Presentation of Information for Link Analysis

Jerome Thomere & Michael Wolverton

Artificial Intelligence Center
SRI International
333 Ravenswood Ave
Menlo Park, California 94025
thomere@ai.sri.com, mjw@ai.sri.com

Abstract

SRI's LAW (Link Analysis Workbench) is a system that helps intelligence analysts detect occurrences of situations of interest by finding pattern instances in vast amounts of data using graph edit distance matching techniques. However to be completely successful it has to convey the results of the such findings to the users in a way that they can quickly grasp, not only to make use of it or to present it, but also to provide the feedback necessary to fine tune the discovery process using LAW.
This paper presents what we think is required in order to achieve this goal, the solutions that were designed and implemented in the current LAW architecture and the approach that we envision in the next installments of LAW
.

Introduction

LAW's main function is to detect instances of a pattern within a given dataset (Thomere et al. 2004). The data in question are relational data, generally in the form of a relational database, so one dataset can be viewed as an extremely large graph. Nodes of that graph are generally entities (persons, organizations, locations…) or events. Links are labeled binary relations between those nodes. A pattern is a similar graph with the exception that each node corresponds to a *class* of events or entities. When LAW is given a pattern and a dataset, it returns matches. Each match (or *hypothesis*) is a subset of the data graph where the nodes are instances of the classes defined in the pattern and the links between those nodes are the same as the links between the nodes in the pattern. To be precise, this is that case for an *exact* match. The match can be partial or inexact, within a threshold determined by the analyst. In addition, a pattern may include constraints, which are not link directly expressed in the data, but relations (often numerical) between attributes of the entities in the data. Such constraints can be temporal, for instance.

Figure 1 shows a simple example from a surveillance application domain to detect the action of an actor moving inside some location. However, patterns can be more complex than that: they can be composed of several levels of sub patterns that may represent logical disjunctions. Moreover to a sub pattern can be associated the concept of cardinality, which determines the number of different possible matches inside the same overall match.
To implement the matching process, LAW uses a graph-edit distance method (Wolverton et al. 2003). This distance corresponds roughly to the number of operations one has to operate on the pattern graph to map a sub graph of the data. The algorithm used is a version of A* which allows the process to be interrupted at any time.
When the process terminates (either by the analyst action or when all the data have been examined) LAW is in possession of a list of hypotheses each of which corresponds to a match (partial or exact) of the pattern, i.e. a set of instances linked by relations.
The challenge we discuss in this paper is how to convey the resulting hypotheses in the most useful possible way to the user.

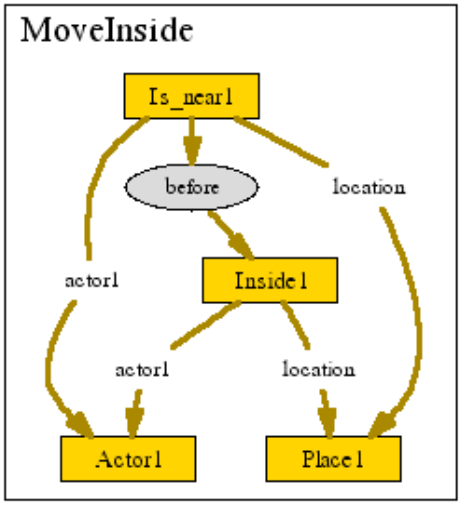

Figure 1: example of a simple pattern

Requirements and Challenges

As presented in (Wolverton et al. 2004) the intended use of the LAW system is as part of a development cycle. In summary, after having built a first pattern with the help of LAW's pattern editor, the analyst examines the results and uses them to refine the pattern, either by adding nodes or links, or by releasing some constraints, or by building another variant of the pattern, or anything she finds appropriate to get more significant results. That means that she needs to understand as much as possible not only the hypotheses, but also the rationales that lead LAW from the pattern to those hypotheses, and how they match the pattern. These general goals lead to the following requirements:

1. The first one is thus naturally to display the pattern to the user in the most explicit way. That also implies that the pattern editing should reflect the display. One of the

