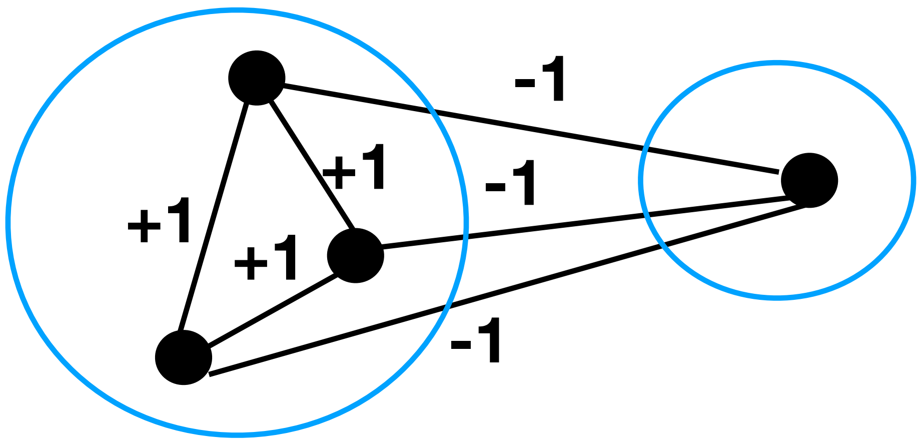
Online and Consistent Correlation Clustering

Andreas Maggiori
Vincent Cohen-Addad, Silvio Lattanzi, Nikos Parotsidis

EPFL Google

Correlation Clustering - Problem Statement

Input: a complete signed graph G = (V, E, s) where s(e) = ‘+’ or ‘-‘ for every edge e



0 disagreements 😀

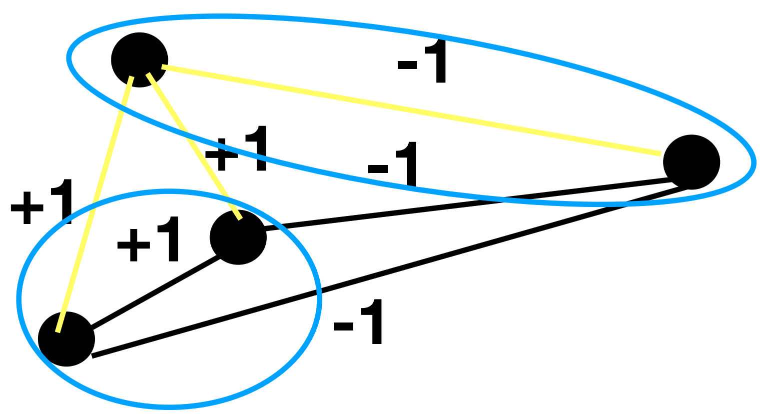
Goal: output a clustering which minimises *disagreements*

‘+’ edge between two nodes assigned to different clusters

‘-‘ edge between two nodes assigned to the same cluster

Correlation Clustering - Problem Statement

Input: a complete signed graph G = (V, E, s) where s(e) = ‘+’ or ‘-‘ for every edge e



3 disagreements 🤨

Goal: output a clustering which minimises *disagreements*

‘+’ edge between two nodes assigned to different clusters

‘-‘ edge between two nodes assigned to the same cluster

Correlation clustering is well studied.

Offline setting

- Introduced by Bansal et al. 2004. The problem is NP-HARD.
- Pivot algorithm is a 3-approximation on expectation. (Ailon et al. 2008)
- LP based 2.06-approximation algorithm (Chawla et al 2015)

Correlation clustering is well studied.

Offline setting

- Introduced by Bansal et al. 2004. The problem is NP-HARD.
- Pivot algorithm is a 3-approximation on expectation. (Ailon et al. 2008)
- LP based 2.06-approximation algorithm (Chawla et al 2015)

Other settings

- Distributed: Chierichietti et al 2014, Ahi et al. 2015, Pan et al. 2015, Cohen-Addad et al. 2021
- Online: Mathieu et al. 2010

Online setting

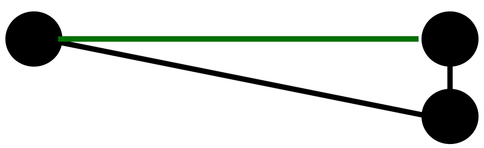
- At each time t: a node arrives, revealing all its incident edges to previously arrived nodes.
- Clustering decisions are irrevocable:
  - create a new singleton cluster with the newly arrived node; or
  - Add that node to a preexisting cluster.
- Mathieu et al. proved that any online algorithm is $\Omega(n)$- competitive. Why: difficult to distinguish if an edge is a bridge between two cliques or it is part of a clique. 😩

