Distributed Learning of Finite Gaussian Mixtures

Qiong Zhang, Renmin University of China

Jiahua Chen, University of British Columbia

Background

Finite Gaussian mixtures

- A probabilistic model where there are finitely many Gaussian subpopulations in the entire population but the observed data have no direct information about which subpopulations they came from.
- Density function of a Gaussian mixture of order K

$$\phi_G(x) = \int \phi(x;\theta)\, dG(\theta) = \sum_{k=1}^{K} w_k \delta_{\theta_k}$$

- Parameter space

$$\mathbb{G}_K = \left\{ G = \sum_{k=1}^{K} w_k \delta_{\theta_k} : w_k \geq 0, \sum_{k=1}^{K} w_k = 1 \right\}$$

Split-and-conquer learning

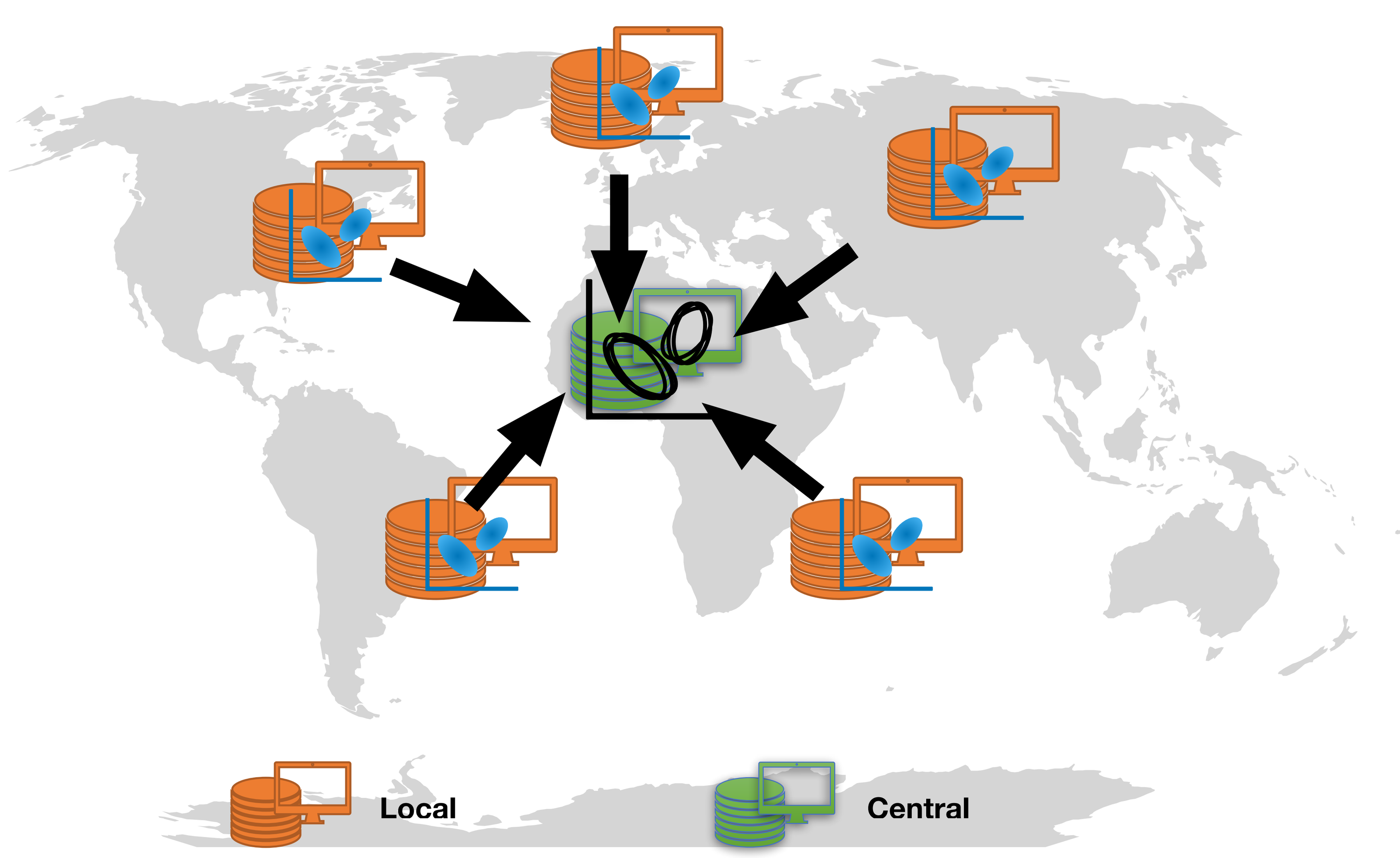


Figure 1. Illustration of distributed data and split-and-conquer approach. Distributed data storage: datasets are too large to be stored on a single facility or collected by different agencies and cannot be shared due to privacy. The split-and-conquer approach is widely used for learning under distributed data storage. It consists of the following 2 steps: 1) local inference: statistical inference on local machine and 2) aggregation: combine local results on a central machine. The most widely used aggregation approach is the linear average of local results. Split-and-conquer only requires one round of communication of summary statistics.

