Coreset-based Machine Learning for Distributed Analytics: An Empirical Study More than

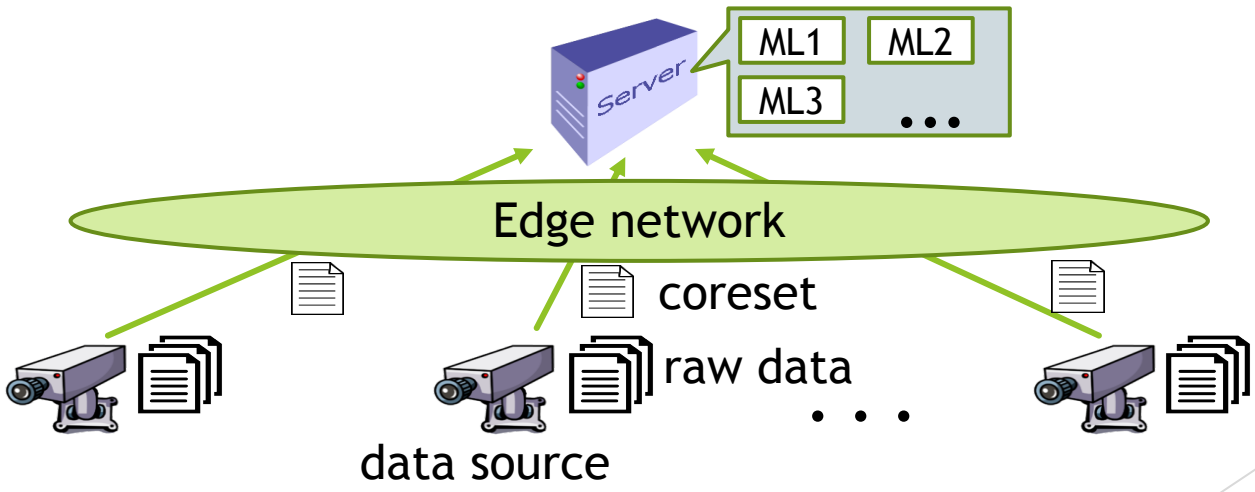
Presented by Hanlin Lu

joint work with Ming-Ju Li (PSU), Ting He (PSU), Shiqiang Wang (IBM), Vijay Narayanan (PSU), Kevin Chan (ARL)

9/12/2018

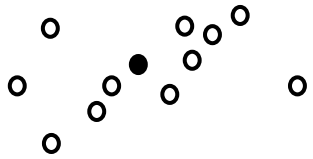
Motivation

- Consider solving a complicated machine learning problem over a large dataset
  - high complexity even if data is at a central location
  - More complex if data is distributed and/or streamed
- *Idea: use a small set (coreset) to replace the raw dataset*
- *Application to distributed analytics:*

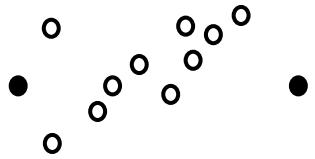


What is “coreset”

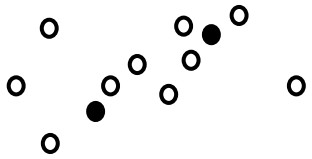
- Given a set *P* of *n* data points, a *coreset* is a *small*, *weighted* set *D* of data points which approximate P in fitting a machine learning model.



a good coreset for 1-clustering



for minimum enclosing ball



for linear regression

More precisely,

- For a set of candidate queries *X*, and a measure function *cost(P, x) (x ∈ X)*, the set *D* is an *ε-coreset* for *P* if *cost(D, x)* approximates *cost(P, x)* up to a multiplicative factor of 1 ± ε for every *x* ∈ *X*, i.e.,

$$(1-\varepsilon)cost(P,x) \leq \sum_{p\in D} w(p)cost(p,x) \leq (1+\varepsilon)cost(P,x)$$

