DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Diversity maximization in MapReduce and Streaming

Under Cardinality and Matroid Constraints

Andrea Pietracaprina
University of Padova

Joint work with:
M. Ceccarello, G. Pucci (U. Padova), and E. Upfal (Brown U.)

[VLDB17] and [WSDM18]

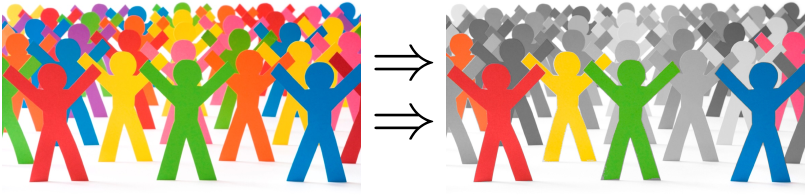
Outline

- Problem definition and applications
- Background
- Summary of results
- Our approach (cardinality constraint):
  - Core-set construction
  - MapReduce implementation
  - Streaming implementation
  - Future space savings
- Partition and transversal matroids
- Experiments
- Conclusions and future work

Problem definition and applications

Objective:

For a given dataset, determine the most diverse subset of given (small) size k



Applications

← News/document aggregators

↑ e-commerce ↑

← Facility location

Diversity maximization: formal definition

Given:

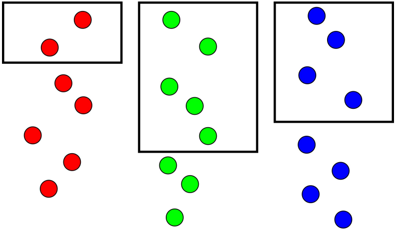
1. Set S of points in a metric space Δ
2. Distance function $d : \Delta \times \Delta \to R^+ \cup \{0\}$
3. (Distance-based) diversity function $\text{div} : 2^\Delta \to R^+ \cup \{0\}$
4. Integer $k > 1$

Return

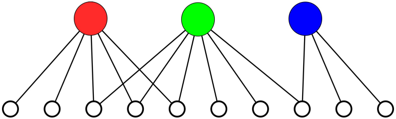
$$S^* \subset S, |S^*| = k \quad \text{s.t.} \quad S^* = \text{argmax}_{S' \subseteq S, |S'|=k} \, \text{div}(S')$$

Matroids allow to express more complicated constraints, like categorization of elements

Partition matroid



Transversal matroid



Matroid over a set S:

$$\mathcal{M} = (S, \mathcal{I}(S))$$

$$\text{rank}(\mathcal{M}) = \max_{X \in \mathcal{I}(S)} |X|$$

Diversity maximization under matroid constraints

Given:

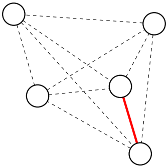
1. Set S of points in a metric space Δ
2. Distance function $d : \Delta \times \Delta \rightarrow R^+ \cup \{0\}$
3. (Distance-based) diversity function $\text{div} : 2^\Delta \rightarrow R^+ \cup \{0\}$
4. Matroid $\mathcal{M} = (S, \mathcal{I}(S))$ of rank $\text{rank}(\mathcal{M})$
5. Integer $1 \leq k \leq \text{rank}(\mathcal{M})$

Return

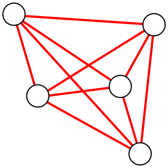
$$S^* \subset S, S \in \mathcal{I}(S), |S^*| = k \quad \text{s.t.} \quad S^* = \text{argmax}_{S' \subseteq S, |S'|=k} \text{div}(S')$$

Diversity measures studied in this work

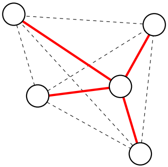
remote-edge



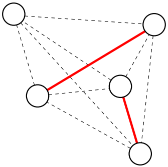
remote-clique



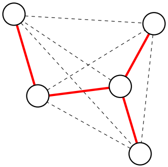
remote-star



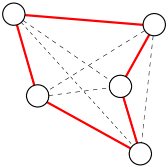
remote-bipartition



remote-tree



remote-cycle



All measures are NP-Hard to optimize

Background

Sequential approximation and hardness results

Problem	Seq. Approx.	LB
Remote-edge	2	≥ 2
Remote-clique	2	$\geq 2-\epsilon$
Remote-star	2	–
Remote-bipartition	3	–
Remote-tree	4	≥ 2
Remote-cycle	3	≥ 2

Specialized results (hardness and better approx. ratios) for remote clique and remote edge under Euclidean distances

β-core-set [Agarwal et al.'95]

- A small subset T (core-set) of input S s.t. $\mathrm{div}_k(T) \geq (1/\beta)\,\mathrm{div}_k(S)$

- Compute final solution on T.

