Augmenting Wargame AI with Data Mining Technology

John Rushing[1], John Tiller[2], Steve Tanner[1], Drew McDowell[1]

[1] Information Technology and Systems Center
University of Alabama in Huntsville
Huntsville, AL 35899
{jrushing,stanner,dmcdowell}@itsc.uah.edu

[2] John Tiller Software
Madison, AL 35758
tiller@hiwaay.net

Abstract

Data mining methods can be used to augment a traditional wargame AI in a variety of ways. They can be used in situational analysis to identify high-level structures such as front lines, weakly defended targets, inadequate supply lines or other items of interest. They may also be used to model and predict the likely responses of players, and learn viable decision making strategies by example. These capabilities can both speed the development of an AI and improve its quality by reducing the amount of explicit programming of rules or scripts that would be required, by allowing the decision making to occur at a higher level, and by allowing the wargame AI to adapt over time and learn from previous experiences. Wargames have a long history of use as tools for training military personnel, and are likely to be used even more in the future. Modern computer wargames are sophisticated software tools that can provide detailed data about the actions taken by each player and the underlying situations within the game scenarios that lead up to these actions. Data mining tools can be used to aid in the analysis of this data in order to improve the feedback provided to the player, revealing general tendencies or weaknesses that could be exploited by an opponent.

Introduction

Data mining is a discipline that draws tools and techniques from many areas, including pattern recognition, machine learning, statistics, database systems and others. The purpose of data mining is to aid in the analysis of very large data sets. In particular, data mining can be used to discover relationships between attributes in data sets, discover similar patterns, and perform classification and prediction. Information provided by data mining queries can be used for decision support. Data mining technologies relevant to wargame AI include association rule, clustering, and classification methods.

Association rules (Srikant, Vu, and Agrawal 1997) are capable of identifying relationships between attributes in large data sets. Association rule mining algorithms produce rules of the form: $(A_1 \wedge A_2 \wedge \ldots \wedge A_N) \Rightarrow (B_1 \wedge B_2 \wedge \ldots \wedge B_M)$. The left side of the rule is known as the antecedent and the right side is the consequent. The rule simply states that whenever the antecedent is true, the consequent is often true as well. There are two parameters associated with the rule: support and confidence. The support indicates how often the rule applies (ie. the percentage of cases where the antecedent and the consequent are true). The confidence is the probability that the consequent is true given that the antecedent is true. Association rules are most frequently used for undirected queries, and should be especially useful in diagnosing tendencies of players.

Clustering involves grouping objects into classes so that similar objects are in the same class and dissimilar objects are in different classes. Clustering may also be used to identify outliers, or objects that to not fit well into any particular class. Clustering algorithms use distance or dissimilarity measures to determine which objects belong together. In order to use clustering methods in a wargame, meaningful attributes and distance measures must be defined. Spatial clustering can be used to identify groups of supporting units, identify front lines, and identify gaps or weak points in lines.

A supervised pattern classifier is trained based on a set of patterns with known class assignment. The classifier is later used to classify patterns where the class assignment is unknown. Supervised pattern classifiers can be used in wargames to recognize important conditions or situations that could be triggers for the AI. For example, a classifier could potentially be used to recognize situations where one or more units are likely to be overrun or surrounded. In order to use supervised classifiers in this way two things are required: a set of features or attributes that contain information from which it is possible to predict the desired event, and training data containing instances of the event and near-miss non-instances.

Data Mining in Wargames

Data mining has many potential uses in wargames. Mining can be used both to augment the wargame AI, and to provide feedback to players in order to help them improve their performance. Data mining queries may be directed or undirected. In the case of a directed query, the analyst has a particular goal in mind, or a particular question that they feel is relevant. Undirected queries may reveal relationships that the analyst may not have considered.

Augmenting Wargame AI

One important application of data mining technology is in analysis of the current situation. Wargame engines can provide detailed information about the exact disposition of friendly and enemy units, their capabilities and other important factors. However, this information is often at a level of abstraction that is far below the level where the decision is being made. Consider the AI for a strategic level ground based wargame. At the highest level, the AI must decide whether to attack, defend the current position, or withdraw. There may be several factors that influence this decision, such as the scenario objectives, overall strength of friendly and enemy forces, access to supplies, available air support, weather and so forth. While some factors like overall strength may be computed in a straightforward way, other less obvious factors may benefit from data mining analysis. Mining methods can be used to recognize situations of interest such as units in danger of being isolated, gaps in an enemy line, opportune targets for air strikes etc.

