Clique percolation method: memory efficient almost exact communities

Alexis Baudin, Maximilien Danisch, Sergey Kirgizov,
Clémence Magnien, Marwan Ghanem

2-4 February 2022

The 17th International Conference on Advanced Data Mining and Applications (ADMA), Sydney, Australia (2021)

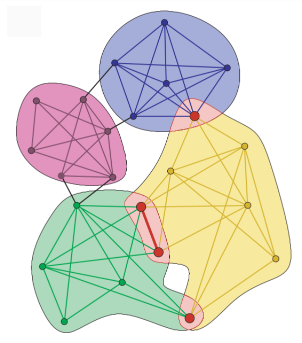
Contents

Parts of the presentation :

Communities in a graph

Community :

- Densely connected inside
- Sparsly connected outside



Palla et al. 2005

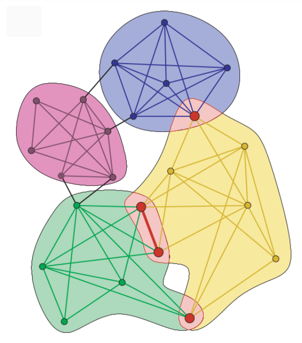
Communities in a graph

Community :

- Densely connected inside
- Sparsly connected outside

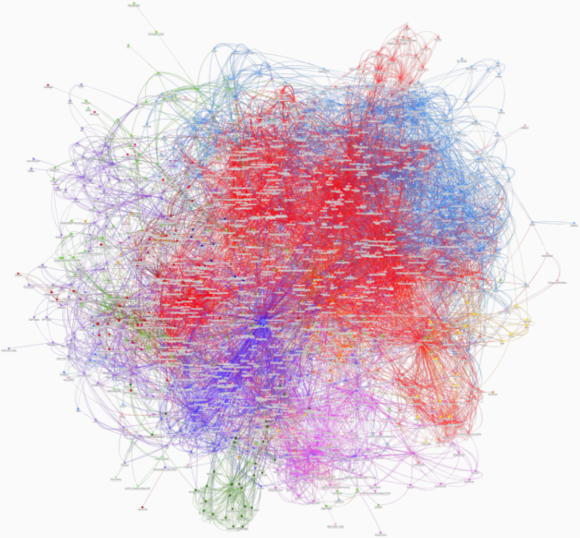
Interest :

- Massive graph : zoom in and out
- Biological interactions
- Content recommendation
- ...



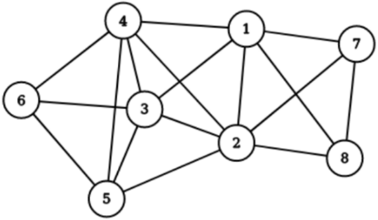
Palla et al. 2005

Scaling up to massive graph

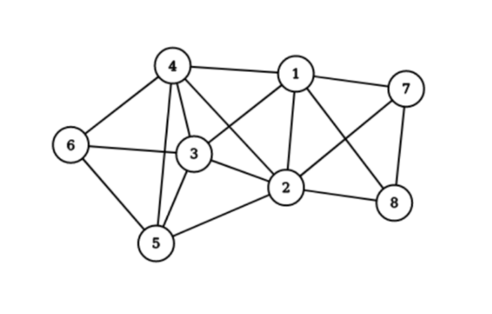


CPM - Definition

CPM - Definition



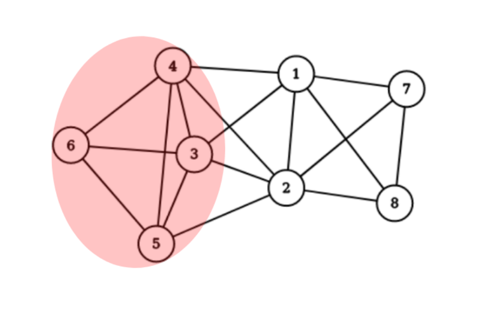
CPM - Definition



k-clique

Set of *k* nodes all connected to each other.

