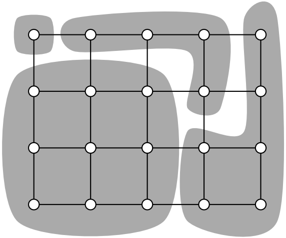
Machine Learning II

Bjoern Andres

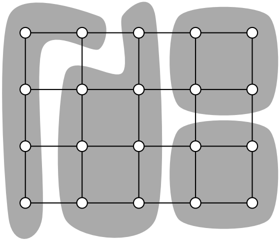
Machine Learning for Computer Vision
TU Dresden

https://mlcv.cs.tu-dresden.de/courses/26-summer/ml2/

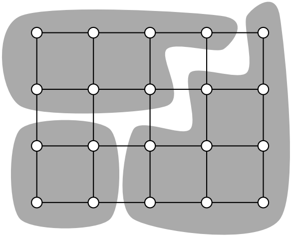
Summer Term 2026



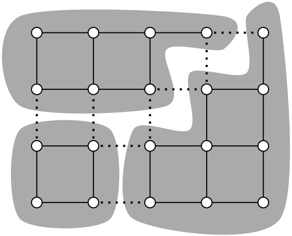
Clustering of a graph $G = (V, E)$



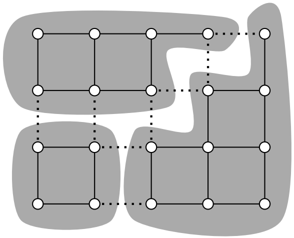
Clustering of a graph $G = (V, E)$



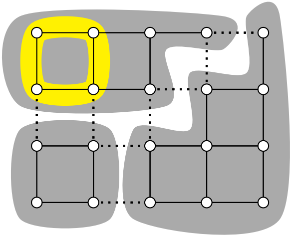
Clustering of a graph $G = (V, E)$



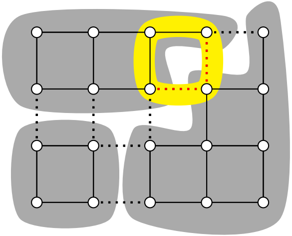
Clustering of a graph $G = (V, E)$



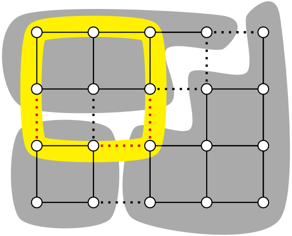
Multicut of a graph $G = (V, E)$



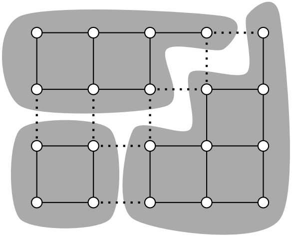
Multicut of a graph $G = (V, E)$



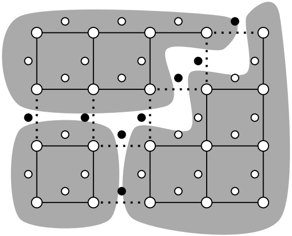
Multicut of a graph $G = (V, E)$



Multicut of a graph $G = (V, E)$



Multicut of a graph $G = (V, E)$



Multicut of a graph $G = (V, E)$

Definition 1. Let $G = (V, E)$ be any graph.

– A subgraph $G' = (U, E')$ of G is called a **cluster** of G iff G' is non-empty, node-induced (i.e. $E' = E \cap \binom{U}{2}$) and connected.

– A partition Π of the node set V is called a **clustering** of G iff for every $U \in \Pi$ the subgraph $(U, E \cap \binom{U}{2})$ of G induced by U is connected (and thus a component of G).

– Let C_G denote the set of all clusterings of G.

Definition 2. Let $G = (V, E)$ be any graph.

- For any $M \subseteq E$, M is called a **multicut** of G iff for every cycle (V_C, E_C) of G: $|E_C \cap M| \neq 1$.
- Let M_G denote the set of all multicuts of G.

Lemma 1. Let $G = (V, E)$ be any graph.

- For any clustering Π of G, the set $\{ab \in E \mid \forall U \in \Pi\colon a \notin U \vee b \notin U\}$ $=: M_\Pi$ of those edges that straddle distinct clusters is a multicut of G.
- For any clustering Π of G, the multicut M_Π is said to be **induced** by Π.
- The map $\Pi \mapsto M_\Pi$ from clusterings to induced multicuts is a **bijection** from C_G to M_G.