Another important use of data mining technology is learning strategies by example. Trainable pattern classifiers can be used to model a decision making process based on large numbers of sample decisions. The classifiers could then be used both to predict the likely course of action of opponents, and directly as components of an AI system. Learning from examples is a challenging process, and there are several factors that influence success. It is important to provide the classifier information at the appropriate level of abstraction. The data provided should contain enough detail to make a decision, while irrelevant details should be eliminated. The process of choosing the influencing variables can be automated to some extent, either by using classifiers with built-in selections mechanisms (such as decision trees), or by using external feature selection algorithms.

Perhaps the most difficult part of the learning process is generating good examples. The examples must cover as many cases as possible to ensure proper generalization of the classifier. The number of games required to produce a good number of samples will depend at least partly on the level of abstraction of the decision. During the course of a single game, a player may make a small number of decisions dealing with overall strategy, and a much larger number of decisions about the actions of individual units or small groups of units. Another important consideration is that the negative instances may far outnumber the positive ones: for every action a player chooses there are vast numbers of alternatives. In cases where the number of examples available is restricted, a hybrid approach could be used. An initial model could be constructed based on expert knowledge, and then the model could be refined based on experience.

Supervised pattern classification has been widely studied, and there are a large number of classification algorithms that are appropriate for use in wargames. Supervised learning algorithms that incrementally learn and can tolerate concept drift (functions or behaviors that change over time (Widmer and Kubat 1996)) are especially valuable in a wargame environment. Versions of supervised learning algorithms exist with both of these capabilities for neural networks (Polikar et. al. 2001), support vector machines (Syed, Liu, and Sung 1999), decision trees (Hulten, Spencer, and Domingos 1997), and ensemble classifiers (Street and Kim 2001). In many cases, producing a user-readable model is also of value. Decision trees and belief networks (Heckerman 1995) are logical choices in those instances. A classifier that is capable of dealing with uncertainty is also valuable. Uncertainty may occur on the input due to fog of war, or on the output if more than one decision is appropriate, or if the case being considered is outside the predictive range of the classifier.

Providing Player Feedback

Data mining tools can be used to help characterize, model, and predict the behavior of human players. The information provided by these tools can then be analyzed for weaknesses or tendencies, and used to provide feedback to the player. It can also be used to help devise or refine AI strategies to challenge the player in future games or scenarios. In general, data mining queries can be partitioned into two classes: directed and undirected. In the case of a directed query, the analyst has a particular goal in mind, or a particular question that they feel is relevant. An undirected query on the other hand may be used to identify relationships or patterns that may be of interest for further consideration. Undirected queries may reveal relationships that the analyst may not have considered.

The attributes available for predicting the player behavior depend on the level of information captured in the game replay logs, and the amount of information available to the player. Depending on how fog of war settings are chosen and implemented, the player may not have exact knowledge of the situation. For example, the player might know that there are incoming aircraft at a given range, but might not be able to distinguish ground attack aircraft (like an A-10) from air superiority fighters (like an F-15) without visual identification.

In order to make the preceding discussion more concrete, it is useful to consider the concepts of directed queries in the context of an actual wargame. Modern Air Power is a game that has potential for use in training for the United States Air Force. In this game, players control aircraft, radars, SAM sites and other resources in an entire theatre of operations. Players must make decisions about when, where, and how to deploy the resources available to them. The players' decision making processes could be captured by constructing classifiers that model key aspects of their behavior. For example, classifiers could be constructed to determine when a player willingly engages in dogfights, how the player uses SAM sites, how they deploy tanker aircraft and so forth.

Another alternative is to frame the data mining queries in extremely general terms, with no specific targets in mind. This is the approach that is often used in association rule mining, where one is interested discovering relationships between attributes. In the case of wargames, the attributes may be user actions, events, game states, or other features computed from the game logs. The mining could be directed to discover patterns present at a particular time, or to discover temporal patterns (Agrawal and Srikant 1995) or sequences of events. This process is complicated somewhat by the fact that the player's decisions may be at least partially governed by factors that are not captured by the attributes. Furthermore, the decisions may not be entirely consistent, as the player may make different decisions in similar situations.

Demonstration

There are many interesting problems in ground-based wargames related to identification of front lines. The AI for a wargame should be able to determine where the fronts are, which units constitute the front, and the locations of any gaps or weak points in the front. There are many possible ways to address these problems, including the use of general-purpose segmentation and clustering algorithms. The use of one such technique, Density Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et. al. 1996), is demonstrated.

