2010 17th IEEE International Conference and Workshops on Engineering of Computer-Based Systems

A Non-Numerical Predictive Model for Asymmetric Analysis

Michael L. Valenzuela, Chuan Feng, Praneel Reddy, Faisal Momen, Jerzy W. Rozenblit and Brian Ten Eyck
Dept. of ECE, University of Arizona
Tucson, Arizona, USA
mvalenz@email.arizona.edu

Ferenc Szidarovszky
Dept. of SIE, University of Arizona
Tucson, Arizona, USA
szidar@sie.arizona.edu

***Abstract*—Predicting asymmetric threats (e.g., terrorist events) is becoming ever more important. Prior works have focused on tactical, statistical, and data-fusion systems. The thrust of our work has been the development of a non-numerical predictive model for amplifying intelligence analysts' recognition of emergent threats. The intelligence community uses a Template schema for assessing courses of action. Our predictive model processes non-numerical data to arrive at automated assessment and confidence scores for these Templates. The predictive model is traceable, transparent, and utilizes Human-in-the-Loop data-fusion. For future work, this predictive model will be further enhanced with behavioral filtering. Behavioral filtering adjusts the assessment and confidence of the predictions by intelligently evaluating characteristic behavioral data. This non-numerical predictive model has been tested and verified in the Asymmetric Threat Response and Analysis Program (ATRAP).**

***Keywords*-ATRAP, Prediction, Non-Numerical, Template, Data-Fusion**

I. INTRODUCTION

The generation of intelligence faces three major challenges. The first challenge is that of asymmetry, meaning opposing forces have very different capabilities. This is manifest in asymmetric financial markets, law enforcement, politics, homeland security, and modern warfare. Second, rapidly changing environments, such as technology and insurgencies [1-3], pose problems for systems that assume a static opponent. Lastly, due to the real-time confluence of information from automated sensors, Internet, unmanned aerial vehicles, and streaming reports, analysts have more data to sift through than previously. Information overload often obscures critical patterns in the data. To further compound this problem, the data analysts use is often in a textual form, such as reports – hence the need for non-numerical predictive models.

As a result of the third challenge – information overload – data-fusion systems have become critical. Data-fusion is broken down into five levels, where the lower levels deal with fusing sensor data and higher levels deal with identifying and predicting threats. The five levels (starting at level 0) according to the Joint Directors of Laboratories (JDL) are: Preprocessing, Object Refinement, Situation Refinement, Impact Assessment, and Process Refinement (see [4, 5] for details). There is also a proposed level 5 (sixth level) of data-fusion, User Refinement [5]. In [7], level 5 data-fusion is called Human-In-The-Loop Data-fusion, in which higher level data-fusion tasks are performed manually by analysts making use of a Template schema. Templates are one way of structuring hypotheses and courses of action (COAs) (see fig. 1). It is this schema that is the basis for the non-numerical predictive model.

We are employing this model in the Asymmetric Threat Response and Analysis Program (ATRAP). The non-numerical predictive model is a key feature of ATRAP, which assists with the tasks of detecting, assessing and responding to COAs promptly and effectively. These COAs are coded up as Templates. The scoring process discussed in section II ties data to Templates, providing assessment and confidence scores for the codified COAs they represent. The scores are derived from sensor data and textual data such as reports, html, emails, documents, and other similar archival artifacts. ATRAP does sport several other sub-systems, but they are outside the scope of this paper.

There are numerous data-fusion systems [6-9]. Most of the recent research has gone into higher level data-fusion (levels 2 and 3). Game Theory and simulation are commonly used tools for determining threat assessments and evaluating enemy courses of action (ECOAs). An attrition-type discrete time dynamic game model was designed to evaluate different approaches to a conflict [6]. The extensively developed game model evaluates COAs from a tactical standpoint, but lacks any mechanism to discover new COAs. In [8, 9], the authors resolve this deficiency using adversarial Markov games, as part of a larger data-fusion system. The Markov games help determine possible tactical enemy courses of action (ECOAs) as well as their possible intents. The work also considers uncertainty and deception possibilities when attempting to predict likely future ECOAs.