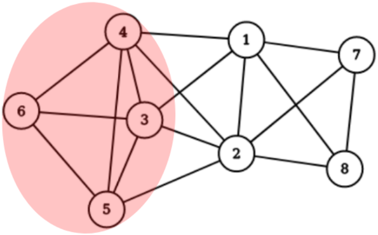
CPM - Definition



k-clique

Set of k nodes all connected to each other.

CPM - Definition



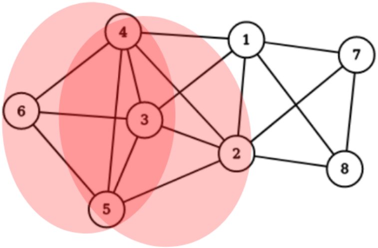
k-clique

Set of k nodes all connected to each other.

Adjacent k-cliques

Two k-cliques are adjacent if they share $k-1$ nodes (a $(k-1)$-clique).

CPM - Definition



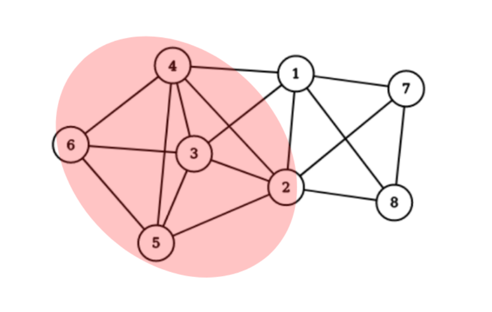
k-clique

Set of k nodes all connected to each other.

Adjacent k-cliques

Two k-cliques are adjacent if they share $k-1$ nodes (a $(k-1)$-clique).

CPM - Definition



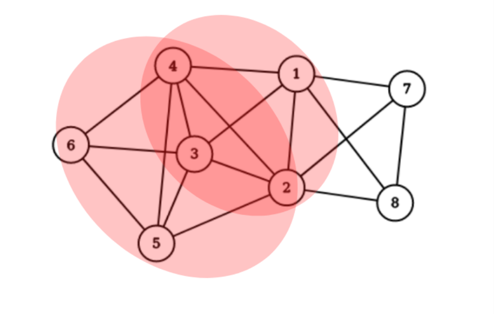
k-clique

Set of k nodes all connected to each other.

Adjacent k-cliques

Two k-cliques are adjacent if they share $k-1$ nodes (a $(k-1)$-clique).

CPM - Definition



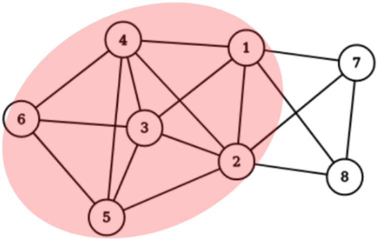
k-clique

Set of k nodes all connected to each other.

Adjacent k-cliques

Two k-cliques are adjacent if they share $k-1$ nodes (a $(k-1)$-clique).

CPM - Definition



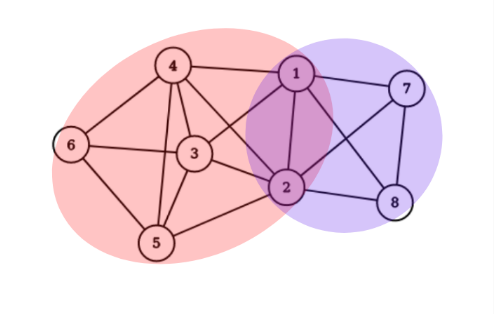
k-clique

Set of k nodes all connected to each other.

Adjacent k-cliques

Two k-cliques are adjacent if they share $k-1$ nodes (a $(k-1)$-clique).

CPM - Definition



k-clique

Set of k nodes all connected to each other.

