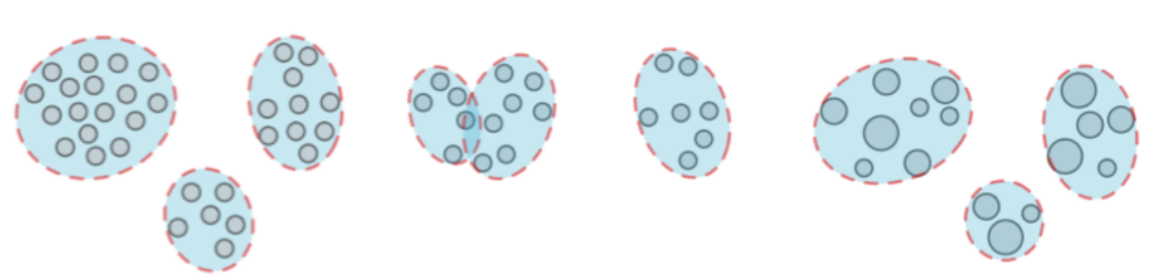
Communication efficient coreset sampling for distributed learning

Yawen Fan, Husheng Li, The University of Tennessee, Knoxville

Motivation

- Modern machine learning problem has large scale and requires distributed setting for storage and computation.
- Communication between the distributed computation unit becomes the bottleneck of the system's performance.
- Consider the trade-off between accuracy, computation and communication in distributed learning framework.

Exchange Information via Sampling

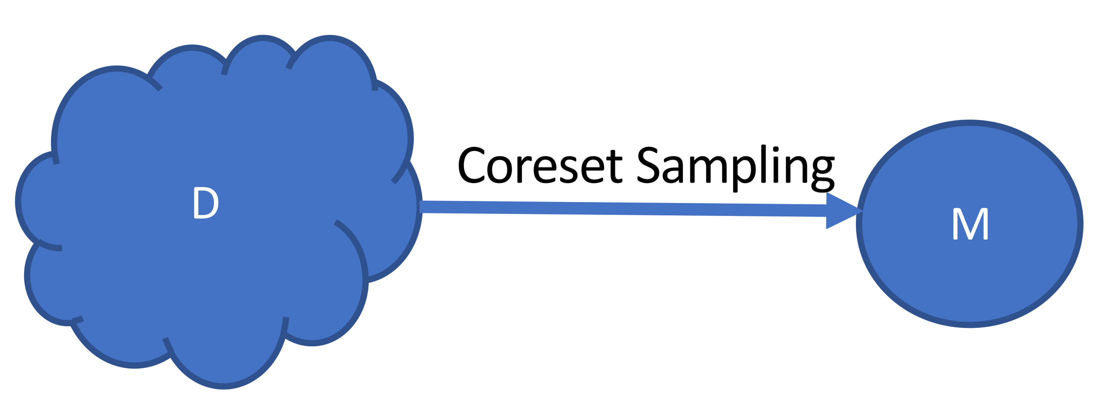


(a) Data set (b) Random Sampling (c) Coreset

- Given the communication constraint, instead of learning based on the whole data set, we prefer to sample a subset of data for learning.
- Random sampling may fail when the size of the subset is small.

Coreset

Coreset is a subset of data with small size and could be considered as a good approximation of the original data set.



$$E_D[L(f(x))] = \sum_{i=1}^{|D|} \frac{1}{D} L(f(x_i)) \qquad E_M[L(f(x))] = \sum_{i=1}^{|M|} w_i L(f(x_i))$$

