Graph sketching-based Space-efficient Data Clustering

SIAM Data Mining, 2018

A. Morvan[1,2] K. Choromanski[3] C. Gouy-Pailler [1] J. Atif [2]

[1]CEA, LIST

[2]Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE

[3]Google Brain Robotics

May 3, 2018

Plan

Plan

Objectives

Context

Resources-limited devices collecting huge volume of data

A clustering algorithm...

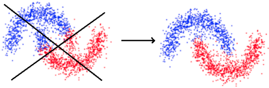
- Recognizing arbitrary non-convex cluster shapes
- With no parameter
- In a time linear to the number of points N
- Under high space constraints

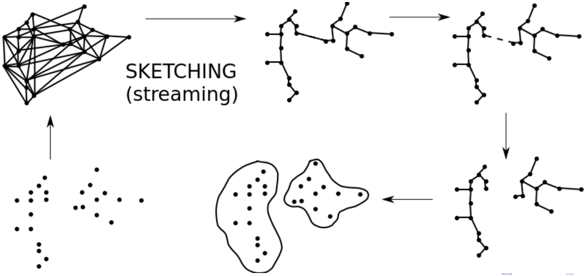
Plan

Principle

Minimum-Spanning-Tree-based (MST) clustering algorithm

- MST: A useful and compact summary of the data dissimilarity graph
- Appealing property: helping to recover arbitrarily-shaped clusters
- Idea: perform suitable cuts on the MST





Related work

Graph clustering [Schaeffer, 2007]

- DenGraph [Falkowski et al., 2007]: graph version of DBSCAN
- [Ailon et al., 2013] recovering clusters with dissimilar sizes
- Convex optimization [Oymak and Hassibi, 2011, Chen et al., 2012, Chen et al., 2014a, Chen et al., 2014b]

MST-based graph clustering

- [Zahn, 1971, Asano et al., 1988, Mitra et al., 2003, Grygorash et al., 2006]

Space-efficient clustering

- Streaming k-means [Ailon et al., 2009]: only the centroïd is stored
- CURE algorithm [Guha et al., 2001]: $O(N^2 \log(N))$ time complexity
- CluStream [Aggarwal et al., 2003] and DenStream [Cao et al., 2006]

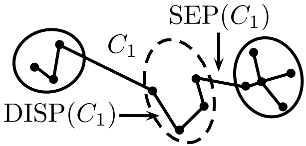
Cluster Dispersion and Separation

Cluster Dispersion

$$\forall i \in [K],\ \mathrm{DISP}(C_i) = \begin{cases} \max\limits_{j,\ e_j \in C_i} w_j & \text{if } \ |E(C_i)| \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Cluster Separation

$$\forall i \in [K],\ \mathrm{SEP}(C_i) = \begin{cases} \min\limits_{j,\ e_j \in Cuts(C_i)} w_j & \text{if } \ K \neq 1 \\ 1 & \text{otherwise.} \end{cases}$$



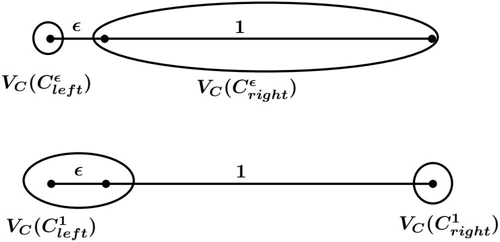
Validity Index of a Cluster and of a Clustering Partition

Validity Index of a Cluster

$$V_C(C_i) = \frac{\mathrm{SEP}(C_i) - \mathrm{DISP}(C_i)}{\max(\mathrm{SEP}(C_i), \mathrm{DISP}(C_i))} \in [-1, 1]$$

Validity Index of a Clustering partition

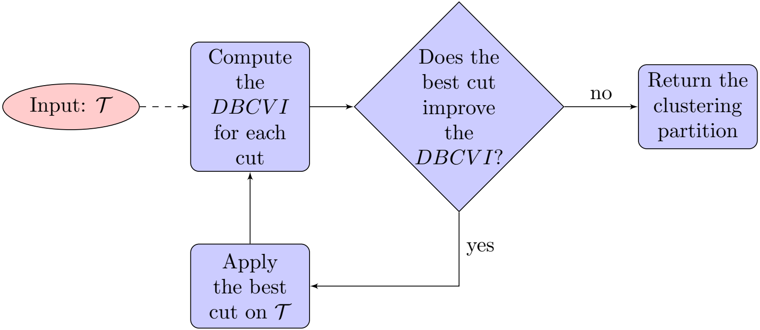
$$\mathrm{DBCVI}(\Pi) = \sum_{i=1}^{K} \frac{|C_i|}{N} V_C(C_i) \in [-1, 1]$$



$$V_C(C^{\epsilon}_{left}) = 1$$
$$V_C(C^{\epsilon}_{right}) = \epsilon - 1 < 0$$

$$V_C(C^{1}_{left}) = 1 - \epsilon > 0$$
$$V_C(C^{1}_{right}) = 1$$

Algorithm DBMSTClu($\mathcal{T}$)