Remark 1. The characteristic function $y \in \{0,1\}^E$ of a multicut $y^{-1}(1)$ makes explicit for every edge $ab = e \in E$ whether the incident nodes a and b belong to the same clusters, $y_e = 0$, or distinct clusters, $y_e = 1$.

Lemma 2. For any $y \in \{0,1\}^E$, the set $y^{-1}(1)$ is a multicut of G iff the following inequalities are satisfied:

$$\forall (V_C, E_C) \in \mathrm{cycles}(G)\ \forall e \in E_C\colon \quad y_e \leq \sum_{e' \in E_C \setminus \{e\}} y_{e'} \tag{1}$$

Learning and inferring clusterings:

- Instead of the problem of learning and inferring clusterings, we consider the problem of learning and inferring multicuts. By Lemma 1, this is w.l.o.g..
- We reduce the problem of learning and inferring multicuts to the problem of learning and inferring decisions, by defining constrained data (S, X, x, Y) with

$$S = E \tag{2}$$

$$\mathcal{Y} = \left\{ y \in \{0,1\}^E \;\middle|\; \forall (V_C, E_C) \in \text{cycles}(G)\ \forall e \in E_C \colon y_e \leq \sum_{e' \in E_C \setminus \{e\}} y_{e'} \right\} . \tag{3}$$

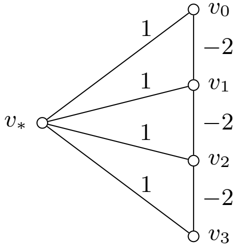
- As inference problem, we obtain the **(minimum cost) multicut problem**

$$\min_{y \in \{0,1\}^E} \quad \sum_{e \in E} \underbrace{(-f_\theta(x_e))}_{=: c_e} y_e \tag{4}$$

$$\text{subject to} \quad \forall (V_C, E_C) \in \text{cycles}(G)\ \forall e \in E_C \colon\ y_e \leq \sum_{e' \in E_C \setminus \{e\}} y_{e'} \ . \tag{5}$$

Theorem 1. The minimum cost multicut problem is NP-hard.

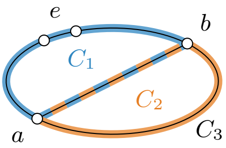
Proof. (Sketch) By reduction of **maximum independent set**:



□

Theorem 2 (Chopra and Rao 1993)**.** In the minimum cost multicut problem, the inequalities (1) are redundant for chordal cycles.

Proof. (Sketch) Consider a cycle C_3 with a chord ab.



By induction, (1) holds for C_1 and C_2. For any $e \in C_1 \setminus \{ab\}$:

$$\begin{aligned} y_e &\leq \sum_{e' \in C_1 \setminus \{e\}} y_{e'} = y_{ab} + \sum_{e' \in C_1 \setminus \{e, ab\}} y_{e'} \\ &\leq \sum_{e' \in C_2 \setminus \{ab\}} y_{e'} + \sum_{e' \in C_1 \setminus \{e, ab\}} y_{e'} = \sum_{e' \in C_3 \setminus \{e\}} y_{e'} \ . \end{aligned}$$

For any $e \in C_2 \setminus \{ab\}$, the argument is analogous. $\square$

Corollary 1 (Chopra and Rao 1993)**.** The multicut problem for a complete graph is isomorphic to the clique partition problem for the node set.

Proof. (Sketch) In a complete graph $G = (V, E)$, the chordless cycles are precisely the triangles.

The inequalities (1) for all triangles are written equivalenly as

$$\forall a \in V \ \forall b \in V \setminus \{a\} \ \forall c \in V \setminus \{a, b\} \colon \quad y_{ac} \leq y_{ab} + y_{bc} \ .$$

Substituting $1 - y'$ for y (i.e. swapping the roles of 0 and 1), yields the transitivity constraints of the clique partition problem:

$$\forall a \in V \ \forall b \in V \setminus \{a\} \ \forall c \in V \setminus \{a, b\} \colon \quad y'_{ab} + y'_{bc} - 1 \leq y'_{ac} \ .$$

