Streaming and Massively Parallel Algorithms for Euclidean Max-Cut

Nicolas Menand [1] Erik Waingarten [1]

[1]University of Pennsylvania

Background

Given dataset $X \subseteq \mathbb{R}^d$, the Euclidean Max-Cut of X asks to compute:

$$\max_{S \subset X} \sum_{x \in S} \sum_{y \in X/S} \|x - y\|_2$$

Massively Parallel Computation: Points distributed across m machines each with $O(dn^\alpha)$ words of memory for $\alpha > 1$. Each Machine can send / receive $O(dn^\alpha)$ words each round.

Dynamic Geometric Streams [Ind04]: Dataset $X \subset [\Delta]^d$ is presented as an arbitrary sequence of insertions & deletions of points $x \in X$.

Prior work [CJK23] gives a dynamic streaming algorithm which finds a $(1+\varepsilon)$ approximation in $\mathrm{poly}(d\log(\Delta)/\varepsilon)$ space, but cannot determine which points lie in S. The prior best known streaming algorithm that provides oracle-access to S requires space exponential in $1/\varepsilon$.

Parallel and Subsampled Greedy Assignment

Consider $X = \{x_1, \dots, x_n\}$ of n points in Euclidean space. We assign points in a greedy fashion similar to [MS08]:

- Each point x_i generates a timeline $\mathbf{A}_i \in \{0,1\}^{t_e}$: at each time $t = 1, \dots, t_e$, it "activates" by sampling $\mathbf{A}_{i,t} \sim \mathrm{Ber}(\boldsymbol{w}_{i,t})$ with probability proportional to it's weight and $1/t$.
- Each x_i samples a mask $\mathbf{K}_i \in \{0,1\}^{t_e}$: at each time t, x_i is "kept" by sampling $\mathbf{K}_{i,t} \sim \mathrm{Ber}(\gamma_t)$, where γ_t is 1 at $t \le t_0$ and γ/t for $t > t_0$.

