KEYBIN: KEY-BASED BINNING FOR DISTRIBUTED CLUSTERING

XINYU CHEN, JEREMY BENSON, TRILCE ESTRADA, COMPUTER SCIENCE DEPARTMENT, UNIVERSITY OF NEW MEXICO

PROBLEM

The Big Data era brings new challenges to machine learning. Traditional learning algorithms often require centralized data, but modern data sets are collected and stored in a distributed way. We are now facing the following problems:

1. Moving data is expensive
2. Privacy concerns restrict data moving
3. Curse of dimensionality
4. Noisy features in high dimensional data

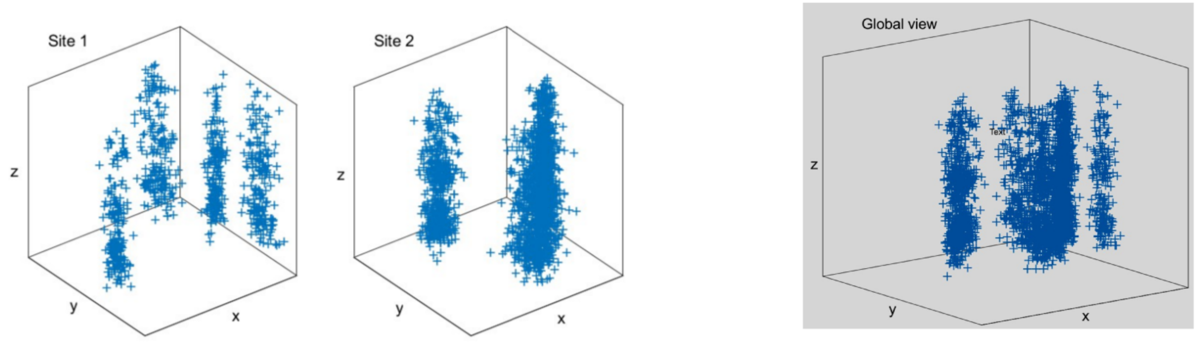
INTRODUCTION

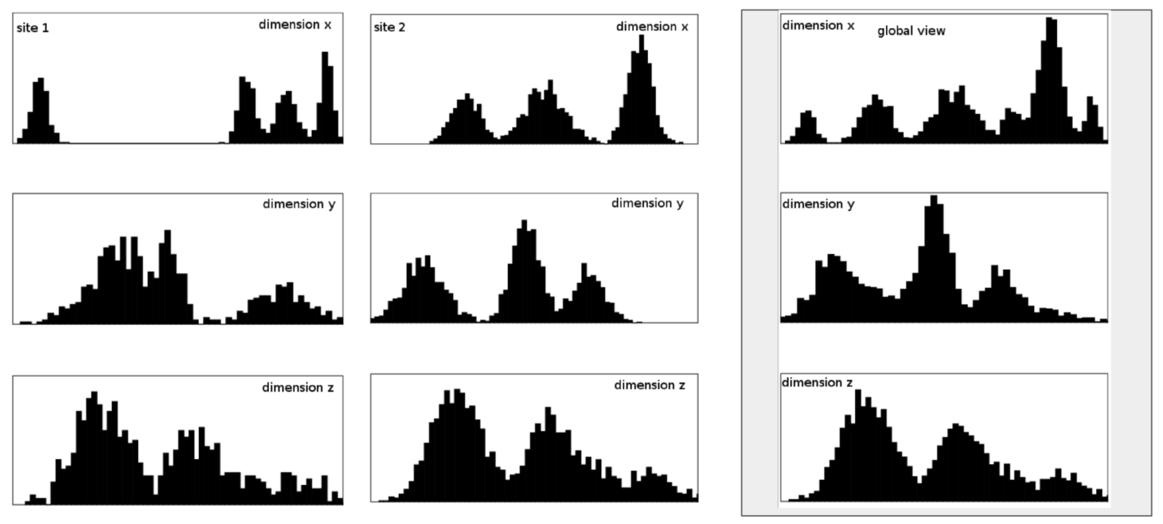
We present keybin, a scalable and accurate clustering algorithm, suitable for distributed and privacy constrained environments. Learning from statistics information, avoid pair-wise distance computations. Our contributions are:

