An Evaluation of a Hierarchical Clustering Method Using Quantum Annealing

Masahito Kumagai, Kazuhiko Komatsu, Masayuki Sato, Hiroaki Kobayashi
Tohoku University

Introduction – Quantum Annealing and Clustering –

- ✓ **Digital computers** (e.g. CPU, Vector Processing Unit (VPU), GPU)
  - ○ General-purpose computing capability for various problems
  - × Performance limited by the Moore's law
  - × Huge amounts of power consumption
- ✓ **Quantum annealing (QA)** (on Quantum Processing Unit (QPU))
  - ○ Power efficient
  - ○ Accelerating **the combinatorial optimization problems**
  - × Special-purpose processing capability for limited problems

Hybrid computing with digital computers and QA machine

- ✓ **Non-hierarchical clustering** (e.g. K-means)
  - ○ Low computational complexity
  - × Need to know the number of clusters in advance
- ✓ **Hierarchical clustering**
  - ○ **No need to know the number of clusters in advance**
  - × High computational complexity

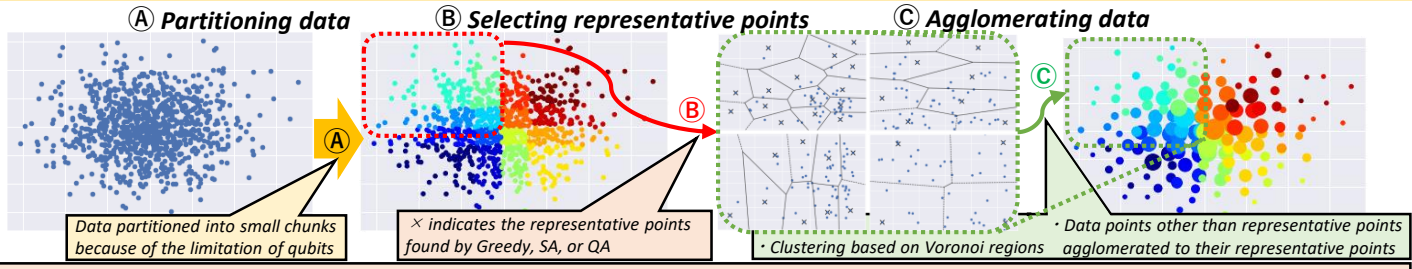
Hierarchical agglomerative clustering (HAC) using the combinatorial optimization problem of Maximum Weighted Independent Set (MWIS) [1]

Possibility of accelerating clustering by combining QA to solve the combinatorial optimization problem and digital computers to process the other

Objective

Clarify features of each processor and each clustering method by comparing the execution time and quality of the clustering method

A Combinatorial Optimization Problem in Hierarchical Agglomerative Clustering



✓How to choose representative points of each cluster by MWIS

- *A chunk having $\{x_1, \ldots, x_n\}$ and each weight of w_i*
- *ε defined as degree of similarity*
- *Binary variables S obtained as solutions of MWIS*

① *Similarity matrix* $N_{ij}^{(\epsilon)} = \begin{cases} 1 & if\ distance(x_i, x_j) < \varepsilon \\ 0 & otherwise \end{cases}$

② *Quadratically Constrained Quadratic Program (QCQP) given by* $\underset{S\in(0,1)^n}{maximize} \sum_{i=1}^{n} s_i w_i \quad subject\ to \sum_{i=1}^{n}\sum_{i<j} s_i N_{ij}^{(\epsilon)} s_j = 0$

③ *1. QCQP is being solved by* ***Greedy algorithm***
2. QCQP is transformed into QUBO[2] to solve by ***SA*** *or* ***QA***

Performance Evaluation

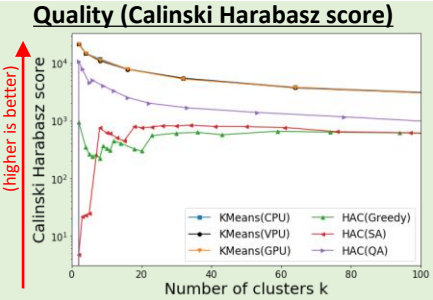
Data set : MoCap Hand Postures

- 5 types of hand postures from 14 users in a motion capture environment
- The number of data is 78,095 and the number of features used for experiments is 9 out of 36 features

