Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment

Goal Recognition with Markov Logic Networks for Player-Adaptive Games

Eun Y. Ha, Jonathan P. Rowe, Bradford W. Mott, and James C. Lester
Department of Computer Science, North Carolina State University, Raleigh, NC 27695, USA
{eha, jprowe, bwmott, lester}@ncsu.edu

Abstract

Goal recognition is the task of inferring users' goals from sequences of observed actions. By enabling player-adaptive digital games to dynamically adjust their behavior in concert with players' changing goals, goal recognition can inform adaptive decision making for a broad range of entertainment, training, and education applications. This paper presents a goal recognition framework based on Markov logic networks (MLN). The model's parameters are directly learned from a corpus of actions that was collected through player interactions with a non-linear educational game. An empirical evaluation demonstrates that the MLN goal recognition framework accurately predicts players' goals in a game environment with multiple solution paths.

Introduction

Over the past several years, AAA digital games have grown increasingly complex. In some cases this complexity has surfaced as ever-increasing production values, which have driven several game genres to move toward linear levels and reduced player options. In other cases, this complexity has taken the form of open environments, missions with multiple solution paths, and ill-defined objectives. Games such as Grand Theft Auto IV (Rockstar 2008) and Fallout 3 (Bethesda 2008) present open environments where players choose which missions to complete and pursue multiple paths through the storyworld. Alternatively, games like Batman: Arkham Asylum (Eidos and Warner Bros 2009) accommodate multiple strategies for dispatching foes during otherwise linear narratives. Minecraft (Mojang 2009) has recently re-invigorated an open-ended style of gameplay that eschews extrinsic objectives in favor of player-defined goals, expansive virtual environments, and creation-focused activities.

In this latter category of games, players' actions may be difficult to interpret and predict. On the one hand, nonlinearity can promote player agency and increased replayability. On the other hand, it presents significant challenges to game designers interested in crafting cohesive narrative experiences that are attuned to players' actions. These tradeoffs can be addressed by *player-adaptive games*, which dynamically generate and augment gameplay experiences according to players' actions and preferences. The Artificial Intelligence and Interactive Digital Entertainment community has investigated player-adaptive game technologies in areas such as drama management (Li and Riedl 2010; Mateas and Stern 2005; Roberts et al. 2007; Thue et al. 2010) and procedural content generation (Jennings-Teats, Smith, and Fruin 2010; Shaker, Yannakakis, and Togelius 2010).

A key challenge posed by player-adaptive digital games is recognizing players' goals. *Goal recognition* is a restricted form of the *plan recognition* problem. Goal recognition involves identifying the specific objectives that a user is attempting to achieve, where the user's goals are hidden from the system and must be automatically inferred from user actions taken in the game environment.

Goal recognition models offer several prospective benefits to game creators. First, they enable player-adaptive systems that preemptively augment game experiences. Models that provide accurate predictions about players' actions are essential for games to simultaneously promote open-ended scenarios and proactively preserve gameplay cohesion, story coherence, and character believability. Second, recognizing players' goals is important in serious games. Interpretations of players' goals and plans contribute to assessments of learning progress, and goal recognition models can inform intelligent tutoring systems within serious games. Third, goal recognizers can provide valuable information for telemetry efforts. Providing detailed descriptions of players' gameplay objectives and problem-solving plans facilitates interpretation of raw game logs, and player goal data can be analyzed to inform subsequent game designs.

Work on goal recognition has traditionally focused on sequences of user actions that are derived from well-defined goals and plans (Blaylock and Allen 2003; Carberry 2001; Charniak and Goldman 1993; Geib and Goldman 2009; Hu and Yang 2008; Kautz and Allen 1986; Pynadath and Wellman 2000). However, nonlinear digital games often present players with complex or ill-defined goals. Players may not have formulated a specific sub-goal when selecting actions to perform; they frequently choose actions in an *exploratory* manner. Exploratory actions may lead to the identification of new sub-goals or even inadvertent achievement of goals. In this case, goals are derived from the preceding sequences of actions, as opposed to actions that are derived from particular goals.

This paper investigates goal recognition in nonlinear game environments with ill-defined goals and exploratory behaviors. To address the problem of goal recognition for exploratory goals in game environments, which are characterized by cyclical relationships between players' goals and actions, a Markov logic goal recognition framework is introduced. Model parameters are learned from a corpus of behaviors that was collected from player interactions within a nonlinear educational game environment.

