Architecture for Machine Learning Techniques to Enable Augmented Cognition in the Context of Decision Support Systems

David Martinez

Lincoln Laboratory, Massachusetts Institute of Technology
244 Wood St., Lexington, MA 02420-9185 U.S.A.
DMartinez@LL.MIT.EDU

Abstract. For a wide range of applications, one of the key challenges is to identify an architecture that is suitable for machine learning techniques to enable important augmented cognition capabilities in the context of complex decision support systems. This overview paper presents an architecture framework. The elements of the architecture are described starting with data formatting, a machine learning algorithm taxonomy, components of courses of action, resource management, and finally the role of augmented cognition within the architecture. The paper includes one cyber security example where the architecture framework is employed. The paper concludes with future work in the development of a recommender system.

Keywords: Machine learning, decision support systems, human-machine interfaces, recommender system.

1 Introduction

Many applications of interest to both military and commercial uses rely on a vast amount of massive data at the input. The input data can be in the form of physical sensors, documents, email, video, images, etc. The users are then tasked to make decisions in a timely manner. The field of augmented cognition facilitates reaching insight after a significant amount of processing is done in the front-end of the decision support system. As shown in Fig. 1, in this paper we are interested in the steps shown after any front-end processing. In a decision support system, these steps can be characterized as information manager, knowledge manager, and decision assistant. As we progress from the information manager to the decision assistant, the role of augmented cognition becomes more critical.

Often the user is either lacking enough data (sparse data) or overloaded with data. The decision support system objective is to drive, via a human-machine interaction, to the shortest decision time with the right amount of data volume. Thus, the goals of the decision support system can be characterized as:

D.D. Schmorrow and C.M. Fidopiastis (Eds.): AC 2014, LNAI 8534, pp. 148–156, 2014.

- Driving data collection when enough information and knowledge are lacking
- Support discovery and association when data is abundant (i.e., overloaded with data)
- Developing human cognition models to support machine learning
- Assisting in developing decision options

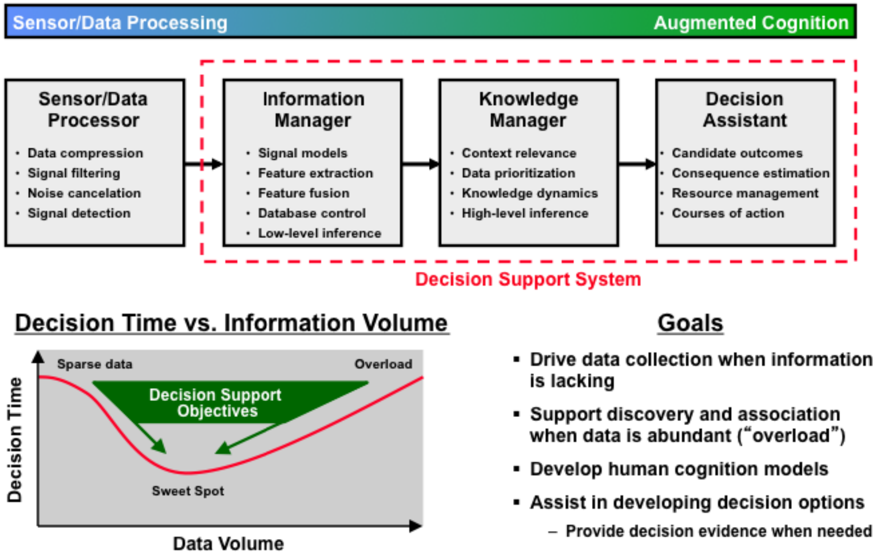

Fig. 1. Decision support system consisting of information manager, knowledge manager, and decision assistant

In the next section, we formulate a decision support architecture containing the key elements of a decision support system consistent with the processing flow shown in Fig. 1. Following a description of the architecture, a cyber botnet example is used to illustrate a representative example of addressing the challenge of data overload. The paper concludes with a look to a recommender system to drive the decision support system to the shortest decision time with the appropriate amount of data volume.

2 Architecture Framework

The decision support architecture shown in Fig. 2 enables a clear formulation of the most important components used in many applications where the interplay between the human and the machine is paramount to reach timely decisions. Many developments discussed in the literature present piece parts solutions. The advent of many sources of data, such as sensors, sources, and events, is causing many users to be uncertain on how to reduce the information entropy to reach actionable decisions. As illustrated in Fig. 2, the user should first identify the needs expected from the decision support system. Similarly, an effective decision support system must be evaluated by a set of measures of performance and measures of effectiveness. We referred to both of these elements in the architecture as the “bookends.” Without a clear approach for the user analysis and performance analysis it is very hard to know how well the overall architecture is working for the user.

