Graph sketching-based Space-efficient Data Clustering (Supplementary material)

Anne Morvan[*,†,‡] anne.morvan@cea.fr
Krzysztof Choromanski[§] kchoro@google.com
Cédric Gouy-Pailler[*] cedric.gouy-pailler@cea.fr
Jamal Atif[†] jamal.atif@dauphine.fr

1 Proofs.

1.1 Proof of Prop. 4.1.

PROPOSITION 4.1. WHEN THE FIRST CUT IS NOT THE HEAVIEST *Let $\mathcal{T}$ be an MST of the dissimilarity data graph with N nodes. Let us consider this specific case: all edges have a weight equal to w except two edges e_1 and e_2 resp. with weight w_1 and w_2 s.t. $w_1 > w_2 > w > 0$. DBMSTClu does not cut any edge with weight w and cuts e_2 instead of e_1 as a first cut iff:*

$$w_2 > \frac{2n_2w_1 - n_1 + \sqrt{n_1^2 + 4w_1(n_2^2w_1 + N^2 - Nn_1 - n_2^2)}}{2(N - n_1 + n_2)}$$

where n_1 (resp. n_2) is the number of nodes in the first cluster resulting from the cut of e_1 (resp. e_2). Otherwise, e_1 gets cut.

Proof. Let $DBCVI_1$ (resp. $DBCVI_2$) be the DBCVI after cut of e_1 (resp. e_2). As w (resp. w_1) is the minimum (resp. maximal) weight, the algorithm does not cut e since the resulting DBCVI would be negative (cf. Lemma 4.2) while $DBCVI_1$ is guaranteed to be positive (cf. Lemma 4.1). So, the choice will be between e_1 and e_2 but e_2 gets cut iff $DBCVI_2 > DBCVI_1$. $DBCVI_1$ and $DBCVI_2$ expressions are simplified w.l.o.g. by scaling the weights by w s.t. $w \leftarrow 1$, $w_1 \leftarrow w_1/w$, $w_2 \leftarrow w_2/w$, hence $w_1 > w_2 > 1$. Then,

$$\begin{aligned}
&DBCVI_2 > DBCVI_1 > 0\\
&\iff \frac{n_2}{N}(\frac{w_2}{w_1} - 1) + (1 - \frac{n_2}{N})(1 - \frac{1}{w_2})\\
&\qquad - \frac{n_1}{N}(1 - \frac{1}{w_1}) + (1 - \frac{n_1}{N})(1 - \frac{w_2}{w_1}) > 0\\
&\iff w_2^2 \underbrace{(N + n_2 - n_1)}_{a} + w_2 \underbrace{(n_1 - 2n_2w_1)}_{b}\\
&\qquad + \underbrace{(n_2 - N)w_1}_{c<0} > 0.
\end{aligned}$$

Clearly, $\Delta = b^2 - 4ac$ is positive and c/a is negative. But $w_2 > 0$, then $w_2 > \frac{-b+\sqrt{b^2-4ac}}{2a}$ which gives the final result after some simplifications.

1.2 Proof of Prop. 4.2.

PROPOSITION 4.2. FIRST CUT ON THE HEAVIEST EDGE IN THE MIDDLE *Let $\mathcal{T}$ be an MST of the dissimilarity data graph with N nodes. Let us consider this specific case: all edges have a weight equal to w except two edges e_1 and e_2 resp. with weight w_1 and w_2 s.t. $w_1 > w_2 > w > 0$. Denote n_1 (resp. n_2) the number of nodes in the first cluster resulting from the cut of e_1 (resp. e_2). In the particular case where edge e_1 with maximal weight w_1 stands between two subtrees with the same number of points, i.e. $n_1 = N/2$, e_1 is always preferred over e_2 as the first optimal cut.*

Proof. A reductio ad absurdum is made by showing that cutting edge e_2 i.e. $DBCVI_2 > DBCVI_1$ leads to the contradiction $w_1/w < 1$. With the scaling process from

*CEA, LIST, 91191 Gif-sur-Yvette, France.
†Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France.
‡Partly supported by the DGA (French Ministry of Defense).
§Google Brain Robotics, New York, USA.