The MLN goal recognition framework was evaluated in CRYSTAL ISLAND, a story-centric educational game for middle school microbiology. In CRYSTAL ISLAND, players are assigned a single high-level objective: solve a science mystery. Players are expected to interleave periods of exploration and deliberate problem solving in order to complete a non-linear narrative scenario. In this setting, goal recognition entails predicting the next narrative sub-goal that the player will complete as part of solving the mystery. Findings are presented from an empirical evaluation comparing the Markov logic network framework approach against unigram and bigram baselines. In the evaluation, the MLN goal recognition framework yields significant accuracy gains beyond these alternative probabilistic approaches for predicting player goals in a nonlinear game environment.

Related Work

Recognizing the goals and plans of players offers significant promise for increasing the effectiveness of digital game environments for entertainment, training, and education. Plan recognition, which seeks to infer users' goals along with their plans for achieving them from sequences of observable actions, has been studied for tasks ranging from natural language understanding to collaborative problem solving and machine translation (Carberry 2001; Kautz and Allen 1986). In story understanding, plan recognition is used to infer characters' goals from their actions (Charniak and Goldman 1993); in dialogue systems, it supports natural language understanding and intention recognition (Blaylock and Allen 2003). Because plan recognition is inherently uncertain, solutions supporting reasoning under uncertainty such as Bayesian models (Charniak and Goldman 1993), probabilistic grammars (Pynadath and Wellman 2000), and variations on Hidden Markov Models (Bui 2003) have been investigated. In the restricted form of plan recognition focusing on inferring users' goals without concern for identifying their plans or sub-plans, goal recognition models have been automatically acquired using statistical corpus-based approaches without the need for hand-authored plan libraries (Blaylock and Allen 2003).

The classic goal recognition problem assumes that a single agent is pursuing a single goal using deterministic actions, and it assumes that a user's plan can be identified using a provided plan library. A major focus of recent work on goal and plan recognition has been probabilistic approaches that relax several of these assumptions. For example, Ramirez and Geffner (2010) describe a plan recognition approach that does not require the provision of an explicit plan library. Hu and Yang (2008) describe a two-level goal recognition framework using conditional random fields and correlation graphs that supports recognition of multiple concurrent and interleaving goals. Geib and Goldman (2009) have devised the PHATT algorithm, which is a Bayesian approach to plan recognition that focuses on plan execution. PHATT provides a unified framework that supports multiple concurrent goals, multiple instantiations of a single goal, partial ordering among plan steps, and principled handling of unobserved actions. Recent work focused on real-world applications of goal recognition has emphasized efficient and early online prediction. Armentano and Amandi (2009) describe an approach for predicting software usage behaviors by learning a variable order markov model classifier for each user goal. Sadilek and Kautz (2010) use Markov logic to investigate multi-agent applications in the related area of activity recognition. The work presented here focuses on goal recognition in complex, nonlinear game environments, which often include ill-defined sub-goals and cyclical relationships between goals and actions.

Within digital games, recent work has explored goal recognition to determine players' objectives in an action-adventure game, support dynamic narrative planning, and create adaptable computer-controlled opponents. Gold (2010) describes training an Input-Output Hidden Markov Model to recognize three high-level player goals in a simple action-adventure game. Mott, Lee, and Lester (2006) explore several probabilistic goal recognition models to support dynamic narrative planning. Kabanza, Bellefeuille, and Bisson (2010) explore challenges with behavior recognition in real-time strategy games and present preliminary results for creating adaptable computer-controlled opponents. The current work investigates a Markov logic network goal recognition framework for an educational game environment, with the eventual aim of dynamically tailoring game experiences to players.

Observation Corpus

In order to investigate goal recognition in a nonlinear game environment involving many possible goals and user actions, data collected from student interactions with the CRYSTAL ISLAND learning environment were used.

CRYSTAL ISLAND (Figure 1) is an educational game for eighth-grade microbiology. It is built on Valve Software's Source™ engine, the 3D game platform for Half-Life 2. The environment features a science mystery where students attempt to discover the identity and source of an infectious disease that is plaguing a research team stationed on the island. Students play the role of a visitor who recently arrived in order to see her sick father, but they are promptly drawn into a mission to save the entire research team from the outbreak. Students explore the research camp from a first-person viewpoint and manipulate virtual objects, converse with characters, and use lab equipment and other resources to solve the mystery. Now in its fourth major iteration, CRYSTAL ISLAND has been the subject of extensive empirical investigation, and has been found to provide substantial learning and motivational benefits (Rowe et al. 2010). Students consistently demonstrate significant learning gains after using CRYSTAL ISLAND, and they report experiencing boredom less frequently than in alternative instructional software. CRYSTAL ISLAND is also challenging for students, with fewer than 50% of students solving the mystery in less than an hour. The current investigation of goal recognition models is part of an overarching research agenda that is focused on artificial intelligence technologies for dynamically shaping students' interactions with game-based learning environments. Prior work has focused on a range of computational modeling tasks, including probabilistic representations for user knowledge modeling (Rowe and Lester 2010) and machine learning frameworks for driving characters' affective behaviors (Robison, McQuiggan, and Lester 2009).

