Near-Optimal LP Rounding for Correlation Clustering

Grigory Yaroslavtsev

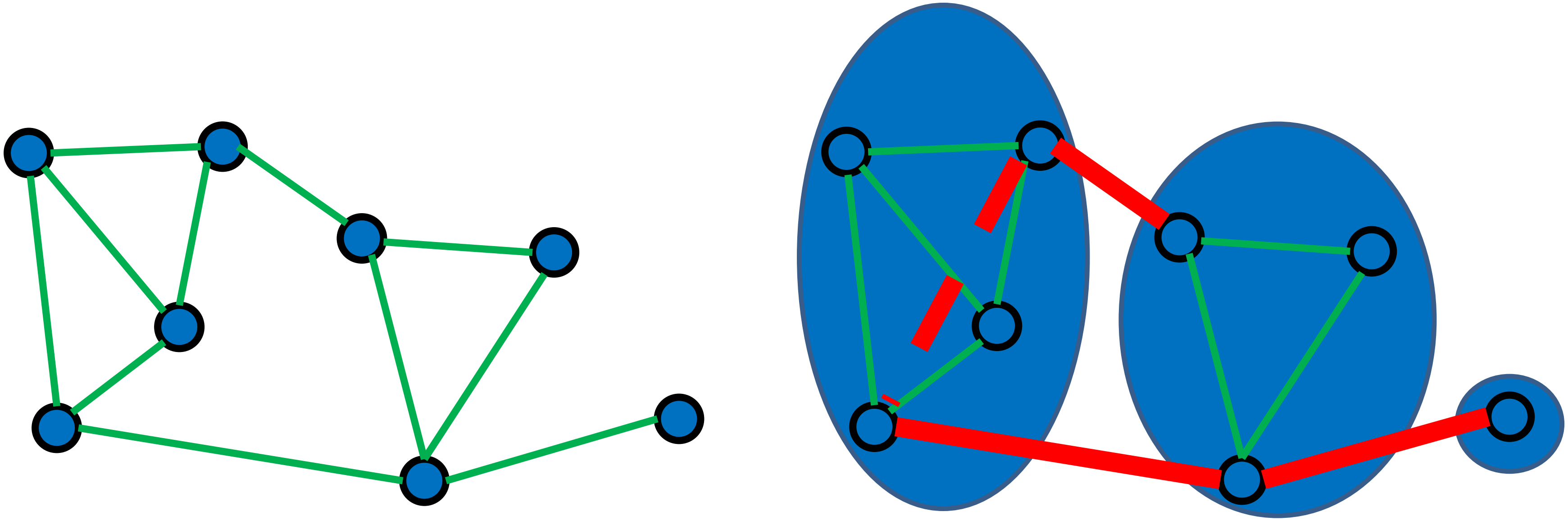
http://grigory.us

University of Pennsylvania

With Shuchi Chawla (University of Wisconsin, Madison),
Konstantin Makarychev (Microsoft Research),
Tselil Schramm (University of California, Berkeley)

Correlation Clustering

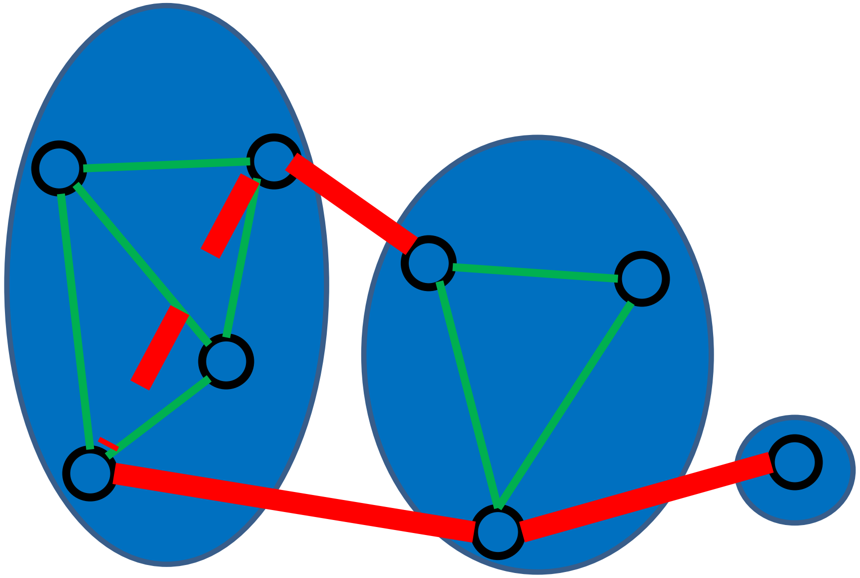
- Inspired by machine learning at WhizBang! LABS
- Practice: [Cohen, McCallum ‘01, Cohen, Richman ’02]
- Theory: **[Blum, Bansal, Chawla ’04]**



