Consensus-based Distributed Clustering for IoT

Hui Chen[†], Hao Yu[†], Shengjie Zhao, Qingjiang Shi

NaMI Lab, Tongji University

[†] Equal contribution

April 15, 2020

Contents

Emerging IoT devices

- IoT is widely used in industry.
- IoT devices are increasing exponentially.
- Huge data is to be mined. A difficult task.

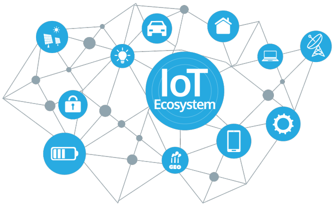


Figure: IoT devices ecosystem

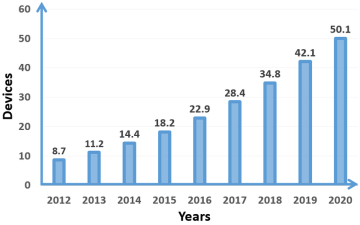


Figure: Number of connected IoT devices (Billion)

Centralized process of data mining from IoT

- Typically, we need to
  1. transmit raw data from all agents to a central device.
  2. upload data to a cloud center.
  3. apply data mining algorithms.
- Challenges
  - Data volume
  - Communication latency
  - Information security
- We need distributed methods to mine IoT data!

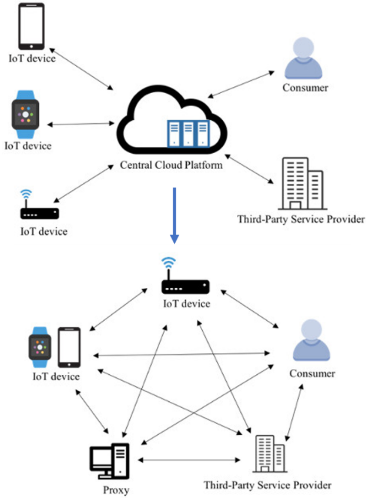


Figure: Centralized IoT system to Distributed IoT system.

Why distributed clustering?

- Clustering analysis is widely used in hidden information mining.
- Most clustering algorithms are cost-efficient so that IoT devices are able to implement them.
- K-means (K-means++) is the most popular and effective algorithm among plentiful clustering methods.

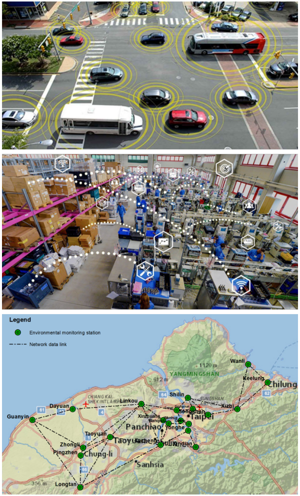
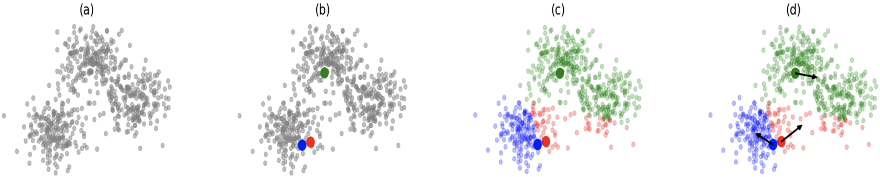


Figure: Clustering analysis in transportation, industry, and environment IoT

k-means algorithm



- Step 1: (b) Initialize the centroids.
- Step 2: (c) Assign each observation to the cluster with the closest centroid.
- Step 3: (d) Update centroids as the average of the corresponding clusters.

Repeat Step 2 and 3 until convergence.

Distributed clustering

- M agents, each with an observation set $\mathcal{X}^{(m)}, m = 1, \cdots, M$.
- To conduct clustering analysis to $\mathcal{X} = \cup_{m=1}^{M} \mathcal{X}^{(m)}$, and to return K centroids $\boldsymbol{c}_k$, $k = 1, \cdots, K$.
- Each agent keeps its own version of centroids $\boldsymbol{c}_k^{(m)}$, $\boldsymbol{c}_k^{(1)} = \boldsymbol{c}_k^{(2)} = \cdots = \boldsymbol{c}_k^{(M)} = \boldsymbol{c}_k$.

