Hierarchy Matching - Preliminary Results

Ron Zohar and Dan Geiger
Computer Science Department
Technion, Israel
ronron@tx.technion.ac.il, dang@cs.technion.ac.il

May 25, 2002

Abstract

A report is a list of labels of various specificity associated with a group of objects. For example, (T55, T88, T88, Tank) is a report about four objects. In this paper we develop a framework and polynomial algorithms for the fusion of consistent and inconsistent reports into a single summary report and for assessing the quality of the given reports. The proposed algorithms are part of a new framework being developed for group tracking.

1 Introduction

We consider a group of objects where each object can be described using labels of various specificity. Labels are arranged in a hierarchy according to their specificity. For example, Tank, IFV, Artillery Vehicle are labels for objects, and T88, T54 are more specific labels then Tank. An example for such a hierarchy is shown in Figure 1. A report is a list of labels associated with a group of objects, as reported by some identification device. For example, (T55, T88, T88, Artillery Vehicle) is a report about four objects. Reports do not associate which object in the group has been given a specific label. In our example, the report does not specify which object is a T55 and which is an Artillery Vehicle. The labels used in each report may be given in various levels of specificity. Furthermore, the information in the reports may be erroneous.

In this paper we develop a framework and efficient algorithms for the conservative fusion of consistent and inconsistent reports into a single summary report and for assessing the quality of the given reports, assuming the number of objects is known. These algorithms are part of a new framework being developed for group tracking.

This problem occurs in military situation assessment scenarios when group-tracking capabilities are available, but no information is available for specific objects [2]. Often, in this situation, an independent sensor is used to identify vehicles on the scene. Then, an algorithm associates the identified findings with the tracks provided by the group-tracking procedure. Given that the track is maintained for some time interval, several reports of the sensor are associated with it. Assuming that the identity of the vehicles on a track is mostly unchanged, it is reasonable to fuse the accumulating sensor reports into a single more accurate report. Since the group-tracking procedure is incapable of tracking individual targets in the arena, one needs to apply group fusion techniques as the ones developed in this paper.

A data fusion algorithm must be conservative before it may be applied in a battlefield because military customers are considerably more tolerant to human errors than to errors caused by a machine. In a conservative environment an algorithm should not try to guess the type of vehicles for which no direct information is received; ignorance is preferred to non-reliable identification and in cases of uncertainty the algorithm should prompt a human operator. In this work a specific effort was made to develop algorithms and concepts that accommodate this restrictive environment.
The rest of this paper is organized as follows. We first provide a formal description of the Hierarchically Matching problem (Section 2). We then describe an intuitive $O(mn^2)$ algorithm (Section 3) to provide background for our $O(nm|E|)$ algorithm (Section 4), where $m$ is the number of objects, $n$ is the number of reports and $l$ is the number of labels in the hierarchy. Then we present the Minimal Modification Hierarchically Matching problem and provide an algorithm for its solution for star hierarchies (Section 5). Finally, we describe how the algorithms developed herein may be used to obtain quality measures for a given set of reports (Section 6).

## 2 Problem formulation

Suppose $L$ is a finite set with a partial order $>$. Each element $l_i \in L$ is called a label and $l_1 > l_2$ is interpreted as $l_2$ is more specific than $l_1$, or equivalently, $l_1$ is more general than $l_2$. For example, $l_1$ = Tank is more general than $l_2$ = T88.

**Definition 1** A hierarchy $H$ is a directed tree $(L, E)$ where each node in $L$ is a label, and if $(l_1, l_2)$ is an edge in $E$, then label $l_2$ is more specific than label $l_1$.

For example, consider the military-domain hierarchy in Figure 1. Tank, IFV, Artillery Vehicle are labels for objects, and T88, T54 are more specific labels than Tank.

We use the term *multiset* to denote a set with repetitions. A multiset $A$ is *compatible* wrt a hierarchy $H = (L, E)$ if all elements of $A$ belong to $L$ and reside on a single directed path in $H$.

We define the Hierarchically Matching problem as follows.

**Instance:** Let $H = (L, E)$ be a hierarchy. Let $R_j = \{l_{j_1}, ..., l_{j_m}\}$, $j = 1, ..., n$, be multisets with elements from $L$ and let $R$ be the union $R_1 \cup R_2 \cup \ldots \cup R_n$ (with repetitions) of the multisets $R_j$. In other words, the multiset $R$ contains $mn$ elements from $L$, not necessarily distinct.

**Query:** Partition $R$ into disjoint multisets $A_1, ..., A_m$ each containing exactly one element from each $R_j$, $j = 1, ..., n$, such that for $1 \leq i \leq m$, $A_i$ is compatible wrt $H$; or output a valid statement that no such partition exists.

When such a partition exists, the multiset $R$ is said to be *consistent* wrt $H$ and $A_1, ..., A_m$ is called a *consistent partition* of $R$. Otherwise it is *inconsistent* wrt $H$.

The Hierarchically Matching problem is motivated by the following interpretation of its components. Consider a group of objects, in which the number of objects $m$ is known. Each object can be described using a label. The set of possible labels is denoted by $L$. Labels are arranged in a hierarchy tree as in Figure 1. A report $R_j$ is a list of labels associated with a group of $m$ objects, as reported by some identification device. For example, (T55, T88, T88, Artillery Vehicle) is a report about four objects. Reports do not associate which object in the group has been given a specific label. In our example the report does not specify which object is a T55 and which is an Artillery Vehicle. The labels used in each report may be given in various levels of specificity.