Correlation Clustering: Example

- **Minimize** # of **incorrectly** classified pairs:

  # Covered non-edges + # Non-covered edges



4 incorrectly classified =
1 covered non-edge +
3 non-covered edges

- Min-CSP, but # labels is unbounded

Approximating Correlation Clustering

- **Minimize** # of **incorrectly** classified pairs
  - ≈ 20000-approximation [Blum, Bansal, Chawla'04]
  - [Demaine, Emmanuel, Fiat, Immorlica'04],[Charikar, Guruswami, Wirth'05], [Williamson, van Zuylen'07], [Ailon, Liberty'08],…
  - 2.5 [Ailon, Charikar, Newman'05]
  - APX-hard [Charikar, Guruswami, Wirth'05]
- **Maximize** # of **correctly** classified pairs
  - $(1-\epsilon)$-approximation [Blum, Bansal, Chawla'04]

Correlation Clustering

One of the most successful clustering methods:

- Only uses **qualitative information** about similarities
- **# of clusters unspecified** (selected to best fit data)
- Applications: document/image **deduplication** (data from crowds or black-box machine learning)
- **NP-hard** [Bansal, Blum, Chawla '04], admits **simple approximation algorithms** with good provable guarantees
- **Agnostic learning** problem

Correlation Clustering

More:

- **Survey** [Wirth]
- **KDD'14** tutorial: "Correlation Clustering: From Theory to Practice" [Bonchi, Garcia-Soriano, Liberty] http://francescobonchi.com/CCtuto_kdd14.pdf
- **Wikipedia** article: http://en.wikipedia.org/wiki/Correlation_clustering

Data-Based Randomized Pivoting

3-approximation (expected) [Ailon, Charikar, Newman]

Algorithm:

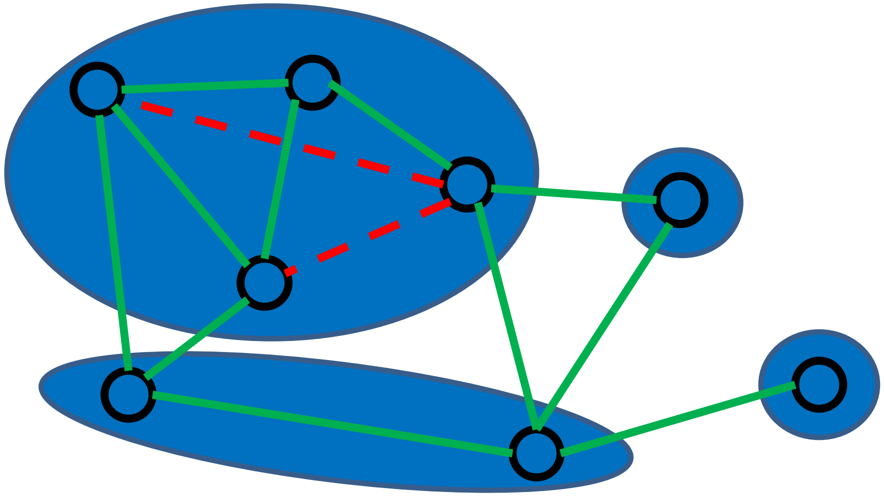
- Pick a random pivot vertex $\boldsymbol{v}$
- Make a cluster $\boldsymbol{v} \cup N(\boldsymbol{v})$, where $N(\boldsymbol{v})$ is the set of neighbors of $\boldsymbol{v}$
- Remove the cluster from the graph and repeat

Modification: $(3 + \epsilon)$-approx. in $O(\log^2 n \,/\, \epsilon)$ rounds of MapReduce [Chierichetti, Dalvi, Kumar, KDD'14]

http://grigory.us/blog/mapreduce-clustering

Data-Based Randomized Pivoting

- Pick a random pivot vertex $\boldsymbol{p}$
- Make a cluster $\boldsymbol{p} \cup N(\boldsymbol{p})$, where $N(\boldsymbol{p})$ is the set of neighbors of $\boldsymbol{p}$
- Remove the cluster from the graph and repeat