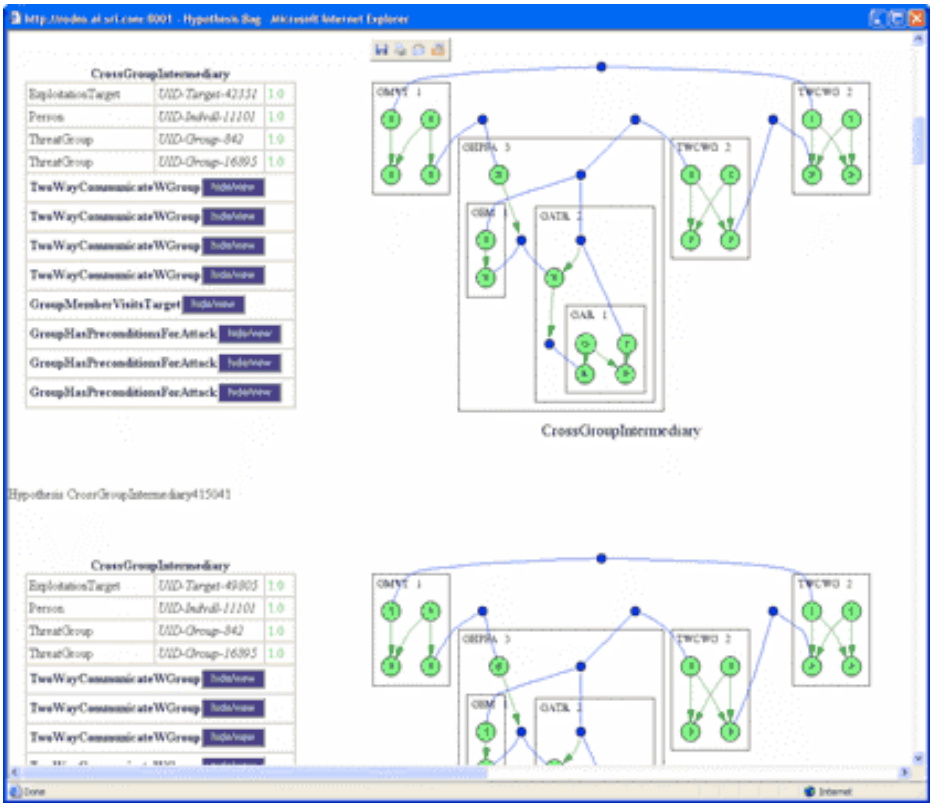

Figure 2: Detailed match results display

Ungroup	28 Matches			
	VulnerabilityMode1	ThreatGroup2	ResourceType2	ExploitationTarget1
Matches with ExploitationTarget1=Ta-15490	Mo-15227 Mo-19021 Mo-5733 Mo-23538 Mo-27772 Mo-8631	Gr-14484 Gr-14818 Gr-21819 Gr-11125 Gr-28968 Gr-11006 Gr-2786	Re-46 Re-30 Re-35 Re-10 Re-8 Re-9 Re-21 Re-32	Ta-15490
Matches with ExploitationTarget1=Ta-5996	Mo-1793 Mo-9048 Mo-12028 Mo-12969 Mo-29701	Gr-14818 Gr-21819 Gr-28968 Gr-11006	Re-20 Re-47 Re-48 Re-18 Re-13	Ta-5996

Figure 3: Results grouped along nodes

benefits of a graph approach is that it naturally translates into a graphical view, as we saw in the previous figure of a pattern.

2. A second requirement is to display each individual hypothesis in a way that is both easy to understand and close to the pattern from which it originates. This can be challenging when the pattern is complex, for instance if it includes several level of embedded sub-patterns or if it uses cardinality.

3. In the first iterations of the development cycle, the matcher may return a huge number of hypotheses. In those cases, the display component of LAW has to organize the results on order to avoid overloading the user with information. That can mean filtering the matches, sorting them, clustering them into significant groups or summarizing the results in one way or another.

4. Conversely, at the beginning, LAW can very well return no matches at all. Then, another requirement is for the user to be able to detect the cause of the null result. That could mean that no situation of interest is occurring, but that could also be that some flaw in the pattern prevented the matching process to move along. That's why the user should be able to visualize the matching process, to figure the possible bottlenecks or false tracks.

5. Whether the matcher does not return enough results, or returns too many, or doesn't return the right ones, the problem may be in a wrong interpretation of the data. That's why it is almost always required to give the analyst the ability to explore directly the data without going through a pattern. For instance, knowing that a pattern hasn't returned a match she expected, the analyst can explore the data linked to a node that should have been part of one of the matches, and explore the links to and from than node.

