Model-Based Reinforcement Learning under Periodical Observability

Richard Klima,[1] **Karl Tuyls,**[1] **Frans A. Oliehoek**[1]
[1]University of Liverpool, United Kingdom
{richard.klima, k.tuyls, frans.oliehoek}@liverpool.ac.uk

Abstract

The uncertainty induced by unknown attacker locations is one of the problems in deploying AI methods to security domains. We study a model with partial observability of the attacker location and propose a novel reinforcement learning method using partial information about attacker behaviour coming from the system. This method is based on deriving beliefs about underlying states using Bayesian inference. These beliefs are then used in the QMDP algorithm. We particularly design the algorithm for spatial security games, where the defender faces intelligent and adversarial opponents.

Introduction and Motivation

In security domains we often face several uncertainties which make acting effectively very difficult. Overcoming the uncertainties is one of the main challenges in order to deploy AI techniques in real-world applications. The reasoning agent has often an access to extra information about the environment which if used properly can help significantly in effective strategy-making. In security games this knowledge can come from several types of surveillance available to the agent. We focus on a model-based approach, where we continually learn and improve our knowledge about the opponent behaviour. The main uncertainty lies in not being able to always observe the opponent location. To tackle this challenge we develop a statistical probability model to enable us to reason about opponent location. We base opponent location modelling on observed frequencies of transition tuples and prior information about the environment e.g. target location. Our proposed algorithm is based on the QMDP (Littman, Cassandra, and Kaelbling 1995) algorithm, which combines the standard Q-learning with belief states in partially observable domains. We extend this algorithm with Bayesian inference update using prior information about the environment.

We describe our work in terms of a taxonomy proposed in (Hernandez-Leal et al. 2017), where the authors discuss a classification in terms of environment observability, opponent adaptation capabilities and how the agent deals with non-stationarity. We assume observability of the agent's local reward and partial observability of opponent's actions. The opponent is assumed to adapt his strategy within some bounds, thus we restrict his behaviour from abrupt/drastic changes. This is explained by the concept of bounded rationality, which is often used in security games (Pita et al. 2010). Such a concept allows us to learn a model of opponent behaviour and use it to form the defender strategy.

This paper is motivated by the domain of Green Security Games (Fang, Stone, and Tambe 2015), with a focus on the problem of Illegal Rhino Poaching (Montesh 2013) and on ways how to learn effective ranger strategies in order to mitigate rhinos killings. Nevertheless, our proposed method is applicable to other spatial security game scenarios which can be modelled on a grid (graph). The problem belongs to a domain of pursuit-evasion games. There has been a lot of work on computing exact solutions and describing their theoretical properties in security games, mostly using the equilibria concepts e.g. Nash equilibria or Stackelberg equilibria (Korzhyk et al. 2011). This line of research has been important as a theoretical underpinning of the field, however, these methods are often difficult to deploy in real world settings due to some strict assumptions or severe simplifications. A different approach from computing exact solution strategy is to learn the strategy from interacting with the environment. This approach helps to overcome some of the assumptions of the theoretical approaches.

The domain of security games can be modelled as a reward-based system, where the agents obtain rewards and thus can learn strategies. The problem can be approached by Multi-agent Reinforcement Learning (MARL) using the Markov Decision Process (MDP) framework. In MARL it is very difficult to learn optimal strategies because of the *moving target* problem (Tuyls and Weiss 2012), where all agents are assumed to be adapting to each others behaviour. In security games we face an additional complexity caused by the uncertainty about the attacker, who can be intelligent and strategic. One of the possible uncertainties about the attacker is his location, which might not be observable or only partially observable. We focus on a special case of partial (limited) observability which is inspired by the board game *Scotland Yard* where the player gets to observe the opponent location only periodically e.g. every 3 time steps. We claim that this type of observability is quite common in security domains where the defender gets to observe an opponent location by obtaining some extra information. For instance

in the green security game scenarios like Rhino Poaching problem, the rangers can be informed by the villagers living nearby about the current location of the poachers, or this information can also come from surveillance by drones (Montesh 2013). In our model we assume an adversarial adaptive opponent who might be able to observe the defender behaviour. Our main goal is to make use of the extra information about the attacker location in reinforcement learning, obtaining an adaptive strategy to apprehend the attacker.