At the input to the decision support system are sensor data, sources, and/or effects. Examples of sensor data are inputs from, for example, electro-optical cameras mounted on the side of an airborne platform. Sensor data are also often referred to as structured data, since the input is characterized and described together with metadata containing details about the sensor. Sources are typically representative of unstructured data, for example, written reports from patrols after returning from a military sortie. This type of data is becoming more prevalent in today’s military conflicts, and it provides very valuable complementary information to the sensor data. Another important input category to a decision support system is effects. Examples of effects are human-perceived intents, patterns of behavior, and locations of critical facilities. Effects are also inputs characterized as unstructured data.

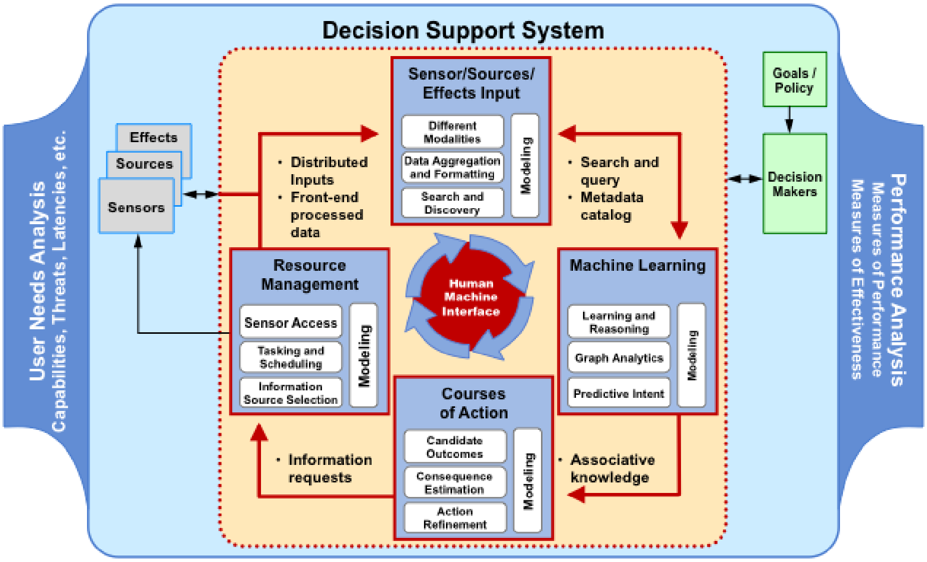

Fig. 2. Decision support system functional architecture

The next important building block of the functional architecture is the sensors/sources/effects data formatting. This is where multiple forms of data inputs are brought together and formatted so that easier data searching and retrieval can be accomplished. It is more common than not to find input data available in a distributed form across many data centers. So this subsystem, within the functional architecture, needs to attend to the distributed form of the available data.

Today, many commercial organizations are developing capabilities to expediently search and discover information by way of machine-to-machine interactions. Commercial organizations like Google, Yahoo, and Amazon, for example, provide their services by efficient use of predefined data aggregation and formatting, and under very efficient algorithms. For example, neural networks, Bayesian belief networks, and hypothesis testing are all examples used in data searching and discovery. The different modalities stage captures the differences between structured data (e.g., sensors) and unstructured data (text, speech, images, etc.). The next important building block is machine learning analytics shown in Fig. 3. This stage is where the decision support system relies on learning and reasoning, inference processing, and hypothesis testing (e.g., in formulating predictive intent) as a way to transform the inputs from information into knowledge, shown in Fig. 1. The results of machine learning feed back to the sensor/sources/effects data-formatting block to aggregate information and, again, to store the results of machine learning processing in the same format as other data to permit later discovery, for example, to perform forensics and data retrieval.