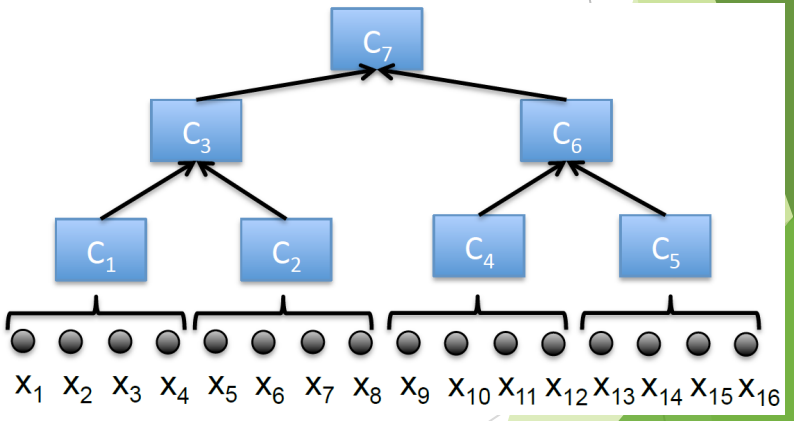
- Advantage over other data summaries: Simply a smaller version of the “same dataset”

History of coreset construction algorithms

- Gradient descent algorithms:
  - Idea: iteratively adding "outliers" to the coreset, until spanning all data points
  - Applications: Minimum Enclosing Ball (MEB), k-center [STOC'02], Support Vector Machine (SVM) [JMLR'05]
- Geometric decomposition algorithms:
  - Idea: partition space into cells, one coreset point per cell
  - Applications: Weighted Facilities [FOCS'06], k-median [STOC'04], k-means [SDM'16], Euclidean graph problems [STOC'05]
- Random sampling algorithms:
  - Idea: randomly sampling from original dataset (sensitivity sampling)
  - Applications: numerical integration [SODA'10], projective clustering [STOC'11], dictionary learning [JMIV'13], dependency networks [AAAI'18]

Pros and cons of coreset

- Advantage: The size of coreset is typically small.
  1. Independent of n (cardinality) and d (dimension) for k-means, PCA, least-squares regression
  2. Low communication overhead, small memory consumption, fast computation
- Applicable to streaming/distributed setting:
  - "merge-and-reduce" framework [NIPS'11]

