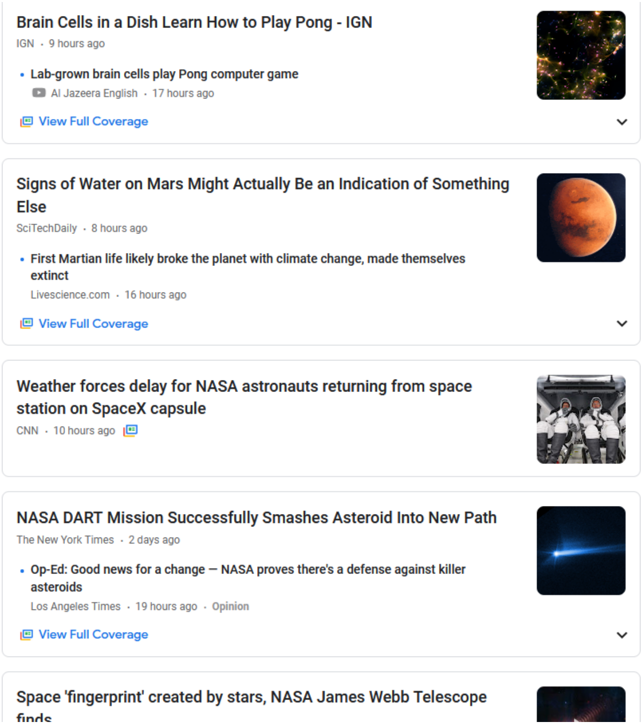
Robust Online Correlation Clustering

Rudy Zhou

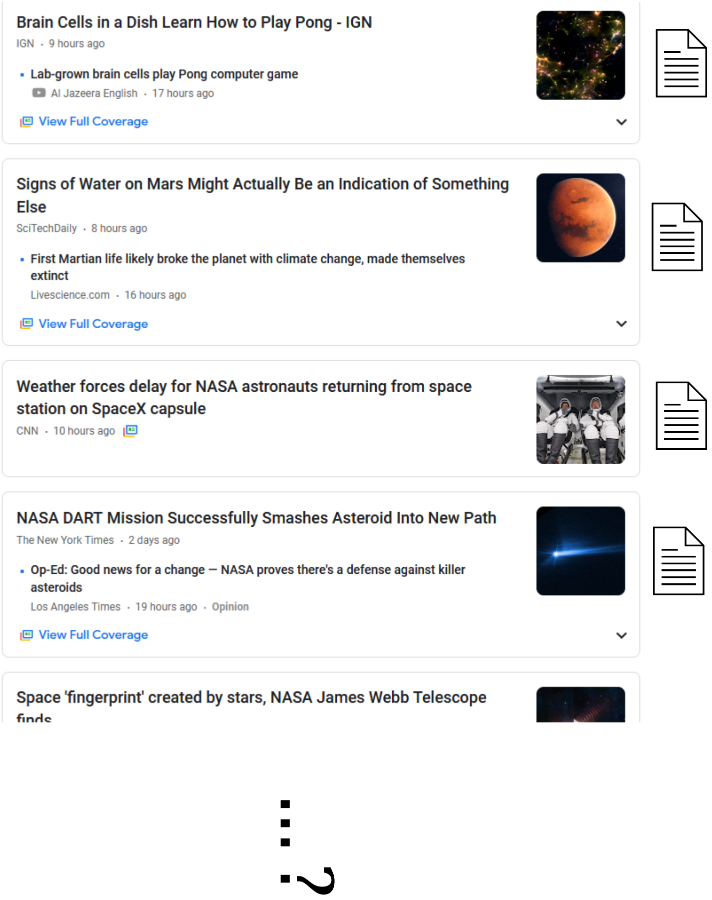
Carnegie Mellon University

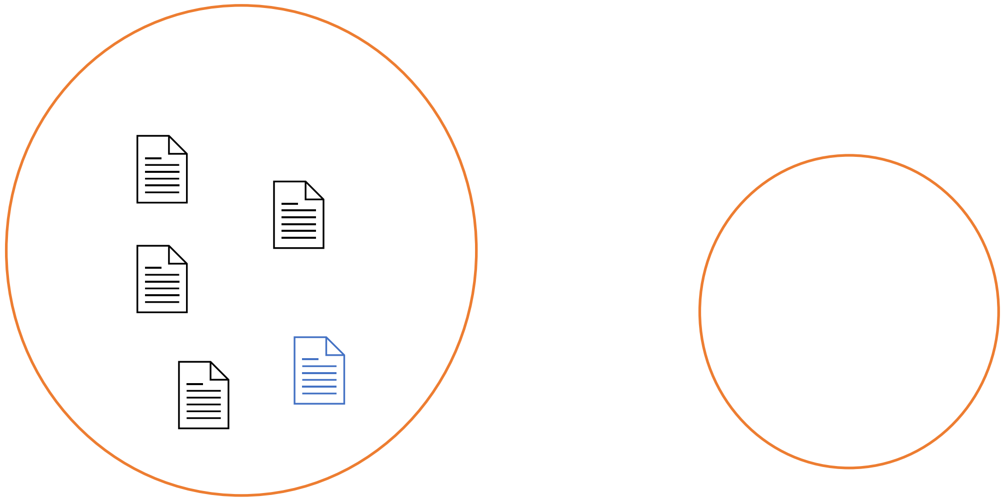
Silvio Lattanzi, Benjamin Moseley, Sergei Vassilvitskii, Yuyan Wang, Rudy Zhou
Robust Online Correlation Clustering
Neural Information Processing Systems (NeurIPS) 2021.