Online setting

- At each time t: a node arrives, revealing all its incident edges to previously arrived nodes.
- Clustering decisions are irrevocable:
  - create a new singleton cluster with the newly arrived node; or
  - Add that node to a preexisting cluster.
- Mathieu et al. proved that any online algorithm is $\Omega(n)$- competitive. Why: difficult to distinguish if an edge is a bridge between two cliques or it is part of a clique. 😩

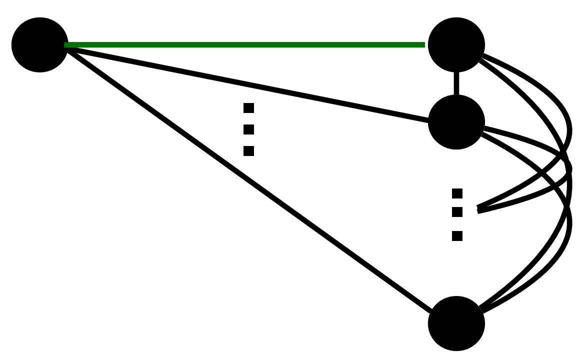
Online setting

- At each time t: a node arrives, revealing all its incident edges to previously arrived nodes.
- Clustering decisions are irrevocable:
  - create a new singleton cluster with the newly arrived node; or
  - Add that node to a preexisting cluster.
- Mathieu et al. proved that any online algorithm is $\Omega(n)$- competitive. Why: difficult to distinguish if an edge is a bridge between two cliques or it is part of a clique.



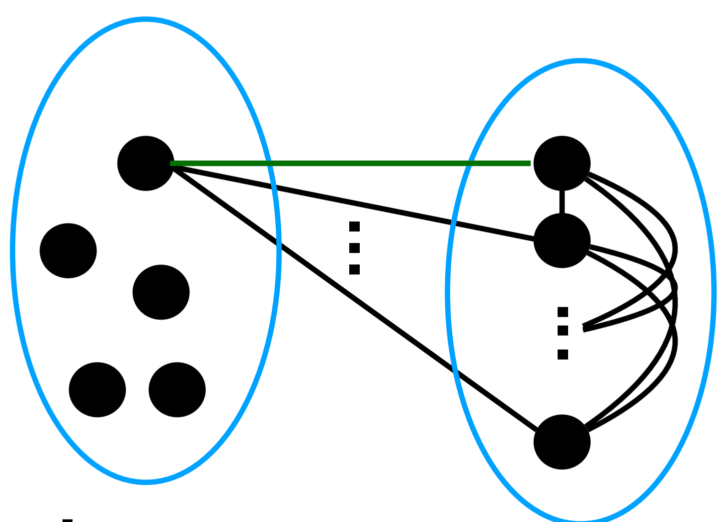
Online setting

- At each time t: a node arrives, revealing all its incident edges to previously arrived nodes.
- Clustering decisions are irrevocable:
  - create a new singleton cluster with the newly arrived node; or
  - Add that node to a preexisting cluster.
- Mathieu et al. proved that any online algorithm is $\Omega(n)$- competitive. Why: difficult to distinguish if an edge is a bridge between two cliques or it is part of a clique.



Online setting

- At each time t: a node arrives, revealing all its incident edges to previously arrived nodes.
- Clustering decisions are irrevocable:
  - create a new singleton cluster with the newly arrived node; or
  - Add that node to a preexisting cluster.
- Mathieu et al. proved that any online algorithm is $\Omega(n)$- competitive. Why: <u>difficult to distinguish if an edge is a bridge between two cliques or it is part of a clique</u>. 😩



Large clique

Online setting with recourse

 # times a node changes cluster

- At each time t: a node arrives, revealing all its incident edges to previously arrived nodes.
- Clustering decisions are **NOT** irrevocable.
- Goal: having at all times a constant factor approximation while minimising the worst case recourse of a node.

Contributions:

- A constant factor approximation algorithm which achieves worst case log(n) recourse per node.
- A matching lower bound.