8 incorrectly classified =
2 covered non-edges +
6 non-covered edges

Integer Program

Minimize: $\sum_{(u,v)\in E} x_{uv} + \sum_{(u,v)\notin E}(1 - x_{uv})$

$$x_{uv} \le x_{uw} + x_{wv} \qquad \forall u, v, w$$

$$x_{uv} \in \{0,1\}$$

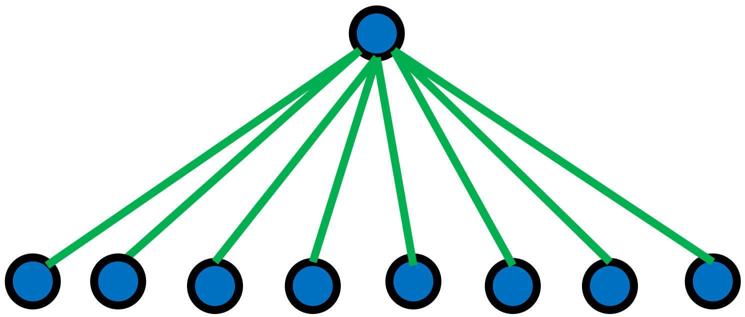
- Binary distance:
  - $x_{uv} = 0 \Leftrightarrow u$ and v in the same cluster
  - $x_{uv} = 1 \Leftrightarrow u$ and v in different clusters
- Objective is exactly MinDisagree
- Triangle inequalities give transitivity:
  - $x_{uw} = 0, x_{wv} = 0 \Rightarrow x_{uv} = 0$
  - $u \sim v$ iff $x_{uv} = 0$ is an equivalence relation, equivalence classes form a partition

Linear Program

- Embed vertices into a (pseudo)metric:

$$\text{Minimize: } \sum_{(u,v)\in E} x_{uv} + \sum_{(u,v)\notin E} (1 - x_{uv})$$
$$x_{uv} \le x_{uw} + x_{wv} \qquad \forall u, v, w$$
$$x_{uv} \in [0,1]$$

- Integrality gap $\geq 2 - o(1)$

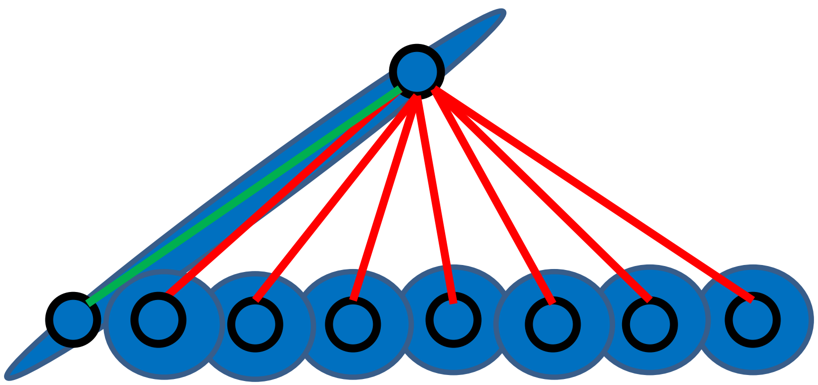


Integrality Gap

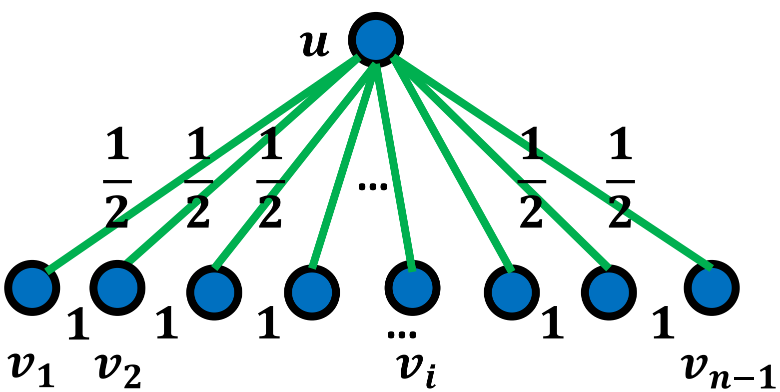
$$\text{Minimize: } \sum_{(u,v)\in E} x_{uv} + \sum_{(u,v)\notin E} (1 - x_{uv})$$

$$x_{uv} \le x_{uw} + x_{wv} \qquad \forall u, v, w$$

$$x_{uv} \in [0,1]$$



- IP cost = n – 2



- LP solution x_{uv}:
  - $\frac{1}{2}$ for edges (u, v_i)
  - **1** for non-edges (v_i, v_j)
  - LP cost = ½ (n - 1)
- IP / LP = 2 – o(1)

Can the LP be rounded optimally?

- **2.06-approximation**
  - Previous: 2.5-approximation [Ailon, Charikar, Newman, JACM’08]
- **3-approximation for objects of k types (comparisons data only between different types)**
  - **Matching 3-integrality gap**
  - Previous: 4-approximation for 2 types [Ailon, Avigdor-Elgrabli, Libety, van Zuylen, SICOMP’11]
- **1.5-approximation for weighted comparison data satisfying triangle inequalities**
  - **Integrality gap 1.2**
  - Previous: 2-approximation [Ailon, Charikar, Newman, JACM’08]

LP-based Pivoting Algorithm [ACN]

