HAL open science

Suppression Distance Computation for Hierarchical Clusterings

François Queyroi, Sergey Kirgizov

▶ To cite this version:

François Queyroi, Sergey Kirgizov. Suppression Distance Computation for Hierarchical Clusterings. Information Processing Letters, 2015, 15 (9), pp.689-693. 10.1016/j.ipl.2015.04.007 . hal-00996090v3

HAL Id: hal-00996090

https://hal.science/hal-00996090v3

Submitted on 21 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Suppression Distance Computation for Hierarchical Clusterings

François Queyroi*, Sergey Kirgizov*

Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005
CNRS, UMR 7606, LIP6, F-75005, Paris, France

Abstract

We discuss the computation of a distance between two hierarchical clusterings of the same set. It is defined as the minimum number of elements that have to be removed so the remaining clusterings are equal. The problem of distance computing was extensively studied for partitions. We prove it can be solved in polynomial time in the case of hierarchies as it gives birth to a class of perfect graphs. We also propose an algorithm based on recursively computing maximum assignments.

Keywords: hierarchical partition, clustering, distance, graphs, vertex cover

1. Introduction

Decomposing a set into patterns of interest is a central problem in data analysis. Evaluating the distance between decompositions is an important task in this context as it allows to study the behaviour of clustering algorithms or study the evolution of a set of patterns over time. The situation where the detected patterns do not overlap is called *partitions*. Measures based on edit distance [3] or on mutual information [6] can be used to assess the distance between those objects. The first corresponds to the minimum number of elements that need to be moved from one group to another for the two partitions to be equal (called *transfer distance* in [7]). It was used for practical applications in bioinformatics [10]. Similar definitions can also applied to different kind of decompositions *e.g.* with overlapping groups (called *set covers*).

This work focuses on *hierarchical clusterings* (also called hierarchies) in which each group can be recursively decomposed into smaller groups. The problem of distance definition between hierarchies is of interest as they can

*Corresponding author
Email addresses: `francois.queyroi@parisgeo.cnrs.fr` (François Queyroi), `sergey.kirgizov@u-bourgogne.fr` (Sergey Kirgizov)

be used to represent and study a system (such as *complex networks* [8]) at different scales. Comparing hierarchical clusterings is related to the comparison of phylogenetic trees [9] in biology although those objects have typically more constraints than the decompositions studied here.

We investigate the problem of finding the minimum number of elements to be removed so that the remaining hierarchical clusterings are equal or, equivalently, the size of of smallest subset of elements for which the decompositions "disagree". After having define the core concepts (Section 2), we will provide two alternative proofs of the main claim (Section 3 and 4). The first links the problem to a class of perfect graphs (generalizing the results of [3]) since the difference between hierarchies can be encoded into a graph (called the *difference graph*) with specific characteristics. The second provides a polynomial algorithm to compute the distance between hierarchical clusterings. Both approaches are based on similar observations (Lemmas 2 and 3). Section 5 provides concluding remarks and directions for future work.

2. Definitions

We assume we have a set S of elements of finite cardinality. A *hierarchy* $\mathcal{H} = (H_1, H_2, \ldots, H_k)$ is a finite multiset of non-empty subsets of S such that if there exist two groups $H_1, H_2 \in \mathcal{H}$ such that if $H_1 \cap H_2 \neq \emptyset$ then either $H_1 \subseteq H_2$ or $H_2 \subseteq H_1$. The relation of inclusion between the sets defines a partial ordered set. It can be represented in a forest fashion, the roots of each tree being the sets that are not include in any other group.

Let $N_i(\mathcal{H})$ denotes the i-th level of $\mathcal{H}$ *i.e.* the groups sitting at depth i in this forest. Notice it is still well defined if $\mathcal{H}$ contains repeated groups. A level $N_i(\mathcal{H})$ is a partition since it does not contain overlapping sets. The *depth* of a hierarchy $d(\mathcal{H})$ is the maximum depth of its groups. We define as $\mathcal{H}[S']$ the *sub-hierarchy induced by* $S' \subseteq S$ as the non-empty sets of $\{S' \cap H_i\}_{1 \leq i \leq k}$. It is the hierarchical clustering of S' obtained after the removal of every elements of $\{S \setminus S'\}$ in each group of $\mathcal{H}$ (discarding empty sets).

