Distributed PCA and k-Means Clustering

Yingyu Liang

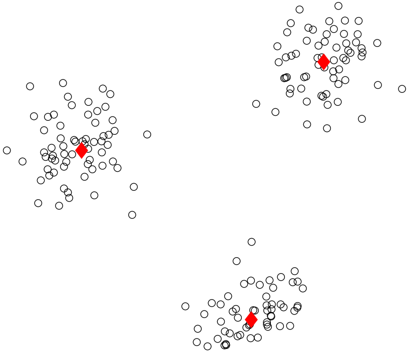
Joint work with Maria Florina Balcan, Vandana Kanchanapally

Georgia Institute of Technology

k-Means Clustering

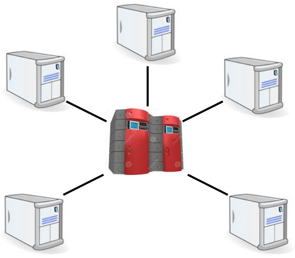
- Given a set P of points in $\mathbf{R}^d$ and #clusters k
- Find centers $\mathbf{c} = \{c_1, \ldots, c_k\}$ to minimize the k-means cost

$$\sum_{p \in P} \min_i ||p - c_i||_2^2$$



Distributed Clustering

- Global data P consists of local data sets $P_1, \ldots, P_s$
  - Distributed databases
  - Images and videos on the Internet
  - Sensor networks ...
- Challenge: how to lower the communication needed?



Our Results

Algorithm for distributed k-means for high dimensional data

- loses $(1 + \epsilon)$-approx factor compared to non-distributed
- #points communicated independent of $|P|$ and dim d
- has positive experimental results

Coreset

Coreset [HarPeled-Mazumdar,STOC04]

short summaries capturing relevant info w.r.t. all clusterings

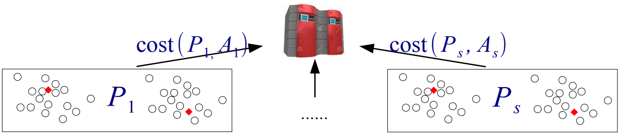
Definition

An ϵ-coreset for P is a set of points D and weights w on D s.t.
$\forall \mathbf{c}, (1-\epsilon)\mathrm{cost}(P, \mathbf{c}) \leq \sum_{q \in D} w_q \mathrm{cost}(q, \mathbf{c}) \leq (1+\epsilon)\mathrm{cost}(P, \mathbf{c})$.

Distributed Coreset and Clustering [Balcan-Ehrlich-Liang,NIPS13]

Distributed coreset construction (two rounds, interactive):

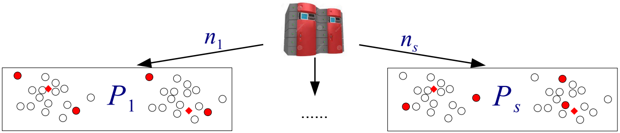
1. Compute a constant approximation solution A_i for P_i.
   Communicate the costs $\mathrm{cost}(P_i, A_i)$
2. Sample $\tilde{O}(kd)$ points.
   #points from P_i obeys multinomial$[\{\mathrm{cost}(P_i, A_i)\}_i]$



Distributed Coreset and Clustering [Balcan-Ehrlich-Liang,NIPS13]

Distributed coreset construction (two rounds, interactive):

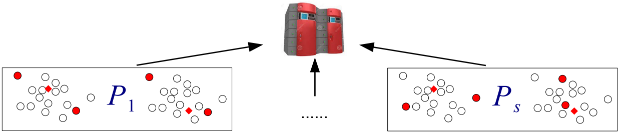
1. Compute a constant approximation solution A_i for P_i.
   Communicate the costs $\mathrm{cost}(P_i, A_i)$
2. Sample $\tilde{O}(kd)$ points.
   #points from P_i obeys Multinomial$\{\mathrm{cost}(P_i, A_i)\}_i$



Distributed Coreset and Clustering [Balcan-Ehrlich-Liang,NIPS13]

Distributed coreset construction (two rounds, interactive):

1. Compute a constant approximation solution A_i for P_i.
   Communicate the costs $\mathrm{cost}(P_i, A_i)$
2. Sample $\tilde{O}(kd)$ points.
   #points from P_i obeys Multinomial$\{\mathrm{cost}(P_i, A_i)\}_i$



Used for distributed k-means clustering:

1. Lose $(1+\epsilon)$ approx factor compared to non-distributed
2. Communication on star: $\tilde{O}(kd + sk)$ points for const ϵ

Distributed k-Means Clustering for High Dimensional Data

Algorithm

1. Perform distributed PCA to $O(k/\epsilon^2)$ dimension
2. Perform distributed clustering on the projected data

- Lose $(1+\epsilon)$ approx factor due to distributed PCA
- Communication cost on star network for constant ϵ:
  - Distributed PCA: $O(sk)$ points in $\mathbf{R}^d$
  - Distributed Clustering: $\tilde{O}(k^2+sk)$ points in $\mathbf{R}^{O(k)}$

