Learning-Augmented Streaming Algorithms for Correlation Clustering

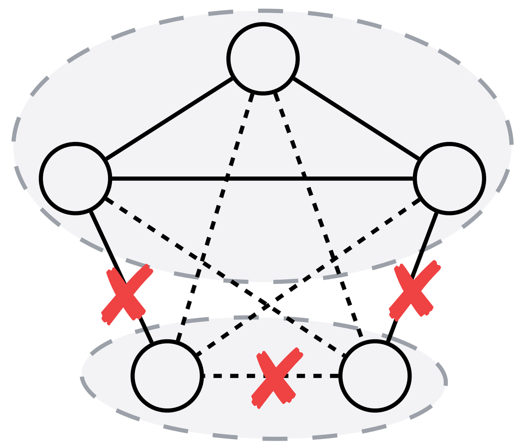
Yinhao Dong[1] Shan Jiang[1] Shi Li[2,3] Pan Peng[1]

[1]University of Science and Technology of China [2]Nanjing University [3]New Cornerstone Science Laboratory

Problem Setting

Correlation Clustering

- **Input:** Graph $G = (V, E = E^+ \cup E^-)$ with each edge labeled as either positive $(+)$ or negative $(-)$
- **Output:** Clustering/Partition of V
- **Goal:** Minimize the number of *disagreements*:
  - # of positive $(+)$ edges across different clusters
  - # of negative $(-)$ edges within same clusters



Graph Streaming Model

- G is presented as a sequence of edge insertions and deletions
  - Insertion-Only Streams: consists of edge insertion only
  - Dynamic Streams: has both insertions and deletions
- **Observation:** Outputting the clustering requires $\Omega(n)$ space
- **Goal:** In one pass, using small space (usually $\tilde{O}(n)$ space), compute the clustering

Learning-Augmented Algorithms

- The algorithm has access to a **learned oracle** providing a certain type of **predictions** about the input instance
- **Goals:**
  - **Consistency:** Better performance when the input has some "learnable" pattern (i.e., under high prediction quality)
  - **Robustness:** Similar worst-case guarantee as the best-known classical algorithms (regardless of the prediction quality)

Our Prediction Model

- Oracle access to **pairwise distance** $d_{uv} \in [0,1]$ between any $u, v \in V$
- Arises in many scenarios: multiple graphs on the same vertex set
  - Healthcare: disease network, provider network, clinical trial network
  - Biology: protein-protein interaction network, gene co-expression network, signaling pathway network
  - Temporal Graphs: same vertices, different edges over time
- **Observation:** Two vertices similar in one network are likely similar in another – cluster structure can thus be extracted!

β-Level Predictor ($\beta \geq 1$)

1. (triangle ineq.) $d_{uw} + d_{vw} \geq d_{uw},\ \forall u, v, w \in V$
2. $\sum_{(u,v)\in E^+} d_{uv} + \sum_{(u,v)\in E^-}(1 - d_{uv}) \leq \beta \cdot \mathrm{OPT}$

Natural LP of Correlation Clustering

$$\begin{aligned} \min \quad & \sum_{(u,v)\in E^+} x_{uv} + \sum_{(u,v)\in E^-} (1 - x_{uv}) \\ \text{s.t.} \quad & x_{uw} + x_{wv} \geq x_{uv} \quad \forall u, v, w \in V \\ & x_{uv} \in [0,1] \quad \forall (u,v) \in E \end{aligned}$$

Our Results

Setting	Best-Known Approx-Space Trade-offs (without Predictions)	Our Results (with Predictions)
Complete Graphs	$(3+\varepsilon)$-approx, $\tilde{O}(\varepsilon^{-1} n)$ total space [CKL+24]	$(\min\{2.06\beta, 3\} + \varepsilon)$-approx, $\tilde{O}(\varepsilon^{-2} n)$ total space
	$(\alpha_{\mathsf{BEST}} + \varepsilon)$-approx, $\tilde{O}(\varepsilon^{-2} n)$ space during the stream, $\mathrm{poly}(n)$ space for post-processing [AKP25]	
General Graphs	$O(\log \lvert E^- \rvert)$-approx, $\tilde{O}(\varepsilon^{-2} n + \lvert E^- \rvert)$ total space [ACG+21]	$O(\beta \log \lvert E^- \rvert)$-approx, $\tilde{O}(\varepsilon^{-2} n)$ total space

