Online Learning Methods for Border Patrol Resource Allocation

Richard Klíma[1,2] *, Christopher Kiekintveld[2], and Viliam Lisý[1]

[1] Department of Computer Science, FEE, Czech Technical University in Prague
klimaric@fel.cvut.cz, lisy@agents.fel.cvut.cz,
[2] Computer Science Department, University of Texas at El Paso
cdkiekintveld@utep.edu

Abstract. We introduce a model for border security resource allocation with repeated interactions between attackers and defenders. The defender must learn the optimal resource allocation strategy based on historical apprehension data, balancing exploration and exploitation in the policy. We experiment with several solution methods for this online learning problem including UCB, sliding-window UCB, and EXP3. We test the learning methods against several different classes of attackers including attacker with randomly varying strategies and attackers who react adversarially to the defender's strategy. We present experimental data to identify the optimal parameter settings for these algorithms and compare the algorithms against the different types of attackers.

Keywords: security, online learning, multi-armed bandit problem, border patrol, resource allocation, UCB, EXP3

1 Introduction

Border security is a major aspect of national security for many countries; in the United States alone billions of dollars are spent annually on securing the borders. However, the scale of the problem is massive, with thousands of miles of land and sea borders and thousands of airports to secure. Allocating limited resources to maximize effectiveness is a serious issue for the United States Customs and Border Protection agency (CBP). Indeed, the most recent strategic plan for the CBP places a great emphasis on mobilizing resources and using risk-based models to allocate limited resources. [1].

Game theory is an increasingly important paradigm for strategically allocating resources in security domains, and we argue that it can also be useful for border security. There are now many examples in which security games [6, 11] have been used as a framework for randomizing security deployments and schedules in homeland security and infrastructure protection. This model has

* Richard Klíma is affiliated with both CTU and UTEP; this research was conducted primarily while he was an exchange student at UTEP.

been successfully used to randomize the deployment of security resources in airports [7, 8], to create randomized schedules for the Federal Air Marshals [6, 12], and to randomized patrolling strategies for the United States Coast Guard [10].

Existing models of security games rely on constructing a game-theoretic model of the security problem including the actions and payoffs of both the attacker and the defender. These models are based on whatever data is available combined with expert analysis and risk assessment to model the attacker preferences. One of the reasons for this style of modeling is that there is relatively little direct evidence about the attackers; we cannot directly elicit their preferences, and attack events are so rare that there is not enough data available to directly construct a model. This lack of data leads to a time intensive, expert-driven modeling process that still faces challenges in trying to validate the models and keep them up to date.

In border security the situation is quite different from many of the areas where security games have been applied. The CBP makes hundreds of thousands of apprehensions annually for illegal entry, smuggling, and other violations. This means that there is a large amount of data available for building and updating game models of the interactions between border patrol and the illegal entrants. The nature of the interaction is also different. A terrorist attack is a one-time, very high stake events. However, border security is more accurately characterized as a large number of repeated interactions with lower stakes. Similar situations with frequent incidents occurs also in cyber security domains, so we expect that our approachs is also applicable to these domains.

We propose to model border security using adversarial learning models that are related to both game theory and machine learning. These models are more dynamic, and account for the possibility of learning about the opponent during repeated interactions of a game. We introduce a basic model for a border security resource allocation task that is closely related to multi-armed bandit problems studied in the online learning literature. We then apply several different online learning algorithms to this model, including algorithms designed for adversarial bandit problems. We present an empirical evaluation of the performance of these algorithms and analyze the results to show the feasibility of modeling resource allocation for border patrol using this approach.

2 Model

We study a simplified model of the problem of resource allocation for border patrol. One of the main challenges that we try to capture in this model is the problem of *situational awareness*, which can also be thought of as a problem of balancing exploration and exploitation. There are limited resources available for patrolling different regions of the border. Ideally, these resources should be used in regions where there is a high level of illegal traffic. However, traffic patterns can change over time as the attackers (e.g., illegal entrants and criminal smuggling organizations) adapt to the border protection strategy. This means that it is

necessary to maintain situational awareness even in areas that currently have low traffic so that any changes in the traffic patterns can be quickly detected.

