Learning & Prediction in Relational Time Series: A Survey

Terence K. Tan
Christian J. Darken
The MOVES Institute
Naval Postgraduate School
700 Dyer Road, Watkins Ext. 265
Monterey, CA 93943-5001
ktan@nps.edu, cjdarken@nps.edu

Keywords:
Learning, Prediction, Relational Time Series, Sense making

ABSTRACT: *Making sense out of a stream of incoming percepts is the first step in any agent's cognition process. The purpose of sense-making is usually to facilitate sound decision making, often by making predictions of future events or actions. In the case that the percepts are relational, the technologies available for this task are mainly based on production systems or statistical graphical model inferencing processes such as Bayesian networks. To apply these approaches, it is necessary that domain knowledge be known or that examples are available to a supervised learning process. Darken (2005) proposed a situation learning (SL) approach to learn a string of percept sequence into a set of overlapping situations. This approach has much potential for learning and predicting in domains that are characterized by high variability and great number of predicates and terms that become known only at runtime, and which feature a trending or moving context environment. In this paper, we attempt to define relational time series (RTS) and its characteristics for evaluating current learning approaches for learning and prediction of RTS. We also report the prediction accuracies of various prediction techniques based on SL in a benchmark environment.*

1. Introduction

Making sense out of a stream of incoming percepts is the first step in any agent's cognition process. This stream of percepts can be described as a relational time series, which is a time series of percepts in first order logic representation. Relational representation is a natural way to express the relations among the constants in the virtual environment. We can use such a time series to learn the behavior of other agents. Furthermore, relational representation allows inference of additional knowledge from the structural properties afforded by the relations among the constants. In particular, such structural properties can help to predict even atoms that we have not seen before.

In this paper, we first describe the relevance of event prediction to behavior representation in modeling and simulation (BRIMS). Next, we frame the problem of event prediction in relational time series and describe its challenging characteristics of learning and prediction. We then qualitatively evaluate Markovian and Non-Markovian approaches against the problem characteristics. Finally, we present an overview of a situation learning (SL) approach to learning, and present a few prediction techniques in conjunction with the SL and experimental results.

2. Relevance of Prediction Task to BRIMS

Prediction capability is important in many applications. In Modeling and Simulation, Kunde and Darken (2006) showed that predictive ability enhanced the realism of the behavior of a simulated military commander. Human beings do not make decisions based only on the current situation, but also on the predicted development (Kurby & Zacks, 2008). Klein (1999) describes the process of prediction as "mental simulation" while Fauconnier and Turner (2002) describe it as "running the blend". The ability to predict future events and to act based on the predicted states can enhance the fidelity of an agent behavior model.

There are several possible ways of using the predictive information. In production system, where decisions are made based on rules, we can use the future states in conjunction with the current state in the precondition of the rules. A rule that delays a call for fire can be described in Figure 1, using CLIPS syntax. The interpretation of this rule is: if the number of enemy sighted is one, and the future number of enemy sighted is one, and then call for fire. If the number of enemy sighted is one, and the future number of enemy sighted is greater than one, then, issue the wait command. The rationale for these rules is that the agent cannot handle more than one enemy. In reinforcement learning, given a situation, the agent chooses an action from its policy that maximizes some goodness measure. In exploitation mode, the agent will simply choose the action with the highest cumulative reinforcement. In exploration mode, the agent will randomly choose other actions. We envision that in exploitation mode, the agent can generate a prediction of a sequence of future events based on each possible action in the policy, and choose the action that has the desirable

state in the predicted sequence. From these examples, we can see that predictive capabilities can be useful to enhance the performance of an agent.

```
(defrule rule1
     (NumberOfEnemySighted 1)
     (Future (NumberOfEnemySighted 1))
      =>
     (assert (Command Fire))
)
(defrule rule2
     (NumberOfEnemySighted 1)
     (Future (NumberOfEnemySighted>1))
      =>
     (assert (Command Wait))
)
```

Figure 1: A CLIPS rule that uses predicted state

3. Relational Time Series

We now define the relational time series (RTS) and the prediction problem, and discuss its characteristics.