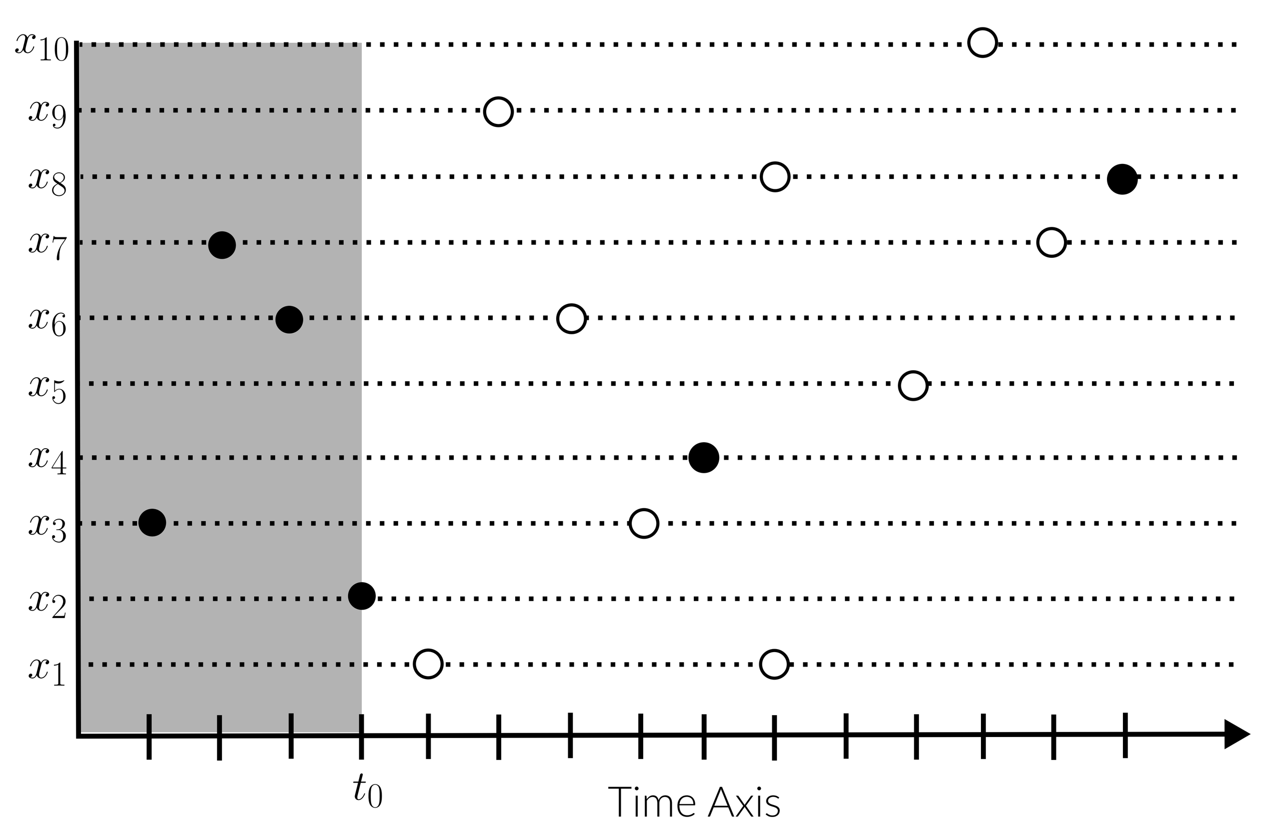

Fig 1. Assignment timeline.

Greedy Cut: We encode a partial cut at each time step $t \in [t_e]$ as a $[n] \times \{0,1\}$ matrix z^t with entries in $[0,1]$.

Each point assigns itself to inside/outside the cut the moment it is first activated.

- Points activated by time t_0 try all cut assignments.
- Points activated after t_0 are assigned greedily.

Greedy assignment: If $\mathbf{A}_{i,t} = 1$ for $t > t_0$, we assign x_i to the cut S if

$$\sum_{j=1}^{n} d(x_i, x_j) \left(\frac{1}{t-1} \sum_{\ell=1}^{t-1} \frac{\mathbf{A}_{j,\ell} \cdot \mathbf{K}_{j,\ell}}{\boldsymbol{w}_{j,\ell} \cdot \gamma_\ell} \right) \boldsymbol{z}_{j,0}^{t-1} \ge \sum_{j=1}^{n} d(x_i, x_j) \left(\frac{1}{t-1} \sum_{\ell=1}^{t-1} \frac{\mathbf{A}_{j,\ell} \cdot \mathbf{K}_{j,\ell}}{\boldsymbol{w}_{j,\ell} \cdot \gamma_\ell} \right) \boldsymbol{z}_{j,1}^{t-1}$$

and otherwise assign it outside.

Theorem 1: [MPC] *There is a $O(1)$-round fully-scalable MPC algorithm which outputs a $(1+\varepsilon)$-approximate Euclidean max-cut using $O(nd) + n \cdot \mathrm{poly}(\log n/\varepsilon)$ total space.*

MPC Algorithm:

1. Use shared randomness to compute and share cascaded $\ell_1(\ell_2)$-sketches.
2. Generate $\mathbf{A}_i$ and $\mathbf{K}_i$ for each x_i using the sketched weight w_i.
3. If $\exists t \le t_e$ where $\mathbf{A}_{i,t} \cdot \mathbf{K}_{i,t} = 1$, communicate (x_i, w_i, t) to all other machines.
4. Estimate the best initial assignment of the points activated by t_0 (using a few additional samples)
5. Assign each x_i by maximizing the weighted contribution to the activated and kept points.

Analysis

Want to prove:

$$\mathbf{E}\left[f(\boldsymbol{z}^{t_e}) - f(z^*) \right] \le \varepsilon \sum_{i=1}^{n} \sum_{j=1}^{n} d(x_i, x_j).$$

Proof Sketch: Like [MS08], define a "fictitious cut" solely for analysis

Fictitious Cut: Sequence of cut matrices $\hat{\boldsymbol{z}}^0, \dots \hat{\boldsymbol{z}}^{t_e}$.