$\square$

Definition 3. For any graph $G = (V, E)$ and any $c \in \mathbb{R}^E$, the **cycle relaxation** of the minimum cost multicut problem is

$$\min_{y \in \mathbb{R}^E} \quad \langle c, y \rangle \tag{6}$$

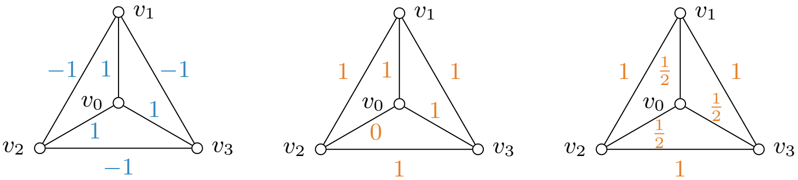
$$\text{subject to} \quad \forall (V_C, E_C) \in \text{cycles}(G)\ \forall e \in E_C\colon\ y_e \leq \sum_{e' \in E_C \setminus \{e\}} y_{e'} \tag{7}$$

$$\forall e \in E\colon \quad 0 \leq y_e \leq 1\ . \tag{8}$$

Remark 2. The cycle relaxation of the minimum cost multicut problem

– can be solved by the simplex algorithm
– yields a lower bound on the minimum cost
– can be solved efficiently
– is not tight (see below)

Example 1. Instance of the multicut problem for an odd wheel:



Instance

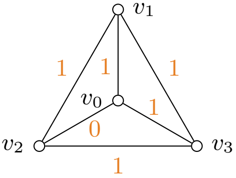
Solution
$\langle c, y\rangle = -1$

Solution to relaxation
$\langle c, y\rangle = -\frac{3}{2}$

Remark 3. Approaches to working with LP relaxations that are not tight:

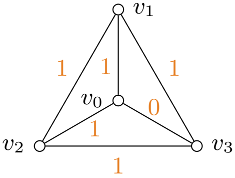
- branching
- cutting planes

Example 2. Branching on $y_{v_0 v_2}$



Solution to relaxation with $y_{v_0 v_2} = 0$

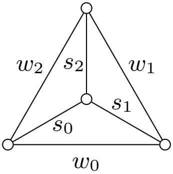
$\langle c, y \rangle = -1$



Solution to relaxation with $y_{v_0 v_2} = 1$

$\langle c, y \rangle = -1$

Example 3. An **odd wheel inequality** as a **Chvátal-Gomory cutting plane**:



From the cycle relaxation of the multicut problem, the inequalities

$$\begin{aligned} y_{w_0} &\le y_{s_0} + y_{s_1} & \qquad y_{w_0} &\le 1 \\ y_{w_1} &\le y_{s_1} + y_{s_2} & \qquad y_{w_1} &\le 1 \\ y_{w_2} &\le y_{s_2} + y_{s_0} & \qquad y_{w_2} &\le 1 \end{aligned}$$

together imply

$$\sum_j y_{w_j} - \sum_j y_{s_j} \le \frac{3}{2}$$

For any integral feasible solution y, the lhs. is an integer. Thus:

$$\sum_j y_{w_j} - \sum_j y_{s_j} \le \left\lfloor \frac{3}{2} \right\rfloor \; .$$

Next, we discuss partial optimality for clustering.

For ease of notation, we consider complete graphs, i.e., the clique partition problem.

Definition 4. For any $A \neq \emptyset$ finite, any $c \colon \binom{A}{2} \to \mathbb{R}$,

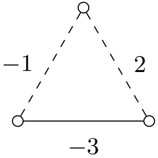
$$Y_A := \Big\{ y \colon \tbinom{A}{2} \to \{0,1\} \;\Big|\; \forall a \in A \; \forall b \in A \setminus \{a\} \; \forall c \in A \setminus \{a,b\} \colon \\ y_{ab} + y_{bc} - 1 \leq y_{ac} \Big\} \tag{9}$$

and $\varphi_c \colon Y_A \to \mathbb{R} \colon y \mapsto \langle c, y \rangle$,

$$\min\{\varphi_c(y) \mid y \in Y_A\} \tag{10}$$

is called the instance of the **(clique) partition problem** wrt. A and c, which we abbreviate as $\mathrm{CPP}(A, c)$.

Example 4.



For any set A and any $U \subseteq A$, we write

$$\partial U := \left\{ \{u, a\} \in \binom{A}{2} \;\middle|\; u \in U \wedge a \notin U \right\} \quad . \tag{11}$$

Definition 5. Let $A \neq \emptyset$ finite and $U \subseteq A$.