Minimize: $\sum_{(u,v)\in E} x_{uv} + \sum_{(u,v)\notin E}(1 - x_{uv})$

$$x_{uv} \le x_{uw} + x_{wv} \qquad \forall u, v, w$$

$$x_{uv} \in [0,1]$$

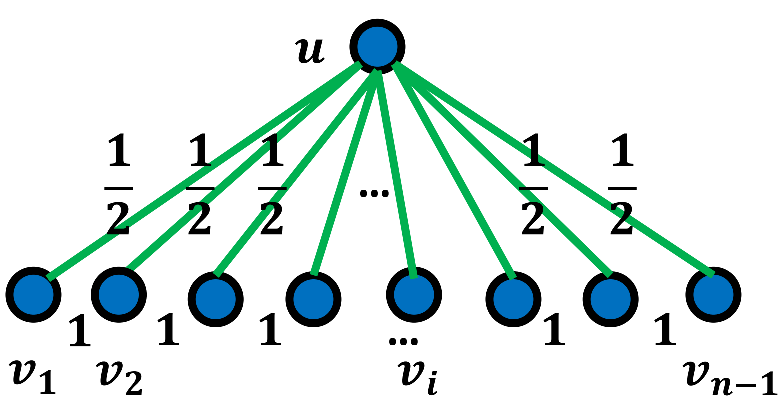
Get all "distances" x_{uv} by solving the LP

- Pick a random pivot vertex $\boldsymbol{p}$
- Let S($\boldsymbol{p}$) be a random set containing every other vertex $\boldsymbol{v}$ with probability $1 - x_{\boldsymbol{p}v}$ (independently)
- Make a cluster $\boldsymbol{p} \cup S(\boldsymbol{p})$
- Remove the cluster from the graph and repeat

LP-based Pivoting Algorithm [ACN]

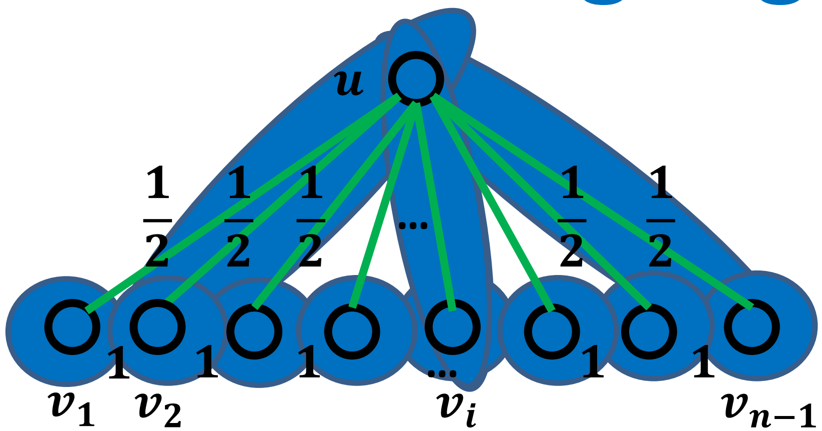
Get all "distances" x_{uv} by solving the LP

- Pick a random pivot vertex $\boldsymbol{p}$
- Let S($\boldsymbol{p}$) be a random set containing every other vertex $\boldsymbol{v}$ with probability $1 - x_{\boldsymbol{p}v}$ (independently)
- Make a cluster $\boldsymbol{p} \cup S(\boldsymbol{p})$
- Remove the cluster from the graph and repeat



- LP solution x_{uv}:
  - $\frac{1}{2}$ for edges (u, v_i)
  - **1** for non-edges (v_i, v_j)
  - LP cost = ½ (n - 1)

LP-based Pivoting Algorithm



- $\boldsymbol{v_i}$ is a pivot (prob. 1 - 1/n)

$$\mathbb{E}[cost|\boldsymbol{v_i} \text{ is a pivot}] \approx ½n \ + \ ½ \ \mathbb{E}[cost]$$

- $\boldsymbol{u}$ is a pivot (prob. 1/n)

$$\mathbb{E}[cost|\boldsymbol{u} \text{ is a pivot}] \approx \frac{n^2}{8}$$

- $\mathbb{E}[cost] \approx \mathbb{E}[cost|\boldsymbol{v_i} \text{ is a pivot}] + \frac{1}{n}\mathbb{E}[cost|\boldsymbol{u} \text{ is a pivot}] =$ $\left(\frac{\text{n}}{2} + \frac{1}{2}\mathbb{E}[cost]\right) + \frac{n}{8} \Rightarrow \ \mathbb{E}[cost] \approx \frac{5n}{4}$
- LP $\approx \frac{n}{2} \ \Rightarrow \ \frac{\mathbb{E}[cost]}{LP} \approx \frac{5}{2} =$ approximation in the ACN analysis

Our (Data + LP)-Based Pivoting

Get all "distances" x_{uv} by solving the LP

- Pick a random pivot vertex $\boldsymbol{p}$
- Let S($\boldsymbol{p}$) be a random set containing every other vertex $\boldsymbol{v}$ with probability $\boldsymbol{f}(x_{\boldsymbol{p}v}, (\boldsymbol{p}, v))$ (independently)
- Make a cluster $\boldsymbol{p} \cup S(\boldsymbol{p})$
- Remove the cluster from the graph and repeat