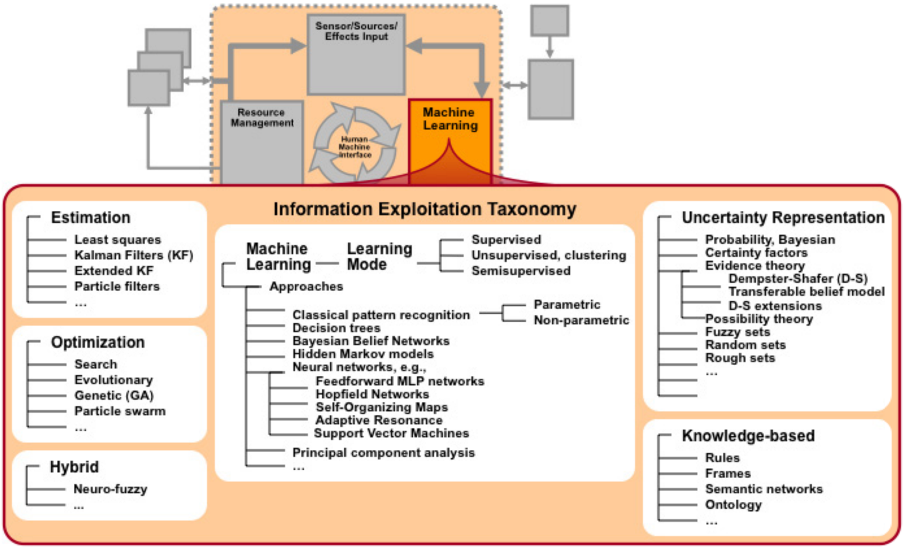

Fig. 3. Representative classes of machine learning techniques [1]

The combination of sensors/sources/effects data management together with machine learning leads to associative knowledge. The associative knowledge is the requisite input needed to identify candidate courses of actions, and this is the next important building block in the decision support system architecture. Courses of action are presented in the form of alternatives to choose from, and, oftentimes, these are characterized under probabilistic values (e.g., the system estimates that the presence of a military threat exists with a probability of 90%). Similarly, this functional block must address consequence estimation, also in the form of probabilistic measures, leading to action refinement.

If the decision maker deems that not enough information is available to make a reliable and accurate decision, information requests are generated as inputs to a resource management stage. The resource management stage is also a complex operation requiring machine learning support. Many military and nonmilitary scenarios involve a complex array of data, and a computer system is more apt to manage and task these providers of data in an efficient manner than the human.

The output from resource management is the input to requesting more sensor data, source of information, and/or effects. The process then proceeds as previously described. A decision support system is not static; additional information and processing are critically important to continue updating the information needed by the decision maker, hopefully leading to a reduction in information uncertainty and increasing the value of the recommended courses of action (decisions). [2] Ultimately, the objective is to achieve the goals shown in Fig. 1.

Each functional block shown in Fig. 2 includes a modeling and simulation subcomponent. This subcomponent simply reinforces the need to generate data, algorithms, courses of actions, or resource management via models and simulation when the real data are not available. For example, the decision support system can be used to model multiple sources of data, algorithms can be evaluated against these data, and courses of actions can also be modeled to better understand consequence estimation. Similarly, resource management can be exercised all in the form of modeled sensors, sources, or effects.

2.1 The Role of Human-Machine Interface (HMI) in a Decision Support System

The HMI function is a very important component of the architecture. It is the media by which the user interfaces with the decision support system. Recent advances in the HMI area are enabling significant improvements in the overall system. [3] Immersive analytics, including collaborations among analysts, are best enabled by advances in augmented cognition, cognitive modeling, and simulations. Several of these technologies are gaining rapid acceptance in many other applications (e.g., game industry).

Each functional component shown in Fig. 2 depends on the HMI effectively interacting across the overall system. It is very common for analysts to make the best decisions when they are presented with data after these have gone through critical stages of machine learning processing (e.g., composable analytics). The decision support system, at the HMI level, can then pose options for courses of action back to the analysts for them to consider.

Fig. 4 depicts an analyst interacting with a broad range of inputs. The stage of composable analytics allows for rapid insertion of machine learning analytics meeting a predefined workflow. These analytics should be able to be changed on-demand and continue to update information to the analysts via both geospatial and graph views.

Another option the analysts have, as they immerse themselves in the data flow within a decision support system, is to perform data exploration. The last section in the paper discusses a recommender system suitable for data exploration.