Definition 1. ***(Suppression Distance)*** *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchies of S. The suppression distance d_s is defined as*

$$d_s(\mathcal{H}_1, \mathcal{H}_2) = \min_{S' \subseteq S} \{|S'| : \mathcal{H}_1[S \setminus S'] = \mathcal{H}_2[S \setminus S']\}$$

A set S' such that $\mathcal{H}_1[S \setminus S'] = \mathcal{H}_2[S \setminus S']$ is called a suppression set*.*

Theorem 1. *The function d_s is a metric.*

Proof. The non-negativity, identity and symmetry properties are straightforward for d_s. Moreover, this distance respects the triangular inequality. Consider

three hierarchies $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$. Let $S_{ij} \subseteq S$ be a minimum suppression set for $(\mathcal{H}_i, \mathcal{H}_j)$. Since $S_{12} \cup S_{23}$ is also a suppression set for $(\mathcal{H}_1, \mathcal{H}_3)$, we have:

$$|S_{13}| \leq |S_{12} \cup S_{23}| \leq |S_{12}| + |S_{23}|$$
$$d_s(\mathcal{H}_1, \mathcal{H}_3) \leq d_s(\mathcal{H}_1, \mathcal{H}_2) + d_s(\mathcal{H}_2, \mathcal{H}_3)$$

□

3. Existence of a polynomial-time solution

We give here a non-constructive proof for the existence of a polynomial time algorithm. It generalizes the results of Gusfield [3] on the equivalence between this problem and the minimum vertex cover problem on perfect graphs. The difference between hierarchies can be encoded in a *difference graph* (Definition 2). Finding a suppression set for two hierarchies is equivalent to find a minimum vertex cover in this graph (Theorem 2). Since, this graph is perfect [5] (Theorem 3), it exists a polynomial time algorithm to solve this problem.

Definition 2. ***(Difference Graph)*** *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hierarchies of a set S. We call $G(\mathcal{H}_1, \mathcal{H}_2) = (S, E)$ the* difference graph *of $(\mathcal{H}_1, \mathcal{H}_2)$* [1] *with*

$$E = \{(s_1, s_2) \in S^2 : |\mathcal{H}_1[\{s_1, s_2\}]| \neq |\mathcal{H}_2[\{s_1, s_2\}]|\}$$

This graph can contain self-loops.

Two elements of S are connected iff they do not appear in the same number of groups together in both hierarchies. An example of hierarchies and their difference graph can be found in Figure 1.

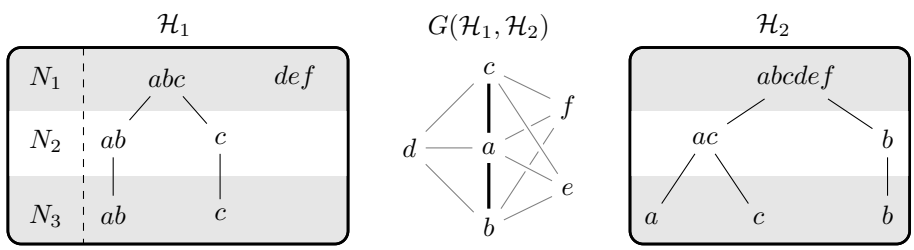

Figure 1: Example of two hierarchies $\mathcal{H}_1, \mathcal{H}_2$ of a set $S = \{a, b, c, d, e, f\}$ and their difference graph $G(\mathcal{H}_1, \mathcal{H}_2)$. The levels of $\mathcal{H}_1$ are $N_1(\mathcal{H}_1) = \{\{a, b, c\}, \{d, e, f\}\}$ and $N_2(\mathcal{H}_1) = \{\{a, b\}, \{c\}\}$. We have $d_S(\mathcal{H}_1, \mathcal{H}_2) = 3$ with the suppression set $S' = \{a, b, c\}$.

Lemma 1. *Given $G = (S, E)$ the difference graph of $(\mathcal{H}_1, \mathcal{H}_2)$ and $S' \subseteq S$, the induced subgraph $G[S']$ is the difference graph of $(\mathcal{H}_1[S'], \mathcal{H}_2[S'])$.*

[1] To simplify notations, G will sometimes be used instead of $G(\mathcal{H}_1, \mathcal{H}_2)$.