6. Finally, an important requirement for the display of results, which is not directly linked to the development cycle, is to help the analyst present them to other analysts, or to the policy maker. It is not enough to get convincing results for a possible thereat situation, for instance, you also have to convince other people so that the results are useful.

Current Design

In the current implementation of LAW we put the emphasis on the pattern editing and the display of the matches, which correspond to the requirements 1, 2 and 3.

Each pattern is created through a graphical pattern editor aided with a simple graph layout program (North et al. 1994). The pattern editor is a simple graph editor, which constrains the node labels and link labels according to the domain ontology.

To display the results, LAW utilizes a dual view for each of the hypotheses. On the left hand side is a table putting the pattern nodes and the data instances in correspondence as well as the links. It also shows for each of them the degree of matching. On the right-hand side is a reduced version of the pattern, which preserves the shape of the graph of the original pattern. One example is shown on Figure 2.

As we have seen in requirement 3, the number of matches can sometimes be high and such a display would become intractable. That is why the user has the option to get a summarized view of the matches. This view can be sorted or grouped according to a particular node in the pattern as shown in Figure 3.

Approaches explored

As it is clear from the previous section, the requirements introduced in this paper have not all been fulfilled in the current state of the LAW system. The main reason was that, in its first phase, the emphasis was put on the capabilities to autonomously return significant and correct matches, with an acceptable performance. Now that this goal has been achieved, it is time to focus on the above requirements, in order to improve the user returns.

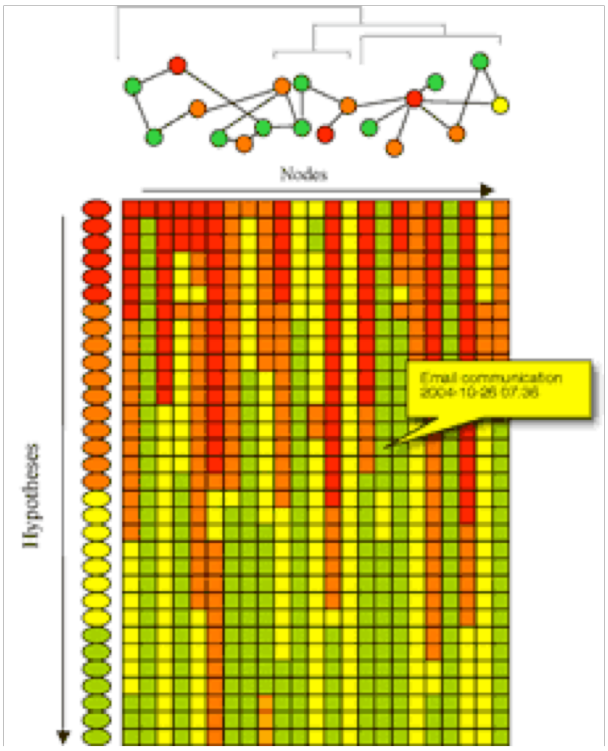

Figure 4: Compact Array of Hypotheses

Here are the different approaches we are envisioning.

Condensed Arrays of Hypotheses

This approach intends to be an improvement over the existing summarized presentation of match results, when the number of matches is extremely large. The idea is to represent the hypotheses as an array of simple colored boxes in which each row is a hypothesis and each column is a node of the pattern. Each box is colored according to

the degree of matching. Above the array is a simplified version of the pattern, with the hierarchy of sub patterns evocated. An example is shown in Figure 4. On the side of the array is represented the match score of the hypothesis.
Every one of these graphical elements is clickable, giving the user more detailed information. For instance, as shown in the figure, if the user clicks on one box, a pop-up window appears, displaying the information on the particular instance, and the corresponding node is highlighted in the pattern. Conversely, if the user clicks on one of the ovals on the left hand side, the overall color scheme of the pattern will change showing notably how well the nodes and links were matched, along with the constraints.

