Graph sketching-based Massive Data Clustering

Anne Morvan[12], Krzysztof Choromanski[3], Cédric Gouy-Pailler[1] and Jamal Atif[2]

[1]CEA, LIST, [2]Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243, LAMSADE, [3]Google Brain Robotics

Objectives

We present a new clustering algorithm DBMSTClu providing a solution to the following issues: 1) detecting arbitrary-shaped data clusters, 2) with no parameter, 3) in a space-efficient manner by working on a limited number of linear measurements, a *sketched* version of the dissimilarity graph G.

Steps of the method

1. The dissimilarity graph of data G with N nodes is handled as a stream of edge weight updates.
2. G is sketched in the dynamic semi-streaming model in one pass over the data into a compact structure requiring $O(N \operatorname{polylog}(N))$ space with method from work [1] relying on ℓ_0-sampling principle [2].
3. From the sketch, an AMST is recovered containing $N-1$ edges with weight $0 < w_i \leq 1$ for all $i = 1, \ldots, N-1$.
4. Apply DBMSTClu on the given AMST (good for expressing the underlying structure of a graph) which successfully detects the right number of non-convex clusters. DBMST is a divisive top-down procedure: at each iteration, a cut among the edges of T is performed creating a new connected component. The best cut to do is identified thanks to a criterion based on *Dispersion* and *Separation* of one cluster.

Cluster Dispersion

The Dispersion of a cluster C_i (DISP) represented by the subtree S_i of MST T is defined as the maximum edge weight of S_i:

$$\forall i \in [K],\ \mathrm{DISP}(C_i) = \begin{cases} \max\limits_{j,\ e_j \in S_i} w_j \text{ if } |E(S_i)| \neq 0 \\ 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{1}$$

Cluster Separation

The Separation of a cluster C_i (SEP) is defined as the minimum distance between the nodes of C_i and the ones of all other clusters $C_j, i \neq j, 1 \leq i, j, \leq K, K \neq 1$ where K is the total number of clusters and $Cuts(C_i)$ denotes the edges incident to cluster C_i:

$$\forall i \in [K],\ \mathrm{SEP}(C_i) = \begin{cases} \min\limits_{j,\ e_j \in Cuts(C_i)} w_j \text{ if } K \neq 1 \\ 1 \qquad\qquad \text{otherwise.} \end{cases} \tag{2}$$

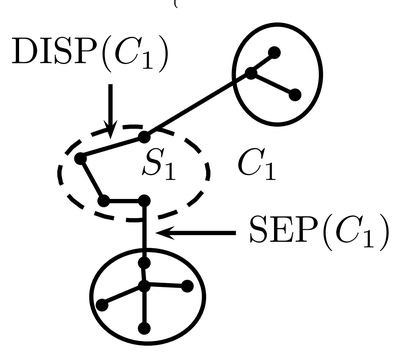

Figure 1: SEP and DISP for cluster C_1 represented by subtree S_1 of T ($N = 12$, $K = 3$).

Validity Index of a Cluster

The Validity Index of a cluster C_i, $1 \leq i \leq K$ is defined as:

$$V_C(C_i) = \frac{\mathrm{SEP}(C_i) - \mathrm{DISP}(C_i)}{\max(\mathrm{SEP}(C_i), \mathrm{DISP}(C_i))} \tag{3}$$

Validity Index of a Clustering Partition

The Density-Based Validity Index of a Clustering partition $\Pi = \{C_i\}, 1 \leq i \leq K$, DBCVI($\Pi$) is defined as the weighted average of the Validity Indices of all clusters in the partition.

$$\mathrm{DBCVI}(\Pi) = \sum_{i=1}^{K} \frac{|C_i|}{N} V_C(C_i) \tag{4}$$

To whom correspondence should be adressed: anne.morvan@cea.fr. Her PhD thesis is partly supported by the *Direction Générale de l'Armement* (French Ministry of Defense).

Algorithm 1 DBMSTClu algorithm

Input: T, the (A)MST
$splitDBCVI \leftarrow -1.0$; $cut\_candidate\_list \leftarrow [\ edges(T)\ ]$; $clusters = [\ ]$
while $splitDBCVI < 1.0$ **do**
  $temp\_cut \leftarrow None$; $temp\_DBCVI \leftarrow splitDBCVI$
  for each cut in $cut\_candidate\_list$ **do**
    $newClusters \leftarrow performCut(clusters, cut)$
    $newDBCVI \leftarrow getDBCVI(newClusters, T)$
    if $newDBCVI \geq temp\_DBCVI$ **then**
      $temp\_cut \leftarrow cut$; $temp\_DBCVI \leftarrow newDBCVI$
  if $temp\_cut \neq None$ **then**
    $clusters \leftarrow performCut(clusters, temp\_cut)$
    $splitDBCVI \leftarrow temp\_DBCVI$
    $remove(cut\_candidate\_list, temp\_cut)$
  else
    break
return $clusters$, $splitDBCVI$