Motivation: Google News

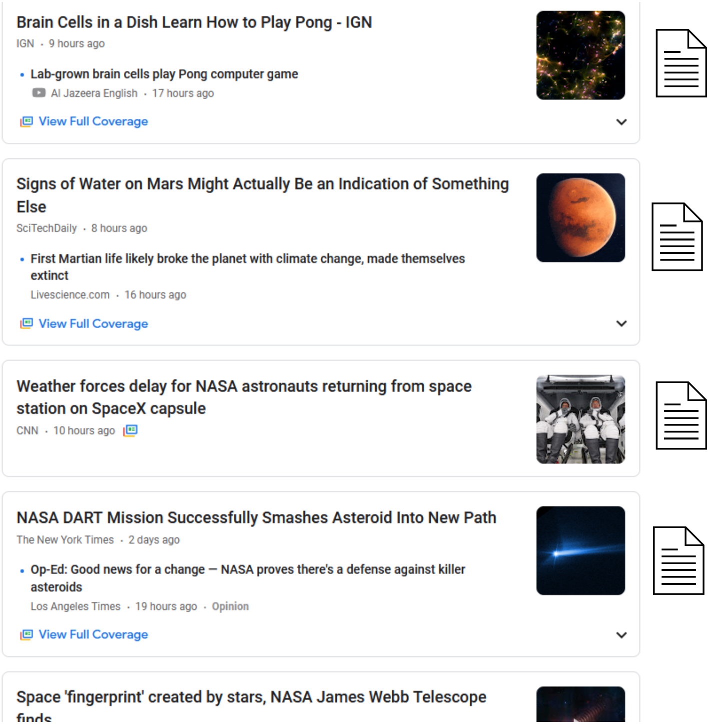


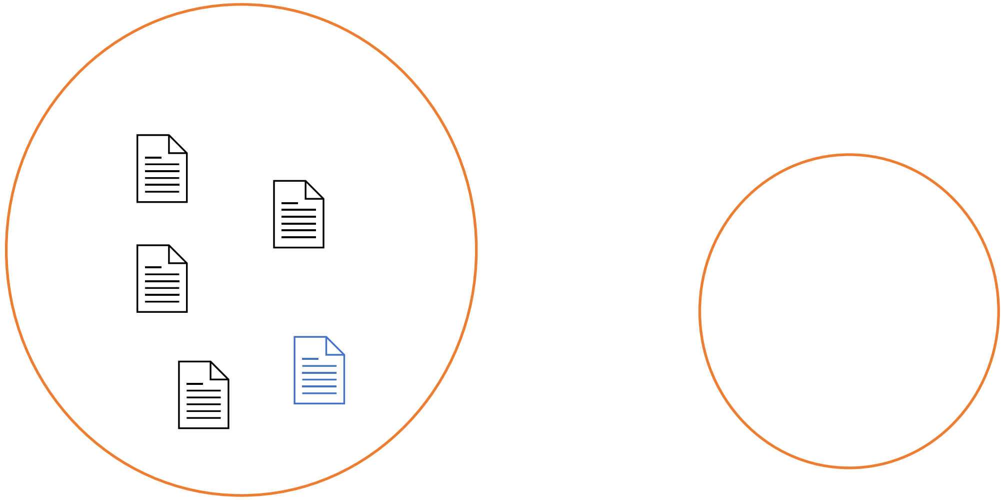
Motivation: Google News



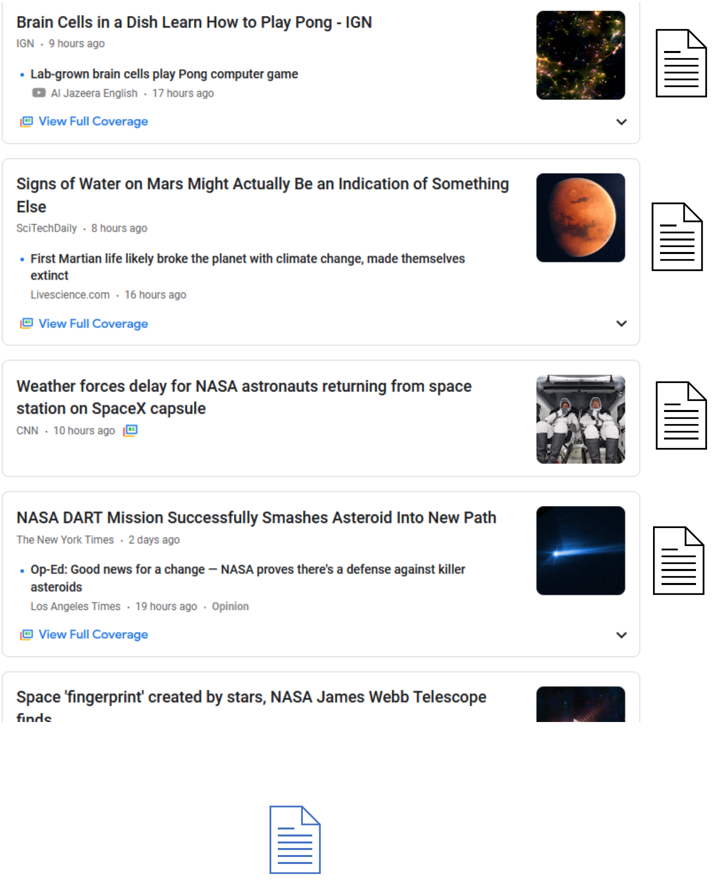


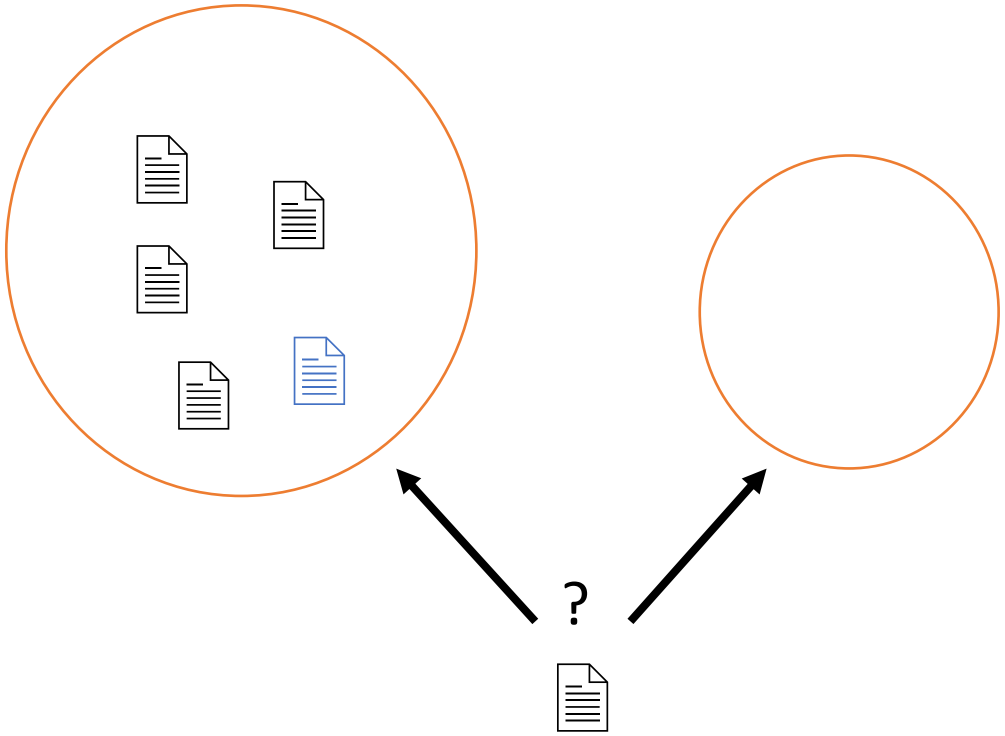
Motivation: Google News





Motivation: Google News



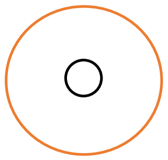


Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster

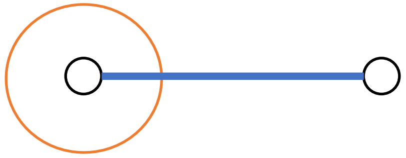
Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



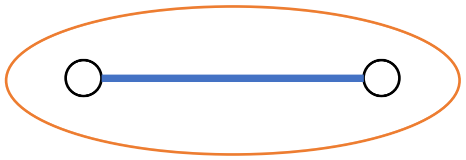
Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



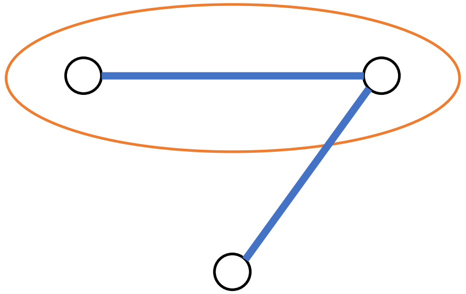
Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



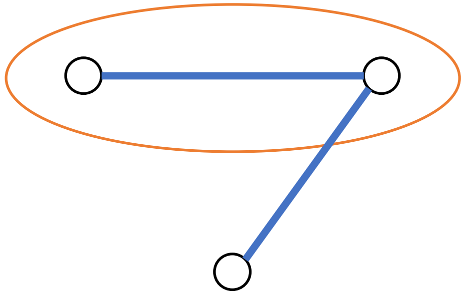
Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Online Correlation Clustering

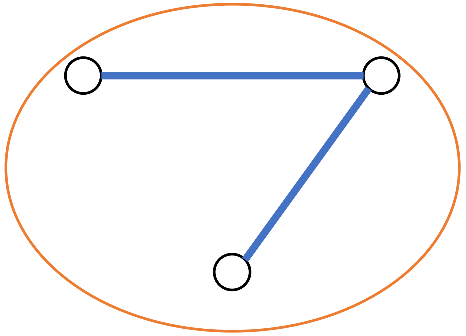
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(disagreements) = \#(edges\ across) + \#(non - edges\ within)$

Online Correlation Clustering

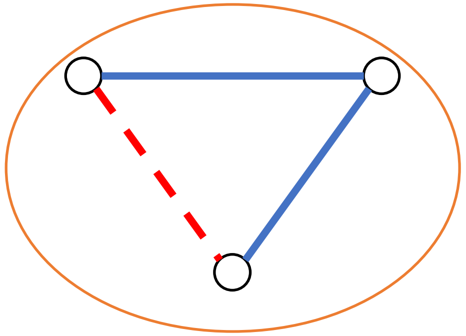
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(\textcolor{red}{disagreements}) = \#(edges\ across) + \#(non - edges\ within)$

Online Correlation Clustering

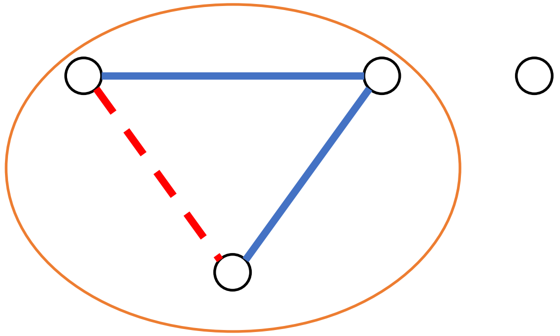
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(disagreements) = \#(edges\ across) + \#(non - edges\ within)$

Online Correlation Clustering

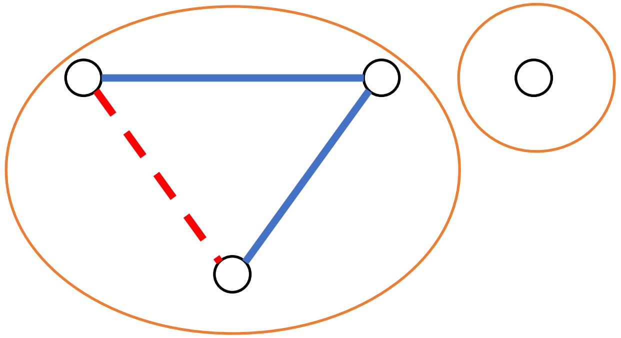
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(\textcolor{red}{disagreements}) = \#(edges\ across) + \#(non - edges\ within)$

Online Correlation Clustering

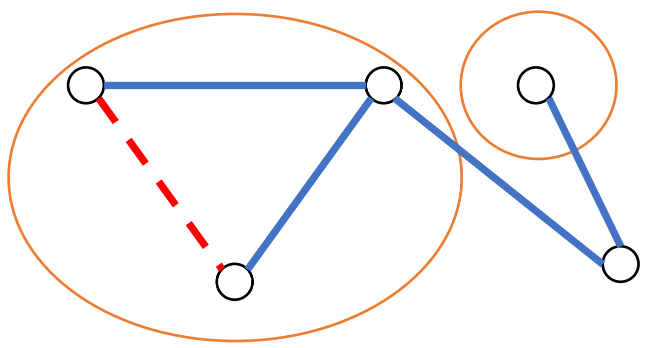
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(disagreements) = \#(edges\ across) + \#(non - edges\ within)$

Online Correlation Clustering

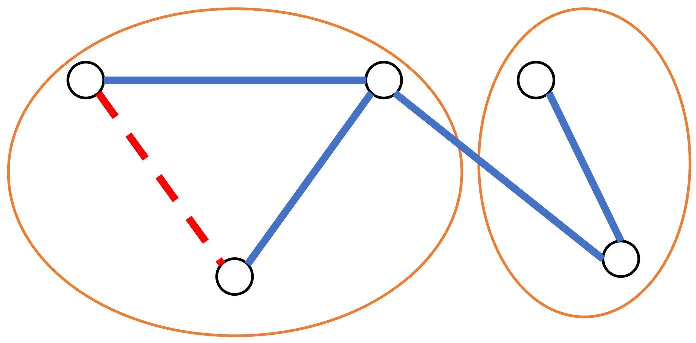
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(\textcolor{red}{disagreements}) = \#(edges\ across) + \#(non - edges\ within)$

Online Correlation Clustering

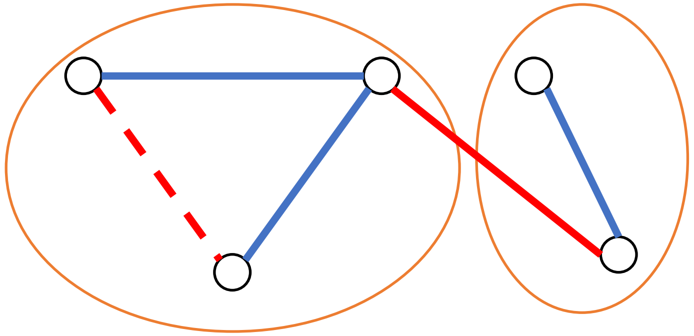
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(\color{red}{disagreements}) = \#(edges\ across) + \#(non-edges\ within)$

Online Correlation Clustering

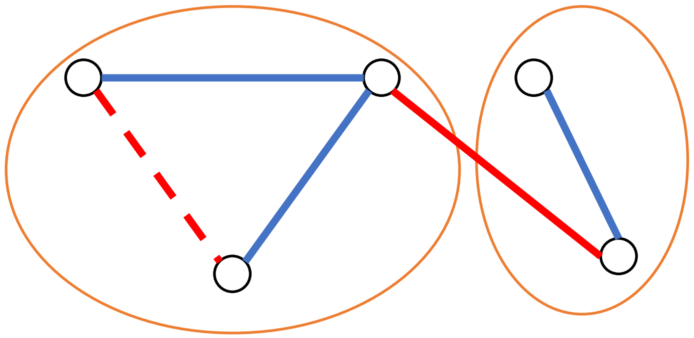
- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



Minimize $\#(disagreements) = \#(edges\ across) + \#(non-edges\ within)$

Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster



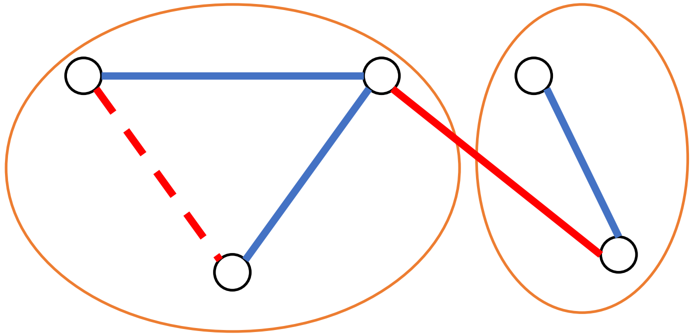
Minimize $\#(\textcolor{red}{disagreements}) = \#(edges\ across) + \#(non - edges\ within)$