Proof. Let $G' = G(\mathcal{H}_1[S'], \mathcal{H}_2[S'])$. First, notice that $V(G') = V(G[S'])$ by definition. Second, we have $E(G') = E(G[S'])$. Indeed, for $i \in \{1, 2\}$, the number of groups where $\{s_1, s_2\} \in S'^2$ appear together is equal in $\mathcal{H}_i$ and $\mathcal{H}_i[S']$ by definition of induced hierarchy. Therefore, we have $E(G') = \{(s_1, s_2) \in S'^2, |\mathcal{H}_1[\{s_1, s_2\}]| \neq |\mathcal{H}_2[\{s_1, s_2\}]|\}$ which is also equal to $E(G[S'])$ by definition of induced subgraph.

□

Theorem 2. *$d_s(\mathcal{H}_1, \mathcal{H}_2)$ is equal to the size of the minimum vertex cover of $G(\mathcal{H}_1, \mathcal{H}_2)$.*

Proof. Let $G = G(\mathcal{H}_1, \mathcal{H}_2)$. We show first that $E(G) = \emptyset \Leftrightarrow \mathcal{H}_1 = \mathcal{H}_2$.

1. $(\mathcal{H}_1 = \mathcal{H}_2) \Rightarrow (E(G) = \emptyset)$ by definition of difference graph.
2. $(E(G) = \emptyset) \Rightarrow (\mathcal{H}_1 = \mathcal{H}_2)$
   (a) $d(\mathcal{H}_1) = d(\mathcal{H}_2) = d$ since G contains no self-loops by hypothesis. Every $s \in S$ belongs to the same number of sets in both hierarchies and $d(\mathcal{H}) = \max_{s \in S} |\mathcal{H}[\{s\}]|$.
   (b) $G = \bigcup_{i=1}^{d} G(N_i(\mathcal{H}_1), N_i(\mathcal{H}_2))$ since all elements in S belong to at most one group at a given level by definition of hierarchy. Indeed, let $(a, b) \in S^2$ such that $|\mathcal{H}_1[\{a, b\}]| = i$ and $|\mathcal{H}_2[\{a, b\}]| = j$, if $i < j$ then both $G(N_{i+1}(\mathcal{H}_1), N_{i+1}(\mathcal{H}_2))$ and G contain the edge (a, b), if $i = j$ then neither any of the $G(N_i(\mathcal{H}_1), N_i(\mathcal{H}_2))$ nor G contain the edge (a, b).
   (c) $\mathcal{H}_1 = \mathcal{H}_2$ iff $N_i(\mathcal{H}_1) = N_i(\mathcal{H}_2)$ for all $i \in [1, d]$ since $\mathcal{H} = \bigcup_{i=1}^{d} N_i(\mathcal{H})$ and any $H \in \mathcal{H}$ only belongs to one level $N_i(\mathcal{H})$ by definition of level.
   (d) By contradiction, assuming $E(G) = \emptyset$ and $\mathcal{H}_1 \neq \mathcal{H}_2$, it exists $i \in [1, d]$ such that $N_i(\mathcal{H}_1) \neq N_i(\mathcal{H}_2)$. In this case, $G(N_i(\mathcal{H}_1), N_i(\mathcal{H}_2))$ should contain at least one edge as the difference graph of two partitions (Lemma 3.1 of [3]). This contradicts the hypothesis $E(G) = \emptyset$.

We show now that a minimum suppression set for $(\mathcal{H}_1, \mathcal{H}_2)$ is also a minimum vertex cover of G. Since $(E(G) = \emptyset) \Leftrightarrow (\mathcal{H}_1 = \mathcal{H}_2)$ and according to Lemma 1, for $S' \subseteq S$, we have $\mathcal{H}_1[S'] = \mathcal{H}_2[S']$ iff $E(G[S']) = \emptyset$. The subset $S \setminus S'$ is therefore a vertex cover of G by definition. □

We assume for the rest of the paper that each element of S belongs to the same number of sets in both hierarchies. Indeed, if it is not the case, the elements that appear in a different number of groups are part of every suppression sets (equivalently, they will have self-loops and belong to every minimum vertex covers of G). Those elements can be found in polynomial time. If $G(\mathcal{H}_1, \mathcal{H}_2)$ contains no self-loops then $\mathcal{H}_1$ and $\mathcal{H}_2$ have the same depth d.