- Data-Based Pivoting:

$$\boldsymbol{f}(x_{\boldsymbol{p}v}, (\boldsymbol{p}, v)) = \begin{cases} 1, \text{ if } (\boldsymbol{p}, v) \text{ is an edge} \\ 0, \text{ if } (\boldsymbol{p}, v) \text{ is a non-edge} \end{cases}$$

- LP-Based Pivoting:

$$\boldsymbol{f}(x_{\boldsymbol{p}v}, (\boldsymbol{p}, v)) = 1 - x_{\boldsymbol{p}v}$$

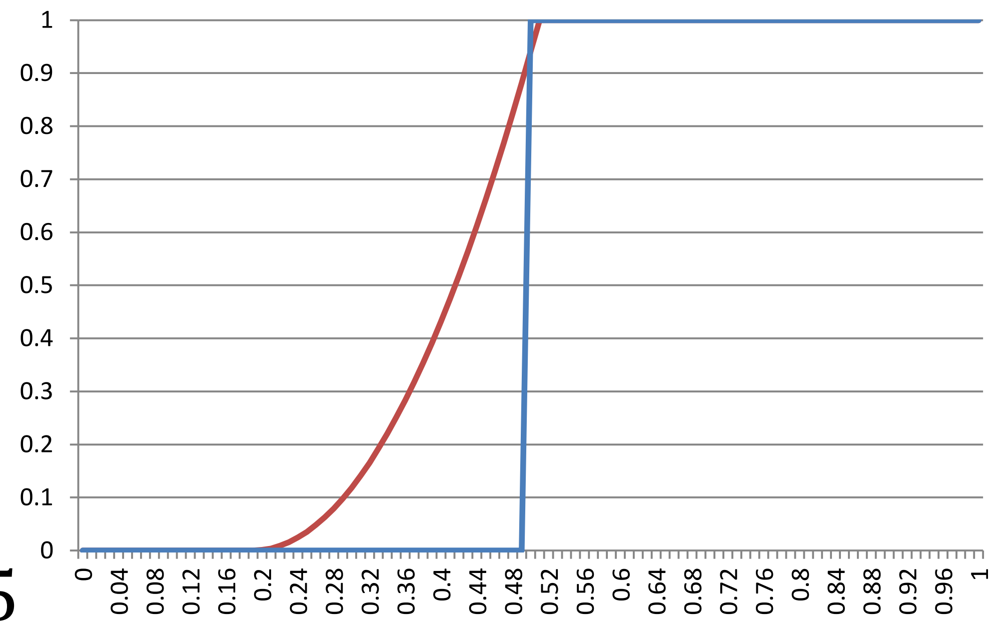
Our (Data + LP)-Based Pivoting

- (Data + LP)-Based Pivoting:

$$f(x_{pv}, (p, v)) = \begin{cases} 1 - f^{+}(x_{pv}), \text{ if } (p, v) \text{ is an edge} \\ 1 - x_{pv}, \text{ if } (p, v) \text{ is a non-edge} \end{cases}$$

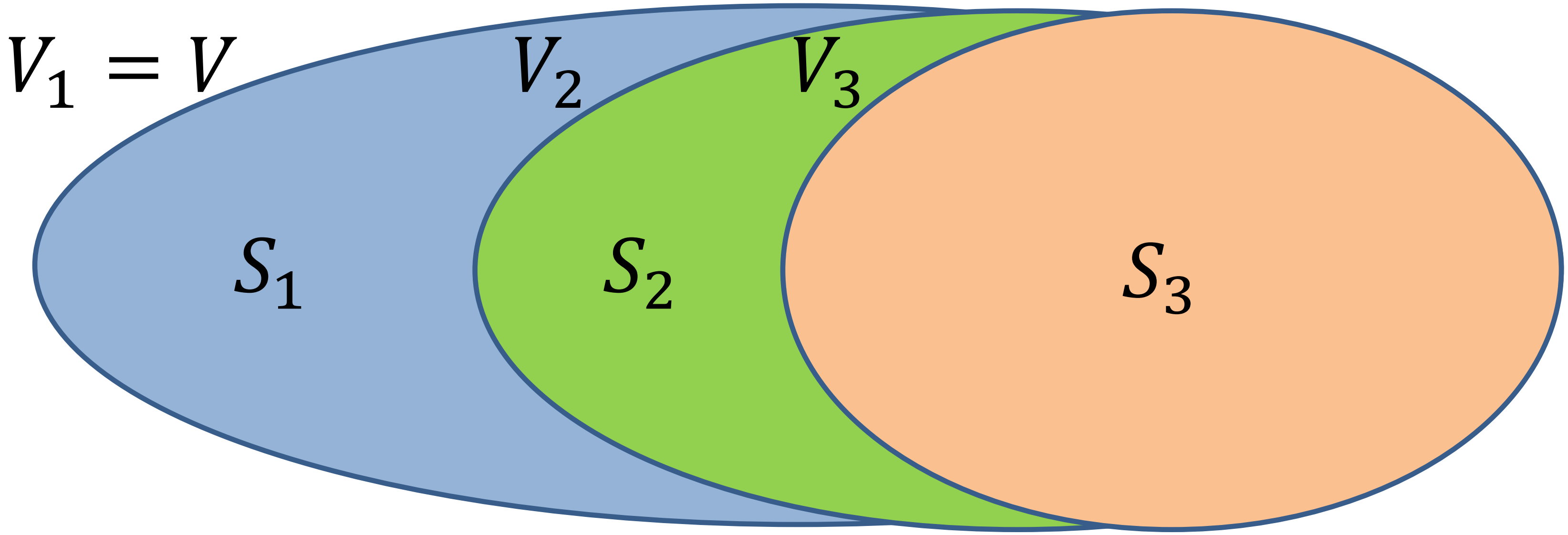
$$f^{+}(x) = \begin{cases} 0, \text{ if } x \le a \\ 1, \text{ if } x \ge b \\ \left(\frac{x - a}{b - a}\right)^2, \text{ otherwise} \end{cases}$$

