Partially Optimal Cubic Subspace Clustering

Research Project Machine Learning

Volodymyr Drobitko

Technische Universität Dresden

21.07.2025

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 $\mathrm{c}\colon \binom{S}{3} \cup \binom{S}{2} \cup \emptyset \to \mathbb{R}$

Instance of the extended cubic clique partition problem:

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

Construct **Improving Maps** for the clustering y

$\to$ **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

→ **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

$\to$ **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

$\rightarrow$ **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 $\rightarrow$ 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:

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 JOIN-Subset

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 JOIN-Subset

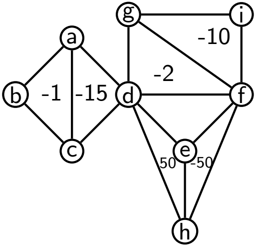
Subproblem-CUT: cut sample subsets $R_1, R_2, \ldots, R_k$ that are only connected via non-negative costs (applied if $k > 1$)

JOIN-Subset: 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 $|R| > 1$)
(Finding the worst bipartition joining cost can be reduced to the Min-Cut-Problem and solved as its dual Max-Flow-Problem)

Subproblem-CUT and JOIN-Subset

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

JOIN-Subset: 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 $|R| > 1$)
(Finding the worst bipartition joining cost can be reduced to the Min-Cut-Problem and solved as its dual Max-Flow-Problem)

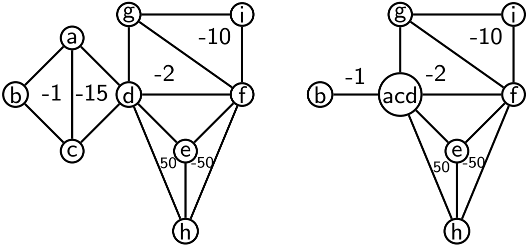


$c_{\emptyset} = 0$

Subproblem-CUT and JOIN-Subset

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

JOIN-Subset: 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 $|R| > 1$)
(Finding the worst bipartition joining cost can be reduced to the Min-Cut-Problem and solved as its dual Max-Flow-Problem)

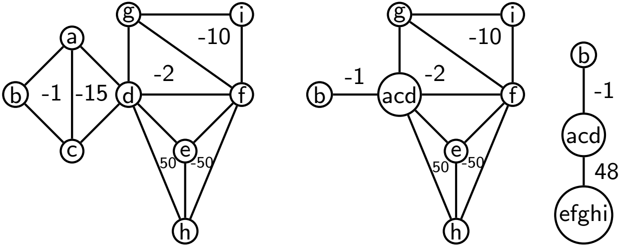


$c_\emptyset = 0$ $\quad$ $c_\emptyset = -15$

Subproblem-CUT and JOIN-Subset

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

JOIN-Subset: 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 $|R| > 1$)
(Finding the worst bipartition joining cost can be reduced to the Min-Cut-Problem and solved as its dual Max-Flow-Problem)

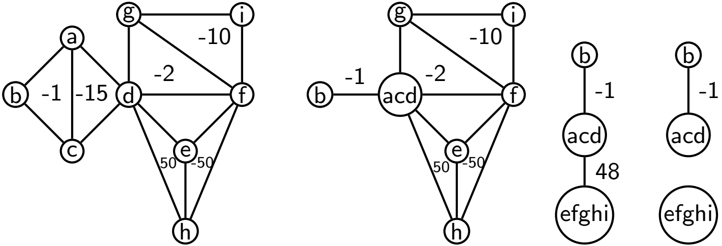


$c_\emptyset = 0$ $c_\emptyset = -15$ $c_\emptyset = -75$

Subproblem-CUT and JOIN-Subset

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

JOIN-Subset: 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 $|R| > 1$)
(Finding the worst bipartition joining cost can be reduced to the Min-Cut-Problem and solved as its dual Max-Flow-Problem)

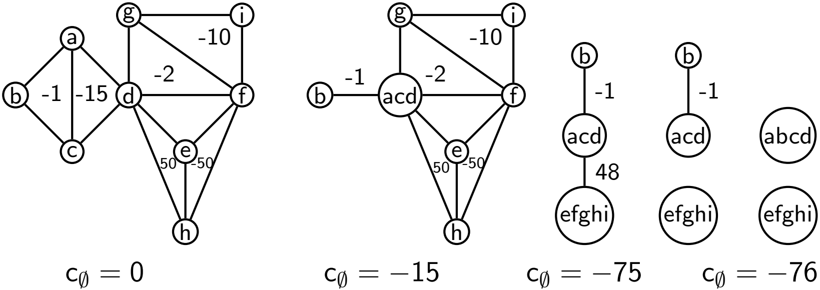


$c_\emptyset = 0$ $c_\emptyset = -15$ $c_\emptyset = -75$

Subproblem-CUT and JOIN-Subset

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

JOIN-Subset: 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 $|R| > 1$)
(Finding the worst bipartition joining cost can be reduced to the Min-Cut-Problem and solved as its dual Max-Flow-Problem)



Other JOIN-conditions and CUT-conditions

Overview of the other join-conditions (with pictures) Overview of the cut-conditions (with pictures) Pyramid example (solvable and unsolvable)

Program Structure

TODO
Class Diagram 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! (add screenshots)