…compared to optimal offline clustering that knows entire graph

Online Correlation Clustering

- Vertices arrive online; reveal edges to previous arrivals
- Assign vertex to existing cluster or make new singleton cluster

Competitive Ratio: An algorithm is ***c*-competitive** if for any input graph and arrival order:
$\#(disagreements\ by\ ALG) \leq c \cdot \#(disagreements\ by\ OPT)$

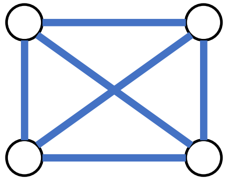


Minimize $\#(\color{red}{disagreements}) = \#(edges\ across) + \#(non-edges\ within)$

…compared to optimal offline clustering that knows entire graph

Prior Work

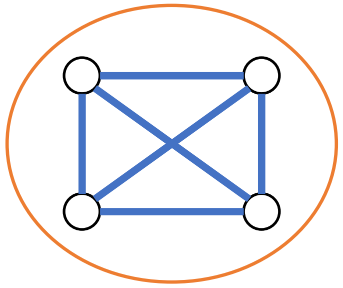
- Every online algorithm is $\Omega(\#\ vertices)$-competitive



C. Mathieu, O. Sankur, and W. Schudy. *Online correlation clustering*. STACS 2010

Prior Work

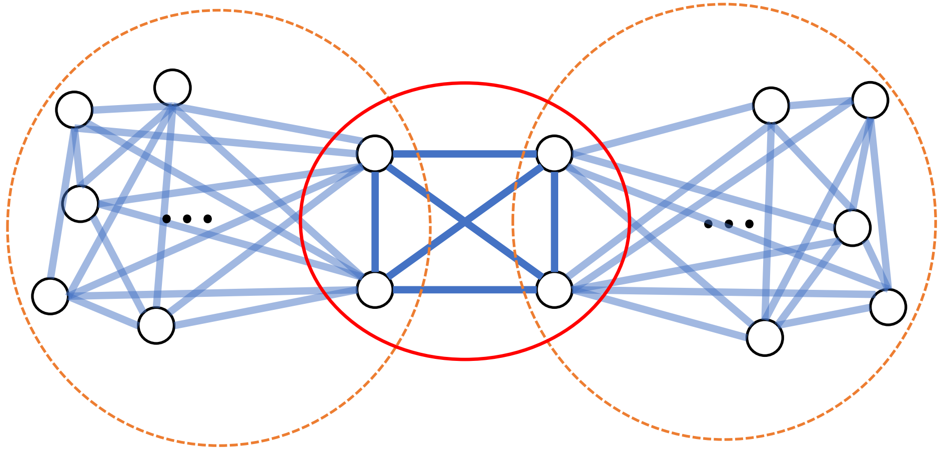
- Every online algorithm is $\Omega(\#\ vertices)$-competitive



C. Mathieu, O. Sankur, and W. Schudy. *Online correlation clustering*. STACS 2010

Prior Work

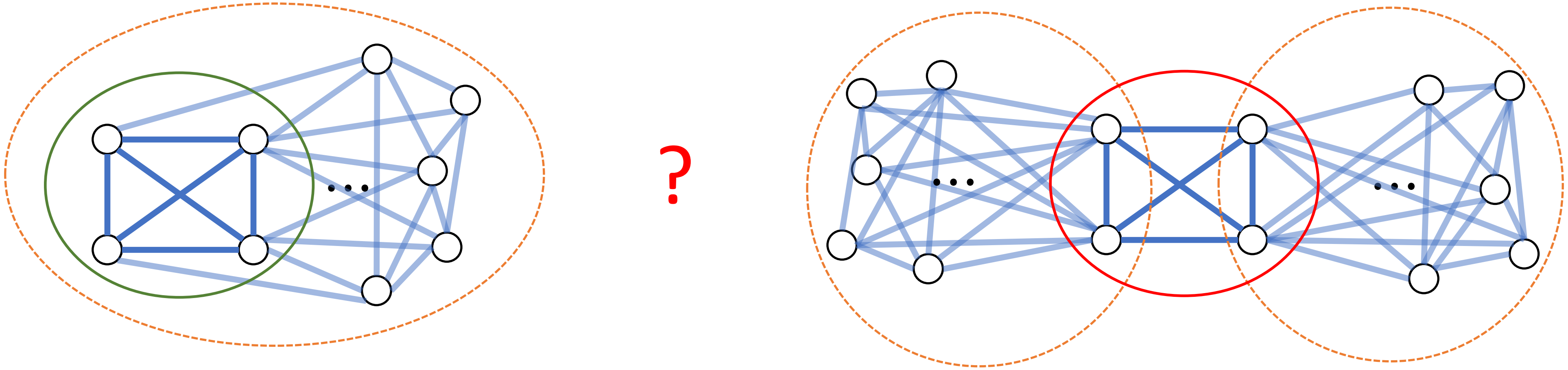
- Every online algorithm is $\Omega(\#\ vertices)$-competitive



C. Mathieu, O. Sankur, and W. Schudy. *Online correlation clustering*. STACS 2010

Prior Work

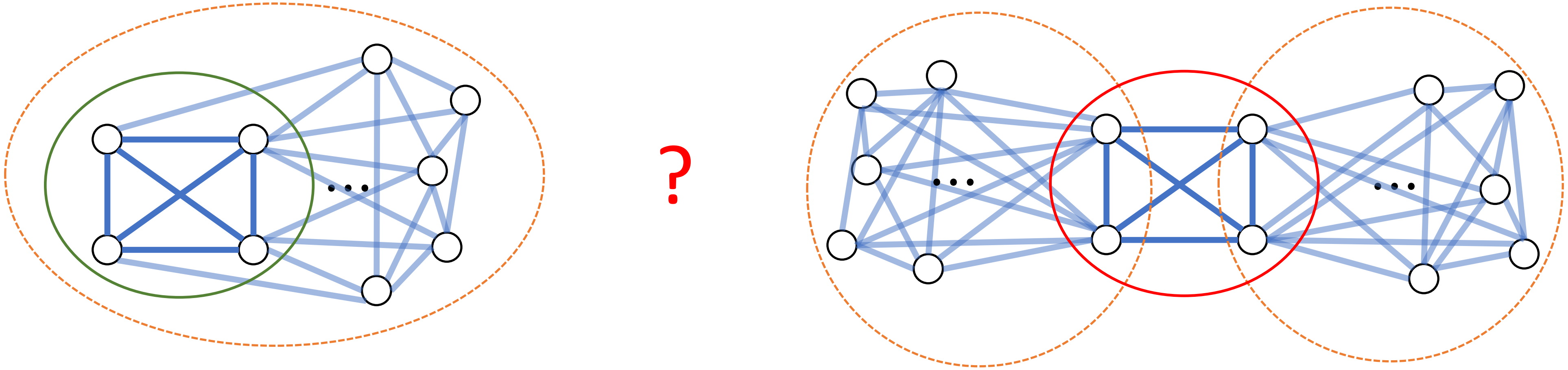
- Every online algorithm is $\Omega(\#\ vertices)$-competitive



C. Mathieu, O. Sankur, and W. Schudy. *Online correlation clustering*. STACS 2010

Prior Work

- Every online algorithm is $\Omega(\#\ vertices)$-competitive



How to overcome online lower bound?

C. Mathieu, O. Sankur, and W. Schudy. *Online correlation clustering*. STACS 2010

How to overcome online lower bound?

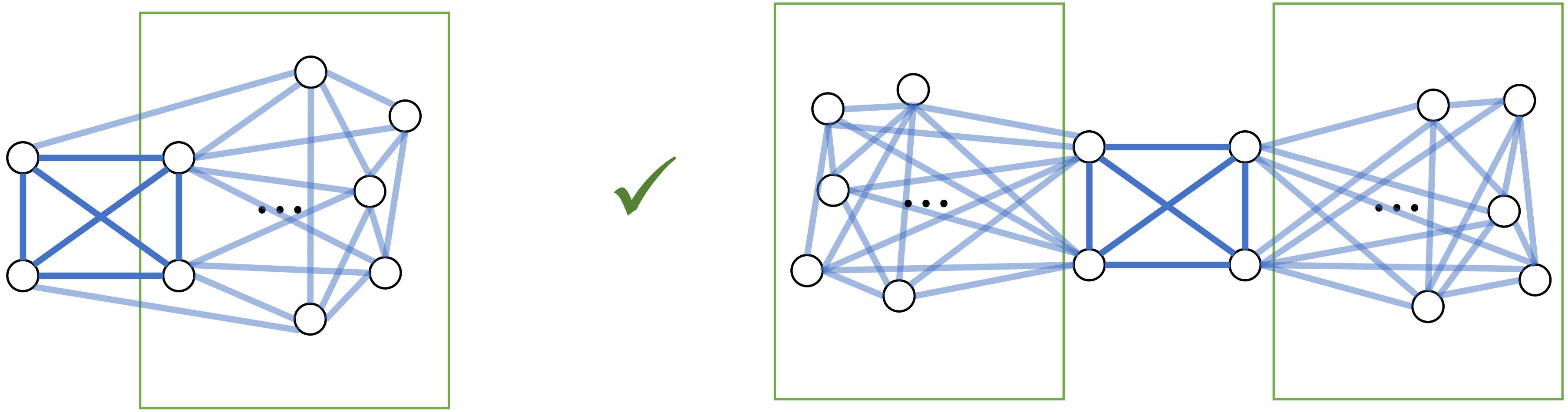
- Augment algorithm with historical data
- Introduce new online model
  - Historical data should be (partially) related to online arrivals
  - No other assumptions on online arrival order

Semi-Online Model

- Two phases: Offline and Online
- **Offline Phase:** corrupted random subgraph of ϵ-fraction of vertices revealed offline
  - Adversary chooses α-fraction of vertices
  - $(\epsilon - \alpha)$-fraction of remaining vertices are randomly chosen
- **Online Phase:** remaining vertices arrive online

Semi-Online Model

- Two phases: Offline and Online
- **Offline Phase:** corrupted random subgraph of ϵ-fraction of vertices revealed offline
  - Adversary chooses α-fraction of vertices
  - $(\epsilon - \alpha)$-fraction of remaining vertices are randomly chosen
- **Online Phase:** remaining vertices arrive online



Our Contribution

- Introduce semi-online model for sequential decision-making problems

Main Theorem: We design an algorithm for semi-online correlation clustering that is $O(\frac{1}{\epsilon - \alpha})$ – competitive*.

* assuming $\alpha \leq \frac{\epsilon}{2}$

Our Contribution

- Introduce semi-online model for sequential decision-making problems

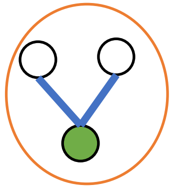
Main Theorem: We design an algorithm for semi-online correlation clustering that is $O(\frac{1}{\epsilon - \alpha})$ – competitive*.