- The **elementary cut map** wrt. U is the $\sigma_U \colon Y_A \to Y_A$ such that $\forall y \in Y_A \ \forall \{a, b\} \in \binom{A}{2}$:

$$\sigma_U(y)_{ab} = \begin{cases} 0 & \text{if } \{a,b\} \in \partial U \\ y_{ab} & \text{otherwise} \end{cases} \quad . \tag{12}$$

- The **elementary join map** wrt. U is the $\sigma'_U \colon Y_A \to Y_A$ such that $\forall y \in Y_A \ \forall \{a, b\} \in \binom{A}{2}$:

$$\sigma'_U(y)_{ab} = \begin{cases} 1 & \text{if } \{a,b\} \in \binom{U}{2} \\ 1 & \text{if } a \in U \wedge \exists u \in U \colon y_{ub} = 1 \\ 1 & \text{if } b \in U \wedge \exists u \in U \colon y_{ua} = 1 \\ 1 & \text{if } (\exists u \in U \colon y_{ua} = 1) \wedge \\ & \quad (\exists u \in U \colon y_{ub} = 1) \\ y_{ab} & \text{otherwise} \end{cases} \quad . \tag{13}$$

Remark 4. σ_U is well-defined, i.e. $\sigma_U(Y_A) \subseteq Y_A$. σ'_U is well-defined. $\sigma'_U \circ \sigma_U$ is well-defined.

To begin with, we establish a trivial partial optimality condition for the clique partition problem:

Lemma 3. Let $A \neq \emptyset$ finite and $c \colon \binom{A}{2} \to \mathbb{R}$. If there exists $U \subseteq A$ such that

$$\forall \{a, b\} \in \partial U : \quad 0 \leq c_{ab} \ , \tag{14}$$

there exists a solution y^* to $\mathrm{CPP}(A, c)$ such that

$$\forall \{a, b\} \in \partial U : \quad y^*_{ab} = 0 \ . \tag{15}$$

Proof. For any $y \in Y_A$, $\sigma_U(y)$ satisfies (15). Moreover, σ_U is improving for $\mathrm{CPP}(A, c)$ because for any $y \in Y_A$ and $y' := \sigma_U(y)$:

$$\begin{aligned}
\varphi_c(y') - \varphi_c(y) &= \sum_{\{a,b\} \in \binom{A}{2}} c_{ab}\, y'_{ab} - \sum_{\{a,b\} \in \binom{A}{2}} c_{ab}\, y_{ab} && (16) \\
&= \sum_{\{a,b\} \in \binom{A}{2}} c_{ab} (y'_{ab} - y_{ab}) && (17) \\
&= \sum_{\{a,b\} \in \partial U} c_{ab} (0 - y_{ab}) && (18) \\
&= - \sum_{\{a,b\} \in \partial U} c_{ab}\, y_{ab} && (19) \\
&\overset{(14)}{\leq} 0 \; . && (20)
\end{aligned}$$

$\square$

For any $r \in \mathbb{R}$, we write

$$[r]_+ := \begin{cases} r & \text{if } r \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

$$[r]_- := \begin{cases} 0 & \text{if } r \geq 0 \\ -r & \text{otherwise} \end{cases} . \tag{22}$$

Next, we establish a less trivial partial optimality condition for the CPP:

Proposition 1. Let $A \neq \varnothing$ finite and $c \colon \binom{A}{2} \to \mathbb{R}$. If there exist $U \subseteq A$ and $\{u, v\} \in \partial U$ such that

$$\sum_{\{a,b\} \in \partial U \setminus \{\{u,v\}\}} [c_{ab}]_- \leq c_{uv} \quad , \tag{23}$$

there exists a solution y^* to $\mathrm{CPP}(A, c)$ such that $y^*_{uv} = 0$.

Proof. Let $\xi\colon Y_A \to Y_A$ such that for all $y \in Y_A$:

$$\xi(y) = \begin{cases} y & \text{if } y_{uv} = 0 \\ \sigma_U(y) & \text{otherwise} \end{cases} \quad . \tag{24}$$

For any $y \in Y_A$ and $y' := \xi(y)$, we have $y'_{uv} = 0$.