Another branch of related research is the improvement of spatial forecasting methods [10, 11]. Recent advances include selecting the appropriate algorithms and intelligently increasing the dimensionality of the data to find new correlations. These methods have been successfully applied to the spatial forecasting of crime and of terrorist activities. The results of the new algorithms showed a significant improvement over the naïve model of spatial forecasting. These new methods have little to do with predicting COAs, but are still a significant contribution to predicting the locations of reoccurring phenomenon.

978-0-7695-4005-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ECBS.2010.44

Algorithms have also been developed for identifying collective behavior. In [12], the authors develop a data-fusion technique for clustering entities into groups. Their approach involves building a minimum spanning tree connecting all entities and breaking the tree up into forests if any connection exceeds a certain threshold. This produces a dynamic number of groups based on an attribute vector, opposed to a fixed number of groups as is the case with k-means clustering. They also present a procedure for determining the spatial objective for a moving group and the confidence in that estimate.

A recent trend has been Human-In-The-Loop Data-fusion [7]. The authors allow a user to define new entity types based on various sensor data requirements, which a data-fusion system can then use for classification. A human offers many unique aspects to data-fusion, especially the higher levels of data-fusion. A human can interject or act on knowledge that is missing from the system. Additionally, a system making use of a credible human would further build confidence in the system.

Not all of these previous work focused exclusively on one level of data-fusion, but rather on a whole data-fusion system [8, 9] or on multiple data-fusion levels [7]. These systems include feedback between different levels. The human-in-the-loop system allows the users to provide feedback to lower level data-fusion levels based on current deficiencies in the system.

II. Scoring Templates

A Template's structure is that of an acyclic graph (see fig. 1). There are usually three levels: Indicators, Situational Information Requests (SIRs), and Specific Order or Request (SORs). The terms are an uncategorized mix of theory, doctrine, and tactics/techniques/procedures. Thus, the names are subject to change.

Indicators are primarily for organization and structure. However, they may also refer to other Templates, providing Templates a fractal nature. SIRs are questions, specifying what is being searched. The SIR query includes entities types, keywords, and associations. There are options to automatically expand the query to include inflections of words and to make use of an ontology. The SORs represent spatial and temporal constraints on the relevant information. The spatial-temporal query represents a 5-tuple $\{A, T, t\text{-}, t\text{+}, r\}$, where A is a well defined area, T is a well defined time range, t- is the allotted time before the start of T, t+ is the allotted time after the end of T, and r is the allotted radius outside of A. A and T represent the perfect match constraints while t-, t+, and r represent extensions to A and T for imprecise matches.

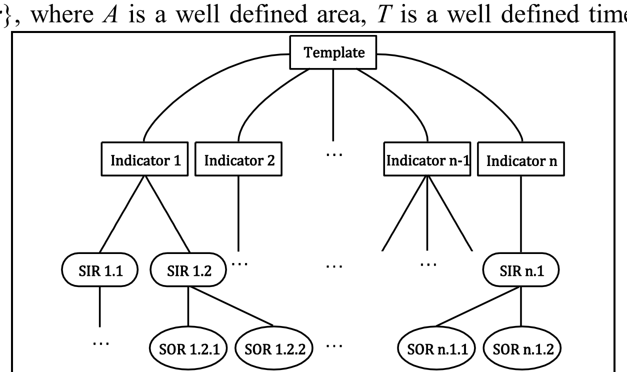

Figure 1. Example Template

By treating the SORs as the location of transduction and adding a weight, threshold, and activation function to each node in the Template, the Template behaves similar to an artificial neural network. We call this extended model a Quasi-Neural-Network (QNN). Fig. 2 depicts an example Template which is looking for an event at either location A or B, but not both. The numbers along the connections are the connection weights and the numbers inside the nodes are the thresholds. All activation functions are linear.

One aspect of our non-numerical predictive model is designing an algorithm which mimics how intelligence analysts rank Templates. To make the ranking of the Templates easier, we introduce the concept of a score. A score is not a percent likelihood. This is because the variance in human behavior and human error is so large, that a probability is hard to estimate. While a user could learn to interpret the score with experience, the purpose of the score is to allow for the comparison of one Template to other Templates.