We use now the edge function $p : E(G) \to \mathbb{N}$ to encode the first level at which $(a, b) \in E$ belongs to a group of $\mathcal{H}_1$ but not $\mathcal{H}_2$ (or the opposite). We denote by G_i the subgraph of G formed by the edges $\{e \in E, p(e) \geq i\}$. Notice we have $G_1 = G$. In Fig. 1, $p(e) = 1$ for the thin edges and $p(e) = 2$ for the tick edges. Observe that subsets of elements connected with edges of values 2 (*e.g.*

$\{a, b, c\}$) belongs to the same group at the first level of $\mathcal{H}_1$ and $\mathcal{H}_2$. Moreover, those subsets are either disconnected or fully pairwise connected (for example, $\{a, b, c\}$ and d form a $K_{3,1}$ when looking only at thin edges). Those observations are generalized in Lemmas 2 and 3.

Lemma 2. *Let $S' \subseteq S$ and $i > 1$, if $G_i[S']$ is connected then, at a given lower level $j < i$, it exists two unique groups $H_1 \in N_j(\mathcal{H}_1)$ and $H_2 \in N_j(\mathcal{H}_2)$ such that $S' \subseteq H_1$ and $S' \subseteq H_2$*

Proof. Assume $G_i[S']$ is a connected subgraph but it exists two non-overlapping subsets A and B of S' with $(A \cup B) = S'$ such that A and B either (1) belong to the same group in $N_j(\mathcal{H}_1)$ but not in $N_j(\mathcal{H}_2)$ (2) do not belong to same group in $N_j(\mathcal{H}_1)$ and $N_j(\mathcal{H}_2)$. By definition of hierarchy, A and B can be split at most one time. Therefore both cases are impossible otherwise the edges between (A, B) would (1) have a value of j (2) have a value lower than j or form an empty set ($j = 0$). This contradicts our hypothesis since we assume $A \cup B$ is a connected component of G_i. □

Lemma 3. *Let $S' \subseteq S$ and $i > 0$ such that $G_i[S']$ is connected, if there exist $u \notin S'$ and $v \in S'$ such that $(u, v) \in E$ and $p(u, v) = j < i$ then $\forall w \in S'$, we have $(u, w) \in E$ with $p(u, w) = j$.*

Proof. According to Lemma 2, the elements in S' belong to the same groups of depth lower than i in both hierarchies. If it exists $u \notin S'$ such that $(u, v) \in E$ and $p(u, v) = j < i$ then there is a group at depth j in $\mathcal{H}_1$ (or $\mathcal{H}_2$) that contains $(u \cup S')$ and a group at depth j in $\mathcal{H}_2$ (or $\mathcal{H}_1$) that contains S' but not u. □

Theorem 3. *Let $\mathcal{H}_1, \mathcal{H}_1$ be two hierarchies of finite depth of a set S, computing $d_s(\mathcal{H}_1, \mathcal{H}_1)$ can be done in polynomial time.*

Proof. First, the difference graph G can be computed in polynomial time.
Let $\Psi_d(S) = \{(\mathcal{H}_1, \mathcal{H}_2) : d(\mathcal{H}_1) = d(\mathcal{H}_2) = d \wedge \forall s \in S, |\mathcal{H}_1[\{s\}]| = |\mathcal{H}_2[\{s\}]|\}$, the pairs of hierarchies of depth d where each element appears in the same number of groups. We show that, for any S, the difference graph G of any pair in $\Psi_d(S)$ is a perfect graph by induction over d.

1. *Basis.* For $d = 1$, $\Psi_1(S)$ corresponds to pairs of partitions and the graph G is therefore perfect (Theorem 3.4 of [3]).
2. *Inductive step.* Assuming it is true for d we show it is also true for $d + 1$. Let $(\mathcal{H}_1, \mathcal{H}_2) \in \Psi_{d+1}(S)$ and $G = G(\mathcal{H}_1, \mathcal{H}_2)$. We denote by $\tilde{G}$ the graph obtained by contraction of the edges of G_2 (*i.e.* with $p(e) \geq 2$) in G. The vertices set of $\tilde{G}$ is $\tilde{S}$. According to Lemma 2, elements within the same connected components of G_2 belong to the same group in the first level. Thus, $\tilde{G} = G(\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2)$ where $(\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2) \in \Psi_1(\tilde{S})$ are obtained via the fusion of each maximal connected components of G_2 into a new element

in the partitions $(N_1(\mathcal{H}_1), N_1(\mathcal{H}_2)) \in \Psi_1(S)$[2]. Therefore, $\tilde{G}$ is perfect as the difference graph of a pair of $\Psi_1(\tilde{S})$.