- … and any semi-online algorithm must be $\Omega(\frac{1}{\epsilon - \alpha})$ – competitive*
- … and the theoretical results are predictive of practice

* assuming $\alpha \leq \frac{\epsilon}{2}$

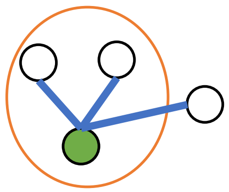
Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster



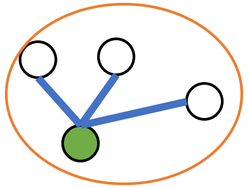
Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster



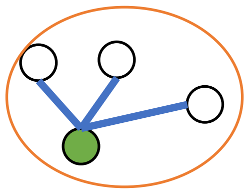
Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster

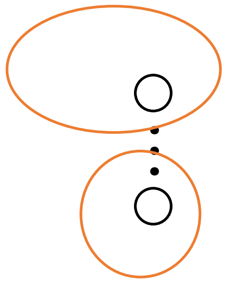


Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster

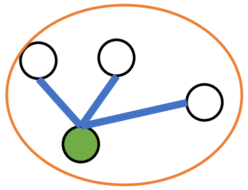


  - Else make v a Pivot, and v starts its own cluster

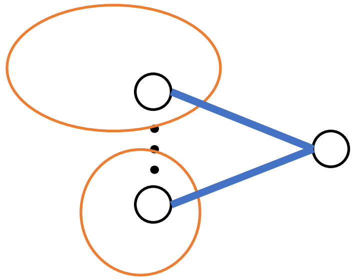


Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster

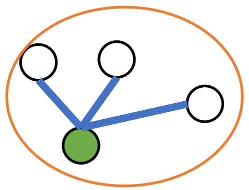


  - Else make v a Pivot, and v starts its own cluster

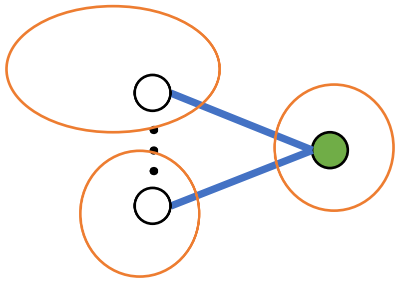


Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster

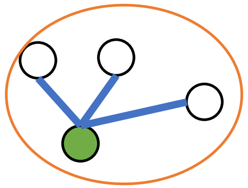


  - Else make v a Pivot, and v starts its own cluster

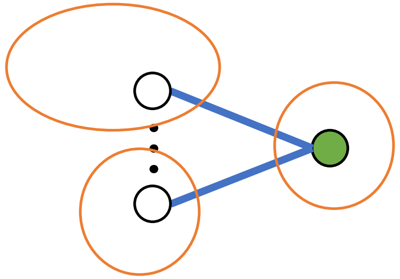


Algorithm

- **Pivot Algorithm**
  - Maintain collection of vertices called **Pivots**
  - Consider vertices in some order
  - If v has an edge to a previous Pivot, then v joins the first such Pivot's cluster



  - Else make v a Pivot, and v starts its own cluster

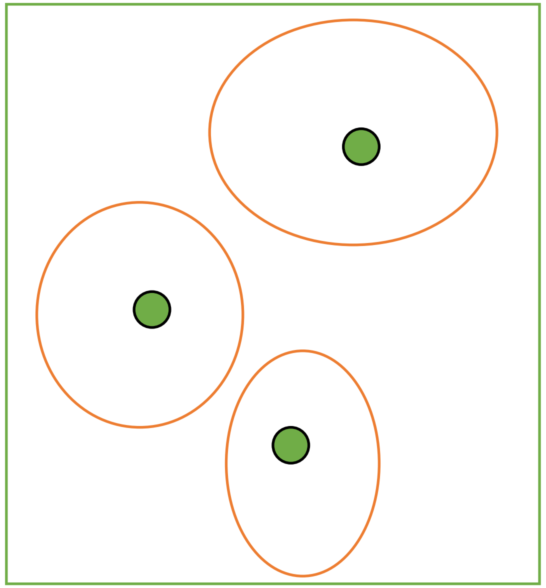


Our Algorithm: Run Pivot in random order in offline phase; then continue in arrival order in online phase

Analysis Overview

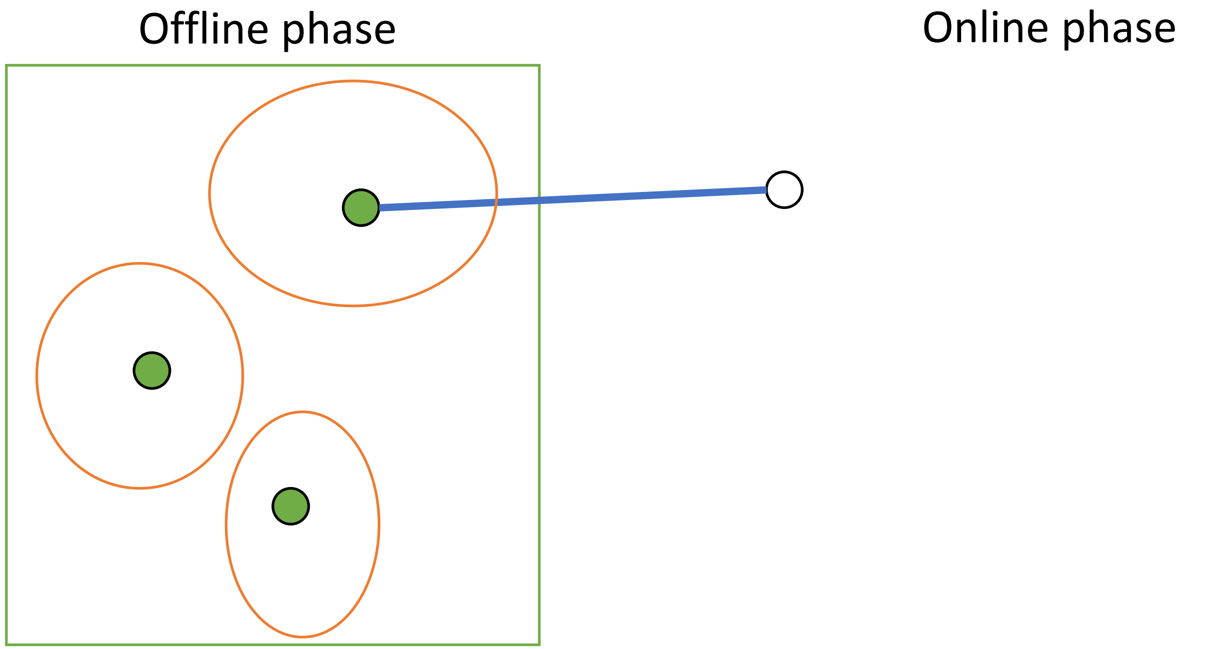
- Assume no corruption
- Use offline phase to pre-cluster online arrivals

