

Establishing a Data-Mining Environment for Wartime Event Prediction with an Object-Oriented Command and Control Database

Marion G. Ceruti

SSC San Diego

S. Joe McCarthy

Space and Naval Warfare Systems Command

INTRODUCTION

The ability to predict attacks and other hostile events during times of conflict is important to military commanders from the standpoint of readiness. The more advanced the notice and the more widespread the notification, the better able all echelons are to respond to threats efficiently and with the correct combination of forces.

The literature is replete with recent research results on data mining and data classification. (See, for example, [1, 2, 3, and 4].) Data mining, data classification, and data correlation are related to data fusion. As these techniques mature, better tools become available to model and to correlate data from complex operational scenarios. The purpose of this research is to create and extend a method to predict attacks on the U.S. Marine Corps using an object-oriented command and control database and data-mining techniques [5].

Data Mining

Data mining is the search for and extraction of hidden and useful patterns, structures, and trends in large, multidimensional, and heterogeneous data sets that were collected originally for another purpose. (See, for example, [4].) Data mining is an art that is supported by a considerable body of science, engineering, and technology. For example, data mining uses techniques from such diverse areas as data management, statistics, artificial intelligence, machine learning, pattern recognition, data visualization, and parallel and distributed computing. Data mining is possible today because of advances in these many fields; however, this multidisciplinary characteristic also makes data mining a difficult subject to teach and learn. Whereas the Structured Query Language (SQL) is inadequate to answer many complex queries, data mining can support searches for patterns in temporal and spatial databases in a more efficient manner. Data mining is important to the military because commanders and the analysts who support them cannot anticipate all future uses of information at the time of data collection.

Limitations of Data Mining

Whereas the goal of data mining is to identify hidden patterns, the search algorithms chosen for the particular task may miss an important and interesting pattern or even a class of similar patterns. A systematic method to preclude this problem is not available.

ABSTRACT

This paper documents progress to date on a research project, the goal of which is wartime event prediction. The paper describes the operational concept, the data-mining environment, and the data-mining techniques that use Bayesian networks for classification. Key steps in the research plan are (1) implement machine learning, (2) test the trained networks, and (3) use the technique to support a battlefield commander by predicting enemy attacks. Data for training and testing the technique can be extracted from the object-oriented database that supports the Integrated Marine Multi-Agent Command and Control System (IMMACCS). The class structure in the IMMACCS data model is especially well suited to support attack classification.

Similarly, there is no guarantee that any given data-mining effort will yield something new and useful, regardless of how many well-designed data-mining tools are used. This is because the data may not contain the desired patterns. Data mining is a search for observational data and the relationships between them, rather than the measurement of experimental data.

CONCEPT OF OPERATIONS

The concept of operations for a future system based on this research is (1) to use data-mining and data-classification algorithms to detect patterns associated with attacks (e.g., to identify factors that indicate an imminent attack) and (2) to correlate these patterns with current events with a view toward supplying military commanders with a prediction of the next attack and a confidence level that pertains to that prediction. A considerable amount of data associated with events that have preceded known attacks is required to model attacks, to search for common features, and to find these patterns in new data.

Success in this effort depends on a characterization of the circumstances that translate to well-defined observables that preceded past attacks. The more detailed the available knowledge, the better the resulting model, and the greater the probability that data instantiating critical variables can be collected. We expect that such detailed data for all variables will not be available prior to future attacks and that all available data may not be useful in predicting attacks (i.e., will function as "noise" in the analysis). Thus, the task involves identification of algorithms that can detect pre-attack features in clutter and the use of pattern recognition. Modern methods of statistical pattern recognition are sufficiently computationally oriented to use a larger dimensional space and are less sensitive to noise than older methods. Success in attack prediction will depend, at least in part, on how well these methods can be implemented with the available data.

GENERAL APPROACH

Hostile events can be characterized with respect to as many relevant variables as are deemed necessary and available to predict future attacks. An object-oriented message-traffic database can be analyzed for the occurrence of telltale signs of pending attacks. Our objective is to generate an event prediction (in terms of a probability) with a confidence value associated with it. Therefore, it is necessary to determine the combinations of events and observations that will have a higher probability of indicating a future attack. A baseline can be modeled from normal operational scenarios and from military events during times of conflict that do not constitute attacks per se.