Environments

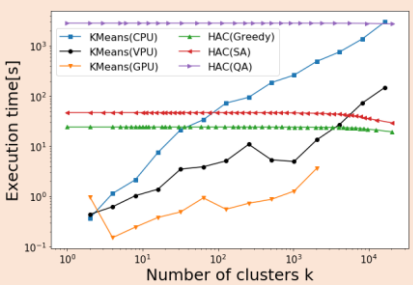
Method		Hardware	Software
K-means	CPU	Intel Xeon Gold 6126	Scikit-learn v0.22.0
	VPU	NEC Vector Engine Type 10B	Frovedis v0.9.5
	GPU	NVIDIA Tesla V100	Rapidsai v0.12.0a
SA		Intel Xeon Gold 6126	OpenJij v0.0.9
Greedy			-
QA		D-Wave 2000Q	Ocean SDK v1.4.0

Quality (Calinski Harabasz score)



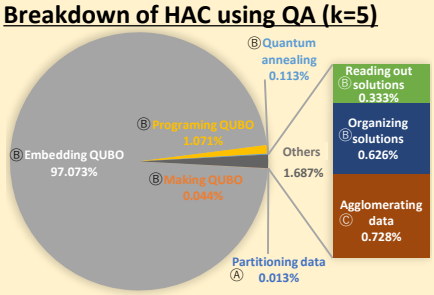
- QA achieves the best quality in HAC
  - The quality of solutions of QCQP is important for HAC
    - QA can search the whole search space by quantum fluctuations
    - SA cannot improve the quality by increasing the annealing time to explore the search space
    - Greedy falls into a local optimum and cannot find the best solution
- K-means is always superior to HAC
  - K-means does not agglomerate data and can avoid loss of information of the original data

Execution time



- Stable execution times of HAC (Greedy, SA, and QA)
  - It is because the number of processed data decreases by agglomerating data for each hierarchy
  - Since HAC can get all clusters at once, it does not take a long time if the number of clusters is known
- Quite low performance of HAC on QA
  - HAC on QA still needs to be accelerated
- The execution times of K-means increase as the number of clusters increases
  - Since K-means needs to sequentially process many times with the different number of clusters, it takes so much time when the number of clusters is unknown.

Breakdown of HAC using QA (k=5)



- The time for quantum annealing is short (3.21s)
- QUBO embedding executed on a CPU dominates the whole execution time
  - QUBO embedding duplicates data to make up for missing connections because not all qubits can be connected in the D-Wave 2000Q machine

Discussion : towards the high performance clustering

- Combination of the hierarchical clustering on QA and non-hierarchical clustering on digital computers could be promising
  - HAC should get the optimal number of clusters, when it is unknown
  - K-means should get the high quality result after the number of clusters is decided by HAC

Conclusions and Future Work

- While the quality of the clustering results by HAC using QA becomes higher than those of SA and Greedy, K-means is always superior to HAC.
- HAC is faster than K-means when the number of clusters is unknown.
- As future work, we will combine hierarchical clustering on QA to get the optimal number of the clusters and non-hierarchical clustering on digital computers to get the high quality result.

Acknowledgments

This research was partially supported by MEXT Next Generation High-Performance Computing Infrastructures and Applications R&D Program, entitled "R&D of A Quantum-Annealing-Assisted Next Generation HPC Infrastructure and its Applications." and by Grants-in-Aid for Scientific Research(A) #19H01095. The authors would like to thank Jij Inc. for helpful comments.

references
[1] Tim, J. et al. :A Quantum Annealing-Based Approach to Extreme Clustering, FICC 2020, 2020 , DOI: 10.1007/978-3-030-39442-4_15.
[2] Boros, E. et al. :Local search heuristics for quadratic unconstrained binary optimization (QUBO), *Journal of Heuristics*, 2007, 13.2: 99-132.