Correlation Clustering

Authors

Nikhil Bansal, Avrim Blum, Shuchi Chawla

FOCS 2002

Presented by

Jai Pillai

Background

- FOCS – Foundations of Computer Science.

Nikhil Bansal
Research Staff,
IBM Watson Research

Avrim, Blum
Professor, Computer Science
CMU

Shuchi Chawla
Professor, Computer Science
Wisconsin Madison

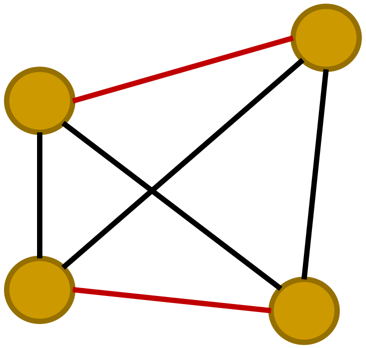
Word of caution – This paper has no empirical results.

Outline

- Problem Definition –
- Properties - NP Hard ☹
- Approximation Algorithm

Problem Definition

- Input
  - A fully connected graph.
  - Vertices are items to be clustered.
  - Edge weights (+ or -)
  - Size of cluster not known



Red – Positive
Black - Negative

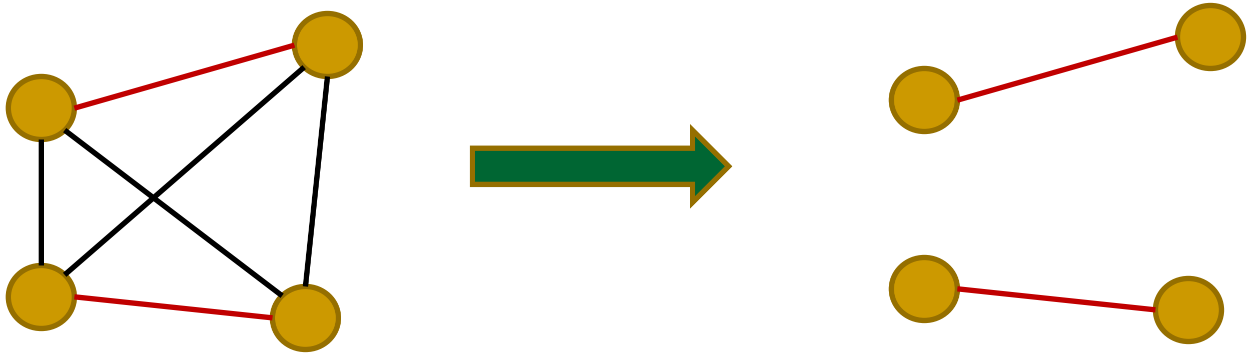
- Desired output
  - Clustering minimizing disagreements.
  - Negative edges within cluster and positive across clusters.

Application

- Consider clustering documents into topics.
- What we donot have
  - Topic information.
  - Similarity measure.
- What we have
  - A binary classifier which says f(A,B) = (+,-)
  - Classifier can make mistakes.

Problem Properties

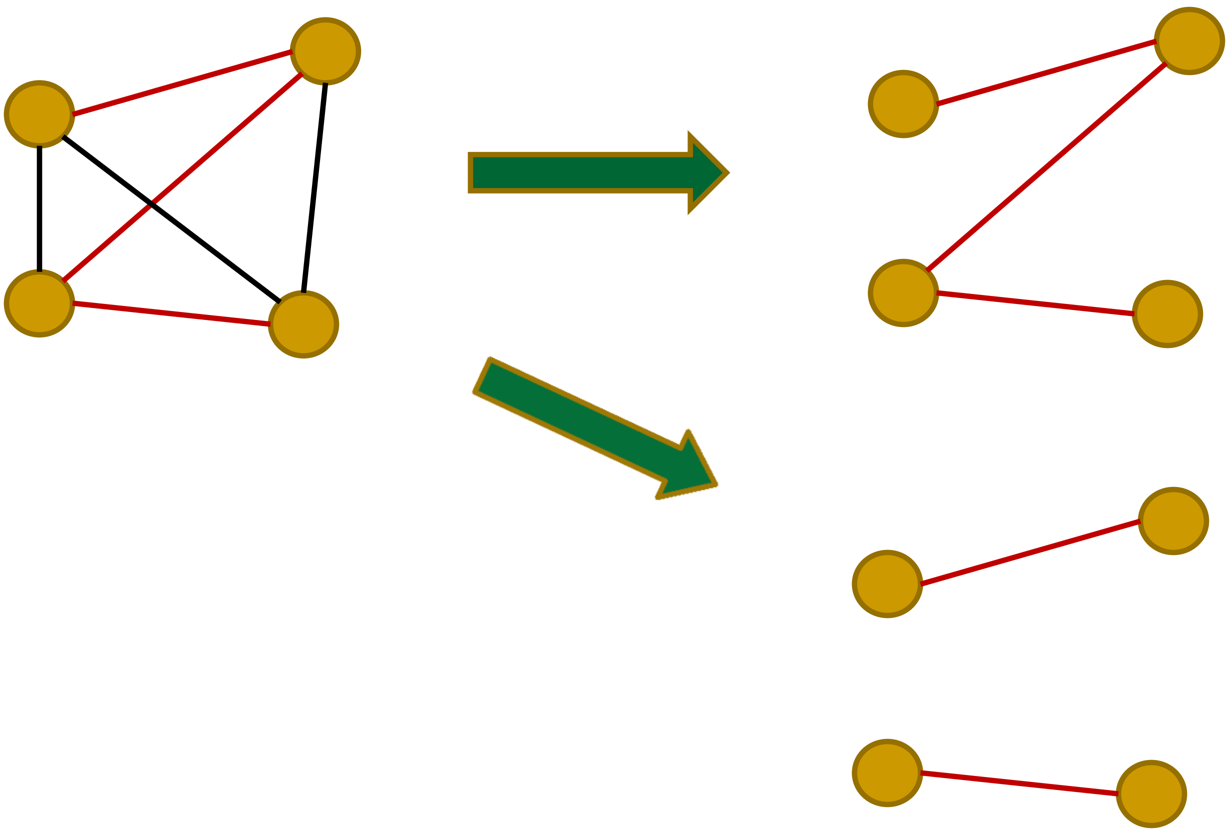
- Trivial solution – anyone?
  - Remove all the negative edges.
  - Retain remaining clusters
  - Minimizes disagreements.