Related Work

This paper is situated in the field of Multi-agent Reinforcement Learning (MARL), which is a very active field of research since there has been substantially less work done in MARL compared to single-agent RL due to the increased complexity. For more information on MARL we refer the reader to surveys (Bloembergen et al. 2015) or (Hernandez-Leal et al. 2017). We divide this section into several fields of research, which are closely related to this paper. These consist of Partially Observable Markov Decision Processes (POMDP), Bayesian Reinforcement Learning and Security Games (SG). We state the related work respectively.

Partially observable problems are often modelled as Partially Observable Markov Decision Processes (POMDP) (Kaelbling, Littman, and Cassandra 1998). Related to our work is algorithm BA-POMDP proposed in (Ross et al. 2007), where the authors combine Bayesian approach with POMDP model or the learning version BA-POMCP (Katt, Oliehoek, and Amato 2017). We also mention Bayesian Q-learning proposed in (Dearden, Friedman, and Russell 1998), which uses Bayesian inference combined with Q-learning to model the value function. The domain of Bayesian learning can be divided into probabilistic modelling of transition function, value function, reward function or policy. In this paper we focus on probabilistic modelling of transition function. We also propose a combination of Bayesian approach and Q-learning, however in substantially different way. Our method uses Bayesian approach to model transition function to derive belief states, modelling the partially observable attacker behaviour.

Security games have gained a lot of attention in recent years due to their successful application on real-world security threats. Examples include the ARMOR system for airport security (Pita et al. 2008) or the PROTECT system for scheduling Coast Guard (Shieh et al. 2012). Additionally some work has focused on Green Security Games for poaching problems (Fang, Stone, and Tambe 2015) or Border Patrol (Klima, Lisy, and Kiekintveld 2015). Some of these security games however, do not consider space or time, i.e. the time it takes the defender to travel to the target node, as part of the model. Recently, reinforcement learning has been applied to spatial security games (Klima, Tuyls, and Oliehoek 2016) to tackle the spatial component. Spatial security games are also often modelled as extensive form games (Korzhyk et al. 2011). There has been lot of work in computing the optimal strategies online or offline, especially for zero-sum games (Bosansky et al. 2016), (Jain et al. 2011). We also mention the work of (An et al. 2012), which computes the optimal defender strategy to a learning attacker who can only partially observe the defender and updates his beliefs using Dirichlet distribution. In this paper we assume the attacker can fully observe the defender past moves and plays fictitious play (Fudenberg and Levine 1996). We address this by learning the defender strategy. Fictitious play is well-defined in 1D space, but it is more complicated in 2D space. Recently, (Heinrich, Lanctot, and Silver 2015) showed the extension of fictitious play into extensive form games implemented in behavioural strategies with similar properties as the original fictitious play.

Model

We study the problem of effective decision making in spatial security games. Our focus is a spatial security game played on a graph with two non-cooperative players with opposing (not strictly, assuming general-sum game) goals. We define these two players as the defender and the attacker. In this work we use the terms defender/agent and attacker/opponent interchangeably. The model is inspired by the Green Security Game framework where we are interested in the problem of Illegal Rhino Poaching. In such a problem the rangers (the defender) tries to apprehend illegal rhino poachers (the attacker) and thus protect the rhinos (targets) from being poached. The environment is a wildlife reservation, which can be modelled as a graph (grid).

We define this framework in terms of Stochastic game (Shapley 1953) using Markov Decision Process (MDP) model. A state is defined as a combination of locations of the defender and the attacker in the grid, an action is defined for the defender as moving from one place in the grid to another and a reward is defined as a positive signal for apprehending the attacker. A Stochastic (Markov) game as described in (Wiering and van Otterlo 2013) chapter 14.3.1. is defined as a tuple $(n, S, A_1 \dots A_n, R_1 \dots R_n, T)$ where n is number of agents in the system, S is a finite set of system states, A_k is the action set of agent k, $R_k : S \times A_1 \times ... \times A_n \to \mathbb{R}$ is the reward function of agent k and $T : S \times A_1 \times ... \times A_n \times S \to [0, 1]$ is the transition function.