We consider a model where a border region is divided into z distinct zones. The border patrol has only one resource available to patrol these zones, and must decide which zone to patrol [3]. The attackers try to cross the border without being detected. They must pick one of the z zones to attempt a crossing. The game is played in a series of n rounds representing discrete time periods (e.g., one day, or one hour). There are t attackers who attempt to cross during each round. Any attackers who chooses the same zone as the defender are apprehended, while attackers that chose different zones cross successfully.

We represent the defender and attacker strategies in a round using probability distributions over the zones. The defender strategy for round i is given by a vector $D^i = \langle d_1^i \dots d_z^i \rangle$ where d_j^i represents the probability that the defender patrols zone j in period i. Similarly, the attacker strategy round i is given by a vector $A^i = \langle a_1^i \dots a_z^i \rangle$ where a_j^i represents the probability that an attacker chooses zone j in period i. We assume that each of the t attackers chooses a zone independently according to the distribution A_i. This assumption is made by the idea that the attackers share common knowledge about the border; where it is more suitable to cross or there is high probability of being caught.

The goal for the defender is to maximize apprehensions. We assume that all zones are identical for the defender. The attacker has a penalty p for being caught as well as a base value that differs across the zones, denoted by c_j. For any zone j we calculate the attacker's expected value in round i as:

$$v_j^i = c_j - (d_j^i * p) \tag{1}$$

where d_j^i is the probability that the attacker will be caught in a given zone in this round, which comes from the defender's strategy. The values for the different zones can be interpreted as the value of successfully crossing in a given zone, less the costs associated with the crossing (e.g., payments to smuggling organizations, and the difficulty and time required to traverse the terrain). The asymmetry introduced by these values is also important, because if all zones are identical for both players there is a trivial equilibrium solution in which both players play the uniform random strategy (analogous to the symmetric game of Rock, Paper, Scissors).

3 Attacker Models

We introduce four different models of attacker behavior that represent a spectrum of levels of adaptation and intelligence. These are also designed to present different challenges for the online learning algorithms.

Random Fixed: This policy is a fixed attacker probability distribution over the zones. The strategy is generated randomly at the beginning of the scenario by

[3] We limit this to one resource to simplify the initial model, but plan to generalize to multiple resources in future work

drawing real random numbers from interval $(0, 1)$ for each zone and normalizing. This is intended as a baseline that should be relatively easy for the online learning methods to learn.

Random Varying: In this models we generate a new random attacker strategy after a fixed number of rounds; the new strategy is unrelated to the previous one. This models an attacker that changes strategies, but not intelligently in response to the defender. We chose to generate large changes intermittently rather than making constant small changes because it allows us to average results over many runs and evaluate how quickly the learning methods are able to detect and respond to sudden changes in attacker behavior.

Adversarial Fixed: This model assumes that the attacker is intelligently adapting in response to the defender's strategy. We assume that the attacker knows the number of times the defender has visited each zone in the past.[4] The attacker adapts his strategy gradually to maximize the value given in Equation 1. Here, the attacker uses the observed frequencies of the defender patrols to estimate the probability of being caught in each zone. This is motivated by *fictitious play*, a well-known learning dynamic in which players play a best response to the history of actions played by the other players [4]. However, we parameterize this learning strategy so that we can control the rate at which the attacker moves towards a best response using the learning rate parameter α. The initial attack strategy is selected randomly, and it is updated on each iteration according to:

$$A^{i+1} = (1 - \alpha) * A^i + \alpha * M \tag{2}$$

where M represents a vector that has a 1 for the zone that gives the maximum value, and 0s for all other zones.