Difficulty

How to make all agents agree on the centroids?

Existing methods

$$\min \; F \text{ (e.g. in-cluster error)}$$
$$s.t. \; \boldsymbol{c}_k^{(i)} = \boldsymbol{c}_k^{(j)}$$
$$\text{for } \; k = 1, \cdots, K \text{ and } agents(i,j) \text{ connected}$$

- Pedro A Forero, Alfonso Cano, and Georgios B Giannakis, "Distributed clustering using wireless sensor networks," IEEE Journal of Selected Topics in Signal Processing, 2011
- Soummya Kar and Brian Swenson, "Clustering with distributed data," arXiv preprint arXiv:1901.00214, 2019

Existing methods

$$\min \;\; F + \lambda \cdot dist(\boldsymbol{c}_k^{(i)}, \boldsymbol{c}_k^{(j)})$$
$$\text{for} \;\; k = 1, \cdots, K \text{ and agents}(i,j) \text{ connected.}$$

Disadvantages:

- no theoretical gaurantee on clustering quality;
- slow convergence when data is huge.

Another idea

How to make all agents agree on the centroids?

Distributed consensus

To get all agents in a network to agree on some specific value.

Our method

How to do k-means in a distributed setting:

- reassign observations ✓
- update centers for $k = 1, \cdots, K$

$$\boldsymbol{c}_k \leftarrow \frac{\sum_m^M \sum_{\boldsymbol{x} \in P_k^{(m)}} \boldsymbol{x}}{\sum_m^M \mid P_k^{(m)} \mid},$$

where $P_k^{(m)}$ is the k-th cluster of agent m.

$$\frac{\sum_m^M \sum_{\boldsymbol{x} \in P_k^{(m)}} \boldsymbol{x}}{\sum_m^M \mid P_k^{(m)} \mid} = \frac{\frac{1}{M}\sum_m^M \sum_{\boldsymbol{x} \in P_k^{(m)}} \boldsymbol{x}}{\frac{1}{M}\sum_m^M \mid P_k^{(m)} \mid} = \frac{\text{average of } \sum_{\boldsymbol{x} \in P_k^{(m)}} \boldsymbol{x}}{\text{average of } \mid P_k^{(m)} \mid}.$$

Calculation of $\boldsymbol{c}_k$ is amenable to average-consensus!✓

Core idea: summation & average.

Implementation

- The distributed k-means++[1] initialization $\Rightarrow$ faster convergence and theoretical gaurantee on clustering quality.
- Most average consensus algorithms are merely asymptotically correct. We use a finite-time average-consensus algorithm[2] $\Rightarrow$ exactly k-means.

[1] David Arthur and Sergei Vassilvitskii, "k-means++: The advantages of careful seeding," in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[2] Shreyas Sundaram and Christoforos N Hadjicostis, "Finite- time distributed consensus in graphs with time-invariant topologies," in 2007 American Control Conference. IEEE, 2007, pp. 711–716.

Distributed k-means

Algorithm 1: Distributed k-means++: agent m.

Data: $\mathcal{X}^{(m)}$, M, K, $N^{(m)}$, ϵ
Result: $\mathbf{C}$, $P_i^{(m)}$, $i = 1, \ldots, |\mathcal{X}^{(m)}|$

$\mathbf{C} \leftarrow$ k-means++ initialization;
while True do
  $\tilde{\mathbf{C}} \leftarrow \mathbf{C}$;
  // assignment
  for $i \leftarrow 1$ to $|\mathcal{X}^{(m)}|$ do
    $P_i^{(m)} \leftarrow \arg\min_k \left\| \boldsymbol{x}_i^{(m)} - \boldsymbol{c}_k \right\|$;
  end
  // update of centers
  for $k \leftarrow 1$ to K do
    $\boldsymbol{S}_k^{(m)} \leftarrow \mathbf{0}$;
    $n_k^{(m)} \leftarrow 0$;
    for $i \leftarrow 1$ to $|\mathcal{X}^{(m)}|$ do
      if $P_i^{(m)} == k$ then
        $n_k^{(m)} \leftarrow n_k^{(m)} + 1$;
        $\boldsymbol{S}_k^{(m)} \leftarrow \boldsymbol{S}_k^{(m)} + \boldsymbol{x}_i^{(m)}$;
      end
    end
    $\boldsymbol{c}_k \leftarrow \frac{\text{avg-con}\left(\boldsymbol{S}_k^{(m)}, N^{(m)}, M\right)}{\text{avg-con}\left(n_k^{(m)}, N^{(m)}, M\right)}$;
  end
  if $\left\| \tilde{\mathbf{C}} - \mathbf{C} \right\| < \epsilon$ then
    break;
  end
end