Observability in spatial security game

In our security game we assume that the defender can always observe his own location but sometimes cannot observe the attacker location, thus cannot fully observe the underlying state. Agent's observations consist of either full observation of the state or an observation of only own location. Therefore, the defender needs to maintain beliefs $b(s)$ over states which give him the probability of being in a state s. In every time step we restrict the set of possible states by (i) physical structure of the map (gridworld) and (ii) by observations of attacker location in previous time steps. We use the notion of *information set* from extensive-form game theory to denote such restricted set of possible states. We define such a restricted state space as a subset of the original state space denoted $\bar{S} \subseteq S$.

In Figure 1 we show an example of a small grid world and corresponding extensive-form tree with information set. The

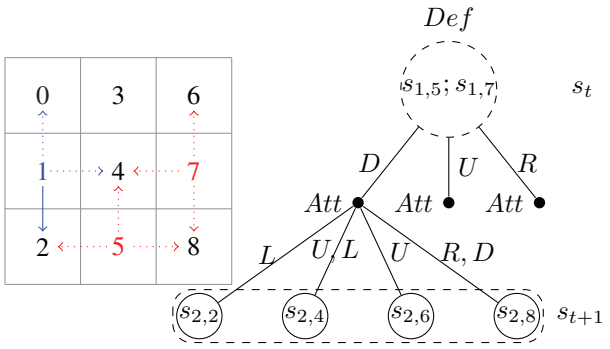


Figure 1: Example of states in information sets for the case where neither the current state nor the succeeding state is observed. We need to reason over two information sets at time t and $t+1$. The defender is in location 1 and chooses action *down* (D), the attacker is in location 5 or 7.

defender is unsure about the state, it is either $s_{1,5}$ or $s_{1,7}$, because the defender can always observe his own location (tile 1) but might be unsure about the attacker location (either tile 5 or tile 7). The figure captures a decision point, where the defender decides to go *down* (D). The defender reasons about the possible attacker actions and about the resulting attacker location in order to form the information set. We update the beliefs only over the states in given information set.

We study a scenario with a periodical observability i.e. the defender gets to observe the attacker location every k steps. This type of observability is inspired by the board game *Scotland Yard*. We compare this type of observability with a full observability of the attacker location and partial observability i.e. knowing only agent's own position.

Attacker behaviour model

In security domains we often face an adversarial opponent who is potentially intelligent and can observe the defender behaviour to some extent and plan his strategy accordingly. In our model the attacker plays a version of fictitious play (Fudenberg and Levine 1996), considering an intelligent and adaptive opponent. We assume that the attacker can observe all the past moves of the defender. This assumption is rather strong but describes the worst-case scenario in security games. We also choose the fictitious play because of its properties. It is a best response to defender past moves and is guaranteed to converge to Nash equilibrium in some games (e.g. zero-sum games (Robinson 1951)).

Statistical approach to uncertainty

We assume that both players know the environmental model i.e. state space, action space and reward function. However the defender is uncertain about the location of the opponent and his strategy. Our main goal is to act efficiently under this uncertainty. In security games the defender has often access to some extra information about the attacker whereabouts, which we use to deal with this uncertainty.

