PIE: A Data-Driven Payoff Inference Engine for Strategic Security Applications

Haipeng Chen, Mohammad T. Hajiaghayi, Sarit Kraus, Anshul Sawant, Edoardo Serra, V.S. Subrahmanian and Yanhai Xiong

Abstract—Although most game theory models assume that payoff matrices are provided as input, getting payoff matrices in strategic games (e.g. corporate negotiations, counter-terrorism operations) has proven difficult. To tackle this challenge, we propose PIE (Payoff Inference Engine) that finds payoffs assuming that players in a game follow a myopic best response or a regret minimization heuristic. This assumption yields a set of constraints (possibly non-linear) on the payoffs with a multiplicity of solutions. PIE finds payoffs by considering solutions of these constraints and their variants via three heuristics: i) We approximately compute a centroid of the resulting polytope of the constraints, ii) We use a soft constraint approach that allows violation of constraints by penalizing violations in the objective function and iii) We develop a novel approach to payoff inference based on Support Vector Machines (SVM). Unlike past work on payoff inference, PIE has the following advantages: (i) PIE supports reasoning about multi-player games, not just one or two players, (ii) PIE can use short histories, not long ones which may not be available in many real-world situations, (iii) PIE does not require all players to be fully rational, (iv) PIE is one to two orders of magnitude more scalable than past work. We run experiments on (i) a synthetic data set where we generate payoff functions for the players and see how well our algorithms can learn them, (ii) a real-world coarse-grained counter-terrorism data set about a set of different terrorist groups, and (iii) a real-world fine-grained data set about a specific terrorist group. As the ground truth about payoffs for the terrorist groups cannot be tested directly, we test PIE by using the payoffs to make predictions about actions of the groups and corresponding governments (even though this is not the purpose of the paper). We show that compared with recent work on payoff inference, PIE has both higher accuracy and much shorter run time.

Index Terms—Game Theory, Payoff Inference, Counter-Terrorism

1 Introduction

Game theory is a classic and powerful tool for modelling strategic behavior of a system of multiple agents who interact with each other. Almost all work in game theory starts with a payoff matrix. In his pioneering study of conflict, Schelling [1] starts out with a payoff matrix for virtually every scenario. While getting a payoff matrix for a game is critical to modelling and understanding the behaviors of the involved agents, it unfortunately poses an enormous challenge in many real-world strategic games where such knowledge does not exist.

In this paper, we address the payoff inference problem in a game of interacting players with our counter-terrorism application. Because counter-terrorism is adversarial, we target non-cooperative scenarios in order to answer the following question: *Given a body of historical data about the interactions of a set of non-cooperative players, is there a way to learn a payoff matrix?* The approach in this paper is partly motivated by our ongoing counter-terrorism research involving the terrorist group Lashkar-e-Taiba (responsible for the 2008 Mumbai attacks) — given the history of interactions of the governments of Pakistan and India and Lashkar-e-Taiba (LeT), we wish to understand their payoff [2], [3]. To answer these questions, we need a realistic game theoretic model that is able to characterize real-world game scenarios (e.g. in counter-terrorism) while being robust to potential deviations from the game models. To this end, we consider the following properties in our model:

- *Best Response.* Given the history of past events (i.e., choices of actions), in each time period, players choose an action that is an approximate best response to the history (subject to the bounded rationality property as described next).
- *Bounded Rationality.* In real-world games such as our counter-terrorism situations, decision makers are not likely to be fully rational but boundedly rational, i.e. players take actions whose payoffs are within ϵ percent of the action with best response payoff. Therefore, we assume bounded rationality, not full rationality.
- *Time Discounting.* Intuitively, players are more likely to be influenced by "recent" history as opposed to events from a distant past. In order to model this, we developed a notion of time-discounted regret.
- *No Correlated Equilibria, Short Histories.* A correlated equilibrium is a status where no player wants to deviate from the recommended strategy from a public signal (assuming the others don't deviate). When long histories are available and some extra assumptions are made e.g. [4], game play can converge to a correlated equilibrium even without a signaling mechanism.

- *Haipeng Chen, V.S. Subrahmanian and Yanghai Xiong are with Dartmouth College.*
  E-mail: haipeng.chen,vs,yanhai.xiong@dartmouth.edu
- *Mohammad T. Hajiaghayi and Anshul Sawant are with University of Maryland.*
  E-mail: Hajiagha@cs.umd.edu, anshul.sawant@gmail.com
- *Sarit Kraus is with Bar-Ilan University.*
  E-mail: sarit@cs.biu.ac.il.
- *Edoardo Serra is with Boise State University.*
  E-mail: edoardoserra@boisestate.edu

Manuscript received xx xx, xxxx; revised xx xx, xxxx.

However, many real-world applications have short histories for which convergence cannot be assumed. As a result, we do not assume correlated equilibria [4] or the existence of a signaling mechanism [5].

There has been extensive prior work on applying game-theoretic models to counter-terrorism and more generally, security problems [3], [6], [7], [8], [9], [10], [11], [12], [13]. These works usually assume that the payoff matrices of the players are known a-priori, which is not the case in our counter-terrorism domain. The payoff inference problem has been studied in economics for various markets [14], [15], [16], [17]. However, their focus is on modeling a particular market and then to use the domain-specific model for model fitting and regression. Therefore, their methods cannot be applied to our problem. Though there are several existing works on inverse reinforcement learning (IRL) which study the payoff inference problem, we will show that (i) most of these works focus on single agent agent [18], [19], [20], (ii) some of these works focus on multiple agents in a cooperative setting [21], [22] and (iii) the works on multi-agent non-cooperative settings [23], [24], [25], [26] generally (including those in (i) and (ii)) target payoff inference problems defined on a Markov decision process (MDP), which is an over-complication of our problem. For payoff inference on games with multiple players, we show that past works lack at least one of these properties. One work that is closely related to our problem is [27]. However, we will show that the model formulated in [27] is computationally inefficient, as solving the model involves convex optimization. Moreover, another distinction of our model from existing works is that we take into consideration all the properties mentioned above (i.e. best response, time discounting, no correlated equilibria) – but existing works lack at least one of the properties in their game theory-based payoff inference model.

We provide two different formulations to model the payoff inference problem. The first formulation is built on top of the concept of regret in repeated decision making problems where we define a set of constraints whose variables represent the tabular representation of payoffs for each player under each joint action. Our constraints informally state that at each time point t in the past, each player i chose to perform the action for which he had the maximal expected time-discounted regret prior to time t. In the second model, we interpret these constraints as a myopic best response to the state of the world (i.e., a history of actions for all the players), with a (possibly) non-linear function form representation of the payoff function. The two models lead to a set of constraints defined on the payoff functions.

To solve the above problems, we propose three approaches: CBS and SCA are devised for the first model, while SVMM is designed for the second model.

1) **Centroid-Based Solution (CBS).** In CBS, the (approximate) centroid of the constraint polytope is picked as the solution.
2) **Soft Constraints Approach (SCA).** In SCA, we allow the rationality constraints to be violated but penalize such violations in the objective function.
3) **SVM-based Method (SVMM).** In SVMM, we propose a heuristic method to map the payoff inference problem onto a support vector machine [28] and build a separator that captures the payoff function we wish to learn.

We implemented CBS, SCA, SVMM, as well as the ICEL algorithm (Inverse Correlation Equilibrium Learning [27]) on both synthetic data and two real-world datasets. We compared all 4 algorithms w.r.t. solution quality and run-time. On synthetic data where we knew the ground truth (because we generated player behavior using known payoff functions), we showed that SVMM outperforms both CBS and SCA w.r.t. both solution quality and run-time. We also compared CBS,SCA, and SVMM on two real-world data sets: (i) the Minorities at Risk Organizational Behavior (MAROB) dataset [29] the contains data on terrorist group behaviors and related government actions and (ii) a much more fine-grained data set [2] about the behavior of the terrorist group Lashkar-e-Taiba (LeT).[1] Again, SVMM outperformed CBS and SCA. We then ran experiments comparing SVMM with ICEL. When we compare the ability of SVMM with that of ICEL to predict true behaviors from learned payoffs on the MAROB data, SVMM's ability to predict behavior from the learned payoffs was much better than that of ICEL (median Spearman Correlation Coefficient of 0.7 for SVMM, compared to just 0.114 for ICEL).