3.1 Definitions

We define a relational time series (RTS) as a sequence of relational percepts. Each percept is a ground atom defined as $p_i = r(c_1, c_2, \ldots, c_m)$, where r is the predicate and $c_{j\in(1..m)}$ are constants that represent objects. An example of a RTS is given in Figure 2. There are two types of percept: point and interval. The point percept exists or is active for a point in time and immediately ceases to exist. For example, a percept that describes "a ball hitting the wall" becomes obsolete immediately after it occurred. An interval percept occurred and remains true until something happens that change its state. For example, a percept that describes "a ball is in the box" is true until the ball is removed. The interval percept has a '+' indicator in the predicate as shown in Figure 2. A percept that is true is said to be active. The interval percept becomes inactive when a special type of point percept arrives, indicated by '-' in the predicate.

P_i	Time	RTS	Semantics
P_1	1	(loc+ Ed road)	Ed is at location road
P_2	2	(loc + Fox1 road)	Fox1 is at location road
P_3	3	(goE Fox1 east)	Fox1 is going east
P_4	3	(loc- Fox1 road)	Fox1 is NOT at location road
P_5	10	(loc + Fox2 road)	Fox2 is at location road
P_6	11	(goE Fox2 east)	Fox2 is going east
P_7	11	(loc- Fox2 road)	Fox2 is NOT at location road

Figure 2: An example of RTS

The prediction problem can then be defined as follows. Let $\{p_1 p_2 \ldots p_n\}$ be the sequence of percepts from the time the agent started learning till the present time, where i in p_i refers to the running index of each incoming percept. A one-step prediction problem is then $\{p_1 p_2 \ldots p_n\} \vdash p_p$ where $\vdash$ is an operator that weakly implies that p_p is the next most likely percept. A two-step prediction problem is defined as $\{p_1 p_2 \ldots p_n p_{p1}\} \vdash p_{p2}$, given that $\{p_1 p_2 \ldots p_n\} \vdash p_{p1}$. This means that the percept predicted by a one-step predictor is used for the second step prediction. The two-step prediction problem can be generalized to a multiple-step prediction problem.

3.2 Problem Characteristics

Learning and prediction in RTS from unknown environments is a hard problem because of a set of challenging characteristics. (1) Since there is no knowledge of the environment, there can be no predefined statistical graphical model or structure for knowing what kinds of atom that will arrive next. This leads to the second characteristic, which is (2) arbitrarily many constants and relations of arbitrary arity. This results in a large state space. To make the matter worse, the sequence of percepts can be (3) chaotic, and a function of a moving context, with different percept subsequences occurring in different contexts. While each atom can be treated as a proposition, ignoring the (4) relational structural properties can miss out opportunities to predict atoms that have not been seen before.

The above characteristics of RTS present many challenges and opportunities for sense-making. We have not seen any research effort that directly addresses the RTS problem. Research areas such as statistical relational learning or operator observable model are the most relevant. However, they do not directly address all RTS characteristics. Sun & Giles (2001) provide a nice introduction and review of approaches for sequence learning. Their review appears to address a sequence of proposition (versus atom), and do not directly address all the characteristics of RTS. In the next section, we will review current possible approaches to RTS and evaluate them against the characteristics of RTS.

4. Current Approaches for RTS

In order to succeed in learning and prediction in RTS from unknown environments, the algorithms must demonstrate online structural flexibility in its learned knowledge base, and flexibility in using the knowledge base to make predictions. Here, we discuss possible learning approaches by organizing them into Markovian and non-Markovian learning approaches.

4.1 Markovian

Markovian approaches refer to approaches that assume Markov properties. These approaches are variations of Markov chain or Hidden Markov Model. In Markov state machine (MSM), each state with the same input can transit probabilistically to different states. Markov state machine is sometimes called Markov Chain (Luger, 2008, Section 9.3.5). If the transition is defined based on the current state, it is termed first order

Markov model. If the transition is based on n previous states, it is termed n-order Markov Model. The main limitation of the Markov lies in its limited potential to generalize to novel situations due to its strict order. A new situation may simply have the order of two states switched, or have extra trivial percepts in between the states, but the Markov model will not detect such a switch and treat it as a new sequence, resulting in over-fitting. Furthermore, these approaches treat each relational atom as propositional, and does not leverage on the relational structure to make inferences.

