P_i
2. Broadcast the costs $\mathrm{cost}(P_i, A_i)$
3. Let $\frac{|S_i|}{\sum_j |S_j|} = \frac{\mathrm{cost}(P_i, A_i)}{\sum_j \mathrm{cost}(P_j, A_j)}$;
   Sample S_i from P_i with probability proportional to $\mathrm{cost}(p, A_i)$
4. Let the coreset $D = \bigcup_i (S_i \cup A_i)$ (with weights specified later)



Coreset Construction Analysis
Intuition for Sampling

Sample a set S uniformly at random from P.
Let $B(x,r) = \{p : d(x,p) \le r\}$.

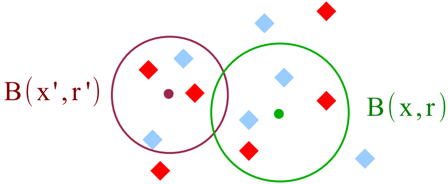
- For fixed $B(x,r)$, w.h.p. $\frac{|B(x,r)\cap S|}{|S|} = \frac{|B(x,r)\cap P|}{|P|} \pm \epsilon$ when $|S| = \tilde{O}(1/\epsilon^2)$



Coreset Construction Analysis
Intuition for Sampling

Sample a set S uniformly at random from P.
Let $B(x,r) = \{p : d(x,p) \leq r\}$.

- For any $B(x,r)$, w.h.p. $\frac{|B(x,r)\cap S|}{|S|} = \frac{|B(x,r)\cap P|}{|P|} \pm \epsilon$ when $|S| = \tilde{O}(\log[\#\text{distinct } B(x,r)\cap P]/\epsilon^2)$



Let F be a set of functions from P to $\mathbf{R}_{\geq 0}$.
For $f \in F$, let $B(f, r) = \{p : f(p) \leq r\}$.

- Special case: $B(f_x, r) = B(x, r)$ when $f_x(p) = d(x, p)$

Let F be a set of functions from P to $\mathbf{R}_{\geq 0}$.
For $f \in F$, let $B(f, r) = \{p : f(p) \leq r\}$.

Sampling Lemma (weighted sampling, general functions)

Let $m_p = \max_{f \in F} f(p)$. Sample S from P with probability proportional to m_p, and let $w_p = \frac{\sum_q m_q}{m_p |S|}$.
If $|S| = \tilde{O}(\log[\#\text{distinct } B(f, r) \cap P]/\epsilon^2)$, then w.h.p.

$$\forall f \in F, \left| \sum_{p \in P} f(p) - \sum_{p \in S} w_p f(p) \right| \leq \epsilon \sum_{p \in P} m_p.$$

Let F be a set of functions from P to $\mathbf{R}_{\geq 0}$.
For $f \in F$, let $B(f, r) = \{p : f(p) \leq r\}$.

Sampling Lemma (weighted sampling, general functions)

Let $m_p = \max_{f \in F} f(p)$. Sample S from P with probability proportional to m_p, and let $w_p = \frac{\sum_q m_q}{m_p |S|}$.
If $|S| = \tilde{O}(\log[\#\text{distinct } B(f, r) \cap P]/\epsilon^2)$, then w.h.p.

$$\forall f \in F, \left| \sum_{p \in P} f(p) - \sum_{p \in S} w_p f(p) \right| \leq \epsilon \sum_{p \in P} m_p.$$

Proof idea: replace p with m_p copies p'; let $f(p') = f(p)/m_p$

Let F be a set of functions from P to $\mathbf{R}_{\geq 0}$.
For $f \in F$, let $B(f, r) = \{p : f(p) \leq r\}$.

Complexity of F: $\log[\#\text{distinct } B(f, r) \cap P]$

- Connection to VC-dimension:

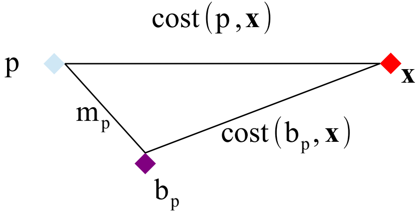
$$I_{f,r}(p) = \begin{cases} +1 & \text{if } p \in B(f, r) \\ -1 & \text{otherwise} \end{cases}$$

$$\log[\#\text{distinct } B(f, r) \cap P] \leq O(1)\text{VC-dimension}(\{I_{f,r}\}).$$

Coreset Construction Analysis
Sampling Lemma for k-median

- Natural attempt: $f_{\mathbf{x}}(p) = \mathrm{cost}(p, \mathbf{x})$
  Fail since $f_{\mathbf{x}}(p)$ unbounded
- Another attempt:
  For $p \in P_i$, let b_p denote its nearest center in A_i.
  Set $f_{\mathbf{x}}(p) = \mathrm{cost}(p, \mathbf{x}) - \mathrm{cost}(b_p, \mathbf{x})$, then $m_p = \mathrm{cost}(p, A_i)$.