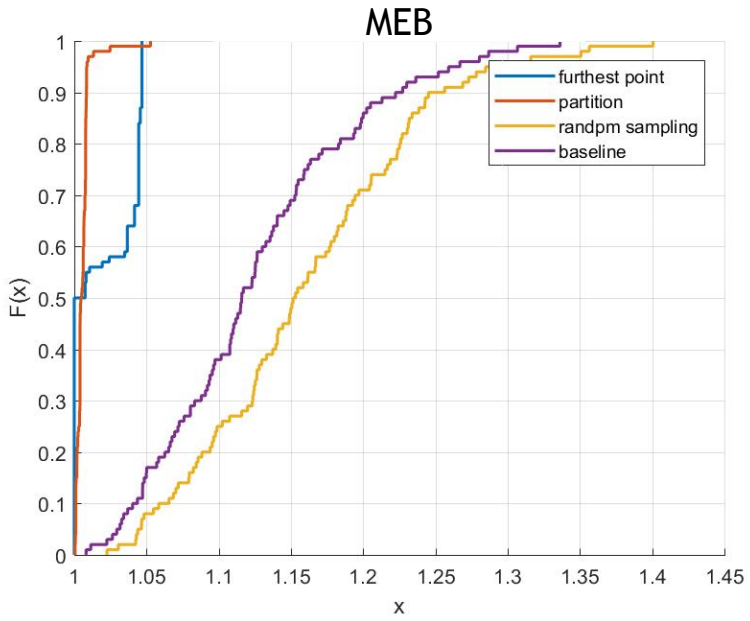


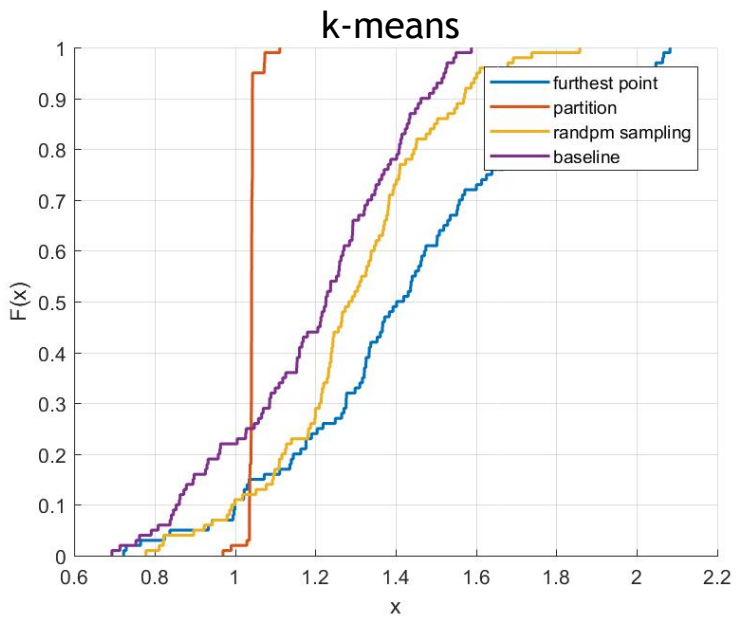
- Limitations:
  - Coresets are each tailor-made to support one machine learning problem.

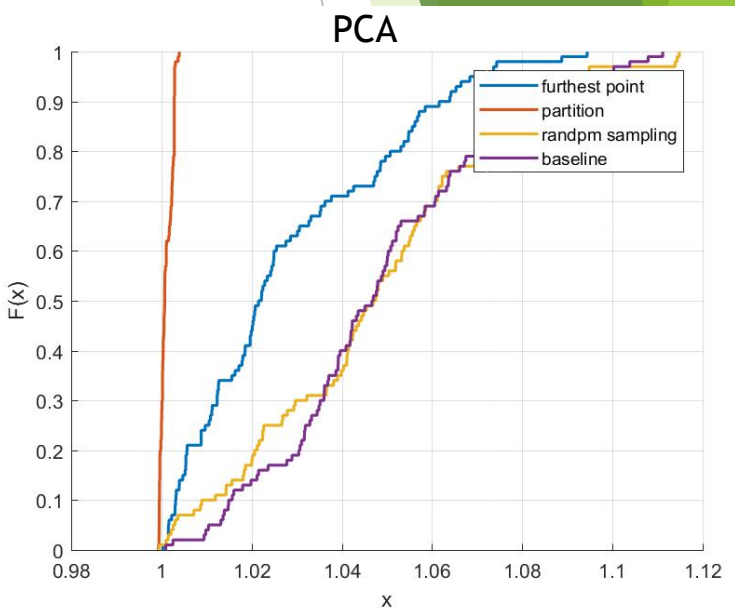
Q: Is there a *universally good* coreset construction algorithm?

Empirical study

- Representative algorithm from each category wrt diverse ML problems
- 4000 points uniformly drawn from $[1, 50]^3$
- Plot CDF of normalized cost ($\text{cost}(c_D, P)$ /$\text{cost}(c^*, P)$) over runs of coreset







- "partition" (k-mean-coreset [SDM'16]) wins!

Q: Coincidence or fundamental?

Theory

k-clustering: argmin_Q c(P, Q) := $argmin_Q \sum_{p \in P} w(p)(min_{q \in Q} dist(p,q))^z$

opt(P,k): optimal cost of k-clustering

- **Theorem 1**. If $opt(P,k) - opt(P,2k) \leq w_{min} \left(\frac{\epsilon}{\rho}\right)^z$, then optimal k-clustering of P gives an ϵ-coreset for P wrt any ML problem with:
  - Overall cost: cost(P,x) = $\sum_{p \in P} w(p) cost(p,x)$ or $max_{p \in P}\ cost(p,x)$
  - Per-point cost: cost(p,x) ≥ 1 and is ρ-Lipschitz-continuous in p ∀x∈X

- Example: ρ=1 for MEB and k-median, ρ=2Δ for k-means (Δ: diameter of sample space)