The observable operator model (Jaeger, 2000) is described to be a generalization of the hidden Markov model. It models a stochastic process in order to compute the probability distribution over all possible future sequences, given that a sequence of observation has been observed. The probability of observing a future sequence is:

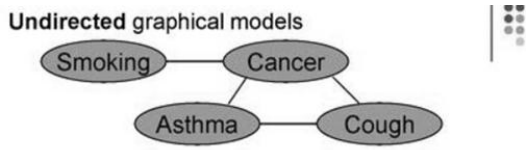
$$P(Y_0 = a_{i_0}, Y_1 = a_{i_1}, \dots, Y_k = a_{i_k}) = \mathbf{1} T_{a_{i_k}} T_{a_{i_{k-1}}} \dots T_{a_{i_0}} w_0$$

Where

- Y_0, Y_1, …, Y_k are random variables in the sequence
- $a_{i_0}, a_{i_1}, \dots a_{i_k}$ are the observables corresponds to Y_0, Y_1, …, Y_k and i refers to different types of observable.
- **1** is an identity vector that attempts to sum the column vector to form the probability value
- $T_{a_{i_k}}$ is the operator corresponds to an observable at position k in the sequence where $T_a=M^T O_a$ where M^T is the transpose of the state transition matrix and O_a is a diagonal matrix that express the conditional distribution of each observation given each state.
- w_0 is the initial distribution of the hidden states.

The learning process requires prior manual estimation of a dimension (number of feature) and the set of features. This is a potential limitation in an unknown environment. Spanczer (2007) identified that learning in OOM, though Simple, is a partially solved problem. He also highlighted the difficulty of choosing characteristics and indicative events in order to have an efficient algorithm. In addition, OOM uses proposition representation, which does not leverage the structural properties to make prediction.

Statistical Relational Learning (SRL) is not strictly a Markovian approaches, but can be implemented with either Markovian or Non Markovian techniques. SRL attempts to combine first order logic with statistical learning (Getoor & Taskar, 2007). The relational learning addresses the relational structure that better represents the world while the statistical learning addresses the uncertainty of the data by relaxing the hard constraint in the relational domain. SRL are usually modeled using graphical model such as Markov Network (MN) or Bayesian Network (BN). While BN models causality, MN models association between two random variables, in the form of an undirected graph. The nodes in the MN are organized into cliques. A potential function is defined for each clique, which are non-negative real values for each state in each clique. The equation and an example for calculating a joint distribution is given in Figure 3. The example shows four random variables. Smoking and caner nodes form one clique while cancer, asthma and cough nodes form another clique. Suppose that we have Φ(Cancer=true, Asthma=true, Cough=true)=5.0, Φ(Smoking=true, Cancer=true, Asthma=true, Cough=true) = (4.5 * 5.0) / Z where Z is a normalizing factor that sum over all possible states. SRL has seen many applications such as relational classification (Jensen et al, 2004), Link based clustering of web search (Wang et al, 2001), link prediction in relational data (Taskar et al, 2004).



Potential functions defined over cliques

$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

Figure 3: An example of Markov Network (Domingos, 2008)

Khosravi & Bina (2010) identified several limitations of SRL. The biggest limitation is the complexity of inferencing because the size of the graph grows exponentially with the number of attributes and objects. Most inferencing methods are based on the standard Bayesian or MN inferencing approaches. MN's inferencing approach requires the computation of the partition function Z, which make the inferencing process NP-Complete. Most of the current researches are focusing on making the inferencing process more efficient. SRL appears to better suit a domain with low variability such as those that has many instant of data that are usually arranged in a relational database. This is due to the great challenge of structural learning in SRL. In RTS where we expect mostly unknown, large and chaotic state spaces, SLR is unsuitable for RTS.

4.2 Non Markovian (NM)

Non Markovian learning approaches do not regard or may relax the sequential order requirement of the RTS, and invalidate Markov assumption. There are many NM techniques that are capable of learning and making

prediction on RTS, with varying learning capabilities. Approaches such as production system (PS), finite state machine (FSM) have no or limited learning capabilities after they are trained and deployed. Many simple event prediction approaches can be implemented based on these approaches that encapsulate domain knowledge. PSs model the domain knowledge as a set of if-then rules. We can define a rule with preconditions that describes the situation to be matched. The consequence of the rule provides the predicted atom. FSMs are similar in that given a state, it can predict the next input/events and the resultant state. These approaches rely heavily on domain knowledge, which can only be created for known environments. Even if the developers have good anticipation capabilities or foresightedness, encoding the large state space is usually prohibitive. Furthermore, the nature of chaos can only be encoded through statistical learning.

Bayesian Network (BN) is another type of NM that encapsulates domain knowledge, usually in the form of causation structure. BNs model the casual relation among the random variables in the form of a directed graph. BN are often used to interpret percept sequences, to derive at possible adversarial goals and actions by computing the posterior probabilities of goals, states, and plans, given the percept sequences (Kott & McEneaney 2006). Given a likely goal or state, other BNs can be used to compute the posterior probability of future actions. Many BNs require the structures to be predetermined and trained offline. Furthermore, when no training example is available, the conditional probability tables are based on human subjective judgments. Hence, BN can only be used if domain knowledge is available. With the large state space and chaotic nature, BN structural learning is unsuitable.