- $\hat{\boldsymbol{z}}^0$ is set to the assignment corresponding to the optimal max-cut z^*.
- When x_i is first activated, set $\hat{\boldsymbol{z}}_i^t = z_i^t$.
- When x_i is not activated, update it to:

$$\hat{\boldsymbol{z}}_i^t = \frac{1}{1 - \boldsymbol{w}_i^t} \left(\frac{t-1}{t} \cdot \hat{\boldsymbol{z}}_i^{t-1} + \frac{1}{t} \cdot \boldsymbol{g}_i^t - \boldsymbol{w}_i^t \cdot \boldsymbol{g}_i^t \right)$$

where $\boldsymbol{g}_i^t$ is either $(1,0)$ or $(0,1)$, based on the greedy decision had x_i been activated at time t

Rewrite in terms of fictitious cut

$$\mathbf{E}\left[f(\boldsymbol{z}^{t_e}) - f(z^*) \right] \le \mathbf{E}\left[f(\hat{\boldsymbol{z}}^{t_e}) - f(\hat{\boldsymbol{z}}^0) \right] = \sum_{t=t_0+1}^{t_e} \mathbf{E}\left[f(\hat{\boldsymbol{z}}^t) - f(\hat{\boldsymbol{z}}^{t-1}) \right]$$

Further decompose the change in cut value per time step into sum of terms representing the change in cut value when reassigning a point and the change for simultaneously updating points.

Fictitious cut defined so as to smooth the error between the true and estimated change when (re)assigning a point, forming a martingale which decays by a factor of $\frac{t-1}{t}$ each time step. Greedy decision always maximizes the estimated change.

This allows us to bound the expression by:

$$\mathbf{E}\left[f(\boldsymbol{z}^{t_e}) - f(z^*) \right] \le \left(\frac{\log(t_e)}{\sqrt{\gamma}} + \frac{\sqrt{\gamma}}{t_0} + \frac{1}{t_0} \right) \sum_{i=1}^{n} \sum_{j=1}^{n} d(x_i, x_j).$$

For a $1+\varepsilon$ approximation, this requires

$$\gamma \ge \log(t_e)/\varepsilon^2 \quad \text{and} \quad t_0 \ge \sqrt{\gamma}/\varepsilon^2 \quad \text{and} \quad t_e \ge n/\varepsilon$$

Total Space: Roughly $O(t_0 + \log(t_e))$ points stored in expectation, gives space complexity:

$$\mathrm{poly}(d\log(n)/\varepsilon)$$

Theorem 2: [Dynamic Streams] *There is a dynamic streaming algorithm using $\mathrm{poly}(d\log\Delta/\varepsilon)$ space which provides oracle access to a $(1+\varepsilon)$-approximate Euclidean max-cut.*

Challenge: Cannot compute weight w_i and timelines $\mathbf{A}_i$ until end of the stream.

Solution: Geometric sampling sketches [CJK23] can sample $\boldsymbol{x}_i \sim X$ with probability proportional to w_i. Use these for the simultaneously "activated" and "kept" points.

During Stream

- Sample one mask $\mathbf{K} \in \{0,1\}^{t_e}$.
- Draw $\mathrm{poly}(d\log(\delta)/\varepsilon^2)$ geometric sampling sketches from the stream.

After Stream

- If $\mathbf{K}_t = 0$, no point is activated and kept at that time. If $\mathbf{K}_t = 1$, use one of the geometric samples $\boldsymbol{x}_i$ and activate it by setting $\mathbf{A}_{i,t} = 1$.
- Estimate the best initial assignment of the points activated by t_0 (using a few additional geometric samples)
- On a query x_j, use the sketch to estimate the weight $\boldsymbol{w}_j$ and generate the remainder of the timeline $\mathbf{A}_j$. Then output the greedy decision for x_j

Modified Analysis: Activations are no longer independent, instead either independent if $\mathbf{K}_t = 0$ or "negatively correlated" if $\mathbf{K}_t = 1$. The changes in cut value in the analysis are largest when simultaneous activations occur, hence the theorems hold under the modified activation timeline.