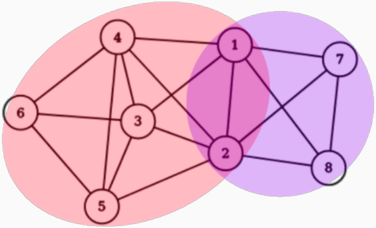
Adjacent k-cliques

Two k-cliques are adjacent if they share $k-1$ nodes (a $(k-1)$-clique).

CPM - Definition

CPM Community

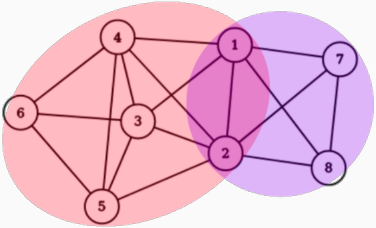
Maximal set of adjacent k-cliques.



CPM - Definition

CPM Community

Maximal set of adjacent k-cliques.



Literature

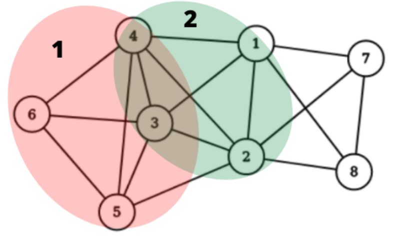
- Palla *et al.* **2005** : first definition
- Kumpula *et al.* **2008** : based on k-clique enumeration
- Reid *et al.* **2012** : based on maximal clique enumeration
- Gregori *et al.* **2013** : parallel version, based on maximal cliques

Exact and approximate algorithms

Exact Algorithm

Exemple :

$k = 4$



$(1,2,3) \to 2$
$(1,2,4) \to 2$
$(1,2,7) \to \times$
$(1,2,8) \to \times$
$(1,3,4) \to 2$

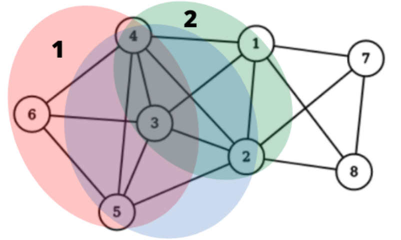
$(1,7,8) \to \times$
$(2,3,4) \to 2$
$(2,3,5) \to \times$
$(2,4,5) \to \times$
$(2,7,8) \to \times$

$(3,4,5) \to 1$
$(3,4,6) \to 1$
$(3,5,6) \to 1$
$(4,5,6) \to 1$

Exact Algorithm

Exemple :

$k = 4$

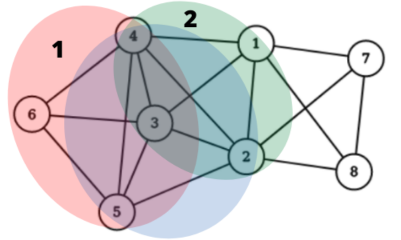


$(1,2,3) \rightarrow 2$	$(1,7,8) \rightarrow \times$	$(3,4,5) \rightarrow 1$
$(1,2,4) \rightarrow 2$	$(2,3,4) \rightarrow 2$	$(3,4,6) \rightarrow 1$
$(1,2,7) \rightarrow \times$	$(2,3,5) \rightarrow \times$	$(3,5,6) \rightarrow 1$
$(1,2,8) \rightarrow \times$	$(2,4,5) \rightarrow \times$	$(4,5,6) \rightarrow 1$
$(1,3,4) \rightarrow 2$	$(2,7,8) \rightarrow \times$	

Exact Algorithm

Exemple :

$k = 4$



$(1,2,3) \rightarrow 2$
$(1,2,4) \rightarrow 2$
$(1,2,7) \rightarrow \times$
$(1,2,8) \rightarrow \times$
$(1,3,4) \rightarrow 2$

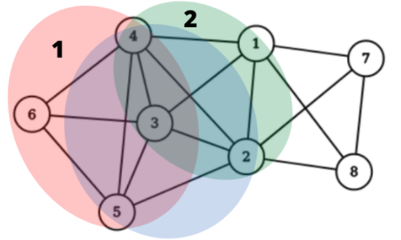
$(1,7,8) \rightarrow \times$
$(2,3,4) \rightarrow 2$
$(2,3,5) \rightarrow \times$
$(2,4,5) \rightarrow \times$
$(2,7,8) \rightarrow \times$