$$a = 0.19, b = 0.5095$$

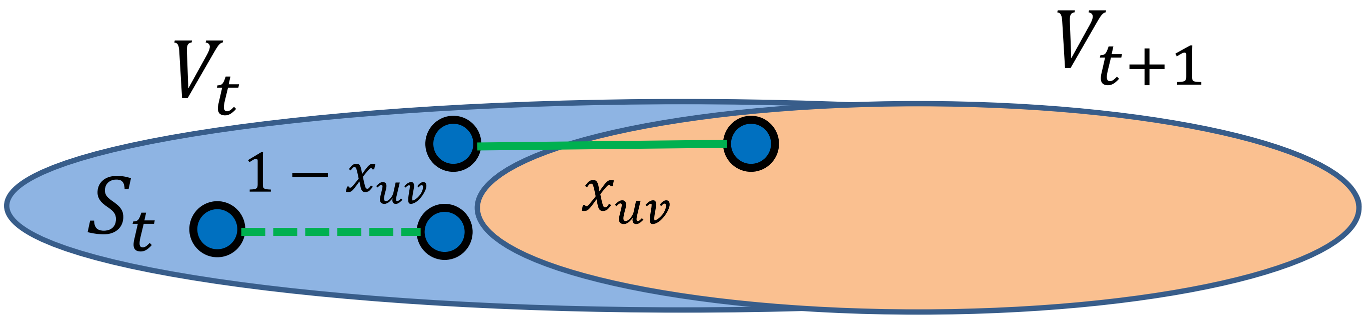


Analysis

- S_t = cluster constructed at pivoting step t
- V_t = set of vertices left before pivoting step t