To ensure that we designed a credible scoring algorithm, we worked with an intelligence analyst Subject Matter Expert (SME) to understand how the scoring should behave under various conditions, what entities should be tracked, and what database design would work best. We broke down the scoring into five separate parts and came up with various constraints and desired attributes for each part. We also came up with several test cases to verify the correct functionality of our scoring algorithm.

A. Information Retrieval

The first question to answer was how to take the query information from the SIRs and SORs and pull the appropriate information from the database. Despite free text being partially analyzed by humans and put into the database with additional fields, much of the information still requires some degree of natural language processing. While ATRAP is working on methods of extracting names, places, and events automatically from free text, differentiating a name of a person verse the name of an organization can be a very difficult task for a computer.

For fields such as the entity type, an ontology is used to optionally expand the search for more general or more

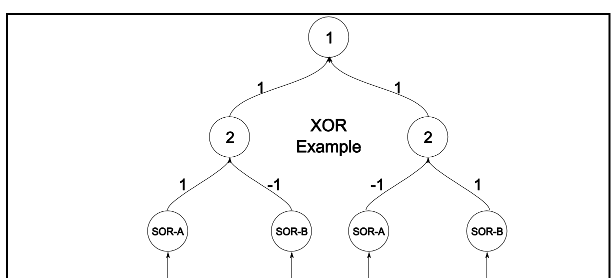

Figure 2. Example Quasi-Neural Network

specific types. For instance, if the user specifies an entity type of type vehicle and includes more specific types, the system may return a hit for an entity of type truck. For fields involving text, inflections of the words can be automatically added (again, if specified by the user). While not currently implemented in the prototype, the optional use of a thesaurus has been discussed to improve the recall at the expense of precision of such queries.

B. *Evaluation of Retrieved Data & Fuzzy Matching*

Once the data and documents have been retrieved which support (or deny) the queries, that information needs to be evaluated and converted into numerical vectors. The vector consists of three terms: confidence in the information, relevance of the match, and the assessment of the match. If the information came from a free-text report, the confidence in the information was entered by a human. If the information came from sensor data, the confidence is assumed to be very high. The relevance reflects how well the query matches the retrieved data. The assessment indicates whether the data supports or denies the query.

An SME provided guidelines for evaluating the data retrieved from the query. Some of these behaviors include the impact of fuzzily matching spatial-temporal constraints, fuzzily matching generalizations of the entity types, user adjustable parameters, and methods for tweaking how detrimental a fuzzy match is compared to an exact match.

The exact variables in the SOR constraint query, *A* and *T*, specify the perfect matches. Data retrieved from the database that fulfills these requirements perfectly do not suffer a penalty to their relevance score for the spatial-temporal constraints. However, if there is a fuzzy match, their relevance is determined by user-selectable falloff functions, which specify how the relevance transitions from the perfect match to a perfect failure. A perfect failure is defined as being outside the specified spatial-temporal region even when expanded by the parameters: *t*-, *t*+, and *r*. An example falloff function is shown in fig. 3.

The relevance score can also be impacted by a generalization of the entity type searched. If the user chooses to include all levels of generalizations of the specified entity type (e.g., truck) when building an SIR query, ambiguous entities may be found (e.g., vehicle). The user can specify how many levels up the entity type ontology to travel when expanding the search and the penalty for each level. For example, every level that must be traversed up the ontology (more general terms), in order to find a match, could multiply the relevance by 0.5. This is demonstrated in