Offline phase



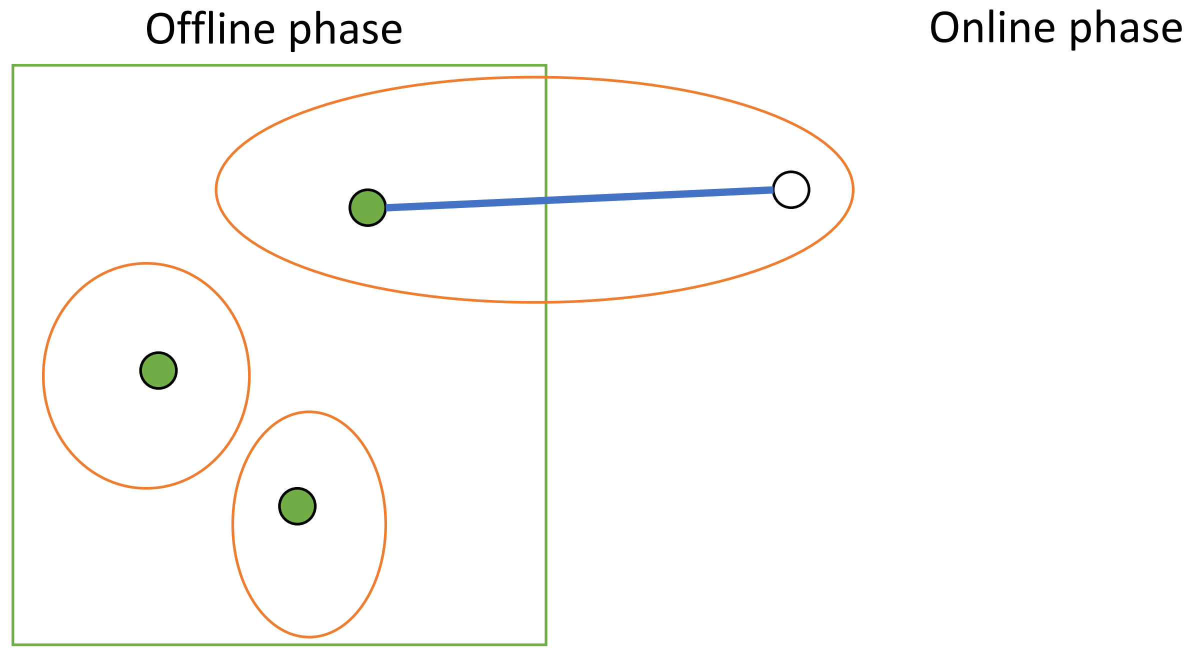
Analysis Overview

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



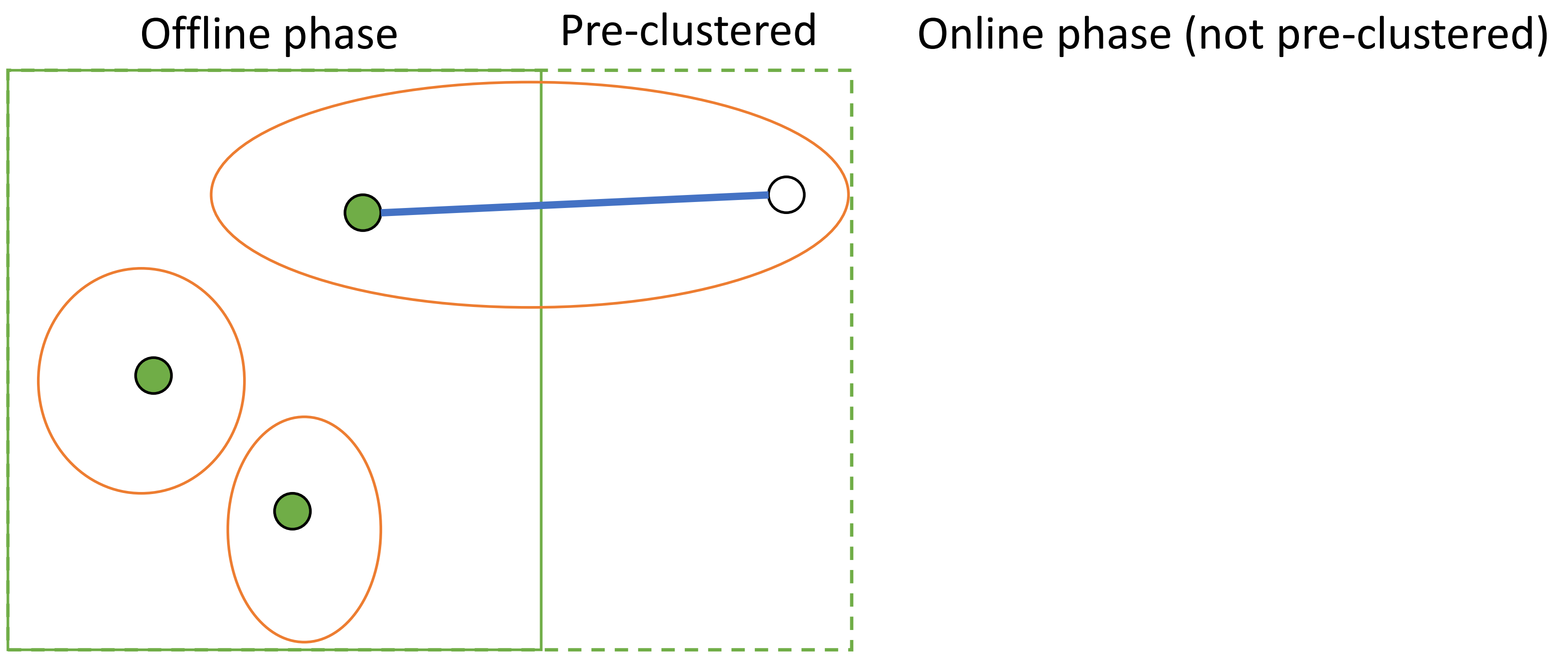
Analysis Overview

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



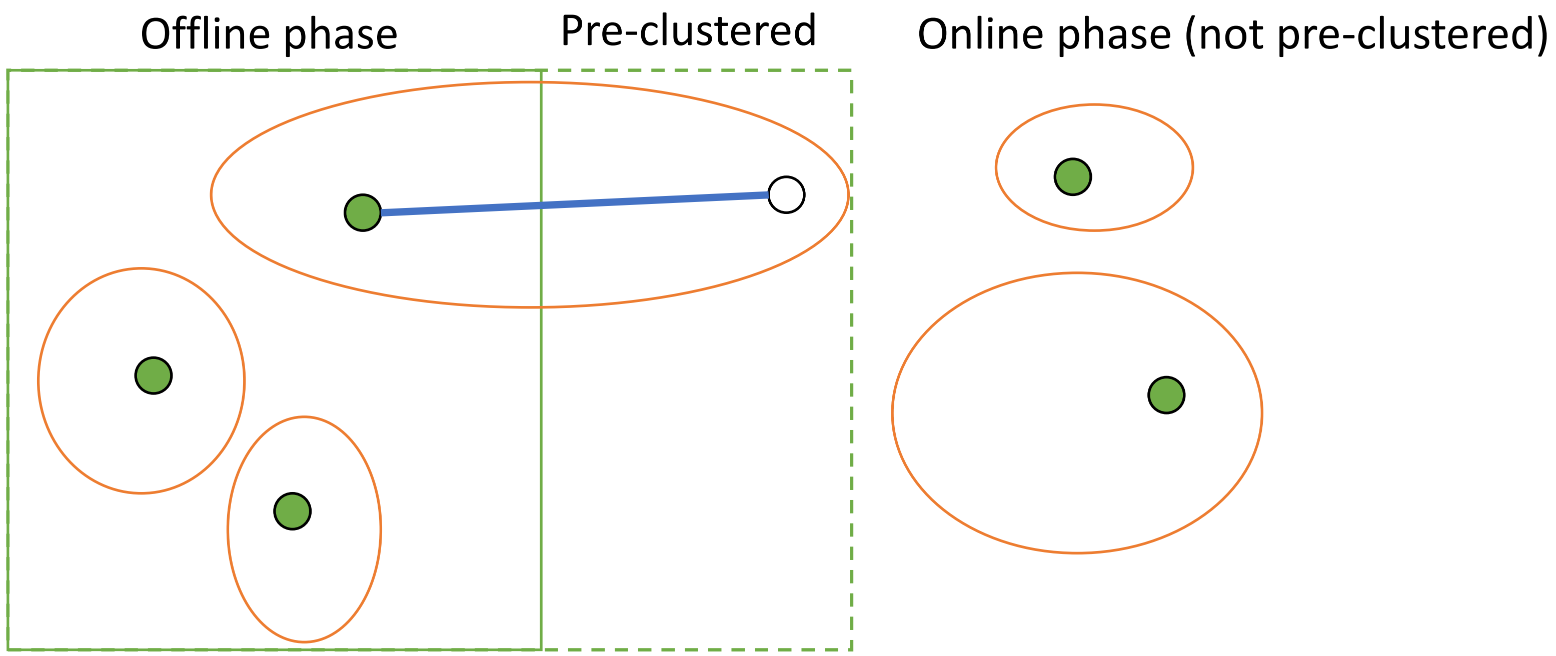
Analysis Overview

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



Analysis Overview

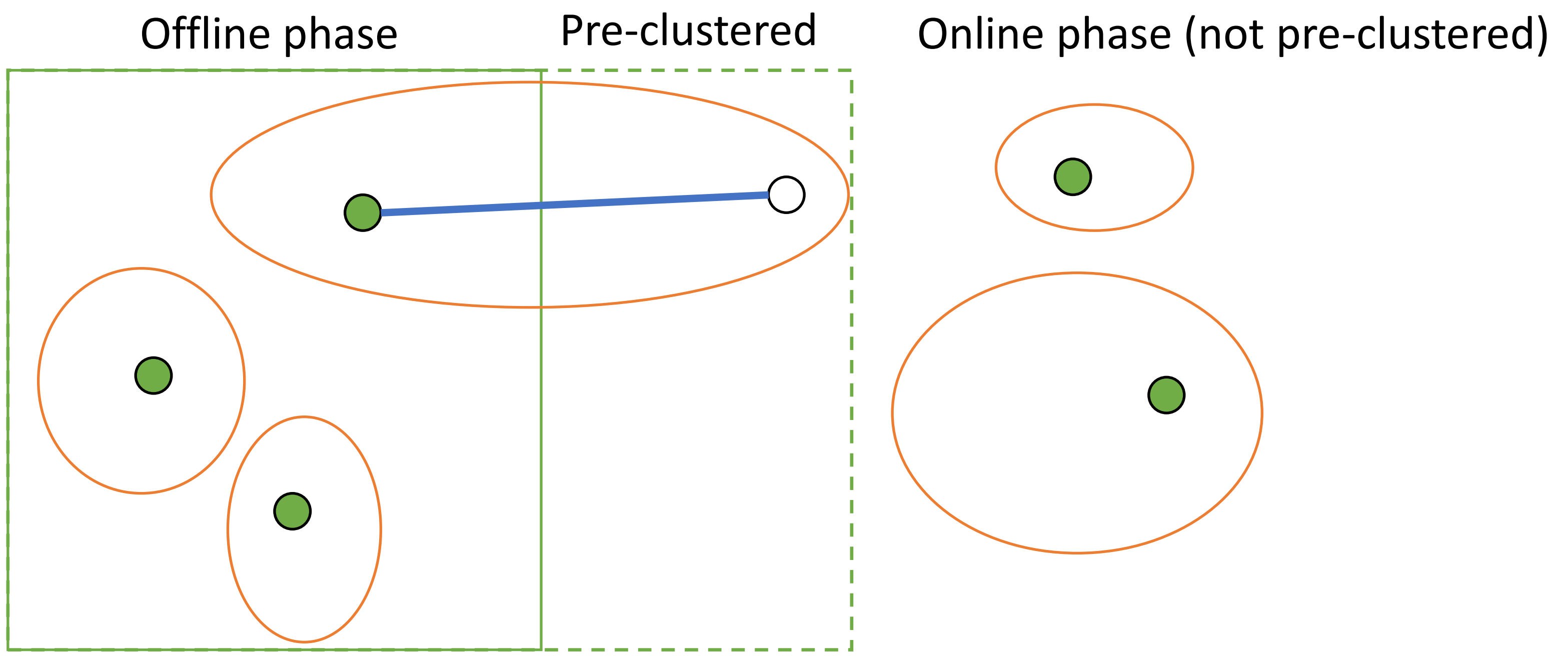
- Assume no corruption
- Use offline phase to pre-cluster online arrivals