We define a discrete random variable X in the restricted space $\bar{S}'$ of the succeeding states given by the information set. Thus, we have a discrete probability distribution of the succeeding states $P(X = s') : \forall s' \in \bar{S}'$ parametrized by a vector θ, where $\sum_i^k \theta_i = 1$ and $P(X = s'|\theta) = \theta_i$. We assume that the defender can observe some of the transitions defined by a transition tuple (s, a, s'). The defender stores these transitions and form a vector $\Phi = (\phi_1, \ldots, \phi_k)$ of transition occurrences; for example $\phi_{s'}^{sa}$ is a number of past observations of a transition from state s taking action a to state s'.[1] The defender in our model forms beliefs about the possible states defined by the information set. The information set is build based either on a direct observation or on reasoning about previous attacker locations (see Figure 1). These beliefs are probabilities defined by the vector θ, e.g. θ_s is the probability of being in state s. The goal is to derive these probabilities given the past observed transitions, thus we need to compute the probability distribution $P(\theta|\Phi)$. Note that the total number of observations (of the succeeding states for given state and action) is $n = \sum_{\bar{S}'} \phi_i^{sa}$, note that $|\bar{S}'|$ is the size of the information set.

Firstly, we assume that probability distribution $P(\Phi|\theta)$ follows a multinomial distribution with parameters n and θ. Thus, we can write the probability mass function of multinomial distribution as:

$$P(\Phi|\theta) \sim f(\Phi|n, \theta) = \frac{n!}{\prod_{\bar{S}'} \phi_i^{sa}!} \prod_{\bar{S}'} \theta_i^{\phi_i^{sa}} \tag{1}$$

Note that $\frac{n!}{\prod_{\bar{S}'} \phi_i^{sa}!}$ is the total number of possible observation sequences giving the vector Φ.

The defender assumably has prior knowledge about the environment e.g. target location, which we use as a prior for Bayesian inference. We define the prior probability as a Dirichlet distribution $Dir(\alpha)$, which is defined for hyperparameters α. $Dir(\alpha)$ is a probability distribution over parameters θ of multinomial distribution and is also its conjugate prior. The hyperparameters α can be seen as pseudo-observations to complement the actual observed transitions i.e. the transition counts Φ. $Dir(\alpha)$ is defined using Γ function as:

$$Dir(\theta|\alpha) = \frac{\Gamma(\sum_i^k \alpha_i)}{\prod_i^k \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{2}$$

We already defined the likelihood as multinomial distribution using the transition counts Φ (see Equation 1) and thus, we can write the posterior using Bayes' rule as:

$$Dir(\theta|\Phi) \propto Multi(\Phi|n, \theta) Dir(\alpha) \tag{3}$$

We can then write $P(\theta|\Phi) = Dir(\Phi + \alpha)$. In this work we are not interested in the full posterior $P(\theta|\Phi)$, because we want only a point estimate to determine a belief about states of the model. We focus on the expected value of the distribution to obtain the belief given past observations of the transitions and prior information. The expected value of

[1] For transition counts we use notation ϕ following the previous work e.g. (Ross et al. 2007).

the posterior distribution is defined for multinomial likelihood and Dirichlet prior in Bayes rule as $E_{Dir(\Phi+\alpha)}[\theta_i] = \frac{\phi_i^{sa}+\alpha_i}{n+\sum_{j=1}^{k}\alpha_j}$. Note that when deriving point estimates of posterior distribution we do not need marginal distribution of data (normalizing constant) $P(\Phi)$.

We can now obtain the belief $b^{oa}(s')$ about the succeeding state s' given the observation o and action a. The observation gives us belief $b(s)$ about the state s, transition counts $\phi_{s'}^{sa}$ and priors α as:

$$b^{oa}(s') = \sum_{s\in\bar{S}} b(s) E_{Dir(\Phi+\alpha)}[\theta_{s'}] = \sum_{s\in\bar{S}} b(s)\frac{\phi_{s'}^{sa}+\alpha_{s'}}{n_{s\cdot}^{a}+\sum_{j\in\bar{S}'}\alpha_j} \tag{4}$$

where $n_{s\cdot}^{a}$ is the sum of all the observations for given state s and action a.