- Natural attempt: $f_{\mathbf{x}}(p) = \text{cost}(p, \mathbf{x})$
  Fail since $f_{\mathbf{x}}(p)$ unbounded
- Another attempt:
  For $p \in P_i$, let b_p denote its nearest center in A_i.
  Set $f_{\mathbf{x}}(p) = \text{cost}(p, \mathbf{x}) - \text{cost}(b_p, \mathbf{x})$, then $m_p = \text{cost}(p, A_i)$.



For $p \in P_i$, let b_p denote its nearest center in A_i.
Set $f_{\mathbf{x}}(p) = \text{cost}(p, \mathbf{x}) - \text{cost}(b_p, \mathbf{x})$, then $m_p = \text{cost}(p, b_p)$.

By Sampling Lemma,

$$\forall \mathbf{x}, \left| \sum_{p \in P} f_{\mathbf{x}}(p) - \sum_{p \in S} w_p f_{\mathbf{x}}(p) \right| \leq \epsilon \sum_{p \in P} m_p.$$

$= |\sum_{p \in P} \text{cost}(p, \mathbf{x}) - \sum_{p \in D} w_p \text{cost}(p, \mathbf{x})|$

For $p \in P_i$, let b_p denote its nearest center in A_i.
Set $f_{\mathbf{x}}(p) = \mathrm{cost}(p, \mathbf{x}) - \mathrm{cost}(b_p, \mathbf{x})$, then $m_p = \mathrm{cost}(p, b_p)$.

By Sampling Lemma,

$$\forall \mathbf{x}, \left| \sum_{p \in P} f_{\mathbf{x}}(p) - \sum_{p \in S} w_p f_{\mathbf{x}}(p) \right| \leq \epsilon \sum_{p \in P} m_p.$$

$= \epsilon \sum_i \mathrm{cost}(P_i, A_i) = O(\epsilon) OPT$

Coreset Construction Analysis

Algorithm 1: Distributed coreset construction

1. Compute a constant approximation solution A_i for P_i;
2. Broadcast the costs $\mathrm{cost}(P_i, A_i)$
3. Sample S_i from P_i with probability proportional to $\mathrm{cost}(p, A_i)$
4. Let the coreset $D = \bigcup_i (S_i \cup A_i)$

Theorem (Distributed Coreset Construction)

Algorithm 1 produces an ϵ-coreset. The size of the coreset is $\tilde{O}(kd + nk)$ for constant ϵ.

- By a geometric argument [Feldman and Langberg, 2011], $\log[\#\text{distinct } B(f, r) \cap P] = O(kd)$

Outline

General Communication Topologies

Message-Passing

▷ `Broadcast messages` $\{I_j\}_{j=1}^n$`, where` I_j `is on node` j

On each node i do:

1. Initialize $R_i = \{I_i\}$ and send I_i to all neighbors.
2. When $R_i \neq \{I_j\}_{j=1}^n$,
   if receive $I_j \notin R_i$,
   then $R_i = R_i \cup \{I_j\}$ and send I_j to all neighbors.

Total communication cost: $O(m \sum_{j=1}^n |I_j|)$

Distributed Clustering on General Topologies

Algorithm 2: Distributed Clustering

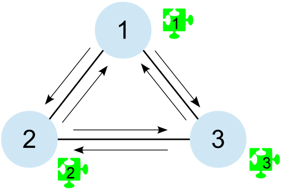
1. Call the distributed coreset construction algorithm
2. Broadcast the local coreset portions by Message-Passing
3. Compute an approximation solution on the coreset

Theorem (Distributed Clustering on General Graphs)

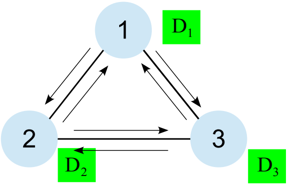
Given any α-approximation algorithm as a subroutine, Algorithm 2 computes a $(1+\epsilon)\alpha$-approximation solution.
The total communication cost is $\tilde{O}(m(kd+nk))$ for constant ϵ.