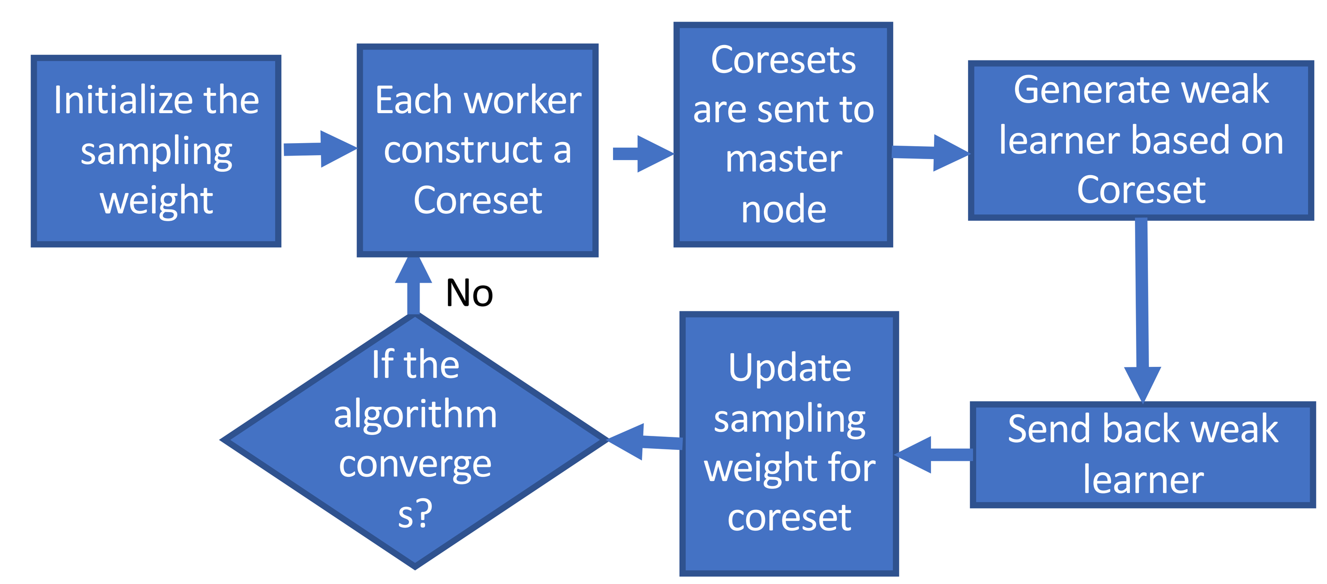
Definition

- A family of target function $f \in F$, Loss function L and Data set D
- With probability $1-\beta$ and $\forall f \in F$, if

$$|E_M[L(f(x))] - E_D[L(f(x))]| \leq \epsilon E_D[L(f(x))]$$

- Then we call M is the ***Coreset*** of D

Distributed Coreset Boosting



- Instead of sending the whole local data set to the master, each worker node selectively sends the Coreset.
- Comparing to random sampling, the master node could learn a better classifier based on coreset.

The sensitivity of the sample

- For each sample $z_i \in D$, we define its sensitivity as

$$\phi_i(\mathcal{F}, L) = \sup_{f \in \mathcal{F}} \frac{L(f(z_i))}{\sum_{j=1}^{|D|} L(f(z_j))}$$

- The sensitivity is large only if there exists at least one function $f \in$ F, such that

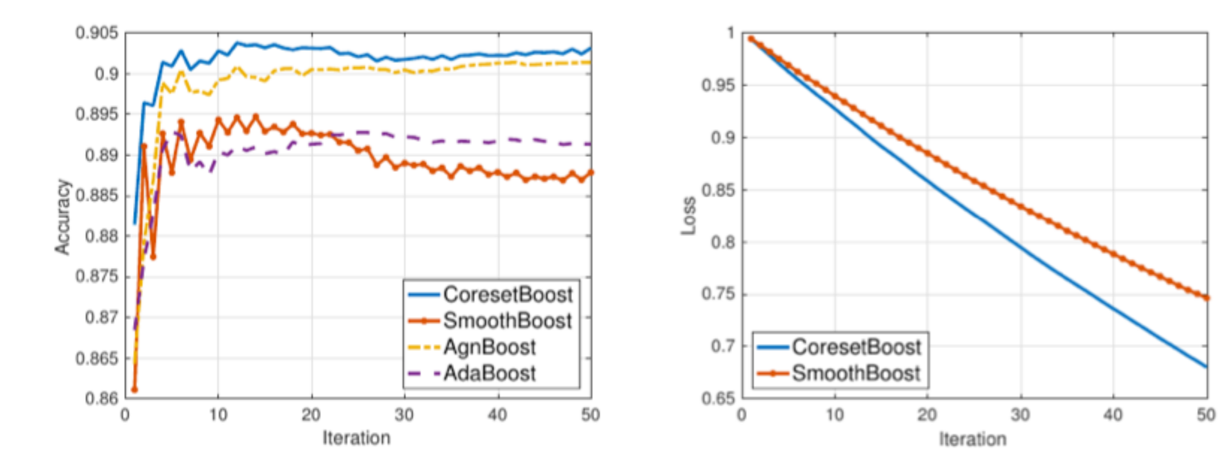
$$L(f(z_i)) \gg L(f(z_j)), \quad \forall j \neq i$$

- Large sensitivity indicates that for the given function family F, the sample z_i has larger loss than any other sample in the data set.