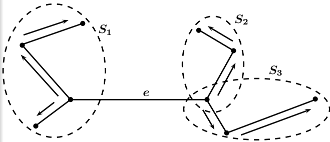
Scalability

MST computation

- Graph sketching [Ahn et al., 2012] in $O(N \log^3(N))$ space complexity in the semi-streaming setting
- Approximate MST recovery from the graph sketch

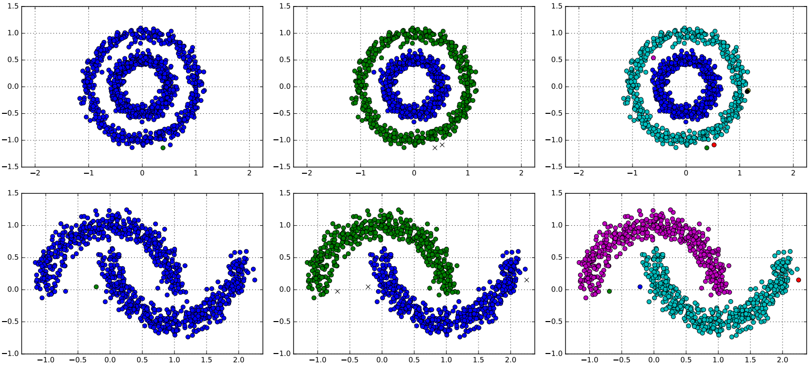
Linear time and space complexities of DBMSTClu

1. A cut in cluster C_i lets $V_C(C_j),\ \forall j \neq i$ unchanged.
2. Recurrence relationship of SEP and DISP in $\mathcal{T}$. Iterative version of the Depth-First Search to determine DBCVI for each cut left and right: Double Depth-First Search.



Plan

Safety of the sketching

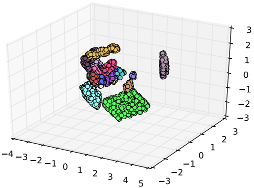


SEMST DBSCAN DBMSTClu

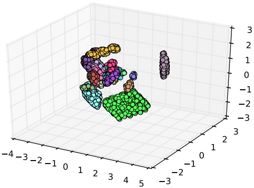
	Silhouette coeff.		ARI		DBCVI	
SEMST	**0.16**	-0.12	0	0	0.001	0.06
DBSCAN	0.02	**0.26**	**0.99**	**0.99**	-0.26	**0.15**
DBMSTClu	-0.26	**0.26**	**0.99**	**0.99**	**0.18**	**0.15**

Scalability of the clustering

Mushroom dataset (8124 nodes), time to recover 23 clusters:



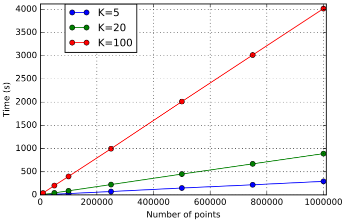
DBSCAN: 9s



DBMSTClu: 3.36s

In the Stochastic Block Model, time (s) to recover the K clusters w.r.t N:

$K \backslash N$	1000	10000	50000	100000	250000	500000	750000	1000000
5	0.34	2.96	14.37	28.91	73.04	148.85	218.11	292.25
20	0.95	8.73	43.71	88.51	223.18	449.37	669.29	889.88
100	4.36	40.25	201.76	398.41	995.42	2011.79	3015.61	4016.13
"100/5"	12.82	13.60	14.04	13.78	13.63	13.52	13.83	13.74



Plan

Conclusion

Take-home message: DBMSTClu is an ...

- MST-based
- parameter-free
- space-efficient clustering algorithm
- for arbitrarily-shaped clusters

https://github.com/annemorvan/DBMSTClu

Future perspectives

- A fully online clustering algorithm
- Exact clustering partition recovery theoretical guarantees (submitted)
- A Differentially Private clustering algorithm based on a private release of the MST (submitted)

https://annemorvan.github.io/

Thank you for your attention!

Today, poster presentation in Salon A-C from 7pm to 9pm!

anne.morvan@cea.fr[1]

[1]Anne is partly supported by the DGA (French Ministry of Defense).

References I

Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003).
A framework for clustering evolving data streams.
In Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03, pages 81–92. VLDB Endowment.