Red – Positive
Black Negative

- But isn't it supposed to be NP hard?

Edge weights need not be consistent



3 disagreements within cluster +
0 agreements across cluster
No of disagreements = 3

0 disagreements within cluster +
1 agreements across cluster
No of disagreements = 1

Problem Properties

- Trivial solution 2 – Agrees with optimal at atleast half the labels?
  - If more positive edges than negative edges, put all vertices in a single cluster.
  - Otherwise, put each vertex as a separate cluster.

Problem properties

Our problem is NP hard.

- Definition
  - NP – Problems that can be verified in polynomial time.
  - NP Hard – Class of problems as hard as the hardest problems in NP.
  - If one NP hard problem can be solved in P, all NP problems can be solved.
- Intuition
  - Exact solution of NP hard problems cannot be found efficiently.

Now what?

- Intuition
  - Exact solution of NP hard problems cannot be found efficiently.
- Trick
  - Go for approximation algorithms
  - Run in polynomial time.
  - Give approximate solution.
  - Prove bounds.
- Coming up
  - Greedy Algorithm for Minimizing Disagreements.

Notations

- Graph $G = (V, E)$
- Positive Neighbor set – "Me and my friends"

$$N^+(u) = \{u\} \cup \{v : e(u, v) = +\}$$

- Negative Neighbor set – "My adversaries"

$$N^-(u) = \{v : e(u, v) = -\}$$

- Optimal Solution OPT
- Set of vertices in same cluster as v - $\mathcal{C}(v)$

Big Picture

- We want to find a clustering with two properties
  - Most of the nodes in my cluster should be my friends.
  - Few of my friends should be outside my cluster.

Quality of vertices and cluster

Definition 1 *A vertex v is called δ-**good** with respect to C, where $C \subseteq V$, if it satisfies the following:*

- $|N^+(v) \cap C| \geq (1 - \delta)|C|$
- $|N^+(v) \cap (V \setminus C)| \leq \delta|C|$

*If a vertex v is not δ-good with respect to (wrt) C, then it is called δ-**bad** wrt C.*

C *is δ-**clean** if all $v \in C$ are δ-good wrt C*

Approximation Bound 1

Result 1 – A clustering with all clusters clean is a near optimal clustering

Lemma 1 *Given a clustering of V in which all clusters are δ-clean for some $\delta \leq 1/4$, then the number of mistakes made by this clustering is at most $8m_{\mathrm{OPT}}$.*

Even this is hard to achieve in polynomial time.

Approximation Bound 2

Result 2 – A clustering with all non singleton clusters clean is a near optimal clustering

Lemma 2 *There exists a clustering* OPT' *in which each non-singleton cluster is* δ*-clean, and* $m_{\mathrm{OPT}'} \leq (\frac{9}{\delta^2} + 1)m_{\mathrm{OPT}}$.

This can be achieved in polynomial time with a Greedy Algorithm. 

Greedy Algorithm

1. Pick a vertex at random.
2. Find its positive neighborhood.
3. Refine its positive neighborhood
    1. Add good vertices.
    2. Remove bad vertices.
4. Use the refined neighborhood as a new cluster.
5. Repeat 1-4 on unclustered vertices.

Greedy Algo - Algorithm Cautious

1. Pick an arbitrary vertex v and do the following:

   (a) Let $A(v) = N^+(v)$.

   (b) (**Vertex Removal Step**): While $\exists x \in A(v)$ such that x is 3δ-bad wrt $A(v)$, $A(v) = A(v) \setminus \{x\}$.

   (c) (**Vertex Addition Step**): Let $Y = \{y | y \in V, y$ is 7δ-good wrt $A(v)\}$. Let $A(v) = A(v) \cup Y$.[2]

2. Delete $A(v)$ from the set of vertices and repeat until no vertices are left or until all the produced sets $A(v)$ are empty. In the latter case, output the remaining vertices as singleton nodes.

Analysis of the Greedy Algorithm

- Algorithm produces singleton and non singleton clusters.
- Mistakes associated with singleton clusters – called external mistakes.
- Mistakes not associated with singleton clusters – called internal mistakes.

External mistakes are bounded by that of OPT'

Internal mistakes $\leq 8m_{\mathrm{OPT}}$

So total mistakes $m_{Cautious} \leq 9(\frac{1}{\delta^2} + 1)m_{\mathrm{OPT}}$

Continuous Weights

- If weights are from [-1, 1] ,
  - Make weights {-1,1} by thresholding.
  - Apply the same Greedy algorithm.
- Its mistakes are bounded roughly by twice the mistakes of the original greedy algorithm.

Conclusion

- Clustering given just quantized binary information.
- Problem is NP hard.
- Polynomial time greedy approximation algorithm.

- Advantages
  - Bounds mean that algorithm is going to work fairly well irrespective of nature of data.
- Disadvantages
  - No empirical results.
  - Is 9 times the optimal error good enough?