β-composable core-set [Indyk et al.'14]

- Partitioned input $S = S_1 \cup S_2 \cup \cdots \cup S_\ell$
- β-composable core-sets $T_i \subset S_i \Rightarrow \bigcup T_i$ is a β-core-set

Known β-composable core-sets for diversity maximization under cardinality constraints

([Indyk et al.'14,Aghamolaei et al.'15])

	β	α_{seq}	$\beta \cdot \alpha_{\text{seq}}$
Remote-edge	3	2	6
Remote-clique	$6+\epsilon$	2	$12+\epsilon$
Remote-star	12	2	24
Remote-bipartition	18	3	54
Remote-tree	4	4	16
Remote-cycle	3	3	9

- α_{seq} = best sequential approximation ratio
- General metric spaces
- Core-set size: k

The case of matroid constraints

- [Abbassi et al. 13]
- Remote-clique measure
- Sequential algorithm for remote-clique based on local search
- $2 + \epsilon$ approximation
- $\Omega(n^2)$ time

MapReduce

- Data represented as multiset of key-value pairs
- Algorithms execute as sequences of rounds
- Architecture: cluster of machines (workers)
- One round (distributed among workers):
  - Map function: applied to each key-value pair
  - Reduce function: applied to subsets of key-value pairs grouped by key.
- Shuffle of data at each round
- Performance indicators: #rounds and space required at each worker to execute map/reduce functions

Streaming

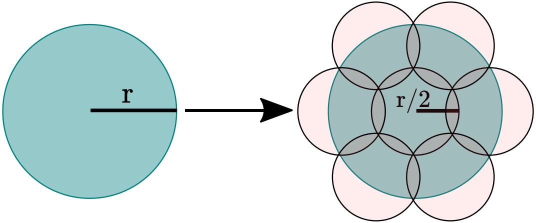
- One processor with limited space
- Input provided as a continuous stream: too large to fit in the available memory
- ≥ 1 passes over the input
- Performance indicators: #passes and space available at the processor.

Proposition: *The known composable core-sets for k-diversity maximization yield 2-round MapReduce and 1-pass Streaming algorithms using* $O\left(\sqrt{k|S|}\right)$ *space.*

Summary of Results

Our setting

Metric spaces of Bounded Doubling Dimension: $\exists D = O(1)$ s.t. any ball of radius r is covered by $\leq 2^D$ balls of radius $r/2$



- Euclidean spaces.
- Shortest-path distances of mildly expanding topologies.
- Low-dimensional pointsets from arbitrary metric space.

Our Results (cardinality constraint):

- Improved β-(composable) core-sets: $\beta = 1 + \epsilon$
- Overall approximation: $\alpha_{\text{seq}} + \epsilon$
- 1-pass Streaming and 2-round MapReduce algorithms using space:

	Streaming	MapReduce
r-edge/cycle	$O(k(c/\epsilon)^D)$	$O\left(\sqrt{k\lvert S\rvert(c/\epsilon)^D}\right)$
other div's	$O(k^2(c/\epsilon)^D)$	$O\left(k\sqrt{\lvert S\rvert(c/\epsilon)^D}\right)$

for a suitable constant c.

Summary of results

- 1 extra pass/round brings space bounds for other div$'s$ down to those for r-edge/cycle

	Streaming	MapReduce
all div's	$O(k(c/\epsilon)^D)$	$O\left(\sqrt{k\|S\|(c/\epsilon)^D}\right)$

for a suitable constant c.

Our Results (remote-clique, matroid constraints):

- 2-rounds in MapReduce, 1 pass in streaming
- $2 + \epsilon$ approximation
- space requirements:

	Streaming	MapReduce
Partition matroid	$O(k^2(c/\epsilon)^D)$	$O\left(k\sqrt{\lvert S\rvert(c/\epsilon)^D}\right)$
Transversal matroid	$O(k^3(c/\epsilon)^D)$	$O\left(k^2\sqrt{\lvert S\rvert(c/\epsilon)^D}\right)$

for a suitable constant c.