$(3,4,5) \rightarrow 1$
$(3,4,6) \rightarrow 1$
$(3,5,6) \rightarrow 1$
$(4,5,6) \rightarrow 1$

Exact Algorithm

Exemple :

$k = 4$

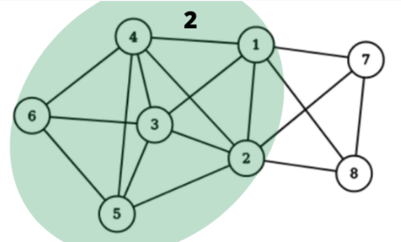


$(1,2,3) \to 2$	$(1,7,8) \to \times$	$(3,4,5) \to 1$
$(1,2,4) \to 2$	$(2,3,4) \to 2$	$(3,4,6) \to 1$
$(1,2,7) \to \times$	$(2,3,5) \to \times$	$(3,5,6) \to 1$
$(1,2,8) \to \times$	$(2,4,5) \to \times$	$(4,5,6) \to 1$
$(1,3,4) \to 2$	$(2,7,8) \to \times$	

Exact Algorithm

Exemple :

$k = 4$



$(1,2,3) \rightarrow 2$
$(1,2,4) \rightarrow 2$
$(1,2,7) \rightarrow \times$
$(1,2,8) \rightarrow \times$
$(1,3,4) \rightarrow 2$

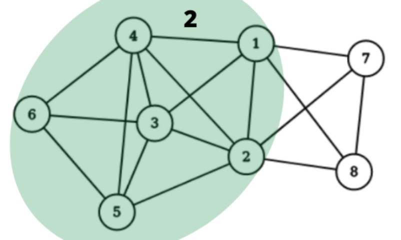
$(1,7,8) \rightarrow \times$
$(2,3,4) \rightarrow 2$
$(2,3,5) \rightarrow 2$
$(2,4,5) \rightarrow 2$
$(2,7,8) \rightarrow \times$

$(3,4,5) \rightarrow 2$
$(3,4,6) \rightarrow 2$
$(3,5,6) \rightarrow 2$
$(4,5,6) \rightarrow 2$

Exact Algorithm

Exemple :

$k = 4$



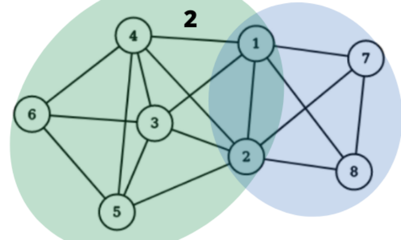
$(1,2,3) \to 2$
$(1,2,4) \to 2$
$(1,2,7) \to \times$
$(1,2,8) \to \times$
$(1,3,4) \to 2$

$(1,7,8) \to \times$
$(2,3,4) \to 2$
$(2,3,5) \to 2$
$(2,4,5) \to 2$
$(2,7,8) \to \times$

$(3,4,5) \to 2$
$(3,4,6) \to 2$
$(3,5,6) \to 2$
$(4,5,6) \to 2$

Exemple :

$k = 4$

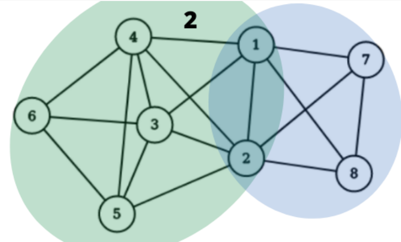


$(1,2,3) \rightarrow 2$	$(1,7,8) \rightarrow \times$	$(3,4,5) \rightarrow 2$
$(1,2,4) \rightarrow 2$	$(2,3,4) \rightarrow 2$	$(3,4,6) \rightarrow 2$
$(1,2,7) \rightarrow \times$	$(2,3,5) \rightarrow 2$	$(3,5,6) \rightarrow 2$
$(1,2,8) \rightarrow \times$	$(2,4,5) \rightarrow 2$	$(4,5,6) \rightarrow 2$
$(1,3,4) \rightarrow 2$	$(2,7,8) \rightarrow \times$	