According to Lemma 3, the graph G can be recovered from $\tilde{G}$ by deleting each $u \in \tilde{S}$ and replacing them by their corresponding connected component S' of G_2, connecting each $v \in S'$ to the vertices previously adjacent to u in $\tilde{G}$ (the operation is called *substitution* of u by S'). Note that $\tilde{G}$ is perfect and every connected subgraphs of G_2 is perfect as the difference graph of pairs in $\Psi_d(S')$ (by hypothesis). Therefore, G is also perfect since it can be obtained after substituting perfects graphs for vertices of a perfect graph (Theorem 1 of [5], p. 255).

By Theorem 2, the distance $d_s(\mathcal{H}_1, \mathcal{H}_1)$ is equal to the size of the minimum vertex cover of G. In our case, G is perfect, so the minimum vertex cover can be computed in polynomial time. □

4. An Algorithm based on recursive maximum assignment

The minimum vertex cover problem can be solved in polynomial time for perfect graphs using the generic *ellipsoid method* [2]. This method is however not very practical. We therefore propose a combinatorial algorithm for computing the suppression distance based on observations made in the previous section (Lemma 3). We prove its correctness (Theorem 4) using the fact that a minimum vertex cover G_2 (see previous section) is a subset of the minimum vertex cover of G (Lemma 4).

We start by discussing the case of partitions $(\mathcal{P}_1, \mathcal{P}_2)$. The distance $d_s(\mathcal{P}_1, \mathcal{P}_2)$ can be computed by solving a MAXIMUM ASSIGNMENT problem based on the size of intersections between all pairs of groups in $\mathcal{P}_1$ and $\mathcal{P}_2$ using the Hungarian algorithm [4]. The resulting complexity is $\mathcal{O}((|\mathcal{P}_1| + |\mathcal{P}_2|)^3 + |S|)$.

As explained in the proof of Theorem 2, two hierarchies are equal iff the pairs $\{(N_i(\mathcal{H}_1), N_i(\mathcal{H}_2))\}_{1 \leq i \leq d}$ are all pairwise equals. However, finding a suppression set S' using a greedy "level-by-level" approach (either top-down or bottom-up) may not lead to an optimal solution. Consider the example given in Fig. 1 where $d_s(\mathcal{H}_1, \mathcal{H}_2) = 3$, a top-down approach may fail since either $\{a, b, c\}$ or $\{d, e, f\}$ can be chosen at level 1 to be part of S'. But choosing $\{d, e, f\}$ would lead to a distance of 4. Alternatively, consider the sub-hierarchies induced by the set $\{a, b, c\}$, a bottom-top approach may also fail since either a or b can belong to S' at the last level. Choosing b would lead to a distance of 2 whereas $d_s(\mathcal{H}_1[\{a, b, c\}], \mathcal{H}_2[\{a, b, c\}]) = 1$.

[2] In Fig. 1, the maximal connected component $\{a, b, c\}$ of G_2 is associated to a new element abc. $\tilde{G}$ is a star whose vertices are $\tilde{S} = \{abc, d, e, f\}$ with center abc (the fusion of $\{a, b, c\}$). The corresponding flat partitions are $\tilde{\mathcal{P}}_1 = \{\{abc\}, \{d, e, f\}\}$ and $\tilde{\mathcal{P}}_2 = \{\{abc, d, e, f\}\}$

Algorithm 1: $MSS(\mathcal{H}_1, \mathcal{H}_2)$

Input: $\mathcal{H}_1, \mathcal{H}_2$ two hierarchies of a set S
Output: $S' \subseteq S$ a minimum suppression set

1 **if** $\mathcal{H}_1 = \mathcal{H}_2 = \emptyset$ **then**
2 | **return** $\emptyset$
3 **end**
4 $S' \leftarrow \emptyset$
5 **for** $C \in \{C_1 \cap C_2 : C_1 \in N_1(\mathcal{H}_1), C_2 \in N_1(\mathcal{H}_2)\}$ **do**
6 | $S' \leftarrow S' \cup MSS(\mathcal{H}_1[C] - C, \mathcal{H}_2[C] - C)$
7 **end**
8 **return** $S' \cup flatMSS(N_1(\mathcal{H}_1[S \setminus S']), N_1(\mathcal{H}_2[S \setminus S']))$

