Abstract - In this paper we develop a computationally efficient multiple hypothesis association algorithm for generation of alternative association hypotheses regarding cluster memberships of intelligence reports represented as belief functions. We have previously a $O(N^2K^2)$ clustering algorithm using a measure of pairwise conflicts, and a fast algorithm for classification of clusters using a more advanced measure. As these measures are similar but not identical and may have different minima we generate additional multiple association hypotheses around the solution found by the clustering algorithm. These hypotheses may then be evaluated by the classification algorithm in order to find the best overall classification of all clusters. In order to maintain the computational complexity we will investigate algorithms that run in no worse than $O(N^2K^2)$ time.

Keywords: Multiple hypotheses association, clustering, Dempster-Shafer theory, intelligence processing, belief function, force aggregation.

1 Introduction

In this paper we develop a computationally efficient multiple hypothesis association algorithm that may serve as an interface between a clustering algorithm and a classification algorithm in order to evaluate multiple different associations. When we cluster large amounts of intelligence data with an efficient clustering algorithm we usually find a good local minimum. This is the case in Dempster-Shafer clustering (DSC) [1–4], when intelligence reports represented as belief functions [5–7] are clustered based on their pairwise conflict [8] into subsets corresponding to different events that should be handled independently. In order to have a computationally efficient algorithm we only use pairwise information.

If the following classification of the intelligence reports in each cluster is made with a more advanced measure, not only taking pairwise information into account, the minimum of this measure is not guaranteed to exactly coincide with the overall minimum of the pairwise conflicts used in the clustering process. This is the reason we are interested in finding a larger set of alternative multiple hypothesis of the association of all intelligence reports. With a larger set of alternative association hypotheses the optimization of the classification measure is improved. However, it is important that the computational complexity of the generation and evaluation of all association hypotheses are no worse than the computational complexity of the clustering algorithm used, so as not to reduce the efficiency of the overall clustering and classification method.

The computational complexity of DSC using Potts spin [9, 10] mean field theory [11] is $O(N^2K^2)$ [12–15], where $N$ is the number of intelligence reports and $K$ the number of clusters, both assumed fixed for the problem. Problems where $N$ is not fixed was studied in [16], and problems where $K$ is not fixed was studied in [17], and problems where neither $N$ nor $K$ was fixed was studied in [18].

We will in this paper investigate different schemes to generate additional association hypothesis that may be evaluated in $O(N^2K^2)$ time and present an algorithm that achieves the computational requirement while maintaining a good balance between the search depth (i.e., iterative changes) and the search width of maintained alternative association hypotheses.

The proposed multiple hypothesis association algorithm, while general in scope, was developed as an interface between two modules for clustering intelligence reports and classification of clustered
subsets in a force aggregation [19–21] method for an
information fusion demonstrator for tactical
intelligence processing [22, 23].

In Section 2 we discuss the idea of multiple
hypothesis association and possible algorithms. An
algorithm is then presented in Section 3, and its
computational complexity is discussed. In Section 4
we give for the sake of completeness a short overview
of the clustering and classification parts of force
aggregation describing how the multiple hypothesis
association algorithm can fit in. Finally (in Section
5), conclusions are drawn.

2 Multiple hypothesis association

2.1 Generation of multiple hypotheses

Let us start with the initial association hypothesis
given as output from the clustering algorithm. At this
level 0 we have one initial hypothesis with \( k \) clusters, Figure 1.

At the next level we generate all possible hypotheses
with one association change. Since we have nine
reports and each report can be moved to any of the
other three clusters we have 27 new hypotheses, or
\( n(k - 1) \) new hypotheses in the general case, where \( n \)
is the number of reports and \( k \) the number of clusters
in the initial hypothesis. In Figure 2 we observe the
first two of these 27 new hypotheses.

Note that the first hypothesis generates two new
clusters that must be evaluated, while the second
hypothesis (and all the other hypotheses generated by
moving the same report) generates only one
additional cluster. In total we have in this level 36
new clusters, or \( nk \) new clusters in the general case.

Let us now study an interesting phenomenon on
levels two and deeper. As an example we continue
with hypothesis one from level one, in Figure 3.

The report that was moved at the previous level
must now be excluded. Moving this report to any
other cluster would give a hypothesis at level 3
already generated at level 2 (or level 1, if it is moved
back to where it came from). For instance, if we
moved it to the blue cluster (top right in Figure 3) we
would obtain hypothesis number 2 at level 1, already
generated.
from C to D on the first level and report 1 from A to B on the next level.