2 Related Work

A major driver for our work is counter-terrorism applications. The development of game-theoretic methods to analyze terrorist behavior and organization has been pioneered by [6], [7], and subsequently adopted by others [8], [9], [10], [11], [12], [13]. However, as described above, these approaches usually assume that the payoff matrices are known in advance by the decision makers, which might not be the case in many real-world problems such as the counter-terrorism applications motivating our work.

Economists have studied payoff inference problems for various markets [14], [15], [16], [17]. However, their focus is on modeling a particular market and then to use various model fitting and regression methods to learn the best parameters. For instance, [16] studies the effect of land use regulations on the mid-scale hotel market. In our problem, the decision making process is in an interactive environment [30], [31] with multiple agents (i.e., governments and terrorist groups) as opposed to the single agent scenario in these works. Therefore, these lines of research cannot be directly applied to our problem.

Inverse Reinforcement Learning [18] learns payoffs of a single-agent operating in a given (usually Markovian) environment. [18] addresses the problem of learning a reward function by observing behavior of MDPs. However, they and a series of subsequent works [19], [20] assume a single rational agent in a given environment. Some recent works have focused on *Multi-agent Inverse Reinforcement*

1. As no ground truth exists about payoffs for real-world players in the MAROB and LeT data sets, we learned player payoffs from a training data set and then validated them on a separate validation data set by making predictions based on learned payoffs. We emphasize the fact that this paper is not about prediction – but about learning payoffs in order to understand group behavior. The goal is to understand the payoff structure for different players for different strategies so diplomats and counter-terrorism agencies can shape policies towards the terrorist groups. We use predictions solely to validate learned payoffs.

Learning (MIRL). In contrast to the scenario we consider in which players are non-cooperative, [21] and [22] study the problem of learning player payoffs in the presence of a centralized coordinator. While several other works [23], [24], [25], [26] study the non-cooperative setting, a major difference between the problem addressed in these works is that they focus on non-cooperative games defined on a Markov decision process (MDP), while the problem targeted in this paper is a one-shot decision making problem (i.e., the decision at the current time step does not affect the decisions at future time steps). Therefore, the approaches to MIRL are distinct from our work. Perhaps the prior work that is closest to our problem setting is by Waugh et al. [27] who proposed an approach to predict player behavior when no payoff matrix is available. A convex optimization formulation finds a maximum entropy solution to find the predicted distribution over joint actions. Finally, payoffs are computed by using the dual of the above optimization convex problem. However, this approach suffers from the computational complexity brought by the convex program formulation.

The assumption of equilibrium is common to most work on MIRL and other payoff learning methods. However, decision theory strongly suggests that human players don't follow equilibrium strategies, even when the equilibrium is unique (which is also rare in real-world problems) [1]. However, decision theory does highlight the importance of recency [32] and regret in human decision making. Anticipated regret is considered an important determinant of choice-behavior [33], [34], [35]. These aspects form the basis for PIE's time discounted regret minimization and myopic best response with exponentially decaying state. Thus, PIE differs from existing work on payoff inference in that we assume myopic rationality and not global rationality (equilibrium). We also assume simple game play dynamics inspired by relevant work from decision theory. In addition, we develop a fast and practical data analytic approach compared to the more theoretical approach taken by most machine learning papers. We show this with experiments on two real-world datasets and show superior performance compared to Waugh et al [27] who only study a very small, toy example.

3 Payoff Inference Model

In this section, we first introduce the preliminaries of the game model, followed by two different formulations of the payoff inference models. The first model is based on the idea of "regret minimization", with a tabular representation of the payoff matrices. The second model, which is motivated by our counter-terrorism application on Lashkar-e-Taiba, represents the payoff matrices in a function approximation form with respect to a "time-weighted history".

Preliminaries

Let $[N] = \{1, \ldots, N\}$ be a set of players. We assume that each player i has an associated set A_i of actions that it can take. Let $\mathcal{A} = A_1 \times \cdots \times A_N$ denote the set of all possible *joint actions*. Given a joint action $a \in \mathcal{A}$, a_i is the action of player i and a_{-i} is the joint action of all other players. Let u_i be an unknown payoff function: $u_i : \mathcal{A} \to [0, 1]$. $u_i(a)$ is the payoff of joint action a for player i. Let $\mathcal{U} = \{u_1, \ldots, u_N\}$ be the set of all (as yet unknown) payoff functions where u_i is the payoff function for player i. Let $[T] = \{1, \ldots, T\}$ be a set of past time points.

Let $m = \sum_{i \in [n]} |A_i|$ be the total number of actions for all players in the game. We encode a joint action as an m-dimensional binary vector. Each action for a player i is indexed from $\sum_{j \in \{1 \ldots, i-1\}} |A_j| + 1$ to $\sum_{j \in \{1 \ldots i\}} |A_j|$ in a fixed but arbitrary order. In other words, the first $|A_1|$ entries in the vector describe the actions for the first player, the next $|A_2|$ entries describe the actions for the second player, and so forth. Let v be an encoding for $a \in \mathcal{A}$. If, in the joint action represented by a, player i plays action $a_i \in A_i$ at time t, then, and only then is $v[a_i] = 1$, otherwise $v[a_i] = 0$.

Example 1. Suppose we have two players $1, 2$ and suppose $A_1 = \{a, b, c\}$ and $A_2 = \{a, e\}$ are the actions they can perform. Then the dimensionality of a joint action is 5 and an example of the vector representation of a joint action is:

pl-1	pl-1	pl-2	pl-2	pl-2
a	b	c	a	e
1	0	0	0	1

The first row is the player's ID and the second row is the action name. Here, the 5-dimensional vector (1,0,0,0,1) tells us that in this joint action, player 1 performed action a and player 2 performed action e.

A *history* is a sequence $H^\tau =< a^1, \ldots, a^\tau >$ where a^t is the vector of joint actions taken at time $t \in [T]$. We represent the history of a game as a matrix, H, where $H[t, a] = 1$ iff player i plays action $a \in A_i$ at time t. Thus, H_t, the t^{th} row of the history matrix represents the joint action taken by all players at time t. Likewise, the i'th column of H tells us that actions taken by player i at each time point.

Example 2. Suppose we have two players $[N] = \{1, 2\}$; player 1 is a terror group and player 2 is the government. Assume that the players' actions are pe_g ("political engagement with the government") for player 1 and pe_tg ("political engagement with the terror group") for player 2. Each of these variables has 3 possible levels of intensity (low, medium, high). Therefore, player 1 has three actions, pe_g(l), pe_g(m), pe_g(h), corresponding to the three levels of intensity of this action, and similarly, player 2 has three actions pe_tg(l), pe_tg(m), pe_tg(h). Let indices of the actions pe_g(l), pe_g(m), pe_g(h) be 1, 2 and 3 respectively for player 1, and 4, 5 and 6 respectively for player 2. Suppose we have three years ($[T] = \{1, 2, 3\}$) of history below:

$[T]$	year	player 1	player 2
1	2010	pe_g(l)	pe_tg(m)
2	2011	pe_g(m)	pe_tg(l)
3	2012	pe_g(h)	pe_tg(m)

Then the history matrix H is given by:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Regret-Based Payoff Inference

In this section, we define the concept of time discounted regret. Classical regret is defined with respect to a class Φ of modification functions. Each modification function $f \in \Phi$ is a mapping $f : A \to A$. Intuitively, a modification function suggests an alternative choice $f(a)$ for an action a. Instead of taking action a, the player takes action $f(a)$.

