Partially Optimal Cubic Subspace Clustering

Research Project Machine Learning

Volodymyr Drobitko

Technische Universität Dresden

21.07.2025

Problem Statement (1)

Finite sample set S, cost function $\mathrm{c}\colon \binom{S}{3} \to \mathbb{R}$.
Instance of the **Cubic Clique Partition Problem**:

$$\min_{\mathrm{y}\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} \mathrm{c}_{\{a,b,c\}}\, \mathrm{y}_{\{a,b\}}\, \mathrm{y}_{\{b,c\}}\, \mathrm{y}_{\{a,c\}}$$

subject to $\mathrm{y}_{\{a,b\}} + \mathrm{y}_{\{b,c\}} - 1 \leq \mathrm{y}_{\{a,c\}}$ for all distinct $a, b, c \in S$.

Problem Statement (1)

Finite sample set S, cost function $\mathrm{c}\colon \binom{S}{3} \to \mathbb{R}$.
Instance of the **Cubic Clique Partition Problem**:

$$\min_{\mathrm{y}\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} \mathrm{c}_{\{a,b,c\}}\, \mathrm{y}_{\{a,b\}}\, \mathrm{y}_{\{b,c\}}\, \mathrm{y}_{\{a,c\}}$$

subject to $\mathrm{y}_{\{a,b\}} + \mathrm{y}_{\{b,c\}} - 1 \leq \mathrm{y}_{\{a,c\}}$ for all distinct $a, b, c \in S$.

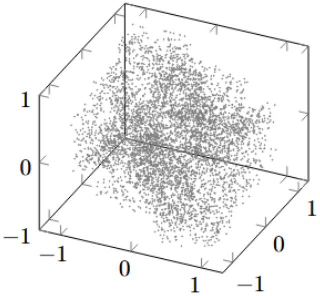
Find a **partially optimal solution**, i.e. fix some labels $\mathrm{y}_{\{a,b\}}$ for distinct $a, b \in S$

$$\begin{cases} \mathrm{y}_{\{a,b\}} = 1 & \text{join } a, b \\ \mathrm{y}_{\{a,b\}} = 0 & \text{cut } a, b \\ \mathrm{y}_{\{a,b\}} = ? & \text{unknown} \end{cases}$$

in such way that there still exists an optimal solution.

Subspace Instances of the Cubic Clique Partition Problem

Samples S: points $S \subset \mathbb{R}^3$

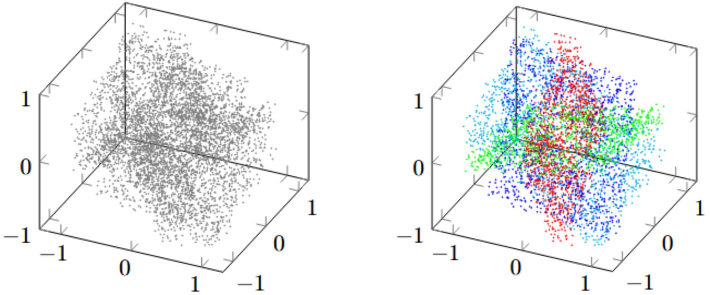


(a) Samples S

Problem Statement (2)

Subspace Instances of the Cubic Clique Partition Problem

Samples S: points $S \subset \mathbb{R}^3$
Point generation: 3 distinct planes containing the origin, noise σ

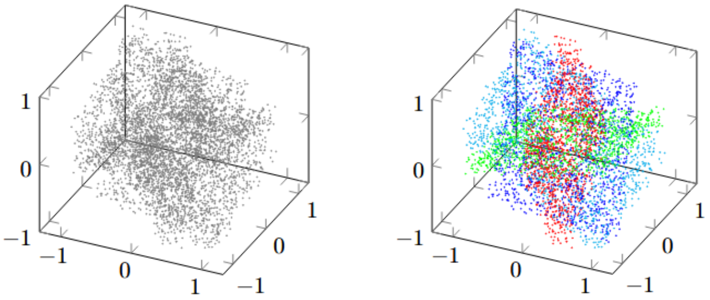


(a) Samples S (b) Optimal labeling y^*

Problem Statement (2)

