An Efficient Cluster-Improving Algorithm for Correlation Clustering

Nathan Cordner

Boston University

12 May 2021

Outline

- Correlation Clustering [BBC]
- Weighted CC and CC-Pivot [ACN]
- Probabilistic Graphs and pKwikCluster [KPT]
- Node Algorithms and Cluster Improvement
- Parallel Algorithms
- Constrained Cluster Sizes [PM]

Correlation Clustering [BBC]

Given a complete graph $G = (V, E)$

- $E = E^+ \cup E^-$

Want to cluster $+$ edges and separate $-$ edges

- Maximize Agreements
- **Minimize Disagreements**

Some Applications

- Classification
- Entity Resolution

Algorithm: CC-Pivot [ACN]

CC-Pivot$(G = (V, E = E^+ \cup E^-))$:

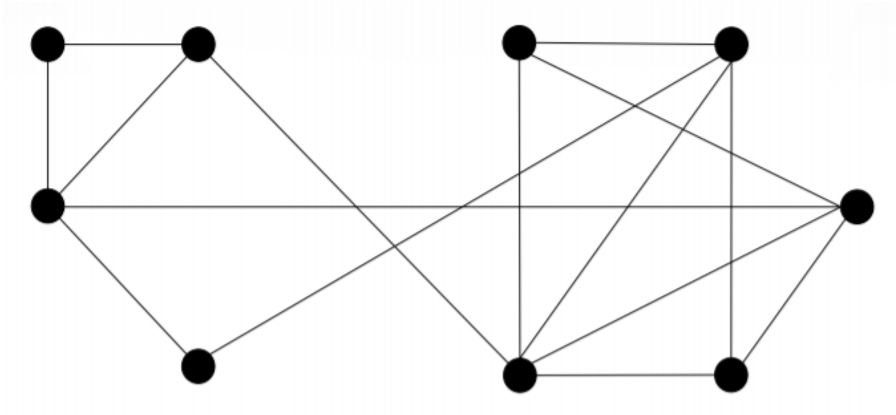
- Pick random pivot $i \in V$
- Set $C = \{i\}$, $V' = \emptyset$
- For all $j \in V \setminus \{i\}$:
  - If $\{i, j\} \in E^+$: Add j to C
  - Else: Add j to V'
- Let G' be the subgraph induced by V'
- Return clustering C, CC-Pivot(G')

Runs in $O(|E|)$ time

Randomized expected 3-approximation

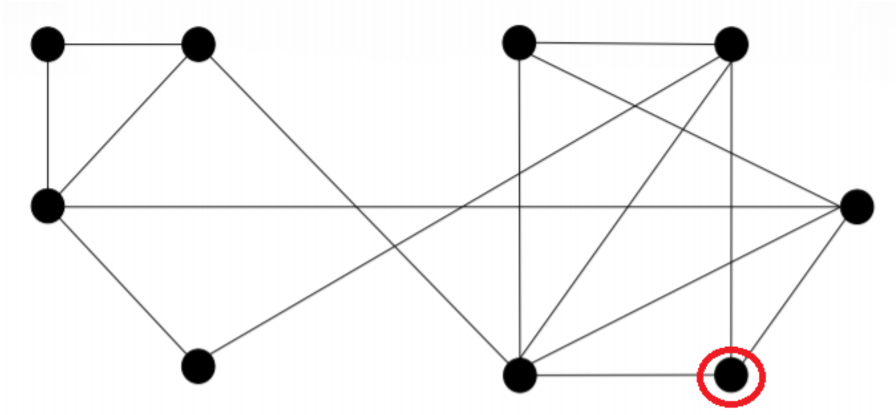
Algorithm: CC-Pivot [ACN]

Example:



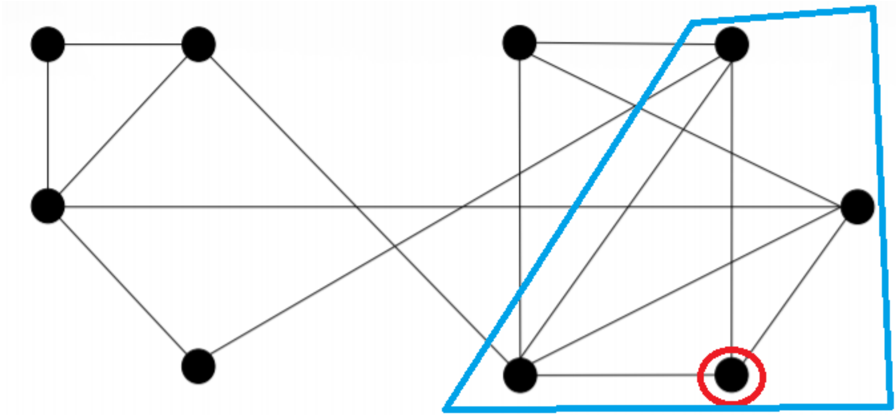
Algorithm: CC-Pivot [ACN]

Example:



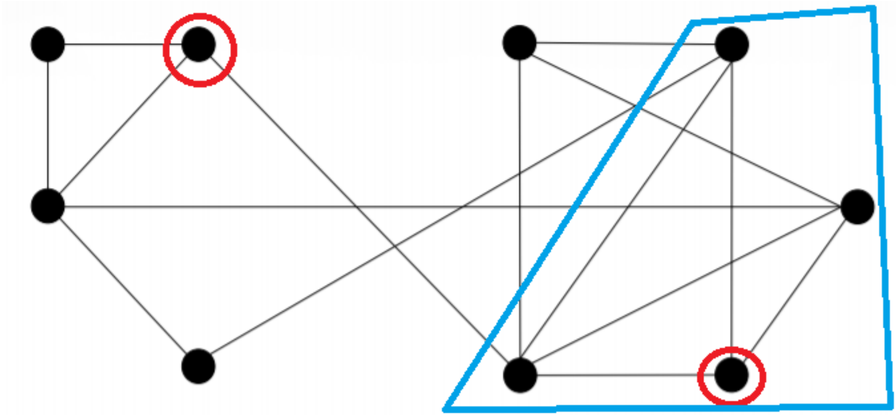
Algorithm: CC-Pivot [ACN]

Example:



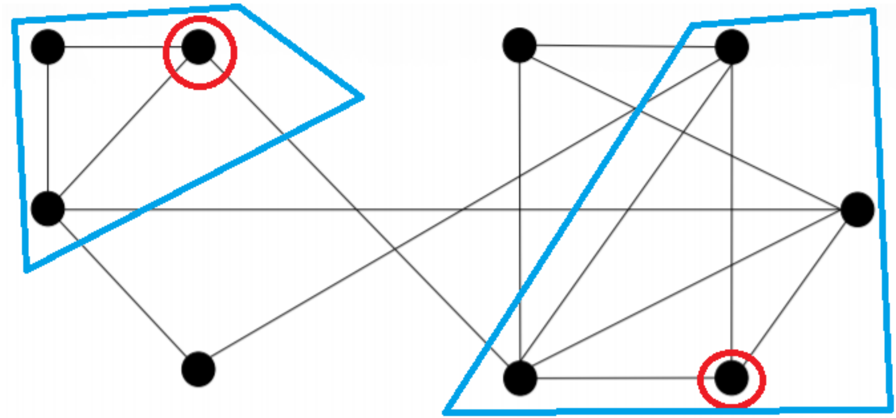
Algorithm: CC-Pivot [ACN]

Example:



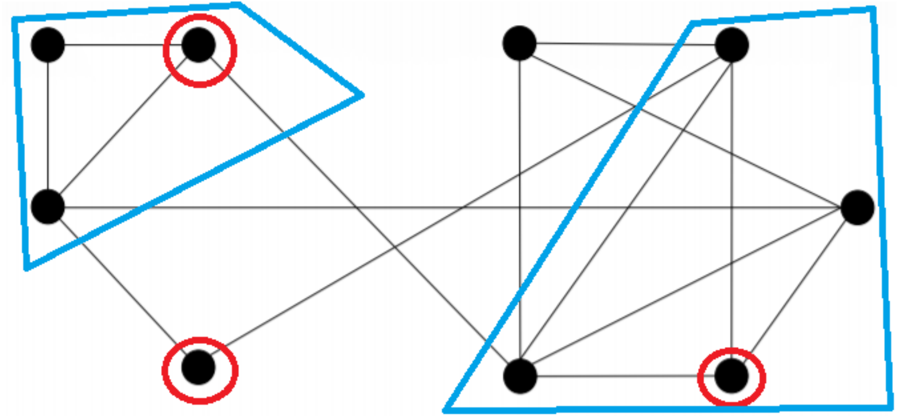
Algorithm: CC-Pivot [ACN]

Example:



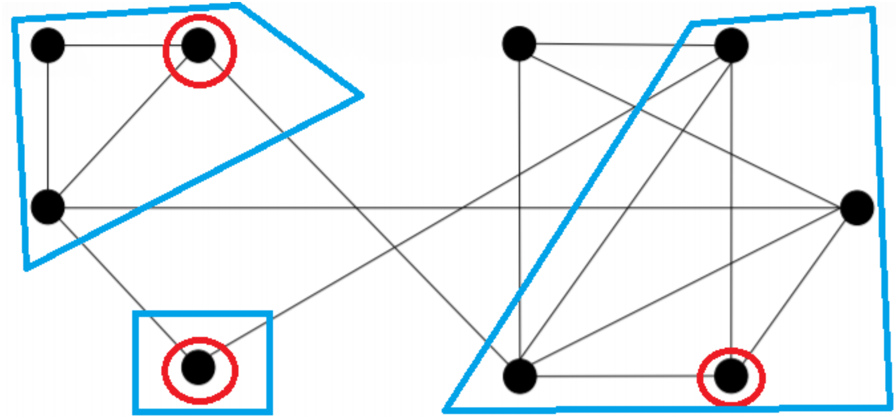
Algorithm: CC-Pivot [ACN]

Example:



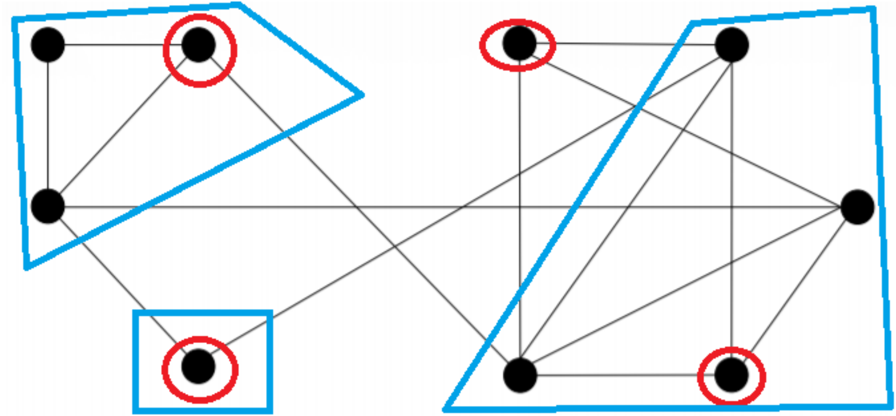
Algorithm: CC-Pivot [ACN]

Example:



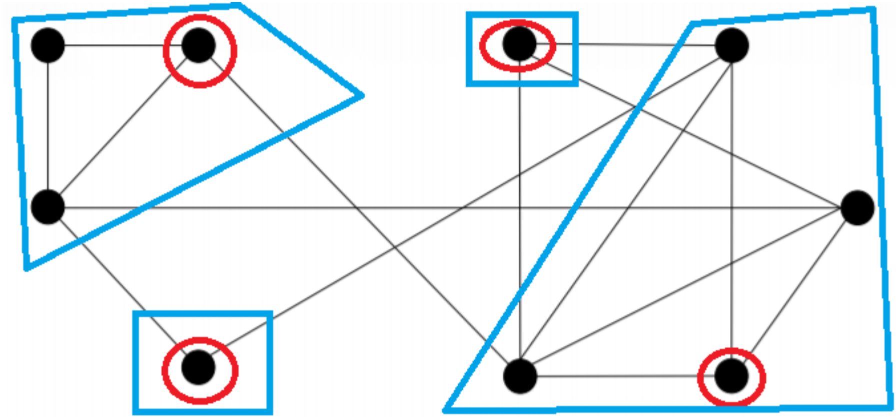
Algorithm: CC-Pivot [ACN]

Example:



Algorithm: CC-Pivot [ACN]

Example:



Weighted Corr. Clustering [BBC, ACN]

Every pair of nodes i, j has weights $w_{ij}^+ \geq 0$ and $w_{ij}^- \geq 0$

- **Probability Constraints**: $w_{ij}^+ + w_{ij}^- = 1$

Clustering Cost:

$$\sum_{i,j \text{ in different clusters}} w_{ij}^+ + \sum_{i,j \text{ in same cluster}} w_{ij}^-$$

Relation to original CC problem:

- $\{i, j\} \in E^+ \Leftrightarrow w_{ij}^+ = 1$ and $w_{ij}^- = 0$
- $\{i, j\} \in E^- \Leftrightarrow w_{ij}^+ = 0$ and $w_{ij}^- = 1$

Extending CC-Pivot [ACN]

Given $G = (V, E, w)$:

- Form the unweighted *majority instance* G_w:
  - Place $\{i, j\}$ in E_w^+ if $w_{ij}^+ > w_{ij}^-$
  - Place $\{i, j\}$ in E_w^- if $w_{ij}^- > w_{ij}^+$
  - Break ties arbitrarily
- Run CC-Pivot on $G_w = (V, E_w = E_w^+ \cup E_w^-)$

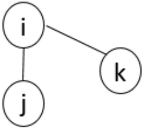
Approximation Results:

- 5-approx if w satisfies the probability constraints

Proof Outline of Approx Bounds [ACN]

"Bad Triangles":

- Two edges are $+$ but one is $-$



Lemma: Approx ratio of CC-Pivot $\leq$ worst cost ratio for bad triangles

- 0/1: $\{i, j, k\}$ has cost 1 but $\{i\}, \{j, k\}$ has cost 3
- Prob: $\{i, j, k\}$ has cost $\frac{1}{2}$ but $\{i\}, \{j, k\}$ has cost $\frac{5}{2}$

Relation to Probabilistic Graphs [KPT]

Probabilistic graph $\mathcal{G} = (V, p)$:

- $p(u, v) =$ probability edge exists between $u, v \in V$
- Possible world $G \sqsubseteq \mathcal{G}$ sampled with probability

$$\Pr(G) = \prod_{\{u,v\} \in E_G} p(u, v) \cdot \prod_{\{u,v\} \notin E_G} (1 - p(u, v))$$

Useful for modelling uncertainty

Relation to Probabilistic Graphs [KPT]

Edit distance: "number of disagreeing edges"