We now discuss the setting of the hyperparameters α. We believe that the attacker behaviour is steered by the location of the targets which is known information to both of the players at the beginning of the game. Therefore, prior for each node (location) is defined as $\alpha_{node} = \frac{1}{SP(node,target)+1} * priorConfidence$, where $SP(node, target)$ is the shortest path to the nearest target from the given node, $priorConfidence$ depends on number of observations and potentially other influences determining the confidence in comparison to actual observations. Note that the prior is defined for a location of the attacker ignoring the location of the defender. This simplification comes from the assumption that the attacker cannot fully observe the defender location in given game episode (but knows the past moves) and is mainly steered by location of the targets.

Saving transition counts in partial observability

The defender uses a model-based learning approach. In each time step he saves a transition tuple observed in the current transition. In the case he cannot fully observe the current or/and the succeeding state he updates the transition counts $\phi_{s'}^{sa}$ proportionally to the beliefs $\phi_{s'}^{sa} \mathrel{+}= b(s)b(s')$. Therefore, the stronger the belief about a particular state is the more he updates the corresponding value in the vector Φ. Note that for fully observed states s and s' the update is equal to 1.

Q-learning with Bayesian Inference

We combine the inference of probabilities of different states in given information set with standard temporal difference learning algorithm TD(0) i.e. Q-learning, where we use QMDP algorithm (Littman, Cassandra, and Kaelbling 1995). We present BayesQMDP in Algorithm 1. The action-selection on line 4 is ϵ-greedy proportional to the belief, meaning that the action a from state s is more likely to be chosen with increasing probability of being in the state s and increasing Q-value for that state and action. On line 5 we update Q-values using the belief about states $b(s)$. The learning rate λ is linked to the belief we have about the state; the less certainty (lower probability) about being in the state the less we update the Q-value and vice-versa (smaller learning rate).[2] The value function on line 6 is a sum over maximal Q-values of the succeeding states weighted by the probability (belief) of going to those states. The belief update on line 7 uses the expected value of the posterior probability distribution as explained in Equation 4. Finally, on line 8 we update the transition count vector $\phi_{s'}^{sa}$.

Algorithm 1 BayesQMDP

1: **Input:** priors α, parameters λ, γ
2: **Init:** $s_0, Q(s,a) = 0$, $\phi_{s'}^{sa} = 0\,\forall s, s' \in S\,\forall a \in A$
3: **for** t in game **do**
4: $\quad\epsilon$-greedy: $a = \arg\max_a \sum_{\bar{S}} b(s) * Q(s,a)$
5: $\quad\forall s$: $Q(s,a) = (1 - b(s)\lambda)Q(s,a) + b(s)\lambda(r + \gamma V(s'))$
6: $\quad$where $V(s') = \sum_{\bar{S}'} b(s') \max_a Q(s',a)$
7: $\quad b^{oa}(s') = \sum_{\bar{S}} b(s)\frac{\phi_{s'}^{sa}+\alpha_{s'}}{n_{s\cdot}^{a}+\sum_{\bar{S}'}\alpha_j}$
8: $\quad\phi_{s'}^{sa} \mathrel{+}= b(s) * b(s')$

Experiments

In this section we compare the proposed BayesQMDP with two baseline algorithms based on standard Q-learning. We show two different gridworlds.