Student interactions with CRYSTAL ISLAND are comprised of a diverse set of actions occurring throughout the seven major locations of the island's research camp: an *infirmary*, a *dining hall*, a *laboratory*, a *living quarters*, the *lead scientist's quarters*, a *waterfall*, and a large *outdoors* region. Students can perform actions that include the following: *moving around the camp*, *picking up* and *dropping objects*, *using the laboratory's testing equipment*, *conversing with virtual characters*, *reading microbiology-themed books* and *posters*, *completing a diagnosis worksheet*, *labeling microscope slides*, and *taking notes*. Students advance through CRYSTAL ISLAND's non-linear narrative by completing a partially ordered sequence of goals that comprise the scenario's plot. Seven narrative goals are considered in this work: *speaking with the camp nurse* about the spreading illness, *speaking with the camp's virus expert*, *speaking with the camp's bacteria expert*, *speaking with a sick patient*, *speaking with the camp's cook* about recently eaten food, *running laboratory tests on contaminated food*, and *submitting a complete diagnosis* to the camp nurse.

Figure 1. CRYSTAL ISLAND virtual environment.

The following scenario illustrates a typical interaction with CRYSTAL ISLAND. The scenario begins with the student's arrival at the research camp. The student approaches the first building, an infirmary, where several sick patients and a camp nurse are located. A conversation with the nurse is initiated when the student approaches the character and clicks the mouse. The nurse explains that an unidentified illness is spreading through the camp and asks for the student's help in determining a diagnosis. The conversation with the nurse takes place through a combination of multimodal character dialogue—spoken language, gesture, facial expression, and text—and student dialogue menu selections. All character dialogue is provided by voice actors and follows a deterministic branching structure.

After speaking with the nurse, the student has several options for investigating the illness. Inside the infirmary, the student can talk to sick patients lying on medical cots. Clues about the team members' symptoms and recent eating habits can be discussed and recorded using in-game note-taking features. Alternatively, the student can move to the camp's dining hall to speak with the camp cook. The cook describes the types of food that the team has recently eaten and provides clues about which items warrant closer investigation. In addition to learning about the sick team members, the student has several options for gathering information about disease-causing agents. For example, the student can walk to the camp's living quarters where she will encounter a pair of virtual scientists who answer

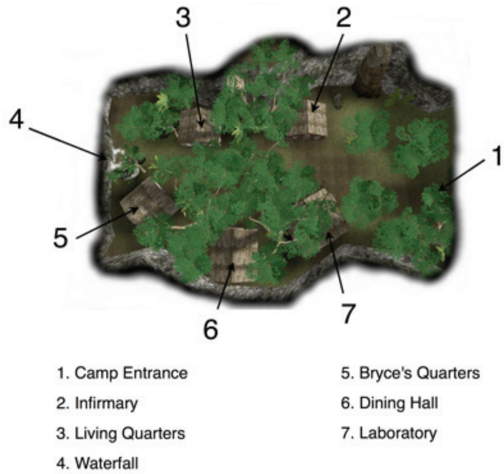

Figure 2. Map of the C*RYSTAL* I*SLAND research camp.*

questions about viruses and bacteria. The student can also learn more about pathogens by viewing posters hanging inside of the camp's buildings or reading books located in a small library. In this way, the student can gather information about relevant microbiology concepts using resources that are presented in multiple formats.

Beyond gathering information from virtual scientists and other instructional resources, the student can conduct tests on food objects using the laboratory's testing equipment. The student encounters food items in the dining hall and laboratory, and she can test the items for pathogenic contaminants at any point during the learning interaction. A limited number of tests are allocated to the student at the start of the scenario, but additional tests can be earned by answering microbiology-themed multiple-choice questions posed by the camp nurse.