- MapReduce algorithms are oblivious to D.
- Streaming algorithms can be made oblivious to D with 1 extra pass.

Our approach

(cardinality constaint)

Core-set construction: algorithm

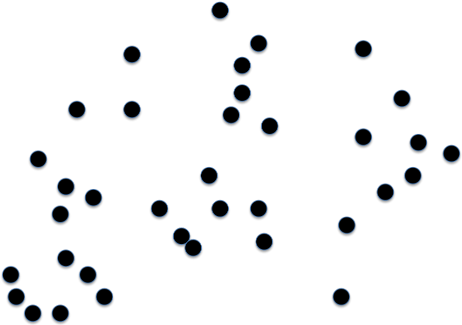
Input dataset: S

Optimal solution $\mathrm{OPT} \subset S$, with $|\mathrm{OPT}| = k$

MAIN IDEA: Compute core-set T such that each $o \in \mathrm{OPT}$ has a (distinct) proxy $p(o) \in T$ with "small"

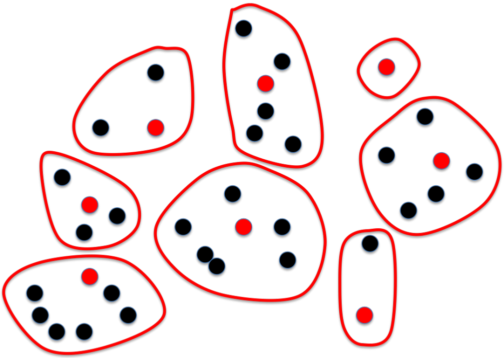
1. Partition S into $\tau > k$ clusters of small radius (τ function of doubling dimension)
2. $T = \{\text{cluster centers}\}$
3. If injectivity of $p(\cdot)$ required (remote-clique/start/bipartition/tree): $T = \{\text{cluster centers}\} \cup \{\leq k-1 \text{ delegates for each cluster}\}$.

Core-set construction: algorithm



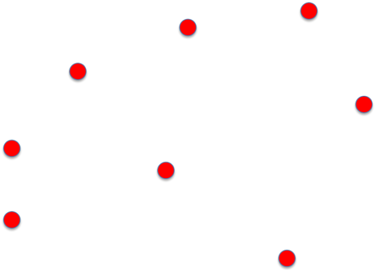
- $k = 3$, $\tau = 8$

Core-set construction: algorithm



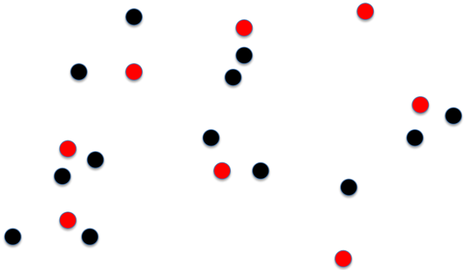
- Compute τ-center clustering

Core-set construction: algorithm



- No injectivity required: $T = \{\text{cluster centers}\}$ $(|T| = \tau)$

Core-set construction: algorithm



- Injectivity required: $T = \{k \text{ points per cluster}\}$ $(|T| \leq k \cdot \tau)$

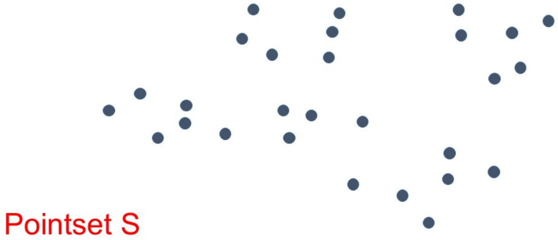
Core-set construction: analysis

- Radius: $r_k =$ min radius of k-clustering of S
- Farness: $\rho_k =$ max min distance between k points of S

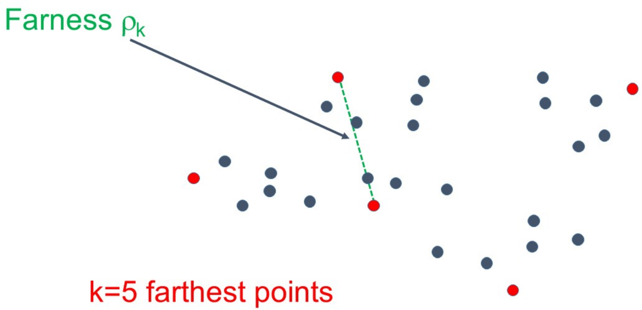
Claim: For every k, $r_k \leq \rho_k$

Proof: take k centers with Gonzalez85's algorithm (Farthest-First traversal). Their pairwise distnce is at least the radius $r \geq r_k$ of the associated clustering