The Hierarchically Matching problem seeks a consistent interpretation of the reports $R_1, ..., R_n$ if such exists by matching the labels of each object across the $n$ reports. If the reports are consistent it is possible to summarize the information they convey using a *consensus report*.

**Definition 2** Given a consistent partition $A_1, ..., A_m$ we say that the consensus report is the multiset constructed of the most specific labels in each of the multisets $A_i, i = 1, ..., m$.

Consider using Figure 1 the reports: $R_1 = (Equipment, Tank)$, $R_2 = (Military, IFV)$. It is readily seen that these reports may be partitioned into the consistent partition: $A_1 = (Military, Tank), A_2 = (Equipment, IFV)$. In this case, the consensus report is the most specific label from $A_1$ and the most specific label from $A_2$ that is (Tank, IFV).

Now consider a third report $R_3 = (Military, Truck)$. Although each pair of the reports $R_1, R_2, R_3$ is consistent, no consistent partition exists for the three reports. Consequently the reports are inconsistent and a consensus report does not exist.

The Hierarchically Matching problem is a restricted version of an N-Dimensional Matching problem, defined as follows. Let $G = (V, E)$ be a graph and let $V_1, ..., V_n$ be a partition of $V$ into disjoint sets each containing exactly $m$ nodes. The problem is to partition $V$ into $m$ disjoint sets
Figure 1: A Hierarchy of Labels

$A_1, \ldots, A_n$ such that each $A_i$ contains exactly one node from each set $V_j$ (namely, $n$ nodes) and for every two nodes $u, v$ in $A_i$ there exists an edge $(u, v)$ in $E$. When $n = 2$, this becomes the b-partite graph matching problem. When $n > 2$, the decision version of this problem is NP complete [1]. However, for the Hierarchy Matching problem, which can be viewed as a restricted version of the $n$-dimensional matching problem, we developed a polynomial algorithm with time complexity $O(n^l + m)$.

3 Algorithm HIERARCHY-MATCHING

The algorithm described in this section is based on the fact that due to the constraint implied by a hierarchy the desired multisets $A_i$, $1 \leq i \leq m$ can be found sequentially in a greedy manner.

Definition 3 Let $R$ be a multiset with elements from a set $L$ with partial order $\succ$. An element $l_1$ is called a least element of $R$ if $l_1 \in R$ and there is no other label $l_2 \in R$ such that $l_1 \succ l_2$.

Definition 4 Let $l, l_1$ be elements of a set $L$ with partial order $\succ$, such that $l_1 \succ l$, and let $R$ be a multiset with elements from $L$. The element $l_1$ is called a least element of $R$ wrt $l$ if $l_1 \in R$ and there is no other element $l_2 \in R$ such that $l_1 \succ l_2 \succ l$.

Definition 5 Let $(H, R_1, \ldots, R_n)$ be an instance of the Hierarchy Matching problem and let the multiset $R$ be the union $R_1 \cup R_2 \cup \ldots R_n$ (with repetitions). Let $A = \{a_1, \ldots, a_n\}$ be a multiset with elements from $R$ such that: (a) The multiset $A$ is compatible wrt $H$ (b) There is an index $k \in [1, \ldots, n]$ such that $a_k$ is a least element of $R$ (c) For $j = 1, \ldots, n$, label $a_j$ is a least element of $R_j$ wrt $a_k$. Then $A$ is said to be a reducible multiset wrt to $(H, R_1, \ldots, R_n)$.

Lemma 3.1 Let $(H, R_1, \ldots, R_n)$ be an instance of the Hierarchy Matching problem where $R$ is the union $R_1 \cup R_2 \cup \ldots R_n$ (with repetitions). Let $A = \{a_1, \ldots, a_n\}$ be a reducible multiset wrt $(H, R_1, \ldots, R_n)$. Then $R$ is consistent iff $R' \equiv R \setminus A$ is consistent.

Proof: First assume that the multiset $R'$ is consistent. Consequently it may be partitioned into disjoint compatible multisets wrt $H$. Since $A$ is a reducible multiset it is also compatible wrt $H$ by requirement (a) of Definition 5. Since $R = R' \cup A$ with repetitions, the multiset $R$ must be consistent as well.

Now assume that $R$ is consistent. We will prove that $R'$ may be partitioned into $m - 1$ disjoint compatible multisets. By definition, $R$ may be partitioned into $m$ disjoint compatible multisets

3
A_1, \ldots, A_m each containing exactly one element from each R_j, j = 1, \ldots, n. Clearly, if one of the multisets A_1, \ldots, A_m is identical to the reducible multiset A then the proof is complete. However, in general, this may not be the case. We address this difficulty in the following.

Since A = \{a_1, \ldots, a_n\} is a reducible multiset, then by requirement (b) of Definition 5 there is an index k \in [1, \ldots, n] such that a_k is a least element of R. Since R is partitioned into the multisets A_1, \ldots, A_m, one of these multisets must also contain the label a_k. Denote this multiset A^* = \{a^*_1, \ldots, a^*_n\}.