Ahn, K. J., Guha, S., and McGregor, A. (2012).
Analyzing graph structure via linear measurements.
In Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12, pages 459–467, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Ailon, N., Chen, Y., and Xu, H. (2013).
Breaking the small cluster barrier of graph clustering.
CoRR, abs/1302.4549.

Ailon, N., Jaiswal, R., and Monteleoni, C. (2009).
Streaming k-means approximation.
In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A., editors, Advances in Neural Information Processing Systems 22, pages 10–18. Curran Associates, Inc.

Asano, T., Bhattacharya, B., Keil, M., and Yao, F. (1988).
Clustering algorithms based on minimum and maximum spanning trees.
In Proceedings of the Fourth Annual Symposium on Computational Geometry, SCG '88, pages 252–257, New York, NY, USA. ACM.

Cao, F., Ester, M., Qian, W., and Zhou, A. (2006).
Density-based clustering over an evolving data stream with noise.
In In 2006 SIAM Conference on Data Mining, pages 328–339.

References II

Chen, Y., Jalali, A., Sanghavi, S., and Xu, H. (2014a).
Clustering partially observed graphs via convex optimization.
J. Mach. Learn. Res., 15(1):2213–2238.

Chen, Y., Lim, S. H., and Xu, H. (2014b).
Weighted graph clustering with non-uniform uncertainties.
In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, pages II–1566–II–1574. JMLR.org.

Chen, Y., Sanghavi, S., and Xu, H. (2012).
Clustering sparse graphs.
In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 25, pages 2204–2212. Curran Associates, Inc.

Cormode, G. and Firmani, D. (2014).
A unifying framework for l_0-sampling algorithms.
Distributed and Parallel Databases, 32(3):315–335.
Special issue on Data Summarization on Big Data.

Falkowski, T., Barth, A., and Spiliopoulou, M. (2007).
Dengraph: A density-based community detection algorithm.
In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, WI '07, pages 112–115, Washington, DC, USA. IEEE Computer Society.

Grygorash, O., Zhou, Y., and Jorgensen, Z. (2006).
Minimum spanning tree based clustering algorithms.
In 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), pages 73–81.

References III

Guha, S., Rastogi, R., and Shim, K. (2001).
Cure: An efficient clustering algorithm for large databases.
Inf. Syst., 26(1):35–58.

Mitra, P., Pal, S. K., and Siddiqi, M. A. (2003).
Non-convex clustering using expectation maximization algorithm with rough set initialization.
Pattern Recognition Letters, 24(6):863 – 873.

Oymak, S. and Hassibi, B. (2011).
Finding dense clusters via "low rank + sparse" decomposition.
CoRR, abs/1104.5186.

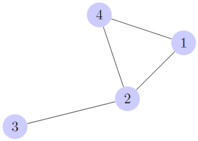
Schaeffer, S. E. (2007).
Survey: Graph clustering.
Comput. Sci. Rev., 1(1):27–64.

Zahn, C. T. (1971).
Graph-theoretical methods for detecting and describing gestalt clusters.
IEEE Trans. Comput., 20(1):68–86.

Plan

Graph sketching [Ahn et al., 2012, Cormode and Firmani, 2014]

A compact structure for $\mathcal{G}$ in $O(N \log^3(N))$



weighted: $\mathcal{G}$ → $\mathcal{G}_0 \quad \cdots \quad \mathcal{G}_i \quad \cdots \quad \mathcal{G}_{\lceil \log_{1+\epsilon}(w_{max}) \rceil}$

unweighted: $\mathcal{G}_i \subset \mathcal{G}_{i+1}$

$$\mathbf{a}^{(1)} = (\ \ 1,\ 0,\ 1,\ 0,\ 0,\ 0)$$
$$\mathbf{a}^{(2)} = (-1,\ 0,\ 0,\ 1,\ 1,\ 0)$$
$$\vdots$$

$$\left.\begin{aligned} \phi &= \sum_j \mathbf{a}_j^{(i)} \\ \iota &= \sum_j j\ \mathbf{a}_j^{(i)} \\ \tau &= \sum_j \mathbf{a}_j^{(i)}\ z^j \bmod p \end{aligned}\right\} \times O(\log^3(N))$$

L levels of representation:

$$\begin{cases} h : [M] \to [L] \\ Pr[h(j) = l] = \frac{1}{2^l} \end{cases}$$

1-sparsity test

If $\tau = \phi\ z^{\frac{\iota}{\phi}} \bmod p$ then $\mathbf{a}^{(i)}$ is 1-sparse. If $\mathbf{a}^{(i)}$ is 1-sparse: always + answer, otherwise − answer with prob. at least $1 - M/p$.

Thank you for your attention!