Genetic algorithms (GA) have been used to generate possible scenarios / plans based on perceived goals and situations (Kott & McEneaney 2006). For example, given the current situation and assumed goals, GA can generate the possible future events to serve as predictions. Each GA requires some fitness evaluation functions, which can be heuristics, or simulators. These evaluation functions can limit the nature of scenarios to be evaluated. Furthermore, these functions are developed for known domains. The other limitation is the assumed adversarial goals, which can be inferred using a BN, or based on subjective expert judgment. Here, GA can only be used if domain knowledge is available. While GA can search a large state space efficiently, the state space cannot be predefined for unknown and chaotic environments.

There are NM approaches that are able to continue to learn after they are deployed. These approaches may not require domain knowledge, and are able to learn from unknown situations. Inductive learning is one such approach. In Inductive learning, an agent learns a general function or a set of rules from specific input-output pairs (Russel & Norvig, 2010, Section 19.5). Inductive Logic Programming is a type of inductive learning that induces first order logic theories from examples in relational form. For example, if we have the following atoms: Father(john, caleb), Father(caleb, timothy), grandfather(john, Sheryl), we can induce a rule: $\forall x \forall y \forall z$, Father(x,y), Father(y,z) $\Leftrightarrow$ GrandFather(x, z). The main limitation is on the strict logic constraints. A rule will not be learnt if there is just one counter example. For example, the Grandfather rule is generally true. However, if there is just one case of abnormal relation in the family that contradicts the rule, that rule will be violated, and will not be induced, even though it may be true statistically. Such contradictory phenomenon is common in a chaotic world. While probabilistic inductive logic programming may seem to solve the problem, the entire ILP algorithm must be rerun for each arriving percepts. This poses a great problem because ILP is exponential in the number of predicates and constants. Hence, ILS is unsuitable for online learning in RTS.

Reinforcement-learning (Luger, 2008, Section10.7) can also be regard as a NM approach that supports online learning. In reinforcement-learning, an agent learns a set of policy for actions selection. The policy contains a set of state-action pairs with a value that describes the historical goodness of applying that action in that state. The goodness value is accumulated based on a reward or penalty function known as "reinforcement". Reinforcement learning is not the same as RTS learning mainly because its main focus is to learn a set of policy, which involves actions taken by the agent, while RTS learning needs to predict environmental states even though they are irrelevant to the reinforcement calculation.

4.3 Discussions

Many current approaches for situation reasoning assume that domain knowledge is known. While these approaches have work well in many applications, they will fail in unknown environments. Unknown environments require agents to be robust and flexible, as well as to be able to learn and to adapt in new environments. While a learning agent is able to improve its performance, the structures of the knowledge representation are usually fixed. Structural or rule learning are usually limited and done offline due to the exponential complexity. We need structural flexibility, or multiple structures to account for the chaotic nature of the RTS.

Methods such as ILP or MSM are either logic constrained, have strict sequence requirement, or based on propositional representation. While Markov model

and its variances have found many success stories, its strict sequence requirement prevent it to be used for unknown situations. Likewise, strict logic constraint does not allow ILP to predict atoms that have not been seen before. Reinforcement learning is not designed for RTS learning and prediction. Furthermore, many methods assume propositional data representation even though the relational formalism is a more natural way of representing the world of objects. While SRL may allow structural and statistical inferencing, their largely constraint topological network structures prevent them for uses in unknown and chaotic environments. Hence, these methods are hard to generalize to predict atoms that have not been seen before.

A summary of the evaluation of the current methods for RTS learning and prediction is given in Figure 4. The scores at the last column provide an indication on how suitable each method is for RTS. The higher the score, the higher it may be used for RTS. Nevertheless, since none of them achieve a full score, we need a new learning and inferencing method for RTS. We will introduce a new situation learning (SL) approach for learning a RTS in unknown environment that features structural agility in its learned knowledgebase, and allows flexible use of the knowledgebase to make predictions for unseen states.

