ICML-97 Workshop "ML Application in the Real World", Nashville, TN, July 1997 (to appear).

Situation development in a complex real-world domain

Mark Devaney
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
markd@cc.gatech.edu

Ashwin Ram
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
ashwin@cc.gatech.edu

Abstract

Applying techniques from Machine Learning to real-world domains and problems often requires considerable processing of the input data, to both remove noise and to augment the amount and type of information present. We describe our work in the task of situation assessment in the domain of US Army training exercises involving hundreds of agents interacting in real-time over the course of several days. In particular, we describe techniques we have developed to process this data and draw general conclusions on the types of information required in order to apply various Machine Learning algorithms and how this information may be extracted in real-world situations where it is not directly represented.

keywords: situation development, pre-processing

1 Introduction

We are investigating the use of various Machine Learning techniques in support of the task of *situation development* in the domain of real-world "mock" training battles conducted by the US Army. Situation development is a process in which information is collected and integrated into a model which is used for assessing the current situation and predicting future situations. Situation development covers a range of sub-tasks such as recognizing significant events during the course of a battle and predicting the movements, tactics, and ultimate goals of enemy forces during the course of an engagement.

Our ultimate goal is to develop an Artificial Intelligence system which is able to use prior knowledge of mock battles learned through experience along with information on-the-fly about an occurring battle scenario to perform prediction and recognition at several levels. The task is extremely difficult in general, and particularly so in our domain which is further complicated by various aspects of the real-world data such as large amounts of noise, missing information, heterogeneous input types, lack of high-level representations, and sheer volume of information. Bares, et. al. (1994) discuss many of these issues in a similar domain and task.

In fact, before a solution to the situation development problem can be found, there are several difficult problems which must be solved first. These problems center around ways for both condensing and enriching the real-world "raw" input data into forms which can be used by other "higher-level" algorithms in the situation development process. Our initial research has focused on identifying the types of techniques we will eventually be employing and developing algorithms for converting our input data into forms suitable for these algorithms.

Our overall methodology has progressed in a top-down direction. We first hypothesized what a solution to the ultimate task of situation development would involve, then identified techniques that might be required to reach the solution. We quickly found that many of the difficult problems in applying these techniques were caused by the lack of information inherent in our input data. This paper describes some of the Machine Learning techniques we have employed for performing pre-processing of our input data such as temporal clustering and estimation, and presents some of our initial approaches to an overall solution. We also discuss some of the difficulties that very large data sets from the real world present to the development of large-scale systems employing Machine Learning techniques.

2 Real-world input

The domain of our research involves data obtained from "mock" battle training exercises conducted at the US Army's National Training Center (NTC). Training battles are very large scale, typically involving four hundred to one thousand participants and last from one to three days. During the course of the battle, positional and other information is recorded in real-time through the use of sensors located on the battlefield and on the participants themselves. This data is transmitted to a control center where it is recorded and used in evaluation and post-training debriefing. Similar data is expected to be available in real battles as well, hence the interest in developing techniques for situation development.

There are two types of information collected at NTC; static information known in advance about the participants and dynamic information collected during the course of the battle. A great deal of information is recorded before and during the mock battles; however, we limit ourselves to information that could realistically be obtained in an actual battle. The static information we employ is the side of the participant (friendly or enemy) and the echelon to which the participant belongs, i.e., who its parent is in the command hierarchy. In the real world, this type of information would be collected through pre-engagement intelligence such as satellite imagery and long-range sensors.

The other type of information available is dynamically collected during the course of the battle. Again, for training and debriefing purposes, a great deal of detailed information is collected at NTC but we limit ourselves to simple positional reports collected during the course of the battle. This data consists of x,y coordinates of individual units on both sides and is collected at 5 or 10 minute intervals throughout the course of the battle. In a real battle, this information would be collected by a combination of global positioning satellites, sensors dispersed throughout the battlefield and short-range surveillance.