Apart from the label a_k which resides in both multisets, A^* and A, the labels a_j, j \neq k need not equal the corresponding a^*_j labels. However, both A^* and A are compatible multisets containing a_k, meaning that all the labels \{a_1, \ldots, a_n\} and \{a^*_1, \ldots, a^*_n\} must reside on the same directed path in H. Furthermore, requirement (c) of Definition 5 yields a^*_j \geq a_j for j = 1, \ldots, n. Since a^*_j is more general than a_j it may replace a_j in any multiset without affecting this multiset compatibility. Consequently it is possible to locate for j = 1, \ldots, n the label a_j in the reports A_1, \ldots, A_m and swap it with a^*_j. The result is a consistent partition of R into m compatible multisets of which one is identical to A.

Lemma 3.1 implies that given an instance of the hierarchy matching problem it is possible to pursue equivalent problems with reduced size by removing reducible-multisets. The following Lemma proves that reducible-multisets can always be found in a consistent set of reports.

**Lemma 3.2** Let (H, R_1, \ldots, R_n) be an instance of the hierarchy matching problem where R is the union R_1 \cup R_2 \cup \ldots R_n (with repetitions). Let a be a least element of R. Then if R is consistent, there exists a reducible multiset A = \{a_1, \ldots, a_n\} wrt (H, R_1, \ldots, R_n) such that a \in A.

**Proof:** Since R is consistent it may be partitioned into m disjoint compatible multisets A_1, \ldots, A_m each containing exactly one element from each R_j, j = 1, \ldots, n. Since R is partitioned into the multisets A_1, \ldots, A_m, one of these multisets must contain the label a. Denote this multiset A^* = \{a^*_1, \ldots, a^*_n\}. The multiset A^* is compatible and contains a least element of R. Indeed, it is possible that for some report R_j, a^*_j is not a least element wrt a. However, this may be easily fixed by replacing a^*_j with the least element of R_j wrt a.

Lemmas 3.1 and 3.2 yield the following algorithm for hierarchy matching: At each iteration the algorithm finds a reducible-multiset. This can be done if the set of reports is consistent (Lemma 3.2). By Lemma 3.1 it is valid to remove the reducible multiset if it exists. This process is iterated. After m iterations either a consistent partition is found or such a partition does not exist and therefore inconsistency is declared. This algorithm is shown in Figure 2.

**Theorem 3.3** Let (H, R_1, \ldots, R_n) be an instance of the Hierarchy Matching problem where R is the union R_1 \cup R_2 \cup \ldots R_n with repetitions. Then, algorithm hierarchy-matching solves the problem in time complexity O(lm^2n) and space complexity O(l + mn).

**Proof:** Denote by R^{(i)} the set of reports before the i-th iteration. At each iteration the algorithm attempts to find a reducible multiset in R^{(i)}. Recall that if R^{(i)} is consistent it is always possible to find a reducible-multiset by Lemma 3.2. Therefore, if a reducible-multiset is not found, the multiset R^{(i)} is inconsistent. By Lemma 3.1, if R^{(i-1)} is consistent then so must be R^{(i)}. Consequently, the inconsistency of R^{(i)} yields inconsistency of R^{(i-1)} and so on until we obtain that R^{(1)} = R is inconsistent as declared by the algorithm.

Consider now the case when the algorithm does not output a statement that R is consistent. This case occurs if R^{(m)} is found to be consistent which implies that R^{(m-1)} is consistent and therefore that R^{(m-2)} is consistent and thus R^{(1)} \equiv R is consistent as well. Furthermore, a consistent partition to reducible multisets is readily given.

We now analyze time and space complexity. At each iteration of FindReducibleMultiset the algorithm finds a least element in R and then a least element in each of the remaining n - 1 reports. Consequently, a naive implementation for FindReducibleMultiset yields time complexity O(lm n) (where it is assumed that the query: given labels l_1, l_2 is l_1 < l_2? is O(l)). This function is called (at most) m times hence the entire algorithm has time complexity O(lm^2 n). Space is needed to store R and the multisets A, as well as the hierarchy labels which yields a space complexity of O(l + mn).
Algorithm hierarchy-matching \((H, R_1, ..., R_n)\)

Main
\[
\begin{align*}
R &\leftarrow R_1 \cup R_2 \cup \ldots \cup R_n \quad \text{(with repetitions)} \\
R^{(1)} &\leftarrow R \\
\text{for } i = 1 \text{ to } m \\
A_i &\leftarrow \text{FindReducibleMultiset}(R^{(i-1)}) \\
R^{(i)} &\leftarrow R^{(i-1)} \setminus A_i \quad \text{(reduction)}
\end{align*}
\]

FindReducibleMultiset \((R)\):
\[
\begin{align*}
a_k &\leftarrow \text{find a least element in } R \\
A &\leftarrow a_k \\
\text{for each report } R_j \text{ do} \\
&\quad j \neq k \\
&\quad a_j &\leftarrow \text{find a least element in } R_j \text{ wrt } a_k \\
&\quad \text{if } a_j \text{ is found then} \\
&\quad &\quad A &\leftarrow A \cup a_j \quad \text{(with repetitions)} \\
&\quad \text{else declare } \text{inconsistent}, \text{ halt}
\end{align*}
\]

return \((A)\)

Figure 2: Algorithm hierarchy-matching

4 Consistency Consumption

In this section we present a more efficient \(O(n(l + m))\) algorithm for determining whether a set of reports \(R\) is consistent and for finding the consensus report. Recall that given a consistent partition \(A_1, ..., A_m\) of algorithm hierarchy-matching the consensus report is the multiset constructed of the most specific labels in each of the multisets \(A_i, i = 1, ..., m\). In this section, we introduce another interpretation for the consensus report which enables a more efficient algorithm for its construction.