$$R' = R \cdot \sigma^{Max(0,L)}, \qquad (1)$$

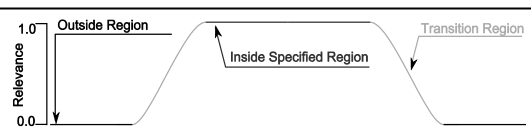

Figure 3. Sinusoidal Falloff Function

TABLE I. PSEUDOCODE RESOLVING MULTIPLE HITS

```
Vect3D scoredDataVectors;
UserParameter a, b, c, d;
UserFunction sigmoid;

List<Vect2D> polarVectors;
foreach( dataV in scoredDataVectors ){
  Vect2D polVect = new Vect2D();
  polVect.cert = pow(dataV.conf, a) *
        pow(dataV.relv, b);
  polVect.asse = pow(dataV.conf, c) *
        pow(dataV.asse, d);
  polarVectors.insert(polVect);
}

Vect2D vectSum(0,0);
Vect2D tmpV;
foreach( polV in polarVectors){
  tmpV.cert = sigmoid.inverse(polV.cert);
  tmpV.asse = polV.asse;
  vectSum = vectorAdd( vectSum, tmpV);
}
List<vect2D> conflictingData;
conflictingData = FindCnfl(polarVectors);
vectSum.cert = sigmoid.forw(vectSum.cert);

double conflictSum = 0.0;
foreach( conflictV in conflictingData){
  conflictSum += conflict.cert;
}
vectSum.cert /= (1 + conflictSum);
SOR.Vector = vectSum;
```

where *R'* is the relevance once adjusted for a generalization of the entity type, *R* is the relevance prior to accounting for the generalization, σ is a user adjustable parameter which determines how detrimental generalization is, and *L* is the number of levels (in the ontology) more generic the found entity type was than the specified entity type.

C. *Resolving Multiple Hits*

The next issue is converting all the informational vectors into a single vector for the SORs that can also be used for the rest of the Template scoring system. It was decided that the confidence in the information could then be combined with assessment and relevance to represent the data as a two-dimensional polar vector for the rest of the scoring procedure. This simultaneously simplifies the task and transforms the problem to something that matches the intelligence analyst SME's thought process better. The explanation of the two-dimensional polar vector is that the assessment reflects what the underlying information indicates. A measure of certainty indicates the strength in that belief. Our mathematics SME recommended Cobb-Douglas products to accomplish this transform, primarily due to their behavior and wide acceptance in business and financial environments. The Cobb-Douglas product is a weighted multiplication. Equation (2) shows a Cobb-Douglas product, where *x* is a vector of components to be combined and α is a vector of weights, both of length as *n*.

$$CDP(\boldsymbol{x}, \boldsymbol{\alpha}) = \prod_{i=1}^{n} x_i^{\alpha_i} \tag{2}$$

Our intelligence analyst SME and team developed 11 key constraints and desired behaviors for resolving multiple hits. A few properties include: conflicting information should lower the confidence, order of elements should not matter, model dimensioning returns when there are many hits, various behaviors for different quality matches, and the simpler the better. When all aspects were combined, competing algorithms were then evaluated based on understandability and simplicity.

The current procedure proceeds as follows (see table 1). First, the dimensionality is reduced to a two-dimensional polar vector using Cobb-Douglas Products. Next, the lengths (certainty) of all vectors are stretched by using the inverse of a user selectable sigmoid function. All vectors are then summed and the resulting vector is normalized using the user-selectable sigmoid function. Lastly if there are two or more pieces of information are in conflict, the length of the normalized resulting vector is adjusted to reflect disagreeing data.

Since the last prototype of the model, we have talked with two SMEs for some additional constraints. This included methods for determining redundant data, measure of similarity, and a more statistical method for fusing the data based on the independence of the data. If a method for determining the dependence of two text-based pieces of information can be formulated, verified, and tested, then a statistical solution can be used for resolving multiple query hits. The intelligence analyst would then be able to select which algorithm to use for this part.

D. *Propagation of Score*

Once two-dimensional polar vector exists for each SOR, the system must propagate the scores to higher levels to answer the SIRs, Indicators, and eventually the Template. The propagation method is the activation function present at each node in the Template.

To have the activation function behave as a generally accepted artificial neuron response without truncating information, a linear activation function was chosen. The slope of the linear function is determined by the inverse of the sum of the absolute values of the weights of all the child links. Thus, this effectively performs a weighted average if the threshold is equal to the sum of the absolute values of the weights.

The behavior for the threshold is still under investigation. There are few options currently being debated. One solution would be to drop the threshold. Another solution involves using only the most influential information. The third solution focuses on making the behavior as similar to a piecewise linear neural network as possible. Each currently proposed solution has some disadvantages.