Contrary to our expectations, the data obtained from NTC is extremely noisy and incomplete. The echelon information is almost entirely non-existent with only approximately 10-20% of the units identified as belonging to any higher-level echelon. The telemetry data is also incomplete; positions are only reported for approximately half the total units at any given time step, and many units are "lost" for long periods of time. This presents a difficulty, for it is not evident whether the unit has remained at its prior position or has moved but been lost by the sensors.

Noise in the data takes the form of incorrect positional reports for units in the dynamic data. The static data also contains noise, mostly in the echelon information, where units are identified as belonging to echelons of the opposing side or to echelons at the same level in the echelon hierarchy. Telemetry data is also occasionally reported for units which were not described in the static information, so there is no way of knowing to what side they belong. This degree of noise is to be expected in an actual battle situation, so any algorithms employed must be capable of robustly handling both the missing and incorrect data. Since there is no external knowledge source on which to rely for corrections, we discard any known noisy data, such as incorrect echelon assignments or units not labeled as belonging to a particular side. Figure 1 provides a visualization of the input data with which we work - a snapshot of the units participating in the battle at the initial time step. Units are typically either armored personnel carriers or tanks, and are the lowest-level at which movements are tracked. The data is simply sequences of such snapshots over time.

Figure 1: Battle participants at time t_0

Often, the largest problem faced when using real-world data is not the noisy and incomplete information received, but that the data is not in a form which is readily used by any Machine Learning technique. This problem is particularly acute in the NTC battle domain, where the input essentially consists of position reports for several hundred agents over one to two hundred time steps. This information is simply at too low a level and in too great a quantity to be delivered as input to any clustering or pattern recognition algorithm with any degree of success. Thus, in our initial research we have investigated and implemented techniques to distill the information present in the real-world raw data into a more compact form. Additionally, we have augmented the information present in the raw data by constructing higher-level information structures which are derived entirely from the input data. The following sections describe some of the techniques we have developed to convert our data into more useful forms.

3 Building useful representations

The core problem in situation development is essentially that of recognizing patterns across instances of temporal data, which at a very abstract level is a supervised learning task. Recognizing these patterns simply through low-level telemetry data is an intractable problem, hence the need for constructing representations which capture relevant similarities in an efficient manner. Certain patterns of movement correspond to purposeful strategies and tactics employed during the course of a battle, and as similar strategies and tactics are used across battles, these patterns will recur.

Two similar patterns should be identified as such regardless of where they occur on the battlefield or when in the course of the battle they occur. Thus, one of the essential components of a representation must abstract absolute positions into a relative egocentric format. This can be accomplished in many different ways according to the particular domain being used. In our research this conversion was inspired by an examination of the manuals used to train Army Intelli-

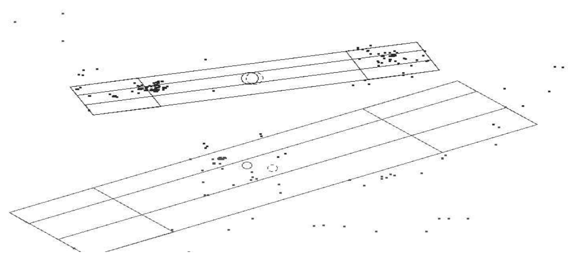

Figure 2: Battlefield layouts at time t_0

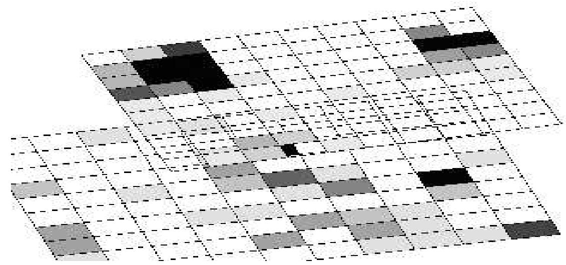

Figure 3: Quadrant representation at time t_0

gence officers.