We start with a slight modification to algorithm hierarchy-matching as follows: We modify \(H\) by adding a parent to root \((H)\) named uproot \((H)\) and we add to each of the reports \(R_j, j = 1, ..., n\) exactly \(m(n-1)\) labels of uproot \(H\) type. Consequently, each modified report has \(mn\) labels. Note that the modified algorithm always terminates after having created the partition \(A_1, ..., A_m\) and never beforehand. This algorithm is named uproot-hierarchy-matching.

Lemma 4.1 Let \((H, R_1, ..., R_n)\) be an instance of the Hierarchy Matching problem and let \(A_1, ..., A_m\) be the multisets created by algorithm uproot-hierarchy-matching. Then \(R\) is a consistent multiset iff \((A_1 \cup A_2 \cup \ldots A_m) \cap \{\text{uproot}(H)\} = \emptyset\).

Proof: if \((A_1 \cup A_2 \cup \ldots A_m) \cap \{\text{uproot}(H)\} \neq \emptyset\) then algorithm uproot-hierarchy-matching terminated before using any of the labels \{uproot \((H)\)\}. Consequently it becomes identical to algorithm hierarchy-matching and then by Theorem 3.3 we have that \(R\) is consistent. If, however, \((A_1 \cup A_2 \cup \ldots A_m) \cap \{\text{uproot}(H)\} \neq \emptyset\) then had algorithm hierarchy-matching been applied, there would be an iteration where a reducible multiset could not be found. Consequently, by Theorem 3.3 \(R\) is inconsistent.

In the following we use additional notations as follows: \(\Gamma(v)\) denotes the set of labels including \(v\) and its descendants in a hierarchy \(H\). \(\Gamma^\ast(v)\) denotes the set of labels of all the descendants of \(v\) (\(v\) excluded), that is, \(\Gamma^\ast(v) \equiv \Gamma(v) \setminus v\). \(\Delta(v)\) denotes the set of labels including \(v\) and its ancestors in \(H\) and \(\Delta^\ast(v) \equiv \Delta(v) \setminus v\).

Definition 6 Let \((H, R_1, ..., R_n)\) be an instance of the Hierarchy Matching problem. Let \(U, V\) be sets of labels in \(H\). We define the match-count \(M_j(U, V)\) as the number of reducible multisets reduced by algorithm uproot-hierarchy-matching such that for each reducible multiset \(A = \{a_1, ..., a_n\}\): (a) \(a_j\) (the label from report \(R_j\)) is in \(U\) (b) the least element of \(A\) is in \(V\).

Definition 7 Let \((H, R_1, ..., R_n)\) be an instance of the Hierarchy Matching problem. We define
the consumption $C(v)$ of a node $v \in H$ with respect to $R = R_1 \cup R_2 \cup \ldots, R_n$ (with repetitions) as follows:
\[
C(v) = \max_j M_j(v, v)
\]

Let $V$ be a set of labels then $C(V) = \sum_{v \in V} C(v)$.

It is important to observe that the consumption $C(v)$ represents the multiplicity of label $v$ in the consensus report, namely, the number of appearances of $v$ in the consensus report (given that such a report exists). The following Lemmas enable the efficient computation of the consumption.

**Lemma 4.2** Let $(H, R_1, \ldots, R_n)$ be an instance of the Hierarchy Matching problem. Let $v \in H$ be a label. Then for $j = 1 \ldots n$ the following identity holds:
\[
C(v) = M_j(\Delta(v), v)
\]

**Proof:** At each iteration of algorithm uproot-hierarchy-matching a reducible multiset containing a label from each report is removed and the problem size is reduced by 1. Occasionally, some of the labels of the reducible multisets are of type $v$, but it is not until $v$ becomes a least element of $R^{(i)}$ (where $i$ is the iteration number) that $v$ is the least element of a reducible multiset.

Denote by $R'_k(v)$ the report with maximal number of labels of type $v$ at the first iteration, $i^*$, for which $v$ becomes a least element of $R^{(i^*)}$. Clearly, the number of labels of type $v$ in $R'_k(v)$ is $M_k(v, v)$.

After the $i^*$th iteration, reducible multisets containing label $v$ as a least element are removed until all the labels of type $v$ from $R_k$ are extracted. Afterwards, there are no further reducible multisets where $v$ is the least element. Consequently $M_k(\Delta(v), v) = M_k(v, v)$.

Since every report contributes a label for each reducible multiset where $v$ is a least element, it holds that $M_j(\Delta(v), v) = M_k(v, v)$ for $j = 1, \ldots, n$.

Noticing that:
\[
C(v) = \max_j M_j(v, v) = M_k(v, v)
\]
completes the proof.

Let $L_j(v)$ denote the multiplicity of label $v$ in report $R_j$. Let $V$ be a set of labels, then $L_j(V) = \sum_{v \in V} L_j(v)$.