As there are many ways in which a player could modify his choice, we consider a set Φ of modification functions. In the context of our running counter-terrorism example, the different modification functions might correspond to all feasible actions that could replace a given action a. The *regret* for a player i is defined as:

$$R_{i,\Phi}(t) \quad = \quad \max_{f \in \Phi} \sum_{\hat{t}=1}^{t-1} u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}).$$

Here, $u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}})$ is the difference in utility for player i had he elected to take action $f(a_i^{\hat{t}})$ instead of whatever action he took at time t in the past. The summation $\sum_{\hat{t}=1}^{t-1} u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}})$ reflects the total regret that player i had w.r.t. his past actions, had he chosen to use modification function f instead of whatever method he used to select his past actions. Had player i used the modification function $f \in \Phi$ that maximizes this summation, then he would have gotten the maximal possible benefit, and the fact that he (maybe) did not use it is what leads to this regret.

When determining what action to take, players in the real world are often more influenced by recent actions than by actions in the distant past. Our notion of *time-discounted regret* takes this into account by allowing a player to discount the past at a rate α s.t. $0 < \alpha \leqslant 1$. After each time point, the "importance" of a past event is reduced by a factor of α. The time-discounted regret is defined as follows:

$$TDR_{i,\Phi}(t) = \max_{f \in \Phi} \frac{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(f(a_i^{\hat{t}}), a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}))}{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}} \tag{1}$$

Because of the α parameter in the definition of TDR, for us, a history is a *timed-stamped* collection of past joint actions. This is very different from [27] which only uses the history to extract the distribution of joint actions and considers it to be a *collection (without timestamps)* of past joint actions. When $\alpha = 1$, the definitions of regret and time-discounted regret coincide.

Suppose Φ_c is the set of all functions from A to A that are *constant* functions, i.e. if f is in Φ_c, there must exist an action $a' \in A$ such that for all $a \in A$, $f(a) = a'$. The *time discounted external regret* w.r.t. Φ_c is then simply given by:

$$TDER_{i,\Phi_c}(t) = \max_{\hat{a} \in A} \frac{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}))}{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}}$$

In other words, $TDER_{i,\Phi_c}$ only considers constant functions when computing time-discounted regret. We define the time-discounted external regret w.r.t. action $\hat{a}$ as:

$$TDER_i(\hat{a}, t) = \frac{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a^{\hat{t}}))}{\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}} \tag{2}$$

Intuitively, $TDER_i(\hat{a}, t)$ is the regret for player i due to the fact that she/he did not use the strategy to always play the action $\hat{a}$ in the past. We assume that for a rational player, the greater the regret w.r.t an action $\hat{a}$ in the past, the more likely it is that the player will play the action $\hat{a}$ in the future.[2]

Example 3. Let us reconsider Example 2 with $\alpha = 0.9$. The time-discounted external regret for player 1 w.r.t. action h in the year 2013 ($t = 4$) is:

$$TDER_1(h, 4) = \frac{0.81(u_1(h,m) - u_1(l,m)) + 0.9(u_1(h,l) - u_1(m,l)) + (u_1(h,m) - u_1(h,m))}{0.81 + 0.9 + 1.0}$$

Observe that the weights 0.81, 0.9 and 1.0 are the weights for years 2010, 2011 and 2012, respectively.

A player is *rational* if, for each time t, the player chooses the action that caused the maximum time-discounted external regret in the past. Thus, our rationality constraints require that $\forall t \in [T]\backslash\{1\}$, $\forall i \in [N]$, $\forall \hat{a} \in A\backslash\{a_i^t\}$ the following condition holds[3]:

$$TDER_i(\hat{a}, t) \leqslant TDER_i(a_i^t, t) \tag{3}$$

or, equivalently,

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a_i^t, a_{-i}^{\hat{t}})) \leqslant 0 \tag{4}$$

Since our goal is to infer the payoff function of players (based on regret maximization) instead of computing an equilibrium, we do not have any constraint at $t = 1$, and the action at $t = 1$ is from the ground truth of the dataset. *Bounded Rationality.* As players in the real world are rarely 100% rational, we introduce a parameter $\epsilon \in [0, 1]$ that captures the degree of rationality. The closer ϵ is to 1, the more rational the player is, while the closer ϵ is to 0, the more irrational the player is. We replace Equation 3 (which assumes complete rationality) with the equation below, which allows weaker notions of rationality:

$$\epsilon \cdot TDER_i(\hat{a}, t) \leqslant TDER_i(a_i^t, t)$$

or equivalently

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}}(\epsilon \cdot u_i(\hat{a}, a_{-i}^{\hat{t}}) - u_i(a_i^t, a_{-i}^{\hat{t}})) \leqslant 0. \tag{5}$$

As ϵ and the α's are constants, this equation is linear. Each $u_i(-)$ term is a variable in this constraint. Let LC be the set of all linear constraints generated by Equation 5 above. We demonstrate them in the next example.

Example 4. By considering only the last two years of the history in Example 2, we observe that H^T is

[T]	year	player 1	player 2
1	2011	m	l
2	2012	h	m

If $\alpha = 0.9$ and $\epsilon = 0.8$, the rationality constraints are:

$$(0.8\, u_1(l,l) - u_1(h,l)) \leqslant 0 \qquad (0.8\, u_1(m,l) - u_1(h,l)) \leqslant 0$$
$$(0.8\, u_2(m,l) - u_2(m,m)) \leqslant 0 \qquad (0.8\, u_2(m,h) - u_2(m,m)) \leqslant 0$$

The result below states that LC is polynomial in size.

Proposition 1. The number of variables occurring in LC is polynomial in the number of players N, the number of actions M and in the size of the history T.

One problem with LC is that it may have multiple solutions, some of which may be trivial. An example of a trivial solution is when the utility function returns the same value for each joint action for each player. For instance,

2. In simple terms: if the player had great regret about not doing something in the past, especially the recent past, then he is more likely to do it in the future, especially in the near future.

3. Note that since we set a constraint for each player, this means that each player is maximizing its regret knowing that the other players are simultaneously maximizing their regret.

the maximal entropy solution of LC (the entropy function is applied to all variables of LC) assigns the same utility to all combinations of players and joint actions.

Proposition 2. Suppose $\mathcal{U} = \{u_1, \ldots, u_N\}$ is a maximal entropy solution for LC. Then for all joint actions a, a' and all players i, j, $u_i(a) = u_j(a')$.

The proofs of the above two propositions are in the appendix. Hence, given the assumptions in this paper, maximal entropy is not a very effective way of choosing a solution.

Function Form Payoff Matrix with Time-Weighted History

PIE was motivated by our ongoing counter-terrorism research. We have applied PIE to a multi-player game involving the terrorist group Lashkar-e-Taiba (responsible for the 2008 Mumbai attacks) and the governments of Pakistan and India as players. In such scenarios, a time-weighted history of players' actions is a representation of state of the world (which is a history of players' actions) at the time the player decides to take a new action. Time-weighted history captures the idea that recent actions may be more relevant than older ones.

The payoffs in Equation 5 are in tabular forms. Suppose we assume that a payoff function for player i at time t is *any* (linear or non-linear) function $\pi_i : \mathbb{R}^{m+1} \mapsto \mathbb{R}$ of the time-weighted history at time t. A *time-weighed history*, w_t, at time t is an m-dimensional vector defined as follows:

$$w_t = \frac{\sum_{i \in \{1 \ldots t\}} \alpha^{t-i} H_i}{\sum_{i \in \{1 \ldots t\}} \alpha^{t-i}}.$$

That is, $\pi_i(a, w_t)$ takes an action a and a time weighted history w_t as input and outputs a payoff value, specifying the payoff to player i of playing a at time t, w.r.t. w_t.