Algorithm 1 can be used to compute a minimum suppression set (MSS) for two hierarchies. It recursively computes a suppression set for two sub-hierarchies whose elements belong to the same groups at the current level. The set $\{C_1 \cap C_2 : C_1 \in \mathcal{P}_1, C_2 \in \mathcal{P}_2\}$ contains the maximal subsets of S that are in the same group in both partitions $\mathcal{P}_1$ and $\mathcal{P}_2$. The function *flatMSS*$(\mathcal{P}_1, \mathcal{P}_2)$ returns a minimum suppression set for partitions $(\mathcal{P}_1, \mathcal{P}_2)$. The intuition behind Algorithm 1, is that if the set S' constructed at line 6 is a minimum suppression set for the sub-hierarchies then it is a subset of an optimal solution for $(\mathcal{H}_1, \mathcal{H}_2)$ (Lemma 4). Theorem 4 shows it is actually the case.

Lemma 4. *Let $G = (S, E)$ be the difference graph of two hierarchies of S, any minimum vertex cover of G_i is a subset of a minimum vertex cover of G_{i-1}.*

Proof. Let C be a minimum vertex cover of G_i, S' be a maximal connected component of G_i. The set $C' = (S' \cap C)$ is a minimum vertex cover of $G_{i-1}[S']$. According to Lemma 3, the edge cut (S', S'') forms a complete bipartite graph where S'' is the set of vertices in $(S \setminus S')$ connected to S'. The minimum vertex cover of G_{i-1} should contain either all S' or all $(S'' \cup C')$. Therefore, C' is a subset of the cover in both cases. Since it is true for the minimum cover of every maximal connected components of G_i, the set C is a subset a minimum vertex cover of G_{i-1}. □

Theorem 4. *For two hierarchies $\mathcal{H}_1, \mathcal{H}_2$ of a set S, MSS$(\mathcal{H}_1, \mathcal{H}_2)$ is a minimum suppression set for $(\mathcal{H}_1, \mathcal{H}_2)$.*

Proof. Termination: The hierarchies are of finite depth d and the recursive call is used on two sub-hierarchies of depth $d - 1$ (the "root" group is removed in both hierarchies in line 6). The condition in line 1 is always met since we assume elements of S appears the same number of sets in both hierarchies.

Correctness: Non-empty sets of $\{C_1 \cap C_2 : C_1 \in N_1(\mathcal{H}_1), C_2 \in N_1(\mathcal{H}_2)\}$ correspond to either independent or maximal connected components of G_2. Assume that at the end of the loop 5–7, the set S' is the union of the elements to be removed so that those sub-hierarchies are equal. According to Lemma 4, S'

is a subset of a minimum suppression set between $(\mathcal{H}_1, \mathcal{H}_2)$. A possible solution is therefore the union of S' and a suppression set of $\mathcal{H}_1[S \setminus S']$ and $\mathcal{H}_2[S \setminus S']$. The latter can be found only looking at the first level of both hierarchies. We can show the assumption on S' to be true by induction since the Algorithm will return a minimum suppression set if $(\mathcal{H}_1, \mathcal{H}_2)$ are partitions. □

We briefly discuss the complexity of Algorithm 1 in terms of $|S|$ and the size of the hierarchies $|\mathcal{H}_2|$ and $|\mathcal{H}_2|$. The groups intersections (line 5) can be computed in $\mathcal{O}(|S| + |N_1(\mathcal{H}_1)||N_1(\mathcal{H}_2)|)$ using an appropriate data structure. There are at most $|N_1(\mathcal{H}_1)||N_1(\mathcal{H}_2)|$ non-empty intersections.
Let $\mathcal{C}_j$ be the subsets of S for which the algorithm is used at depth j, it is the union of all intersections computed at depth $j-1$ during the algorithm execution (when $j > 1$). For $C \in \mathcal{C}_j$, the number of required operations is $\mathcal{O}((|N_j(\mathcal{H}_1[C])| + |N_j(\mathcal{H}_2[C])|)^3 + |C|)$ due to the computation of *flatMSS* (line 8). For $i \in \{1, 2\}$,

$$\sum_{C \in \mathcal{C}_j} |N_j(\mathcal{H}_i[C])| \leq |N_j(\mathcal{H}_1)||N_j(\mathcal{H}_2)|$$