After running several tests, the student discovers that the sick team members have been consuming contaminated milk. Upon arriving at this finding, the student is instructed to see the lab technician, Elise, for a closer look. The screen momentarily fades to black to indicate elapsing time, and Elise returns with an image of the contaminated specimen, which she explains was taken using a microscope. At this point, the student is presented with a labelling exercise where she must identity the contamination as bacterial in nature. After successfully completing this activity, the student can use the camp's books and posters in order to investigate bacterial diseases that are associated with symptoms matching those reported by the sick team members. Once she has narrowed down a diagnosis and recommended treatment, the student returns to the infirmary in order to inform the camp nurse. If the student's diagnosis is incorrect, the nurse identifies the error and recommends that the student keep working. If the student correctly diagnoses the illness and specifies an appropriate treatment, the mystery is solved.

All student actions are logged by the CRYSTAL ISLAND software and stored for later analysis. The data used for creating the MLN goal recognition system was collected from a study involving the eighth grade population of a public middle school. There were 153 participants in the study. Data for sixteen of the participants was removed from the analysis due to incomplete data or prior experience with CRYSTAL ISLAND. Participants whose data was included had no prior experience with the software.

MLN-based Goal Recognition

Following previous work on goal recognition (Blaylock and Allen 2003), our work defines goal recognition as the task of predicting the most likely goal for a given sequence of observed player actions in the game environment. The current work assumes that a given sequence of actions maps to a single goal, and no interleaving occurs between actions associated with different goals. Under these conditions, goal recognition is cast as a classification problem, in which a learned classifier predicts the most likely goal associated with the currently observed player action. The player's actions in the game environment are encoded using the following three properties:

- **Action Type**: The type of current action taken by the player, such as *moving to a particular location*, *opening a door*, and *testing an object using the laboratory's testing equipment*. To avoid data sparsity issues, only the predicate (e.g., *OPEN*) of the action is considered, ignoring the associated arguments (e.g., *laboratory-door*). Our data includes 19 distinct types of player actions.
- **Location**: The location in the virtual environment where a current player action is taken. This includes 39 fine-grained and non-overlapping sub-locations that decompose the seven major camp locations in CRYSTAL ISLAND.
- **Narrative State**: An indication of the player's progress in solving the narrative scenario. Narrative state is represented as a vector of four binary variables. Each variable represents a milestone event within the narrative. The four milestone events are: *Discuss the illness with the nurse*, *Test the contaminated object, Submit a diagnosis to the nurse,* and *Submit a correct diagnosis to the nurse*. If a milestone event has been accomplished, the associated variable is assigned a value of 1. Otherwise the value of the variable is 0.

The data exhibits two sources of complexity that pose modeling challenges. First, individual goals are not independent of one another. Goals represent key problem-solving steps in the science mystery, and some goals are naturally completed in rapid succession to other goals. In particular, the island's layout can lead to co-occurrence patterns among goals. Thus, goals should be inferred in relation to one another rather than in isolation. Second, players are not provided explicit goals to achieve; players learn about the goals while interacting with the virtual environment. As a consequence, player actions may not be motivated by well-defined goals in all cases. Instead, causality between player actions and goals is potentially cyclical. A goal could influence a player's next action if she has a particular goal in mind. However, it is also possible that the player's current action will influence which goal will be pursued next. This would occur when the player does not know what the next goal will be, but the current action reveals it (e.g., the player enters a new location and observes a character that she can engage in conversation).

To address these challenges, the current work employs *Markov logic networks (MLNs)* (Richardson and Domingos 2006). MLNs are a type of statistical relational learning framework, and they are well suited for machine learning tasks in domains with complex associations between modeled entities. MLNs encode undirected probabilistic graphical models with structures that are determined by first-order logic formulae and associated weights. In contrast to directed graphical models (e.g., Bayesian networks, hidden Markov models), undirected graphical models are well suited for representing cyclical relationships between entities. In addition, MLNs leverage the full representational expressiveness of first-order logic. This capability contrasts with traditional machine learning frameworks that rely on propositional expressions, such as hidden Markov models (Rabiner 1989). To illustrate the representational benefits of MLNs, a standard hidden Markov model would require 11,856 observation symbols (19 actions x 39 locations x 16 narrative states) to represent our goal recognition domain. This would lead to prohibitive data sparsity when using a player-generated training corpus. With the expressive power of first-order logic in MLNs, the data can be modeled more compactly. Our domain was represented in a MLN using 74 symbos (19 actions + 39 locations + 16 narrative states) and 13 logic formaulae.