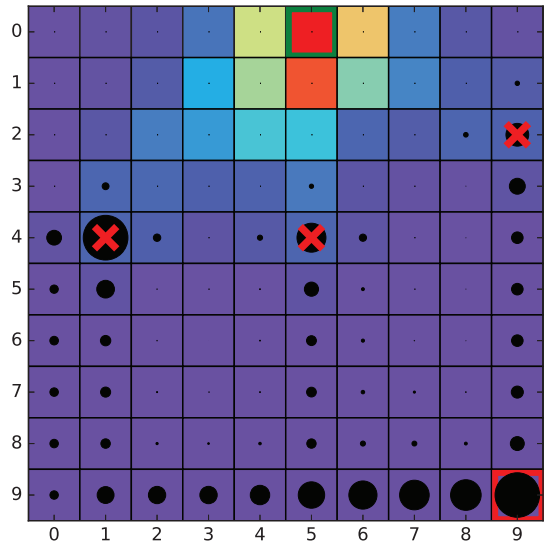
Security game gridworld

We perform the experiments on a grid of size 10x10, thus the state space has size 100^2 i.e. 100 possible locations for each player (as explained before a state is defined by the location of the defender and the attacker). The defender starts on top in the middle and the attacker starts in the right bottom corner. In our model the attacker chooses a best response to defender past locations in the grid world, which is the shortest path from start node to target location weighted by defender visits in each node over all the targets. The attacker chooses his path at the beginning of every episode. Every target has some probability p of success; for example once the poacher (attacker) reaches the target, he has p probability of poaching a rhino in which case the game ends. If the attacker gets to a target (e.g. area with a rhino) and is not successful, he makes a random move from the target node and tries again in the next time step.

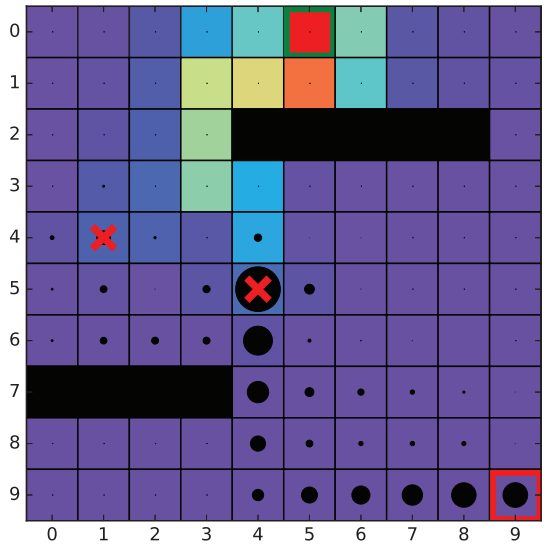
We present experiments with two and three targets, each with probability $p = 0.3$ of successful attack. If the defender is in the same location as the attacker, the attacker is apprehended and the defender receives a positive reward. If the defender apprehends the attacker or the attacker successfully attacks a target, the game (episode) ends. As a performance metric we use the percentage of defender wins i.e. the percentage of attacker apprehensions.

In our experiments we compare the proposed algorithm BayesQMDP with two baselines. The first baseline is a standard Q-learning with full observability of the attacker. The second baseline is also a standard Q-learning but this time

[2] For learning rate we use λ instead of the common notation α to distinguish from the hyperparameter.



(a) Gridworld 1: three targets



(b) Gridworld 2: two targets with obstacles

Figure 2: 10x10 gridworlds, targets depicted by red crosses, defender starts at position [0,5] (green square), attacker starts at position [9,9] (red square). The heatmap shows defender visits in each tile and the black dots show attacker visits in each tile (size of the black dots represents the number of visits).