Prop. 4.1'proof:

$$DBCVI_1 = \frac{1}{2}(1-\frac{1}{w_1}) + \frac{1}{2}(1-\frac{w_2}{w_1}) = 1 - \frac{1}{2w_1} - \frac{w_2}{2w_1}$$

$$\begin{aligned} DBCVI_2 &= \frac{n_2}{N}(\frac{w_2}{w_1}-1) + (1-\frac{n_2}{N})(1-\frac{1}{w_2}) \\ &= 1 - \frac{1}{w_2} + \frac{n_2}{N}\underbrace{(\frac{w_2}{w_1}+\frac{1}{w_2}-2)}_{=A} \end{aligned}$$

There is $w_2 > w = 1$, so $\frac{1}{w_2} < 1$. Besides $w_2 < w_1$ so $\frac{w_2}{w_1} < 1$ thus, $A < 0$. Let now consider w.l.o.g. that edge e_2 is on the "right side" (right cluster/subtree) of e_1 (similar proof if e_2 is on the left side of e_1). Hence, it is clear that for maximizing $DBCVI_2$ as a function of n_2, we need $n_2 = n_1 + 1$. Then,

$$\begin{aligned} & DBCVI_2 > DBCVI_1 \\ \iff & -\frac{1}{w_2} + (\frac{1}{2}+\frac{1}{N})(\frac{w_2}{w_1} - 2 + \frac{1}{w_2}) > -\frac{1}{w_1} - \frac{w_2}{w_1} \\ \iff & (\frac{1}{2w_1} + \frac{1}{Nw_1} + \frac{1}{2w_1})w_2 - 1 - \frac{2}{N} + \frac{1}{2w_1} \\ & \quad + (-1 + \frac{1}{2} + \frac{1}{N})\frac{1}{w_2} > 0 \\ \iff & \underbrace{(1+\frac{1}{N})}_{a>0} w_2^2 + w_2\underbrace{(\frac{1}{2} - w_1(1+\frac{2}{N}))}_{b<0} \\ & \quad + w_1\underbrace{(\frac{1}{N}-\frac{1}{2})}_{c<0} > 0 \end{aligned}$$

As $c/a < 0$ and $w_2 > 0$, $w_2 > \frac{N}{2(N+1)}$ $[\ w_1(1+\frac{2}{N}) - \frac{1}{2} + \sqrt{\Delta}\ ]$ with $\Delta = (w_1(1+\frac{2}{N}) - \frac{1}{2})^2 + 4(1+\frac{1}{N})(\frac{1}{2}-\frac{1}{N})w_1$. This inequality is incompatible with $w_1 > w_2$ since:

$$\begin{aligned} & w_1 > w_2 \iff w_1 > \frac{N}{2(N+1)}\ [\ w_1(1+\frac{2}{N}) - \frac{1}{2} + \sqrt{\Delta}\ ] \\ \iff & w_1 + \frac{1}{2} > \sqrt{\Delta} \\ \iff & \frac{4}{N}\ w_1^2\ (1+\frac{1}{N}) + \frac{4}{N}w_1(-1-\frac{1}{N}) < 0 \\ \iff & w_1 < 1\ \ : ILLICIT \end{aligned}$$

Indeed, after the scaling process, $w_1 < 1 = w$ is not possible since by hypothesis, $w_1 > w$. Finally, it is not allowed to cut e_2, the only remaining possible edge to cut is e_1.

1.3 Proof of Prop. 4.3.

Proposition 4.3. Fate of negative V_C cluster *Let $K = t+1$ be the number of clusters in the clustering partition at iteration t. If for some $i \in [K]$, $V_C(C_i) < 0$, then DBMSTClu will cut an edge at this stage.*

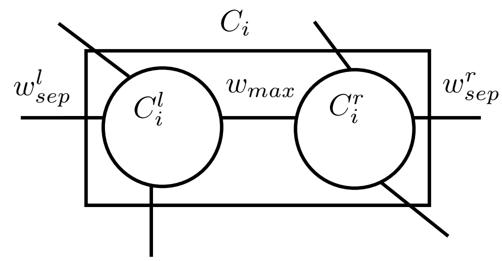

Figure 1: Generic example for proof of Prop. 4.3 and 4.5.