Moreover, ξ is improving for $\mathrm{CPP}(A, c)$ because for all $y \in Y_A$ and $y' := \xi(y)$, the following holds: If $y_{ab} = 0$ then $\varphi_c(y') - \varphi_c(y) = \varphi_c(y) - \varphi_c(y) = 0 \leq 0$. Otherwise:

$$\begin{aligned}
\varphi_c(y') - \varphi_c(y) &= \sum_{\{a,b\} \in \binom{A}{2}} c_{ab}(y'_{ab} - y_{ab}) && (25)\\
&= c_{uv}(0-1) + \sum_{\{a,b\} \in \partial U \setminus \{\{u,v\}\}} c_{ab}(0 - y_{ab}) && (26)\\
&= -c_{uv} - \sum_{\{a,b\} \in \partial U \setminus \{\{u,v\}\}} c_{ab}\, y_{ab} && (27)\\
&\leq -c_{uv} + \sum_{\{a,b\} \in \partial U \setminus \{\{u,v\}\}} [c_{ab}]_- && (28)\\
&\overset{(23)}{\leq} 0 \ . && (29)
\end{aligned}$$

$\square$

Next, we establish a non-trivial partial optimality condition for the CPP:

Lemma 4. Let $A \neq \varnothing$ finite and $c\colon \binom{A}{2} \to \mathbb{R}$. If there exist $U \subseteq A$ such that

$$\sum_{\{u,a\} \in \partial U} [c_{ua}]_- \leq \min_{\{s,t\} \in \binom{U}{2}} \min_{\substack{y \in Y_U \mid \\ y_{st}=0}} \sum_{\{u,v\} \in \binom{U}{2}} (-c_{uv})(1 - y_{uv}) \ , \qquad (30)$$

there exists a solution y^* to $\mathrm{CPP}(A, c)$ such that $\forall \{u,v\} \in \binom{U}{2} \colon y^*_{uv} = 1$.

Proof. Let $\xi\colon Y_A \to Y_A$ such that for all $y \in Y_A$:

$$\xi(y) := \begin{cases} (\sigma'_U \circ \sigma_U)(y) & \text{if } \exists \{u,v\} \in \binom{U}{2} \colon y_{uv} = 0 \\ y & \text{otherwise} \end{cases} \quad . \tag{31}$$

For any $y \in Y_A$, $y' := \xi(y)$ and all $\{u,v\} \in \binom{U}{2}$, we have $y'_{uv} = 1$.

Moreover, ξ is improving because for all $y \in Y_A$ and $y' := \xi(y)$, the following condition holds: If $\forall \{u,v\} \in \binom{U}{2} \colon y_{uv} = 1$ then $\varphi_c(y') - \varphi_c(y) = \varphi_c(y) - \varphi_c(y) = 0 \leq 0$. Otherwise:

$$\varphi_c(y') - \varphi_c(y) = \sum_{\{u,a\} \in \partial U} c_{ua}(0 - y_{ua}) + \sum_{\{u,v\} \in \binom{U}{2}} c_{uv}(1 - y_{uv}) \tag{32}$$

$$\leq \sum_{\{u,a\} \in \partial U} [c_{ua}]_- + \max_{\{s,t\} \in \binom{U}{2}} \max_{\substack{y \in Y_U | \\ y_{st}=0}} \sum_{\{u,v\} \in \binom{U}{2}} c_{uv}(1 - y_{uv}) \tag{33}$$

$$\leq \sum_{\{u,a\} \in \partial U} [c_{ua}]_- - \min_{\{s,t\} \in \binom{U}{2}} \min_{\substack{y \in Y_U | \\ y_{st}=0}} \sum_{\{u,v\} \in \binom{U}{2}} (-c_{uv})(1 - y_{uv}) \tag{34}$$

$$\overset{(30)}{\leq} 0 \ . \tag{35}$$

$\square$

Even if set $U \subseteq A$ is given, Condition (30) of Lemma 4 cannot be checked efficiently: In general, the calculation of

$$\min_{\{s,t\} \in \binom{U}{2}} \min_{\substack{y \in Y_U | \\ y_{st} = 0}} \sum_{\{u,v\} \in \binom{U}{2}} (-c_{uv})(1 - y_{uv}) \tag{36}$$

requires solving CPPs with the additional constraint $y_{st} = 0$.

However, in the special case where $\forall \{u,v\} \in \binom{U}{2}$: $c_{uv} \leq 0$, these problems become minimum st-cut problems that can be solved efficiently.

Hence, one way of applying Lemma 4 algorithmically is in two steps:

1. heuristically search for a set U such that
   - inside U, all costs are non-positive
   - on the boundary of U, the sum of the negative costs is large.
2. efficiently test (30) for this U