Subspace Instances of the Cubic Clique Partition Problem

Samples S: points $S \subset \mathbb{R}^3$
Point generation: 3 distinct planes containing the origin, noise σ
Optimal labeling y^*: original planes

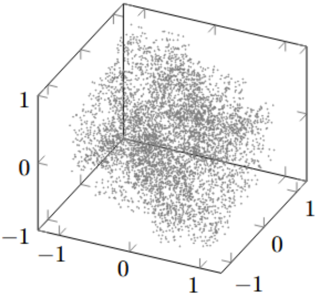


(a) Samples S (b) Optimal labeling y^*

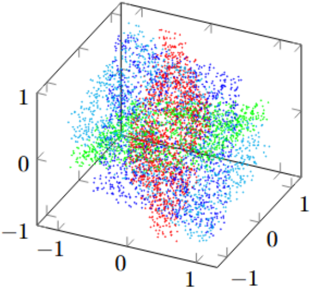
Problem Statement (2)

Subspace Instances of the Cubic Clique Partition Problem

Samples S: points $S \subset \mathbb{R}^3$
Point generation: 3 distinct planes containing the origin, noise σ
Optimal labeling y^*: original planes
Cost function c? (no concrete plane information given)



(a) Samples S



(b) Optimal labeling y^*

Research Goals and Contributions

Task	Solution
read ‘Partial Optimality in Cubic Correlation Clustering’ [?]; implement the algorithm for establishing partial optimality to the cubic clique partition problem;	

Research Goals and Contributions

Task	Solution
read 'Partial Optimality in Cubic Correlation Clustering' [?]; implement the algorithm for establishing partial optimality to the cubic clique partition problem;	my own implementation in C++ of the suggested algorithm with some adjustments

Research Goals and Contributions

Task	**Solution**
read 'Partial Optimality in Cubic Correlation Clustering' [**?**]; implement the algorithm for establishing partial optimality to the cubic clique partition problem;	my own implementation in C++ of the suggested algorithm with some adjustments
construct subspace instances of increasing difficulty by generating the sample points and by defining a **suitable cost function** for the point triples	

Research Goals and Contributions

Task	**Solution**
read ‘Partial Optimality in Cubic Correlation Clustering’ [**?**]; implement the algorithm for establishing partial optimality to the cubic clique partition problem;	my own implementation in C++ of the suggested algorithm with some adjustments
construct subspace instances of increasing difficulty by generating the sample points and by defining a **suitable cost function** for the point triples	point generation using the linear algebra methods; experimentally determined geometric cost function with a significant noise tolerance;

Research Goals and Contributions

Task	Solution
read 'Partial Optimality in Cubic Correlation Clustering' [**?**]; implement the algorithm for establishing partial optimality to the cubic clique partition problem;	my own implementation in C++ of the suggested algorithm with some adjustments
construct subspace instances of increasing difficulty by generating the sample points and by defining a **suitable cost function** for the point triples	point generation using the linear algebra methods; experimentally determined geometric cost function with a significant noise tolerance;
apply the algorithm to the constructed subspace instances, assess partial optimality, accuracy with respect to truth, and computation time	

Research Goals and Contributions

Task	Solution
read 'Partial Optimality in Cubic Correlation Clustering' [**?**]; implement the algorithm for establishing partial optimality to the cubic clique partition problem;	my own implementation in C++ of the suggested algorithm with some adjustments
construct subspace instances of increasing difficulty by generating the sample points and by defining a **suitable cost function** for the point triples	point generation using the linear algebra methods; experimentally determined geometric cost function with a significant noise tolerance;
apply the algorithm to the constructed subspace instances, assess partial optimality, accuracy with respect to truth, and computation time	systematic empirical assessment of the algorithm application to the problem subspace instances with my cost function proving its quality