Experimental results

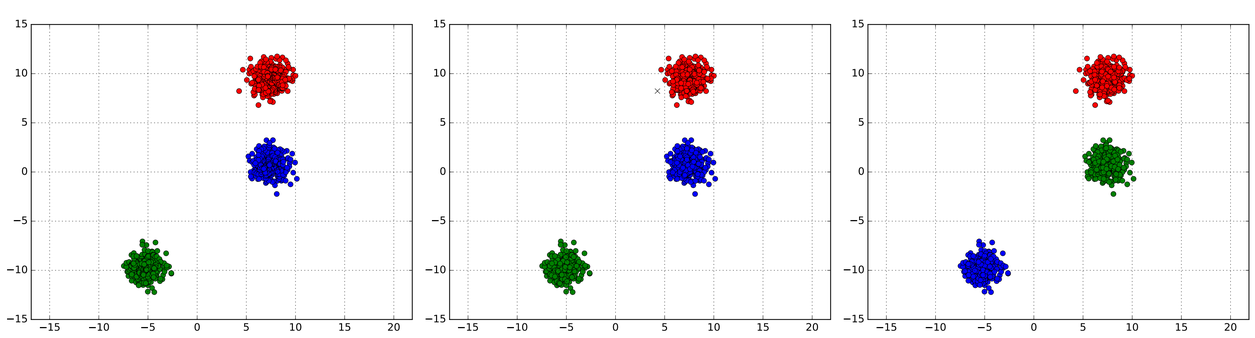

Figure 2: Three blobs: SEMST, DBSCAN ($\epsilon = 1.0$, $minPts = 5$), DBMSTClu with AMST

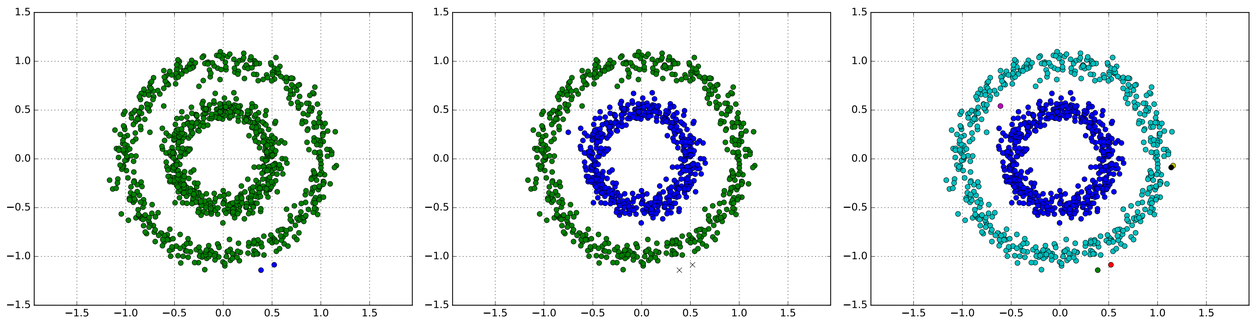

Figure 3: Noisy circles: SEMST, DBSCAN ($\epsilon = 0.15$, $minPts = 5$), DBMSTClu with AMST

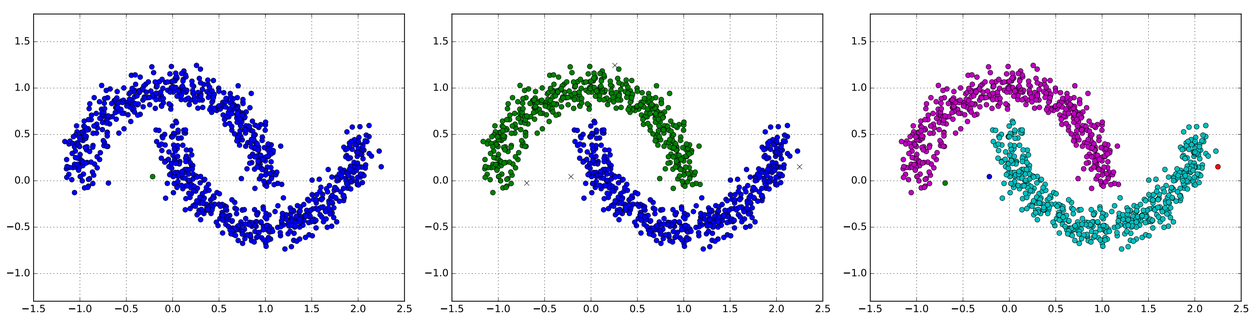

Figure 4: Noisy moons: SEMST, DBSCAN ($\epsilon = 0.15$, $minPts = 5$), DBMSTClu with AMST

Conclusion and perspectives

- We introduced a novel non-parametric space-efficient density-based clustering algorithm relying on a sketch built dynamically on the fly as new edge weight updates are received. Its robustness has been assessed by using as input a sketch of the MST rather than the MST itself.
- Further work would be to use DBMSTClu in privacy issues and adapt both the MST recovery and DBMSTClu to the fully online setting i.e. update current MST and clustering partition as new edge weight updates are seen.

[1] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor.
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, 2012. Society for Industrial and Applied Mathematics.

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

[3] Anne Morvan, Krzysztof Choromanski, Cédric Gouy-Pailler, and Jamal Atif.
Graph sketching-based massive data clustering.
CoRR, abs/1703.02375, 2017.