DBSCAN is a density based clustering algorithm. It identifies groups of samples or patterns that are connected by contiguous dense regions in the pattern space. Unlike many commonly used clustering techniques, it can form clusters of arbitrary shape and size. The algorithm scales very well as it only requires a single pass through the input data set. DBSCAN can identify outliers, or patterns that do not fit well in any particular cluster. DBSCAN is typically used in clustering applications with sets of continuous (real-valued) variables. These variables constitute an N dimensional space where each dimension or axis corresponds to a variable. The variables are often normalized to have the same range so that they are given equal weights by the distance function.

DBSCAN can be used to identify fronts in ground-based wargames by identifying continuous groups of units that are close enough together to provide mutual support. Friendly units that constitute a continuous front line will be grouped into a single cluster, while isolated units will be identified as outliers. Gaps in the front will be exposed, as

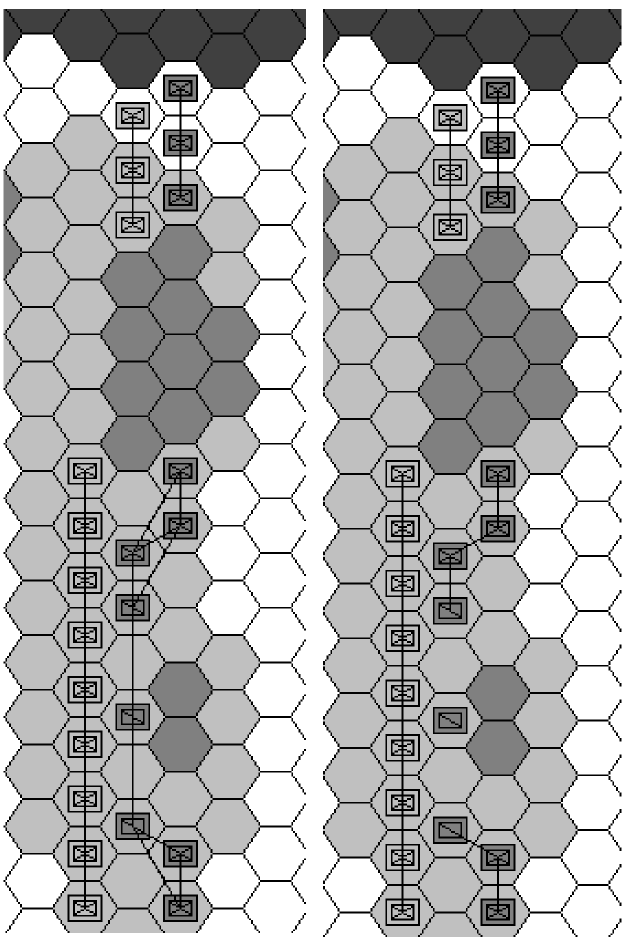

Figure 1: Lines Broken by Terrain

(a) DBSCAN run with low thresholds

(b) DBSCAN run with high thresholds

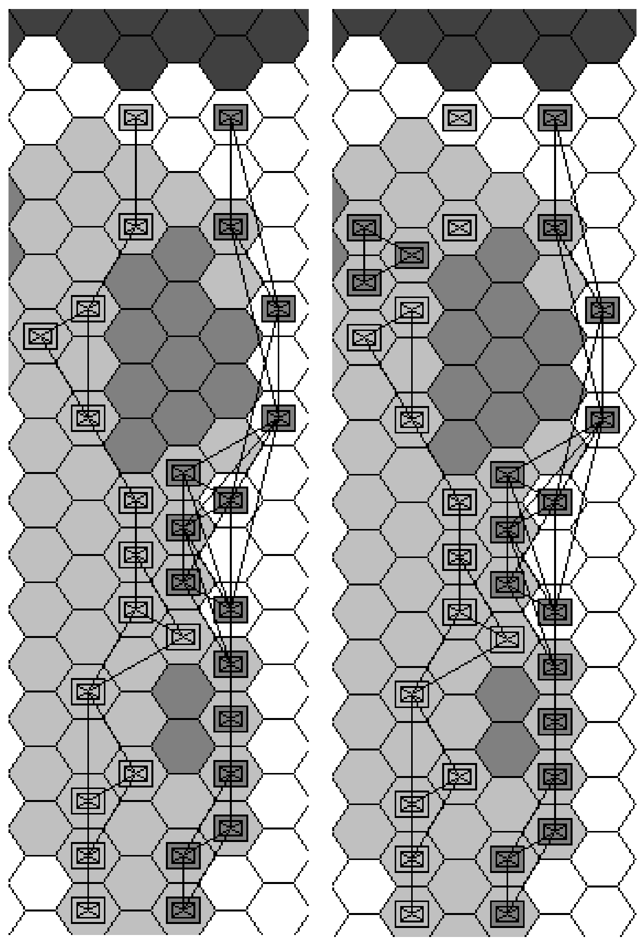

Figure 2: Simulated Parachute Drop

(a) Before drop, light gray line complete

(b) After drop, light gray line split

the units on either side of the gap will be in different clusters. The DBSCAN algorithm assumes that there is a meaningful distance measure that is defined for the input patterns. In the case of an N-dimensional pattern space, distance measures such as Manhattan distance and Euclidean distance are commonly used. For ground-based wargames, it makes sense to consider the movement cost of the shortest viable path from unit to unit. This path must take into account variable movement costs for different types of terrain, impassable hexes, features such as roads, rivers and railroads, and the presence of enemy units. It may be desirable to vary the radius by unit type, as some units are much more mobile than others.

Another important consideration is the strength of the units. A group of very weak units may not constitute a viable front, even if there are large numbers of them. The total strength of the units in a particular region is therefore more interesting than the number of units. Assignment of a strength rating will depend on the game engine and rules of the game, and may consider factors such as the size of the unit, its supply state, its morale level, any attrition it has suffered and other factors. The maximum distance and minimum strength acceptable are conveyed to DBSCAN through tunable parameters (epsilon for distance, minStrength for unit strength).

In order to demonstrate the algorithm, some simple cases were devised. In each of the figures in this section, there are two opposing sides, light gray and dark gray. There are infantry units with strength of 5 and cavalry units with strength of 2, represented by the standard symbols. There are four types of terrain: clear, rough, mountain, and ocean. The clear hexes are white, rough hexes light gray, mountain hexes darker gray and ocean hexes darkest gray. The clear hexes have a movement cost of 1, rough hexes 2, and the mountains and ocean are assumed to be impassible.