Challenges of aggregation under finite Gaussian mixtures

- When the parameter space is Euclidean, aggregate via linear average

$$\bar{\theta} = M^{-1} \sum_{m=1}^{M} \hat{\theta}_m$$

where M is the # of local machines and $\hat{\theta}_m$ is the local estimate on the mth machine
- Under mixture model
  - The simple average $\bar{G} = M^{-1} \sum_m \hat{G}_m \notin \mathbb{G}_K$ is **NOT** in the parameter space
  - From mixture point of view: $\phi_{\bar{G}}(x)$ is a good estimate for $\phi_{G^*}(x)$
  - We could find an approximation to $\bar{G}$ from the desired parameter space

Proposed Aggregation Method

Overview of the method

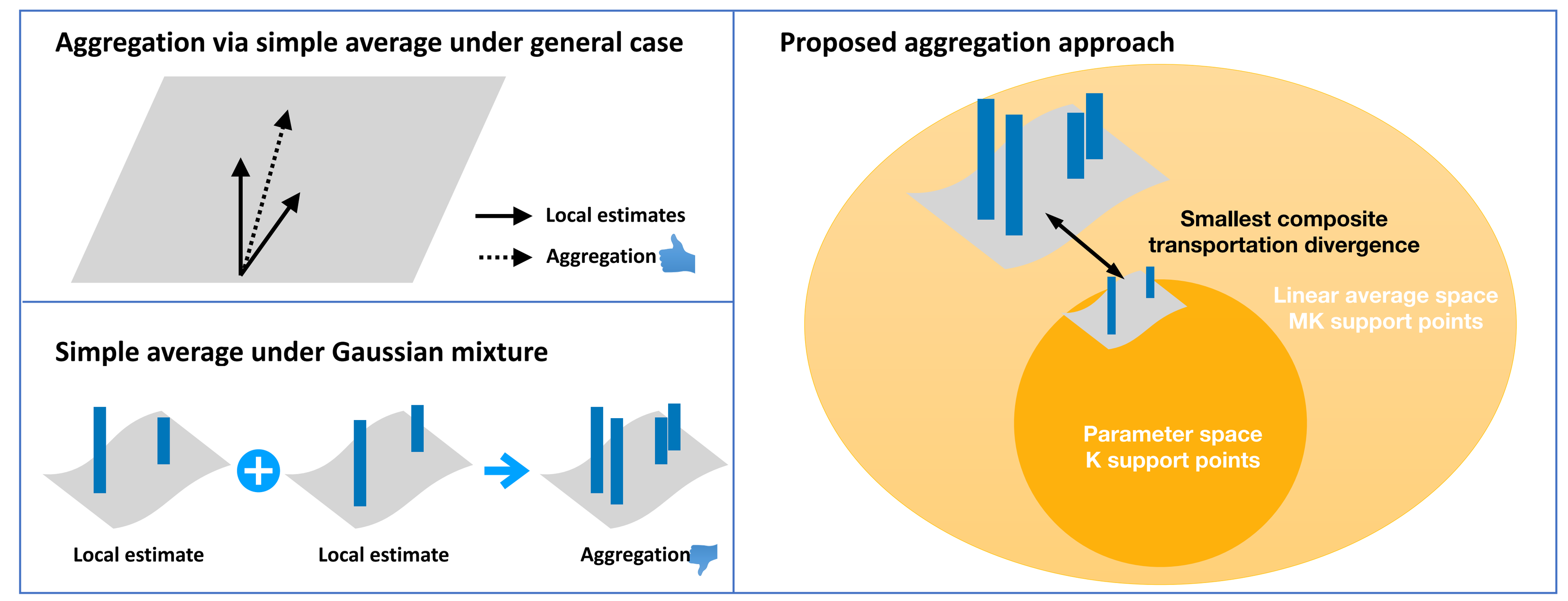


Figure 2. Illustration of the proposed aggregation method. When the parameter space of a model is a vector space, the local estimates are usually aggregated via their linear average. Under mixture models, the linear average cannot be directly used since the parameter space consists of discrete distributions with fixed number of support point and the linear average no longer belongs to this space. We instead propose to search for a parameter in the desired space that minimizes the composite transportation divergence to the mixing distribution obtained via linear average.