Adversarial Varying: This model is identical to the previous one, except that we randomly change the base values c_j for each zone after a fixed number of rounds, similar to the random varying policy. This model simulates an attacker that adapts intelligently, but also has preferences that can change over time.

4 Defender Strategies

Our model captures one of the central difficulties in accurately estimating traffic, which is the limited observations that the defender makes about the attacker's strategy. The defender only observes the level of traffic in the zone that is patrolled in each time period, and not in the other zones, just like a patrol in the real world. This means that a defender strategy that always tries to patrol the zones with the highest levels of activity to maximize apprehensions risks developing "blind spots" as the attacker strategy changes. What were previously low traffic zones may have increased traffic due to adaptations by the attacker, but the

[4] This assumes a very knowledgeable attacker, but is fairly realistic since major transnational smuggling organizations use sophisticated surveillance to track border patrol presence.

defender cannot observe this unless it allocates some resources to exploration–patrolling zones that are believed to have low traffic to detect possible changes in the traffic levels over time.

This becomes a problem of balancing exploration and exploitation when allocating the patrolling resources [9]. The online learning literature contains many examples of models that focus on this basic problem, including the well-known multi-armed bandit problem [2]. In a multi-armed bandit, a player must select from a set of possible arms to pull on each iteration. Each arm has a different sequence of possible rewards that is initially unknown to the player. The player selects arms with the goal of maximizing the cumulative reward received, and must balance between selecting arms that have a high estimated value based on the history of observations, and selecting arms to gain more information about the true expected value of the arm.

From the defender's perspective, our model very closely resembles the basic stochastic multi-armed bandit problem if we assume a Random Fixed attacker. The zones in our model map to arms in the bandit model, and the defender must select zones both to maximize apprehensions based on the current estimates of the attacker strategy, but also to explore other zones to improve the estimate of the strategy. Based on this mapping, we apply variations of some of the existing solution methods for multi-armed bandits to our border patrol scenario. For the other attacker models this is no longer a stochastic multi-armed bandit problem because the underlying distribution of rewards changes over time (in some cases based on an adversarial response). Therefore, we also consider solution algorithms that have been developed for other variations of the bandit problem that make different assumptions about how the underlying rewards can change. We now describe in more detail the specific algorithms we consider.

Uniform Random: A baseline in which the defender chooses a zone to patrol based on a uniform random distribution in every round.

Upper Confidence Bound (UCB): One of the standard policies used for multi-armed bandits is UCB [2]. This method follows a policy that selects the arm that maximizes the value of the following equation in each round:

$$x_j + \sqrt{\frac{2\ln(n)}{n_j}} \tag{3}$$

where x_j is the average reward obtained from arm j, n_j is the number of times arm j has been selected so far and n is the number of rounds completed.

Sliding-Window UCB: This algorithm is a variant of the standard UCB that is more suitable for non-stationary bandit problems [5]. This algorithm should do well in an abruptly changing environment which is suitable for our attacker models that can change strategies (or underlying preferences) quickly. The main difference from the standard UCB is that the algorithm uses a fixed window of data from the previous rounds to calculate the estimated average rewards. At time step t we get average of rewards not from the whole history but only the τ previous rounds. SW-UCB chooses a zone which maximize the

sum of exploitation and exploration part. The exploitation part of the UCB formula is a local average reward:

$$\bar{X}_t(\tau, i) = \frac{1}{N_t(\tau, i)} \sum_{s=t-\tau+1}^{t} X_s(i) \mathbb{1}\{I_s = i\} \tag{4}$$

where N_i is the number of times arm i was played. $X_s(i)$ is a reward in time step s of ith zone and the indicator function returns a value of one if the chosen zone in the time step s is equal to ith zone, and zero otherwise.
The exploration part of the formula is defined by:

$$c_t = (\tau, i) = B\sqrt{\log(t \wedge \tau)/(N_t(\tau, i))} \tag{5}$$

where $(t \wedge \tau)$ denotes the minimum of two arguments and τ is a constant. B is a constant which should be tuned appropriately to the environment, which we address in out experiments.