Theory cont'd

k-clustering is NP-hard → use a suboptimal k-clustering algorithm

approx(P,k): cost of the k-clustering algorithm

- **Theorem 2**. If $\text{approx}(P,k) - approx(P,2k) \leq w_{min}\left(\frac{\epsilon}{\rho}\right)^z$, then the k-clustering algorithm gives an ϵ-coreset for P wrt any ML satisfying conditions in Theorem 1, if the algorithm satisfies:
  - optimal for k=1 (local optimality)
  - $approx(P,2k) \leq \sum_{i=1}^{k} approx(\widetilde{P}_i, 2)$ (self-consistency)
  - $approx(P,2) \leq c(P, \{\mu(P), p^*\})$ (greedy dominance)

$\{\widetilde{P}_i\}_{i=1}^{k}$: partition by the k-clustering algorithm

$\mu(P)$: 1-clustering center

p^*: point with highest 1-clustering cost

Algorithm

Algorithm 1: Universally Good Coreset(P, ϵ, ρ)

input : A weighted set P with minimum weight $w_{\min}$, approximation error $\epsilon > 0$, Lipschitz constant ρ

output: An ϵ-coreset S for P w.r.t. a cost function satisfying Theorem 3.1

1 **foreach** $k = 1, \ldots, |P|$ **do**

2 | **if** $approx(P, k) - approx(P, 2k) \leq w_{\min}(\frac{\epsilon}{\rho})^z$ **then**

3 | | break;

4 $(\{\mu(\widetilde{P}_i)\}_{i=1}^k, \{\widetilde{P}_i\}_{i=1}^k) \leftarrow k$-clustering$(P, k)$;

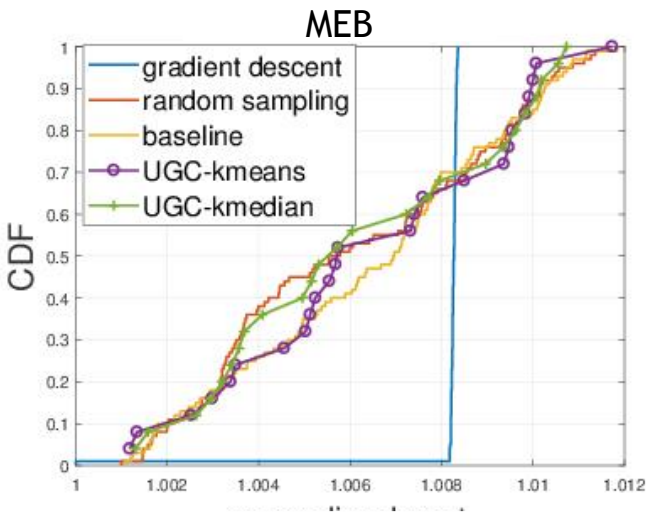
5 $S \leftarrow \{\mu(\widetilde{P}_i)\}_{i=1}^k$, where $\mu(\widetilde{P}_i)$ has weight $\sum_{p \in \widetilde{P}_i} w(p)$;

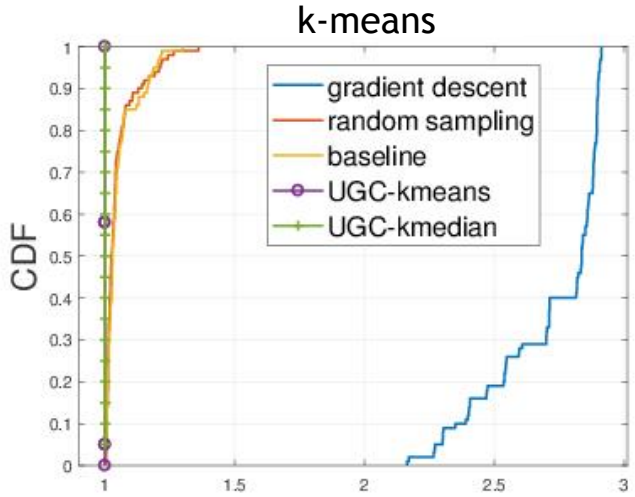
6 return S;