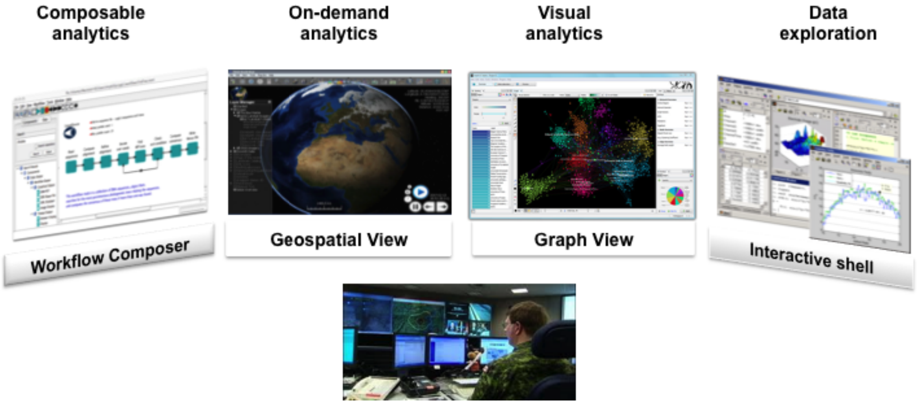

Fig. 4. Example of human-machine interface to achieve timely decisions

3 Anomaly Detection via Graph Analytics

It is useful to cement the architecture framework presented earlier via an example. We illustrate the case of finding a cyber anomaly by inputting a large number of proxy logs (over four million logs; 4M). The scenario consists of a group of source computers, represented by IP (internet protocol) addresses. These computers communicate to a set of web server IPs captured in the traffic proxy logs. The number of source computers is over four thousand (4K). The number of web server IPs is over 16K. The challenge here is to find one infected source computer representing 16K event logs out of the total 4M proxy logs. This is a very good example of large data volume with an anomaly buried in 0.4% of the infected proxy logs traffic (16K/4M)

Following the architecture illustrated in Fig. 2, an interconnected graph containing all the source computer IPs and the web server IPs is formed. [4] This step is represented as the sensors/sources/effects data-conditioning block. It also includes a model of normalcy, referred to as the graph model construction in the graph residual construction step shown in Fig. 5 and formulated as E[G] (graph estimate). These outputs are then operated on using a class of machine learning technique commonly referred to as principal component analysis (PCA). The PCA step permits dimensionality reduction by identifying the top two most relevant singular vectors. The last stage is the anomaly detection step (or signal detection). The two techniques found to be most effective were Chi-Squared and K-Means statistical clustering to separate the anomaly from the rest of the uninfected data.

The output of the anomaly detector is the set of infected traffic proxy logs between the one IP (single computer) and the malicious web server. This output is characterized as the associative knowledge (knowledge of infected IP) in Fig. 2. Of course, this example, used for illustration purposes of a decision support system, is a very simple case but very difficult to accomplish if not done with the help of PCA analytics. However, in many applications one is confronted with many outputs as data is streaming through the decision support system in real time. Therefore, it is imperative that the decision support system continues to operate on streaming data while the decision support system is tasked with providing candidate courses of action to the user. Future work in this area will leverage a recommender system discussed in the next section.

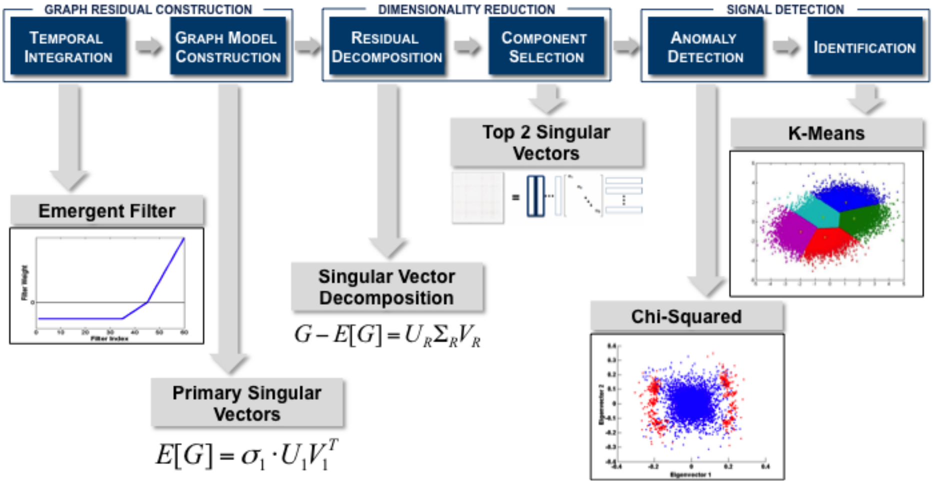

Fig. 5. Anomaly detection of a malware botnet using principal component analysis

4 Future Work

The previous sections described an architecture identifying the critical functional blocks in a decision support system. An example was used to demonstrate the value of the architecture for a cyber anomaly detection application. This section expands the discussion to include a type of capability to enable rapid decisions leading to recommending courses of actions. A recommender system is of interest since, in many cases, the analysts are presented with an abundant set of options potentially resulting in false positives and/or false negatives. Therefore, an approach to minimize the entropy in the consequence estimation function shown in Fig. 2, within the courses of action functional block, is to employ a recommender system as illustrated in Fig. 6.