**Lemma 4.3** Let $(H, R_1, \ldots, R_n)$ be an instance of the Hierarchy Matching problem. Let $v \in H$ be a label. Then for $j = 1 \ldots n$ the following identity holds:
\[
M_j(v, v) = \max(0, L_j(v) - M_j(\Delta(v), \Gamma^*(v)))
\]

**Proof:** Clearly it holds that:
\[
M_j(v, \Delta(v)) + M_j(v, \Gamma^*(v)) = L_j(v)
\]

since $M_j(v, \Delta(v)) = M_j(v, v)$ we have:
\[
M_j(v, v) = L_j(v) - M_j(v, \Gamma^*(v))
\]  

(4.1)

Note that no reducible multiset with a least element in $\Gamma^*(v)$ contains labels in $\Delta^*(v)$ from report $R_j$ unless there are no remaining labels of type $v$ in this report. Therefore we have that either $M_j(v, \Gamma^*(v)) < L_j(v)$ where $M_j(v, \Gamma^*(v)) = M_j(\Delta(v), \Gamma^*(v))$ or that $M_j(v, \Gamma^*(v)) = L_j(v)$. Consequently, equation 4.1 can be rewritten as:
\[
M_j(v, v) = \max(0, L_j(v) - M_j(\Delta(v), \Gamma^*(v)))
\]  

//

**Lemma 4.4** Let $(H, R_1, \ldots, R_n)$ be an instance of the Hierarchy Matching problem. Let $v \in H$ be a label. Then for $j = 1 \ldots n$ the following identity holds:
\[
M_j(\Delta(v), \Gamma^*(v)) = C(\Gamma^*(v)) - L_j(\Gamma^*(v))
\]
Proof: We use an induction over the hierarchy structure. We start by showing that the equation holds for leaves. If \( v \) is a leaf in \( H \), then \( \Gamma^*(v) = 0 \) so both sides of the equation equal zero. Next we assume that the lemma holds for all the children \( u_i \) of a node \( v \) and show it must then hold for \( v \). Since \( u_i \) is a child of \( v \) it holds that:

\[
M_j(\Delta(v), \Gamma^*(u_i)) = M_j(\Delta(u_i), \Gamma^*(u_i)) - M_j(u_i, \Gamma^*(u_i))
\]

and similarly:

\[
M_j(\Delta(v), u_i) = M_j(\Delta(u_i), u_i) - M_j(u_i, u_i)
\]

using Lemma 4.2:

\[
M_j(\Delta(v), u_i) = C(u_i) - M_j(u_i, u_i) \tag{4.2}
\]

notice that \( M_j(u_i, \Delta(u_i)) = M_j(u_i, u_i) \) and since \( M_j(u_i, \Delta(u_i)) + M_j(u_i, \Gamma^*(u_i)) = L_j(u_i) \) Equation 4.2 can be written as follows:

\[
M_j(\Delta(v), u_i) = C(u_i) - (L_j(u_i) - M_j(u_i, \Gamma^*(u_i))]
\]

it holds that:

\[
M_j(\Delta(v), \Gamma^*(u_i)) = M_j(\Delta(v), \Gamma^*(u_i)) + M_j(\Delta(v), u_i)
\]

consequently we obtain:

\[
M_j(\Delta(v), \Gamma^*(u_i)) = M_j(\Delta(u_i), \Gamma^*(u_i)) + C(u_i) - L_j(u_i)
\]

by substitution of the induction assumption we get:

\[
M_j(\Delta(v), \Gamma^*(u_i)) = C(\Gamma^*(u_i)) - L_j(\Gamma^*(u_i)) + C(u_i) - L_j(u_i) =
\]

\[
C(\Gamma^*(v)) - L_j(\Gamma^*(v))
\]

by summing over the children \( u_i \) of node \( v \) we obtain:

\[
M_j(\Delta(v), \Gamma^*(v)) =
\]

\[
\sum_i M_j(\Delta(v), \Gamma^*(u_i)) = C(\Gamma^*(v)) - L_j(\Gamma^*(v))
\]

Lemma 4.5 Let \((R_1, R_2, ..., R_n)\) be an instance of the Hierarchy Matching problem. The consumption of node \( v \) may be calculated using:

\[
C(v) = \max(0, \max_{j \in \{1, ..., n\}} L_j(\Gamma(v)) - C(\Gamma^*(v)))
\]

Proof: using Lemma 4.3 we get:

\[
M_j(\Delta(v), \Gamma^*(v)) =
\]

\[
\max(0, L_j(\Delta(v)) - M_j(\Delta(v), \Gamma^*(v)))
\]

using Lemma 4.4 we get:

\[
M_j(\Delta(v), \Gamma^*(v)) = \max(0, L_j(\Delta(v)) - (C(\Gamma^*(v)) - L_j(\Gamma^*(v))))
\]

the consumption of node \( v \) is then by Definition 7:

\[
\max_{j \in \{1, ..., n\}} (\max(0, L_j(\Delta(v)) - (C(\Gamma^*(v)) - L_j(\Gamma^*(v))))
\]

replacing the maximization order, noticing that the sum over \( C(u) \) is independent of \( j \) and rearranging completes the proof. //
Lemma 4.6 Let \((H, R_1, \ldots, R_n)\) be an instance of the Hierarchy Matching problem. Let \(m\) be the size of each of the multisets \(R_j\). Then, the multiset \(R = R_1 \cup R_2 \cup \ldots \cup R_n\) (with repetitions) is consistent iff \(C(\Gamma(\text{root}(H))) = m\). Otherwise, \(C(\Gamma(\text{root}(H))) > m\).