We assume that a player chooses an action at time t that has highest payoff with respect to the state of the world at time $t-1$. Let a be the action of player i at time t, thus

$$\pi_i(a, w_t) \geqslant \pi_i(a', w_t) \; \forall a' \in A_i, i \in [N], t \in [T] \qquad (6)$$

One such constraint needs to be written for each player i and each time t. *Note that these constraints may be non-linear.* It is easy to see that Equation 6 generalizes Equation 5. Note that this system of inequalities is feasible as assigning identical payoffs for all actions always satisfies Equation 6. However, this solution is trivial. Hence, it is important to choose a "robust" solution to the above system. For real-world problems, we require that the selected solution satisfies the following properties:

(P1) The family of functions to which our payoff functions π_i belong should not have arbitrary complexity, i.e., our hypothesis space should not allow arbitrary payoff functions to avoid overfitting. On the other hand, we should allow somewhat complicated non-linear payoff functions to avoid over-simplification.

(P2) While it is reasonable to assume that players react to game history and use actions which would generate high payoffs, we cannot assume that each and every player always adheres to this heuristic at all times. Therefore, our algorithm must admit the possibility that some points in the history may violate Equation 6. However, Equation 6 should hold for most of the game history.

(P3) Last but not the least, there must be a tractable algorithm to select a solution of these constraints so that PIE can apply to real-world strategic games involving many players and dozens of actions. While the current best approach [27] in the literature has been applied to games of upto 6 players with 3 actions for each player, they don't discuss the runtime of their approach. Our experiments will show that our best approach is 1 to 2 orders of magnitude faster.

4 Solutions

In this section, we present three approaches to select a solution of the system of constraints defined in the previous section. Note that the CBS and SCA approaches are designed for the LC constraints in Equation (5), while the SVMM is devised for the constraints in Equation (6).

Centroid Based Solution (CBS)

The Centroid Based Solution uses LC (Equation (5)). The classical way to choose one solution of LC is to choose the maximum entropy solution. However, as proved earlier in Proposition 2, this is not useful as the maximum entropy solution assigns the same utility to all combinations of players and joint actions. In order to avoid this, we choose a centroid based approach. The *centroid solution* of LC is the mean position of all points satisfying LC. Unfortunately, computing the centroid of a convex region is computationally very complex — even approximating it is $\#P$-hard [36]. We therefore approximate the centroid by using Hit-and-Run (HAR) sampling [37]. In HAR sampling, we start with a randomly selected solution of the constraints (point in the polytope). We then randomly identify a direction and distance and head in that direction for the selected distance from the last sampled point. If we are still within the polytope, this becomes our next sampled point. If the new point is outside the polytope, we regenerate a distance and direction till a valid point within the polytope is found. This process is iterated till the desired number of sample points is generated. HAR sampling allows us to sample points from a convex polytope uniformly at random in time polynomial in the number of dimensions (number of variables of LC). We approximate the centroid by taking the component-wise mean of the sampled payoffs.

Proposition 3. The centroid approximation described above is a solution of LC.

This proposition follows as the centroid approximation is a convex combination of solutions of LC.

Soft Constraints Approach (SCA)

In the Soft Constraints Approach, we again use only LC (Equation 5) and allow the rationality constraints to be violated by introducing a slack variable in each constraint in LC. These slack variables are denoted $s_{i,a,t}$ in the revised linear program RLP given below. We then find a solution of RLC that minimizes the sum of the slack variables which, in a sense, minimizes the amount of violation of the constraint. The *Revised Linear Program RLP* is shown below.

$$\underset{s,u}{\text{Minimize}} \sum_{i,a,t} s_{i,a,t}$$

Subject to:

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}} \left(\epsilon u_i(a, a_{-i}^{\hat{t}}) - u_i(a_i^t, a_{-i}^{\hat{t}}) + s_{i,a,t}\right) \leqslant 0, \forall i \in [N], a \in A, t \in [T] \tag{7}$$

The slack variables in Equation 7 are inside the parentheses to normalize for the history length and time decay of payoffs.

Payoff Inference using SVMM

In this section, we present a novel approach that uses Support Vector Machines (SVM) to find a "good" candidate solution to the system of inequalities given in Equation 6. Here, we use a set $CONS$ of constraints which are generated by Equation 6. Each constraint generated by Equation 6 has a left hand side and a right hand side. We encode the left and right hand sides of the inequalities in Equation 6 as points in a space. If a point encodes the right hand side of an inequality, it is assigned a label 1, otherwise, it is assigned a label 0. We then run the classification algorithm. The decision function for the learned classifier is the desired payoff function. We now describe this method in more detail.

Encoding points in game history. Let the number of actions of player i be $n_i = |A_i|$. Let a be an action of player i, whose index is equal to $\sum_{j \in \{1 \dots i-1\}} |A_j| + k$. Consider an $m+1$-tuple (a, h), where $a \in A_i$ and h is an m-dimensional point. Let $V : \mathbb{R}^{m+1} \mapsto \mathbb{R}^{(m+1)n_i}$ be a map that takes this $m+1$'th-tuple as input and outputs an $(m+1) * n_i$-dimensional vector. Map V is defined as follows:

$$V(a,h)[(k-1)*(m+1)+1] = 1$$
$$V(a,h)[(k-1)*(m+1)+1+j] = h[j], \quad \forall j \in \{1 \dots m\}$$

All other entries of $V(a,h)$ are 0. Suppose player i plays action a at time t. Then $V(a, w_{t,i}, w_{t-1,-i})$ is labeled 1. For all actions $a' \neq a$, $V(a', w_{t,i}, w_{t-1,-i})$ are labeled 0. The following example shows how this works.

Example 5. Consider a 3 player game with players $\{1,2,3\}$ with 2 actions (namely, Action 1 and Action 2) for each player. A point in the history of the game is represented by a 6-dimensional binary vector, e.g. the vector (1,0,0,1,1,0) represents the fact that players 1, 2 and 3 played actions 1, 2 and 1 respectively. Let the current state of the world be given by the vector $w_t = (w_1, w_2, w_3, w_4, w_5, w_6)$. Assume that at any time, player 1 plays a myopic best response to this state of the world. For simplicity, let payoffs be a linear function of the state of the world. The payoff for playing action 1 by player 1 is $p_1 = a_1 + \sum_{i \in \{1..6\}} a_{1i} w_i$. Similarly for action 2, $p_2 = a_2 + \sum_{i \in \{1..6\}} a_{2i} w_i$. Thus, the payoff function can be represented as a 14-dimensional vector $p = (p_1, p_2)$. Further, assume that player 1 actually chooses action 1 as the best response, then:

$$p_1 >= p_2 \tag{8}$$

We now encode the RHS as $V(1, w_t) = (1, w_1, w_2, w_3, w_4, w_5, w_6, 0, 0, 0, 0, 0, 0, 0)$ and the LHS as $V(2, w_t) = (0, 0, 0, 0, 0, 0, 0, 1, w_1, w_2, w_3, w_4, w_5, w_6)$. If we use SVM to learn a separating hyperplane W for points $V(1, w_t)$ and $V(2, w_t)$ such that $V(1, w_t)$ is on the positive side and $V(2, w_t)$ is on the negative side, then we have:

$$W^T V(1, w_t) > 0, \quad W^T V(2, w_t) < 0 \tag{9}$$

Thus, we have $W^T V(1, s) > W^T V(2, s)$ and W is a 14-dimension vector representing a feasible solution of Eq. 9.

Going back to the general case, let $\mathcal{E}$ be the function that takes a given game history as input and outputs a payoff value, the labeling, and encoding of points as defined above. We now describe the relationship between the SVM classifier applied to points given by mapping V and the system of inequalities given by Equation 6 with the help of the following two propositions - proofs are in the Appendix.

Proposition 4. The system of inequalities given in Equation 6 is feasible for a game history H, i.e., we can find payoff functions such that all the inequalities are satisfied if the SVM algorithm can find a separator for encoding $\mathcal{E}(H)$.

Proposition 5. If the SVM algorithm can find a separator that misclassifies n_1 points with label 1 and n_0 points with label 0, then we can find payoff functions such that at most $n_0 + n_1$ of the inequalities in Equation 6 are violated.