The solution proposed using the most influential information would work as follows: the child vectors in order of greatest influence on the SIR or Indicator is selected and added until the sum of their weights equals the threshold. The influence is determined by a projection of the assessment and confidence. This solution has a few problems. While the confidence of the resulting vector is bounded by the lowest and highest child vectors' confidences, it does not follow any predictable pattern. It does not mimic the threshold behavior of neural network configurations.

The third option would involve decomposing the polar vectors into non-polar components. The vectors would be decomposed into two non-polar components (confirm-deny data and supporting but neutral data). The first step is the same, sum of all the vectors. Hold onto the un-normalized vector, but set the confidence to the normalized length. The confirm-deny component would be set to the minimum of the confidence and itself divided by the threshold. The other component is then calculated to preserve the confidence. This third option behaves most like a piecewise linear neural-network, but the confidence remains constant with respect to the threshold. Due to the nature of this third option, the confidence could be calculated in another manner to fix this.

E. *Template Score and Ranking*

The last step in evaluating Templates is providing scores and ranking all of the Templates selected for ranking. The first part of this procedure is identical to the propagation of the score, where all of the Template's Indicators' vectors are averaged to produce an assessment and confidence. This produces a final two dimensional vector which must be transformed into a one dimensional number for ranking purposes.

A single number is generated from taking the projection of the confidence and assessment onto an axis. There are three reasons why a projection is used instead of the aforementioned Cobb-Douglas product. First, projection is more intuitive due to the polar nature of the vectors. Secondly, both have similar behavior if either number becomes zeros. Additionally, projection is simpler when working with negative numbers. The resulting number can then be used for ranking purposes.

There is one more step when it comes to Template scoring. It is dangerous to report a number which ranges between -1.0 and 1.0, since the number can be misinterpreted. Positive scores could accidentally be taken as likelihoods. Thus the last step is a monotonic distortion to prevent the user from interpreting the score as a percent likelihood. The monotonic property preserves the relative order of the Templates.

III. APPLICATION AND IMPLEMENTATION

Because the non-numerical predictive model for asymmetric analysis uses a Template schema, its applications are not limited to military intelligence. Virtually any domain which uses significant non-numerical data can benefit from this predictive model since hypotheses can be codified as Templates. Applications range from criminal investigations, financial markets, and homeland security to evaluating political outcomes. One specific application is that for ATRAP, a practical and flexible toolbox for intelligence analysts. The model provides automatic generation of an assessment and confidence for hypotheses (Templates). We worked with SMEs to help develop a non-numerical predictive model that intelligence analysts could use. We have tested and verified the algorithms via the use of ATRAP.

The non-numerical predictive model has been implemented as part of ATRAP, written for Microsoft Windows XP and Vista. The programming language used was C#, using Microsoft .NET Framework 3.5. The database system was built upon .netTiers and Microsoft SQL Server 2005.

IV. Experiments and Results

We have performed multiple preliminary experiments to help verify and validate the predictive model. The primary focus of the experiments was to test if the model behaved as our SMEs wanted (e.g., described in Section II.*C*). For brevity, only four types of experiments will be discussed. The *white* test case has a database containing information to confirm a Template. Another type of test case is the *black* test case. It involves a database with irrelevant material for the Templates being scored. There is a *gray* test case, which uses a much larger database, with both relevant and irrelevant data. Lastly, the database remains constant, but several different Templates are evaluated at the same time in a *mixed* test case.

When the model was tested, the *white* case provided very good hits to all of the Indicators, meaning all their children also scored highly. The average Indicator assessment was approximately +0.707 (appears to be happening) with very high certainty. The Indicators had a neutral assessment of 0.000 (non-negative and non-positive) with very low confidence in the *black* case. The *gray* case did not take noticeably longer to run, despite the database being an order-of-magnitude larger. Some of the Indicators found relatively good matches while other found little to no data. The *mixed* case results showed a spectrum of results, ranging from a couple Templates performing well (indicating that they appear to be happening) to most Templates performing similar to the results from the *black* case (indicating a lack of data). These results reflected what the SMEs desired.