**Proof:** By Lemma 4.4 we have:

\[
M_j(\Delta(\uproot(H)), \Gamma^*(\uproot(H))) = C(\Gamma(\text{root}(H))) - L_j(\Gamma(\text{root}(H))).
\]

since \(L_j(\Gamma(\text{root}(H))) = m\) for \(j = 1, \ldots, n\), since \(\Delta(\uproot(H)) = \uproot(H)\) and since root is the only child of \(\uproot\) we obtain:

\[
M_j(\uproot(H), \Gamma(\text{root}(H))) = C(\Gamma(\text{root}(H))) - m.
\]

by Lemma 4.1 we yield that \(R\) is consistent iff no labels of type \(\uproot\) exist in the partition to \(A_1, \ldots, A_m\), that is iff \(M_j(\uproot(H), \Gamma(\text{root}(H))) = 0\). Thus we conquer that \(C(\Gamma(\text{root}(H)))\) must equal \(m\) for the reports to be consistent. Since \(M_j(\uproot(H), \Gamma(\text{root}(H))) \geq 0\), we obtain that if the reports are inconsistent then, \(C(\Gamma(\text{root}(H))) > m\).

Direct application of the properties in Lemma 4.5 and Lemma 4.6 enables the development of an \(O(n(l + m))\) algorithm for determining whether a set of reports is consistent and for finding the consensus report. The algorithm is named consistency-consumption and is presented in Figure 3.

**Theorem 4.7** Let \((H, R_1, \ldots, R_n)\) be an instance of the Hierarchy Matching problem. Algorithm consistency-consumption determines if the multiset \(R = R_1 \cup R_2 \cup \ldots \cup R_n\) (with repetitions) is consistent and finds the consensus report if exists in time complexity \(O(n(l + m))\) and space complexity \(O(l + m)\) with \(l\) being the number of labels in \(H\).

**Proof:** Algorithm consistency-consumption correctly finds the consumption of root(H) since it uses recursive calculations of the consumption as in Lemma 4.5. By Lemma 4.6 the reports
are consistent iff \( C(\Gamma(root(H))) = m \). Consequently, if \( C(\Gamma(root(H))) > m \) then the reports are declared inconsistent. Otherwise, the consensus report is readily found by taking \( C(v) \) labels of type \( v \) for each \( v \in H \).

We turn to analyzing time and space complexity. First, reading the reports and calculating \( L_j(v) \) requires \( O(mn) \) operations. Next, each label is traversed twice and at the second traversal an \( O(n) \) operation is performed. Consequently, a naive implementation yields time complexity \( O(nl + mn) \).

Space requirements for the algorithm consist of that used for saving the consumption values for each node and for the reports which yields space complexity of \( O(l + mn) \).

It should be noted that time complexity may be slightly improved by pruning the hierarchy to contain only the labels received in the reports.

5 Minimal Modification Hierarchy Matching

In this section we consider the case of inconsistent reports. We define the Min Modification Hierarchy Matching problem as follows:

Instance: Let \( H = (L, E) \) be a hierarchy. Let \( R_j = \{l_{j1}, ..., l_{jm}\} \), \( j = 1, ..., n \), be multisets with elements from \( L \) and let \( R \) be the union \( R_1 \cup R_2 \cup ... R_n \) (with repetitions). The multiset \( R \) contains \( mn \) elements not necessarily distinct.

Query: Construct a consistent report \( \hat{R} \) from \( R \), by switching minimal number of elements \( K \) in \( R \) with the label \( root(H) \).

Note that if \( R \) is consistent then \( \hat{R} \) is identical to \( R \), and that if \( R \) is inconsistent, it is always possible to construct a consistent multiset \( \hat{R} \) from \( R \) by switching all the elements in \( R \) with the label \( root(H) \).

In the following we modify our notations for \( L_j(u) \) and \( C(u) \) to explicitly pronounce their dependency in \( R \) using \( L^R_j(u) \) and \( C^R(u) \) or in \( R \) using \( L_j^R(u) \) and \( C^R(u) \).

Consider the case where the hierarchy \( H \) is a tree of height 1. In this star hierarchy we have a hierarchy root, \( root(H) \) and labels \( u_1, ..., u_l \) which are leaves of \( H \) and are directly connected to \( root(H) \).

Lemma 5.1 Let \( (H, R_1, ..., R_n) \) be an instance of the Min Modification Hierarchy Matching problem with \( m \) being the size of the multisets \( R_j \). Let \( H \) be star hierarchy with leaf-labels \( u_1, ..., u_l \). Let the multiset \( R = R_1 \cup R_2 \cup ... R_n \) (with repetitions) be an inconsistent set of reports. Then for every solution \( \hat{R} \) constructed from \( R \) by switching elements in \( R \) with the label \( root(H) \) it holds that \( \sum_{i=1}^{l} max_j L^R_j(u_i) = m \).

Proof: In our case, \( \Gamma(root(H)) = root(H) \cup \bigcup_{i=1}^{l} u_i \). Consequently and since the multiset \( R \) is inconsistent we have by Lemma 4.6, \( C^R(\Gamma(root(H))) = C^R(root(H)) + \sum_{i=1}^{l} C^R(u_i) > m \). Using Lemma 4.5 and noticing that \( L_j(\Gamma(root(H))) = m \) for all \( j \), we have:

\[
C^R(root(H)) = \max(0, m - C^R(\Gamma^*(root(H))))
\]

notice that if \( C^R(root(H)) > 0 \) then \( C^R(root(H)) = m - C^R(\Gamma^*(root(H))) \), that is, \( C^R(\Gamma(root(H))) = m \). As we have seen, this may not be the case when \( R \) is inconsistent. Consequently, we obtain that \( C^R(root(H)) = 0 \) and therefore \( \sum_{i=1}^{l} C^R(u_i) > m \).