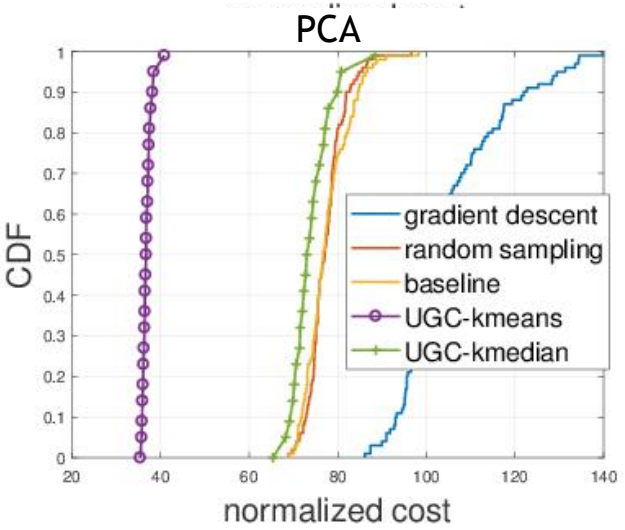
- If coreset size is predetermined, start from line 4

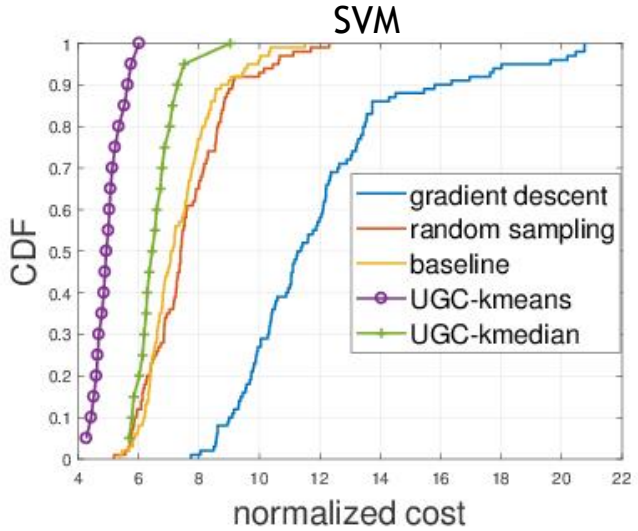
Evaluation

- A trimmed version of MNIST data: 60, 000 images (20 * 20 each) of handwritten digits.









Algorithms based on k-clustering (*partition*, *kmeans*, *kmedian*) are always among the best

References

- [STOC'11] *A Unified Framework for Approximating and Clustering Data*. D. Feldman, M. Langberg STOC' 2011
- [JMIV'13] *Learning Big (Image) Data via Coresets for Dictionaries*. Dan Feldman, Micha Feigin, Nir Sochen. JMIV' 2013 Journal of Mathematical Imaging and Vision
- [SDM'16] *k-Means for Streaming and Distributed Big Sparse Data*. Artem Basrger and Dan Feldman, SDM 2016
- [STOC'02] *Approximate Clustering via Core-Sets*. Mihai Bādoiu, Sariel Har-Peled, Piotr Indyk. STOC'02
- [JMLR'05] *Core Vector Machines: Fast SVM Training on Very Large Data Sets*. Ivor W. Tsang, James T. Kwok, Pak-Ming Cheung. JMLR'05
- [FOCS'06] *Coresets for Weighted Facilities and Their Applications*. Dan Feldman, Amos Fiat, Micha Sharir . FOCS'06