Analysis

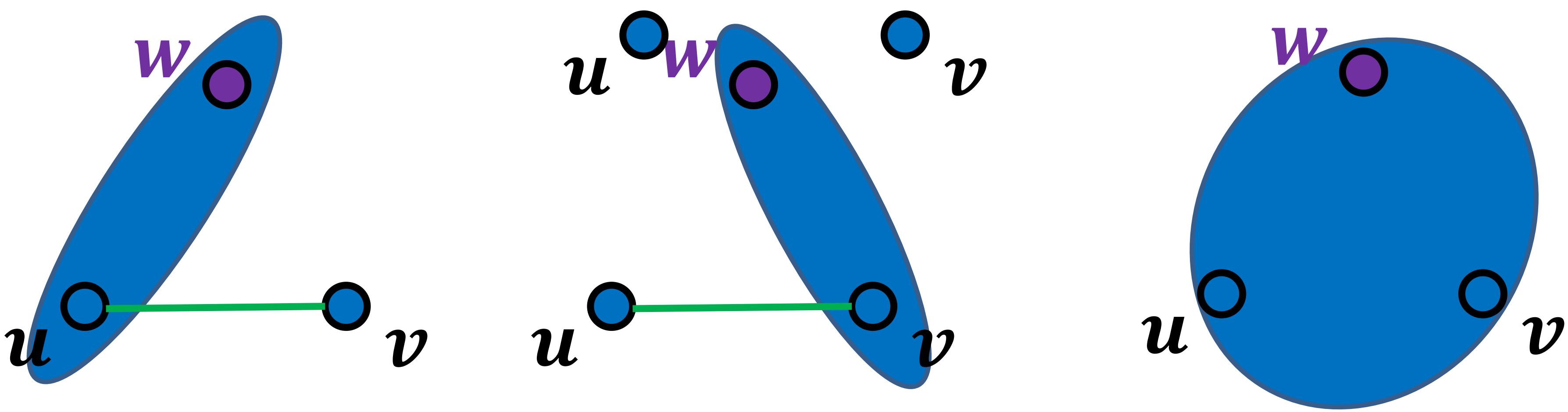


- $ALG_t = \sum_{\substack{(u,v)\in E \\ u,v\in V_t}} \big(\mathbb{1}(u \in S_t, v \notin S_t) + \mathbb{1}(u \notin S_t, v \in S_t)\big) + \sum_{\substack{(u,v)\notin E \\ u,v\in V_t}} \mathbb{1}(u \in S_t, v \in S_t)$
- $LP_t = \sum_{\substack{(u,v)\in E \\ u,v\in V_t}} \mathbb{1}(u \in S_t \text{ or } v \in S_t)\, x_{uv} + \sum_{\substack{(u,v)\notin E \\ u,v\in V_t}} \mathbb{1}(u \in S_t \text{ or } v \in S_t)\,(1 - x_{uv})$
- Suffices to show: $\mathbb{E}[ALG_t] \le \boldsymbol{\alpha}\, \mathbb{E}[LP_t]$
- $\mathbb{E}[ALG] = \mathbb{E}[\sum_t ALG_t] \le \boldsymbol{\alpha}\, \mathbb{E}[\sum_t LP_t] = \boldsymbol{\alpha}\, LP$

Triangle-Based Analysis: Algorithm

- $ALG_{\boldsymbol{w}}(\boldsymbol{u},\boldsymbol{v}) =$
  $\mathbb{E}[error\ on\ (\boldsymbol{u},\boldsymbol{v})|\ \boldsymbol{p} = \boldsymbol{w};\ \boldsymbol{u} \neq\ \boldsymbol{v}, \boldsymbol{w} \in V_t]$

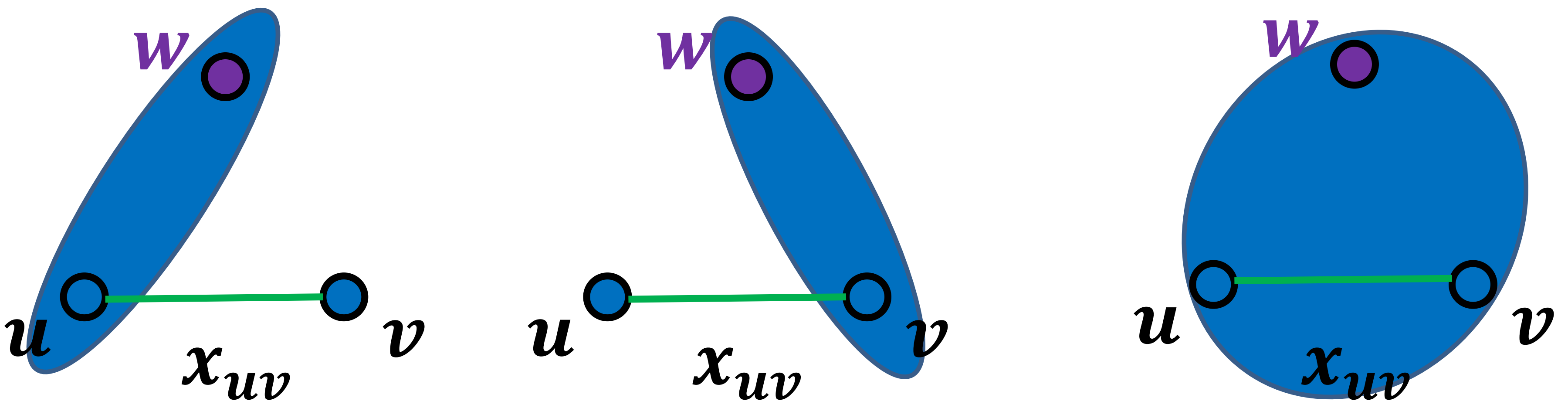
$$= \begin{cases} \boldsymbol{f}(x_{\boldsymbol{wu}})(1 - \boldsymbol{f}(x_{\boldsymbol{wv}})) + \boldsymbol{f}(x_{\boldsymbol{wv}})(1 - \boldsymbol{f}(x_{\boldsymbol{wu}})), & \text{if } (\boldsymbol{u},\boldsymbol{v}) \in E \\ \boldsymbol{f}(x_{\boldsymbol{wu}})\, \boldsymbol{f}(x_{\boldsymbol{wv}}), & \text{if } (\boldsymbol{u},\boldsymbol{v}) \notin E \end{cases}$$