Since the labels \( u_i \) are hierarchy leaves and using Lemma 4.5 we obtain that \( C(u_i) = \max_j L^R_j(u_i) \) and therefore \( \sum_{i=1}^{l} max_j L^R_j(u_i) > m \). Since \( \hat{R} \) is consistent we must hold that \( C^R(\Gamma^*(root(H))) \leq C^R(\Gamma(root(H))) = m \). Since \( C^R(\Gamma^*(root(H))) \equiv \sum_{i=1}^{l} max_j L^R_j(u_i) \) we obtain that:

\[
\sum_{i=1}^{l} max_j L^R_j(u_i) \leq m
\]

Assume by contradiction that \( \sum_{i=1}^{l} max_j L^R_j(u_i) < m \). Since \( \sum_{i=1}^{l} max_j L^R_j(u_i) > m \) and since \( \hat{R} \) is constructed from \( R \) by switching elements in \( R \) with the label \( root(H) \) it is always possible to
Algorithm \textsc{min-mod.-hierarchy-matching}

\begin{verbatim}
R[0] \leftarrow R
k \leftarrow 0
while \( \sum_{i=1}^{l} \max_{j} L_j^{R_i}(u_i) > m \)
    for \( i \) from 1 to \( l \),
        \( c_i \leftarrow \text{setsize}(\arg \max_{j} L_j^{R_i}(u_i)) \)
        \{ number of elements needed to be switched with the label \( \text{root}(H) \) to reduce \( \max_{j} L_j^{R_i}(u_i) \) by 1 \}
    end for
    \( p_{\text{min}} \leftarrow \min_{i} c_i \)
    \( R^{k+1} \leftarrow \text{switch appropriate labels with the label } \text{root}(H) \) to reduce \( \max_{j} L_j^{R_i}(u_{\text{min}}) \) by 1
    \( k \leftarrow k + 1 \)
end while
\end{verbatim}

Figure 4: Algorithm \textsc{min-modification-hierarchy-matching}

build a consistent report \( \hat{R} \) by un-switching elements from \( \text{root}(H) \) until \( \sum_{i=1}^{l} \max_{j} L_j^{\hat{R}}(u_i) = m \). The report \( \hat{R} \) is consistent by Lemma 4.6 and its creation requires that less elements are switched with the label \( \text{root}(H) \). This is in contradiction to the fact that \( R \) is optimal. Consequently, for every optimal solution \( \hat{R} \) it must hold that \( \sum_{i=1}^{l} \max_{j} L_j^{\hat{R}}(u_i) = m \).

Following the reasoning in Lemma 5.1, we have developed an algorithm which produces the report \( R \) by continuously (and greedily) switching elements in \( R \) with the label \( \text{root}(H) \) up to the point where \( \sum_{i=1}^{l} \max_{j} L_j^{R}(u_i) = m \). This algorithm is presented in Figure 4.

\textbf{Theorem 5.2} Let \((H, R_1, \ldots, R_l)\) be an instance of the \textsc{Min Modification Hierarchy Matching} problem with \( m \) being the size of the multisets \( R_j \). Let \( R \) be the union \( R_1 \cup R_2 \cup \ldots \cup R_l \) (with repetitions) and let \( H \) be star hierarchy with leaf labels \( u_1, \ldots, u_l \). Then algorithm \textsc{min-modification-hierarchy-matching} solves the matching problem in time complexity \( O(lmn) \) and space complexity \( O(l + mn) \).

\textbf{Proof:} The proof has two parts. In the first part we show that every optimal solution to the \textsc{Min Modification Hierarchy Matching} problem consists of a series of steps where \( \max_{j} L_j^{R}(u_i) \) is reduced exactly by 1 for exactly one node \( u_i \), \( i \in \{1, \ldots, l\} \). In the second part we show that the series of such steps obtained by an optimal algorithm is not better than the series produced by algorithm \textsc{min-modification-hierarchy-matching} and therefore that the algorithm is optimal.

Since obtaining \( R \) from \( \hat{R} \) involves switching elements with the label \( \text{root}(H) \) it follows that for every \( i \in \{1, \ldots, l\} \), \( \max_{j} L_j^{R}(u_i) \leq \max_{j} L_j^{\hat{R}}(u_i) \). Consequently, every report \( R_j \) for which \( L_j^{R}(u_i) \geq \max_{j} L_j^{\hat{R}}(u_i) \) must switch elements with the label root(H) until the number of labels remaining in \( u_i \) equals \( \max_{j} L_j^{\hat{R}}(u_i) \). Let a \textit{step} on a node \( u_i \) be termed the process in which a single element is switched with the label root(H) from each of the reports \( R_j \) for which \( L_j(u_i) \) is maximal. After a step has been made on \( u_i \), the maximal number of elements in \( u_i \) is reduced exactly by 1. The process of obtaining \( R \) from \( \hat{R} \) can be viewed as a series of consecutive steps, which terminates when all the reports satisfy \( L_j^{R}(u_i) = \max_{j} L_j^{\hat{R}}(u_i) \).

