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. 2014. hal-00996090v2

HAL Id: hal-00996090
https://hal.science/hal-00996090v2

Preprint submitted on 14 Oct 2014 (v2), last revised 21 Apr 2015 (v3)

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.

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 the suppression 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 equals. The problem of distance computing was studied by [2] 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 recursive 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 [2, 6] or on mutual information [5] can be used to assess the distance between those objects. In particular, the partition-distance [2] is used for practical applications in bioinformatic [9].

*Corresponding author

Email addresses: `francois.queyroi@lip6.fr` (François Queyroi), `sergey.kirgizov@lip6.fr` (Sergey Kirgizov)

This work focuses on *hierarchical clusterings* (also called hierarchies) in which patterns can be recursively decomposed into smaller patterns with similar properties. The problem of distance definition between hierarchies is of interest [7]. It can be related related to the comparison of phylogenetic trees [8] in biology although those objects have typically more constraints than the decompositions studied here.

2. Definitions and Problem Statement

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 collection of non-empty subsets of S $(H_1, H_2, \ldots, H_k)$ 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 define 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 $N_i(\mathcal{H})$ denote 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. Moreover, we call $\mathcal{H}[S']$ the induced sub-collection of groups that contain $S' \subseteq S$ *i.e.* the hierarchical clustering of S' obtained after the removal of every elements of $\{S \setminus S'\}$.

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*

$$d_s(\mathcal{H}_1, \mathcal{H}_2) = \min_{S' \subseteq S} \{|S'|, \mathcal{H}_1[S \setminus S'] = \mathcal{H}_2[S \setminus S']\} \tag{1}$$

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

This definition is the same as the one introduce in [2]. As show in Theorem 1, the measure is still a distance in the case of hierarchies.

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

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

Consider three hierarchies $\mathcal{H}_1, \mathcal{H}_2$ and $\mathcal{H}_3$. Let $S_{ij} \subseteq S$ such that $\mathcal{H}_i[S \backslash S_{ij}] = \mathcal{H}_j[S \setminus S_{ij}]$. Observe that $\mathcal{H}_1[S \setminus (S_{12} \cup S_{23})] = \mathcal{H}_3[S \setminus (S_{12} \cup S_{23})]$, so we have:

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

□

Our objective is the computation of the suppression distance given in Def. 1. It is worthwhile to note that 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. Its evaluation relates to *hypergraph matching* used in pattern recognition [1]. Indeed, the problem is equivalent to finding a *maximum common sub-hypergraph*, which is $\mathcal{NP}$-hard.

3. Existence of a polynomial-time solution

We give here an non-constructive proof for this claim. 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 (Theorem 3), it exists a polynomial 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 $(\mathcal{H}_1, \mathcal{H}_2)$ 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 the same number of groups together in both hierarchies. An example of hierarchies and their difference graph can be found in Figure 1.

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 $(\mathcal{H}_1[S'], \mathcal{H}_2[S'])$.*

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, $\forall S'' \subseteq S'$ and $i = [1, 2]$, $|\mathcal{H}_i[S'']| = |\mathcal{H}_i[S']|$ 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 the definition of induced subgraph.

□

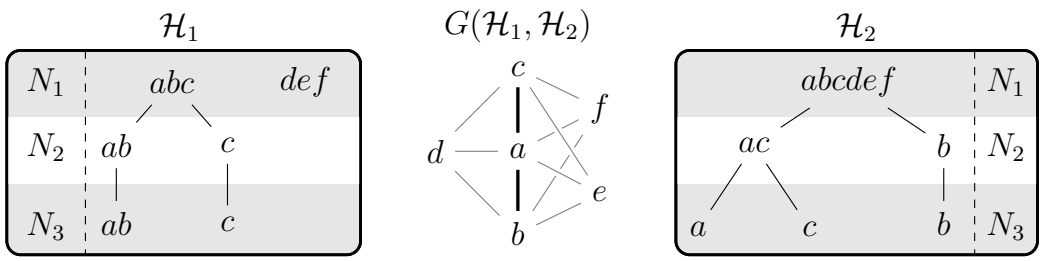

Figure 1: Example of two hierarchies $\mathcal{H}_1, \mathcal{H}_2$ of a set $S = \{a, b, c, d, e, f\}$ and their difference graph $G(\mathcal{H}_1, \mathcal{H}_2)$ ($p(e) = 1$ for gray edges and $p(e) = 2$ for black edges).