The attack alarm-generation process and the reduction of false positives can be approached using constraints from models of known attacks. The identification of the appropriate features (and groups of features) that can flag imminent attacks is the most challenging part of the process. One approach is to explore the generation of a knowledge base encoded in Bayesian networks.

A literature search was conducted for publications on various subjects that relate to data mining, including algorithms and their applications. Data-mining algorithms can be used to identify complex patterns in the

data that correlate well to hostile events. Criteria can be developed for sufficient correlation and confidence levels in data associations. For example, one metric that could be used is correlation strength, which is the ratio of the joint probability to the individual probability of observing a pattern [1].

BAYESIAN NETWORKS

Bayesian networks can be used to classify data into categories. Bayesian networks are:

- probabilistic networks,
- directed acyclic graphs that encode certain dependences between nodes that represent random variables,
- knowledge bases with knowledge in the network's structure and in its conditional probability table, and
- structures that can be used to infer causality.

Naive Bayesian Networks

A naive Bayesian network is a very simple structure in which all random variables representing observable data have a single, common parent node—the class variable. The naive Bayesian classifier has been used extensively for classification because of its simplicity, and because it embodies the strong independence assumption that, given the value of the class, the attributes are independent of each other.

Naive Bayesian networks work remarkably well considering that this independence assumption may not be valid from a logical standpoint. The performance of a naive Bayesian network can be improved with the addition of trees that provide augmenting edges to a naive Bayesian network by representing correlations between the attributes.

Tree Augmented Naive (TAN) Bayesian Classification Algorithm

SSC San Diego has access to SRI International's classifier algorithms developed under the Defense Advanced Research Projects Agency's High Performance Knowledge Base Program. For example, SRI's Tree Augmented Naive (TAN) Bayesian Classification Algorithm is a classifier algorithm based on Bayesian networks with the advantages of robustness and polynomial computational complexity [2 and 3].

Bayesian networks have some drawbacks that SRI has addressed in the TAN algorithm. In ordinary naive Bayesian networks, the variables (data) are assumed to be conditionally independent given the class. Logically, this is not always true. For example, suppose enemy troops are observed at location X and enemy tanks are observed at location Y. When using naive Bayesian networks, one assumes that these events are independent. However, both events may be part of the overall enemy battle plan. In the TAN algorithm, the trees provide edges that represent correlation between the variables.

Bayesian networks, especially with tree augmentation, are a suitable technology for data-mining classification and event prediction for the following reasons:

- First, one need not provide all joint probability values to specify a probability distribution for collections of independent variables [6].
- Second, one could mix modeling (e.g., explicit knowledge engineering for knowledge elicited from experts) with statistical data induction and

adaptivity. This mix would require fewer data values to induce better quality models.

- Third, one could use these models to compute the value of information. For example, having seen signs "A" and "B" of an imminent attack, what is the best information to collect next to confirm that hypothesis?
- Fourth, one could characterize explicitly the kinds of attacks. For example, given an attack of type "air attack," what are the most likely signals? These signals could be collected regularly to fill the database used as input into the TAN algorithm.

The TAN algorithm makes some tradeoffs between accuracy and computation. It approximates a probability distribution using some constraints on the complexity of the representation; however, it is extremely fast (low polynomial), efficient (one pass over the data), and robust (low-order statistics).

The TAN algorithm accepts data sets as input and induces Bayesian networks as output. Specifically, the TAN algorithm is intended to be used as a classification algorithm, which means that the input would be a file with tuples of the form $\{x_1, x_2, x_3, \dots, x_n, c\}$ where the x_i are values that variable X_i takes and c is the value that a class (C) variable can take. To set the range of each variable, the TAN algorithm needs an auxiliary file that contains a description of each variable, including the range of values representing the degree of intensity.

The TAN algorithm's output is a Bayesian network encoding of $P(C, X_n, \dots, X_1)$ in an efficient manner. To use TAN as a classifier, one simply computes $P(C|x'_n, \dots, x'_1)$. Given a new vector X'_n, \dots, X'_1 and having a probability distribution over c , one can select the event with highest probability as the one to classify. To compute the confidence in this value, the bootstrap method can be used [7].