5 Implementation and Experiments

We implemented CBS, SCA, SVMM as well as the ICEL algorithm [27]. Section 5 uses synthetic data (with known payoff functions to evaluate these algorithms' accuracy). Section 6 uses the real-world MAROB data about 10 terrorist groups [29] with actions by both the terrorist groups and the government of the country involved. Section 6 also uses a very fine-grained counter-terrorism data set with three actors: the terror group Lashkar-e-Taiba [2] which carried out the Mumbai attacks and the governments of Pakistan and India. Section 7 compares our best algorithm with the ICEL algorithm. We used the MAROB data to compare ICEL and our SVMM method. Because of Proposition 2, we could not apply ICEL to the synthetic data — and because the Lashkar-e-Taiba data contained a host of environmental variables, we could not apply ICEL to that either. For the experiments on both the synthetic and real datasets, the discount factor α is set to 0.9.

Generation of Synthetic Data

We wrote R code to generate random games with random linear payoff functions and a random state of the world at each time. A payoff function is represented as a vector of coefficients of a linear function. Each of the payoff functions and the state of the world at each time point is a uniformly randomly directed positive vector of norm 1. After generating the payoffs and the state of the world at different times, an action history for all players is generated assuming best response. We are not simulating a game. Instead, each time point is a "what if" scenario, where each player is presented with a state of the world and they choose the best response as per their payoff functions. The code to generate random games varies the following inputs:

The state of the world is thus an $(na * np)$-dimensional vector. Payoff for each action is a linear function of the

np	Number of players
na	Number of actions for each player
n	Length of history for each game
noise	Probability an action is chosen uniformly at random
seed	The seed for random number generation

state of the world and hence is represented as an $(na * np)$-dimensional vector of coefficients. Thus, each player has na such vectors. [4] We introduce noise into our experiments by allowing each player, at each time step, to either play a random response independently at random with probability given by parameter "noise", or a best response to the current state of the world. To evaluate quality of payoffs learned, we learn the $np * na^2$ length vector of parameters of each player's payoff function. We measure the quality of our three payoff learning algorithms by comparing this vector with the actual payoff function vectors using Pearson Correlation Coefficients (PCCs for short).

The following subsections compare the three algorithms presented in this paper in order to identify which one is best - both from an accuracy and from a run-time perspective.

Performance of SVM based method

We use a linear soft margin SVM classifier using the R interface to libsvm [38]. The hyper-parameter for tuning this SVM is the cost of mis-classification C. We tried values of $C \in \{0.01, 0.1, 1, 10, 100\}$ and chose the best-forming SVM model. However, we also report the overall results (encompassing all five values of C). The choice of C turns out to not be critical to the performance of our algorithms. *SVMM performs very well with median PCC above* 0.8 *for games with* 5 *players and* 5 *actions for each player and median PCC between* 0.6 *and* 0.8 *for most of the smaller games.* In addition, performance degrades slowly with noise. We now analyze SVMM's performance in more detail.

Effect of Dimension of Payoff Function (SVMM): Figure 1 shows the effect of dimension of the payoff function on performance of SVMM.[5] Here, the number of sampled history points is 1000 and the noise parameter is set to 0. Surprisingly, the performance improves with dimension of the payoff function. This will be discussed in detail later in this section.

Effect of Noise (SVMM): Figure 2 shows the effect of noise on SVMM's performance. The number of samples is 1000 and the dimension of the payoff function is 9. We note that performance degrades gracefully under noise. For zero noise, the median PCC value is 0.73, whereas even with noise as high as 0.3 (i.e., with probability 0.3 a player chooses to play a random action instead of the best response), we still get a median PCC of 0.66.

Effect of Length of History (n) (SVMM): Figure 3 shows the effect of the length of the history on SVMM's performance. There is slight improvement in median PCC as n increases from 200 to 1000.

4. For most experiments on synthetic data, we have $na = np = 3$. Thus, each payoff function is a 9 dimensional vector. As there are 3 payoff functions per player (one per action), we are trying to learn a total of 9 vectors, each of which is 9-dimensional.

5. The figure was plotted using the standard "boxplot" function in R (http://www.r-bloggers.com/boxplots-and-beyond-part-i/). The boxes denote the range of 25th and the 75th percentiles. The line in the box is the median. The upper and lower lines outside the box is the "nominal" range of values and the circles are outliers.

Performance of SCA

In this section we describe the performance of the SCA method and show that it is inferior to SVMM.

Effect of Dimension of Payoff Function (SCA): Figure 4 shows the effect of dimension of the payoff function on SCA's performance. The number of sample history points is 1000 and the noise parameter is set to 0. Performance degrades with dimension of the payoff function.

Effect of Noise (SCA): Figure 5 shows the effect of noise on SCA's performance. The number of samples is 200 and dimension of the payoff function is 9. While performance does not degrade significantly with noise, overall performance is poor (Overall PCC median of 0.22).

Effect of History length (SCA): Figure 6 shows the effect of history length on SCA's performance. Here noise is 0 and dimension of the payoff function is 9. Somewhat counter-intuitively, performance degrades with history length. This could be because the number of slack variables increases linearly with the length of history. Thus, the degrees of freedom of the model is potentially higher with a longer history and thus a longer history can lead to overfitting.

Performance of Centroid Based Method

In this section we study CBS's performance and show that it is far inferior to SVMM.

Effect of ϵ (CBS): Figure 7 shows the effect of ϵ on CBS's performance. Here, the length of history is 200 and dimension of the payoff function is 9. The noise is 0. The overall performance of CBS is better than SCA but much worse than the SVM based method (PCC median of 0.45 for $\epsilon = 0.9$).

Effect of Dimension of Payoff Function (CBS): Figure 8 shows the effect of dimension of the payoff function on performance of CBS. Here, the number of sample history points is 1000 and the noise parameter is set to 0. Performance shows no discernible trend.

Effect of Noise: Figure 9 shows the effect of noise on performance of CBS. Here, the number of samples is 1000 and dimension of the payoff function is 9. Performance degrades sharply with noise and even with 10% noise, is close to random.

Effect of Length of History (CBS): Figure 10 shows the effect of history length on performance of CBS. Here noise is 0 and dimension of the payoff function is 9. Performance does not improve with n and shows no discernible trend.

Runtime Comparison of CBS, SCA, SVMM

Figure 11 shows the relative runtime performance of the three methods for varying lengths of histories. SVMM is faster than SCA by an order of magnitude and faster than CBS by 2-3 orders of magnitude. For this comparison, na and np are 3. The actual performance time of SVMM for varying values of n, na and np are given in Figure 12.

Discussion of the Results

Effect of Dimension of Payoff Function: The effect of the dimension of the payoff functions on the performance of the SVMM, SCA and CBS is depicted in Figures 1, 4 and 8 respectively. We see that SVMM's performance improves

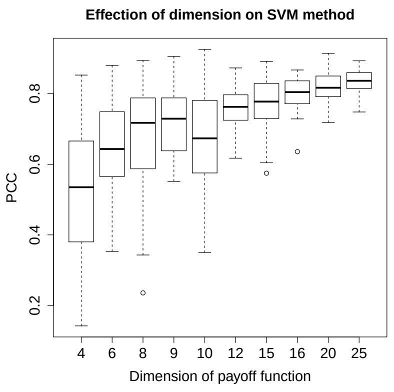

Fig. 1. Effect of payoff function dimension on performance of SVM method for synthetic data