Taking this into account we may calculate the total number of new hypotheses and clusters for all branches at each level. For instance, for level 2 we start out with the \( n(k-1) \) hypotheses at level 1 and with a branching factor of \( (n-1)(k-1) \), eliminating for any duplicate generation of hypothesis in different branches, we receive a total of

\[
\frac{n(n-1)}{2}(k-1)^2
\]  

(1)

new hypotheses at level 2. The total number of hypotheses for level 0 to 2 becomes

\[
\frac{n(n-1)}{2}(k-1)^2 + n(k-1) + 1,
\]  

(2)

equality continued until a depth of \( p < n \) we will have a number of hypotheses equal to

\[
O\left(\binom{n}{p}k^p\right).
\]  

(3)

Similarly, the number of new clusters may be calculated level by level. At level 1 we have \( nk \) new clusters distributed over the hypotheses. At level 2 we receive \( n(n-1)(k-1)^2 \) new clusters and if we continue to a depth of \( p \) we have

\[
O\left(\frac{n!}{p!}k^p\right)
\]  

(4)

number of clusters, Table 1.

Table 1: Total number of hypotheses and evaluated clusters for each maximum depth.

<table>
<thead>
<tr>
<th>Depth</th>
<th># Clusters</th>
<th># Hypotheses</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( k )</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>( nk + k )</td>
<td>( n(k-1) + 1 )</td>
</tr>
<tr>
<td>2</td>
<td>( n(n-1)(k-1)^2 + nk + k )</td>
<td>( \frac{n(n-1)}{2}(k-1)^2 + n(k-1) + 1 )</td>
</tr>
<tr>
<td>( p )</td>
<td>( O\left(\frac{n!}{p!}k^p\right) )</td>
<td>( O\left(\binom{n}{p}k^p\right) )</td>
</tr>
</tbody>
</table>

When \( p \) is a constant and \( p \ll n \) a full search to level \( p \) has a computational complexity of \( O(n!k^p) \) which is much worse than that of DSC. If we intend to develop an \( O(n^2k^2) \) algorithm (the computational complexity of DSC) for generation and evaluation of multiple hypothesis association, this is only acceptable if \( p \leq 2 \). However, such a wide but shallow search space for finding a good solutions is hardly what we are looking for. By reducing the width of the search we may increase the depth to something more applicable.

2.2 Depth vs. width in hypotheses

Both the width and the depth of the search should be some function of \( N \), the number of intelligence reports, and \( K \), the number of clusters.

Let us investigate seven different algorithms with different widths and depths, in Table 2. All hypotheses at a particular level are evaluated. For instance, \( L_{1w}NK \) indicate the number of hypotheses evaluated at level 1, etc. The selection is then made of the top ranking hypotheses based on the evaluation. \( L_{0w}, L_{1w}, \) and \( L_{2w} \) indicate the number of hypotheses selected for that level (i.e., the width). This process is repeated level-by-level until the specified depth.

Table 2: Algorithms A1-A7, their widths, depths, and their computational complexity (last row). Example with \( N = 1000 \) and \( K = 10 \).

<table>
<thead>
<tr>
<th></th>
<th>( L_{0w} )</th>
<th>( L_{1w} )</th>
<th>( L_{2w} )</th>
<th>( L_{3w}NK )</th>
<th>( L_{2w}NK )</th>
<th>( L_{1w}NK )</th>
<th>( L_{0w}NK )</th>
</tr>
</thead>
<tbody>
<tr>
<td>width</td>
<td>1</td>
<td>1</td>
<td>( K )</td>
<td>( N/K )</td>
<td>( N/K )</td>
<td>( \sqrt{NK} )</td>
<td>( \sqrt{NK} )</td>
</tr>
<tr>
<td>depth</td>
<td>( N )</td>
<td>( K )</td>
<td>( K )</td>
<td>( K )</td>
<td>( K/K )</td>
<td>( \sqrt{NK} )</td>
<td>( \sqrt{NK} )</td>
</tr>
<tr>
<td>( L_{0w}NK )</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>( L_{1w}NK )</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>100</td>
<td>1,000</td>
<td>1,000</td>
<td>100</td>
</tr>
<tr>
<td>( L_{2w}NK )</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>1,000,000</td>
<td>10,000,000</td>
<td>1,000,000</td>
<td>1,000,000</td>
</tr>
<tr>
<td>( L_{3w}NK )</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000,000</td>
<td>10,000,000</td>
<td>1,000,000</td>
<td>1,000,000</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

\( O(\cdot) \) \( O(n^{3K}) \) \( O(n^{3K^2}) \) \( O(n^{3K^3}) \) \( O(n^{3K^4}) \) \( O(n^{3K^5}) \) \( O(n^{3K^6}) \)

We may observe in Table 2 that algorithms A1-A2 are very fast, but with a width equal to one. A6 is much too slow, while A3-A5 and A7 are computationally reasonable. Here A3 has some width but is shallow. A4-A5 have good width, but are both shallow. Only A7 is fast with a good width and depth. To summarize:

A1: Fast, no expansion.
A2: Faster, no expansion, but shallow.
A3: Fast, some expansion, but shallow.
A4: Fast, with expansion, but shallow.
A5: Good speed, expansion, but shallow.
A6: Slow, and deep.
A7: Good speed, expansion, and depth.

