Correlation Clustering in Map-Reduce

Flavio Chierichetti
Sapienza

Nilesh Dalvi
Trooly

Ravi Kumar
Google
Correlation Clustering

[Bansal, Blum, Chawla]

- Clustering tool that has been used in many data mining tasks (e.g., [Arasu et al, ICDE ’09], [Bonchi et al, KDD ’12], [Cesa-Bianchi et al, JMLR ’12], [Elmagarmid et al, TKDE ’12], [Günnemann et al, TKDD ’12])
Correlation Clustering

[Bansal, Blum, Chawla]

• Clustering tool that has been used in many data mining tasks (e.g., [Arasu et al, ICDE ’09], [Bonchi et al, KDD ’12], [Cesa-Bianchi et al, JMLR ’12], [Elmagarmid et al, TKDE ’12], [Günnemann et al, TKDD ’12])

Web pages
Users
Ads
…
Correlation Clustering
[Bansal, Blum, Chawla]

• Clustering tool that has been used in many data mining tasks (e.g., [Arasu et al, ICDE ’09], [Bonchi et al, KDD ’12], [Cesa-Bianchi et al, JMLR ’12], [Elmagarmid et al, TKDE ’12], [Günnemann et al, TKDD ’12])

Web pages
Users
Ads
…
Correlation Clustering

[Bansal, Blum, Chawla]

- Clustering tool that has been used in many data mining tasks (e.g., [Arasu et al, ICDE ’09], [Bonchi et al, KDD ’12], [Cesa-Bianchi et al, JMLR ’12], [Elnagarmid et al, TKDE ’12], [Günnemann et al, TKDD ’12])

- The goal is to find a clustering that keeps together similar items, and that places dissimilar items in distinct clusters.
Correlation Clustering

[Bansal, Blum, Chawla]

• Clustering tool that has been used in many data mining tasks (e.g., [Arasu et al, ICDE '09], [Bonchi et al, KDD '12], [Cesa-Bianchi et al, JMLR '12], [Elmagarmid et al, TKDE '12], [Günnemann et al, TKDD '12])

• The goal is to find a clustering that keeps together similar items, and that places dissimilar items in distinct clusters.
Large-Scale Computation

- Web, social, mobile data is too large to be held on a single computer,
- but it can be distributed across multiple machines
Large-Scale Computation

• Web, social, mobile data is too large to be held on a single computer,

• but it can be distributed across multiple machines

• Data processing has to be fast, parallel and has to have a tiny memory footprint
Large-Scale Computation

- Web, social, mobile data is too large to be held on a single computer,
- but it can be distributed across multiple machines
- Data processing has to be fast, parallel and has to have a tiny memory footprint

- Pregel
- Map-Reduce
- Streaming
Correlation Clustering
[Bansal, Blum, Chawla]
Correlation Clustering

[Bansal, Blum, Chawla]
Correlation Clustering
[Bansal, Blum, Chawla]
Correlation Clustering

[Bansal, Blum, Chawla]
Correlation Clustering

[Bansal, Blum, Chawla]

0 mistakes
Correlation Clustering

[Bansal, Blum, Chawla]

1 mistake
Correlation Clustering

[Bansal, Blum, Chawla]

The goal is to partition the elements in disjoint clusters so to minimize the number of mistakes.
Correlation Clustering

[Bansal, Blum, Chawla]

The goal is to partition the elements in disjoint clusters so to minimize the number of mistakes.

The number of clusters does not have to be selected by hand.
Correlation Clustering

"Perfect clustering"
(0 mistakes)
Correlation Clustering
Correlation Clustering
Correlation Clustering
How to find the best clustering?
How to find the best clustering?

How to minimize the number of mistakes?
The Pivot Algorithm
[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$
   and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm
[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$
   and all the elements “similar” to it
3. Remove the cluster, and repeat

![Diagram of a network with a pivot node $p$.]
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm
[Ailon, Charikar, Newman]

1. Pick an element \( p \) uniformly at random
2. Create a cluster containing \( p \) and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element \( p \) uniformly at random
2. Create a cluster containing \( p \)
   and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$
   and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm
[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element \( p \) uniformly at random
2. Create a cluster containing \( p \) and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element \( p \) uniformly at random
2. Create a cluster containing \( p \) and all the elements “similar” to it
3. Remove the cluster, and repeat

The Pivot Algorithm gives an (expected) 3 approximation to the problem of minimizing the number of mistakes
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element \( p \) uniformly at random
2. Create a cluster containing \( p \) and all the elements “similar” to it
3. Remove the cluster, and repeat