Triangle-Based Analysis: LP

- $LP_{\boldsymbol{w}}(\boldsymbol{u},\boldsymbol{v}) =$
$\mathbb{E}[LP\ contribution\ of\ (\boldsymbol{u},\boldsymbol{v}) |\ \boldsymbol{p} = \boldsymbol{w}; \boldsymbol{u} \neq \boldsymbol{v}, \boldsymbol{w} \in V_t\ ]$

$$= \begin{cases} \big(\boldsymbol{f}(x_{\boldsymbol{wu}}) + \boldsymbol{f}(x_{\boldsymbol{wv}}) - \boldsymbol{f}(x_{\boldsymbol{wu}})\boldsymbol{f}(x_{\boldsymbol{wv}})\big)\boldsymbol{x}_{\boldsymbol{uv}}, & \text{if } (\boldsymbol{u},\boldsymbol{v}) \in E \\ \big(\boldsymbol{f}(x_{\boldsymbol{wu}}) + \boldsymbol{f}(x_{\boldsymbol{wv}}) - \boldsymbol{f}(x_{\boldsymbol{wu}})\boldsymbol{f}(x_{\boldsymbol{wv}})\big)(1 - \boldsymbol{x}_{\boldsymbol{uv}}), & \text{if } (\boldsymbol{u},\boldsymbol{v}) \notin E \end{cases}$$



Triangle-Based Analysis

- $\mathbb{E}[ALG_{\boldsymbol{t}}] = \sum_{\boldsymbol{u},v \in V_t} \left(\frac{1}{|V_t|} \sum_{\boldsymbol{w} \in V_t} ALG_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v}) \right) = \frac{1}{2|\mathrm{V_t}|} \sum_{u,v,w \in V_t, u \neq v} ALG_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v})$
- $\mathbb{E}[LP_{\boldsymbol{t}}] = \sum_{\boldsymbol{u},v \in V_t} \left(\frac{1}{|V_t|} \sum_{\boldsymbol{w} \in V_t} LP_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v}) \right) = \frac{1}{2|\mathrm{V_t}|} \sum_{u,v,w \in V_t, u \neq v} LP_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v})$
- Suffices to show that for all triangles $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$

$$ALG_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v}) \leq \boldsymbol{\alpha} LP_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v})$$

Triangle-Based Analysis

- For all triangles $(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$

$$ALG_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v}) \leq \boldsymbol{\alpha} LP_{\boldsymbol{w}}(\boldsymbol{u}, \boldsymbol{v})$$

- Each triangle:
  - Arbitrary edge / non-edge configuration (4 total)
  - Arbitrary LP weights satisfying triangle inequality
- For every fixed configuration functional inequality in LP weights (3 variables)
- $\boldsymbol{\alpha} \approx 2.06$! $\boldsymbol{\alpha} \geq 2.025$ for **any** $\boldsymbol{f}$!

Our Results: Complete Graphs

Minimize: $\sum_{(u,v)\in E} x_{uv} + \sum_{(u,v)\notin E}(1 - x_{uv})$

$$x_{uv} \leq x_{uw} + x_{wv} \qquad \forall u, v, w$$

$$x_{uv} \in \{0,1\}$$

- **2.06**-approximation for complete graphs
- Can be derandomized (previous: [Hegde, Jain, Williamson, van Zuylen '08])
- Also works for real weights satisfying probability constraints

Our Results: Triangle Inequalities

Minimize: $\sum_{(u,v)}(1 - \boldsymbol{c}_{uv})x_{uv} + \boldsymbol{c}_{uv}(1 - x_{uv})$

$$x_{uv} \le x_{uw} + x_{wv} \qquad \forall u, v, w$$

$$x_{uv} \in \{0,1\}$$

- Weights satisfying triangle inequalities and probability constraints:
  - $\boldsymbol{c}_{uv} \in [0,1]$
  - $\boldsymbol{c}_{uv} \le \boldsymbol{c}_{uw} + \boldsymbol{c}_{wv}\ \forall \mathrm{u}, \mathrm{v}, \mathrm{w}$
- **1.5**-approximation
- **1.2** integrality gap

Our Results: Objects of $\boldsymbol{k}$ types

Minimize: $\sum_{(u,v)\in \boldsymbol{E}} (1 - \boldsymbol{c}_{uv})x_{uv} + \boldsymbol{c}_{uv}(1 - x_{uv})$

$$x_{uv} \leq x_{uw} + x_{wv} \qquad \forall u, v, w$$

$$x_{uv} \in \{0,1\}$$

- Objects of k-types:
  - $\boldsymbol{c}_{uv} \in \{0,1\}$
  - $\boldsymbol{E}$ = edges of a complete $\boldsymbol{k}$-partite graph
- **3**-approximation
- **Matching 3**-integrality gap

Thanks!

Better approximation:

- Can stronger convex relaxations help?
  - Integrality gap for natural Semi-Definite Program is $\geq \frac{1}{2-\sqrt{2}} \approx 1.7$
  - Can LP/SDP hierarchies help?

Better running time:

- Avoid solving LP?
- < 3-approximation in MapReduce?

Related scenarios:

- Better than 4/3-approximation for **consensus clustering**?
- o(log n)-approximation for arbitrary weights (would improve MultiCut, no constant –factor possible under UGC [Chawla, Krauthgamer, Kumar, Rabani, Sivakumar '06])