Non-Distributed PCA

SVD on data

1. Perform SVD $A = UDE^T$
2. $D^{(t)}$: first t columns of D
   $E^{(t)}$: first t columns of E
3. Let $A^{(t)} = UD^{(t)}(E^{(t)})^T$

Equivalent:

Eigen-factorize covariance

1. Compute $S = A^T A$ and eigen-factorize $S = E\Lambda E^T$
2. Project the data on $E^{(t)}$

Distributed PCA

Algorithm: PCA onto dimension t

▷ *Round 1: Local PCA*

- Each server: SVD $P_i = U_i D_i E_i^T$
- Each server: communicate $D_i^{(t)}$ and $E_i^{(t)}$ to the coordinator

▷ *Round 2: Global PCA*

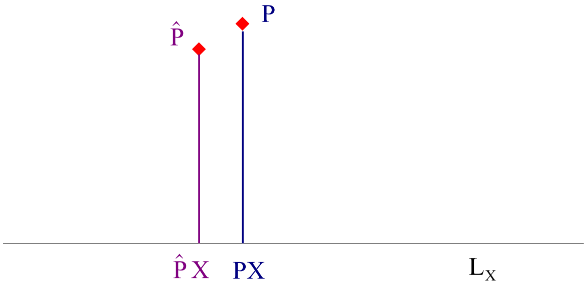
- Coordinator: compute covariance $S = \sum_i E_i^{(t)} D_i^{(t)} D_i^{(t)} (E_i^{(t)})^T$
  factorize $S = E \Lambda E^T$
- Coordinator: communicate $E^{(t)}$ to each server
- All servers: project the data on $E^{(t)}$ to get $\hat{P}$

$$P=\begin{bmatrix}P_1\\ \vdots\\ P_s\end{bmatrix}\begin{array}{c}\xrightarrow{\text{Local PCA}}\\ \vdots\\ \xrightarrow{\text{Local PCA}}\end{array}\begin{bmatrix}P_1^{(t)}\\ \vdots\\ P_s^{(t)}\end{bmatrix}=P^{(t)}\xrightarrow{\text{Global PCA}}\hat{P}$$

Property of Distributed PCA

Theorem (informal): Distributed PCA

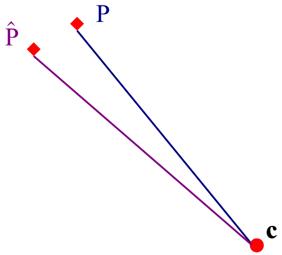
Let L_X be a k-dim subspace.
When $t \gg k$, the projections of $\hat{P}$ and P on L_X are close.



Property of Distributed PCA

Theorem (informal): Distributed PCA

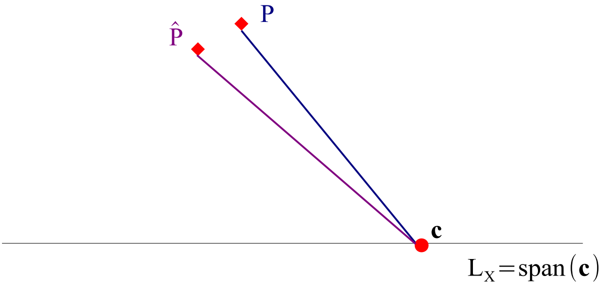
Let L_X be a k-dim subspace.
When $t \gg k$, the projections of $\hat{P}$ and P on L_X are close.



Property of Distributed PCA

Theorem (informal): Distributed PCA

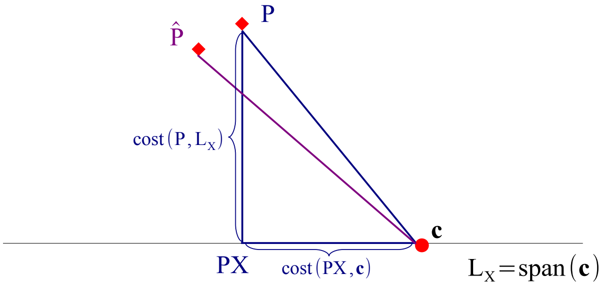
Let L_X be a k-dim subspace.
When $t \gg k$, the projections of $\hat{P}$ and P on L_X are close.



Property of Distributed PCA

Theorem (informal): Distributed PCA

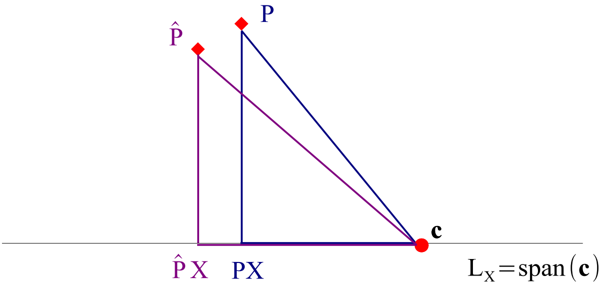
Let L_X be a k-dim subspace.
When $t \gg k$, the projections of $\hat{P}$ and P on L_X are close.