- Between possible worlds

$$D(G, G') = |E_G \setminus E_{G'}| + |E_{G'} \setminus E_G|$$

- Between probabilistic graph and possible world

$$D(\mathcal{G}, G') = \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}}[D(G, G')] = \sum_{G \sqsubseteq \mathcal{G}} \Pr(G) D(G, G')$$

- Efficient calculation

$$D(\mathcal{G}, G') = \sum_{\{u,v\} \in E_{G'}} (1 - p(u, v)) + \sum_{\{u,v\} \notin E_{G'}} p(u, v)$$

Relation to Probabilistic Graphs [KPT]

pClusterEdit: find clustering C that minimizes $D(\mathcal{G}, C)$

- Same objective as CC with probabilistic weights

$$w^+_{uv} = p(u,v),\ w^-_{uv} = 1 - p(u,v)$$

pKwikCluster: CC-Pivot for probabilistic graphs

- cluster nodes with $p(u,v) \geq 1/2$
- Same running time / approximation ratios

pKwik Applications

pKwikCluster has been successfully used on

- Social network graphs [KPT]
- Protein-protein interaction graphs [KPT; HWH]
- Event graphs generated from news stories [CMB]

Other Algorithms for CC Problem

Deterministic CC-Pivot [ZW]

- Fixed “best” order of choosing pivots
- Same approximation ratios as CC-Pivot
- Runs in $O(|V|^3)$ time

LP rounding methods

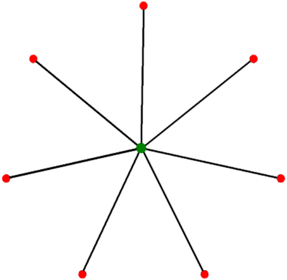
- 2.5-approx for probability weights [ACN]
- 2.06-approx for $0/1$ weights [CMSY]
- Run time dominated by LP solver

CC-Pivot / pKwik still most efficient for large graphs

Drawbacks of CC-Pivot

CC-Pivot / pKwik performs poorly on star graphs

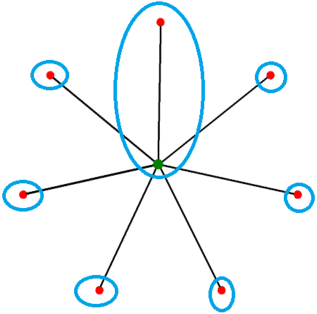
- Expected $1.5 \times$ OPT for $0/1$ star
- Expected $2 \times$ OPT for $1/2$ edge weight star

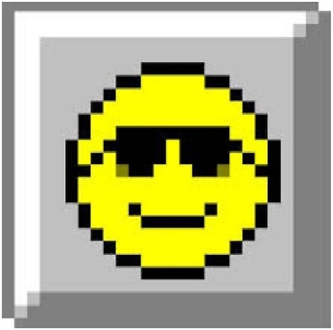


Drawbacks of CC-Pivot

CC-Pivot / pKwik performs poorly on star graphs

- Expected $1.5 \times$ OPT for 0/1 star
- Expected $2 \times$ OPT for 1/2 edge weight star

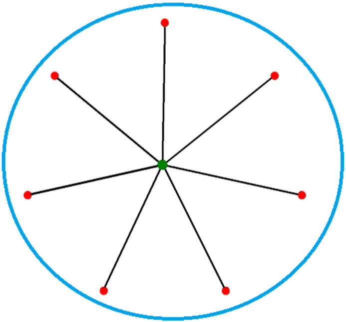




Drawbacks of CC-Pivot

CC-Pivot / pKwik performs poorly on star graphs

- Expected $1.5 \times$ OPT for $0/1$ star
- Expected $2 \times$ OPT for $1/2$ edge weight star



A New Approach: RandomNode*

Pick unclustered nodes one at a time

- First node creates its own cluster
- All others: add to existing cluster, or create own
- Greedily minimize growth of objective function

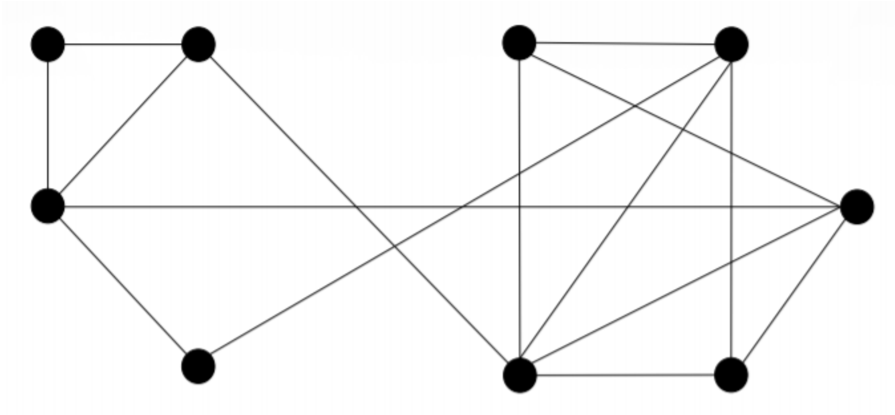
Properties:

- Also linear in the number of edges
- Each cluster has average edge weight $\geq 1/2$

* Inspired by Node algorithm for oracle query reduction [VBD]; similar to greedy algorithm for online CC [MSS]

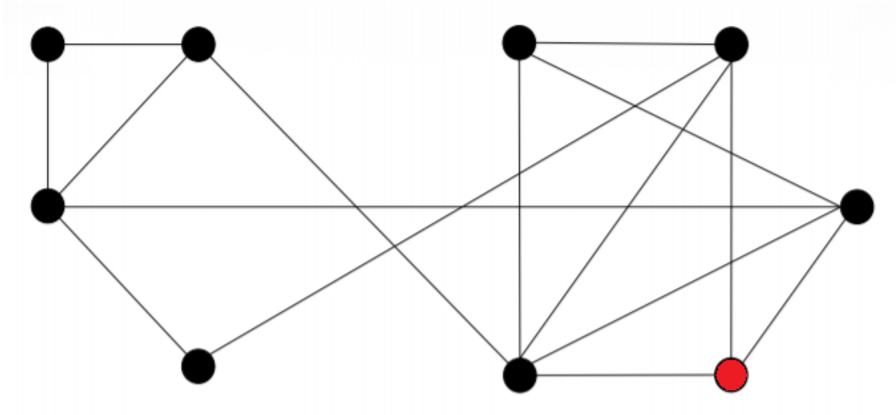
A New Approach: RandomNode

Example review:



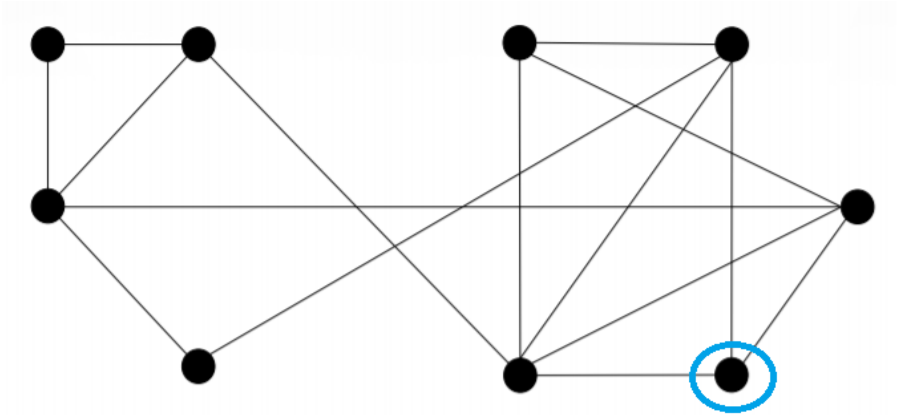
A New Approach: RandomNode

Example review:



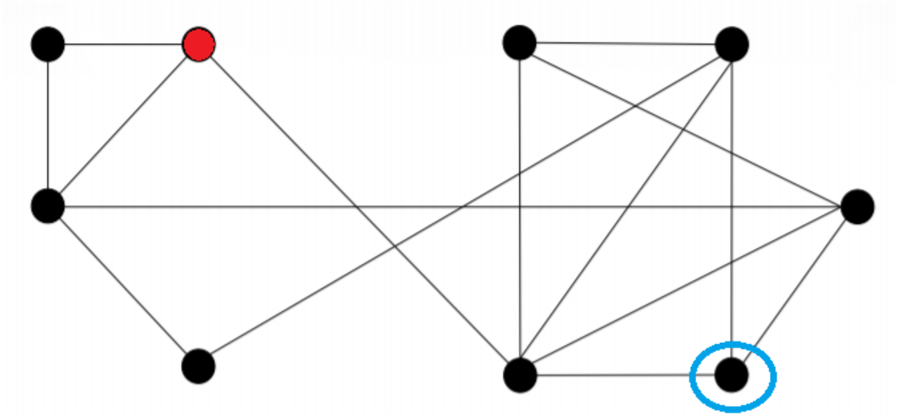
A New Approach: RandomNode

Example review:



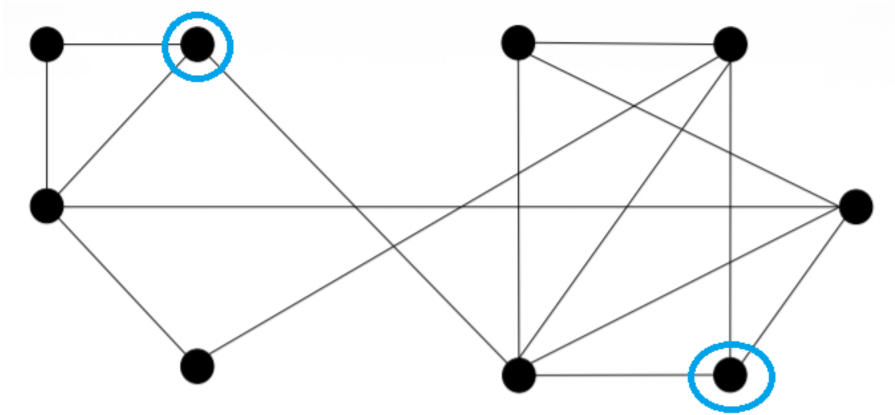
A New Approach: RandomNode

Example review:



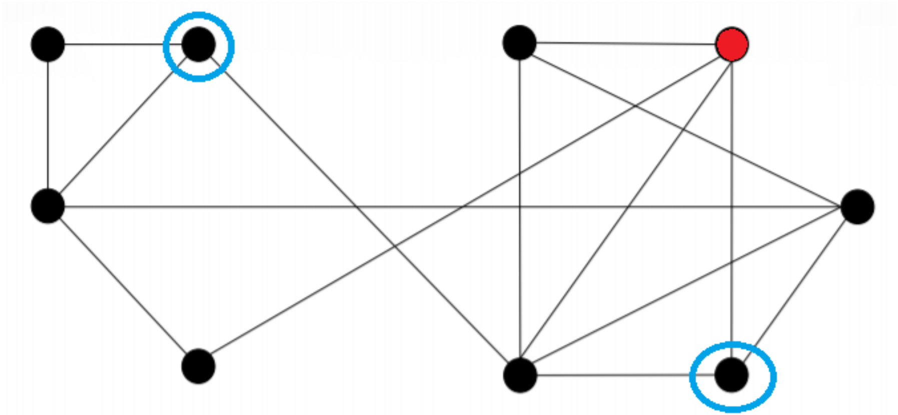
A New Approach: RandomNode

Example review:



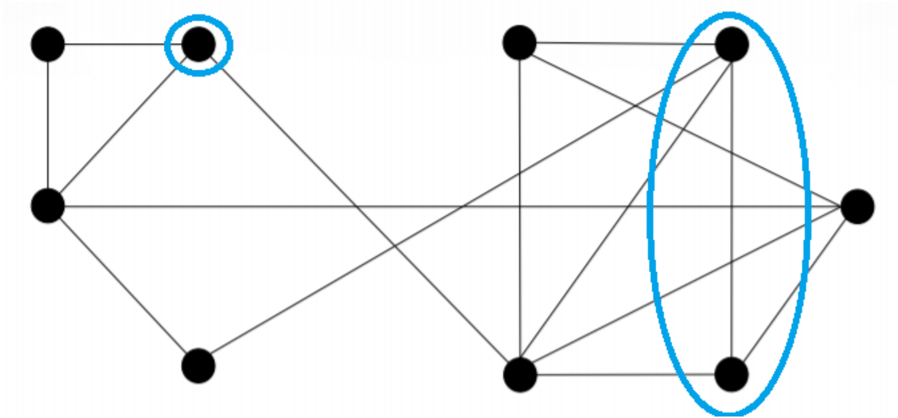
A New Approach: RandomNode

Example review:



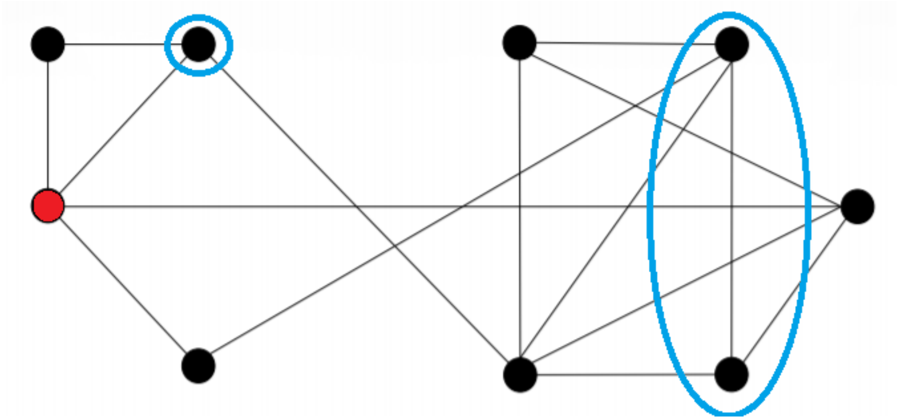
A New Approach: RandomNode

Example review:



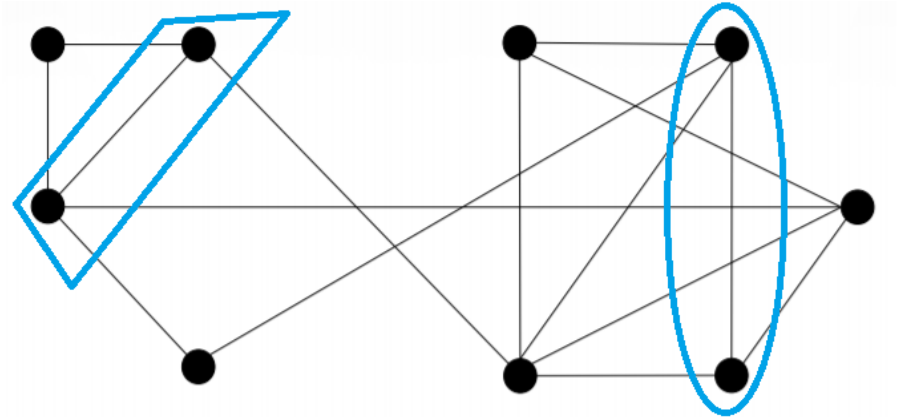
A New Approach: RandomNode

Example review:



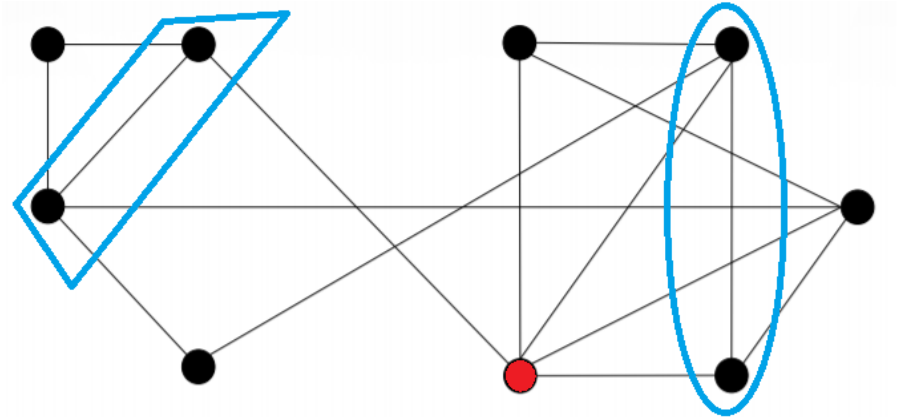
A New Approach: RandomNode

Example review:



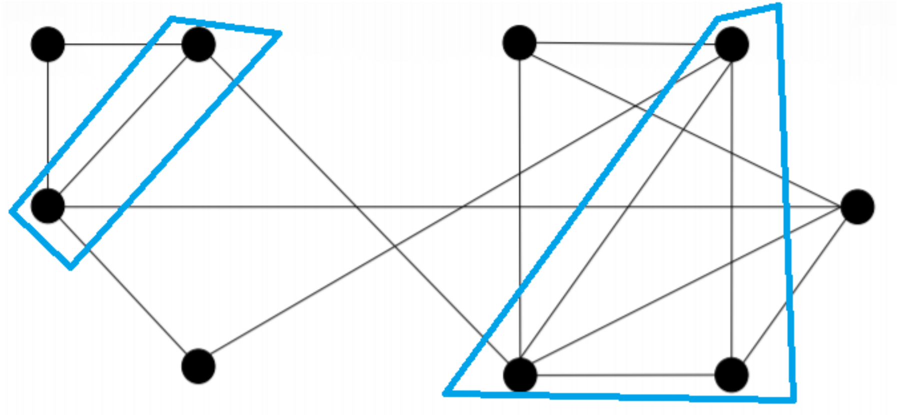
A New Approach: RandomNode

Example review:



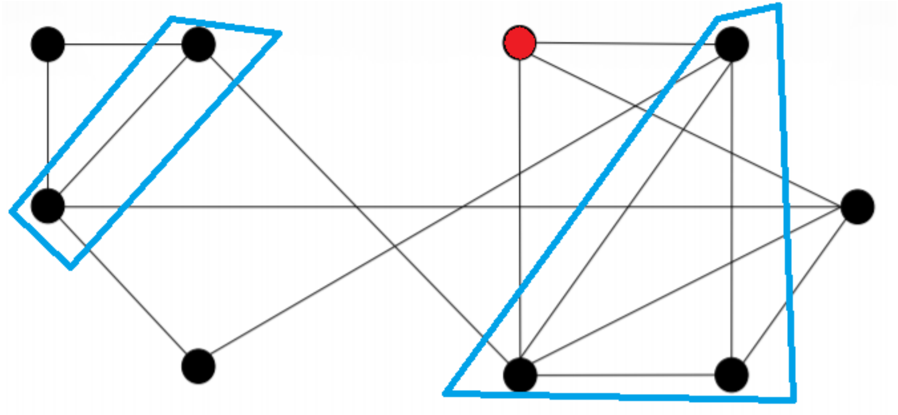
A New Approach: RandomNode

Example review:



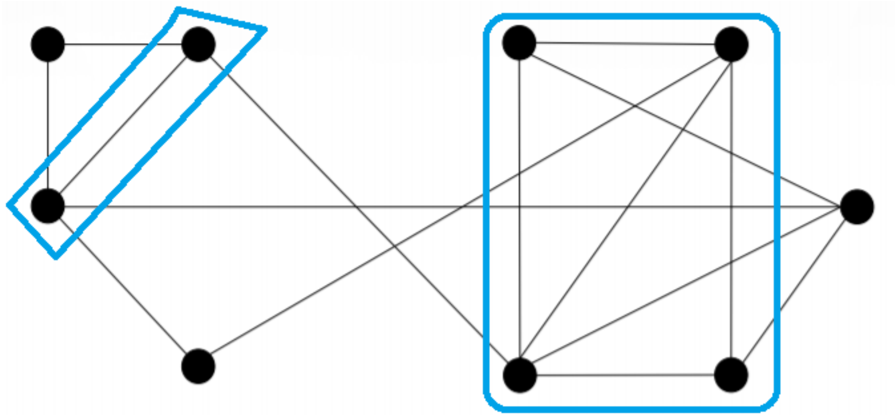
A New Approach: RandomNode

Example review:



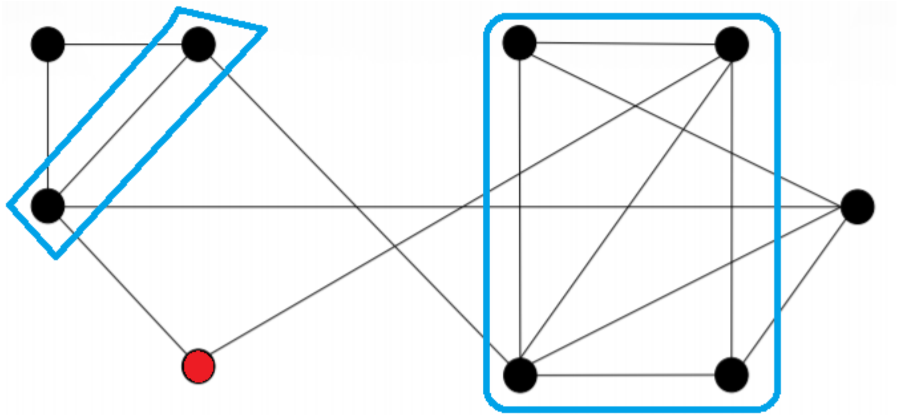
A New Approach: RandomNode

Example review:



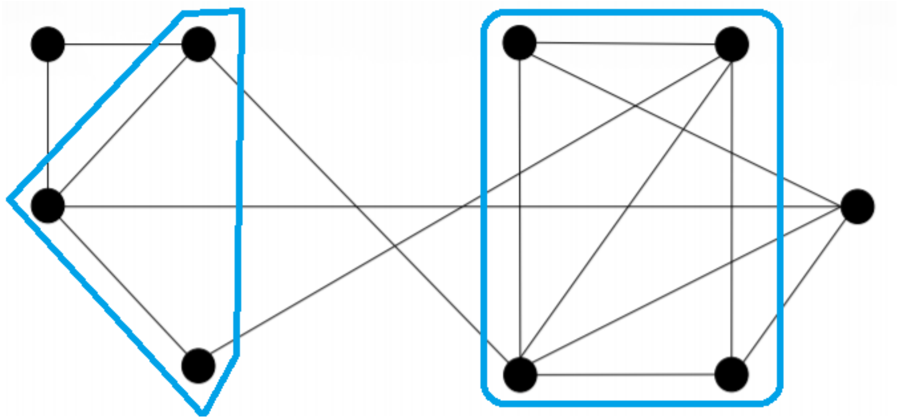
A New Approach: RandomNode

Example review:



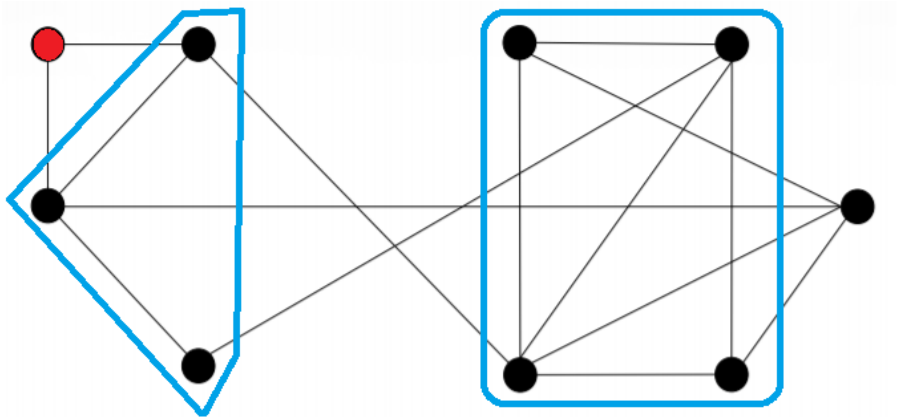
A New Approach: RandomNode

Example review:



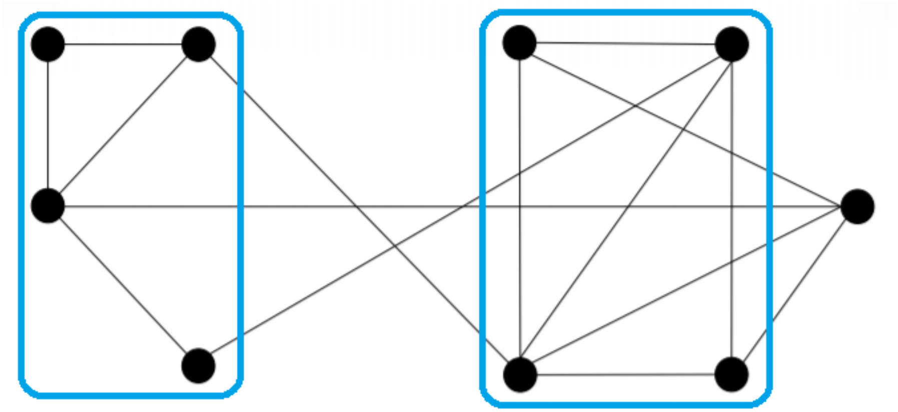
A New Approach: RandomNode

Example review:



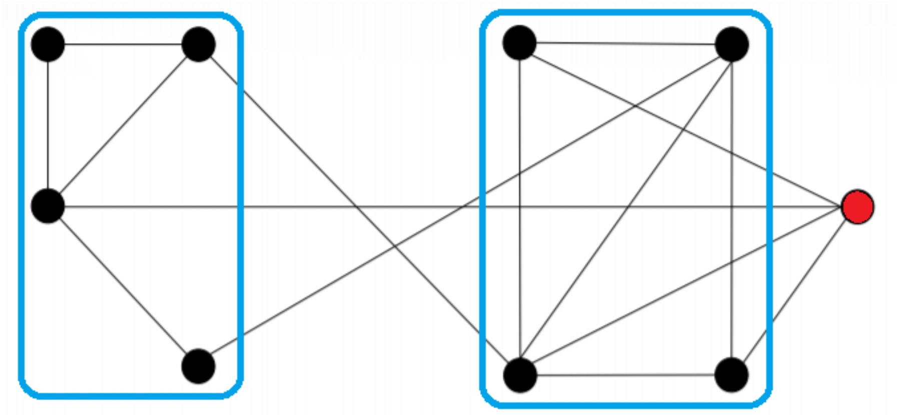
A New Approach: RandomNode

Example review:



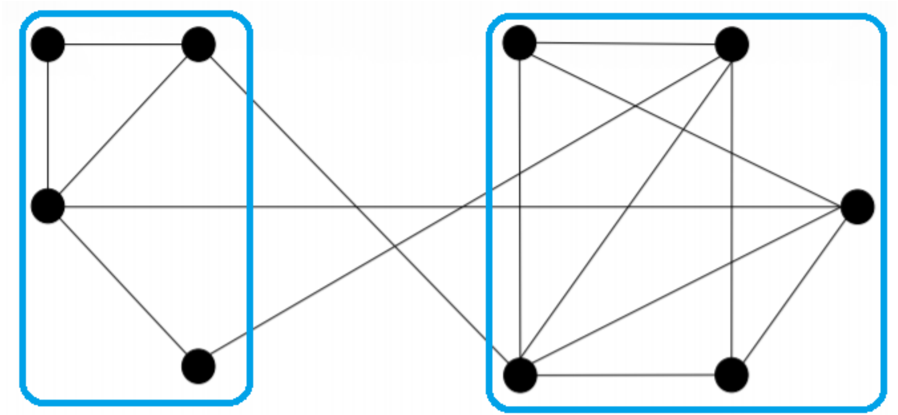
A New Approach: RandomNode

Example review:



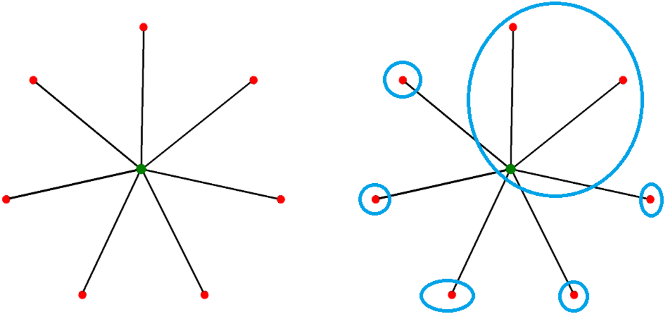
A New Approach: RandomNode

Example review:



A New Approach: RandomNode

Example review: Star graphs



Any ordering produces optimal-valued clustering

A New Approach: RandomNode

Lemma: RandomNode runs in $O(|V|^2) = O(|E|)$ time

Proof: decision for each node requires $O(|V|)$ time

- $S =$ previously settled nodes; $u =$ current node
- Store $\sum_{v \in C} p(u, v)$ for each current cluster C
- Cost of adding to existing cluster C:

$$\sum_{v \in C} (1 - p(u, v)) + \sum_{v \in S \setminus C} p(u, v) = |C| + \sum_{v \in S} p(u, v) - 2 \sum_{v \in C} p(u, v)$$

A New Approach: RandomNode

Lemma: RandomNode clusters have avg weight $\geq 1/2$

Proof: $S =$ settled nodes; $u =$ current node

- Adding u to cluster C implies:

$$\sum_{v \in C} (1 - p(u,v)) + \sum_{v \in S \setminus C} p(u,v) \leq \sum_{v \in S} p(u,v)$$