One of the preliminary steps in performing intelligence assessment of the battlefield is to identify the significant portions of the battlefield, such as the forward and rear areas, flanks, and general orientation of the troops. This allows the position of a unit to be represented in a much more abstract manner (e.g., "left flank rear") than a simple x,y coordinate pair allows. This representation also has the immediate benefit of cutting the space of possible troop dispersions in half, since each side is represented identically. However, there is no standard technique that we are aware of for automatically segmenting the battlefield into its constituent parts. The task is further complicated by the fact that no higher-level information can be used, the description of the battlefield must be determined entirely from the disposition of troops within it.

Our initial heuristic solution to this problem has been to first calculate the center of mass of each side in the battle using the positions of all units on the battlefield. As mentioned above, there is a significant amount of noise in the data which often manifests itself as erroneous positional reports, which tend to skew the layout calculations. We minimize the influence of noise by first calculating a vector through the center of the data for each side, minimizing distance in both x and y directions. A bounding rectangle is then calculated for all units within 1 standard deviation from this line (Euclidean distance), these rectangles are called the "area of operations" for each side. The areas of operation are then divided into forward, rear, and flank areas, defined as percentages of the total bounding rectangle. This approach has proven fairly successful, generating battlefield layouts which domain experts have deemed reasonable. Figure 2 shows the unit data from 1 with the battlefield layouts for each side superimposed.

The layout representation has been useful in representing positions at a very gross level and has also proven useful in describing large-scale movements of units across the battlefield. However, this representation has proven too coarse to capture some of the information necessary to performing situation development. Instead, an intermediate representation was needed, one less general than that provided by the layout algorithm but still ego-centric and more abstract than that of individual positions. For this representation we have divided the bounding rectangle of each side into quadrants, similar to the technique described in Huber & Durfee (1992). The resolution of the quadrants can be dynamically modified; currently this is done by the human user of the system, but eventually we intend for this resolution to be automatically adjusted according to troop dispersal and current hypotheses. Figure 3 shows the unit positions of figure 1 depicted as quadrant concentrations. The shading of the various quadrants indicates the concentration of units within that area.

4 Recovering groupings

The techniques described above are ways of reducing the size of input data by grouping units according to their location in certain areas of the battlefield, essentially increasing the granularity of the coordinate system. Another way to reduce the size of the input data is to ignore the positions of individual units and instead employ the positions of *groups* of units. Initially, we had hoped to do this by using the echelon information supplied in the data as received and performing analyses on higher-level groups of units, such as companies or platoons. However, there was very little echelon information present in the data, and after removing erroneous echelon identifications, we were left with almost no data of this type. In a real battle situation, it is doubtful that much of this information would be possible to obtain anyway, so an alternative approach which derived this information from the data is not only necessary, but preferable.

The Machine Learning technique of *clustering* can be a powerful tool for generating groups from single instances, which is exactly the problem faced in creating pseudo-echelon information from individual units. Our first attempt employed a hierarchical Bayesian clustering algorithm (Cheeseman & Stutz, 1995) which we hoped would not only generate second-order echelons, but the entire echelon hierarchy. However, this approach did not perform well as there was no bias in the algorithm to keeping units within the same group which caused the clusterings to appear and disappear very frequently.