The computational complexity in the last row of Table 2 is calculated as \( depth \times width \times NK \). For most applications algorithm A7 seems appropriate.
2.3 Evaluation of multiple hypotheses

At each level all hypotheses are evaluated using some problem specific measure of fitness. We assume that the evaluation of all hypotheses (except for the first evaluation at level $L0$) can be performed incrementally in constant time.

For instance, the evaluation of the fitness $\Gamma(u^*)$ of a new hypothesis $u^*$ need only concern the two clusters involved in the incremental change if we use the previous fitness, $\Gamma(u)$. This makes the computational complexity independent of $K$.

Similarly, the evaluation of the fitness $\mu$ of the two clusters is preceded by an incrementally performed combination and decombination between the belief function moved and the belief functions corresponding to the intelligence of the two clusters, followed by the actual calculation of fitness, all usually assumed independent of $N$. Thus, allowing the evaluation to run in constant time, i.e., independent of $N$ and $K$.

3 A multiple hypothesis association algorithm

Let us present an algorithm (A7) for generation and evaluation of multiple hypothesis association of belief functions, Figure 4. We assume that all belief functions have been clustered and we receive the initial association hypothesis $\{\chi_a\}$ as the first hypothesis $u$ and input to the A7-algorithm.

In order to achieve a good balance between the $depth$ and the width of the search with an $O(N^2K^2)$ algorithm we choose $depth = width = \lfloor \sqrt{NK} \rfloor$. All clusters are classified and the fitness for each cluster $\mu$ and for the entire hypothesis $\Gamma$ is calculated.

For each successive level $\{u^*, \Gamma(u^*)\}$ is stored in a hash list and $u^*$ is looked up before $\Gamma(u^*)$ is calculated in order to avoid duplicate evaluations.

The algorithm is presented in Figure 4. Here, $n$ is a belief function, $k$ is a cluster (of belief functions) and $u$ is a hypothesis (of clusters). Let $k(n)$ be the cluster that $n$ belongs to, $u(k)$ the $k^{th}$ cluster of $u$. We have $u(k(n))$, the cluster of $u$ that $n$ belongs to.

3.1 Computational complexity

We maintain the $\lfloor \sqrt{NK} \rfloor$ best hypotheses (using the width) after evaluation at each level. Each of these hypotheses generates $NK$ new hypotheses on the next level for a total of $NK \lfloor \sqrt{NK} \rfloor$ evaluations at each level.

Performing the analysis to a search depth of $\lfloor \sqrt{NK} \rfloor$ gives a computational complexity for the A7-algorithm of $O(N^2K^2)$.

4 Multi hypothesis force aggregation

We have defined evidential force aggregation as the combination of two processes: DSC using neural

\[
\text{INITIALIZE}
\]

\[
depth = \lfloor \sqrt{NK} \rfloor;
\]

\[
width = \lfloor \sqrt{NK} \rfloor;
\]

\[
u = \{\chi_a\};
\]

\[
u.\text{freezelist} = [ ];
\]

\[
\text{hashlist} = [ ];
\]

\[
\Gamma(u) = \prod_a \mu(u(a));
\]

\[
\text{bestlist} = [ ];
\]

\[
\text{insort}\{u, \Gamma(u), \text{bestlist}\};
\]

\[
\text{REPEAT}
\]

\[
\forall p.0 \leq p \leq \text{depth Do:}
\]

\[
\text{temp} = [ ];
\]

\[
\forall u \in \text{bestlist Do:}
\]

\[
\forall n \in u.\text{freezelist Do:}
\]

\[
\forall k \neq k(n) Do:
\]

\[
\cdot u^*(k) = u(k) + \{n\};
\]

\[
\cdot u^*(k(n)) = u(k(n)) - \{n\};
\]

\[
\cdot \text{IF } u^* \notin \text{hashlist}
\]

\[
\Gamma(u^*) = \frac{\mu(u^*(k))\mu(u^n(k(n)))}{\mu(u(k))\mu(u(k(n)))} \Gamma(u);
\]

\[
\text{insort}\{u^*, \Gamma(u^*), \text{hashlist}\};
\]

\[
\cdot u^*.\text{freezelist} = u.\text{freezelist} + \{n\};
\]

\[
\cdot \text{insort}\{u^*, \Gamma(u^*), \text{hashlist}\};
\]

\[
\text{RETURN}
\]

\[
\{\chi_a^*\} = \text{bestlist}[1];
\]

Figure 4: A multiple hypothesis association algorithm

(Algorithm-A7).