- Subtract $\sum_{v \in S \setminus C} p(u,v)$:

$$\sum_{v \in C} (1 - p(u,v)) \leq \sum_{v \in C} p(u,v)$$

- Therefore $|C| \leq 2 \sum_{v \in C} p(u,v)$ and $\frac{1}{2} \leq \frac{1}{|C|} \sum_{v \in C} p(u,v)$

Best Ordering: DeterministicNode

Expected Cluster Size [VBD]: $\mathrm{ECS}(u) = \sum_{v \in V \setminus \{u\}} p(u, v)$

Theorem: Ordering nodes greatest to least by ECS is the best ordering for RandomNode (in expectation)

Properties:

- Still linear in $|E|$
  - Compute ECS for each node: $O(|V|^2)$
  - Order nodes by ECS: $O(|V| \log |V|)$
- Edges between clusters now have avg weight $\leq 1/2$

Best Ordering: DeterministicNode

Theorem: ECS is best expected order for RandomNode

Proof: S_i = settled nodes; u_i = current node; $n = |V|$

- Cost increase at iteration i is $\leq \sum_{v \in S_{i-1}} p(u_i, v)$
- Estimate using $\frac{i-1}{n-1} \sum_{v \in V \setminus \{u_i\}} p(u_i, v) = \frac{i-1}{n-1}\mathsf{ECS}(u_i)$
- Total expected cost bounded by $\sum_{i=2}^{n} \frac{i-1}{n-1}\mathsf{ECS}(u_i)$
- Minimized when $\mathsf{ECS}(u_2) \geq \cdots \geq \mathsf{ECS}(u_n)$

Best Ordering: DeterministicNode

Lemma: Edges between clusters have avg weight $\leq 1/2$

Proof:

- First node u_i to form a new cluster satisfies

$$\sum_{v \in C_1} p(u_i, v) < \sum_{v \in C_1} (1 - p(u_i, v)) \Rightarrow \frac{1}{|C_1|} \sum_{v \in C_1} p(u_i, v) < \frac{1}{2}$$

- Approximate value of $\frac{1}{n-1}\mathsf{ECS}(u_i) < 1/2$
- Each subsequent node u_j has $\mathsf{ECS}(u_j) \leq \mathsf{ECS}(u_i)$
- Expected avg edge weight between clusters is

$$\frac{1}{n-1}\mathsf{ECS}(u_j) \leq \frac{1}{n-1}\mathsf{ECS}(u_i) < 1/2$$

Cluster Improvement

Lemma: for any G, the cost of the DeterministicNode clustering of G is $\leq$ the cost of G being one cluster

Proof:

- $A :=$ intra-cluster edges of DNode(G)
- $B :=$ inter-cluster edges of DNode(G)
- $p(e) :=$ weight of edge e
- DNode $\Rightarrow \sum_{e \in B} p(e) \leq |B|/2$
- $\Rightarrow 2 \sum_{e \in B} p(e) \leq |B|$
- $\Rightarrow \sum_{e \in B} p(e) \leq \sum_{e \in B}(1 - p(e))$
- Thus

$$\sum_{e \in A}(1 - p(e)) + \sum_{e \in B} p(e) \leq \sum_{e \in A}(1 - p(e)) + \sum_{e \in B}(1 - p(e))$$

Cluster Improvement

Hybrid Algorithm: on graph G

- Obtain clusters $C_1, \ldots, C_k$ from pKwik(G)
- Let G_i be the graph induced by C_i
- Return DNode$(G_1), \ldots,$ DNode(G_k)

Properties:

- Still linear in $|E|$
  - $\sum_{i=1}^{k} |C_i|^2 \leq \left(\sum_{i=1}^{k} |C_i| \right)^2 = |V|^2$
- Improves cluster scores from pKwikCluster

Experiments

Test Data Sets [FSS, GFSS]:

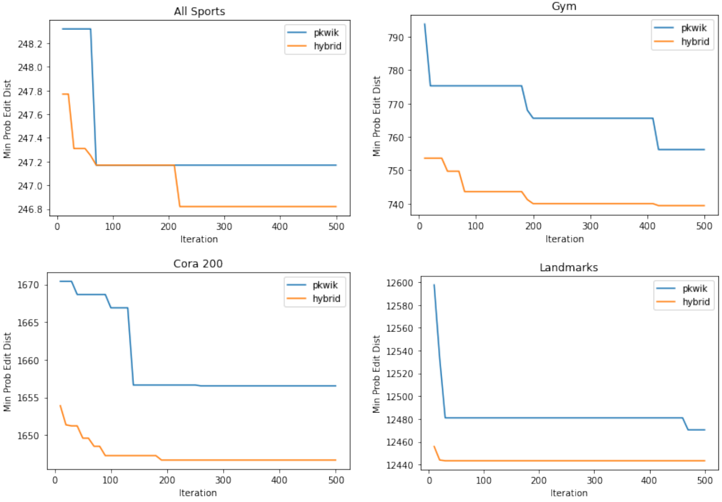
- All Sports: 200 nodes, 19900 edges
  - Images of athletes from 10 different sports
- Gym: 94 nodes, 4371 edges
  - Images of gymnastics athletes
- Cora 200: 190 nodes, 17955 edges
  - Title, author, venue, and date of scientific papers
- Landmarks: 266 nodes, 35245 edges
  - Images of landmarks in Paris and Barcelona

Experiments: pKwik vs Hybrid

pKwik and Hybrid Objective Value Comparison:

- Run pKwikCluster for 500 iterations
- Improve each iteration by hybrid algorithm
- Report rolling minimum score (every 10 iters)

Experiments: pKwik vs Hybrid



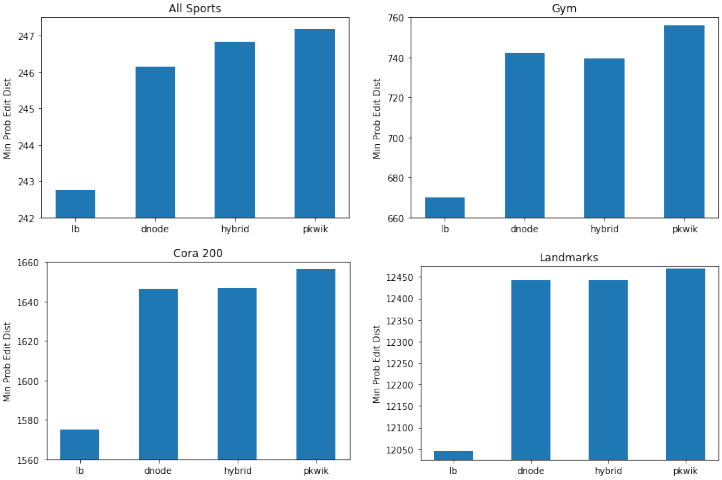
Experiments: Best Value

Best Objective Value Comparison:

- Run DeterministicNode on each graph
- Compare with best results from pKwik and Hybrid
- Lower bound: best clusterings across sets of 3 nodes

$$\mathsf{LB}(G) = \frac{1}{n-2}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=j+1}^{n}\mathsf{best}(i,j,k)$$