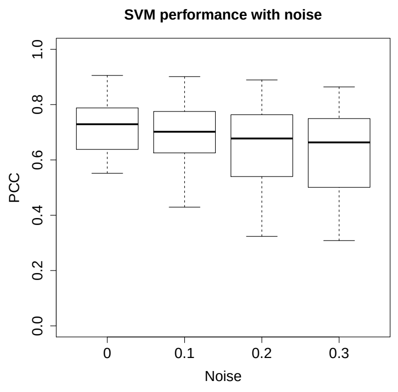

Fig. 2. Effect of noise on performance of SVM method for synthetic data

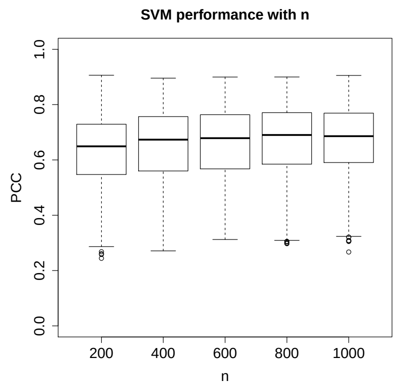

Fig. 3. Effect of history length on performance of SVM method for synthetic data

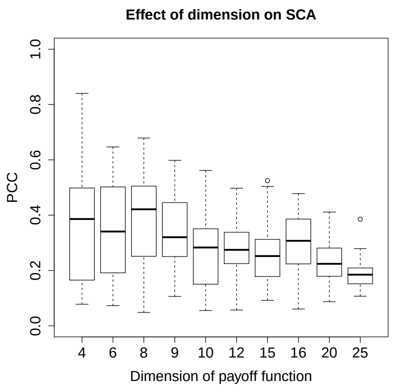

Fig. 4. Effect of dimension of payoff function on performance of SCA for synthetic data

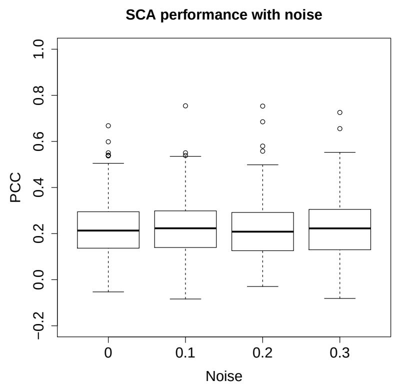

Fig. 5. Effect of noise on performance of SCA for synthetic data

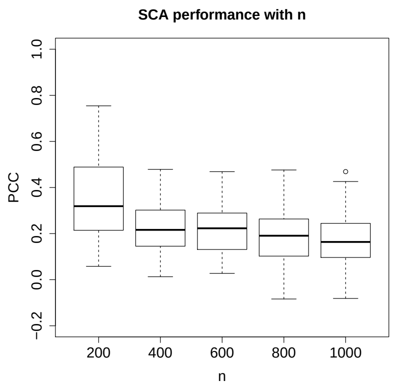

Fig. 6. Effect of history length on performance of SCA for synthetic data

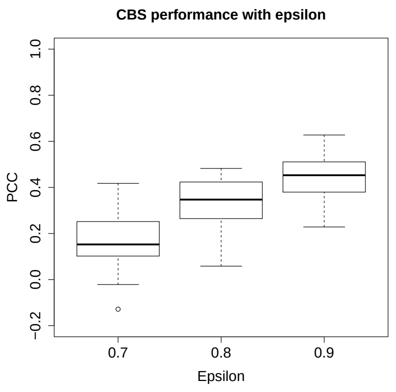

Fig. 7. Effect of epsilon on performance of CBS for synthetic data

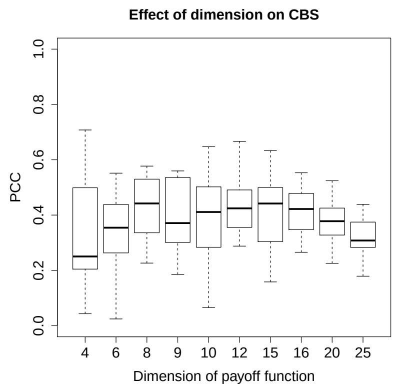

Fig. 8. Effect of dimension of payoff function on performance of CBS for synthetic data

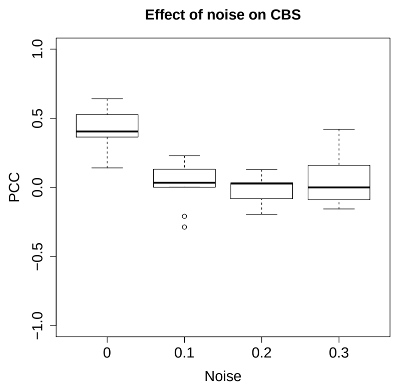

Fig. 9. Effect of noise on performance of CBS for synthetic data

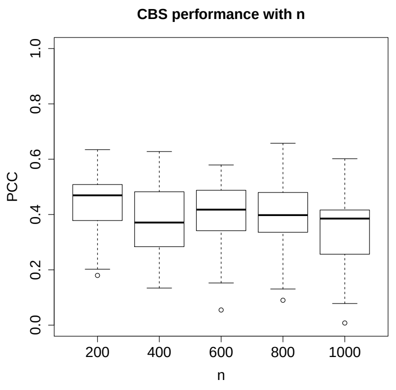

Fig. 10. Effect of history length on performance of CBS for synthetic data

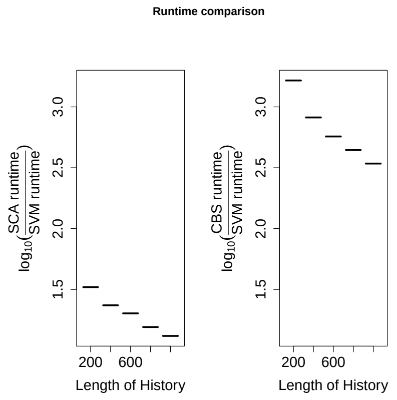

Fig. 11. Comparison of runtime of various methods for synthetic data

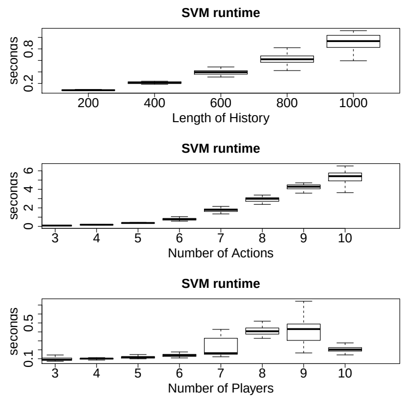

Fig. 12. Runtime of the SVM method

with dimensionality. This is counter-intuitive as performance of most classifiers degrades with dimension. However, for the payoff inference problem, the number of constraints and hence the number of points increases with the payoff function's dimension (for a fixed length of history). Thus, while the complexity of the classifier increases with data dimensionality, we have more data to learn from and hence, performance improves with dimensionality. In the case of SCA, we see that performance degrades with dimensionality. For SCA, the number of slack variables is the product of history length and dimension of the payoff function. Thus, the increase in dimension leads to an increase in number of the slack variables. We hypothesize that in SCA, this increase in number of slack variables with increase in dimensionality leads to performance degradation. CBS shows no discernible trend with increasing dimensionality. First, we are only approximating the centroid. Second, the centroid is very sensitive to individual constraints. Thus, CBS chooses an approximation to a feasible representative solution that is very sensitive to individual constraints. Therefore, it is not surprising that its performance is erratic.

Effect of Noise: The effect of noise on the performance of the SVMM, SCA and CBS algorithms is depicted in Figures 2, 5 and 9 respectively. SVMM's performance degrades gracefully with noise. As soft margin SVMs evolved from hard margin SVMs to handle mis-classification, this graceful degradation is expected. SCA's performance remains more or less constant and very poor with and without noise. SCA accommodates noisy points by allowing constraints to be violated by allowing negative slack variables and hence some robustness to noise is expected. However, a total lack of trend is a bit surprising. As noted earlier, centroid is very sensitive to individual constraints and hence extreme sensitivity to noise, as depicted in Figure 9 is expected.

Effect of History Length: The effect of history length on the performance of SVMM, SCA and CBS is depicted in Figures 3, 6 and 10 respectively. SVMM's performance improves slightly when n increases. Thus, SVMM learns a better classifier with more data. Surprisingly, for synthetic data, it seems that SVMM is able to learn a very good classifier even with $n = 200$. SCA again shows the trend of degrading performance with the increasing number of constraints. Performance of CBS is again erratic.