- α-approx: $\mathrm{OPT} \leq \mathrm{ALG} \leq \alpha \cdot \mathrm{OPT}$
- α_{BEST}: best approx ratio of any poly-time classical alg for Correlation Clustering

Our Streaming Algorithm for Complete Graphs

- **Building Blocks**: Two pivot-based algorithms. In each iteration, randomly pick a pivot p from the current graph, construct a cluster $C \ni p$, and add the remaining vertices v to C:
  - 3-Approx Combinatorial Algorithm (Pivot) [ACN08]: iff $(p, v) \in E^+$
  - 2.06-Approx LP Rounding Algorithm [CMSY15]: with prob. $1 - f(x_{pv})$
- **Challenge:** Solving LPs in streaming is difficult!

Our Algorithm with Predictions

- During the Stream: Maintain a truncated subgraph G' of G (refer to [CKL+24])
- After the Stream (Post-Processing):
  - Run the 3-approx Pivot algorithm on G', obtain clustering $\mathcal{C}_1$
  - Run the 2.06-approx LP rounding algorithm on G' (use predictions d_{uv} to replace LP solution x_{uv}), obtain clustering $\mathcal{C}_2$
  - Output the clustering with the lower cost between $\mathcal{C}_1$ and $\mathcal{C}_2$

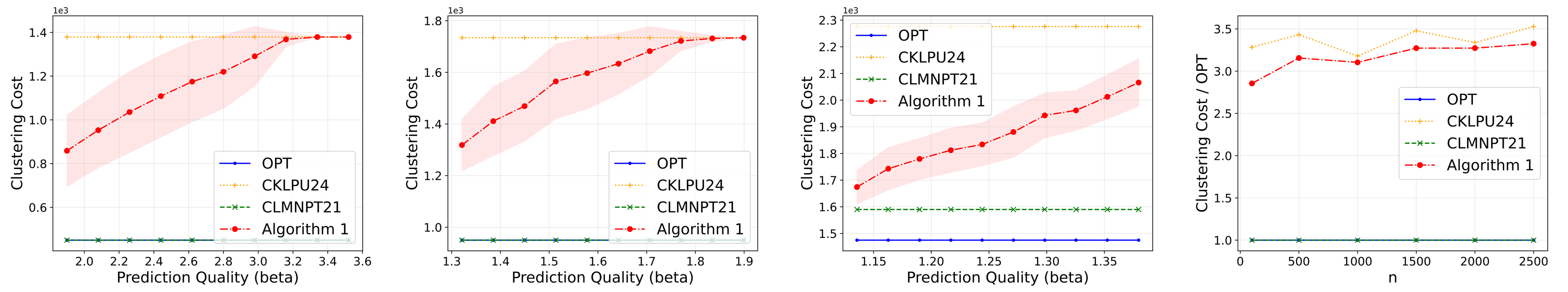
Our Streaming Algorithm for General Graphs

Our Algorithm with Predictions

- During the Stream: Maintain a spectral sparsifier H^+ for $G^+ = (V, E^+)$
- After the Stream (Post-Processing): Perform ball-growing on H^+
  - In each iteration:
    1. Pick an arbitrary vertex u from the current graph
    2. Grow a ball $B(u, r_u)$ using predictions d_{uv} as distance metric, until a certain condition is satisfied
    3. Remove the ball from the current graph
  - Output the balls as the final clustering
- Unlike [ACG+21], our algorithm does not solve an LP and therefore does not require storing E^- during the stream.