EXP3: The Exponential-weight algorithm for Exploration and Exploitation (EXP3) [3] is designed for adversarial bandit problems in which an adversary can arbitrarily change the rewards returned by the arms. It is the most pessimistic algorithm due to the very weak assumptions about the structure of the rewards, but is still able to bound the total regret, similar to the guarantees provided by UCB for the standard model. It tends to result in greater rates of exploration than the UCB policies. The details of the algorithm are somewhat more complex, so due to space limitations we refer the reader to [3] for the full details.

5 Experiments

We now present the results of our initial empirical study of the performance of the different online learning strategies for the defender in the border patrol scenario. We test the algorithms against the four different attacker models that represent increasing levels of adaptation and intelligence. The performance of the learning strategies is evaluated based on the apprehension rate, which is the ratio between the number of apprehensions and total number of attackers that attempt to cross the border.

Unless otherwise specified, all of our experiments are conducted on a simulation with 8 zones. The simulation runs for 10000 rounds, and there are 10 attackers that attempt to cross each round according to the distribution specified by the attacker's strategy. Results are averaged over 50 runs of the simulation.

5.1 Parameter selection

We begin by testing different parameter settings for the learning methods to find the best settings to compare the performance of the different methods (random and UCB do not have parameters). We are also interested in the sensitivity of the algorithm's performance to the parameter settings in our domain. Many of the parameters balance the tradeoff between exploration and exploitation,

so we expect that different settings will perform well against relatively static opponents compared to more adaptive adversarial opponents. We choose the parameters which give the best result for adversarial attacker.

EXP3: We first present parameter tuning results for the parameter γ of EXP3 that controls the level of exploration. The parameter has values in the interval $(0, 1]$, and for higher values the algorithm becomes similar to playing randomly. Table 1 shows the apprehension rates for different values against the four different attacker models. For values of γ close to 1 we get behavior identical to a random defender. The best value of γ is 0.7 against adversarial attacker, so we use this value in the next experiments.

Table 1: EXP3 parameter tuning

γ value	random	random with changes	adversarial	adversarial with changes
0.1	20.05%	15.33%	14.65%	15.03%
0.3	19.47%	15.11%	15.18%	15.84%
0.5	16.77%	14.54%	16.06%	16.69%
0.7	15.13%	13.74%	16.58%	15.90%
1	12.53%	12.51%	12.50%	12.52%

Sliding-window UCB: There are no parameters in the basic version of UCB, but in sliding-window UCB there are several parameters specified in the original implementation [5]. We tune the parameter B which controls the exploration rate. The parameter τ controls the size of sliding window of history used in the calculations. For higher values of τ the algorithm of SW-UCB converges to standard UCB. We run several combinations of these parameters against two of the attacker models: the fixed random attacker and the adversarial attacker.

Table 2: SW-UCB τ tuning with different B parameter

	random fixed attacker					adversarial attacker				
τ value	0.5	1	5	10	20	0.5	1	5	10	20
50	21.11%	19.67%	14,25%	13.51%	12.92%	19.57%	18.21%	26.38%	26.52%	24.05%
100	21.74%	20.10%	14.93%	13.85%	13.15%	18.34%	19.03%	22.96%	24.26%	22.11%
500	21.64%	20.61%	17.08%	15.27%	13.82%	15.87%	16.39%	19.00%	21.91%	23.88%
1000	21.37%	23.08%	18.66%	15.87%	14.66%	15.09%	15.90%	18.72%	22.17%	26.53%
3000	21.58%	21.41%	19.75%	17.86%	15.08%	12.97%	13.45%	19.38%	23.02%	28.50%
5000	21.43%	21.51%	21.79%	18.81%	16.28%	12.68%	13.34%	19.84%	23.72%	28.98%

In Table 2 we present the performance for random fixed and adversarial attacker for different settings of the SW-UCB algorithm. As expected, against a fixed attacker the best results come from the longest sizes of sliding windows; if the environment is fixed, it does not make sense to throw out older data using

a sliding window since this data is still informative. Also we can observe that lower values of B parameter give higher apprehension rates The value of 1 for the B parameter results in the best performance here.