MLNs have recently been applied to tasks that are related to goal recognition, such as probabilistic abduction for plan recognition (Kate and Mooney 2009) and multi-agent activity recognition (Sadilek and Kautz 2010). The following sections describe how our goal recognition data was represented using MLNs.

Background

Markov logic (ML) combines first-order logic with probabilistic graphical models (Richardson and Domingos 2006). In contrast to traditional first-order logic, in which possible worlds are assigned a binary value (*true* when a world satisfies all the logic formulae, *false* otherwise), ML allows certain logic formulae to be violated in a given world, by associating a weight to each logic formula. Thus, while the logic formulae in traditional first-order logic are hard constraints over possible worlds, the weight-associated logic formulae in ML represent soft constraints. The weight reflects the strength of the constraint represented by the associated logic formula.

A *Markov Logic Network (MLN)* consists of a set of weighted first-order logic formulae written in Markov logic. Together with constants that represent objects in the domain, an MLN defines a *Markov network*, an undirected graphical model whose nodes have a Markov property described by the structure of the graph. Each ground predicate in a MLN has a corresponding binary node in the Markov network and each ground logic formula is a feature.[1] The value of a binary node is 1 if the ground predicate is true; otherwise it is 0. Similarly, the value of a feature is 1 if the ground logic formula is true, otherwise 0. An MLN defines a probability distribution over possible worlds. The probability of a possible world x in a MLN is given by equation (1), in which F represents the number of logic formulae in the MLN, w_i is the weight of the i^{th} logic formula f_i, $n_i(x)$ is the number of true groundings of the formula f_i in the given world x, and Z is a normalization constant (Richardson and Domingos 2006). The weights of the logic formulae are learned from training data. An MLN can be viewed as a template by which one can construct Markov networks.

$$P(X = x) = \frac{1}{Z}\exp\left(\sum_{i=1}^{F} w_i n_i(x)\right) \quad (1)$$

MLN for Goal Recognition

We first defined a set of predicates as the basic building blocks to represent the proposed MLN for goal recognition. There are two types of predicates, *observed* and *hidden*. Observed predicates are those that are fully observable while a player is interacting with the game environment. In contrast, hidden predicates are those that are not directly observable by the game environment. Instead, the groundings of the hidden predicates are predicted from the groundings of the observed predicates based on a learned model. In other words, hidden predicates represent the target phenomena to be modeled with the MLN. Table 1 shows the observed and the hidden predicates in our MLN. The three observed predicates, *action(t, a), loc(t, l),* and *state(t, s)*, are the properties that characterize player actions as described earlier in this

[1] MLNs make classification decisions based on the values of the features.

Predicate		Description
Observed	*action(t, a)*	Player takes an action *a* at time *t*.
	loc(t, l)	Player is at a location *l* at time *t*.
	state(t, s)	The narrative state at time t is s.
Hidden	*goal(t, g)*	Player pursues a goal *g* at time *t*.

Table 1. Observed and Hidden Predicates

section. The hidden predicate *goal(t, g)* represents the player's goal at a given time *t*.

By combining these observed and hidden predicates with logical operations, a total of 13 logic formulae were constructed. As an example, formula (2) considers the relation between actions and goals. Given an observed action *a* at time *t*, this formula predicts that the player's goal at time *t* will be *g*. However, it is unrealistic to assume a one-to-one mapping between actions and goals. To allow potential many-to-many mappings between actions and goals, the formula was defined as a soft constraint by assigning a weight function to it. Analogous soft constraints were defined for user location and game state as well. Formula (3) reflects an implication that the same action could imply a different goal at a different narrative state. Formula (4) considers the previous player action as well as the current action. These were also defined as soft constraints. In addition to the soft constraints, our MLN includes one hard constraint, which is represented by Formula (5). This hard constraint asserts that there should exist exactly one goal per action at any given time.

$$\forall t,a,g : action(t,a) \Rightarrow goal(t,g) \quad (2)$$
$$\forall t,a,s,g : action(t,a) \wedge state(t,s) \Rightarrow goal(t,g) \quad (3)$$
$$\forall t,a_1,a_2,g : action(t,a_1) \wedge action(t-1,a_2) \Rightarrow goal(t,g) \quad (4)$$
$$\forall t,a : action(t,a) \Rightarrow |\exists g : goal(t,g)| = 1 \quad (5)$$

The weights for the soft constraints were learned by using Cutting Plane Inference (CPI) in *Markov theBeast*, an off-the-shelf tool for MLNs (Riedel 2008). CPI is a maximum a posteriori inference (MAP) technique for Markov logic, which instantiates a small fraction of a given complex MLN incrementally and solves it using a conventional MAP inference method such as integer linear programming (ILP) and MaxWalkSAT. CPI has been shown to improve the efficiency of MAP inference compared to conventional methods alone. As the base inference method for the proposed goal recognition model, ILP was used, which provides exact solutions to the MLN.