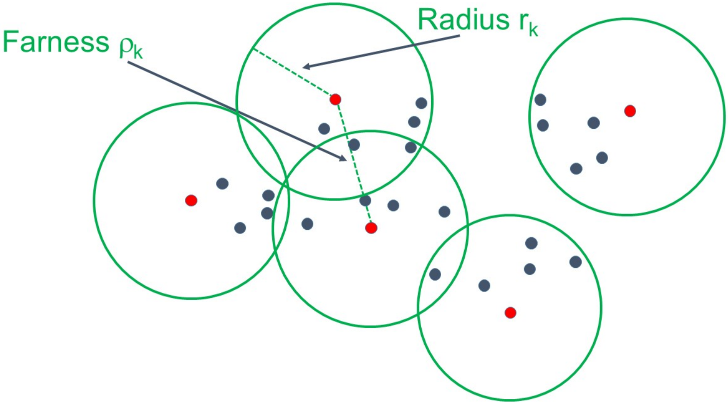
Core-set construction: analysis



Core-set construction: analysis



Core-set construction: analysis



Core-set construction: analysis

Claim: If S has doubling dimension D and $\tau = (16/\epsilon)^D k$ then

$$r_\tau \leq \epsilon/8 r_k$$

- Focus on remote-clique (similiar for other div's)
- Let $\rho = \text{div}(\text{OPT})/\binom{k}{2}$
- Observe that: $\rho \geq \rho_k \geq r_k$

Theorem

For $\epsilon < 1/2$ and $\tau = (16/\epsilon)^D k$, T is a $(1+\epsilon)$-core-set for S of size $O\left(k^2(16/\epsilon)^D\right)$

Proof.

- $\exists$ an injective $p(\cdot)$ such that for each $o \in \text{OPT}$, $p(o) \in T$ and

$$d(o, p(o)) \le 2r_T \le (\epsilon/4)r_k \le (\epsilon/4)\rho_k \le (\epsilon/4)\rho$$

- Hence:

$$\begin{aligned}
\text{div}_k(T) &\ge \sum_{o_1,o_2 \in OPT} d(p(o_1), p(o_2)) \text{ (injectivity!)} \\
&\ge \sum_{o_1,o_2 \in OPT} [d(o_1, o_2) - d(o_1, p(o_1)) - d(o_2, p(o_2))] \\
&\ge \sum_{o_1,o_2 \in OPT} d(o_1, o_2) - \binom{k}{2} 2(\epsilon/4)\rho \ge \frac{\text{div}_k(S)}{(1+\epsilon)}
\end{aligned}$$

$\square$

Core-set construction: other issues

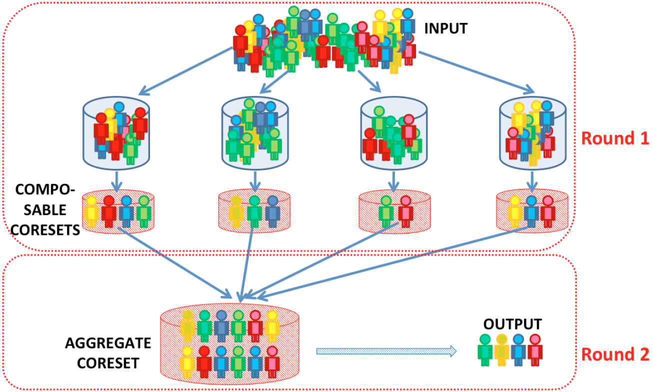
Good clustering?

- Optimal k-center clustering is NP-hard.
- $O(1)$-approximation (e.g., [Gonzalez85]) suffices.

Composability?

- Let $S = S_1 \cup S_2 \cup \cdots \cup S_\ell$
- Extract a core-set $T_i \subset S_i$ as before
- $T = \bigcup T_i$ is a $(1+\epsilon)$-core-set of size $O\left(\ell k^2 (1/\epsilon)^D\right)$.