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

$\rightarrow$ **Partial Optimality Conditions:**

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

$\rightarrow$ **Partial Optimality Conditions:**

1. Subproblem-CUT-condition (cut subset from its complement)

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

$\rightarrow$ **Partial Optimality Conditions:**

1. Subproblem-CUT-condition (cut subset from its complement)
2. CUT-conditions (cut pairs and triples)

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

→ **Partial Optimality Conditions:**

1. Subproblem-CUT-condition (cut subset from its complement)
2. CUT-conditions (cut pairs and triples)
3. JOIN-conditions (join subsets, pairs and triples)

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c \colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y \colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}} \, y_{\{a,b\}} \, y_{\{b,c\}} \, y_{\{a,c\}} + c_{\{a,b\}} \, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

→ **Partial Optimality Conditions:**

1. Subproblem-CUT-condition (cut subset from its complement)
2. CUT-conditions (cut pairs and triples)
3. JOIN-conditions (join subsets, pairs and triples)

CUT-conditions can be applied simultaneously
JOIN-conditions cannot be applied simultaneously!

Partial Optimality for Cubic Clique Partition Problem

Extended cost function $c\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

$$\min_{y\colon \binom{S}{2} \to \{0,1\}} \sum_{\{a,b,c\} \in \binom{S}{3}} c_{\{a,b,c\}}\, y_{\{a,b\}}\, y_{\{b,c\}}\, y_{\{a,c\}} + c_{\{a,b\}}\, y_{\{a,b\}} + c_{\emptyset}$$

Construct **Improving Maps** for the clustering y

→ **Partial Optimality Conditions:**

1. Subproblem-CUT-condition (cut subset from its complement)
2. CUT-conditions (cut pairs and triples)
3. JOIN-conditions (join subsets, pairs and triples)

CUT-conditions can be applied simultaneously
JOIN-conditions cannot be applied simultaneously!

Apply partial optimality connditions → solve subproblems

Partial Optimality Algorithm

Partial Optimality Algorithm:

Input: clustering y without fixed labels

while condition applied **do**

  apply subproblem-CUT-condition exhaustively

  apply one of JOIN-conditions (in effective order)

end while

apply CUT-conditions exhaustively

Output: partially optimal clustering y with some fixed labels

Partial Optimality Algorithm

Partial Optimality Algorithm:

Input: clustering y without fixed labels

while condition applied **do**

  apply subproblem-CUT-condition exhaustively

  apply one of JOIN-conditions (in effective order)

end while

apply CUT-conditions exhaustively

Output: partially optimal clustering y with some fixed labels

Reduction to subproblems:

1. Subproblem-CUT-condition: fix CUT labels for element pairs from different sample subsets; solve each subset as an independent problem and accumulate the results in $c_\emptyset$;

Partial Optimality Algorithm

Partial Optimality Algorithm:

Input: clustering y without fixed labels

while condition applied **do**

  apply subproblem-CUT-condition exhaustively

  apply one of JOIN-conditions (in effective order)

end while

apply CUT-conditions exhaustively

Output: partially optimal clustering y with some fixed labels

Reduction to subproblems:

1. Subproblem-CUT-condition: fix CUT labels for element pairs from different sample subsets; solve each subset as an independent problem and accumulate the results in $c_\emptyset$;
2. JOIN-Conditions: fix JOIN labels for elements of the sample subset; add the join-cost to $c_\emptyset$; solve the problem where the subset is considered as one sample;

Subproblem-CUT and Subset-JOIN

Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

Subproblem-CUT and Subset-JOIN

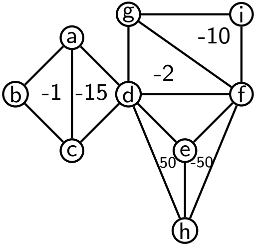
Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

Subset-JOIN: join sample subset R with only non-positive costs if its worst bipartition joining cost is less than or equal to the reward of joining R with $\overline{R}$ (applied if $|R| > 1$)
(The worst bipartition joining cost $\approx$ min-cut)