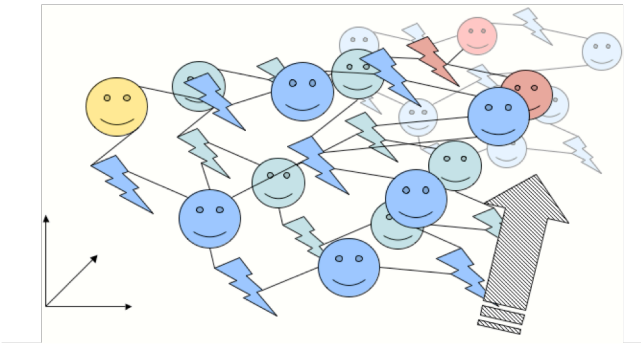

Figure 5: 3D Graph Representation of Data

Visualizing the matching process

This approach intends to respond to requirement 4. The idea here is to use the anytime aspect of the matching algorithm, in the sense that even if the user has not stopped the matching process, the system can do that in parallel to show temporary results. The user will be shown in a continuously updated display how much of the pattern has been explored, what the best partial matches are so far and which branches are currently being explored. This could be displayed both on the pattern, as an animation of the nodes and links, and in the same way the match results are displayed. Naturally, at the time of the display, the matcher can be engaged in a wrong track. Such a display will nonetheless give precious information to the analyst, analog to a chess program showing its line of reasoning during a game.

Data graph mining in 3D

To fulfill requirement 5, the user has to be able to navigate through the data, taking advantage of its relational nature. The approach here is to display a regional (with respect to the node being examined) sub graph of the data as a 3D representation, with the apparent virtual distance approximately figuring the graph distance. The benefits of a 3D representation are multiple:

- It makes the layout of binary graphs possible without intersecting arcs,
- It allows to confer the notion of *point of view*, important since only an extremely partial view of the data is possible at any time, (this point is not clear)
- The notion of exploration can be rendered dynamically as a motion within the graph, with the nodes explored behind your current point of view,
- In some cases one of the dimensions can naturally represent a characteristic of the domain, the most obvious being time, in the context of streaming data.

A simplistic view of what this rendering could look like is presented in Figure 5. Nodes would be represented as icons[1] linked by their relationships. The colors would figure the degree of certainty attached to the data information.

Template-driven Hypothesis Visualization

This proposition aims at responding to the requirement 6, i.e. the ability to present the results to others. In this approach, each pattern has associated with it one or more visualization templates, which describe how to organize a hypothesis presentation and how to display each of its individual elements. Our display language extends PHERL (Murray et al., 2005), which is an emerging standard for representing hypotheses and patterns within the link analysis community. The display language associates display directives with PHERL elements, from high-level (pattern-level) directives that determine overall layout, to mid-level (sub pattern-level) directives that impose structure on the display, to low-level (entity-level) directives that describe placement and display of individual elements of data.
The language includes:

- High-level display directives, associated with the entire pattern, that dictate the display layout for the hypothesis.
- Mid-level display directives, associated with sub patterns within the high-level pattern, that provide organization on elements within the display.
- Low-level display directives, associated with individual pattern elements, that dictate where and how those elements are displayed.
- Methods for switching between multiple element display primitives.
- Methods for highlighting missing information. One of the key functions that many link analysis tools perform is to match patterns even when some of the elements of the pattern may be missing. These methods may also be used to highlight the elements of an evolving hypothesis

[1] Such a solution works best when the domain ontologies are not excessively rich and deep.