Data sets and network topology

Table: Data Set Descriptions. (N: # points, D: # features)

ID	Data Set	N	D
1	S4	5000	2
2	Cloud	1024	10
3	Air-Quality Data	35065	18
4	Activity recognition	75128	9
5	Wave Energy Converters	72000	49

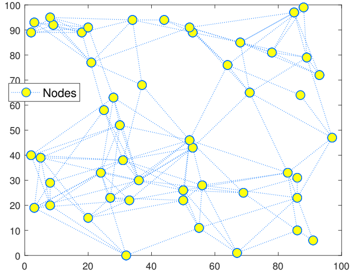


Figure: An example of network topology diagram with nodes = 50.

- DKM, DKM++ iterations almost completely match the CKM, CKM++.
- K-means++ outperforms RI in terms of convergence rate and clustering quality.

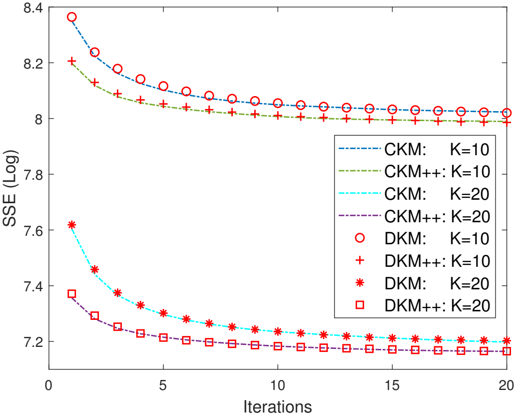


Figure: Average SSE curves of DKM and CKM with $K = 10,\ 20$ and two initialization methods: Random initialization (RI) and K-means++ (or DKM++ in distributed cases), S4 data set (ID: 1), 100 Monte Carlo runs.

Comparison with DCWSN

Table: Performance comparison between DKM and DCWSN

Data Set ID	Index	DKM++	DCWSN-Z	DCWSN-P
1	SSE	2.91E+03	3.09E+03	2.99E+03
	Ratio	1.00	1.06	1.03
2	SSE	1.52E+07	1.74E+07	1.63E+07
	Ratio	1.00	**1.15**	1.07
3	SSE	1.53E+09	1.58E+09	1.57E+09
	Ratio	1.00	1.03	1.02
4	SSE	6.91E+07	1.25E+08	1.17E+08
	Ratio	1.00	**1.81**	1.70
5	SSE	1.12E+14	1.17E+14	1.14E+14
	Ratio	1.00	1.04	1.02

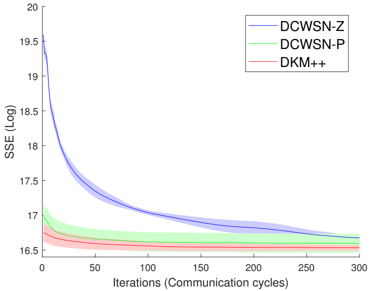


Figure: SSE curves of three algorithms, Cloud data set (ID: 2), 10 Monte Carlo runs.

Compared with existing work DCWSN, the proposed DKM and DKM++ have

- Better clustering quality
- Faster convergence

- A network composed by environmental monitoring stations (agents).
- Clustering analysis for environmental monitoring station data sets.
- Study weather and air pollution patterns of the area.

Figure: A typical environmental monitoring station

Figure: A network of environmental monitoring stations

Conclusion and Discussion

- Distributed K-means and K-means++.
- Same performance with the centralized counterparts but with less data traffic.
- Better performance than the existing distributed clustering algorithms.
- A journal article that covers distributed soft clustering and hard clustering algorithms:

  H. Yu, H. Chen, S. Zhao and Q. Shi, "Distributed Soft Clustering Algorithm For IoT Based on Finite Time Average Consensus," in IEEE Internet of Things Journal.

Thank you for listening.

Stay strong, stay safe!