Subproblem-CUT and Subset-JOIN

Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

Subset-JOIN: join sample subset R with only non-positive costs if its worst bipartition joining cost is less than or equal to the reward of joining R with $\overline{R}$ (applied if $|R| > 1$)
(The worst bipartition joining cost $\approx$ min-cut)

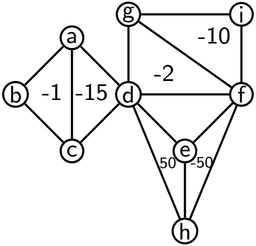


$c_\emptyset = 0$

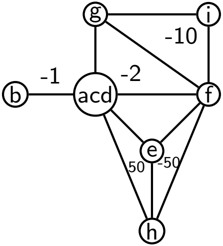
Subproblem-CUT and Subset-JOIN

Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

Subset-JOIN: join sample subset R with only non-positive costs if its worst bipartition joining cost is less than or equal to the reward of joining R with $\overline{R}$ (applied if $|R| > 1$)
(The worst bipartition joining cost $\approx$ min-cut)



$c_{\emptyset} = 0$

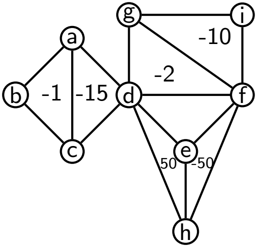


$c_{\emptyset} = -15$

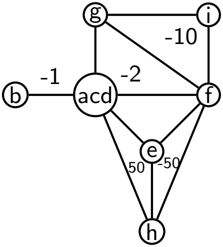
Subproblem-CUT and Subset-JOIN

Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

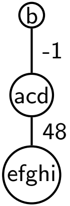
Subset-JOIN: join sample subset R with only non-positive costs if its worst bipartition joining cost is less than or equal to the reward of joining R with $\overline{R}$ (applied if $|R| > 1$)
(The worst bipartition joining cost $\approx$ min-cut)



$c_\emptyset = 0$



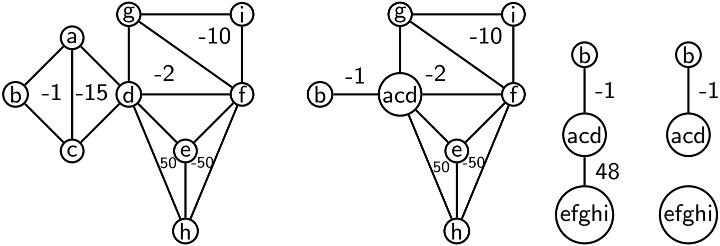
$c_\emptyset = -15$



$c_\emptyset = -75$

Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

Subset-JOIN: join sample subset R with only non-positive costs if its worst bipartition joining cost is less than or equal to the reward of joining R with $\overline{R}$ (applied if $|R| > 1$)
(The worst bipartition joining cost $\approx$ min-cut)

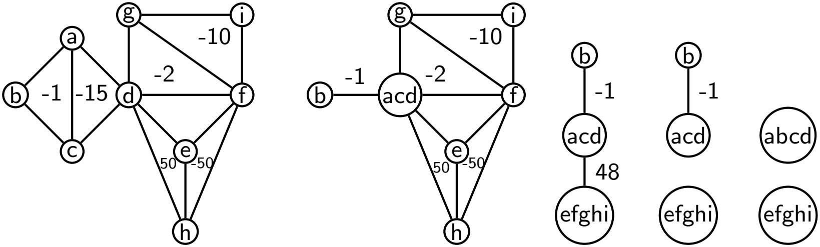


$c_\emptyset = 0$ $c_\emptyset = -15$ $c_\emptyset = -75$

Subproblem-CUT and Subset-JOIN

Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

Subset-JOIN: join sample subset R with only non-positive costs if its worst bipartition joining cost is less than or equal to the reward of joining R with $\overline{R}$ (applied if $|R| > 1$)
(The worst bipartition joining cost $\approx$ min-cut)



$c_\emptyset = 0$ $c_\emptyset = -15$ $c_\emptyset = -75$ $c_\emptyset = -76$

Other JOIN-conditions

Pair-JOIN-1: join samples i, j if their overall joining reward $\geq$ the sum of rewards and penalties for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Other JOIN-conditions