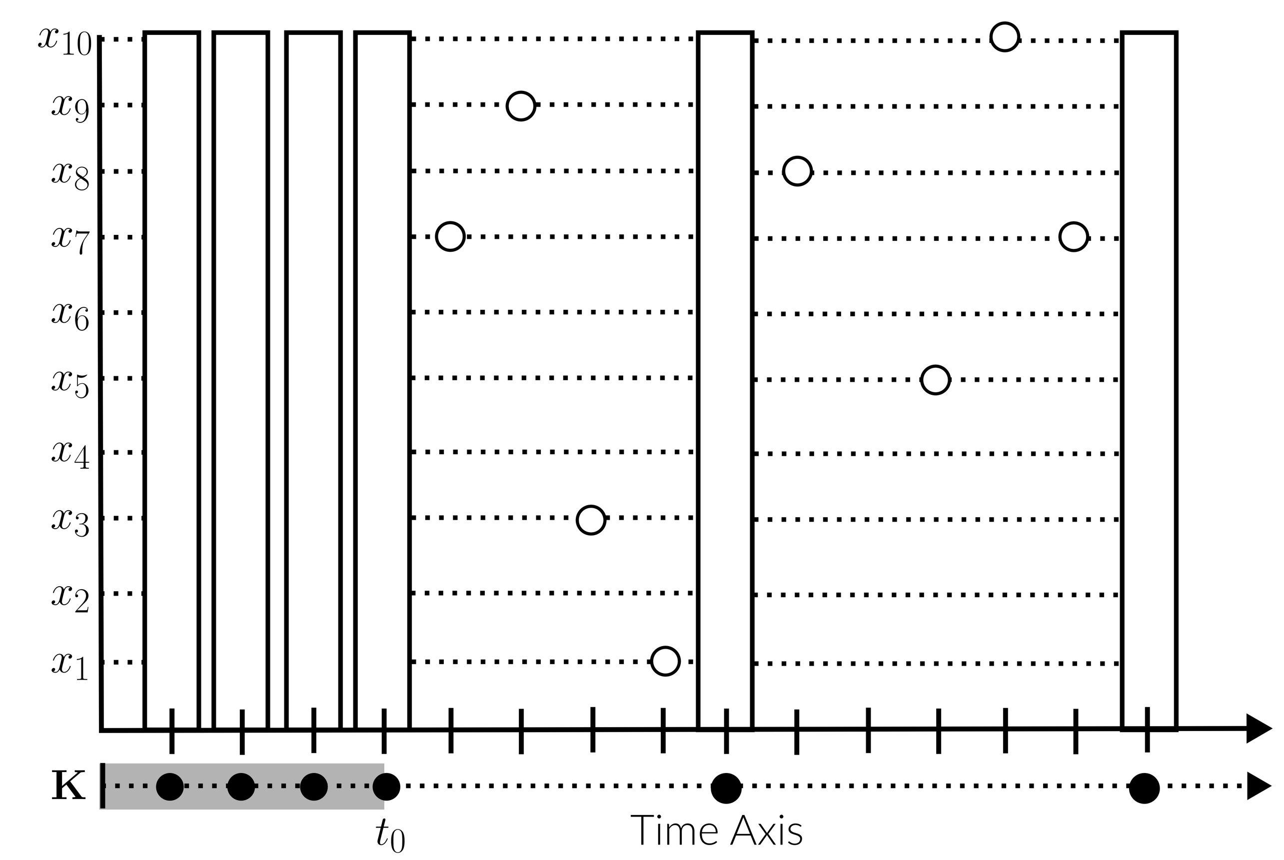

Fig 2. Modified Assignment timeline for dynamic streams.

References

[CJK23] Xiaoyu Chen, Shaofeng H.-C. Jiang, and Robert Krauthgamer. Streaming euclidean max-cut: Dimension vs data reduction. In *Proceedings of the 55th ACM Symposium on the Theory of Computing (STOC '2023)*, pages 170–182, 2023.

[Ind04] Piotr Indyk. Algorithms for dynamic geometric problems over data streams. In *Proceedings of the 36th ACM Symposium on the Theory of Computing (STOC '2004)*, pages 373–380, 2004.

[MS08] Claire Mathieu and Warren Schudy. Yet another algorithm for dense max cut: go greedy. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA '2008)*, 2008.