Method	Relational Data	Unknown Environment	Huge State Space	Probabilistic Data	Online Structural learning	Noisy Inference	Score
PS	✓	×	✓	×	×	×	2/6
FSM	×	×	✓	×	×	×	1/6
BN	×	×	×	✓	×	✓	2/6
GA	×	×	✓	×	×	×	1/6
ILP	✓	✓	✓	×	✓	×	4/6
MSM	×	✓	✓	✓	✓	×	4/6
RL	✓	✓	×	✓	×	×	3/6
OOM	×	✓	✓	✓	×	×	3/6
SRL	✓	×	×	✓	×	✓	3/6

Figure 4: A Summary of evaluating current approaches for RTS learning and prediction

5. A Situation learning Approach to RTS Learning and Prediction

The set of RTS characteristics is challenging. Many current learning methods are not designed to directly address these challenges. In this section, we describe a possible solution that shares some properties as the recent event segmentation theory (EST, Kurby & Zacks, 2008), which decompose the RTS into a set of situations. Unlike EST, the boundary of each situation has no semantic correspondence to the real world event, but is based on a temporal function. An advantage of the situation learning approach is that, it does not have event boundary, and hence avoids the high transient error rate at the event boundaries.

5.1 learning

The situation learning (Darken 2005) approach learns a RTS into a set of situations (not to be confused with the related notion of situation in situation calculus). The approach appears to use a sliding time window to identify sets of percepts called "situations". When a new percept arrives, this new percept serves as a reference and forms a situation that contains older percepts that were received and are still active within the time window from the time stamp of this new percept. This new percept becomes the predictive target atom of the situation that has just been formed. If the situation already exists in the knowledge base, the number of occurrence of this situation is incremented. Otherwise, this situation will be added into the knowledge base. Note that this is not a Markovian approach. An active atom can be received long time ago and still persist even though other later atoms have become inactive. Hence, the sequential order is lost. In fact, the sequential order may be relaxed to achieve better prediction accuracy.

Instead of learning the entire RTS with one graphical model such as a BN or MN, the approach effectively generates multiple simple networks of two layers as the time window slides through the RTS. Given a relational time series as shown in Figure 2, the agent starts with zero knowledge and forms the situations as soon as the first percept arrives as shown in Figure 5.

{}	1	(loc+ Ed road)	1
{[loc+ Ed road]}	2	(loc+ Fox1 road) (loc+ Fox2 road)	1 1
{[loc+ Ed road] [loc+ Fox1 road]}	1	(goE Fox1 east)	1
{[loc+ Ed road] [loc+ Fox1 road] [goE Fox1 east]}	1	(loc- Fox1 road)	1
{[loc+ Ed road] [loc+ Fox2 road]}	1	(goE Fox2 east)	1
{[loc+ Ed road] [loc+ Fox2 road] [goE Fox2 east]}	1	(loc- Fox2 road)	1

Figure 5: A collection of situations (left column) and their associated prediction (right column)

When the learning process starts, there is no percept. The current situation is an empty set. When a percept (loc+ Ed road) arrives, it becomes a reference point and a time window is cast in retrospect to determine which percepts are currently active in the window. Since there is no active percept, the situation that predict (loc+ Ed road) is an empty set (first row). When the second percept (loc+ Fox1 road) arrives, it becomes the next reference. Assuming we have a 5sec time window, we have a situation {[loc+ Ed road]} that predicts (loc+ Fox1 road). When the 3rd percept arrives, we have a situation of two active percepts {[loc+ Ed road] [loc+ Fox1 road]} that predicts (goE Fox1 east). When the 4th percept arrives, we have a situation of {[loc+ Ed road] [loc+ Fox1 road] [goE Fox1 east]} that predicts (loc- Fox1 road). Note that the percept (loc- Fox1 road) deactivates the percept (loc+ Fox1 road). When the 5th percept (loc + Fox2 road) arrives, the current active

percepts are only{[loc+ Ed road]}, which is the same as our second situation. Hence, (loc+ Fox2 road) is added as another possible predicted percept for the situation {[loc+ Ed road]}. When the 6th percept (goE Fox2 east) arrives, we have a current situation of {[loc+ Ed road] [loc+ Fox2 road]} that predict (goE Fox2 east). When the 7th percept (loc- Fox2 road) arrives, 3 active percepts form the situation {[loc+ Ed road] [loc+ Fox2 road] [goE Fox2 east]}

Formally, the situation learning approach processes the percept sequence $\{p_1 p_2 \dots p_n\}$ into smaller disjoint set of percepts (called a situation) $\{s_i\}, i = 1..S$ where S is an integer that defines the number of situation. Let $\tau_a(p_i)$ refers to the time in which the atom p_i is active, $\max_\tau[\tau_a(p_i)]$ refers to the latest time in which p_i is active, τ_c refers to the current time when a new atom p_{new} is received and τ_w refers to the time window duration. $p_i \in s_i$ if $\max_t[\tau_a(p_i)] + \tau_w \geq \tau_c$. Let p_j refers to the percepts encountered after s_j. We write a consequence c_j as a tuple, $c_j = (s_j, p_j)$, such that p_j follows s_j.

