Network Based Hard/Soft Information Fusion
Stochastic Graph Matching
Geoff Gross, Kedar Sambhoos, Rakesh Nagi (co-PI)
Tel. (716) 645-3471, Email: gagross@buffalo.edu

Objectives
- Represent soft data uncertainties in a unified framework
- Utilize uncertainty in situation assessment process
  - Indicate level of uncertainty in the potential situation of interest or hypothesis
- Implement incremental methods to allow for maintenance of real-time situation awareness
- Enable intelligent querying methods within graph matching algorithm

Scientific/Technical Approach
- Incremental graph matching result updating developed to produce result better than or equal to batch Truncated Search Tree (TruST) run
- Graph matching algorithm extended for intelligent templates
  - AND/OR templates

Accomplishments
- Incremental graph matching methodology produces results better than or equal to batch run with significant runtime benefit
- AND/OR templates enable many scenarios which previously required multiple templates to be located in a single graph matching run

Challenges
- Limitations of Human Factors Literature in observational uncertainty quantification
- Only crisp semantic scoring mechanisms proposed in the literature
Motivation

• Intelligence reports continuously flow to intelligence analysts, potentially augmenting a partially matched situation of interest
  – Volume of streaming data renders batch processing methods infeasible for obtaining real time situation awareness
    • Must develop incremental search methods, harnessing previously completed computations

• Analysts respond to the existence of many different threats with some common elements
  – Rigid template structure requires many templates with similar elements, resulting in largely redundant calculations
    • Must develop flexible (AND/OR) templates to reduce overall number of graph matching executions

AND/OR Template Example

- threats against prominent personalities
  - or
- or public works
Scientific/Technical Accomplishments

• Developed incremental graph matching approach which produces results better than or equal to batch graph matching approach

• Enabled the use of templates with both AND and OR relationships on template graph nodes and edges
Publication List
Situation Assessment: Stochastic Graph Matching


Degree Awarded
• M.S. – Geoff Gross

Students supported
• Number of Graduate Students: 1

Publications
• Journal papers – 2 in review
• Conference papers – 4
Technical Approach

Stochastic Graph Matching Sequence of Events

1. Uncertainty Alignment
2. Uncertainty Transformations
3. Similarity Calculations
   a. Attribute-Attribute level similarity
   b. Aggregated to Node-Node level
4. Fuzzy Scores Ranked
5. TruST Execution on Ranked Fuzzy Scores
   a. Neighborhood scores aggregated to form overall match score
6. Incremental Score Updating
   a. Search tree reconstruction
   b. Determining affected state space
   c. Reconstruction point identification
7. Intelligent template creation
   a. AND/OR templates
• **Observation**: person, age 42 years old
  – **Observer Characteristic**: trained in age estimation
  – **Uncertainty Function**: from MURI Human Source Characterization efforts
    • **Bias**: overestimate age by 4.07 years
    • **Variance**: error is normally distributed with standard deviation of 1.65 years

![Uncertainty Function for Observed Age of 42 Years by Observer Trained in Age Estimation](image-url)
• Truncated Triangular Transformation (TTT) used to convert from probabilistic to possibilistic uncertainty function
  – Computationally simple transformation
  – Satisfies consistency principle for given portion of PDF
    • Results in least loss of specificity on calculated interval
• Node-Node similarity calculated by aggregating over all target node attributes
  – Edge-Edge scores calculated by same method

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Target Person</th>
<th>Observed Person</th>
<th>Attribute Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>6.08'</td>
<td>tall</td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>36-39</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>fuzzy{200, 215, 230}</td>
<td>210-220</td>
<td></td>
</tr>
</tbody>
</table>
Technical Approach
Chen and Chen Fuzzy Number Ranking Method

- Fuzzy score ranking method required for match priority determination
- Chen and Chen (2009) method identified for ranking fuzzy numbers
  - Considers both fuzzy number centroid as well as standard deviation
  - Assigns each fuzzy number \([0, 1]\) value which can then be sorted
Technical Approach
Graph Matching Search Tree Execution

<table>
<thead>
<tr>
<th>$v_1^T$</th>
<th>$v_2^T$</th>
<th>$v_3^T$</th>
<th>$v_4^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_1^T$</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>$v_2^T$</td>
<td>0.8</td>
<td>0.85</td>
<td>0.45</td>
</tr>
<tr>
<td>$v_3^T$</td>
<td>0.8</td>
<td>0.7</td>
<td>0.25</td>
</tr>
<tr>
<td>$v_4^T$</td>
<td>0.4</td>
<td>0.2</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Template Graph

Data Graph

Solution 1
Solution 2
Solution 3

![Graph Diagram](image-url)
Technical Approach

Results Display
Technical Approach
Incremental Result Updating

• Increments (incoming graph elements) are batched based on sampling rate and runtime knowledge
  – Increments can include insertion and deletion of both nodes and edges
• Incremental results are based on existing search tree
  – Only branches with elements incident to updates may require recalculation (based on increment scores)
Stochastic Graph Matching
Result Updating

**INPUT**
- Streaming Merged Enhanced Data
  Graph Additions
- Subscribers
  Increment Batching Utility
  Data Association
- Merged Enhanced Cumulative Data Graph

**INPUT**
- New Template Graph
  Scoring Functions
  - Node-Node Scoring
  - Neighborhood Scoring
  - Truncated Search Tree Execution

**Incremental Graph Matching Update**
- Determine Affected State Space
- Identify Reconstruction Points
- Rebuild Search Tree from Reconstruction Points