Main Theorem

Suppose the feature X is scaled to [0,1]. Assume $h(X)$ is η-bounded and the empirical loss for $h^t(x)$ satisfies $\hat{L}_{sm}^M(h^t) \leq (1+\beta)(1-\alpha)$, then with probability $1-\delta$, the output of proposed algorithm could achieve error rate $\min_{h \in H} Err(h) + \epsilon$ and converges in $O(\frac{1}{\epsilon^{2-2c}})$ iterations.

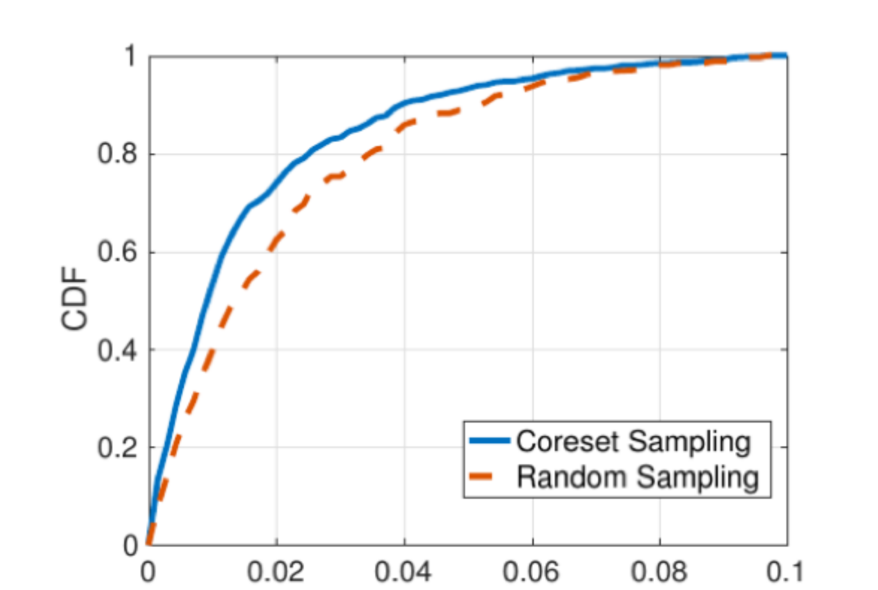
Result



- AdaBoost (Freund, Yoav. "An adaptive version of the boost by majority algorithm." Machine learning 43.3 (2001): 293-318.)
- SmoothBoost (Servedio, Rocco A. "Smooth boosting and learning with malicious noise." Journal of Machine Learning Research 4.Sep (2003): 633-648.)
- AgnBoost (Chen, Shang-Tse, Maria-Florina Balcan, and Duen Horng Chau. "Communication efficient distributed agnostic boosting." Artificial Intelligence and Statistics. 2016.)

Data set	WebSpam	CovType	Yahoo!
SmoothBoost Coreset Sampling			
Acc_{tr} %	**91.54 (0.2)**	**75.45 (0.4)**	**62.90 (0.3)**
Acc_{te} %	**90.19 (0.3)**	**75.15 (0.3)**	**62.35 (0.2)**
Time	104.1s	200.9s	1200.3 s
Clustering	14.1s	30.2s	198.3 s
Boosting	90 s	170.7s	1002.0 s
SmoothBoost Random Sampling			
$Accu_{tr}$ %	89.49 (0.2)	73.06 (0.3)	60.11 (0.4)
$Accu_{te}$ %	88.75 (0.2)	72.90 (0.5)	60.01 (0.2)
Time	**82.1s**	84.1s	903.5s
AgnosticBoost with Subset			
$Accu_{tr}$ %	90.16 (0.2)	74.32 (0.2)	61.14 (0.5)
$Accu_{te}$ %	90.00 (0.1)	73.09 (0.4)	61.01 (0.4)
Time	93.1s	210.1s	1223.5s
AdaBoost with Random Sampling			
$Accu_{tr}$ %	89.38 (0.1)	73.32 (0.3)	59.14 (0.5)
$Accu_{te}$ %	88.97 (0.1)	71.09 (0.4)	58.11 (0.4)
Time	84.1s	**75.3**s	**870.2**s

Accuracy



Approximation Quality