On the right side of Table 2 we have SW-UCB algorithm against an adversarial attacker. We can see that the results are almost opposite than in random fixed attacker case. We get better results with higher values of B parameter and with longer sliding-windows. The best results here are with a *high* value of the exploration rate of 20, compared to the opposite result for the fixed attacker.

In remaining experiments we set the γ parameter for EXP3 equal to 0.7. For sliding-window UCB we will use parameter B equal to 20 and parameter of sliding window τ equal to 5000. For adversarial attacker we will use a learning rate of 0.5 learning rate. For attacker strategies with changes we select a new random strategy or set of preference parameters every 2000 rounds.

5.2 Comparing Apprehension Performance

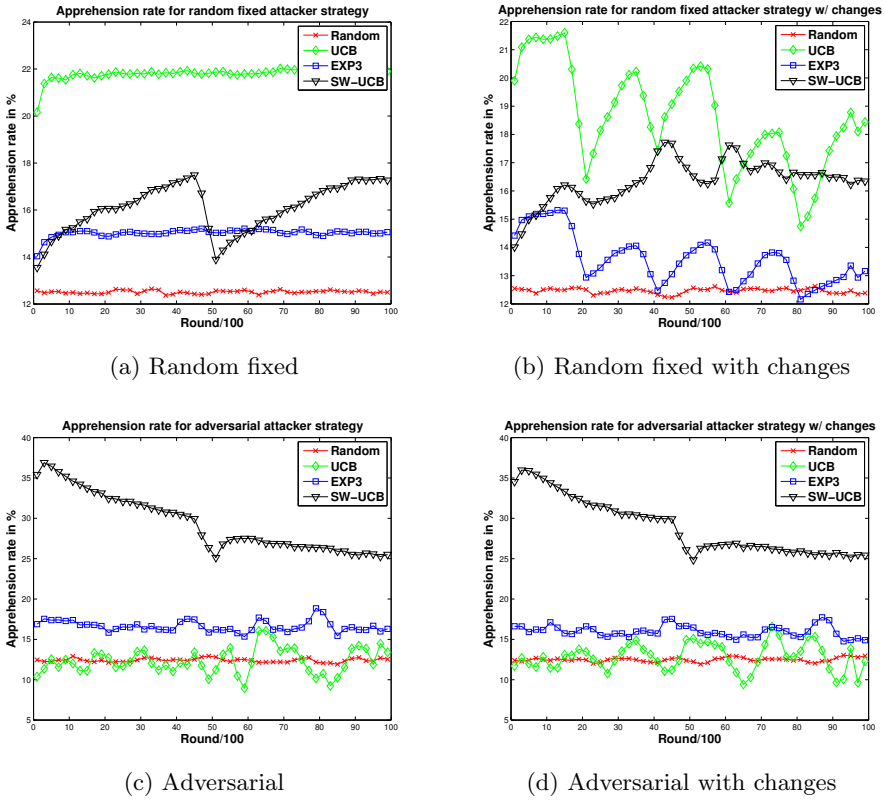
We now present initial results directly comparing the performance of the different learning methods against the different attacker strategies. The figures show how the apprehension rates of the algorithms evolve over the course of the 10000 round simulation. Results are averaged over 50 runs, and the plots are also smoothed using a moving average over 500 round buckets.

The results in Figure 1a show the learning process of the defender strategies against the random fixed attacker. The x-axis shows the number of simulation rounds divided by 100, and the y-axis shows the apprehension rate. All of the learning methods show the ability to learn against the fixed attacker. The standard version of UCB learns the fastest and has the best total apprehension rate, while EXP3 has somewhat poorer performance due to a higher exploration rate. For the SW-UCB there is a drop in the apprehension rate after the size of the sliding window equal to 5000.