Exact Algorithm

Exemple :

$k = 4$



$(1,2,3) \rightarrow 2$
$(1,2,4) \rightarrow 2$
$(1,2,7) \rightarrow \times$
$(1,2,8) \rightarrow \times$
$(1,3,4) \rightarrow 2$

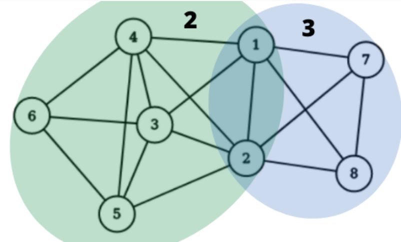
$(1,7,8) \rightarrow \times$
$(2,3,4) \rightarrow 2$
$(2,3,5) \rightarrow 2$
$(2,4,5) \rightarrow 2$
$(2,7,8) \rightarrow \times$

$(3,4,5) \rightarrow 2$
$(3,4,6) \rightarrow 2$
$(3,5,6) \rightarrow 2$
$(4,5,6) \rightarrow 2$

Exact Algorithm

Exemple :

$k = 4$



$(1, 2, 3) \to 2$
$(1, 2, 4) \to 2$
$(1, 2, 7) \to 3$
$(1, 2, 8) \to 3$
$(1, 3, 4) \to 2$

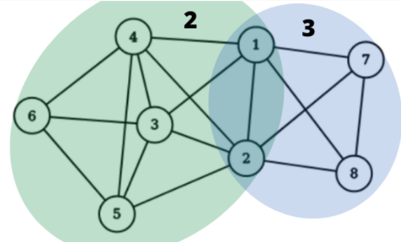
$(1, 7, 8) \to 3$
$(2, 3, 4) \to 2$
$(2, 3, 5) \to 2$
$(2, 4, 5) \to 2$
$(2, 7, 8) \to 3$

$(3, 4, 5) \to 2$
$(3, 4, 6) \to 2$
$(3, 5, 6) \to 2$
$(4, 5, 6) \to 2$

Exact Algorithm

Exemple :

$k = 4$

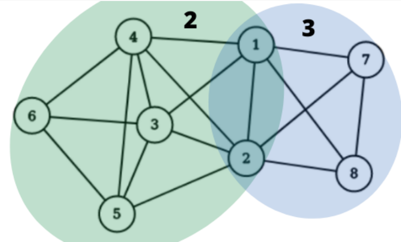


$(1,2,3) \rightarrow 2$	$(1,7,8) \rightarrow 3$	$(3,4,5) \rightarrow 2$
$(1,2,4) \rightarrow 2$	$(2,3,4) \rightarrow 2$	$(3,4,6) \rightarrow 2$
$(1,2,7) \rightarrow 3$	$(2,3,5) \rightarrow 2$	$(3,5,6) \rightarrow 2$
$(1,2,8) \rightarrow 3$	$(2,4,5) \rightarrow 2$	$(4,5,6) \rightarrow 2$
$(1,3,4) \rightarrow 2$	$(2,7,8) \rightarrow 3$	

Exact Algorithm

Exemple :

$k = 4$



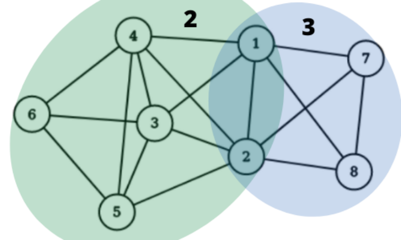
$(1, 2, 3) \to 2$	$(1, 7, 8) \to 3$	$(3, 4, 5) \to 2$
$(1, 2, 4) \to 2$	$(2, 3, 4) \to 2$	$(3, 4, 6) \to 2$
$(1, 2, 7) \to 3$	$(2, 3, 5) \to 2$	$(3, 5, 6) \to 2$
$(1, 2, 8) \to 3$	$(2, 4, 5) \to 2$	$(4, 5, 6) \to 2$
$(1, 3, 4) \to 2$	$(2, 7, 8) \to 3$	