**OUTPUT**
- Identification of Situations of Interest within Observed Data

Arrows:
- ↔ initialization (new TG)
- ↔ incremental batch update
Incremental Result Updating
Determining Affected State Space

- **Affected State Space** – the data graph nodes which could have a change in their one hop neighborhood score due to increment changes
  - All nodes which are incident to the changes make up the affected state space
  - Their one hop neighborhood scores must be recalculated

Example
The one hop neighborhood scores for data graph nodes 2 and 9 must be recalculated while the one hop neighborhood score for node 10 must be initialized
**Reconstruction Point** – a point in the search tree at which lower search tree levels must be reformed

- Caused by additions/deletions in the data graph

Two conditions must be met to identify reconstruction points

**Condition 1**
- A changed score is greater than the score added at that level

**Condition 2**
- A changed score is adjacent to existing partial solution (solution prior to reconstruction point)
**Incremental Result Updating**

**Search Tree Reconstruction**

- All search tree results do not need to be retained in memory.
- Graph matching search tree can be rebuilt from result leaf nodes saved to disk.
  - Only leaf nodes need to be retained between incremental search tree updates.
- Search tree rebuilt through use of *descending adjacency sort*.

**Template Graph**

- \( v_1^T \rightarrow v_2^T \rightarrow v_7^T \rightarrow v_4^T \)

**Data Graph**

- \( v_1^D \rightarrow v_2^D \rightarrow v_3^D \rightarrow v_8^D \rightarrow v_6^D \)
- \( v_2^D \rightarrow v_3^D \rightarrow v_4^D \rightarrow v_5^D \)

**Descending Adjacency Sort** — Sort by descending score value while enforcing topological relationships:

- \( v_2^T - v_2^D = 1.00 \)
- \( v_1^T - v_7^D = 0.80 \)
- \( v_4^T - v_9^D = 0.70 \)
- \( v_3^T - v_8^D = 0.45 \)

**Adjacent not enforced**
Incremental Result Updating
Search Tree Execution

**Incremental versus Batch Search Tree Score**

- **Incremental Score**: (every increment)
- **Batch Run Score**: (every 200 increments)

**Incremental versus Batch Cumulative Execution Time**

- **Cumulative Incremental Time**
- **Cumulative Batch Processing Time**
• Indicators for priority information requests (PIRs) often contain common elements
  – Recognized through example PIRs/indicators from PSU SYNCOIN scenario (Col. Jake Graham)

• **Example Indicator** - Threats against public works, utilities, or transportation; threats of violence against prominent personalities.
Intelligent Querying Methods

AND/OR Template Example

Example AND/OR Template

**Branching Precedence Tree**
- Specifies search tree branching order
- Allows common elements to be branched on initially
- Breath parameters can be specified on each branch
  - Parameters can be used to maximize solution variety or obtain other solution mixes
Intelligent Querying Methods
AND/OR Templates

Execution Times (sec.) by Template Type

<table>
<thead>
<tr>
<th>Template Type</th>
<th>Average Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Templates without Precedence Info</td>
<td>53.5</td>
</tr>
<tr>
<td>Individual Templates with Precedence Info</td>
<td>46.2</td>
</tr>
<tr>
<td>AND/OR Template with Precedence Info</td>
<td>8.6</td>
</tr>
<tr>
<td>AND/OR Template (no neighborhood scoring)</td>
<td>8.3</td>
</tr>
</tbody>
</table>

AND/OR Template with Precedence Info vs. Individual Templates (w/o precedence)

- 83.9% runtime improvement
- Solutions where AND/OR template result is better or equal to individual template:
  - Terminal Node 1: 47
  - Terminal Node 6: 8
  - Terminal Node 8: 4
  - Terminal Node 9: 4
Intelligent Querying Methods
AND/OR Templates

AND/OR Template with Precedence Info vs. Individual Templates (w/ precedence)

- 81.4% runtime improvement
- Solution count where AND/OR template result is better or equal to individual template:
  - Terminal Node 1: 6
  - Terminal Node 6: 9
  - Terminal Node 8: 4
  - Terminal Node 9: 4

AND/OR Template (w/o neighborhood scoring) vs. with neighborhood scoring

- 3.5% runtime improvement
- Solution count where AND/OR template without neighborhood information result is better or equal to AND/OR template with neighborhood information:
  - Terminal Node 1: 0
  - Terminal Node 6: 0
  - Terminal Node 8: 0
  - Terminal Node 9: 0
- Neighborhood information is still important while specifying branching precedence tree


Year 3 Plans
Situation Assessment: Stochastic Graph Matching

• **Capability Goal:** Complete implementation and testing of increment batching utility and graph matching algorithm on SYNCOIN dataset

• **Research Goals:**
  – Explore incremental templates in support of analyst-query interaction
  – Optimize increment batching methodology
  – Temporal layering of data graph
  – Enable temporal templates specifying precedence and time range constraints on graph elements
  – **Experimental Studies**
    • Perform uncertainty alignment study to quantify alignment benefit
    • Further utilize feature level data from hard sensors
      – Supplement soft data with additional associated hard data
    • Investigate entity-attribute weighting to improve entity to entity matches
Option Year Plans
Situation Assessment: Stochastic Graph Matching

• Implementation capable of working with the (DCGS-A) Global Graph (or MapHT)
• Cloud storage of Global Graph: Using HBase
• Parallel Processing for Speedup: MapReduce processing (where applicable, e.g. scoring)
• Implementation of Query-Response system with AND/OR Agile templates