since, for $j > 1$, each group in $N_j(\mathcal{H}_1)$ (resp. $N_j(\mathcal{H}_2)$) can intersect with at most $|N_{j-1}(\mathcal{H}_2)|$ (resp. $|N_{j-1}(\mathcal{H}_1)|$) groups. Therefore, we have

$$\begin{aligned} \sum_{j=1}^{d} \sum_{C \in \mathcal{C}_j} (|N_j(\mathcal{H}_1[C])| + |N_j(\mathcal{H}_2[C])|)^3 &\leq \sum_{j=1}^{d} 8|N_j(\mathcal{H}_1)|^3 |N_j(\mathcal{H}_2)|^3 \\ &\leq 8|\mathcal{H}_1|^3 |\mathcal{H}_2|^3 \end{aligned}$$

where $d = d(\mathcal{H}_1) = d(\mathcal{H}_2)$. Moreover, for $j \in [1, d]$, $\sum_{C \in \mathcal{C}_j} |C| \leq |S|$. The time complexity of Algorithm 1 is therefore $\mathcal{O}(|\mathcal{H}_1|^3 |\mathcal{H}_2|^3 + d|S|)$.

5. Conclusion and Future Work

We introduced a generalisation of suppression distance, defined for partitions, to hierarchical clusterings. Algorithm 1 is polynomial in term of hierarchies sizes and the number of elements being clustered. Although the number of groups seems to be a limitation, we believe this method is efficient in practice since it recursively removes partial solutions from the hierarchies (which is not taken into account in the complexity analysis).

Hierarchies are a subclass of *set covers i.e.* a collections of (overlapping) subsets of S. The same definition of distance can be used. In this case, finding a minimum suppression set is equivalent to the MAXIMUM COMMON SUB-HYPERGRAPH problem, which is $\mathcal{NP}$-hard [1]. The same vertex cover technique could not be directly applied to the most general set covers. However, it might be potentially useful for other similar structures with nested objects like hierarchies.

The suppression distance is a simplistic form of edit distance (minimum number of element transfers from one group to another). Both concepts are equivalent for partitions but it is not the case for hierarchical clusterings since the transfer of an element from a group can violate the inclusion constraint. The suppression distance can however be seen as a lower bound in this case. We want to investigate possible definitions of edit distances for hierarchies (*e.g.* with transfers that respect the hierarchy constraints) and their computation based on the results presented in this paper.

References

[1] Bunke, H., Dickinson, P., Kraetzl, M., Neuhaus, M., Stettler, M., 2008. Matching of hypergraphs—algorithms, applications, and experiments. In: Applied Pattern Recognition. Springer, pp. 131–154.

[2] Grötschel, M., Lovász, L., Schrijver, A., 1993. Stable sets in graphs. In: Geometric Algorithms and Combinatorial Optimization. Springer, pp. 272–303.

[3] Gusfield, D., 2002. Partition-distance: A problem and class of perfect graphs arising in clustering. Information Processing Letters 82 (3), 159–164.

[4] Kuhn, H. W., 1955. The hungarian method for the assignment problem. Naval research logistics quarterly 2 (1-2), 83–97.

[5] Lovász, L., 1972. Normal hypergraphs and the perfect graph conjecture. Discrete Mathematics 2 (3), 253–267.

[6] Meilă, M., 2003. Comparing clusterings by the variation of information. In: Learning theory and kernel machines. Springer, pp. 173–187.

[7] Porumbel, D. C., Hao, J. K., Kuntz, P., 2011. An efficient algorithm for computing the distance between close partitions. Discrete Applied Mathematics 159 (1), 53–59.

[8] Queyroi, F., Delest, M., Fédou, J.-M., Melançon, G., 2014. Assessing the quality of multilevel graph clustering. Data Mining and Knowledge Discovery 28 (4), 1107–1128.

[9] Robinson, D., Foulds, L. R., 1981. Comparison of phylogenetic trees. Mathematical Biosciences 53 (1), 131–147.

[10] Sheikh, S. I., Berger-Wolf, T. Y., Khokhar, A. A., Caballero, I. C., Ashley, M. V., Chaovalitwongse, W., Chou, C.-A., DasGupta, B., 2010. Combinatorial reconstruction of half-sibling groups from microsatellite data. Journal of bioinformatics and computational biology 8 (02), 337–356.