Figure 3 shows a partial graphical representation of the described MLN. Shaded and clear nodes indicate hidden and observed predicates, respectively. Two nodes are connected with an undirected arc when they appear together in at least one of the MLN formulae.

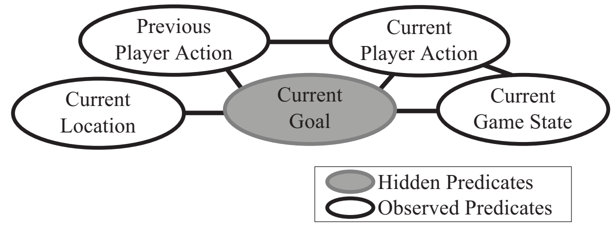

Figure 3. Graphical Representation of the proposed MLN

Evaluation

To train and test the proposed MLN, the data from the observation corpus was processed in several steps. First, all player actions that achieve goals were identified. Second, all actions in the observation sequence that precede the current goal but follow the previous goal were labeled with the current goal. Third, actions that achieve goals were removed from the data. Removing goal-achieving actions was necessary to ensure that model training was fair, because it would be trivial to predict goals from the goal-achieving actions. Finally, all actions that were taken after achievement of the last goal were ignored, since those actions have no direct mapping to any goal. Summary statistics about the training data are shown in Table 2. Table 3 shows the set of goals considered in this work and their distribution in the processed corpus data.

Total Number of Observed Player Actions	77182
Total Number of Goals Achieved	893
Average Number of Player Actions per Goal	86.4

Table 2. Statistics for Observed Actions and Goals

Running laboratory test on contaminated food	26.6%
Submitting a diagnosis	17.1%
Speaking with the camp's cook	15.2%
Speaking with the camp's bacteria expert	12.5%
Speaking with the camp's virus expert	11.2%
Speaking with a sick patient	11.0%
Speaking with the camp nurse	6.4%

Table 3. Distribution of Goals

For evaluation, the proposed MLN was compared to one trivial and two non-trivial baseline systems. The trivial baseline was the majority baseline, which always predicted the goal that appears most frequently in the training data. The non-trivial baselines included two *n*-gram models, *unigram* and *bigram.* The unigram model predicted goals based on the current player action only, while the bigram model considered the previous action as well. In the *n*-gram models, player actions were represented by a single aggregate feature that combines all three action properties: action type, location, and narrative state. Although simplistic, the *n*-gram models have been shown to be effective for goal recognition in spoken dialogue systems (Blaylock and Allen 2003) and interactive narrative game environment (Mott, Lee, and Lester 2006). In their work, both unigram and bigram models achieved higher prediction accuracies than a more sophisticated Bayesian Network model.

The *n*-gram comparison models were created in ML. The unigram model was represented with the single weighted formula (6). The weighted formula defined for the bigram model was similar but considered two consecutive player actions at the same time.

$$\forall t,a,l,s,g : action(t,a) \wedge location(t,l) \wedge state(t,s) \Rightarrow goal(t,g) \qquad (6)$$

The two *n*-gram models and the proposed MLN model were trained using one-best MIRA (Crammer and Singer 2003) as the update rule that is provided by *Markov theBeast*. The three models were evaluated with ten-fold cross validation on the entire data set. The models' performance was measured using *F1*, which is the harmonic mean of *precision* and *recall*. It should be noted that in our data, each observed player action is associated with a single goal. The goal recognition model predicts the most likely goal for each player action. Thus, the values of *precision*, *recall*, and *F1* are the same. Table 4 shows the average performance of each model over ten-fold cross validation. The MLN model scored 0.484 in *F1*, achieving 82% improvement over the baseline. The unigram model performed better than the bigram model. A one-way repeated-measures ANOVA confirmed that the differences among the three compared models were statistically significant ($F(2,18) = 71.87$, $p < 0.0001$). A post hoc Tukey test revealed the differences between all pairs of the three models were statistically significant ($p < .01$).