Composite transportation divergence

The composite transportation divergence between two mixtures $\phi_G(x) = \sum_n w_n \phi(x;\theta_n)$ and $\phi_{G'}(x) = \sum_m w'_m \phi(x;\theta'_m)$ is

$$\mathcal{T}_c(\phi_G, \phi_{G'}) = \left\{ \sum_{n,m} \pi_{nm} c(\phi(\,\cdot\,;\theta_n), \phi(\,\cdot\,;\theta'_m)) : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = w'_m \right\}$$

The proposed aggregated estimator GMR is

$$\bar{G}^R = \mathrm{argmin}_{G \in \mathbb{G}_K} \mathcal{T}_c(\phi_{\bar{G}}, \phi_G)$$

Majorization-minimization algorithm

Equivalent optimization problem: let

$$\mathcal{J}_c(\phi_{\bar{G}}, \phi_G) = \left\{ \sum_{n,m} \pi_{nm} c(\phi(\,\cdot\,;\bar{\theta}_n), \phi(\,\cdot\,;\theta_m)) : \sum_m \pi_{nm} = \bar{w}_n \right\}$$

then we show that

$$\bar{G}^R = \mathrm{argmin}_{G \in \mathbb{G}_K} \mathcal{J}_c(\phi_{\bar{G}}, \phi_G) \quad \bar{w}_k^R = \sum_n \pi_{nk}(\bar{G}^R)$$

where $\pi(G) = \mathrm{argmin}\, \mathcal{J}(\phi_{\bar{G}}, \phi_G)$

Majorization function

$$\mathcal{K}(G \,|\, G^{(t)}) = \sum_{n,k} \pi_{nm}(G^{(t)}) c(\phi(\,\cdot\,;\bar{\theta}_n), \phi(\,\cdot\,;\theta_k))$$

Statistical properties

C1 The data are IID observations from ϕ_{G*} with order K.

C5 **Local triangular inequality** $A^{-1}\|\phi_1 - \phi_2\|_2^2 \leq c(\phi_1, \phi_2) \leq A\|\phi_1 - \phi_2\|_2^2$