Analysis Overview

Pivot is $O(1)$-competitive in random order

- Assume no corruption
- Use offline phase to pre-cluster online arrivals

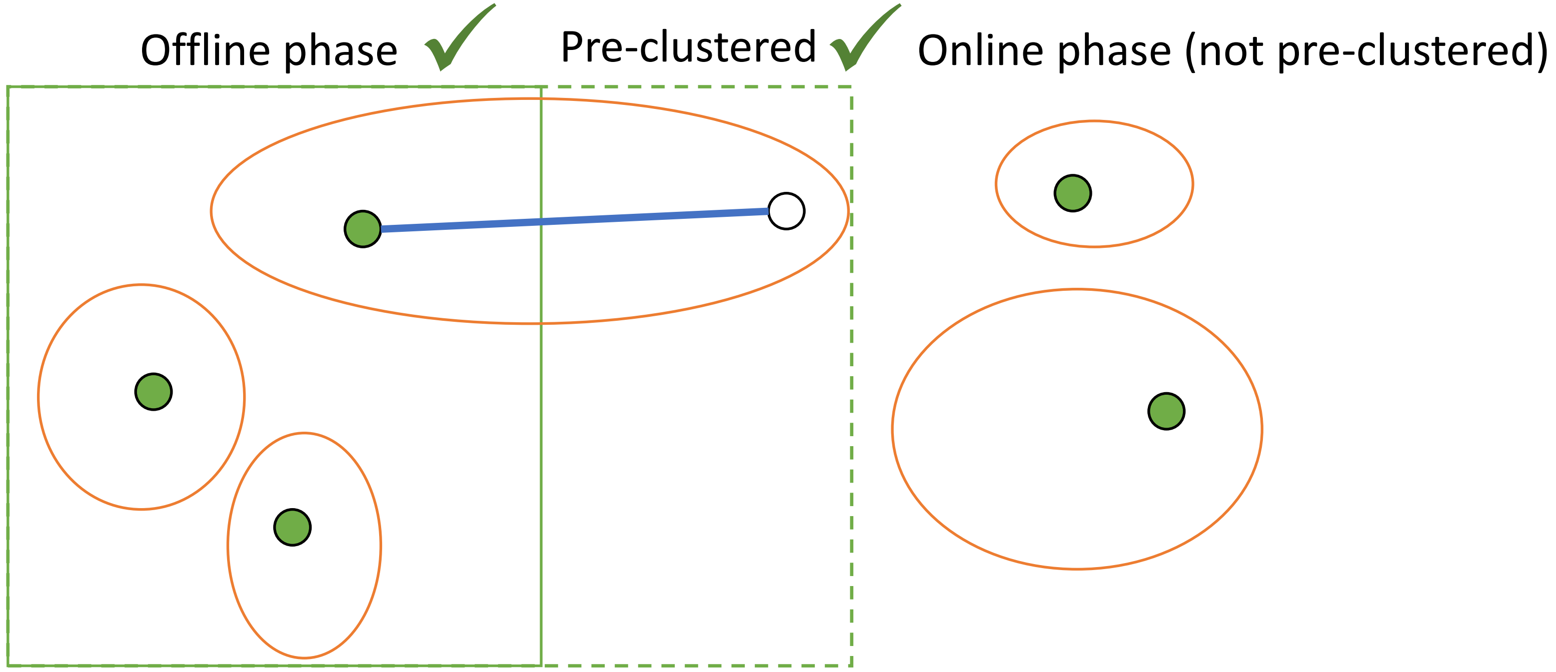


N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

Analysis Overview

Pivot is $O(1)$-competitive in random order

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



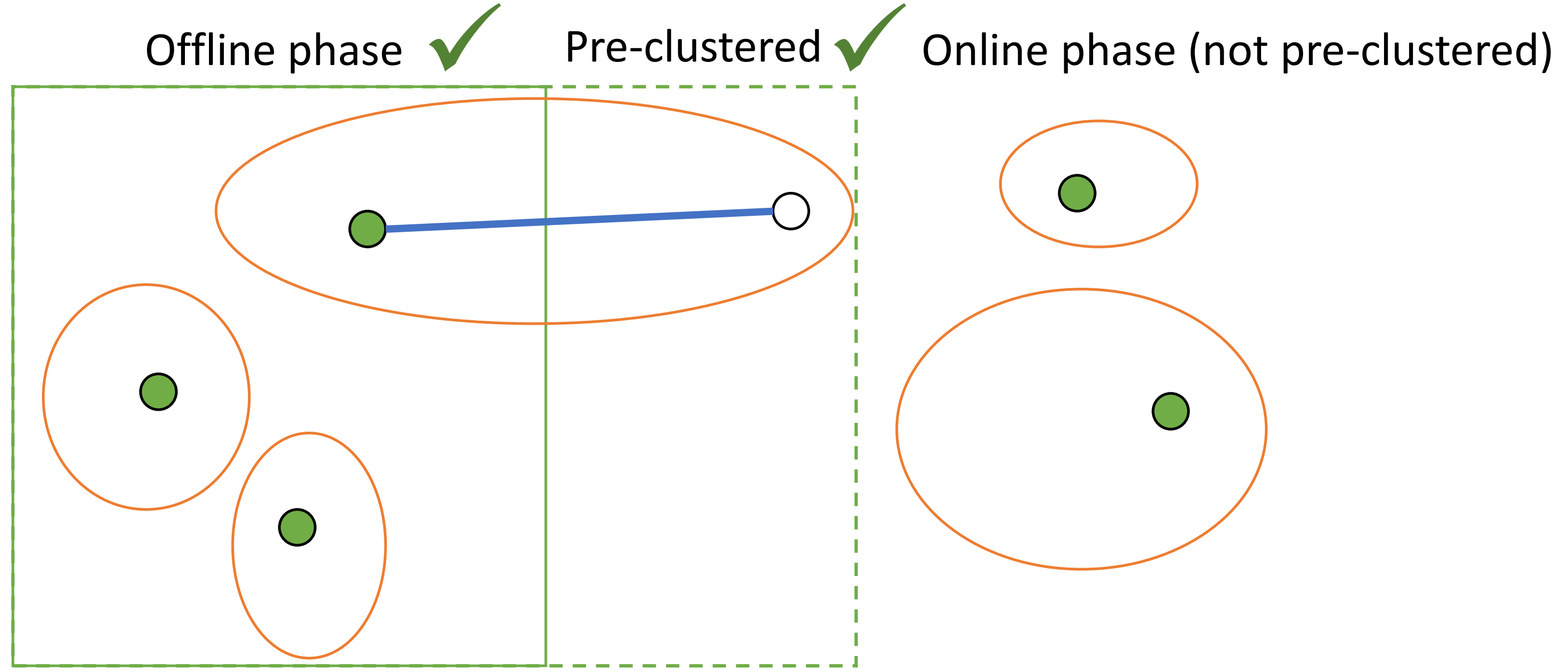
N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

Analysis Overview

Pivot is $O(1)$-competitive in random order

Not pre-clustered graph is sparse

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



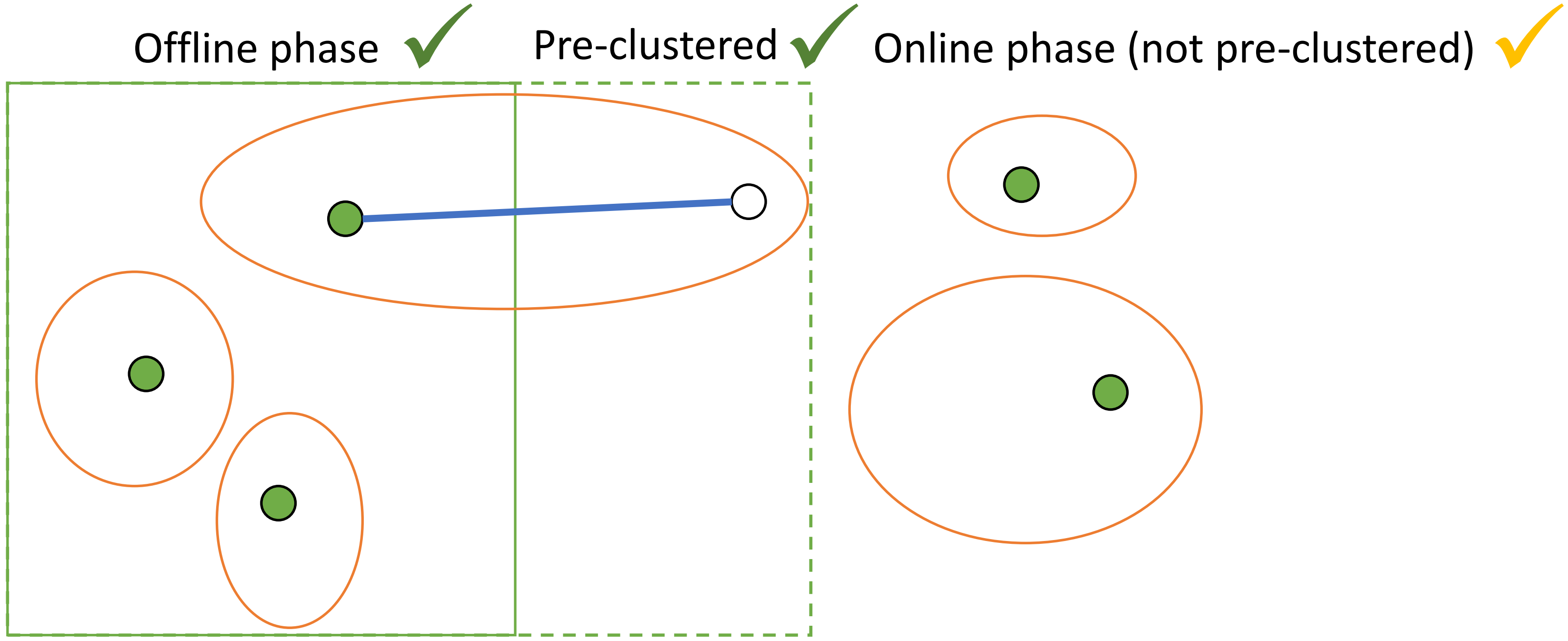
N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

Analysis Overview

Pivot is $O(1)$-competitive in random order

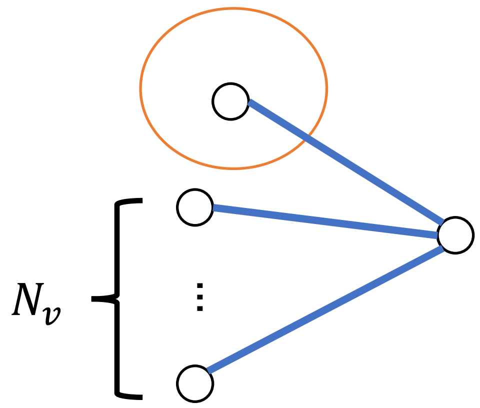
Not pre-clustered graph is sparse

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

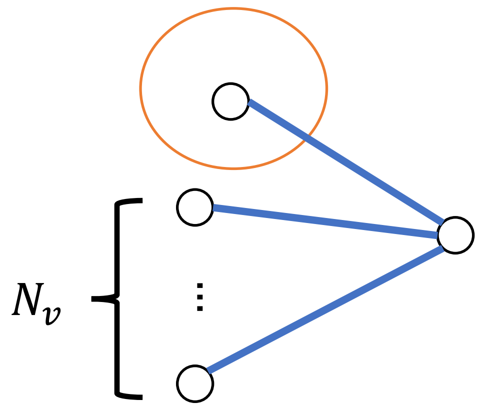
Lemma: For any vertex v, define the random variable N_v such that $N_v = \#(not\ pre - clustered\ neighbors\ of\ v)$ if v is not pre-clustered, and $N_v = 0$ otherwise. Then $\mathbb{E}\, N_v = O(\frac{1}{\epsilon})$.



Our Algorithm: Run Pivot in random order in offline phase; then continue in arrival order in online phase

Lemma: For any vertex v, define the random variable N_v such that $N_v = \#(not\ pre-clustered\ neighbors\ of\ v)$ if v is not pre-clustered, and $N_v = 0$ otherwise. Then $\mathbb{E}\, N_v = O(\frac{1}{\epsilon})$.

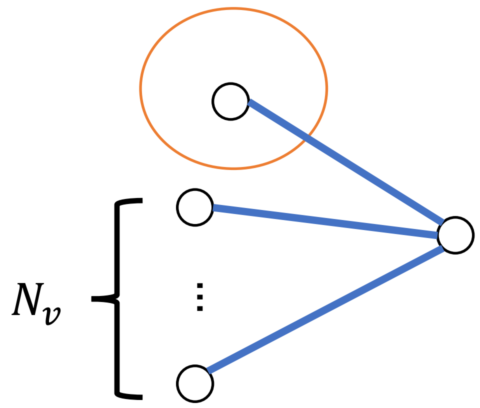
- Assume no corruption
- $N_v \geq k \Rightarrow$ For each arrival in offline phase, v still has k not pre-clustered neighbors, and none of them arrive next (A_i)



Our Algorithm: Run Pivot in random order in offline phase; then continue in arrival order in online phase

Lemma: For any vertex v, define the random variable N_v such that $N_v = \#(not\ pre-clustered\ neighbors\ of\ v)$ if v is not pre-clustered, and $N_v = 0$ otherwise. Then $\mathbb{E}\, N_v = O(\frac{1}{\epsilon})$.