Pair-JOIN-1: join samples i, j if their overall joining reward $\geq$ the sum of rewards and penalties for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Pair-JOIN-2: join samples i, k if there exists a sample triple ijk that fulfills 3 conditions (2 conditions $\approx$ min-cut, 1 explicit condition)

Other JOIN-conditions

Pair-JOIN-1: join samples i, j if their overall joining reward $\geq$ the sum of rewards and penalties for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Pair-JOIN-2: join samples i, k if there exists a sample triple ijk that fulfills 3 conditions (2 conditions $\approx$ min-cut, 1 explicit condition)

Pair-JOIN-3: join samples i, j if $\mathrm{c}_{\{i,j\}} \leq$ the sum of reward costs for joining pairs and triples containing i or j

Other JOIN-conditions

Pair-JOIN-1: join samples i, j if their overall joining reward $\geq$ the sum of rewards and penalties for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Pair-JOIN-2: join samples i, k if there exists a sample triple ijk that fulfills 3 conditions (2 conditions $\approx$ min-cut, 1 explicit condition)

Pair-JOIN-3: join samples i, j if $\mathrm{c}_{\{i,j\}} \leq$ the sum of reward costs for joining pairs and triples containing i or j

Pair-JOIN-4: join samples i, k if there exists a sample triple ijk such that 7 explicit conditions hold

Other JOIN-conditions

Pair-JOIN-1: join samples i, j if their overall joining reward $\geq$ the sum of rewards and penalties for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Pair-JOIN-2: join samples i, k if there exists a sample triple ijk that fulfills 3 conditions (2 conditions $\approx$ min-cut, 1 explicit condition)

Pair-JOIN-3: join samples i, j if $c_{\{i,j\}} \leq$ the sum of reward costs for joining pairs and triples containing i or j

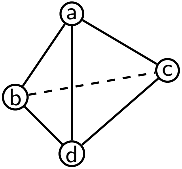
Pair-JOIN-4: join samples i, k if there exists a sample triple ijk such that 7 explicit conditions hold

Triple-JOIN: join samples i, j, k if the condition holds (similar to Pair-JOIN-1) (rhs $\approx$ min-cut)

Pyramid Instance and CUT-conditions

$\mathsf{c}_{\{b,c,d\}} = 10$

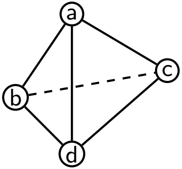
$\mathsf{c}_{\{a,b,c\}} = \mathsf{c}_{\{a,b,d\}} = \mathsf{c}_{\{a,c,d\}} = -50$

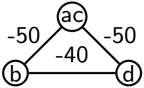


Pyramid Instance and CUT-conditions

$c_{\{b,c,d\}} = 10$

$c_{\{a,b,c\}} = c_{\{a,b,d\}} = c_{\{a,c,d\}} = -50$



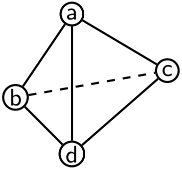


Pair-JOIN-2

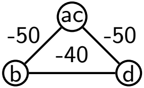
Pyramid Instance and CUT-conditions

$c_{\{b,c,d\}} = 10$

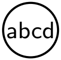
$c_{\{a,b,c\}} = c_{\{a,b,d\}} = c_{\{a,c,d\}} = -50$



Pair-JOIN-2



Subset-JOIN

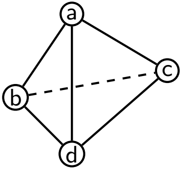


$c_{\emptyset} = -140$

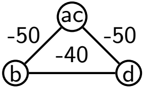
Pyramid Instance and CUT-conditions

$c_{\{b,c,d\}} = 10$

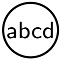
$c_{\{a,b,c\}} = c_{\{a,b,d\}} = c_{\{a,c,d\}} = -50$