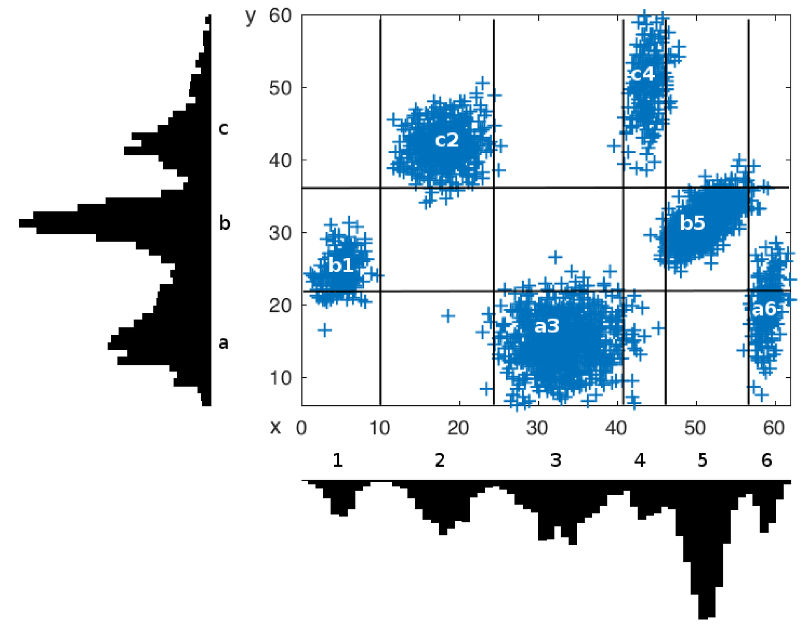
1. A scalable and accurate clustering approach
2. A math method to discard noisy features
3. Use limited view of data to preserve privacy
4. Compare with other clustering algorithms

AN EXAMPLE OF LEARNING HETEROGENOUS STRUCTURES

Consider data distributed on two sites has different distributions. The left plot shows these two sites and their local data. The shaded plot shows the true global patterns we want to learn. However, to move them to a central location is either expensive or restricted. keybin computes histograms on each site then aggregates to a global view of the whole data to assign final clusters.



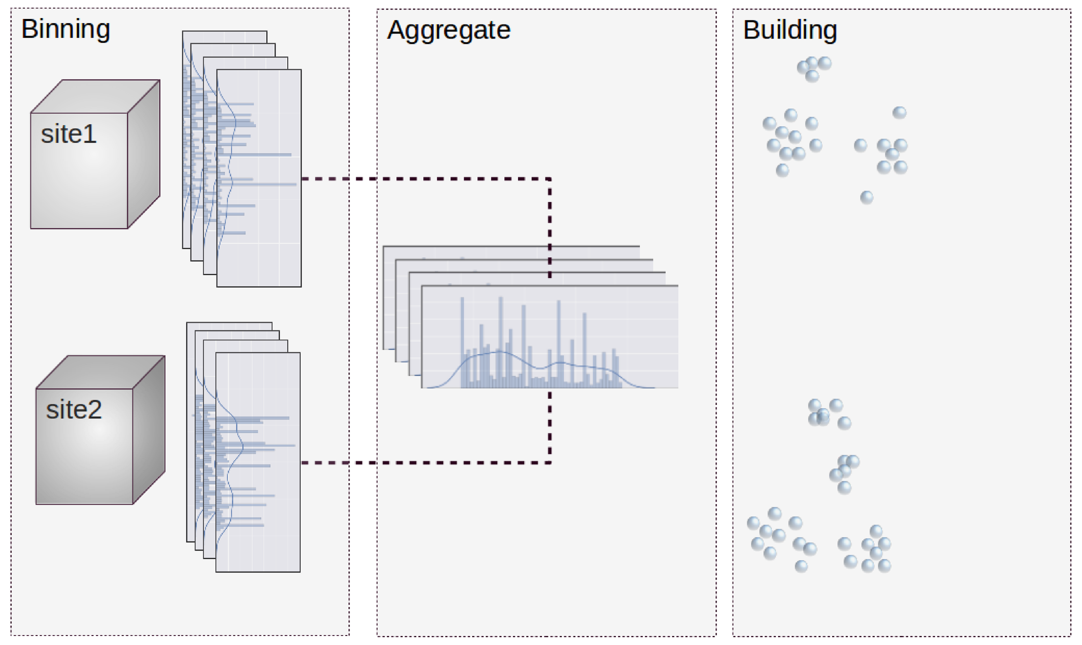




METHODS

keybin follows the following method:

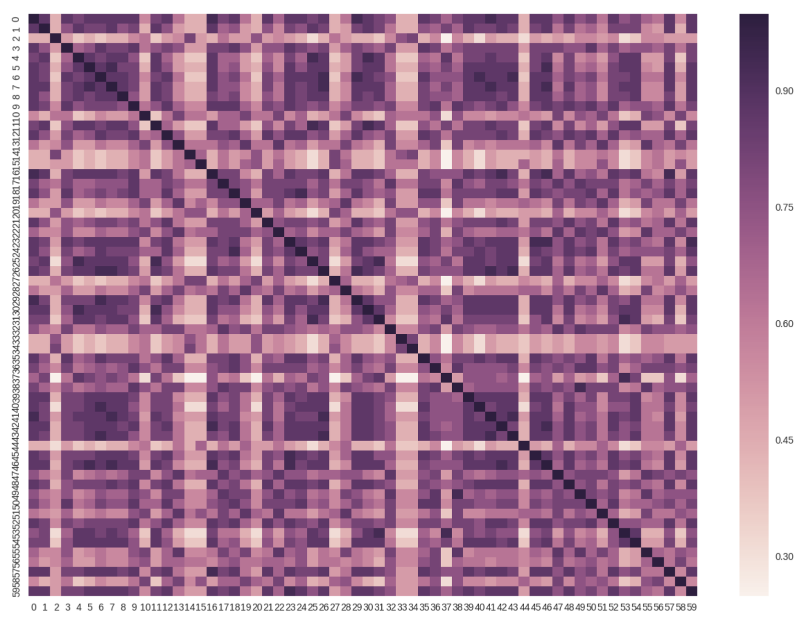
1. Assign keys to data points
2. Aggregate global densities
3. Collapsing noisy features
4. Build primary clusters
5. Reduce to final clustering

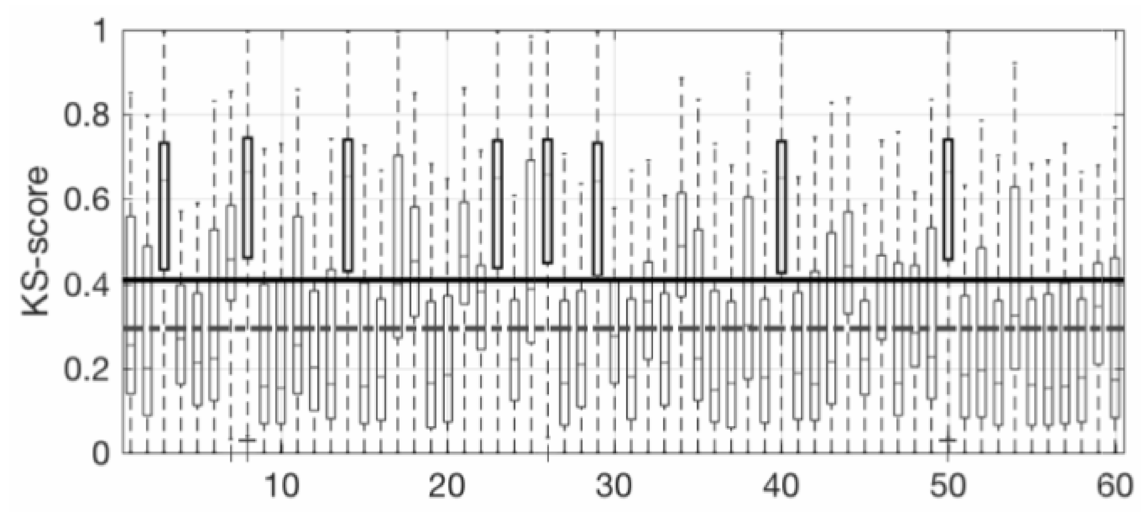


Inspired by hierarchical clustering algorithms. High dimensional clusters consist of lower dimensional primary clusters. Points do not know other points. Features don not affect each other. Ideal for embarrassing parallel implementation.

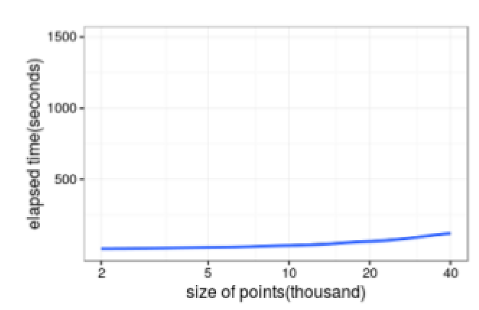
COLLAPSING DIMENSIONS

Some features in high-dimensional data contain noises. We use Kolmogorov-Smirnov Test to filter out noises. We first compute a expected KS-score for all features. Then we discard features that are different(0.5 σ from the median KS-score).