Runtime: The runtimes of the SVMM, SCA and CBS are compared in Figure 11. CBS is easily the worst. SVMM's runtime increases with length of the history (Figure 12 because the problem size varies linearly with the length of the history. The corresponding increase with the number of actions is faster as the problem size varies quadratically with the number of actions. SVMM's runtime increases with the number of players in general, however, when the number of players is 10 it suddenly drops. We don't have a good explanation for this behavior hope to find one in future work.

BOTTOM LINE: We conclude by stating that of the three algorithms presented in this paper, SVMM achieves significantly higher accuracy than SCA and CBS, is more robust to noise, and performs much better and faster than the other two methods. It gives excellent performance on relatively large games (Median Pearson Correlation Coefficient is above 0.8 *for games with* 5 *players,* 5 *actions per player).*

6 Experiments on Real-World Data Sets

MAROB Experiments

We ran tests on all 10 terror groups in the Minorities at Risk Organizational Behavior (MAROB) [29] data set for which at least 20 rows of data are available. We aggregated low level MAROB actions (both by the group and the government of the nation where the group is based) into high level actions. The high level actions involved two actions each for the group (political engagement with the government, militant activities) and for the government (political engagement with the group and suppression of the group). Each of these actions can be carried out at mild, medium, intense extents. Hence, each player can take one of 9 actions, leading to 81 total joint actions. We ran experiments with data about 10 group/nation pairs.[6] As the ground truth about payoffs for the terrorist groups cannot be tested directly, in order to test validity of the payoffs learned, we made predictions about actions carried out by terror groups and governments and checked the accuracy of these predictions. The mean number of actions for the government player, denoted by G is 4.1 and the mean for the terror organization, denoted TO is 5.6. The

6. We pair group and nations based on the simplified assumption that groups have very limited interactions. While there might be a few exceptions, this is true in most real-world cases.

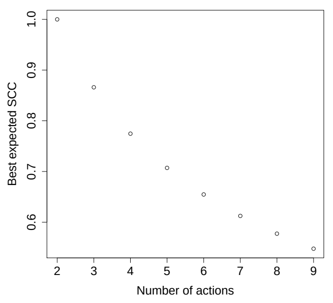

Fig. 13. Best expected SCC

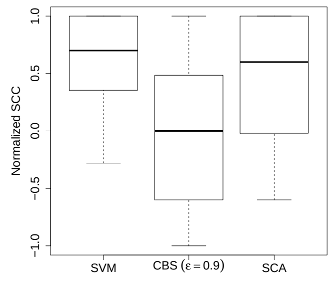

Fig. 14. Comparative performance of 3 methods on MAROB data set

mean number of joint actions (product of actions of G and TO) is 25.30. The history length of each game is between 20 and 25.

There are two issues related to this data that must be mentioned: (i) we only have 20 data points for each group/nation pair, (ii) we don't know the ground truth. Therefore, we evaluate the quality of the learned payoffs as follows. We compute the Spearman Rank Correlation Coefficient (SCC) between predicted payoffs and the binary vector representing actual actions performed during the time period. While the payoffs are reals in $[0, 1]$, we are correlating them with binary variables in $\{0, 1\}$, thus even in the best case, we cannot expect the correlation to be 1. For example, for 5 actions, a point in history may be $(1, 0, 0, 0, 0)$. It will be correctly predicted by a payoff vector such as $p = (1, p_2, p_3, p_4, p_5)$, $p_i < 1 \; \forall i \in \{2 \dots 5\}$. Assuming that $p_i \neq p_j \, \forall i, j \in \{1 \dots 5\}$, the expected SCC for this data (assuming p_i is uniformly distributed over $[0, 1)$) is 0.71 which is extremely good, considering the paucity of data. Figure 13 gives the best expected SCC as a function of the number of actions of a player. We normalized the SCC between predicted payoffs and the actual binary action vector to arrive at normalized SCC (NSCC) based on the number of actions of a player.

Comparison of performance of the three methods (Marob): Figure 14 compares CBS, SCA, and SVMM. On this dataset, SVMM uses the radial basis function as the kernel and the model is selected based on leave-one-out cross validation. Here again, SVMM performs well (median NSCC=0.7) and comfortably outscores SCA (median NSCC=0.6) and CBS (median NSCC=0). While for some part of the data set CBS does well with 75^{th} percentile NSCC value close to 0.4, the average CBS performs very poorly, with a median NSCC of close to 0 indicating near random performance. SCA performs well but not as well as SVMM.

As in the case of synthetic data (Figure 9) CBS is very sensitive to noise. SVMM and SCA perform well because they both allow some constraints to be violated. As shown in Figures 4 and 6, SCA's performance degrades with the number of constraints. However, since all the histories in the MAROB data set are short ($\sim$ 20), SCA performs quite well. SVMM performs even better than SCA. We hypothesize that this is because SVMM accommodates non-linear payoff functions using kernel methods and SCA allows only linear payoffs. This may be because real players' payoffs functions are not necessary linear.

LeT Experiments

We conducted extensive tests on the Lashkar-e-Taiba (LeT) dataset [2] which contains 252 rows (months) of data about 700+ variables.[7] We choose 24 variables which we considered relevant to our problem. These variables include six variables for various types of attacks carried out by LeT and eight for actions taken by the Pakistani government and military. The other 10 variables such as existence of an international ban, existence of conflict within LeT, split in LeT etc. are treated as *environmental variables*.[8]

The LeT dataset models a relatively big game. Players can take many actions simultaneously. If we encode each combination of actions as a separate action, we will end up with 64 actions for LeT and 256 actions for Pakistani government.[9] This leads to a very high dimensional encoding for this dataset. As an illustration, for a given history, H of this game, $\mathcal{E}(H)$ for Pakistani government would be $256 * (1 + 336) = 86272$ dimensions (ignoring the environment variables). We now describe how we tackle the dimensionality problem.

Independent Payoffs for Simultaneous Actions

One natural way to reduce dimensionality is to assume that payoffs for simultaneous actions are independent and additive. However, different actions may require different levels of effort for the player and it is reasonable to assume that payoffs are proportional to effort. For example, if attacking a security installation requires double the effort of attacking civilian transport, then the payoffs for the two actions are comparable only if the payoff from attacking the security installation is double the payoff from attacking civilian transport. This is because the capability and resources of an organization are limited and thus, to maximize the payoff, effort should be spent on actions that give maximum payoff for each unit of effort. Therefore, for this approach, we need to assign effort-based weights to players' actions. However, there is no reliable way of knowing how much effort was needed for each action of the player. *Therefore, we rejected this approach.*

Instead, we relax the constraints in Equation 6 by assuming that regret for each action actually played at time t is greater than or equal to regret for actions not played at time t. For example, assume that at time t, a player played actions $(0, 1, 0, 1, 0, 0)$ indicating that they took actions 2 and 4 out of possible actions in $\{1, \dots, 6\}$. We then assume that regrets for actions 2 and 4 were higher than regrets for other actions at time t. The other alternative could have been encoding each of the possible combinations of actions as a separate action, which leads to 64 possible distinct actions at each time step.

7. It includes details about attacks carried out by LeT, communications campaigns and rallies organized by LeT. It also includes actions by the state (Pakistan) and international actors (US, India, EU etc.) such as arrests, tribunals, killings related to members of LeT.

8. Environment variables can be seen to be actions of another player (similar to the *chance player* in classical game theory), whose actions we cannot predict (or are not interested in predicting). Nevertheless, these actions do have an effect on payoffs of other players.