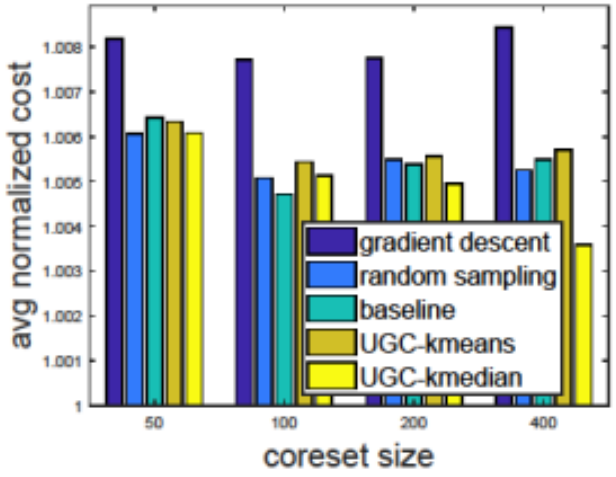
References

- [STOC'05*] Coresets in Dynamic Geometric Data Streams*. Gereon Frahling, Christian Sohler. STOC'05
- [NIPS'11] *Scalable Training of Mixture Models via Coresets*. Dan Feldman, Mikhail Volkov, Andreas Krause. NIPS' 2011
- [STOC'04] *On coresets for k-means and k-median clustering*. Sariel Har-Peled, Soham Mazumdar. STOC'04
- [SODA'10] Universal ϵ approximators for integrals. Langberg, M., and Schulman, L. J. SODA'10
- [AAAI'18] *Core dependency networks*. Molina, A., Munteanu, A., and Kersting, K.

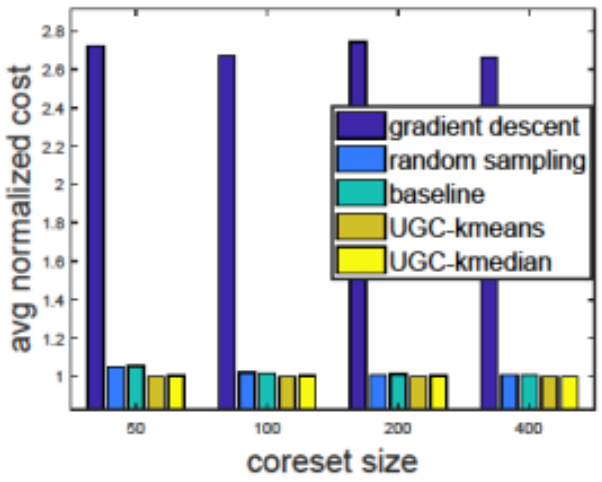
Evaluation

- A trimmed version of MNIST data: 60,000 images (20 * 20 each) of handwritten digits.

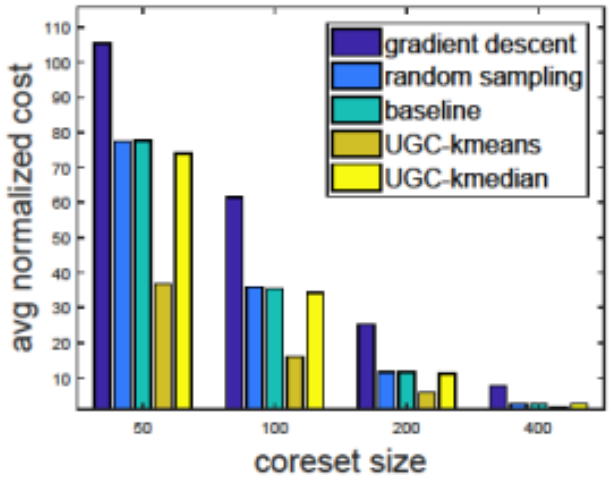
MEB



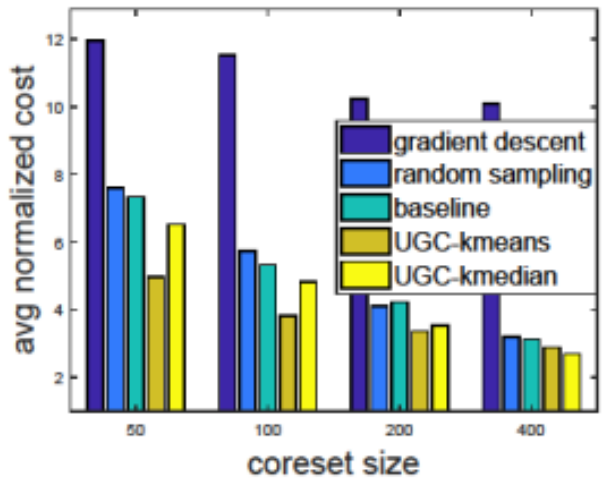
k-means



PCA



SVM



Algorithms based on k-clustering (UGC-*kmeans*/*kmedian*) are always among the best