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.
   (b) $G = \bigcup_{i=1}^{d} G(N_i(\mathcal{H}_1), N_i(\mathcal{H}_2))$ since all element 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) $\mathcal{H}_1 = \mathcal{H}_2 \Leftrightarrow \bigcap_{i=1}^{d} N_i(\mathcal{H}_1) = N_i(\mathcal{H}_2)$ 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 hierarchy and levels.
   (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 [2]). 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' is therefore a vertex cover of G by definition. $\square$

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 appears a different number of groups are part of every possible suppression sets (equivalently, they will have self-loops and belong to every possible minimum vertex cover 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 the edge function $p : E(G) \to \mathbb{N}$ to encode the first level at which the couple $(a, b) \in E$ 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 $G_1 = G$. Lemma 2 and 3 provide important properties for the difference graph.

Lemma 2. *Let $S' \subseteq S$ and $i > 0$ such that $G_i[S']$ is connected, the elements of S' all belong to the same groups of depth lower than i in both $\mathcal{H}_1$ and $\mathcal{H}_2$.*

Proof. Assume $G_i[S']$ is a connected subgraph but there exist at least two non-overlapping subsets A and B of S' that belong to different groups of depth lower than i in both $\mathcal{H}_1$ and $\mathcal{H}_2$. It means that either A and B belong to the same groups in $\mathcal{H}_1$ but not in $\mathcal{H}_2$ or both do not belong to the same groups in both hierarchies. Notice that both cases are impossible otherwise all edges would have value a value lower than i (first case) or there would be no edges between the two groups (second case). 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 $u \notin S'$ 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' all belongs to the same groups of depth lower than i in both hierarchies. If there exist $u \notin S'$ such that $(u, v) \in E$ and $p(u, v) = j < i$, it means there is a group at depth j in $\mathcal{H}_1$ that contain $(u \cup S')$ and a group at depth j in $\mathcal{H}_2$ that contains S' but not u. Therefore u should be connected to every elements of S' with an edge of value j. □

One important consequence of Lemma 3 is that two connected components of G_2 either have no edges between them or form a complete bipartite subgraph in G. We now prove the main theorem.

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 de done in polynomial time.*

Proof. First, the difference graph G can be computed in polynomial time. Now, let Ψ_d denote the set of pairs of hierarchies of a set S with common depth d such that each element appears in the same number of sets in both hierarchies. We show that the difference graph G of any pair in Ψ_d is a perfect graph by induction over d.

1. *Basis.* For $d = 1$, Ψ_1 corresponds to pairs of partitions and the graph G is therefore perfect (Theorem 3.4 of [2]).
2. *Inductive step.* Assuming it is true for any d we show it is also true for $d + 1$. Let $\tilde{G}$ be the graph obtained by contraction of edges of G_2 in G. According to Lemma 2, elements within the same connected components of G_2 belongs to the same group at depth 1. The graph $\tilde{G}$ is therefore the difference graph of the two partitions $(\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2)$ obtained by the fusion of each maximal connected components of G_2 into a new element in the partitions $(N_1(\mathcal{H}_1), N_1(\mathcal{H}_2))$. There, $\tilde{G}$ is perfect (Theorem 3.4 of [2]).
   According to Lemma 3, the graph G can be recovered from $\tilde{G}$ by expanding each vertex $u \in V(\tilde{G})$ by its corresponding connected component of G_2 and connecting each element to the vertices previously adjacent to u. Now, since $\tilde{G}$ is perfect and every connected subgraphs of G_2 is perfect as the difference graph of two hierarchies of depth d, the graph G of depth $d + 1$ is also perfect according to the Theorem 1 of [4].