- Update with **Union-Find** datastructure

Exact Algorithm

Exemple :

$k = 4$



$(1,2,3) \rightarrow 2$	$(1,7,8) \rightarrow 3$	$(3,4,5) \rightarrow 2$
$(1,2,4) \rightarrow 2$	$(2,3,4) \rightarrow 2$	$(3,4,6) \rightarrow 2$
$(1,2,7) \rightarrow 3$	$(2,3,5) \rightarrow 2$	$(3,5,6) \rightarrow 2$
$(1,2,8) \rightarrow 3$	$(2,4,5) \rightarrow 2$	$(4,5,6) \rightarrow 2$
$(1,3,4) \rightarrow 2$	$(2,7,8) \rightarrow 3$	

- Update with **Union-Find** datastructure
- Complexity per k-clique : $\mathcal{C}_k \approx \mathcal{O}(k)$

Approximate Algorithm

Limit on massive graphs

Massive graph : the larger k is, the more k-cliques there are.

We work on $k \sim 3 - 15$.

Reduce memory

- Exact algorithm : Store all $\underline{(k-1)\text{-cliques}}$.
- Approximate algorithm : Store all $\underline{z\text{-cliques}}$, $z < k - 1$.

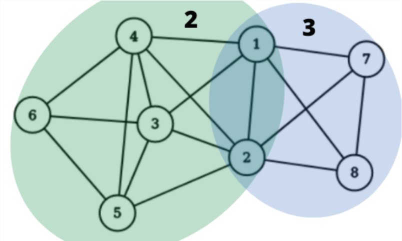
Principle of the Approximate Algorithm

$(k-1)$-clique $\approx$ the set of its z-cliques.

Approximate Algorithm

Example :

$k = 4, z = 2$



$(1,2) \to \{2,3\}$	$(1,8) \to \{3\}$	$(2,7) \to \{3\}$	$(3,6) \to \{2\}$
$(1,3) \to \{2\}$	$(2,3) \to \{2\}$	$(2,8) \to \{3\}$	$(4,5) \to \{2\}$
$(1,4) \to \{2\}$	$(2,4) \to \{2\}$	$(3,4) \to \{2\}$	$(4,6) \to \{2\}$
$(1,7) \to \{3\}$	$(2,5) \to \{2\}$	$(3,5) \to \{2\}$	$(5,6) \to \{2\}$
			$(7,8) \to \{3\}$

- Update with **Union-Find** datastructure
- Complexity per k-clique : $\mathcal{C}_k \approx \mathcal{O}\left(k \cdot \binom{k-1}{z}\right)$

Exact and approximate algorithms

Result of Approximate Algorithm

Some exact communities are merged together.

Accuracy between Exact and Approximate Algorithm

Tool : ONMI *(McAid et al 2013)*

On all the experiments :

$\underline{z=2}$	$\underline{z=3}$
$> 93.8\%$	$> 99.5\%$
Mean : 98.6%	Mean : 99.95%
Median : 99.4%	Median : 100%

Conclusion

Conclusion

Conclusion

- Exact algorithm, gain of time ;
- Almost exact algorithm, gain of memory
  $\Rightarrow$ best scale ;
  $\Rightarrow$ accurate approximate communities.

Conclusion

Conclusion

- Exact algorithm, gain of time ;
- Almost exact algorithm, gain of memory
  $\Rightarrow$ best scale ;
  $\Rightarrow$ accurate approximate communities.

Perspectives

- can bad merges of communities be avoided ?
- order of k-cliques on the accuracy of the result ;
- link streams ;