Figure 1b shows the performance for the defender strategies against the fixed random attacker strategy with a change in the probability distribution every 2000 rounds. The points when the attacker strategy changes are clear in the plot, since the performance for all of the learning strategies drops off abruptly. However, the strategies are able to quickly respond and re-learn the adversaries strategy. We note that UCB has a high variance here, but SW-UCB shows more stable behavior, since it is designed for environments with abrupt changes.

In Figure 1c we present the behavior of the defender strategies against the adversarial attacker. In this case SW-UCB performs the best out of all defender strategies but tend to decrease over time. EXP3 gives the second best result and is quite stable. For all of the algorithms the performance later on is poorer, which is due to the intelligent adversary adapting to the defender strategy over time. All of the algorithms appear to be converging to an equilibrium strategy with the attacker over time.

Finally, Figure 1d shows the results against an adversarial attacker but with underlying zone preferences that change every 2000 rounds. The SW-UCB method again gives the best results, but the performance slightly decreases over



(a) Random fixed

(b) Random fixed with changes

(c) Adversarial

(d) Adversarial with changes

Fig. 1: Apprehension rates for different types of the attacker

time. We see here that the changes in the attacker preferences are not as dramatic as the direct changes in the attacker strategy, since they are muted and have an effect over time. The performance of the learning algorithms is somewhat degraded, but not dramatically worst than against the fixed adversarial attacker.

6 Conclusion

We have introduced a mathematical model for border patrol resource allocation that captures the important problem of allocating resources to maintain situational awareness via exploration. We have proposed several candidate solution methods drawn form the online learning literature that are suitable for making these decisions, including UCB, SW-UCB, and EXP3. They offer different levels of theoretical guarantees against changing and adversarial environments, with EXP3 providing bounds on performance in even the most adversarial settings.

Here, we have provided an initial empirical study comparing the performance of these algorithms in a simple border patrol scenario. We tested the parameters of the algorithms and determined the best settings, while also noting that the practical performance of the algorithms depends heavily on the parameter settings combined with how quickly the adversary changes. In comparison, SW-UCB often gives the best performance in the more adversarial cases, but all of the algorithms showed the ability to learn quickly and adapt even in the face of rapidly changing, adaptive adversaries. This demonstrates the potential for practical applications of these learning methods for resource allocation and situational awareness for border patrol.

Acknowledgements

This research was supported by the Office of Naval Research Global (grant no. N62909-13-1-N256) .

References

1. 2012–2016 border patrol strategic plan. U.S. Customs and Border Protection, 2012.
2. P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.
3. P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1), 2001.
4. D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. The MIT Press, 1998.
5. A. Garivier and E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. Technical report, 2008.
6. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, and M. Tambe. Computing optimal randomized resource allocations for massive security games. In *AAMAS-09*, 2009.
7. J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, and P. Parachuri. Depoloyed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. In *AAMAS-08 (Industry Track)*, 2008.
8. J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald. GUARDS - game theoretic security allocation on a national scale. In *AAMAS-11 (Industry Track)*, 2011.
9. J. Predd, H. Willis, C. Setodji, and C. Stelzner. Using pattern analysis and systematic randomness to allocate u.s. border security resources. 2012.
10. E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. Direnzo, G. Meyer, C. W. Baldwin, B. J. Maule, and G. R. Meyer. PROTECT : A Deployed Game Theoretic System to Protect the Ports of the United States. *AAMAS*, 2012.
11. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.
12. J. Tsai, S. Rathi, C. Kiekintveld, F. Ordóñez, and M. Tambe. IRIS - A tools for strategic security allocation in transportation networks. In *AAMAS-09 (Industry Track)*, 2009.