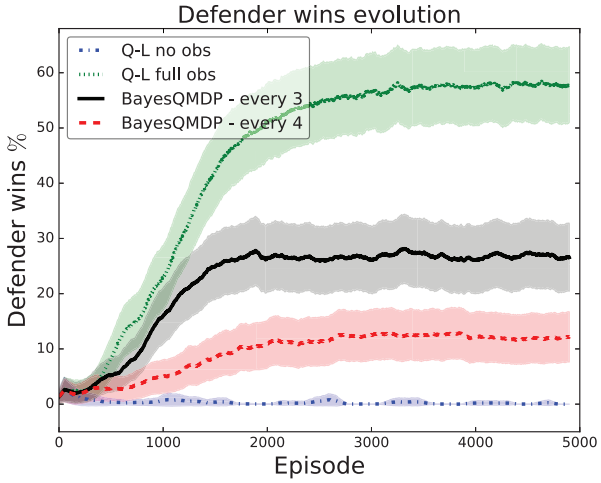


Figure 3: Defender wins for BayesQMDP - Gridworld 1

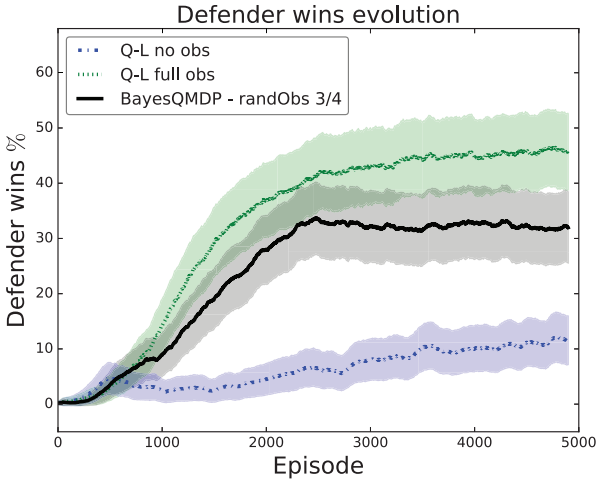


Figure 4: Defender wins for BayesQMDP - Gridworld 2

with no observability of the attacker. In this method the state is defined as the defender location only i.e. ignoring the attacker. All the algorithms use standard settings of learning rate $\lambda = 0.05$, discount factor $\gamma = 0.99$ and fading exploration rate $\epsilon = 0.01 + \frac{0.99}{e^{0.001t}}$. We experiment with different number of periodical observability steps. We use two different gridworlds; Gridworld 1, which has three targets and no obstacles and Gridworld 2, which has two targets and some obstacles. See Figure 2a showing Gridworld 1, the green and red hollow rectangles show the players starting nodes - defender and attacker respectively. The red crosses represent the targets. The heatmap shows defender visits in every node and the black dots show attacker visits (the bigger the dot the more often the attacker was in that node). The gridworlds are shown for one of the baselines - Q-learning with full observability.

In Figure 3 we show the performance of BayesQMDP against the baseline algorithms in GridWorld 1 (Figure 2a) with 95% confidence intervals. The black solid curve is for the case where the defender gets to observe the attacker location every 3rd time step and the red dashed curve is observing every 4th time step. The full observability Q-learning (the green dotted curve) performs the best which is expected, however the no observability Q-learning (the dash-dot blue

Grid	*Q-L full obs*	*Q-L no obs*	*BayesQMDP*
1	$58.0\%, \pm 6.82\%$	$0\%, \pm 0\%$	$26.7\%, \pm 6.12\%$
2	$45.8\%, \pm 6.89\%$	$11.6\%, \pm 4.43\%$	$32.2\%, \pm 6.47\%$

Table 1: Average wins over last 100 episodes with 95% confidence intervals

curve) gets exploited by the attacker's fictitious play. The BayesQMDP algorithm gives us a good performance in the partial observability. Note that observing the attacker every 4th time step can lead to information set size of 25 states in the worst case (4 actions in every state - without repeating the same states).

In Figure 4 there is BayesQMDP compared to the two baselines for Gridworld 2 (Figure 2b). In this experiment we do not assume fixed number of steps to observe the attacker location, instead we sample uniform at random between observing the attacker every 3rd and every 4th time steps to account for any potential synchronisation. One can observe that BayesQMDP gives superior performance compared to no observability case and is close to full observability case. This result shows the effective behaviour of BayesQMDP in partial observability.

Every experiment is run 200 times with 5000 episodes each and averaged over to get significant results. In Table 1 we show the defender wins in the last 100 episodes for all the compared algorithms, we also state 95% confidence intervals. Note that for BayesQMDP we state the results for observability every 3rd time step for Gridworld 1 and random observability between 3rd and 4th step for Gridworld 2.

Conclusion

We have proposed a new algorithm combining QMDP and Bayesian inference called BayesQMDP, which can effectively use partial information about attacker location. We compared this algorithm with two very simple baseline algorithms to demonstrate the initial performance and promising behaviour. The algorithm is experimentally shown to converge against our version of fictitious play. This is a preliminary experimental evaluation of BayesQMDP and we leave further analysis of the proposed algorithm for future work. The next step is comparing BayesQMDP to stronger baseline algorithms such as BA-POMCP (Katt, Oliehoek, and Amato 2017) or DRQN (Hausknecht and Stone 2015).