	Baseline	Unigram	Bigram	MLN
F1	0.266	0.396	0.330	**0.484**
Improvement over Baseline	N/A	49%	24%	**82%**

Table 4. F1 scores for MLN and baseline goal recognition models.

Discussion

While all three models performed better than the baseline, the MLN model achieved the best performance, suggesting that the proposed MLN goal recognition framework is effective in predicting player goals from actions in a complex game environment. The *F1* score of 0.484 may appear low. However, this result is encouraging given the challenges posed by the data. The chance probability of accurately recognizing the seven goals in our data is 0.143. The MLN model's improved performance compared to the *n*-gram models can be partially explained by the MLN's relational learning framework, which facilitates explicit modeling of associations between goals. Furthermore, the structural flexibility of undirected graphical models, which permit cyclical relations, enables MLNs to model richer relations between actions and goals than *n*-gram models. The finding that the unigram model achieved higher performance than the bigram model is consistent with the result reported by Mott, Lee, and Lester (2006). Among the possible reasons for the unigram model's superiority over the bigram model is data sparsity. The bigram model considers two consecutive previous goals, which would lead to each training instance for each bigram becomes sparser than in the unigram model.

Inducing accurate goal recognition models has several prospective benefits for current work on CRYSTAL ISLAND. First, goal recognizers can be used to inform player-adaptive decisions by *narrative-centered tutorial planners,* which comprise a particular class of drama managers that simultaneously reason about interactive narrative and pedagogical issues. Data-driven approaches to narrative-centered tutorial planning are the subject of active research by the CRYSTAL ISLAND research team. They offer a method for dynamically tailoring events during students' game-based learning experiences in order to individualize pedagogical scaffolding and promote student engagement. Second, goal recognizers can be used during data mining to inform the analysis and design of future iterations of the CRYSTAL ISLAND software. By automatically recognizing players' goals, and identifying which actions are likely to be associated with those goals, researchers can gain insights into common types of problem-solving paths and challenges encountered by students. Finally, recognizing players' goals will enrich in-game assessments of student learning and problem solving, which is a critical challenge for the serious games community.

Conclusions

Effective goal recognition holds considerable promise for player-adaptive games. Accurately recognizing players' goals enables digital games to proactively support gameplay experiences that feature nonlinear scenarios while preserving cohesion, coherence and believability. This paper has introduced a goal recognition framework based on Markov logic networks that accurately recognizes players' goals. Using model parameters learned from a corpus of player interactions in a complex, nonlinear game environment, the framework supports the automated acquisition of a goal recognition system that outperforms three baseline models.

Acknowledgements

This research was supported by the National Science Foundation under Grant DRL-0822200. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Armentano, M. G. and Amandi, A. 2009. Goal Recognition with Variable-Order Markov Models. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, 1635-1640. Menlo Park: AAAI Press.

Bethesda Softworks. 2008. *Fallout 3*. Retrived from http://fallout.bethsoft.com/.

Blaylock, N. and Allen, J. 2003. Corpus-based, Statistical Goal Recognition. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 1303-1308. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Bui, H. H. 2003. A General Model for Online Probabilistic Plan Recognition. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 1309-1315. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Carberry, S. 2001. Techniques for Plan Recognition. *User Modeling and User-Adapted Interaction*, 11(1-2): 31-48.

Charniak, E. and Goldman, R. P. 1993. A Bayesian Model of Plan Recognition. *Artificial Intelligence*, 64(1): 53-79.

Crammer, K. and Y. Singer. 2003. Ultraconservative online algorithms for multiclass problems. *The Journal of Machine Learning Research*, 3: 951-991.

Eidos Interactive and Warner Bros. Interactive. 2009. *Batman: Arkham Asylum*. Retrieved from http://www.batmanarkhamasylum.com/.

Geib, C. W. and Goldman, R. P. 2009. A Probabilistic Plan Recognition Algorithm Based on Plan Tree Grammars. *Artificial Intelligence*, 173(11): 1101-1132.

Gold, K. 2010. Training Goal Recognition Online from Low-Level Inputs in an Action-Adventure Game. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Interactive Digital Entertainment*, 21-26. Menlo Park, CA: AAAI Press.

Hu, D. H. and Yang, Q. 2008. CIGAR: Concurrent and Interleaving Goal and Activity Recognition. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence*, 1363-1368. Menlo Park, CA: AAAI Press.

Jennings-Teats, M., Smith, G. and Wardrip-Fruin, N. (2010). Polymorph: A Model for Dynamic Level Generation. In *Proceedings of the 6th Artificial Intelligence and Interactive Digital Entertainment Conference*, 138–143. Menlo Park, CA: AAAI Press.