The Pivot Algorithm gives an (expected) \( 3 \) approximation to the problem of minimizing the number of mistakes

Unfortunately the Pivot algorithm is inherently sequential
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$
   and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element \( p \) uniformly at random
2. Create a cluster containing \( p \) and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$
   and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm
[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat
The Pivot Algorithm

[Ailon, Charikar, Newman]

1. Pick an element $p$ uniformly at random
2. Create a cluster containing $p$ and all the elements “similar” to it
3. Remove the cluster, and repeat

$n$ iterations

$n$ can be huge ($\sim 10^3$)
How to speed up the computation?
Our Contribution

• We propose a parallel version of the Pivot Algorithm, that

• requires only $O(\log^2 n)$ iterations, and

• still guarantees a $3 + \epsilon$ approximation.
Our Contribution

• We propose a parallel version of the Pivot Algorithm, that
  • requires only $O(\log^2 n)$ iterations, and
  • still guarantees a $3 + \epsilon$ approximation.
Our Contribution

- We propose a parallel version of the Pivot Algorithm, that
  - requires only $O(\log^2 n)$ iterations, and
  - still guarantees a $3 + \epsilon$ approximation.
- The new algorithm can be easily implemented in Map-Reduce, Pregel, streaming, etc.
Our Contribution

• We ran our algorithm on a number of datasets (CORA, WDC, Twitter)

• The largest had 41M elements, with 2.5B positive edges
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
    • Activate each element independently with probability $\epsilon/\Delta^+$
    • Deactivate all the active elements that are connected through a positive edge to other active elements
    • The remaining active nodes become pivots
    • Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
    • Activate each element independently with probability $\epsilon/\Delta^+$
    • Deactivate all the active elements that are connected through a positive edge to other active elements
    • The remaining active nodes become pivots
  • Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  -Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)

$\Delta^+ = 2$
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
    - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\frac{\epsilon}{\Delta^+}$
    - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
- The remaining active nodes become pivots
- Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
  • Activate each element independently with probability $\frac{\epsilon}{\Delta^+}$
  • Deactivate all the active elements that are connected through a positive edge to other active elements
  • The remaining active nodes become pivots
  • Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\frac{\epsilon}{\Delta^+}$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
    - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
  • Activate each element independently with probability $\epsilon/\Delta^+$
  • Deactivate all the active elements that are connected through a positive edge to other active elements
  • The remaining active nodes become pivots
  • Create one cluster for each pivot (breaking ties randomly)

![Diagram of parallel pivot](image)
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
  • Activate each element independently with probability $\epsilon / \Delta^+$
  • Deactivate all the active elements that are connected through a positive edge to other active elements
  • The remaining active nodes become pivots
  • Create one cluster for each pivot (breaking ties randomly)

$\Delta^+ = 1$
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
  • Activate each element independently with probability $\epsilon/\Delta^+$
  • Deactivate all the active elements that are connected through a positive edge to other active elements
  • The remaining active nodes become pivots
  • Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

• While the instance is non-empty
  • Let $\Delta^+$ be its current maximum positive degree
  • Activate each element independently with probability $\frac{\epsilon}{\Delta^+}$
  • Deactivate all the active elements that are connected through a positive edge to other active elements
  • The remaining active nodes become pivots
  • Create one cluster for each pivot
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\frac{\epsilon}{\Delta^+}$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\frac{\epsilon}{\Delta^+}$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)

At the outset, we assign a UAR priority to each element
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\frac{\varepsilon}{\Delta^+}$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)

At the outset, we assign a UAR priority to each element

Ties are broken by priority
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot \textit{(breaking ties randomly)}

At the outset, we assign a UAR priority to each element

Ties are broken by priority
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)

Easy to implement it in distributed frameworks (Pregel, Hadoop, …)
Parallel Pivot

- While the instance is non-empty
  - Let $\Delta^+$ be its current maximum positive degree
  - Activate each element independently with probability $\epsilon/\Delta^+$
  - Deactivate all the active elements that are connected through a positive edge to other active elements
  - The remaining active nodes become pivots
  - Create one cluster for each pivot (breaking ties randomly)

Easy to implement it in distributed frameworks (Pregel, Hadoop, …)

The algorithm requires $O(\log n)$ iterations if $\Delta^+$ is constant
Twitter Dataset

[Kwak et al]
Twitter Dataset

[Kwak et al]

41M elements/users
2.5B positive (follow) edges
2.9M maximum degree
Twitter Dataset

[Kwak et al]

Remaining Elements vs. Iteration
Twitter Dataset

[Kwak et al]
Thanks!