By Theorem 2, the distance $d_s(\mathcal{H}_1, \mathcal{H}_1)$ can be computed in polynomial time. □

4. An Algorithm based on recursive maximum assignment

If $\mathcal{P}_1$ and $\mathcal{P}_2$ are partitions, 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 [3]. The computation of the intersections takes $\mathcal{O}(n)$ and the assignment is computed in $\mathcal{O}((|\mathcal{P}_1| + |\mathcal{P}_2|)^3)$. The suppression set corresponds here to the elements that are not covered by the maximum assignment.

As explained in the proof of Theorem 2, two hierarchies are equals iff the set of couple $\{(N_i(\mathcal{H}_1), N_i(\mathcal{H}_2))\}_{1 \leq i \leq d}$ 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 Figure 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 belongs to S' at the last level. Choosing b would lead to a distance of 2 whereas $d_s(\mathcal{H}_1[\{a, b, c\}], \mathcal{H}_2[\{a, b, c\}]) = 1$.

Algorithm 1 can be used to compute a minimum suppression set for two hierarchies. It recursively computes a suppression set for two sub-hierarchies whose element belongs to the same groups at the current level.

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

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

```
if H1 = H2 = ∅ then
    return ∅
else
    S' ← ∅
    for C ∈ maxCommonGroups(N1(H1), N1(H2)) do
        S' ← S' ∪ suppressionSet(H1[C] − C, H2[C] − C)
    end
    return S' ∪ flatSuppressionSet(N1(H1[S \ S']), N1(H2[S \ S']))
end
```

The function *maxCommonGroups*$(\mathcal{P}_1, \mathcal{P}_2)$ returns the maximal subsets of vertices that are together in both partitions $\mathcal{P}_1$ and $\mathcal{P}_2$. The function *flatSuppressionSet*$(\mathcal{P}_1, \mathcal{P}_2)$ returns a minimum suppression set of elements S' such that $d_s(\mathcal{P}_1, \mathcal{P}_2) = |S'|$. Lemma 4 is used to show that the set returned by a recursion call is a subset of one optimal solution.

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, Algorithm 1 always terminates and returns a suppression set S' such that $d_s(\mathcal{H}_1, \mathcal{H}_2) = |S'|$.*

Proof. Termination: First the algorithm will terminate since the hierarchies are of finite depth 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). Moreover, the condition in line 1 is always met at some point since we assume elements of S appears the same number of sets in both hierarchies.

Correctness: Observe the function *maxCommonGroups*$(\mathcal{P}_1, \mathcal{P}_2)$ returns sets that correspond to either independent or maximal connected components of G_2. Let C be one of these sets, the sub-hierarchies $\mathcal{H}_1[C]-C$ and $\mathcal{H}_2[C]-C$ correspond to the sub-hierarchies induced by C minus the first level. 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 equals. According to Lemma 4, the set S' is a subset of a minimum suppression set between $(\mathcal{H}_1, \mathcal{H}_2)$ *i.e.* a minimum vertex cover of G_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 flat partitions. □

We shall discuss the complexity of Algorithm 1. The function *maxCommonGroups* can be implemented in $\mathcal{O}(n \log(n))$. Notice the worst case scenario is achieved when $\mathcal{H}_1 = \mathcal{H}_2$ since the results of each recursive call will be the empty set. Assuming the first $\Theta(d)$ levels correspond to one repeated group S and the last levels correspond to $\Theta(n)$ repeated groups, the time complexity is $\mathcal{O}(n^3 + dn \log n)$.

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

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

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

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

[6] 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.

[7] 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.

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

[9] 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.