Distributed Clustering on General Topologies

Our algorithm: $\tilde{O}(m(kd + nk))$



COMBINE: $\tilde{O}(mnkd)$



Distributed Clustering on Rooted Trees

Algorithm 3: Distributed Clustering on Rooted Trees

1. Call the distributed coreset construction algorithm
2. Send the local coreset portions to the root
3. Compute an approximation solution on the coreset

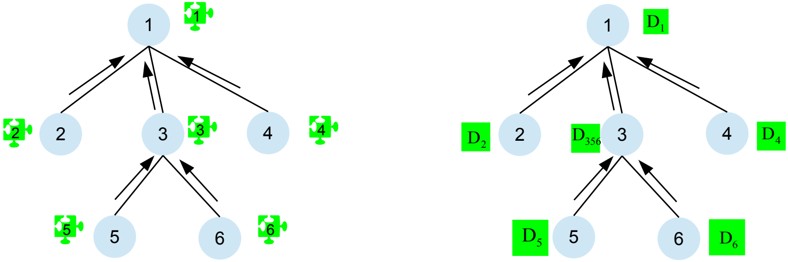
Theorem (Distributed Clustering on Rooted Trees)

Given any α-approximation algorithm as a subroutine, Algorithm 3 computes a $(1+\epsilon)\alpha$-approximation solution.
The total communication cost is $\tilde{O}(h(kd+nk))$ for constant ϵ, where h is the height of the tree.

Distributed Clustering on Rooted Trees

Our Algorithm:
$\tilde{O}(h(kd + nk))$

[Zhang et al., 2008]:
$\tilde{O}(h^2nkd)$ for k-median
$\tilde{O}(h^4nkd)$ for k-means

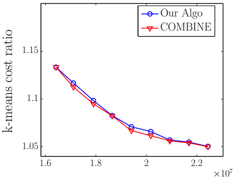


Outline

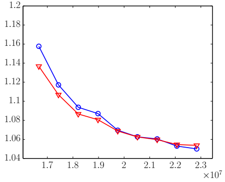
Experiment Setup

- Data set: YearPredictionMSD (≈ 0.5m points in $\mathbf{R}^{90}$)
- Communication graphs: random, grid, preferential
- Partition into 100 local data sets;
  Partition methods: uniform, weighted, similarity/degree-based
- Evaluation criteria:
  k-means cost ($k = 50$) at the same communication budget

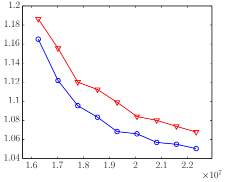
Experiments for Distributed Clustering
On Graphs



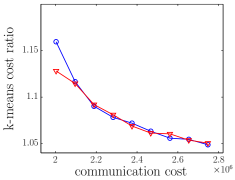
random graph, uniform



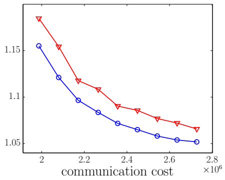
random graph, similarity-based



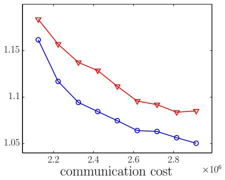
random graph, weighted



grid graph, similarity-based

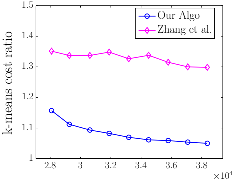


grid graph, weighted

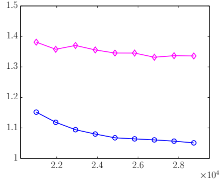


preferential graph, degree-based

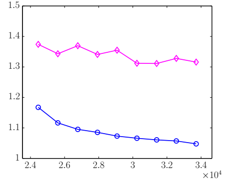
Experiments for Distributed Clustering
On Spanning Trees



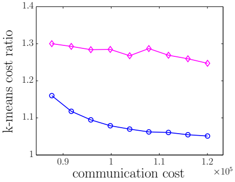
random graph, uniform



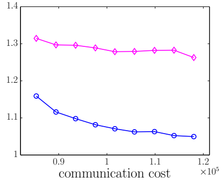
random graph, similarity-based



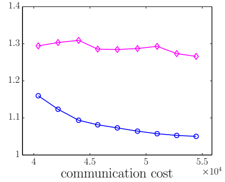
random graph, weighted



grid graph, similarity-based



grid graph, weighted



preferential graph, degree-based

Current Work

- Improve communication cost
- More experiments on high dimensional data
- Distributed optimization

$$\min_{\mathbf{x}} \sum_{i} \sum_{p \in P_i} f_{\mathbf{x}}(p)$$

Thanks!

Feldman, D. and Langberg, M. (2011).
A unified framework for approximating and clustering data.
In *Proceedings of the Annual ACM Symposium on Theory of Computing*.

Har-Peled, S. and Mazumdar, S. (2004).
On coresets for k-means and k-median clustering.
In *Proceedings of the Annual ACM Symposium on Theory of Computing*.

Zhang, Q., Liu, J., and Wang, W. (2008).
Approximate clustering on distributed data streams.
In *Proceedings of the IEEE International Conference on Data Engineering*.