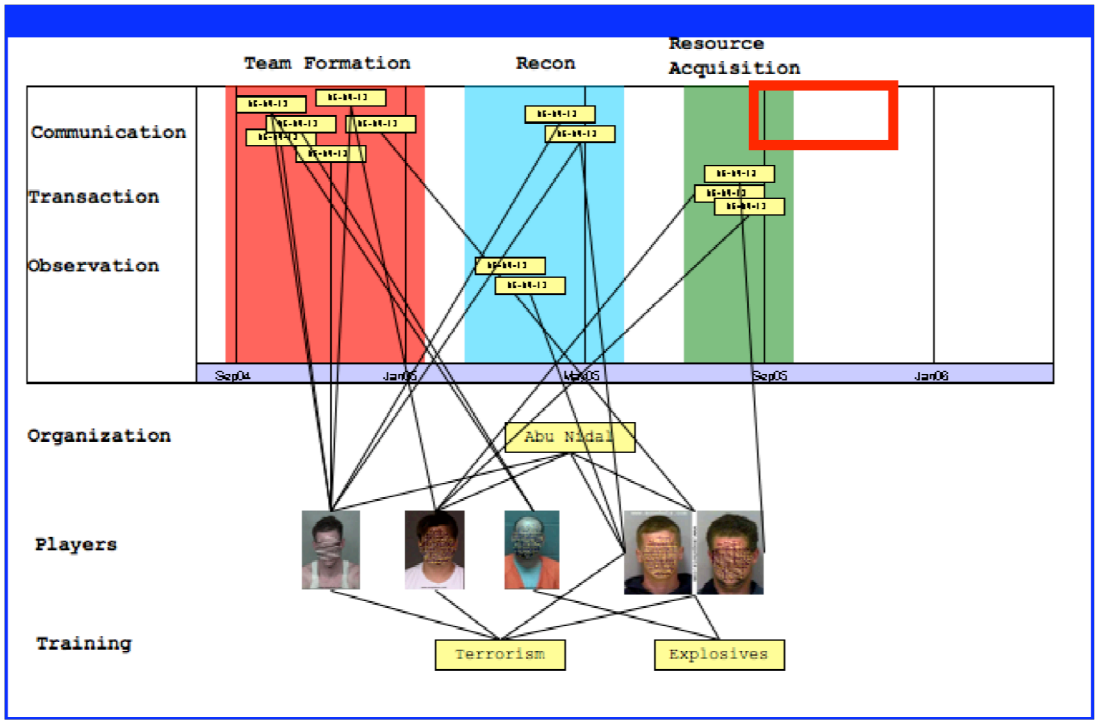

Figure 6: Template-driven Hypothesis Visualization

that have not been seen yet but are the next elements expected to come in.

- Methods for displaying provenance, i.e., for associating hypothesis elements with the evidence for to support them. This evidence may take the form of a document, or it may be a more complex structured argument from another analysis tool.

Figure 6 shows an example of one possible layout for a hypothesis about an evolving terrorist plan. The high-level display directives break the display into a timeline (upper half) for the events in the plan, and a combined graph/table layout (lower half) for information about the plan's participants. Both the timeline and the graph are organized vertically by type of entity (Communication, Transaction, Organization, etc.). Directives attached to the Communication, Transaction, and Observation events place those events horizontally according to when they occurred, and other directives specify that they are displayed as rectangular nodes and described by text denotations of the date on which they occurred. Display directives associated with the subpatterns in the hypothesis provide additional structure to the timeline, color-coding the intervals associated with the Team Formation, Recon, and Resource Acquisition subpatterns. Directives also highlight the information within the hypothesis—the pattern expects communication between the team members in the latter portion of the Resource Acquisition phase—with a red box.

Conclusion

The list of approaches presented in this paper is definitely not exhaustive but we believe it will contribute to making the use of LAW more successful in term of accuracy and efficiency. On the other hand, it is not certain that all of them will be implemented completely in the near future. Ultimately this will all depend on the feedback of users in real life experiments with the end-to-end system.

References

Wolverton, M. and Berry, P. and Harrison, I. and Lowrance, J. and Morley, D. and Rodriguez, A. and Ruspini, E. and Thomere, J. (2003) *LAW: A Workbench for Approximate Pattern Matching in Relational Data*, in The Fifteenth Innovative Applications of Artificial Intelligence Conference (IAAI-03)
– http://www.ai.sri.com/pubs/files/931.pdf

Murray, K., Harrison, I., Lowrance, J., Rodriguez, A., Thomere, J., Wolverton, M. (2005) *PHERL: an Emerging Representation Language for Patterns, Hypotheses, and Evidence*, in Proceedings of the AAAI Workshop on Link Analysis
-http://www.ai.sri.com/pubs/files/1145.pdf

Stephen C. North and Eleftherios Koutsofios. (1994) *Application of graph visualization*. In GI'94 Graphics Interface, pages 235–245, Banff, Alberta, Canada
-http://citeseer.nj.nec.com/221206.html

Thomere, J. and Harrison I. and Lowrance J. and Rodriguez A. and Ruspini E. and Wolverton M. (2004) *Helping Intelligence Analysts Detect Threats in Overflowing, Changing and Incomplete Information*, in Proceedings of the 2004 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety pp. 39-45
- http://www.ai.sri.com/pubs/files/1042.pdf

Wolverton, M. and Harrison, I. and Lowrance, J. and Rodriguez, A. and Thomere, J (2005) *Supporting the Pattern Development Cycle in Intelligence Gathering*, in Proceedings of the International Conference on Intelligence Analysis (IA'05)
- http://www.ai.sri.com/pub_list/1110