The TAN algorithm outperforms naive Bayesian networks while maintaining its robustness and computational simplicity (polynomial vs. exponential complexity).

The TAN algorithm captures the best of both discrete and continuous attributes. Therefore, the TAN algorithm achieves classification performance that is at least as good as, and in some cases better than, models that use purely discrete or purely continuous variables. Studies at SRI have demonstrated that the TAN algorithm performs competitively with other state-of-the-art methods.

TAN, and similar algorithms, can be made to perform the classification of certain battlefield situations for the Marine Corps. Much work needs to be done in this area, particularly with regard to data-set selection, data cleansing, and the refinement of the algorithm to meet specific needs.

In addition to the TAN algorithm, SRI has more general algorithms for inducing Bayesian networks that do not make the compromises that the TAN algorithm does. These algorithms try to fit the best distribution possible with no constraints. The disadvantage is that the computation of these models is slower; however, this may be acceptable and desirable in some cases. Algorithms can be implemented with the same data and the results compared.

GaussMeasurePredict Program

The GaussMeasurePredict program was developed by Nir Friedman to measure the performance of an induced TAN model. (See, for example, [2]).

The input of GaussMeasurePredict consists of the following items: (1) an induced Naive Bayesian network from TAN, (2) the name of the variable to predict, and (3) a test data set that contains instance information.

When testing the Bayesian network model, the variable to predict is specified and known to be correct. Usually this will be the outcome of the class variable.

GaussMeasurePredict also has the option to calculate and display the probability of each class value for each instance in the input file. This feature is particularly useful for receiver operating characteristic (ROC) curves as well as for determining other statistics [8]. Thus, with this option, GaussMeasurePredict can output the probability distribution for each instance in addition to a summary.

The output of GaussMeasurePredict is a prediction of the accuracy of the network in the TAN Bayesian network .bn file. It can be used to predict the accuracy of other classifier algorithms as long as the output file matches the format of TAN's Bayesian network file.

GaussMeasurePredict is intended to be used to measure the accuracy of predictions and not to generate predictions for unlabeled instances. Unlike the TAN algorithm, GaussMeasurePredict does not accept instances with "?" for missing values in an instance input file. All variables must have filled values in each instance. However, because GaussMeasurePredict compares the induced Bayesian network to the test data set, it also can be used to infer the class of an unknown instance by filling in the class (Outcome) variable with a guessed value. Using the option described above, GaussMeasurePredict can output a predicted class probability for each class value. The class with the highest probability is the predicted class for that instance.

Fortunately, in the simplest case of attack predictions, only two values are possible for the class variable: `ATTACK_LIKELY` and `ATTACK_NOT_LIKELY`. In more detailed cases of attack predictions in which specific attack types are listed in the data-definition input file, the class variables may assume $2N$ values where N is the total number of attack types considered in the class. (The $2N$ arises from including the negation of the likelihood of an attack of each type.)

SOFTWARE IMPLEMENTATION AND PLANS

Data-mining software was tested for correct operation with clean data sets designed specifically for testing. The programs described below are included in the research environment. The software includes the TAN algorithm and the GaussMeasurePredict that uses the output of the TAN algorithm. Inputs to GaussMeasurePredict must be complete. Plans include the acquisition of additional algorithms that are designed to operate on incomplete data sets.

TAN 2.1 Availability

The TAN version 2.1 software and user's manual are available for download via file transfer protocol (FTP) from SRI's Web site: <http://edi.erg.sri.com/tan/TANintro.htm>. The user is required to register with a name and password. To obtain the TAN algorithm, Netscape is recommended and may be required. The Solaris CDE Web browser, HotJava, is not recommended to download TAN. The TAN user manual is included with the software (See, for example, [8]).

The TAN software was downloaded from SRI's Web site onto a Solaris SPARC Station 20 computer running the Solaris 2.7 UNIX operating system and using the Common Desktop Environment (CDE).

TAN 2.1 constitutes the main data-mining tool in the research environment of this project. TAN can be used as a base classifier and also as a method to fuse the output of other data-mining and classification algorithms. When algorithms have been tested and programmed, data visualization tools can be identified, tested, and used to view the data and to continue the pattern-recognition process.

GaussMeasurePredict Availability

The GaussMeasurePredict program is available along with the TAN software from SRI's Web site. The program is included with the TAN package and can be executed when files are "unzipped" and when the appropriate input files are available.