The anomaly detection block shown in Fig. 6 encapsulates the steps shown in Fig. 5. Since the example described earlier operates on uncued data, there is a need to further reduce the options presented to the analysts. One approach is to take the outputs of the anomaly detector and identify contextual relevance. This step is referred to in Fig. 6 as the embedding & filtering function. This step, within a recommender system, identifies a contextual subspace that allows for interest prediction. One machine learning analytics applicable for this step is random subspace projection. [5]

The next stage is active learning. This step permits the analysts to interact with the system to separate the most relevant information. The value in performing this step is to reduce false positives and/or false negatives. The last steps in the processing flow shown in Fig. 6 are the visual-analytics interface and the reinforcement learning. These steps permit inputs from a group of analysts to refine the courses of action options (action refinement) as new experiences by the users change their relevance model shown in Fig. 2.

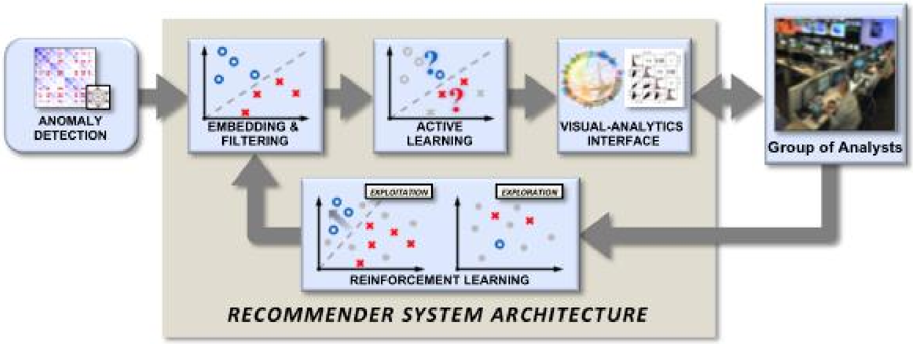

Fig. 6. Candidate recommender system to facilitate timely courses of action

The development of the recommender system shown in Fig. 6 is an area of future research applicable to a broad range of applications, including the cyber anomaly detection described in the previous section. Such an approach will incorporate multiple disciplines in data aggregation, machine learning techniques, augmented cognition models, and probabilistic estimates in reaching the shortest decision time within the courses of action function of a decision support system.

Acknowledgments. This work was sponsored by the Assistant Secretary of Defense for Research and Engineering under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government. The author also wishes to thank many of the technical staff members of the Computing and Analytics Group at MIT Lincoln Laboratory who are working on the material described in Section 3, and who have been part of many discussions with the author in formulating the future work described in Section 4. Mr. Brad Dillman and the Technical Communications personnel from MIT Lincoln Laboratory contributed to the formatting of the paper.

References

1. Braun, J., Glina, Y.: Hybrid Methods for Multisource Information Fusion and Decision Support. In: Proc. SPIE, vol. 6242 (2006)
2. Chryssanthacopoulos, J.P., Kochenderfer, M.J.: Decomposition Methods for Optimized Collision Avoidance with Multiple Threats. Journal of Guidance, Control, and Dynamics 35(2), 398–405 (2012)
3. Crouser, J.R., Ottley, A., Chang, R.: Balancing Human and Machine Contributions in Human Computation Systems. In: Human Computation Handbook. Springer (2013)
4. Miller, B.A., Arcolano, N., Bliss, N.T.: Efficient Anomaly Detection in Dynamic, Attributed Graphs. In: IEEE International Conference on Intelligence and Security Informatics, Seattle, WA (June 2013)
5. Blum, A.: Random Projection, Margins, Kernels, and Feature-Selection. In: Saunders, C., Grobelnik, M., Gunn, S., Shawe-Taylor, J. (eds.) SLSFS 2005. LNCS, vol. 3940, pp. 52–68. Springer, Heidelberg (2006)