5.2 Prediction

The one step prediction task is then: given a set of consequences $\{c_j\}$ and the current situation $s_c = \{p_{c+1} p_{c+2} \dots p_{c+z}$, generate a percept $p_p$that represents the prediction for next future percept p_{c+z+1}. Prediction is correct if $p_p = p_{c+z+1}$. Let $a = \{0,1,2,\dots,i\}$ be a counter that records the number of correct prediction. Prediction accuracy is $\frac{a}{n}$.

5.3 Prediction Techniques

Darken (2005) provides two simple techniques of prediction. The two techniques are Statistical Look-up Table (SLT) and Variable Matching (VM). SLT searches the situation table to look for a situation that exactly matches the current situation. If a match is found, the percept that follows the matched situation with the greatest frequency will be the predicted percept. VM replaces all constants in the atom with variables. Multiple instances of a constant use the same variable. The matching of situations becomes the problem of variable matching with substitution. A substitution is a list of variable bindings, e.g. θ={?a/?b} where variable ?a from one situation is bound to variable ?b in another situation. SUBST(θ,α) denotes the result of applying substitution θ to situation α. A match is then defined as a bijection of variables between the current situation and a match situation. Finding matches is a graph isomorphism problem. An example of the variable representation is shown in Figure 6. In both techniques, there is no prediction when there is no matched between the current situation and any situation in the situation table.

Constant	Variable
[loc+ Ed road]	[loc+ ?x ?y]
[loc+ Ed grass]	[loc+ ?x ?z]

Figure 6: Constant versus Variables Representation

The above two techniques offer some insights into other possible techniques of prediction. Given a set of consequences and a current situation, the predictive target atom can be derived by simple common inferencing techniques such as pattern matching, Bayesian network or Markov chain in conjunction with the SL. We can interpret each consequence as a network (See Figure 7). Note that the SL-Markov approach assumes that the atoms in a situation are in sequential order, even though the order is lost. These techniques provide means to generate the predictive atom given s_c and $\{c_j\}$. Without the SL approach, these three techniques will face exponential complexity in the learning and inferencing process.

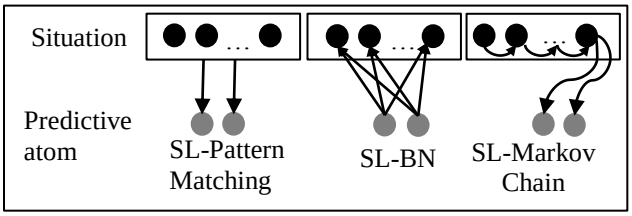


Figure 7: Possible problem formulations for prediction

These multiple simple networks avoid the current challenges in the statistical relational learning (structural learning and exponential inferencing process) by turning the problem into a situation matching and simple inferencing process. The SL approach addresses all challenging characteristics of RTS. Firstly, SL stores the relational data and allows prediction techniques to use the structure of a relational framework. For each unknown situation, SL creates a new situation-prediction tuple, and immediately uses it to predict the next atom. Each situation can accommodate any combination of atoms, regardless of how large the state space is. It manages probabilistic data by having multiple predictive target atoms. Chaos is managed by a simple creation of additional networks for new situations. It can handle noisy inferencing by allowing partial order matching.

We developed additional prediction techniques based on Variable Order Markov Models (VOMM), Multiple Simple Bayesian (MSB) network, and Simple Bayesian Mixture (SBM) in conjunction with SL. When the RTS is decomposed into a set of situations, we can build one simple Bayesian network for each situation with the predictive target atom as the parent node, effectively forming multiple simple Bayesian networks. Since MSB cannot learn certain functions such as Exclusive-OR, we implemented Simple Bayesian Mixtures. SBM

contains probability mixture densities, constructed by normalizing a linear combination of two or more Simple Bayesian Networks probability densities having the same domain and range. SBM is implemented using the Estimate & Maximize (EM) algorithm. VOMM is an extension to the Markov chain models in which a variable order is used in place of a fixed order. We implemented a VOMM model using context trees (Buhlmann & Wyner 1999).