Proof. Let $i \in [K]$ s.t. $V_C(C_i) < 0$ i.e. $\mathrm{SEP}(C_i) < \mathrm{DISP}(C_i)$. We denote w_{sep}^l the minimal weight outing cluster C_i and w_{max} the maximal weight in subtree S_i of C_i i.e. $\mathrm{SEP}(C_i) \stackrel{def}{=} w_{sep}^l$ and $\mathrm{DISP}(C_i) \stackrel{def}{=} w_{max}$. Hence, $w_{sep}^l < w_{max}$. By cutting the cluster C_i on the edge with weight w_{max}, we define C_i^l and C_i^r resp. the left and right resulting clusters.

Let us look at $V_C(C_i^l)$. If $\mathrm{SEP}(C_i^l) \geq \mathrm{DISP}(C_i^l)$ then $V_C(C_i^l) \geq 0 \geq V_C(C_i)$ else $V_C(C_i^l) = \frac{\mathrm{SEP}(C_i^l)}{\mathrm{DISP}(C_i^l)} - 1$. The definition of the Separation as a minimum and our cut imply that

$$\mathrm{SEP}(C_i^l) \geq \min(\mathrm{SEP}(C_i), w_{max}) \geq \mathrm{SEP}(C_i).$$

Also the definition of the Dispersion as a maximum implies that $\mathrm{DISP}(C_i^l) \leq \mathrm{DISP}(C_i)$. Hence we get that $\frac{\mathrm{SEP}(C_i^l)}{\mathrm{DISP}(C_i^l)} - 1 \geq \frac{\mathrm{SEP}(C_i)}{\mathrm{DISP}(C_i)} - 1$ i.e. $V_C(C_i^l) \geq V_C(C_i)$ in this case too. The same reasoning holds for C_i^r showing that $V_C(C_i^r) \geq V_C(C_i)$. Finally,

$$\begin{aligned} DBCVI_{aftercut} &= \sum_{j \neq i} \frac{n_j}{N} V_C(C_j) + \frac{n_i^l}{N} V_C(C_i^l) + \frac{n_i^r}{N} V_C(C_i^r) \\ &\geq \sum_{j \neq i} \frac{n_j}{N} V_C(C_j) + \frac{n_i^l}{N} V_C(C_i) + \frac{n_i^r}{N} V_C(C_i) \\ &= DBCVI_{beforecut}. \end{aligned}$$

Hence cutting the edge with maximal weight in C_i improves the resulting DBCVI.

1.4 Proof of Prop. 4.4.

Proposition 4.4. Fate of positive V_C cluster I *Let $\mathcal{T}$ be an MST of the dissimilarity data graph and C a cluster s.t. $V_C(C) > 0$ and $\mathrm{SEP}(C) = s$. DBMSTClu does not cut an edge e of C with weight $w < s$ if both resulting clusters have at least one edge with weight greater than w.*

Proof. Let us consider clusters C_1 and C_2 resulting from the cut of edge e. Assume that in the associated subtree of C_1 (resp. C_2), there is an edge e_1 (resp. e_2) with a weight w_1 (resp. w_2) higher than w s.t. without loss of generality, $w_1 > w_2$. Since $V_C(C) > 0$, $s > w_1 > w_2 > w$. But cutting edge e implies that for $i \in \{1, 2\}$, $\mathrm{DISP}(C_i) > \mathrm{SEP}(C_i) = w$, and thus $V_C(C_i) < 0$. Cutting edge e would therefore mean to replace a cluster C s.t. $V_C(C) > 0$ by two clusters s.t. for $i \in \{1, 2\}$, $V_C(C_i) < 0$ which obviously decreases the current DBCVI. Thus, e does not get cut at this step of the algorithm.

1.5 Proof of Prop. 4.5.

PROPOSITION 4.5. FATE OF POSITIVE V_C CLUSTER II *Consider a partition with K clusters s.t. some cluster C_i, $i \in [K]$ with $V_C(C_i) > 0$ is in the setting of Fig. 1 i.e. cutting the heaviest edge e with weight w_{max} results in two clusters: the left (resp. right) cluster C_i^l (resp. C_i^r) with n_1 points (resp. n_2) s.t.* $\mathrm{DISP}(C_i^l) = d_1$, $\mathrm{SEP}(C_i^l) = w_{sep}^l$, $\mathrm{DISP}(C_i^r) = d_2$ *and* $\mathrm{SEP}(C_i^r) = w_{sep}^r$. *Assuming w.l.o.g. $w_{sep}^l > w_{sep}^r$, e gets cut iff:*

$$\frac{\left(\frac{n_1 d_1 + n_2 d_2}{n_1 + n_2}\right)}{w_{max}} \leq \frac{w_{max}}{w_{sep}^r}.$$