Property of Distributed PCA

Theorem (informal): Distributed PCA

Let L_X be a k-dim subspace.
When $t \gg k$, the projections of $\hat{P}$ and P on L_X are close.



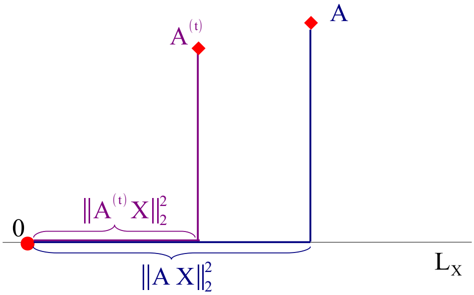
Property of SVD

Lemma: SVD Truncation [Feldman-Schmidt-Sohler,SODA13]

Let $A = UDE^T$ and its SVD Truncation $A^{(t)} = UD^{(t)}(E^{(t)})^T$.
For any k-dim subspace L_X, when $t \geq O(k/\epsilon^2)$:

1) $0 \leq ||AX||_2^2 - ||A^{(t)}X||_2^2 \leq \epsilon^2 \text{cost}(A, L_X)$.

2) $0 \leq ||AX - A^{(t)}X||_2^2 \leq \epsilon^2 \text{cost}(A, L_X)$.



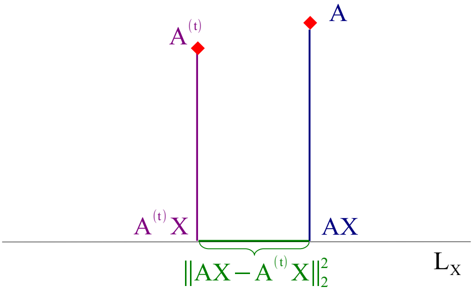
Property of SVD

Lemma: SVD Truncation `[Feldman-Schmidt-Sohler,SODA13]`

Let $A = UDE^T$ and its SVD Truncation $A^{(t)} = UD^{(t)}(E^{(t)})^T$.
For any k-dim subspace L_X, when $t \geq O(k/\epsilon^2)$:

1) $0 \leq ||AX||_2^2 - ||A^{(t)}X||_2^2 \leq \epsilon^2 \text{cost}(A, L_X)$.

2) $0 \leq ||AX - A^{(t)}X||_2^2 \leq \epsilon^2 \text{cost}(A, L_X)$.



Property of Distributed PCA

Theorem: Distributed PCA

For any k-dim subspace L_X, when $t \geq O(k/\epsilon^2)$:

1) $0 \leq ||PX||_2^2 - ||\hat{P}X||_2^2 \leq \epsilon^2 \mathrm{cost}(P, L_X).$

2) $0 \leq ||PX - \hat{P}X||_2^2 \leq \epsilon^2 \mathrm{cost}(P, L_X),$

Proof: combine the bounds for the SVD truncations

Distributed k-Means Clustering for High Dimensional Data

Algorithm

1. Perform distributed PCA to $O(k/\epsilon^2)$ dimension
2. Perform distributed clustering on the projected data

- Lose $(1+\epsilon)$ approx factor due to distributed PCA
- Communication cost on star network for constant ϵ:
  - Distributed PCA: $O(sk)$ points in $\mathbf{R}^d$
  - Distributed Clustering: $\tilde{O}(k^2+sk)$ points in $\mathbf{R}^{O(k)}$

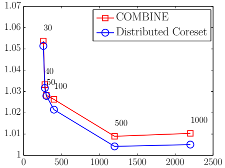
Experiments

Experiments on UCI data sets: (#clusters $k = 20$)

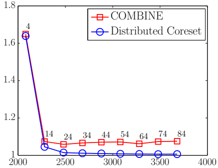
- sports activities: $9,210$ points in $\mathbf{R}^{5625}$, $s = 10$
- MNIST handwritten digits: $70,000$ points in $\mathbf{R}^{784}$, $s = 100$
- BOWnytimes: $300,000$ points in $\mathbf{R}^{102660}$, $s = 100$

Experiment results: can reduce dimension to around 20 while increasing the k-means cost by less than 10%

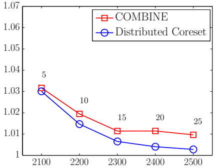
Experiments



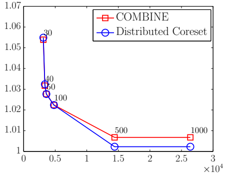
sports activities, star network



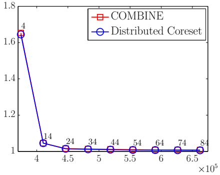
MNIST, star network



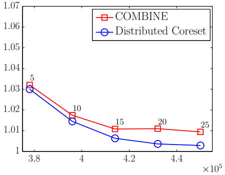
BOWnytimes, star network



sports activities, grid network



MNIST, grid network



BOWnytimes, grid network

Figure: k-means cost v.s. communicated #points. number labels: PCA dimensions.

Thanks!