Coreset-based Machine Learning for Distributed Analytics

Hanlin Lu (PSU), **Ming-Ju Li** (PSU), **Ting He** (PSU), **Shiqiang Wang** (IBM US), **Vijay Narayanan** (PSU), **Kevin Chan** (ARL)

Motivation

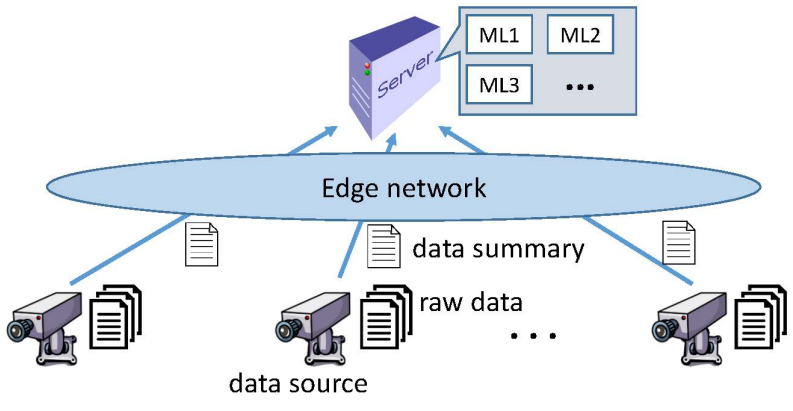

Figure 1: Example of an application scenario

Goal: Reduce the communication overhead in learning *multiple machine learning models* over distributed data

Approach: Learn on *coreset*, which is a "proxy" of original dataset, but much smaller.

Categories of coreset construction algorithms: *gradient descent*, *random sampling, geometric decomposition*

Challenge: Existing algorithms are tailor-made for specific ML problems → Need many coresets to summarize one dataset!

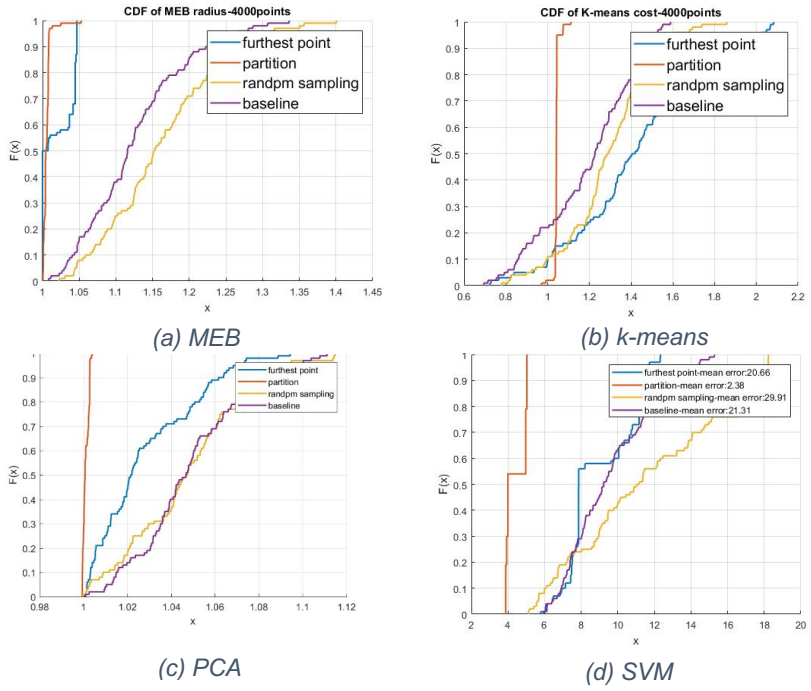

Figure 2: CDF of normalized costs for models learned on different coresets

Q: Does "partition" algorithm give a universally good coreset?

Solution

Main theorem

If $opt(P,k) - opt(P,2k) \leq w_{min}(\varepsilon/\rho)^z$, then the optimal k-clustering of dataset *P* gives an ε-coreset for *P* w.r.t. both the sum cost and the maximum cost for per-point cost function satisfying (1) $cost(p,x) \geq 1$, (2) $cost(p,x)$ is ρ-Lipschitz-continuous in data sample p, $\forall x \in X$.

Algorithm

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

input : A weighted set P with minimum weight $w_{\min}$, approximation error $\epsilon > 0$, continuity parameter ρ
output: An ϵ-coreset S for P w.r.t. a cost function satisfying Theorem III.1

foreach $k = 1, 2, \ldots$ **do**
  if $approx(P,k) - approx(P,2k) \leq w_{\min}(\frac{\epsilon}{\rho})^z$ **then**
    break;
$(\{\mu(\widetilde{P}_i)\}_{i=1}^k, \{\widetilde{P}_i\}_{i=1}^k) \leftarrow k$-clustering$(P,k)$;
$S \leftarrow \{\mu(\widetilde{P}_i)\}_{i=1}^k$, where $\mu(\widetilde{P}_i)$ has weight $\sum_{p \in \widetilde{P}_i} w(p)$;
return S;

Evaluation

Clustering-based coresets ("partition", "kmeans", "kmedian") are "universally good".

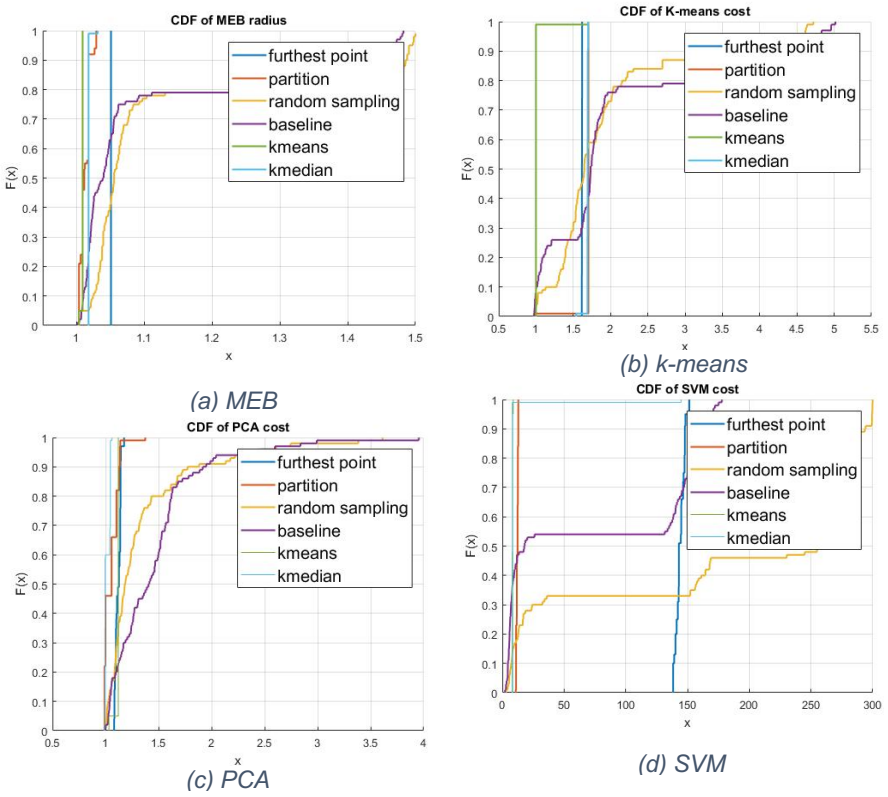

Figure 3: CDF of normalized costs for models learned on Fisher's iris data.

Take-away: Coreset construction → k-clustering (can apply existing distributed clustering alg.)