- Assume no corruption
- $N_v \geq k \Rightarrow$ For each arrival in offline phase, v still has k not pre-clustered neighbors, and none of them arrive next (A_i)
- $\mathbb{P}(A_i \mid A_{i-1}, \dots, A_1) \leq \left(1 - \frac{k}{n}\right) \Rightarrow \mathbb{P}(\mathrm{N_v} \geq k) \leq \left(1 - \frac{k}{n}\right)^{\epsilon n} \leq e^{-\epsilon k}$
- Integrating over tail gives $\mathbb{E}\, N_v = O(\frac{1}{\epsilon})$.



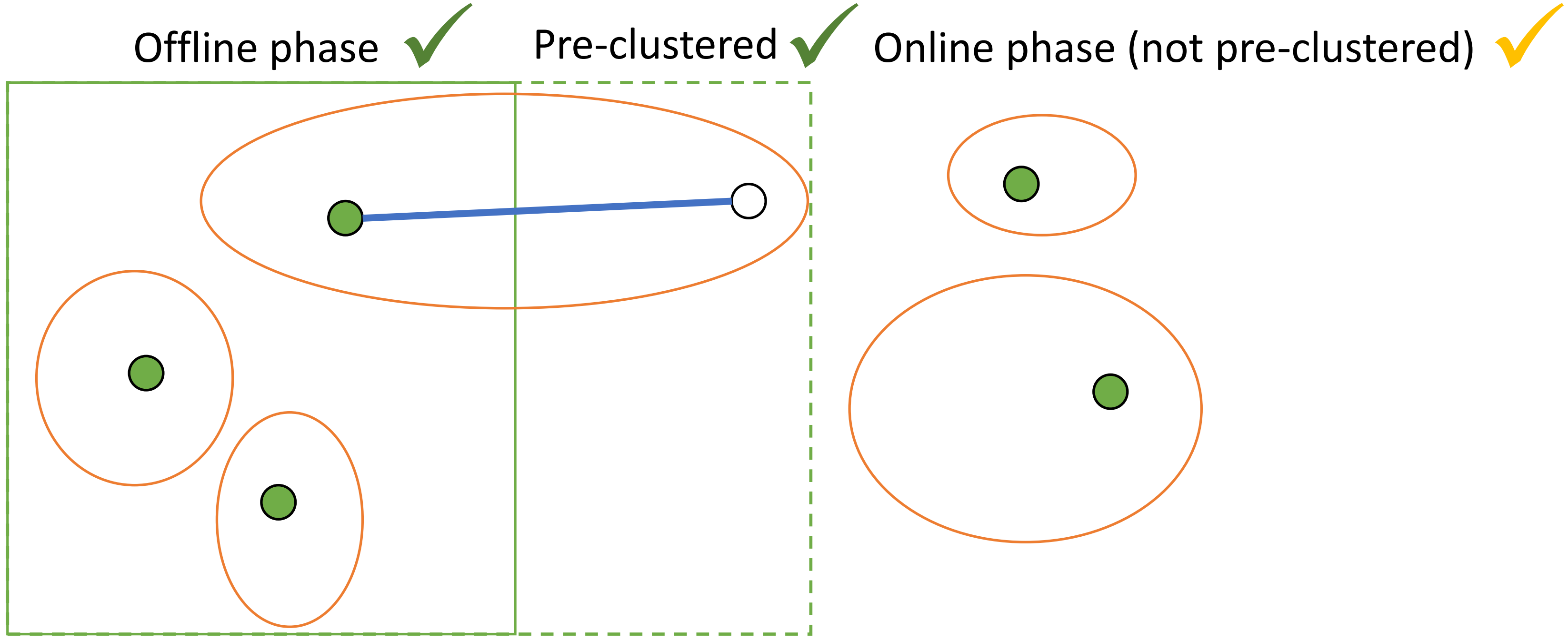
Our Algorithm: Run Pivot in random order in offline phase; then continue in arrival order in online phase

Analysis Overview

Pivot is $O(1)$-competitive in random order

Not pre-clustered graph is sparse

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



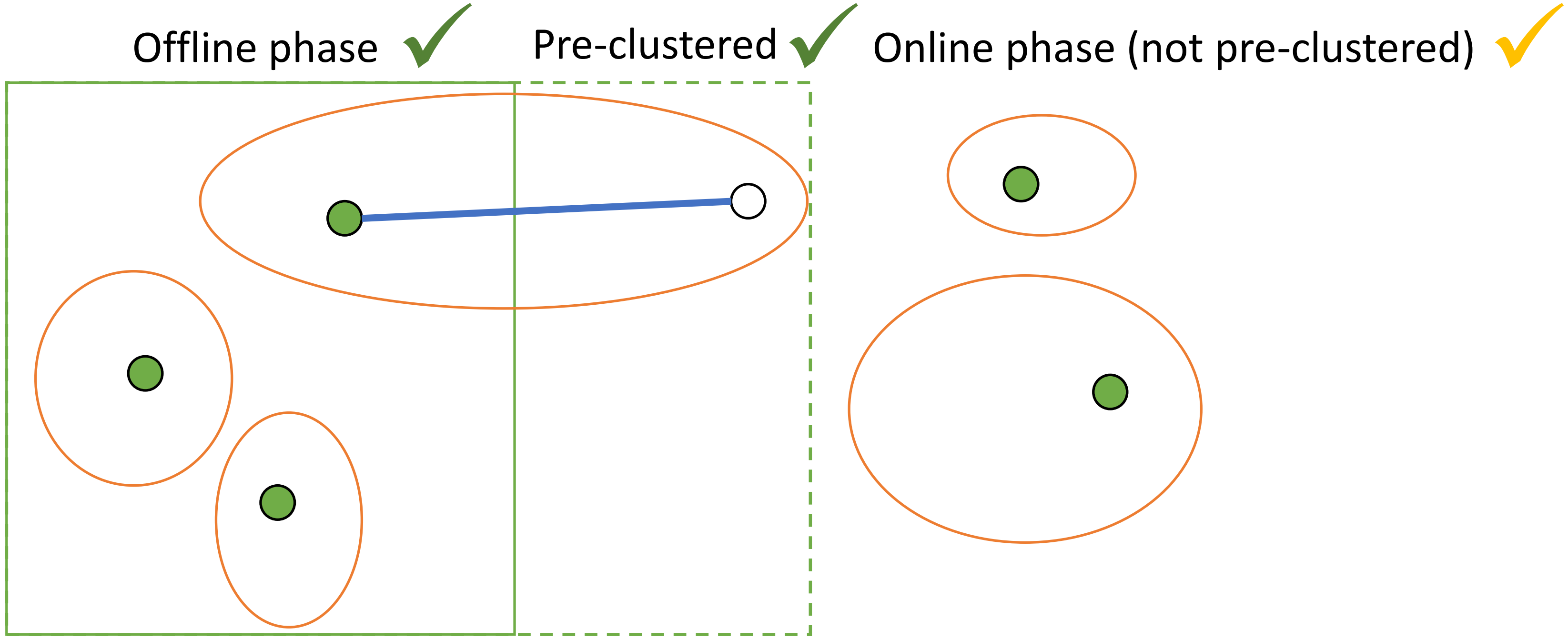
N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

Analysis Overview

Pivot is $O(1)$-competitive in random order

Not pre-clustered graph has expected degrees all $O(\frac{1}{\epsilon})$

- Assume no corruption
- Use offline phase to pre-cluster online arrivals



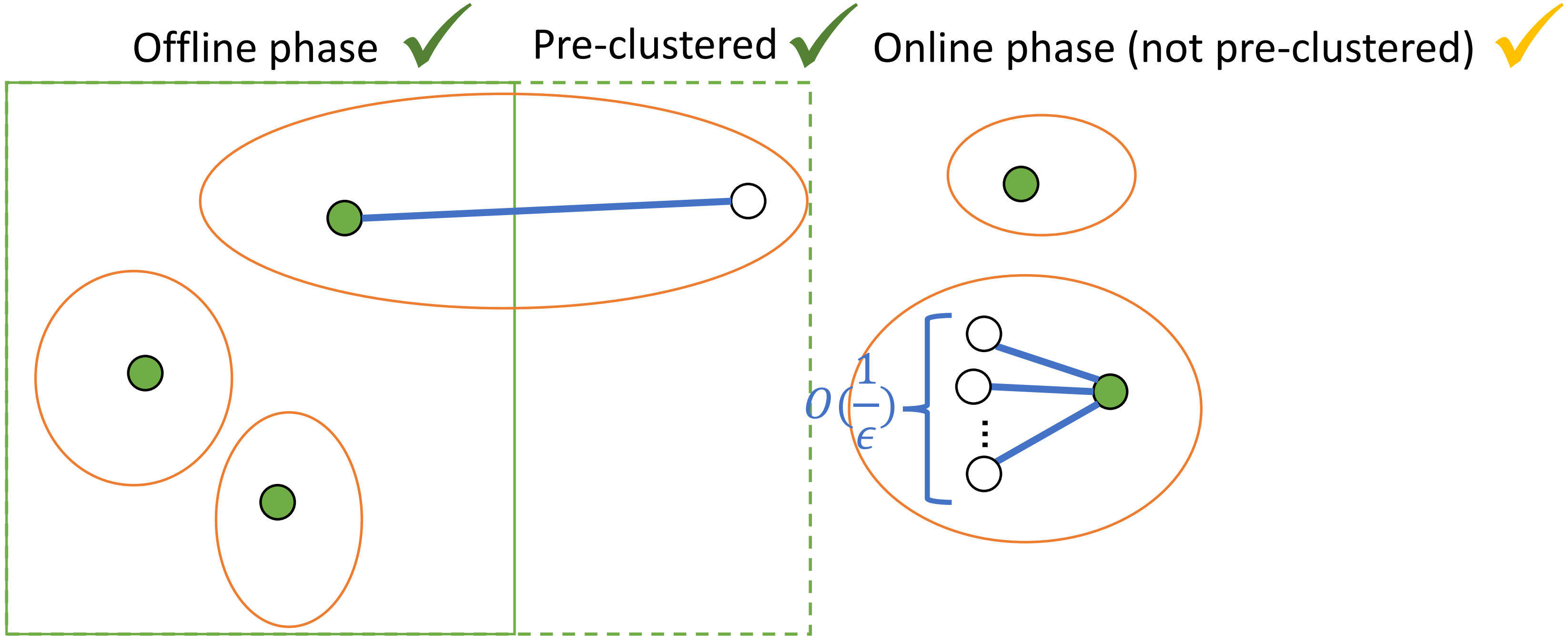
N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