MapReduce implementation



MapReduce implementation

- 2 rounds with $O(k\sqrt{n(c/\epsilon)^D})$ space.
- Obliviousness to D:

  use Farthest-First Traversal algorithm for τ-center stopping at $\tau > k$ ensuring sufficiently small radius.
- Approximation guarantee: $\alpha_{\mathrm{seq}} + \epsilon$.
- Random partition yields $O(\sqrt{kn \log n(c/\epsilon)^D})$ space w.h.p.
- Further decrease in space with multi-round recursion.

Implementation

- Compute a $(1+\epsilon)$-core-set using a variant of the $(2+\delta)$-approximate τ-center algorithm of [McCuthcen et al.'08], with $\tau = O\left((c/\epsilon)^D\right)$ (knowledge of D required!).
- Run sequential approximation on the coreset.

Performace

- 1 pass, $O\left(k^2(c/\epsilon)^D\right)$ space (no dependence on $|S|$!)
- Approximation guarantee: $\alpha_{\text{seq}} + \epsilon$.
- Obliviousness to D can be obtained with an extra pass.

Coping with injective proxy functions

The diversity problems requiring injective proxy functions incur a $\Theta(k)$ space blowup (core-sets $=\{k$ points per cluster$\}$)

Workaround (idea)

- Generalize diversity problems to multisets and adapt sequential approximation algorithms to work on multisets
- Generalized core-set: multiset of cluster centers $\{c_i\}$, each with multiplicity $m_i = \min\{|C_i|, k\}$
- Generalized core-sets feature optimal solutions that can be istantiated into good solutions for the original problem

Space-efficient Streaming and MapReduce Algorithms

- Compute approximate solution to the generalized problem through generalized core-set
- Second pass (Streaming) or third round (MapReduce) to construct the solution with k distinct points
- Space savings: $\Theta(k)$ (Streaming) and $\Theta(\sqrt{k})$ (MapReduce)
- Optimal $O(k)$ streaming space for constant ϵ and D

- Remote-clique problem only.
- Coreset construction (as before)
  - τ-center clustering, with $\tau = O\left(k(c/\epsilon)^D\right)$.
  - for each cluster: select suitable set of delegates

Delegates for partition matroid

For each cluster select an independent set of size $\leq k$

Delegates for transversal matroid

For each cluster select:

- Independent set of size k (if exists)
- Oterwise, up to k points for each category of the matroid

Remark: generalized to arbitrary matroids at the expense of larger space

Experiments

Cardinality constraints

- Synthetic data: Euclidean spaces
- Real data: *musiXmatch* dataset
  - $\approx 250K$ songs.
  - bag-of-words model, cosine distance

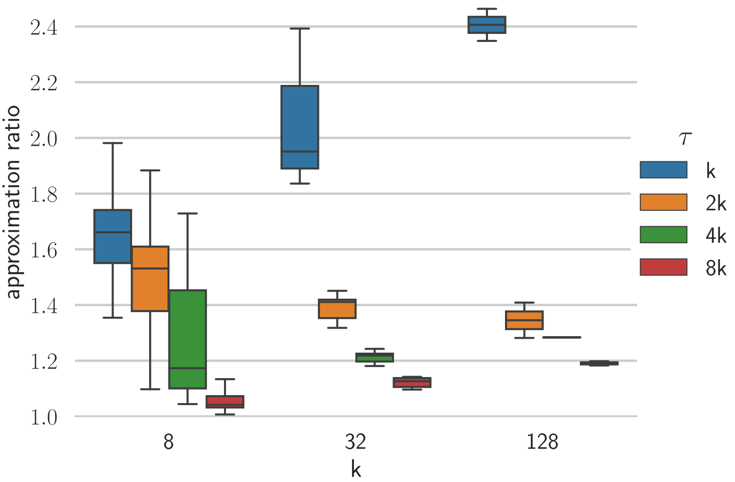
Matroid constraints

- Wikipedia dump:
  - $\approx$ 5M pages
  - partition matroid of rank 100
- MetroLyrics:
  - $\approx$ 264K songs
  - transversal matroid of rank 89

Platform:

- 16-node cluster (Intel i7)
- 18GB-RAM/256GB-SSD per node
- 10Gbps ethernet
- Apache Spark (open source code: `github.com/Cecca/diversity-maximization`)