Proof. As $V_C(C_i) > 0$, there is $\mathrm{SEP}(C_i) = w_{sep}^r > w_{max}$. Then, the DBCVI before (K clusters) and after cut of w_{max} ($K+1$ clusters) are:

$$DBCVI_K = \sum_{j \neq i}^{K} V_C(C_j) + \frac{n_1 + n_2}{N}\left(1 - \frac{w_{max}}{w_{sep}^r}\right)$$

$$DBCVI_{K+1} = \sum_{j \neq i}^{K} V_C(C_j) + \frac{n_1}{N}\left(1 - \frac{d_1}{w_{max}}\right) + \frac{n_2}{N}\left(1 - \frac{d_2}{w_{max}}\right)$$

DBMSTClu cuts w_{max} iff $DBCVI_{K+1} \geq DBCVI_K$. So the result after simplification.

2 Complements on experiments.

Experiments were conducted using Python and scikit-learn library [1] on a single-thread process on an intel processor based node.

2.1 Safety of the sketching. Fig. 2 shows another result on a synthetic dataset: three blobs generated from three Gaussian distributions. With the three blobs, each method SEMST, DBSCAN and DBMSTClu performs well: they all manage to retrieve three clusters.

Quantitative results for the three synthetic datasets are shown in Table 1: the achieved silhouette coefficient, Adjusted Rand Index (ARI) and DBCVI. For all the indices, the higher, the better. Silhouette coefficient (between -1 and 1) is used to measure a clustering partition without any external information. For DBSCAN it is computed by considering noise points as singletons. We see that this measure is not very suitable for nonconvex clusters like noisy circles or moons. The ARI (between 0 and 1) measures the similarity between the experimental clustering partition and the known groundtruth. DBSCAN and DBMSTClu give similar almost optimal results. Finally, the obtained DBCVIs are consistent, since the best ones are reached for DBMSTClu.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, 12 (2011), pp. 2825–2830.

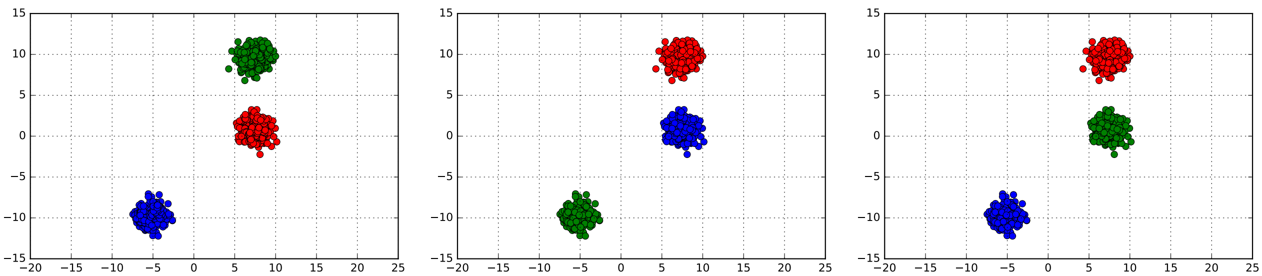

Figure 2: Three blobs: SEMST, DBSCAN ($\epsilon = 1.4$, $minPts = 5$), DBMSTClu with an approximate MST.

	Silhouette coeff.			Adjusted Rand Index			DBCVI		
SEMST	**0.84**	**0.16**	-0.12	**1**	0	0	**0.84**	0.001	0.06
DBSCAN	**0.84**	0.02	**0.26**	**1**	**0.99**	**0.99**	**0.84**	-0.26	**0.15**
DBMSTClu	**0.84**	-0.26	**0.26**	**1**	**0.99**	**0.99**	**0.84**	**0.18**	**0.15**

Table 1: Silhouette coefficients, Adjusted Rand Index and DBCVI for the blobs, noisy circles and noisy moons datasets with SEMST, DBSCAN and DBMSTClu.