Analysis Overview

Pivot is $O(1)$-competitive in random order

Not pre-clustered graph has expected degrees all $O(\frac{1}{\epsilon})$

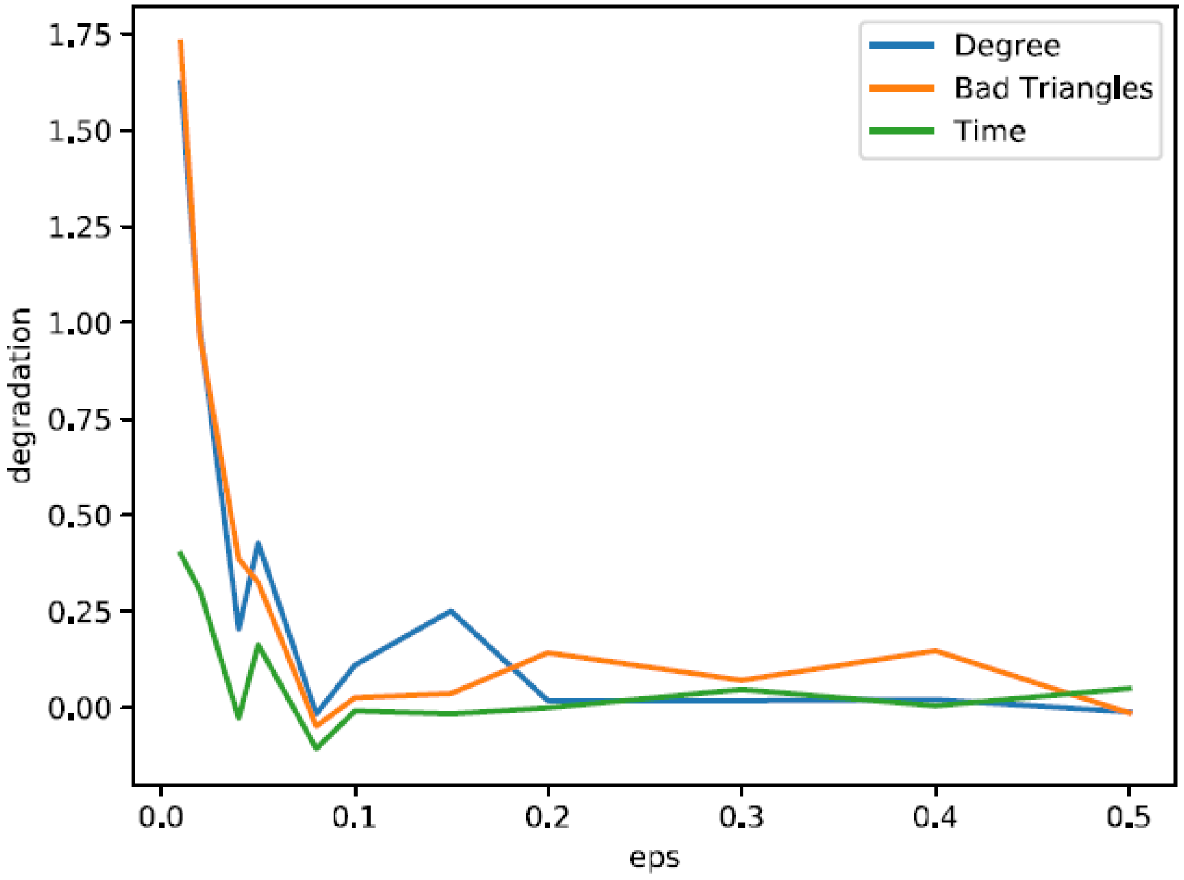
- Assume no corruption
- Use offline phase to pre-cluster online arrivals



N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. JACM 2008.

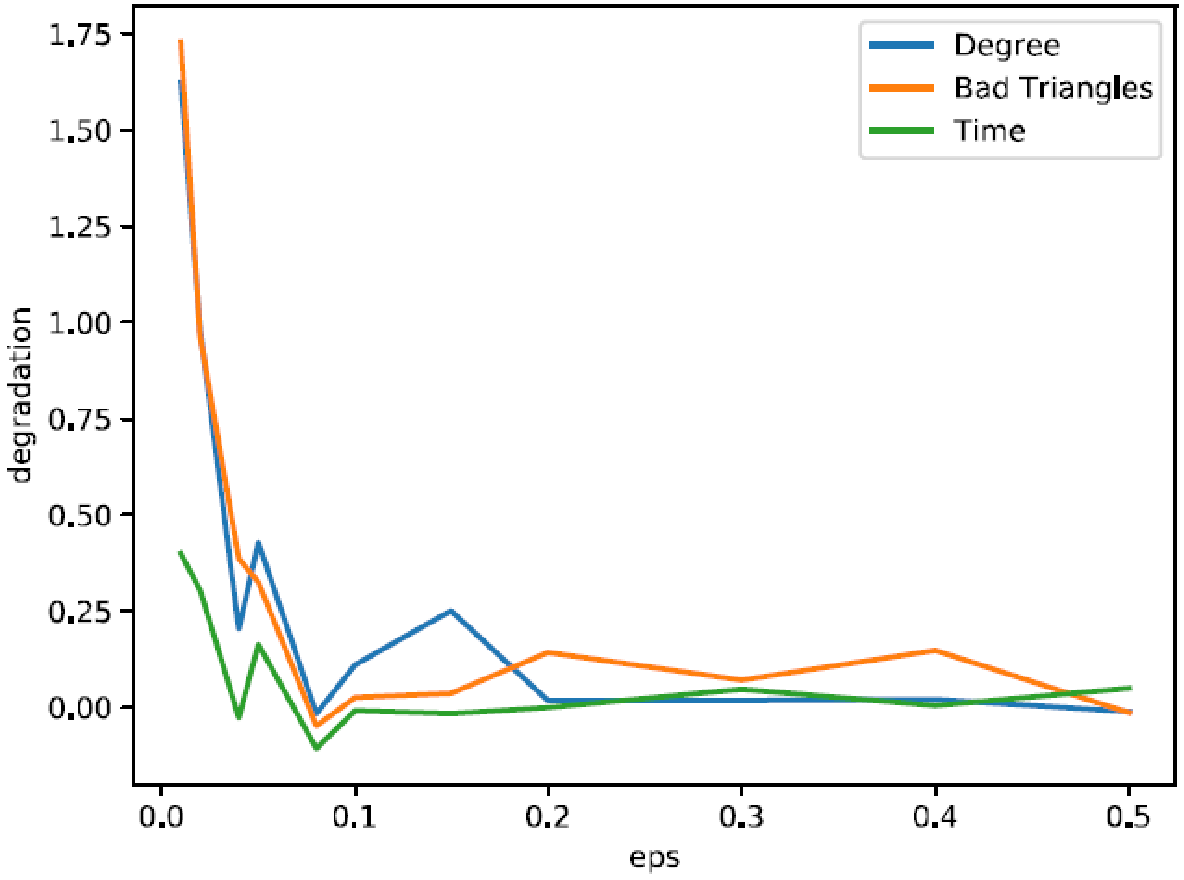
Experimental Results

- Algorithm is competitive with offline baseline for moderate ϵ

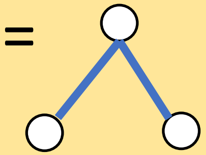


Experimental Results

- Algorithm is competitive with offline baseline for moderate ϵ



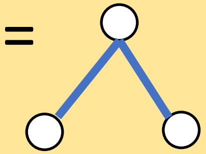
$$Degradation = \frac{Algorithm\ - Baseline}{Baseline}$$

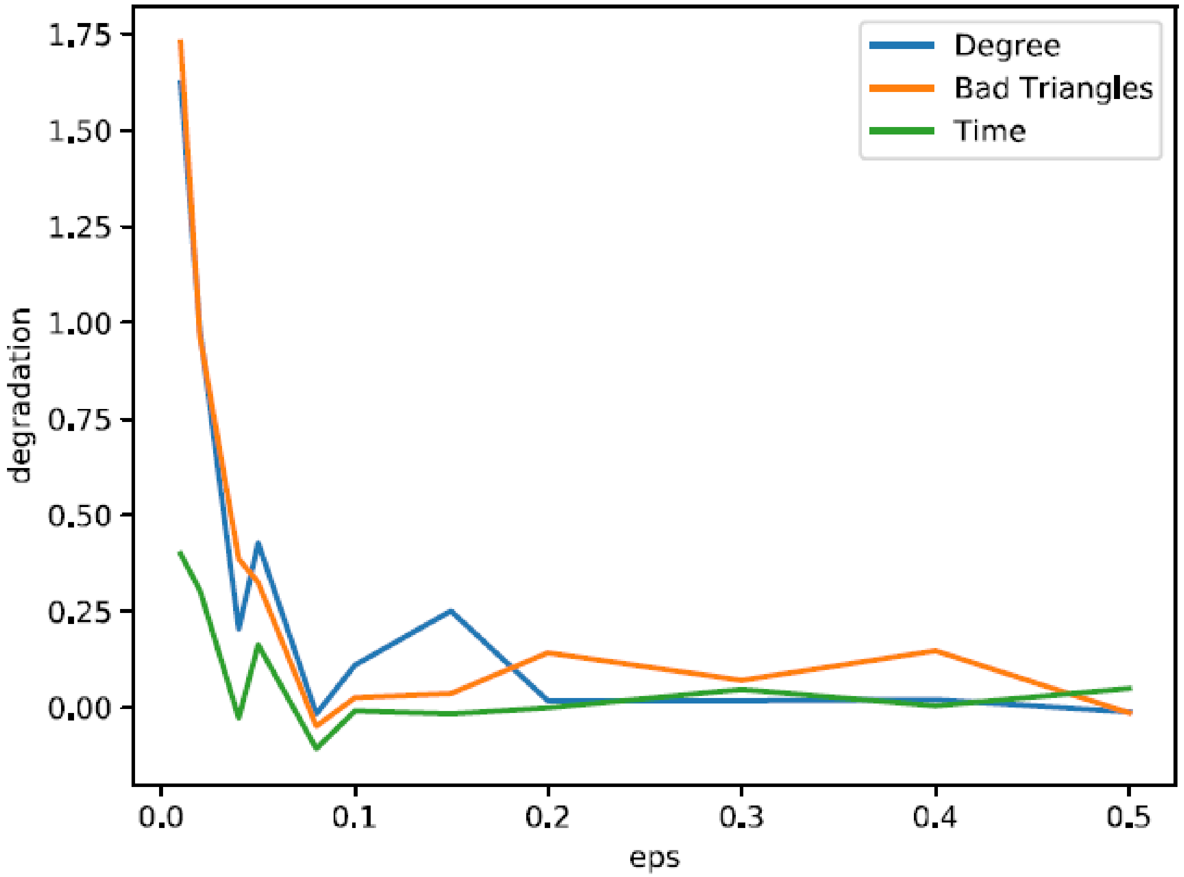
Bad triangle = 

Experimental Results

- Algorithm is competitive with offline baseline for moderate ϵ
- … and robust to adversarial corruptions
- … and random sample can be practically obtained from past data

$$Degradation = \frac{Algorithm\ - Baseline}{Baseline}$$

Bad triangle = 



Summary

- Introduced semi-online model with adversarial corruptions $\sim$ **add data-driven decision making to online algorithms**
- Designed novel semi-online algorithm for correlation clustering with tight competitive ratio $\sim$ **best possible way to use historical data**
- $\Omega(\#vertices)$ lower bound online $\Rightarrow$ $O(1)$-competitive semi-online
- Theory predictive of practice