Kabanza, F., Bellefeuille, P. and Bisson, F. 2010. Opponent Behavior Recognition for Real-Time Strategy Games. In *Proceedings of the AAAI-10 Workshop on Plan, Activity, and Intent Recognition*. Atlanta, GA.

Kate, R. J. and Mooney, R. J. 2009. Probabilistic Abduction using Markov Logic Networks. In *Proceedings of the IJCAI-09 Workshop on Plan, Activity, and Intent Recognition*. Pasadena, CA.

Kautz, H. and Allen, J. F. 1986. Generalized Plan Recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, 32-37. Menlo Park, CA: Morgan Kaufmann.

Li, B., and Riedl, M. O. (2010). An Offline Planning Approach to Game Plotline Adaptation. In *Proceedings of the 6th Artificial Intelligence and Interactive Digital Entertainment Conference*, 45-50. Menlo Park, CA: AAAI Press.

Mateas, M. and Stern, A. 2005. Structuring Content in the Façade Interactive Drama Architecture. In *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*, 93-98. Menlo Park, CA: AAAI Press.

Mojang. 2009. *Minecraft*. Retrieved from http://www.minecraft.net/.

Mott, B., Lee, S., and Lester, J. 2006. Probabilistic Goal Recognition in Interactive Narrative Environments. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 187-192. . Menlo Park, CA.: AAAI Press.

Pynadath, D. V. and Wellman, M. P. 2000. Probabilistic State-Dependent Grammars for Plan Recognition. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 507-514. Stanford, CA: Morgan Kaufmann Publishers Inc.

Rabiner, L. R.1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77 (2): 257-286.

Ramirez, M. and Geffner, H. 2010. Probabilistic Plan Recognition using off-the-shelf Classical Planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1121-1126. . Menlo Park, CA: AAAI Press.

Richardson, M. and Domingos, P. 2006. Markov Logic Networks. *Journal of Machine Learning,* 62(1-2): 107-136.

Riedl, M. O., Stern, A., Dini, D. M., and Alderman, J. M. 2008. Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *International Transactions on Systems Science and Applications*, Special Issue on Agent Based Systems for Human Learning, 3(1): 23-42.

Riedel, S. 2008. Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *Proceedings of the Twenty-Fourth Annual Conference on Uncertainty in Artificial Intelligence*, 468-475. Helsinki, Finland: AUAI Press.

Roberts, D. L., Cantino, A. S., and Isbell, C. L. (2007). Player Autonomy Versus Designer Intent: A Case Study of Interactive Tour Guides. In *Proceedings of the 3rd Artificial Intelligence and Interactive Digital Entertainment Conference*, 95-97. . Menlo Park, CA: AAAI Press.

Robison, J., McQuiggan, S. and Lester, J. (2009). Evaluating the Consequences of Affective Feedback in Intelligent Tutoring Systems. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction,* 37-42. IEEE.

Rockstar Games. 2008. *Grand Theft Auto 4*. Retrieved from http://www.rockstargames.com/IV/.

Rowe, J. and Lester, J. 2010. Modeling User Knowledge with Dynamic Bayesian Networks in Interactive Narrative Environments. In *Proceedings of the Sixth International Conference on Artificial Intelligence in Interactive Digital Entertainment,* 57-62. Menlo Park, CA: AAAI Press.

Rowe, J., Shores, L., Mott, B., and Lester, J. (2010). Integrating Learning and Engagement in Narrative-Centered Learning Environments. In *Proceedings of the Tenth International Conference on Intelligent Tutoring Systems*, 166-177. Heidelberg, Germany: Springer.

Sadilek, A. and Kautz, H. 2010. Recognizing Multi-Agent Activities from GPS Data. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1134-1139. Menlo Park, CA: AAAI Press.

Shaker, N., Yannakakis, G. and Togelius, J. (2010). Towards Automatic Personalized Content Generation for Platform Games. In *Proceedings of the 6th Artificial Intelligence and Interactive Digital Entertainment Conference*, 63–68. Menlo Park, CA: AAAI Press.

Thue, D., Bulitko, V., Spetch, M., and Webb M. (2010). Socially Consistent Characters in Player-Specific Stories. In *Proceedings of the 6th Artificial Intelligence and Interactive Digital Entertainment Conference*, 198-203. Menlo Park, CA: AAAI Press.