Pair-JOIN-2



Subset-JOIN



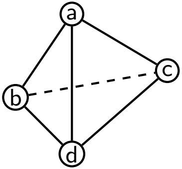
$c_{\emptyset} = -140$

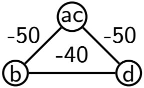
Pair-CUT: cut samples i, j if the direct joing penalty $\geq$ the sum of rewards for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

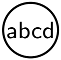
Pyramid Instance and CUT-conditions

$c_{\{b,c,d\}} = 10$

$c_{\{a,b,c\}} = c_{\{a,b,d\}} = c_{\{a,c,d\}} = -50$







Pair-JOIN-2 Subset-JOIN $c_{\emptyset} = -140$

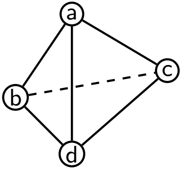
Pair-CUT: cut samples i, j if the direct joing penalty $\geq$ the sum of rewards for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Triple-CUT: cutsamples i, j, k if the condition holds (similar to Pair-CUT) (rhs $\approx$ min-cut)

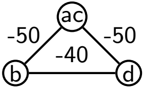
Pyramid Instance and CUT-conditions

$c_{\{b,c,d\}} = 10$

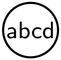
$c_{\{a,b,c\}} = c_{\{a,b,d\}} = c_{\{a,c,d\}} = -50$



Pair-JOIN-2



Subset-JOIN



$c_\emptyset = -140$

Pair-CUT: cut samples i, j if the direct joing penalty $\geq$ the sum of rewards for joining some subset R with $i \in R$ and $\overline{R}$ with $j \in \overline{R}$ (rhs $\approx$ min-cut)

Triple-CUT: cutsamples i, j, k if the condition holds (similar to Pair-CUT) (rhs $\approx$ min-cut)

Samples in the pyramid with $c_{\{b,c,d\}} = 100$ are unjoinable!
Triple-CUT is applied to the triple *bcd*

Program Structure

TODO: class Diagram TODO: features Algorithm implementation in ClusteringProblem Features: ClusteringProblem is generally defined for all types of Cubic Clique Partition Problem (not necessarily points), cost function + sparse costs!, label computation, cut triples, logs joins and cuts!

Pyramid Instance: Program Logs

Clustering problem has been inited with 4 relevant triples,
(negative: 3, positive: 1)
Trying independent subproblem cut (3.1)...
Trying subset join (3.11)...
Trying pair join (3.4)...
Trying complex pair join (3.6)...
* Applying the complex pair join (3.6)
Join: a c

Trying independent subproblem cut (3.1)...
Trying subset join (3.11)...
* Applying the subset join (3.11)
Join: ac b d

(0: cut; 1: joint; x: unknown)
Labeling:
a b c d
a - 1 1 1
b 1 - 1 1
c 1 1 - 1
d 1 1 1 -

Unjoinable cluster triples:
Clustering:

Cluster 0: abcd
Problem solved: completely

Cost: -140

Plane and Point Generation

plane Generation point Generation save as the labeled data, use the unlabeled data + cost function to init the ClusteringProblem

Cost Function

My contribution!!! Cost function does not depend on the point distribution bounds!!! Steps to assign the cost !!!!!

Experiments

My laptop characteristics, Random cubic subspace instances with different seeds max point component size 100 (noise are the percents then) no noise 0, small noise 1, significant noise 3, large noise 5, (Table: instance size + noise + instance count)

Cost Function Evaluation

blue and red dots, conflicts and and their effect (picture of the typical cost function evaluation)

Experiment Results for 3x7 Points

3x7 points: all results + time-optimality-accuracy (min-max-average)

Experiment Results

3x(7x12x17x22) time-optimality-accuracy (min-max-average) DIAGRAM!!! mention the coefficients to prove the efficiency, partial optimality and accuracy!!!

Research Results

Partial optimality reimplementation, Dedicated Cost Function, Efficiency and Accuracy (shown by the experiments)

Future Work

Optimize the partial optimality algorithm Determine better parameters for the cost function Extend the cost function with more advanced conditions