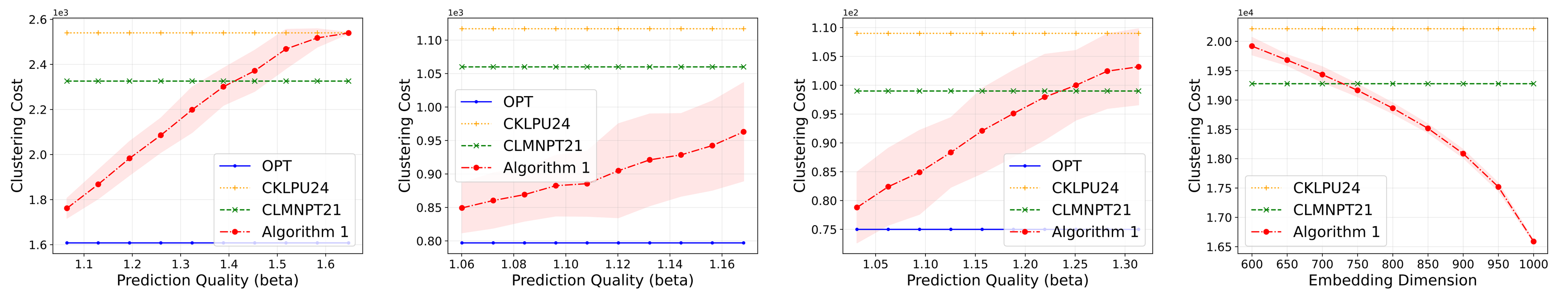
Experiments

- **Datasets**
  - Synthetic Datasets: generated from the Stochastic Block Model (SBM) with parameter $p > 0.5$
  - Real-World Datasets: EmailCore, Facebook, LastFM, DBLP from Stanford SNAP Collection
- **Predictors:** Noisy predictor, Spectral embedding, Binary classifier
- **Baselines**
  - [CKL+24]: $(3+\varepsilon)$-approx streaming algorithm without predictions
  - [CLM+21]: based on agreement decomposition, 701-approx in theory, performs well on certain types of graphs in practice
- **Performance on Synthetic Datasets**



(a) $p = 0.9$, vary β (b) $p = 0.8$, vary β (c) $p = 0.7$, vary β (d) $\beta \approx 2.8$, vary n

- **Performance on Real-World Datasets**



(a) FB 0, vary β (b) FB 414, vary β (c) FB 3980, vary β (d) EmailCore, vary d

Takeaways

1. Better performance under good predictions; robust under bad predictions
2. Empirical performance much better than theoretical guarantee

Open Problems

- Better approx ratio in $\tilde{O}(n)$ total space for complete graphs?
- Better approx-space trade-off for general graphs?

References

[ACG+21] Kook Jin Ahn, Graham Cormode, Sudipto Guha, Andrew McGregor, and Anthony Wirth. Correlation Clustering in Data Streams. *Algorithmica*, 83(7):1980–2017, 2021. Conference version in *ICML*, 2015.

[ACN08] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating Inconsistent Information: Ranking and Clustering. *J. ACM*, 55(5):23:1–23:27, 2008.

[AKP25] Sepehr Assadi, Sanjeev Khanna, and Aaron Putterman. Correlation Clustering and (De)Sparsification: Graph Sketches Can Match Classical Algorithms. In *STOC*, 2025.

[CKL+24] Mélanie Cambus, Fabian Kuhn, Etna Lindy, Shreyas Pai, and Jara Uitto. A $(3+\varepsilon)$-Approximate Correlation Clustering Algorithm in Dynamic Streams. In *SODA*, 2024.

[CLM+21] Vincent Cohen-Addad, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub Tarnawski. Correlation Clustering in Constant Many Parallel Rounds. In *ICML*, 2021.

[CMSY15] Shuchi Chawla, Konstantin Makarychev, Tselil Schramm, and Grigory Yaroslavtsev. Near Optimal LP Rounding Algorithm for Correlation Clustering on Complete and Complete k-partite Graphs. In *STOC*, 2015.