OBJECT-ORIENTED DATA IMPLEMENTATION

The object model, on which the Integrated Marine Multi-Agent Command and Control System (IMMACCS) database is based, is a detailed representation of the battlespace with objects derived from the March 1998 Urban Warrior Advanced Warfighting Exercise [9 and 10]. Object attributes and their associations, as well as class inheritance, are also described in [10]. The IMMACCS database uses the Unified Modeling Language symbolic representation method [10].

The IMMACCS database includes in its structure the following topics of interest to the Marine Corps: aircraft; ground vehicles; sea-surface vehicles; weapons and weapon systems; electronic devices of many kinds; terrain; bodies of water; logistics information; transportation infrastructure; various specialized units; personnel data; and most importantly for this application, military events. Class inheritance paths and allowed values are specified [10]. The use of an object-oriented database and the representation of military entities in object form provide a degree of interoperability and extensibility that allows multiple services to use and add to this common tactical picture [9].

The data sets for this data-mining effort will come from IMMACCS. The class structure in the IMMACCS data model is especially well-designed for adaptation to the attack/non-attack classification task. When data fill becomes available, especially for the attributes and object classes of interest, the IMMACCS database will be a very desirable data source for reasons described in the next subsection.

CONSTRUCTION OF TRAINING DATA SETS

The following discussion illustrates the strategy for constructing training data sets using certain IMMACCS object-oriented data classes as examples. The data-mining classification task is to identify the value of the Bayesian-network class variable of an unknown data set. Initially, two Bayesian-network class variables will be considered, "imminent attack likely" or "imminent attack not likely." To train the TAN algorithm, the value of the Bayesian-network class variable will be identified in the training data sets for both classes.

Various types of attacks and defenses are listed as allowed values (among others) in the MILITARY_EVENT object class in the IMMACCS database.

These are AIR_ATTACK, GROUND_ATTACK, AIR_DEFENSE, GROUND_DEFENSE, and SMALL_SCALE_ATTACK. Only instances that correspond to attacks from hostile forces on the Marine Corps will be considered. Any attack launched by the Marine Corps on hostile forces will not be counted in the "attack" category. In contrast, defenses by the Marine Corps against hostile attacks, whether the attacks are launched from the air or the ground, are likely to play a role in the over-all model when they influence subsequent enemy attacks. For example, enemy commanders may select a battle plan that does not involve an air attack on an area with a strong Marine Corps air defense.

Several naive Bayesian networks can be induced, one for each attack type and one for the combined data for all attack types. For the combined attacks, the class variable can take multiple values, corresponding to the likelihood of a particular attack type and the likelihood that this attack type will NOT occur. Initially, all attack types will be assumed to be independent, although this is rarely true in actual battles. For example, ground attacks are more likely to follow air attacks at the same location than vice versa.

For the non-attack training instances, data associated with the other values of the MILITARY_EVENT object class will be used, such as WITHDRAWAL_EVENT, DELAYING_ACTION, AIR_REINFORCEMENT, or DRILL_EVENT. Other non-attack training instances also can be derived, for example, from the AIR_DEFENSE and GROUND_DEFENSE values, provided the instances pertain to events associated with enemy air defenses and ground defenses.

The date-time groups (DTGs) associated with each instance, both of attack and non-attack situations, will be noted and other data objects with the same DTGs (and with DTGs just prior to the event) will be included in the training data sets. The training data also could include objects present in the same vicinity as the attack or non-attack event that do not have DTGs. This will provide as comprehensive a description of the battlespace at the time and place of the attack as is possible, given the level of data granularity. This method of formulating training data sets can be extended by including in each data set the data that pertain to DTGs several days prior to the event to ascertain whether this will yield better results. The exact time span that each data set should cover is an open research issue.

Design Considerations in the Construction of Test Data Sets

Changes can be made in the test data sets, depending on the desired outcome of the test. For example, to determine how far in advance an attack can be predicted, the instances that pertain to an entire day immediately prior to the attack can be omitted systematically from test data sets. If the algorithm still makes the correct prediction, one can conclude, at least as far as that test data set is concerned, that an attack can be predicted 24 hours in advance. Similarly, if 2-days worth of data immediately preceding the attack can be omitted without a significant decline in the prediction accuracy, this is an indication that attacks can be predicted 48 hours in advance.