Instead, we made the assumption that the initial deploy-

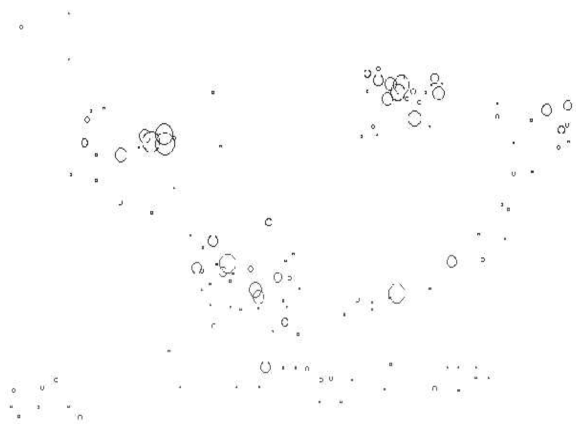

Figure 4: Generated clusterings at time t_0

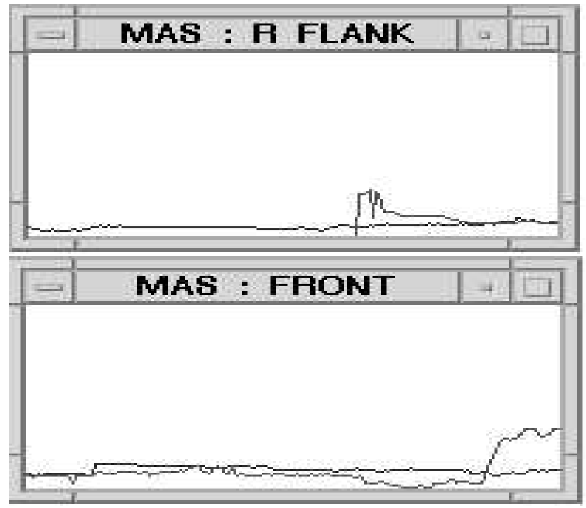

Figure 5: Indicator values

ment of the units on the battlefield corresponded at least roughly to the existing organizational structure and that this structure should be largely preserved over the course of an engagement. We employed a nearest neighbor clustering algorithm (Cover & Hart, 1967) to generate an initial set of clusters for the starting deployment of all the units on the battlefield. As the battle progresses over time these clusters should tend to remain cohesive but not entirely so, as the initial disposition of units may not entirely correspond to actual echelon groupings. Additionally, original clusters may split and/or converge during the course of a battle, particularly in complicated maneuvers. Simply regenerating clusters at each time interval independently of the prior grouping would capture these dynamic changes but would tend to lose the initial groupings which are hypothesized to be more informative than those that occur during a chaotic engagement. For this reason, we modified the non-temporal algorithm to keep units in the same cluster as the previous time step unless a distance threshold is exceeded.

We have not yet formally evaluated the clusters generated by this approach and we are still refining and automating the parameterization but the initial results appear promising. Figure 4 shows the visualization of the clusters as circles which bound their constituent individual units (shown as solid dots).

5 Employing derived representations

The ultimate goal of augmenting and condensing the raw input data is to be able to use the derived information in the process of identifying and recognizing significant patterns in the temporal telemetry data. This is the core of the situation development task and one of the central problems to be solved in pursuit of this task is the ability to identify *events* from low-level *actions* (Bares, et. al., 1994).

In the domain of Army Intelligence analysis (ARMY, 1990), events are a type of *indicator*, significant occurrences on the battlefield which have proven to be evidence for certain actions. For example, a large scale movement to the forward area may indicate preparation for an attack. The calculation of indicators is very difficult and yet extremely important to the process of situation development. Our initial approach toward automatically calculating indicators has been to combine the information in the battlefield layout with the low-level telemetry data to generate indicators which have been described as important in the Army Intelligence literature.

Examples of such indicators are movement into the flanks, massing in the forward or rear, and concentrations of troops in the front or rear of the battlefield. A human expert uses combinations of these indicators along with other types of knowledge to generate hypotheses of potential enemy courses of action. We intend to use this information as well in the plan recognition process. In order to produce a rough idea of the viability of this process we have attempted to recognize patterns in the *indicators* using the SINS continuous case-based reasoning system (Ram & Santamaria, 1997). A simple evaluation has demonstrated that after training on one battle, the system can recognize similar patterns of indicators in a novel but visually similar battle.