V. Conclusion and Future Work

The non-numerical predictive model has been tested and verified in ATRAP. The model is made malleable from its many options, including: an entity ontology to optionally expand search parameters, adjustable Cobb-Douglas products, various sigmoid and falloff functions, and weights and thresholds for Indicators, SIRs, and SORs. It provides means for automatically surveying codified hypotheses (Templates) by providing an assessment and confidence from non-numerical data. The predictive model has primarily five parts: information retrieval, evaluation of the retrieved data, resolving multiple hits, score propagation, and final score generation.

Another on-going research aspect is enhancing the predictive capabilities (Template evaluation) by making use of behavioral data. Behavioral filtering adjusts the confidence and assessment scores of each prediction modeled by a Template. Characteristic attributes which are easily and cheaply measurable, will be the basis for these adjustments. A powerful and flexible prototype inference engine can be integrated into the predictive model to enhance the accuracy.

Future versions of the non-numerical predictive model will include further validation of Template scoring algorithms, computational scalability with orders-of-magnitude larger datasets, integration of the inference engine and other possible improvements to the model.

References

[1] [1] D. Woods and S. Dekker, "Anticipating the effects of technological change: a new era of dynamics for human factors," Theoretical Issues in Ergonomics Science, vol. 1, pp. 272-282(11), 1 July 2000.

[2] E. Roth, R. Scott, S. Deutsch, S. Kuper, V. Schmidt, M. Stilson and J. Wampler, "Evolvable work-centred support systems for command and control: creating systems users can adapt to meet changing demands," Ergonomics, vol. 49, pp. 688, 2006.

[3] A. H. Cordesman, The Lessons of Afghanistan : War Fighting, Intelligence, and Force Transformation. CSIS Press, Washington, D.C., 2002.

[4] D. McDaniel, "An information fusion framework for data integration," in Proceedings of the 13th Software Technology Conference, 2001.

[5] A. N. Steinberg, C. L. Bowman and F. E. White, "Revisions to the JDL data-fusion model," in Sensor Fusion: Architectures, Algorithms, and Applications III, 1999, pp. 430-441.

[6] D. Shen, G. Chen, J. B. Cruz, L. S. Haynes, M. Kruger and E. Blasch, "Game-Theoretic Modeling and Control of Military Air Operations with Retaliatory Civilians," Aerospace Conference, 2007 IEEE, pp. 1-10, 2007.

[7] S. Mahoney, J. Pfautz, T. Fichtl, S. Guarino, E. Carlson and M. Farry. "Enabling robust C2 systems through evolvable human-in-the-loop data-fusion," Presented at International Command and Control Research and Technology Symposium, vol. 6, pp. 14, 2009. [Online] Available: http://www.dodccrp.org/events/14th_iccrts_2009/papers/147.pdf

[8] G. Chen, D. Shen, C. Kwan, J. B. Cruz and M. Kruger, "Game Theoretic Approach to Threat Prediction and Situation Awareness," Information Fusion, 2006 9th International Conference on, pp. 1-8, 2006.

[9] M. Wei, G. Chen, J. Cruz Jose B., L. Haynes and M. Kruger, "Applying Spatial-Temporal Model and Game Theory to Asymmetric Threat Prediction." 12th ICCRTS, vol. 12, no. 3, pp. 63, 2007.

[10] D. Brown, J. Dalton and H. Hoyle, "Spatial Forecast Methods for Terrorist Events in Urban Environments," Lecture Notes in Computer Science., vol. 3073, pp. 426-435, 2004.

[11] M. B. Mitchell, D. E. Brown and J. H. Conklin, "A Crime Forecasting Tool for the Web-Based Crime Analysis Toolkit," Systems and Information Engineering Design Symposium, 2007. SIEDS 2007. IEEE, pp. 1-5, 2007.

[12] A. Hossain, N. Walmsley and P. Pearce, "A Minimum Spanning Tree Approach to Identifying Collective Behaviour and Inferring Intent for Combat Models," International C2 Journal, vol. 2, no. 2, September 2008. [Online] Available: http://www.dodccrp.org/files/IC2J_v2n2_01_Hossain.pdf. Sept. 19, 2009.

[13] R. M. Akita, "User based data-fusion approaches," Proceedings of the Fifth International Conference on Information Fusion, 2002., vol. 2, pp. 1457-1462.