The units forming the line are linked by solid lines plotted from center to center of the connected units. A cluster consists of all connected units. Figure 1 shows the light and dark lines split by a mountain range. In addition, there is a weak point in the southern part of the dark line. When DBSCAN is run with liberal parameters (epsilon = 2, minStrength = 6), the southern part of the red line appears unbroken. When DBSCAN is run with more strict parameters (epsilon = 4, minStrength = 7), a gap in the line appears, and one of the cavalry units is shown to be isolated. Figure 2 shows long lines through rough terrain, and illustrates the result of a hypothetical parachute drop by dark. Figure 2a shows the situation before the drop, and Figure 2b shows the situation afterwards. The result is a break in the light line, and two isolated light units to the north of the pass.

Conclusions

Wargames can benefit from the use of data mining technologies, both in augmenting the wargame AI and in providing feedback to players. Mining methods can aid in situational analysis, decision making, adversarial response prediction, and modeling. A wide variety of techniques including clustering, classification and association rule mining are applicable. A simple application of density based clustering, namely the identification of front lines and isolated units in a ground based wargame, was illustrated.

Acknowledgements

Phase I and Phase II STTR funding from the Air Force Office of Scientific Research, Dr. Robert Barker program manager, is gratefully acknowledged.

References

Agrawal R. and Srikant, R. 1995. Mining Sequential Patterns, In *Proc. 11th Intl. Conf. on Data Engineering*, 3-14.

Ester, M., Kriegel, H-P., Sander, J. and Xu, X. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, In *Proc. 2nd International Conf. on Knowledge Discovery and Data Mining (KDD-96)*, 48-66.

Heckerman, D. 1995. A Tutorial on Learning with Bayesian Networks. Technical Report. MSR-TR-95-06. Microsoft Corporation.

Hulten, G., Spencer, L., and Domingos, P. 1997. Mining Time-Changing Data Streams. In *Proc. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 97-106.

Polikar, R. Upda, L., Upda, S.S. and Honavar, V. 2001. Learn++: An Incremental Learning Algorithm for Supervised Neural Networks. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 31(4): 497-508.

Srikant, R., Vu, Q., and Agrawal, R. 1997. Mining Association Rules with Item Constraints. In *Proc. 3rd Intl. Conf. On Knowledge Discovery and Data Mining*, 67-73.

Street, W.N., and Kim, Y.S. 2001. A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification. In *Proc. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 377-382.

Syed, N.A., Liu, H. and Sung, K.K 1999. Handling Concept Drifts in Incremental Learning with Support Vector Machines. In *Proc. Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 317-321.

Widmer, G. and Kubat, M. 1996. Learning in the Presence of Concept Drift. *Machine Learning* 23: 69-101.