Figure 5 illustrates the visualization of these indicators as line plots. Each graph indicates the strength (0 - 1) of the corresponding indicator for each side. These indicators show a rapid spike in the right concentration of the enemy units followed by a rise in the forward concentration, indicating that an attack from the right flank may be about to occur.

Another interesting aspect of this research is that while indicators do provide information which facilitates advance prediction of enemy intents, it is also the case that this information may not become apparent until the predicted plan is well underway. A more useful approach combines information about the past and present situations with *predictions* about future events. While the task of situation development is essentially a prediction task, it is a prediction task at many levels. At the highest level the output is a hypothesis of the overall intention of the enemy forces, and at the middle layers it is predictions of intermediate-level maneuvers and actions. Finally, at the lowest level it may be predictions of very short-term movements of individual units or small groups of units. A key feature of this hierarchical property of the process is that predictions at lower-levels can be used as input to the higher-level predictions.

We are still working on the intermediate and high-level prediction and recognition tasks, but have implemented algorithms for predicting the movements of individual units which allows us to calculate indicators using not only current information but (hypothesized) future information as well. We use the ARINA (autoregressive integrated moving average) forecasting model (Box, Jenkins, & Reinsel, 1994) to make predictions of unit positions various distances into the future. One of the most useful properties of this model is that the predictions are refined over time as more information becomes available so that the indicator calculation and ultimately the higher-level predictions can be refined and updated as information is received during the course of the engagement.

6 Conclusions

Applying Machine Learning techniques to real-world problems can be hard for many reasons. One such reason is that the type and quantity of data available from the real world may not be directly usable by any Machine Learning algorithm with any appreciable results. Therefore, it is often necessary to perform a great deal of processing on the data, to both remove noise and to increase the semantic content of the information received.

We have described our experiences in such a domain, that of real-world "mock" battles conducted by the U.S. Army's National Training Center. These battles generate telemetry data for hundreds of individual units over the course of several days. Our ultimate goal is to construct a system to perform the task of situation development, an extremely difficult problem which requires a great deal of high-level input which is, unfortunately, not directly present in the data with which we must work.

Our task thus far has been to solve several interesting problems of how to generate higher-level information from our low-level telemetry data without employing any external knowledge sources or information. We have described several of the approaches we have taken and commented on their usefulness to not only our problem, but problems faced by Machine Learning researchers in similar domains. It is our hope that these techniques may be employed in other real-world domains to help bridge the gap between what the real world supplies and the algorithms expect as input.

Acknowledgments

Eric Jones, Juan Carlos Santamaria, and Sunil Mishra at Georgia Tech and Uma Movva and Mike Salonish from the Army Research Laboratory have also contributed to this work. We wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was funded by the US Army Research Laboratory under contract C36-W02.

References

Army Headquarters. (1990). *FM 34-3 Intelligence Analysis*. Washington D.C.: Department of the Army.

Bares, M., Canamero, D., Delannoy, J., & Kodratoff, Y. (1994). XPLANS: Case-based Reasoning for Plan Recognition. *Applied Artificial Intelligence*, 8, 617-634.

Box, G.E.P., Jenkins, G.M., & Reinsel, G.C. (1994) Time Series Analysis. Englewood Cliffs, New Jersey: Prentice Hall.

Cheeseman, P. & Stutz, J. (1995). Bayesian Classification (AutoClass): Theory and Results. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.) *Advances in Knowledge Discovery and Data Mining*. Menlo Park: The AAAI Press.

Cover, T.M. & Hart, P.E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13, 21-27.

Huber, M. & Durfee, E. (1992). Plan Recognition for Real-World Autonomous Robots: Work in Progress. In *Working Notes: Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots*, AAAI Fall Symposium, pp. 68-75. Boston, MA : American Association for Artificial Intelligence.

Ram, A. & Santamaria, J.C. (1997). Continuous Case-Based Reasoning. *Artificial Intelligence*, (90)1-2:25–77. (to appear)