Online setting with recourse

Agreement Algorithm of Cohen-Addad et al 2021 as a subroutine

- Creates dense clusters with few outgoing edges
- Constant factor approximation

Online setting with recourse

Agreement Algorithm of Cohen-Addad et al 2021 as a subroutine

Creates dense clusters
with few outgoing edges

Constant factor approximation

Online Agreement Algorithm

1. Rerun the Agreement Algorithm

Online setting with recourse

Agreement Algorithm of Cohen-Addad et al 2021 as a subroutine

Creates dense clusters
with few outgoing edges

Constant factor approximation

Online Agreement Algorithm

1. Rerun the Agreement Algorithm

Worst case linear recourse

😩

Online setting with recourse

Agreement Algorithm of Cohen-Addad et al 2021 as a subroutine

- Creates dense clusters with few outgoing edges
- Constant factor approximation

Online Agreement Algorithm

1. Rerun the Agreement Algorithm
2. Stabilize the resulting clustering

- Follow (blindly) the advice of the *Agreement* Algorithm to get constant competitive ratio for free
- Change the clustering as little as possible to obtain low recourse

Online setting with recourse

Agreement Algorithm of Cohen-Addad et al 2021 as a subroutine

Creates dense clusters with few outgoing edges

Constant factor approximation

Online Agreement Algorithm

1. Rerun the Agreement Algorithm
2. Stabilize the resulting clustering

Follow (blindly) the advice of the *Agreement* Algorithm to get constant competitive ratio for free

Change the clustering as little as possible to obtain low recourse

Balance these two opposite objective by:

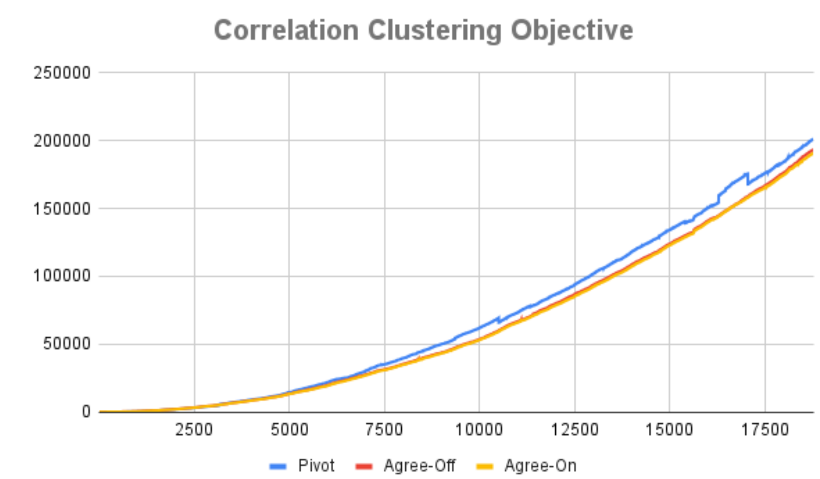
- Keep track of evolving clusters
- Use dense structure of the clustering computed by the *Agreement* Algorithm

Online setting with recourse

Agreement Algorithm of Cohen-Addad et al 2021 as a subroutine

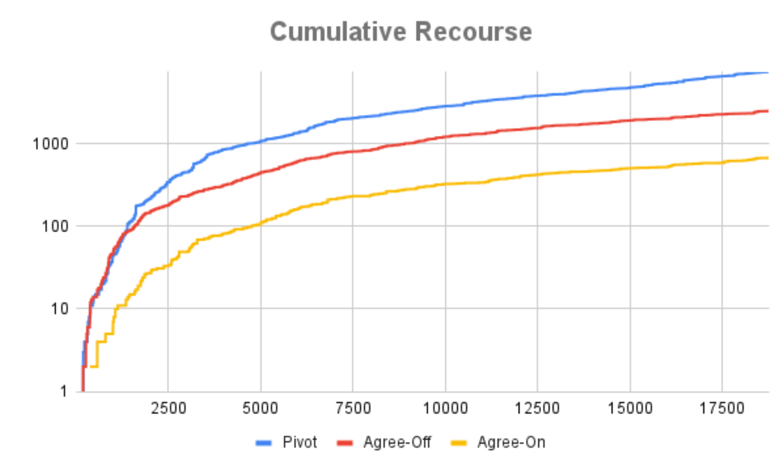
Creates dense clusters with few outgoing edges

Constant factor approximation



Online Agreement Algorithm

1. Rerun the Agreement Algorithm
2. Stabilize the resulting clustering



Change the clustering as little as possible to obtain low recourse

Follow (blindly) the advice of the *Agreement* Algorithm to get constant competitive ratio for free

Balance these two opposite objective by:

- Keep track of evolving clusters
- Use dense structure of the clustering computed by the *Agreement* Algorithm

Thank you for your attention!