We expect, however, that omitting more and more data that pertain to the days just prior to an attack will cause the attack-prediction accuracy to degrade. The exact functionality of this degradation (linear, exponential,

logarithmic, etc.) is another open research question. This type of testing can enable researchers to determine the number of days to include in the data collection and the specific data elements to be collected necessary to formulate as accurate a prediction as possible.

Test and training data sets will be formulated according to an n-fold cross-validation procedure. For example, to implement the first cycle of a five-fold cross validation with a data set consisting of 1,000 records, the first 800 records can be selected for training, with the last 200 records being reserved for testing. During the second phase of training and testing, the first 600 records and the last 200 records together will comprise the test data set, and the remaining records will be used for testing. In the third phase, the first and last 400 records will be used for training and the middle 200 for testing, etc. The advantage of this procedure is that it can be used to identify anomalies in the testing and training so that if the results are comparable for all five tests, a higher level of confidence in the method is obtained.

CONCLUSION

This paper describes a data-mining environment designed to support wartime event prediction using Bayesian networks to perform a data-classification task. The TAN algorithm was selected to induce a network using data extracted from an object-oriented database that contains information from exercise message traffic. Future work could include a user-friendly interface designed on top of the algorithms to provide automated input of selected data sets to the algorithm of choice. Success in this research project will pave the way for a more precise indication-and-warning system for the U.S. Marine Corps.

ACKNOWLEDGMENTS

The authors thank SSC San Diego's Science and Technology Initiative and the Defense Advanced Research Projects Agency for their financial support.

REFERENCES

1. Clifton, C. and R. Steinheiser. 1998. "Data Mining on Text," *Proceedings of the 22nd Annual IEEE International Computer Software and Applications Conference, COMPSAC'98*, pp. 630–635.
2. Friedman, N., D. Geiger, and M. Goldszmidt. 1997. "Bayesian Network Classifiers," *Machine Learning*, vol. 29, no. 2/3, November/December, pp. 131–163.
3. Friedman, N., M. Goldszmidt, and T. J. Lee. 1998. "Bayesian Network Classification with Continuous Attributes: Getting the Best of Both Discretization and Parametric Fitting," *Proceedings of the International Conference on Machine Learning '98, ITAD-1632-MS-98-043*.
4. Thuraisingham, B. M. 1999. *Data Mining: Technologies, Techniques, Tools and Trends*, CRC Press, Boca Raton, FL.
5. McCarthy, S. J. and M. G. Ceruti. 1999. "Advanced Data Fusion for Wartime Event Correlation and Prediction," *Proceedings of the 16th Annual AFCEA Federal Database Colloquium and Exposition, AFCEA*, pp. 243–249.
6. Charniak, E. 1991. "Bayesian Networks without Tears," *AI Magazine*, pp. 50–63.



Marion G. Ceruti

Ph.D. in Chemistry, University of California at Los Angeles, 1979

Current Research: Information systems analysis, including database and knowledge-base systems, artificial intelligence, data mining, cognitive reasoning, software scheduling and real-time systems; chemistry; acoustics.

S. Joe McCarthy

Ph.D. in Solid-State Electronics, University of Washington, 1973

Current Work: Assistant Program Manager for Processing and Analysis, Space and Naval Warfare Systems Command.

7. Friedman, N., M. Goldszmidt, and A. Wyner. 1999. "On the Application of the Bootstrap for Computing Confidence Measures on Features of Induced Bayesian Networks," *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*.
8. Lee, T. J. and M. Goldszmidt. 1998. "TAN Tree Augmented Naive Bayesian Network Classifier Version 2.1 User Manual," <http://edi.erg.sri.com/tan/TANintro.htm>, pp. 1–27.
9. Alderson, S. L. 1999. "Urban Warrior Advanced Warfighting Experiment: Information Dominance in the Battlefield," *Proceedings of the 16th Annual AFCEA Federal Database Colloquium and Exposition*, AFCEA, pp. 213–228.
10. Leighton, R. and J. Pohl. 1998. The IMMACCS Object Model and Database, OBDATA00, November, IOM Version 1.5, Cal Poly, San Luis Obispo, CA.

