Decentralized clustering

Amira Soliman

BIDAF

Eliciting Structure in Data
Why Decentralization

◎ The world is getting more **centralized**
  ➢ Our **data** it’s mostly produced by many **ubiquitous and autonomous** individual sources.

◎ We need decentralized ML & AI with data governance, security and privacy assurances **by design**.
Machine Learning @ Scale

(1) Distributed Learning

1. IID data (no bias in splitting the data)
2. Scalability issues (central authority)

(2) Federated Learning

This new model has learned directly from our users' data without centralizing any of it!

Eliciting Structure in Data
Decentralized Data and Decentralized Learning (D³L)

**Gossip Learning**: Nodes learn locally, update their models upon contact with each other.

**How to perform the update?**

- Existing techniques assume random overlay, **homogeneous** nodes and (nearly) **synchronous** environment.
- Combining models using weighted averaging.
Decentralized Data and Principles of Non-IIDness Learning

Non-IIDness of decentralized learning

- Heterogeneity among data sources,
- Individual view is incomplete, biased, or even misleading

Concepts of non-IIDness learning


- Heterogeneity between parts,
- Coupling between local views:
  - Reasoning
  - Inference
- Local to global view
Decentralized Clustering

- **No central authority**: nodes exchange their incremental updates in P2P manner,
- **Non-IID data**: no guarantees on data distribution,
- **Asynchronous learning**: nodes communicate in different rates.
Decentralized and Adaptive $K$-Means Clustering for Non-IID Data using HyperLogLog Counters

Collaboration Between RISE and Halmstad University

THE 24TH PACIFIC-ASIA CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (PAKDD2020)
Probabilistic Data Structures : HyperLogLog (HLL)

\[ x: \begin{array}{cccccccc}
1234 & 66 & 1234 & 88 & 66 & 900 & 555 & 88 & 555 & 88 \\
\end{array} \]

\[ \text{hash}(x): \begin{array}{cccccccc}
0.2 & 0.8 & 0.2 & 0.4 & 0.8 & 0.3 & 0.7 & 0.4 & 0.7 & 0.4 \\
\end{array} \]

Bit-map of HLL

0.2 Estimated cardinality = 1/0.2 = 5

Merging of two HLLs is possible, accordingly:

- We can estimate the number of unique data points among multiple nodes.
Adaptive Decentralized $k$-means with HLL Counters

The steps are as follows:

1. Nodes share random $k$ centroids in the beginning,
2. Nodes update centroids locally,
3. Nodes exchange centroids as well as HLL representation of each cluster,

We developed two approaches for merging the intermediate clustering results computed by the nodes:

a) **Method 1**: nodes compute another HLL that is using min-hash algorithm to merge only overlapped clusters.

b) **Method 2**: nodes apply $K$-means to merge the closest centroids, then apply **BFR** (Bradley-Fayyad-Reina) algorithm to further merge similar clusters.
Results: PAMAP2 - Physical Activity Dataset

![Graphs showing accuracy over rounds for Syn/non-iid and Asyn/non-iid scenarios.]
Decentralized and Adaptive GMM via Merge and Split optimizations
Split and merge EM algorithm for improving GMM

**The basic idea:** after the convergence of the usual EM algorithm, split-and-merge operation are used to update the values of some parameters among all the parameters.

**Split-Merge-Scores:** operations are simultaneously performed so that the number of components remains unchanged.

Decentralized GMM (Split and Merge)

The steps are as the following:
1. Nodes perform the usual EM algorithm till it reach convergence,
2. Nodes update EM components locally via split-merge-EM,
3. Nodes exchange EM components to further merge similar components.

The merge score: merge mostly correlated components w.r.t estimated posterior (i.e. components in an overpopulated region)
The split score: split most spread component (Gamma test)
Summary

Non-IIDness & Decentralized Learning

- Heterogeneity in data.
- Heterogeneity in learning outcomes.
- Coupling relationships among local views.
- Mapping from local to global understanding of the world.

Decentralized Clustering

- Our algorithms efficiently handle non-IID data as well as asynchronous updates.
- Adaptive K-means by integrating HLL counters to enforce fairness among all data points.
- Adaptive GMM via sequence of Merge and Split Optimizations.
Thanks 😊