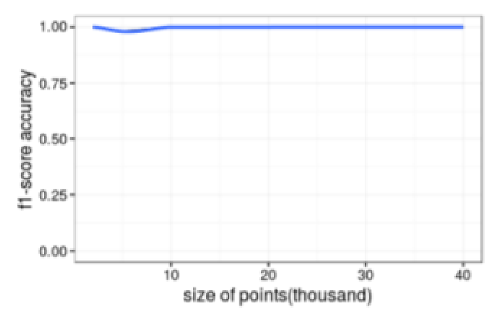




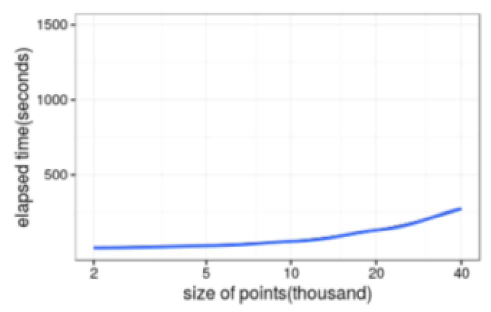
EVALUATIONS AND CONCLUSION



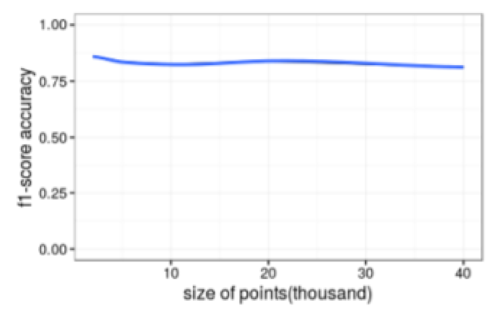
(a) keybin - time



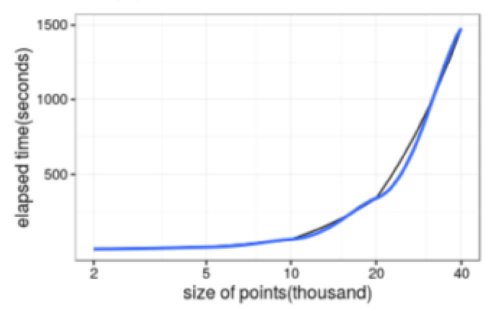
(b) keybin - f1 score



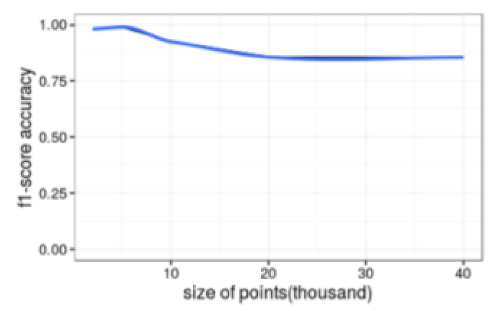
(c) K-means - time



(d) K-means - f1 score



(e) DBSCAN - time



(f) DBSCAN - f1 score

Algorithm	Scale w/ size	Scale w/ dimension	Need merge	Correct clusters
keybin	✓	✓		✓
K-Means	✓	✓	✓	
DBSCAN			✓	✓
PDSDBSCAN	✓	✓	✓	
GPUMAFIA	✓		✓	

- No pair-wise distance computation
- Learn with limited communications
- Scalable with size and dimensionality

REFERENCES

[1] Agrawal et al, Fast algorithms for mining association rules , VLDB, (1994).

[2] Agrawal et al, Automatic subspace clustering of high dimensional data for data mining applications, ACM, (1998).

LIMITATION AND FUTURE RESEARCH

keybin assumes features are orthogonal to each other. In cases when correlated features exist, the projection of some clusters overlap on correlated dimensions. This leads false positive in keybin.

We use synthetic data sets to test our methods. We need to apply keybin to real data sets and find out methods to deal with overlapping clusters.

Full paper is available: http://cs.unm.edu/ xychen/keybin-cluster17.pdf

ACKNOWLEDGEMENTS

This research was supported by the National Science Foundation for the grant entitled *CAREER: Enabling Distributed and In-Situ Analysis for Multidimensional Structured Data* (NSF ACI-1453430).