Under conditions C1-C5, up to permutations, we have

$$\bar{\phi}^R - \phi_k^* = O(N^{-1/2}),\ \bar{w}^R - w_k^* = O(N^{-1/2})$$

Experiments

Methods for comparison

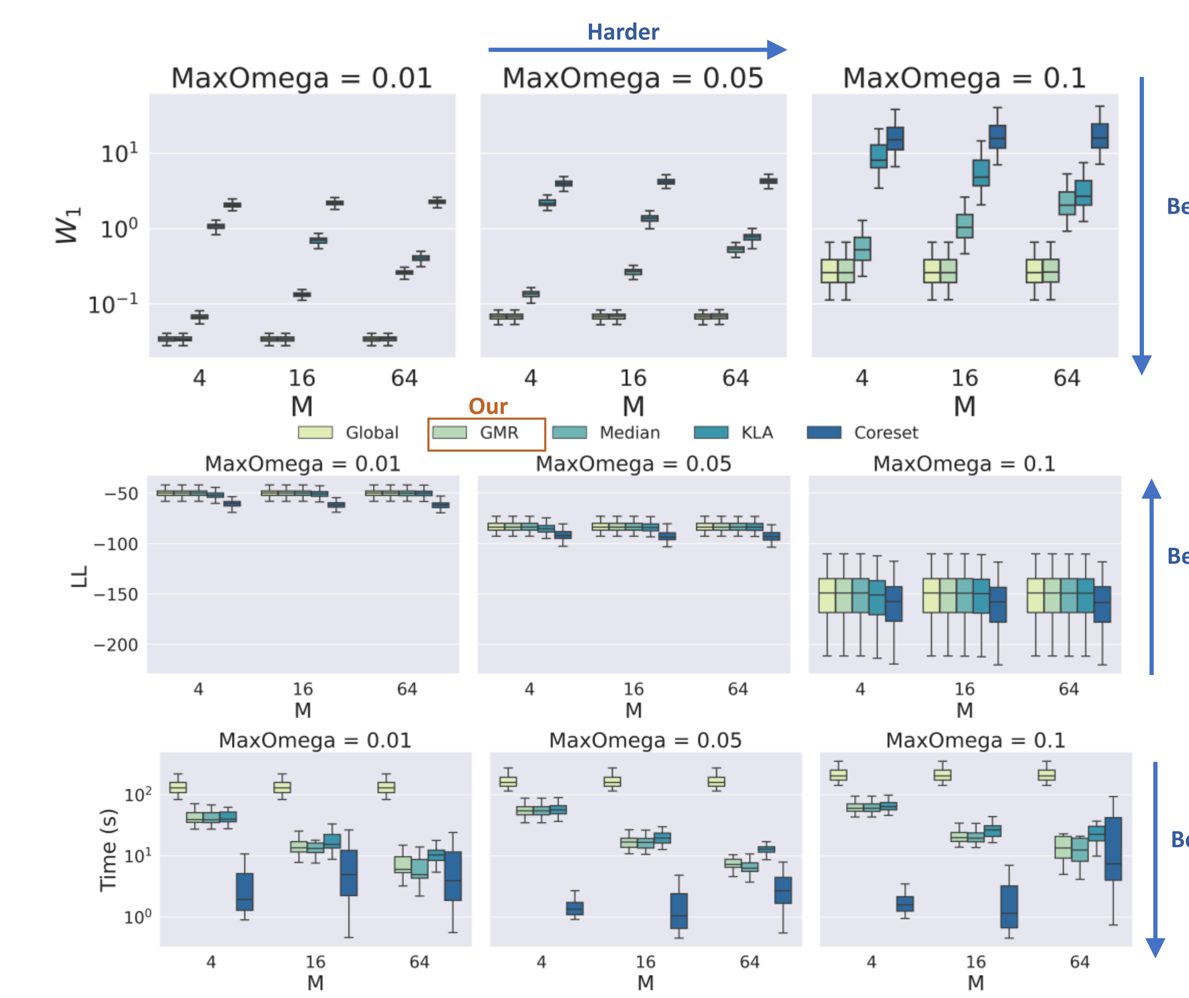
- Global: estimator based on the full dataset (ideal case)
- GMR: our proposed estimator with KL divergence as cost function
- Median: the "best" local estimator
- KLA: the aggregation approach in [1]
- Coresets: the method learns a coreset locally and combine the coresets for learning

Performance criterion

- W_1: Wasserstein distance between the estimator and truth (the lower the better)
- LL: per observation log-likelihood value (the higher the better)
- Time: computational time (the lower the better)

Simulated Dataset

- Total sample size $N = 2^{21}$, number of local machines $M = 4,16,64$



Benchmark Dataset

Table 1. The per observation log-likelihood (LL) value of different learning approaches on benchmark datasets. Our proposed method GMR has comparable LL value (the higher the better) to the global estimator and outperforms other split-and-conquer based existing methods. GMR has much shorter computational time than the global estimator.

Dataset	N	d	K	M	Global	GMR	Median	KLA	Coreset
Median (IQR) LL values (the larger the better)									
MIGIC04	19020	10	10	4	-24.15	-24.30(0.07)	-26.60(0.05)	-26.73(0.07)	-27.16(0.55)
MiniBooNE	130065	50	10	4	-19.46	-22.00(0.53)	-24.60(0.32)	$-6.41(1.95) \times 10^3$	$-8.6(2.56) \times 10^9$
KDD	145751	74	10	4	-221.80	-223.25(0.42)	-232.93(8.02)	-235.00(8.96)	-374.43(193.58)
MSYP	515345	25	50	16	-166.56	-167.05(0.04)	-171.10(0.04)	-170.72(0.01)	-181.64(1.78)
Median (IQR) computation times in seconds									
MIGIC04	19020	10	10	4	19.3	7.0(3.2)	6.7(3.2)	10.2(3.1)	2.2(0.6)
MiniBooNE	130065	50	10	4	346.9	313.1(162.6)	313.2(162.6)	511.3(213.2)	26.6(64.3)
KDD	145751	74	10	4	1033.9	544.4(309.5)	543.0(310.0)	706.0(290.3)	4.3(64.0)
MSYP	515345	25	50	16	67048.8	2611.6(474.0)	1777.5(511.2)	5515.9(1629.7)	67.4(12.6)

References

Liu et al. (2014). "Distributed estimation, information loss and exponential families." In: 2014 Advances in Neural Information Processing Systems 27, pp. 1098-1106.

Lucic et al. (2017). "Training Gaussian mixture models at scale via coresets." In: The Journal of Machine Learning Research, 18(1), pp. 5885-5909.