Streaming algorithm on musiXmatch dataset



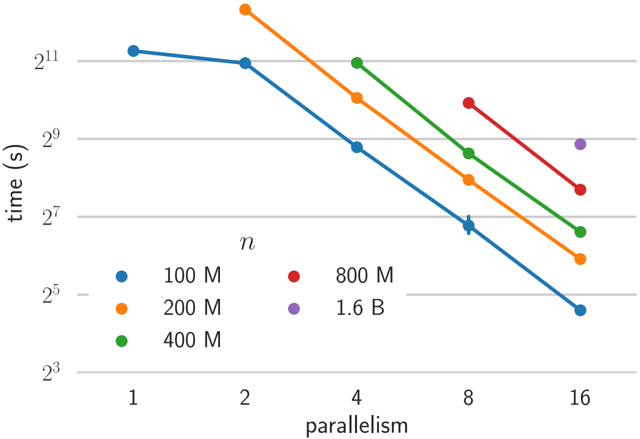
Our algorithm (CCPU) vs. [Aghamolaei et al.'15] (AFZ)

- MapReduce using 16 machines
- remote-clique measure
- $4M$ points in $\mathbb{R}^3$ (max feasible for AFZ)
- Our algorithm: $\tau = 128$

	approximation		time (s)	
k	AFZ	CPPU	AFZ	CPPU
4	1.023	1.012	807.79	1.19
6	1.052	1.018	1,052.39	1.29
8	1.029	1.028	4,625.46	1.12

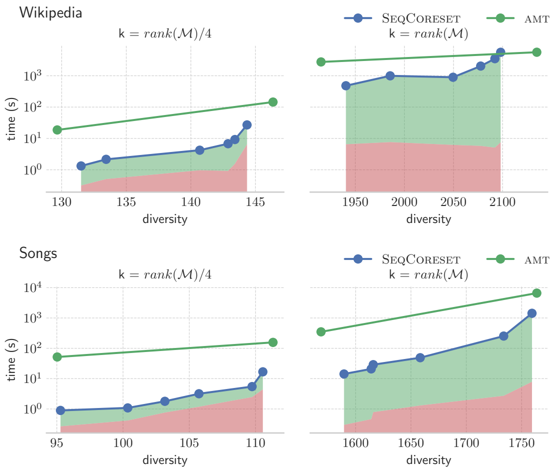
Experiments: scalability (cardinality constraint)

- Synthetic data: points from $\mathbb{R}^3$
- 1 processor $\equiv$ streaming algorithm
- $\tau = 2048$

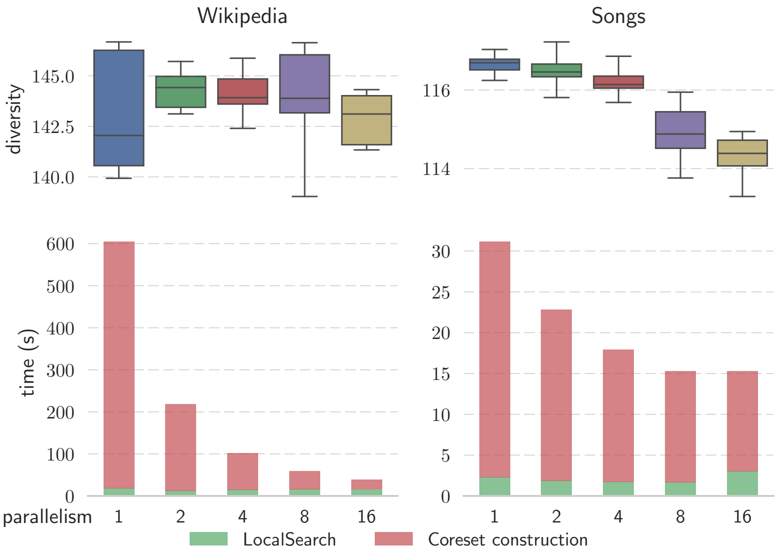


Experiments: comparison with previous work (matroid constraint)

Sequential implementation of the algorithm vs. state of the art local search [Abbassi et al '13]



Experiments: MR scalability and quality (matroid constraints)



Conclusions

Conclusions

- $(1+\epsilon)$-(composable) core-set construction on metric spaces of constant doubling dimension
- Space savings with additional rounds/passes
- Experiments on real and synthetic data demonstrate effectiveness, efficiency and scalability of our approach
- Open problems: improved space requirements (e.g., get rid of exponential dependency on D); better space/round tradeoffs in MapReduce

THANK YOU!