We now add the intermediate component of multiple hypothesis association (MHA). Combining the developed MHA in this paper with two previous results, DSC and ECM, we achieve a multiple hypothesis force aggregation, Figure 5.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{Multiple hypothesis force aggregation.}
\end{figure}

We include the algorithms of DSC, Figure 6, and ECM, Figure 7, here for the sake of completeness of this paper. For a full description see [14, 20].

From DSC we receive a complete partitioning $\chi_a$ of all intelligence into subsets. This will serve as the initial association hypothesis in MHA. MHA will generate a large but manageable number of alternative associations around the initial solution with a computational complexity not worse than DSC. These alternative associations $u$ are evaluated using ECM.

We have

\[
\Gamma(u) = \prod_a \mu(u(a)),
\]

(5)
fitness between template aggregation performance. classification algorithm and the overall force functions of, Figure 7.

We have shown that it is possible to start out from an already performed partitioning of intelligence into subsets, and from there generate and classify additional multiple hypothesis associations around the initial partitioning. This can be achieved with a computational complexity no worse than that of DSC.

\[ J_{ij}^{\text{sp}} = -\log(1 - p_{ij})/(1 - \delta_{ij} \in A) \quad \forall i, j; \]

\[ J_{ij}^{\text{sp}} = -\log(1 - p_{ij})/(1 - \delta_{ij} \in A) \quad \forall i, j; \]

\[ s = 0; \quad t = 0; \quad \epsilon = 0.001; \quad \tau = 0.9; \quad \gamma = 0.5; \]

\[ \tau^0 = T_c \quad \text{(a critical temperature)} \]

\[ = \frac{1}{K} \max(-\lambda_{\min} \lambda_{\max}), \text{where } \lambda_{\min} \text{ and } \lambda_{\max} \text{ are the extreme eigenvalues of } M, \text{ where} \]

\[ M_{ij} = J_{ij} - J_{ij}^+ + \alpha - \gamma \delta_{ij}; \]

\[ V_{ia}^0 = \frac{1}{K} + \epsilon \cdot \text{rand}[0,1] \quad \forall i, a; \]

\[ \text{REPEAT} \]

\[ \forall i \text{ Do:} \]

\[ H_{ia}^s = \sum_{j=1}^{N} (J_{ij} - J_{ij}^+ + \alpha)V_{ja}^s, \quad f_{\geq i}^+ s \gamma V_{ia}^s \forall a; \]

\[ F_i^s = \sum_{a=1}^{K} e^{-H_{ia}^s/T_i'}; \]

\[ V_{ia}^{s+1} = \frac{e^{-H_{ia}^s/T_i'}}{F_i^s} + \epsilon \cdot \text{rand}[0,1] \quad \forall a; \]

\[ s = s + 1; \]

\[ \text{UNTIL} - 2 \]

\[ \frac{1}{N} \sum_{i,a} V_{ia}^s - V_{ia}^{s-1} \leq 0.01; \]

\[ T_i^{t+1} = \tau \cdot T_i^t; \]

\[ t = t + 1; \]

\[ \text{UNTIL} \]

\[ \frac{1}{N} \sum_{i,a} (V_{ia}^s)^2 \geq 0.99; \]

\[ \text{RETURN} \]

\[ \{ \chi_{a} \forall S_i \subseteq \chi_{a}, \forall a \neq a \quad V_{ia}^s > V_{ib}^s \}; \]

Figure 6: Clustering algorithm.

Figure 4, where

\[ \mu(u(a)) = \pi_{\oplus J_s}(T_i) \forall j \neq i, \quad \pi_{\oplus J_s}(T_i) > \pi_{\oplus J_s}(T_j) \quad (6) \]

\[ \pi(\oplus J_s(T_i)) \forall j \neq i, \quad \pi_{\oplus J_s}(T_i) > \pi_{\oplus J_s}(T_j) \quad (6) \]

and \( \{ T_j \} \) is a set of templates and \( \pi_{\oplus J_s}(T_i) \) is the fitness between template \( T_i \) and the combined belief functions of \( \chi_a \), Figure 7.

This should improve the quality of the classification algorithm and the overall force aggregation performance.

\[ \text{References} \]