6. Experiments and Results

We compared the prediction performance in a benchmark environment that is used in Darken (2005) in which, an agent wanders around and performs actions randomly. Actions include "go eastward", "pick up weapon", "equipped weapon", "hit", and many more. There are other agents (monsters) in the environment such as goblins, trolls and dragons. There are three types of weapon: pitchfork, dagger and sword. Each weapon may be more effective against each type of monsters. Each time a monster is killed, it will leave behind a weapon. Each monster, weapon, agent and location has a unique name constant. The sequence of percepts describes what the agent sees, such as its location, weapons, and monsters. Our prediction task is to predict the next percept the agent may see, given the past percept sequence. Darken (2005) tested the prediction performance by running the algorithms through more than 250,000 percepts. In this study, we want to know how the SL-prediction techniques work in harsh and new environments. We clear off the memory after 100 percepts have been processed and examine the results after 40 batches of 100 percepts are processed. To simulate noisy environment, we randomly swapped the order of two atoms in the current situation. All experiments were run on a Dell XPS Laptop i7 1.87Ghz 16GB RAM with Windows 7.

The prediction accuracies are given in Figure 8. Each bar in the chart represents the mean prediction accuracy with its associated standard error of a predictor. From the standard error indicators, we can see that the differences are significant for at least at alpha $\alpha = 0.05$ for a statistical student-T test with degree of freedom df=39. There is no significant difference between the SLT and VM, and both techniques are significantly worse off than the SL-Bayesian and SL-VOMM techniques. This is due to the strict requirement of exact matching. When the environment is unknown and noisy, the current situation can hardly match the learned situations in the memory. Figure 9 shows the mean number of no-match for 40 batches of 100 percepts. No-match occurs when the algorithm is unable to find a reasonable situation. SL-VOMM handles the no-match problem by varying the order of Markov Model. The SL-Bayesian techniques handle the problem using Laplace distribution.

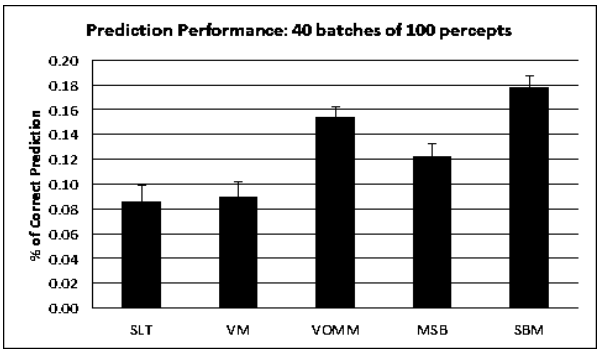


Figure 8 Comparison of Prediction Accuracy for prediction techniques in conjunction with situation learning.

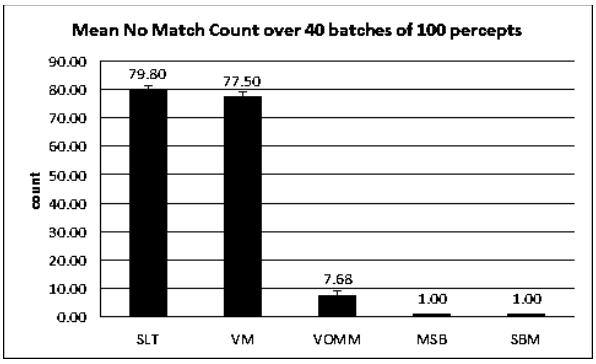


Figure 9 Comparisons of No-Match for prediction techniques in conjunction with situation learning.

The VOMM is a popular approach in sequential and online learning, and handles novel situations better than SL-SLT and SL-VM. While the SL-VOMM does not require exact atom to atom matching, and even allow partial matching, it requires exact sequential adjacency ordering. For example, the sequence of words [The Blue Fish is eating] will not match the sequence [The Fish is eating]. In addition, SL-VOMM treats each atom as a proposition. The multiple simple Bayesian network is able to handle novel situations with the Laplace method of assigning probabilities to newly encountered atoms. Its performance is limited for several reasons. Firstly, there are too many novel percepts. The prior probability for each percept can be very low. The Laplace method assigns a probability that can be unfairly large to new atoms. Secondly, Bayesian network cannot handle exclusive-OR relation. There are atoms that are mutually exclusive. Thirdly, atoms in the sequence are not independent and identically distributed. The SL-SBM performs better than the SL-MSB. However, it also suffers some of the limitations found in SL-MSB. Nevertheless, the purpose of this experiment is to demonstrate the robustness of the SL approach. After a RTS has been decomposed into a set of situation, we can apply different kind of prediction techniques in the

inferencing process. One surprising finding in this study is that, non Markovian techniques can perform better than the Markovian one, even though the Markovian techniques are the popular techniques for sequence learning and prediction.