References

An, B.; Kempe, D.; Kiekintveld, C.; Shieh, E.; Singh, S.; Tambe, M.; and Vorobeychik, Y. 2012. Security Games with Limited Surveillance. In *AAAI Conference on Artificial Intelligence*, 1241–1248.

Bloembergen, D.; Tuyls, K.; Hennes, D.; and Kaisers, M. 2015. Evolutionary Dynamics of Multi-agent Learning: A Survey. *Journal of Artificial Intelligence Research* 53:659–697.

Bosansky, B.; Lisy, V.; Lanctot, M.; Cermak, J.; and Winands, M. H. M. 2016. Algorithms for Computing Strategies in Two-player Simultaneous Move Games. *Artificial Intelligence* 237:1–40.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian Q-learning. *American Association of Artificial Intelligence (AAAI)* 761–768.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. *International Joint Conference on Artificial Intelligence* 2589–2595.

Fudenberg, D., and Levine, D. 1996. *The Theory of Learning in Games*. The MIT Press.

Hausknecht, M., and Stone, P. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. *arXiv preprint arXiv:1507.06527*.

Heinrich, J.; Lanctot, M.; and Silver, D. 2015. Fictitious Self-Play in Extensive-Form Games. In *International Conference on Machine Learning*.

Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; and Munoz de Cote, E. 2017. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity. *arXiv preprint arXiv:1707.09183*.

Jain, M.; Korzhyk, D.; Vaek, O.; Conitzer, V.; Pechouček, M.; and Tambe, M. 2011. A Double Oracle Algorithm for Zero-Sum Security Games on Graphs. In *Autonomous Agents and Multiagent Systems*, 327–334.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* 101:99–134.

Katt, S.; Oliehoek, F. A.; and Amato, C. 2017. Learning in POMDPs with Monte Carlo Tree Search. In *International Conference on Machine Learning*.

Klima, R.; Lisy, V.; and Kiekintveld, C. 2015. Combining Online Learning and Equilibrium Computation in Security Games. *International Conference on Decision and Game Theory for Security* 130–149.

Klima, R.; Tuyls, K.; and Oliehoek, F. 2016. Markov Security Games: Learning in Spatial Security Problems. *NIPS Workshop on Learning, Inference and Control of Multi-Agent Systems* 1–8.

Korzhyk, D.; Yin, Z.; Kiekintveld, C.; Conitzer, V.; and Tambe, M. 2011. Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. *Journal of Artificial Intelligence Research* 41:297–327.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning Policies for Partially Observable Environments: Scaling Up. In *International Conference on Machine Learning*, 1–59.

Montesh, M. 2013. Rhino Poaching: A New Form of Organised Crime. Technical report, College of Law Research and Innovation Committee of the University of South Africa.

Pita, J.; Jain, M.; Marecki, J.; Ordonez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR Protection: The Application of a Game Theoretic Model for Security at the Los Angeles Interna-

tional Airport. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, 1805–1812.

Pita, J.; Jain, M.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2010. Robust Solutions to Stackelberg Games: Addressing Bounded Rationality and Limited Observations in Human Cognition. *Artificial Intelligence* 174(15):1142–1171.

Robinson, J. 1951. An Iterative Method of Solving a Game. *The Annals of Mathematics* 54(2):296–301.

Ross, S.; Chaib-draa, B.; Pineau, J.; Chaib-draa, B.; and Pineau, J. 2007. Bayes-adaptive POMDPs. *Advances in Neural Information Processing Systems* 1225–1232.

Shapley, L. S. 1953. Stochastic Games. *Proceedings of the National Academy of Sciences of the United States of America* 39(10):1095–100.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States. *International Conference on Autonomous Agents and Multiagent Systems* 1:13–20.

Tuyls, K., and Weiss, G. 2012. Multiagent Learning: Basics, Challenges, and Prospects. *AI Magazine* 33(3):41–52.

Wiering, M., and van Otterlo, M. 2013. *Reinforcement Learning: State-of-the-Art*. Springer.