Define the \textit{cost} of a step the number of labels that are switched with the label root(H) during
Assessing Reports Quality

In this section we describe how to use consistency-consumption and min-modification-hierarchy-matching to assess the quality of a given set of reports.

**Definition 8** Let \( H = (L,E) \) be a hierarchy. Let \( S \) be a set of \( m \) objects having true labels \( s_1, \ldots, s_m \). Let \( R = \{ r_1, \ldots, r_m \} \) be a report for \( S \). Let \( R \) be the union \( R \cup S \) (with repetitions). The report \( R \) is said to be **correct wrt** \( H \) if the multiset \( R \) is consistent wrt \( H \). Otherwise, the report \( R \) is **false wrt** \( H \).

The following Lemma formalizes the relationship between consistency and correctness.

**Lemma 6.1** Let \( H \) be a hierarchy of labels. Let \( R_j = \{ l_{j1}, \ldots, l_{jn} \}, j = 1, \ldots, n \), be a set of reports for a group \( S \) of \( m \) objects having true labels \( s_1, \ldots, s_m \). Let \( R \) be the union \( R_1 \cup R_2 \cup \ldots R_n \) (with repetitions). Then if each of the reports \( R_j, j = 1, \ldots, n \) is correct wrt \( H \) then \( R \) is consistent wrt \( H \).

**Proof:** Since all the reports \( R_j, j = 1, \ldots, n \) are correct, using Definition 8, for each \( s_k, k = 1, \ldots, m \), an element \( l_{jk}^* \) exists in each report, \( R_j \), such that \( l_{jk}^* > s_k \). Since \( H \) is a tree, the labels \( l_{jk}^* \), \( j = 1, \ldots, n \) must all reside on a single directed path from root \( (H) \) to \( s_k \). Consequently, the multiset \( R \) is consistent wrt \( H \).

Lemma 6.1 provides a basic assessment tool for the reports since if the multiset \( R \) is inconsistent, then at least one of the reports \( R_j, j = 1, \ldots, n \) must be false. A more quantitative tool is presented next.
**Definition 9** Let $H = (L, E)$ be a hierarchy. Let $S$ be a set of $m$ objects having true labels $s_1, \ldots, s_m$. Let $R_i = \{h_{i1}, \ldots, h_{in_i}\}$ be a report for $S$. The **corrective distance** of report $R_i$ wrt $S$, denoted by $D_S(R_i)$, is defined as the minimal number of labels in $R_i$ that need to be fixed for $R_i$ to be correct wrt $H$. Let $R$ be the union of reports $R_1 \cup R_2 \cup \ldots R_n$ with repetitions. Then the **corrective distance** of $R$ wrt $S$ is defined as $D_S(R) \triangleq \sum_j D_S(R_j)$.

**Lemma 6.2** Let $S$ be a set of $m$ objects having true labels $s_1, \ldots, s_m$. Let $(H, R_1, \ldots, R_n)$ be an instance of the **Min Modification Hierarchy Matching** problem whose solution requires that exactly $K$ labels be switched to $root(H)$. Let the reports $R_1, \ldots, R_n$ be received for $S$ and let $m$ be the size of each of the multisets $R_j$. Let $R$ be the union $R_1 \cup R_2 \cup \ldots R_n$ (with repetitions). Then the following holds:

$$D_S(R) \geq K \geq C(\Gamma(root(H))) - m \quad (6.3)$$

**Proof:** We first prove the left hand side of Eq. 6.3: Since fixing $D_S(R)$ labels yields the correctness of $R$, switching the same $D_S(R)$ labels with the labels $root(H)$ must also yield the correctness of $R$ and by Lemma 6.1 the consistency of $R$. Consequently $D_S(R) \geq K$.

Observe that switching a single label with the label $root(H)$ may reduce $C(\Gamma(root(H)))$ by not more than 1. Consequently, the right hand side of Eq. 6.3 is readily proven since using Lemma 4.6 at least $C(\Gamma(root(H))) - m$ labels must be switched to $root(H)$ for consistency to occur. //

Eq. 6.3 provides a means for assessing the quality of a given set of reports by calculating a lower bound on $D_S(R)$. It can be seen that the bound computed by algorithm **MIN-Modification-Hierarchy-Matching** is better than the bound computed by algorithm **CONSISTENCY-CONSUMPTION**.

7 Future Work

In this paper we introduced the **Hierarchy Matching** problem and an $O(n(l + m))$ algorithm for solving it, where $n$ is the number of reports, $l$ is the number of labels in a given hierarchy $H$ and where $m$ is the number of elements in each report. We have defined the **Minimal Modification Hierarchy Matching** problem and provided an $O(lmn)$ algorithm for solving it where $H$ is a star hierarchy. Finally, we provide lower bounds of reports quality.

Although our algorithm for the **Min Modification Hierarchy Matching** problem currently supports only star hierarchies, we believe that it may be extended to more general hierarchies. We currently address this issue as part of a new framework being developed for group tracking.

Another improvement being developed includes introducing an error model and a modified algorithm that provides the most likely summary report. Given an error model, it is interesting to develop theoretic quality bounds which formalize the relation between specificity and correctness of a summary report.

Acknowledgment

The authors thank David Reinitz for useful conversations. This paper is based on part of the first author’s Ph.D research proposal submitted November 2001.

References