Experiments: Best Value

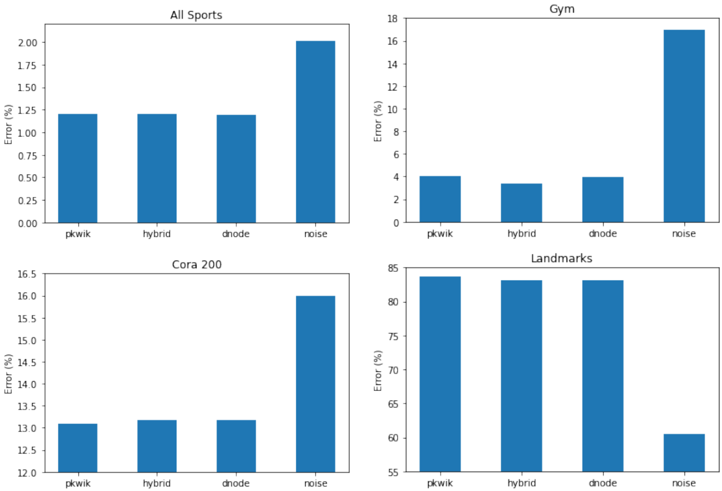


Experiments: Clustering Quality

Check algorithm results against ground truth graph

- Edit distance: $\#$ disagreements between clusterings
- Error rate: edit distance divided by $|E|$
- Noise rate: obj val of ground truth divided by $|E|$
  - G matches ground truth $=$ 0% noise
  - G opposite of ground truth $=$ 100% noise

Experiments: Clustering Quality



Parallel Algorithms

Multi-threaded versions of CC-Pivot [PPORRJ]

- Version 1: Concurrency Control (3-approx)
  - Choose k pivots with a precedence order
  - Form k clusters on parallel threads
  - Award conflicting claims to higher precedence pivot
- Version 2: Coordination-Free ($(3 + \epsilon)$-approx)
  - Sample a small number of working pivots
  - Ignore any edges between sampled pivots
  - Create clusters for each pivot in parallel
- Both versions expected to finish in polylogarithmic number of rounds

Parallel Algorithms

Multi-threaded version of RandomNode

- k threads, n/k rounds
- Stage 1: parallel compute cluster costs for k nodes
  - Keep k lowest cluster choices per node
- Stage 2: add nodes to clustering one at a time
  - Update cluster choices for remaining nodes in parallel
- Runtime: Stage 1 $= O(n)$, Stage 2 $= O(k^2)$
  - Overall: $O(n^2/k + kn)$

DNode: parallel compute ECS and sort in $O(n^2/k)$ time

Parallel Algorithms

Parallel Cluster Improvement

- Run multi-threaded pKwikCluster on single machine
- Distribute resulting clusters to separate machines
- Improve each cluster using multi-threaded DNode
- Collect new clusters together

Constrained Cluster Sizes [PM]

Given integer $K \geq 1$, all clusters must have size $\leq K$

- LP rounding alg: 6-approx
  - All probabilistic graphs
- CC-Pivot adaptation: 7-approx
  - $0/1$ graphs only
  - Efficient version: 11-approx
  - Works well experimentally though

Constrained CC-Pivot [PM]

Constrained CC-Pivot: on graph G with $0/1$ weights,

- "Remove" smallest number of edges from G so that every node has at most $K - 1$ neighbors
- Run CC-Pivot on new graph and return clustering

Probabilistic Graphs: use "majority instance"

- $0/1$ graph formed by rounding edge weights

Constrained CC-Pivot [PM]

Approximation Ratios: CC-Pivot α-approx for G,

- Constrained CC-Pivot: $2\alpha + 1$
- $0/1$ graphs: 7-approx
- Probabilistic graphs: 11-approx

Efficiency:

- Finding smallest edge set: $O(\sqrt{K|V|}|V|^2)$ time
- Instead, let every node choose $K - 1$ neighbors
- Approximation ratio increases to $3\alpha + 2$
- 11 for $0/1$ graphs, 17 for probabilistic graphs

Bounded pKwik / Hybrid

Tweak for Probabilistic Graphs:

- Run pKwikCluster directly on G
- Each node chooses *top* $K - 1$ neighbors

Hybrid approach for cluster improvement:

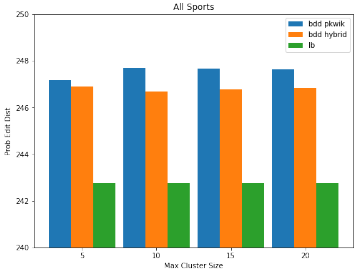
- All clusters from Bounded pKwik have size $\leq K$
- Run DNode on each cluster—sizes can only decrease
- Cluster cost still expected to improve

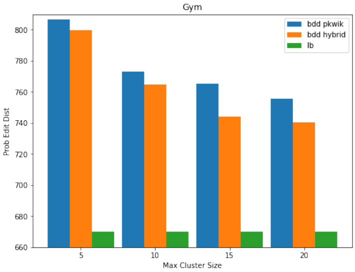
Experiments: Constrained Cluster Sizes

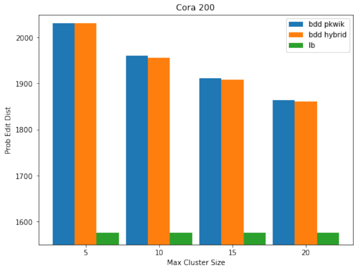
Test Bounded pKwik and Hybrid on four data sets

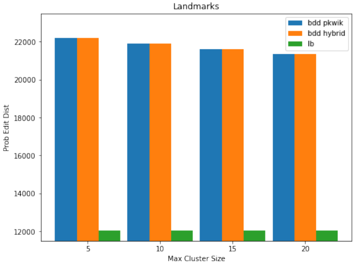
- Max cluster sizes $K = 5, 10, 15$, and 20
- Report best result after 500 iterations
- Compare against the (unconstrained) lower bound
- Algs perform much better than the 17-approx ratio

Experiments: Constrained Cluster Sizes









Future Work

Apply cluster improvement to known CC-Pivot variants

- Data stream model
- Fairness constraints

Find efficient algorithms for other CC variants

- Non-uniform cluster size constraints
- Generalized weight systems
- Number of clusters constraints

References

ACN Ailon, Charikar, and Newman. *Aggregating inconsistent information: ranking and clustering*. 2008

BBC Bansal, Blum, and Chawla. *Correlation clustering*. 2004

CMSY Chawla, Makarychev, Schramm, and Yaroslavtsev. *Near optimal lp rounding algorithm for correlation clustering on complete and complete k-partite graphs*. 2015

CMB Christiansen, Mobasher, and Burke. *Using uncertain graphs to automatically generate event flows from news stories*. 2017

FSS Firmani, Saha, and Srivastava. *Online entity resolution using an oracle*. 2016

GFSS Galhotra, Firmani, Saha, and Srivastava. *Robust entity resolution using random graphs*. 2018

HWH Halim, Waqas, and Hussain. *Clustering large probabilistic graphs using multi-population evolutionary algorithm*. 2015

KPT Kollios, Potamias, and Terzi. *Clustering large probabilistic graphs*. 2011

MSS Mathieu, Sankur, and Schudy. *Online Correlation Clustering*. 2010

PPORRJ Pan, Papailiopoulos, Oymak, Recht, Ramchandran, and Jordan. *Parallel correlation clustering on big graphs*. 2015

PM Puleo and Milenkovic. *Correlation clustering with constrained cluster sizes and extended weights bounds*. 2015

VBD Vesdapunt, Bellare, and Dalvi. *Crowdsourcing algorithms for entity resolution*. 2014

ZW Zuylen and Williamson. *Deterministic pivoting algorithms for constrained ranking and clustering problems*. 2009