7. Conclusion

Prediction tasks play an important role in agent cognition processes such as planning and decision-making. However, many current prediction approaches assume that we have domain knowledge. To improve the predictive power in unknown domain, this paper suggests a situation learning approach to learn a RTS. This approach makes possible the use of pattern matching, Bayesian network and VOMM as techniques for prediction. Initial implementations have produced encouraging results. With improved predictive power, it may be possible to develop multi-step event prediction to look for events of interest and to quantify their likelihoods. The challenging benchmark environment consists of too many novel situations for SL-LUT, SL-VM, SL-VOMM, SL-MSB and SL-SBM. Any algorithm that attempts to excel in novel situation prediction may have to possess properties of human creativity. At this point, SL-SBM appears to be the best performer.

For future work, we will explore the theory of Cognitive Integration, also known as Conceptual Blending (Fauconnier and Turner 2002). This theory explains the human creative process, which may help to improve the prediction accuracy in unknown and chaotic environments.

8. Acknowledgement

This research effort is partially funded by the Temasek Defence Systems Institute, a strategic alliance between National University of Singapore and US Naval Postgraduate School. http://www.tdsi.nus.edu.sg/

9. References

Darken, C. (2005). Towards Learned Anticipation in Complex Stochastic Environments. In Proc. Artificial Intelligence for Interactive Digital Entertainment 2005, 27-32

Domingos, P. (2008). Markov Logic: A Unifying Language for Information & Knowledge Management. Retrieved from the web: Nov. 19, 2011, http://videolectures.net/cikm08_domingos_mlmaul/

Fauconnier, G., and Turner, M. (2004). The Way We Think: Conceptual Blending and The Mind's Hidden Complexities. New York: Basic Books

Getoor, L., and Taskar, B. (2007). Introduction to Statistical Relational Learning. MIT Press

Jaeger, H., Zhao, M., and Kolling, A. (2005). Efficient estimation of OOMs. In Proc. of NIPS.

Jensen, D., Neville, J., and Gallagher, B. (2004). Why collective inference improves relational classification. In Proc of the 10th ACM SIGKDD International Conf on Knowledge Discovery and Data Mining, pp. 593–598

Khosrav, H., and Bina, B. (2010). A Survey on Statistical Relational Learning. In Proc of Canadian Conference on AI'2010. pp.256~268

Klein, G. (1999). Sources of Power: How People Make Decisions. Cambridge MA, MIT Press.

Kott A., and McEneaney, W. (2006). Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind. Chapman and Hall/CRC

Kunde, D. and Darken, C. (2006). A Mental Simulation-based Decision-Making Architecture Applied to Ground Combat. In Proc of BRIMS 2006.

Kurby, C. A. and Zacks, J. M. (2008). Segmentation in the perception and memory of events. Trends in Cognitive Sciences Vol.12 No.2

Luger, G., (2008). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th Edition), Addison Wesley,

Spanczer, I. (2007). Observable Operator Models. Austrian Journal of Statistics Volume 36 (2007), Number 1, 41–52

Sun, R. and Giles, C.L., (2001). Sequence Learning: From Recognition and Prediction to Sequential Decision. Making, IEEE intelligent systems

Taskar, B., Wong, M., Abbeel, P., and Koller, D. (2004). Link prediction in relational data. Proc of Neural Information Processing Systems, 659-666

Wang, Y., Wang, Y., and Kitsuregawa, M. (2001). Link based clustering of web search results. LNCS, pp. 225–236. Springer, Heidelberg

Author Biographies

TERENCE KIAN-MOH TAN is a PhD candidate at the MOVES Institute, Naval Postgraduate School. He is also a senior member of technical staff at DSO National Laboratories in Singapore.

CHRISTIAN DARKEN is an Associate Professor of Computer Science at the NPS, and an academic faculty member of the MOVES Institute. His research focus is on human behavior simulations. Previously he was Project Manager at Siemens Corporate Research. He was also the AI programmer on the team that built Meridian 59, the first 3D massively-multiplayer game. He received his Ph.D. from Yale University in 1993.