9. The dataset does not attribute other actions to specific third part players such as the US or India.

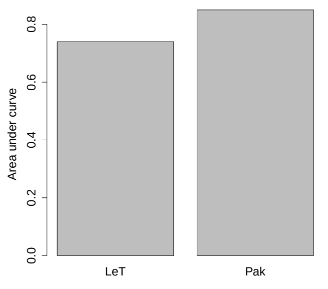

Fig. 15. Predictive performance of SVMM on LeT dataset

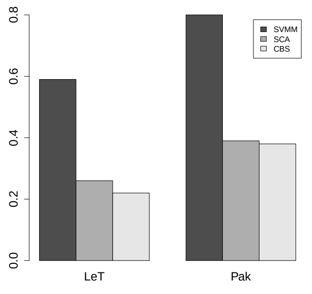

Fig. 16. Performance of SVMM, SCA and CBS on LeT dataset

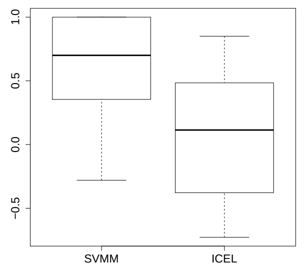

Fig. 17. Comparative performance of ICEL and SVMM

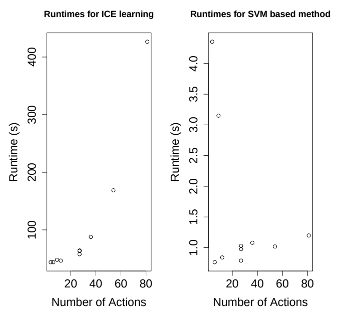

Fig. 18. Comparative runtime for ICEL and SVMM

Evaluation of Quality of Learned Payoffs

We evaluate the quality of prediction in two ways. First, we compute the SCC of events and payoffs for each time period from the test data. We compute SCC between predicted payoffs for actions and the binary vector representing actual occurrence of the events during the time period. We use this method to compare the performance of SVMM, CBS and SCA. Second, for SVMM, we compute the quality of predictions using Area Under the RoC curve as our metric. We don't use predictions to evaluate CBS and SCA methods because these methods don't extend naturally to prediction and prediction is not our primary objective.

Comparison of the Methods

Figure 16 presents comparative performance of the three methods on the LeT dataset. Again, SVMM performs well and clearly outperforms SCA and CBS. The NSCC for SVMM when the player considered is LeT in the dataset is 0.59. The NSCC for SVMM when the player considered is Pakistan is 0.80. The predictive performance of SVMM is good with area under single point RoC curve of 0.74 and 0.85 for LeT and Pakistan respectively (Figure 15).

The performance of SVMM is much better than the other two methods. We think that the reasons are three-fold. First, SVMM is robust to noise. Second, SVMM allows for non-linear payoff functions. Third, SVMM performs better with more constraints and higher dimensional payoff functions.

Policy Options based on Payoffs Learned about LeT

We used the payoffs learned for different actions to qualitatively estimate the values of the different actions that the players in the LeT case study can perform. In particular, the payoffs suggest the value or lack of value of certain actions by the two players (LeT and the Government of Pakistan), as well as actors who can reshape the environment surrounding LeT such as India or the US.

Actions by Pakistan. The payoffs we inferred using PIE have made some concrete discoveries. First, they suggest that arrests of LeT personnel by Pakistan are ineffective – a finding consistent with [2] where the authors found that arrests of LeT personnel can actually be followed by attacks on soft targets. Second, they suggest that release of operatives from prison is also not effective in curbing attacks by LeT. This is consistent with Rule (PST-4) in [2] which says that LeT attacks symbolic sites three months after months when the Pakistani government releases 0-9 LeT prisoners and LeT has locations across the border in India. It is also consistent with Rule (AA-4) in [2] which says that LeT attacks carries out attempted attacks one month after 0–2 LeT personnel are released.

PIE has additionally shown that a ban on LeT by the Pakistani government can have an effect on curbing attacks on transportation sites, public sites, and tourist sites. This is a new result, consistent with results in [2] which shows that having a ban is linked to LeT backed attacks on Hindus and that not having a ban is linked to attacks on professional security forces, security installations, and armed clashes.

Another interesting new finding by PIE is that a freeze of LeT assets by Pakistan are linked to a much higher payoff for targeting professional security forces, security installations and Hindus.

Finally, PIE's payoffs confirm conventional wisdom that military support by Pakistan leads to higher payoffs for targeting security forces, installations and public structures.

Actions by International Actors (primarily India and the US)

We know from [2] that arrests of LeT personnel are linked to a reduction in attacks on hard targets (such as security installations) but are linked to an increase in attacks on softer targets — PIE's inferred payoffs confirms this by showing that such arrests lead to increased attacks on civilians on the basis of ethnicity (Hindus) and tourist sites.

Likewise, the imposition of international bans can lead to more attacks on civilian targets and fewer attacks on security installations, suggesting that external pressure on LeT causes them to move from attacking hard targets to softer ones.

A new result derived by PIE is that asset freezes reduce the payoffs for LeT to carry out attacks on civilian targets.

7 Comparison with ICEL

We compare the performance of our best algorithm (SVMM) against the *Inverse Correlated Equilibrium Learning (ICEL)* algorithm of [27]. ICEL assumes that players play a correlated equilibrium. The input to ICEL is the game history and corresponding outcomes, which depend on joint actions of the players. The output is a joint distribution over the player's actions. The learned distribution is a Correlated Equilibrium for the corresponding inferred payoffs.

We evaluated the performance of ICEL against SVMM using the NSCC metric on the MAROB dataset. We could not evaluate ICEL on the LeT dataset because it has environmental variables in addition to player actions and ICEL is not applicable to games with environmental variables. If the payoffs learned by ICEL correspond to a Correlated Equilibrium actually played by the players, we can expect good rank correlation between chosen actions and payoffs. However, as can be seen from Figure 17, this is not the case. The median NSCC is 0.1140 over all games (which suggests that ICEL is only marginally better than random noise). NSCC is above 0.5 in only 3 out of 20 instances (10 games, 2 players per game). In comparison, SVMM exhibits far superior performance with median NSCC of 0.7.

We believe the poor performance of ICEL stems from two factors. First, we have no knowledge of the outcomes, only of game history. We empirically observed that in the absence of any information about the outcomes, the ICEL convex program simply converges to the distribution of actions actually played by the players (mean Kullback-Leibler divergence between actual and learned distributions is 0.043, max 0.088). Thus, the method corresponds to predicting that whatever happened in past will happen in future with no notion of recency and dynamics. Second, in our real-world data, the players may not play a correlated equilibrium and our proposed dynamics, which are based on regret minimization and recency, may be a closer approximation of reality.

The runtime comparison is shown in Figure 18. Here again, SVMM is 1-2 orders of magnitude faster than ICEL. However, this is a bit of an apples and oranges comparison as the SVMM code is in R with a Libsvm backend and the ICEL code uses the Python code provided by authors publicly at their website. We note that ability to use highly optimized, stable and mature libraries provided by machine learning community for classification tasks is one of the advantages of our approach over other extant approaches.

8 Conclusion

In this paper, we have developed, for the first-time, a method to infer payoffs for real-world games, under much more reasonable assumptions than past work. Specifically, unlike most past work, PIE is applicable to multi-player games, allows players to not be fully rational, does not assume a coordination mechanism, does not assume a symmetric game and is scalable, while other works are lacking in at least one of these aspects. Moreover, we apply our theory to real-world strategic games with two real world counter-terrorism datasets based on two widely influential previous studies [2], [39]. Such individuals seek *understanding* - and our goal in learning these payoffs were to facilitate explaining the payoffs to such senior decision makers - rather than prediction. Toward this end, we propose three heuristics that may be used to learn payoffs of players in *multi-player* real-world games including one that builds upon Support Vector Machines - a tested technique in data mining that has never been used before for learning payoffs. Though the goal of this paper is not prediction, we test our methods in three ways. We use a synthetic data set and a well-known terrorism data set [29] to see how well we can predict known payoff functions (synthetic data) and actions (MAROB data). Even though we have small amounts of data in both cases, they are bigger than those in previous studies, and our best algorithm (SVMM) achieves good correlations. Our third test looks at 10 years of data about the terror group Lashkar-e-Taiba (responsible for the 2008 Mumbai attacks). We show that SVMM is both faster and much more accurate than ICEL [27] one of the best prior algorithms in the literature. Much work remains to be done. Even though PIE is more scalable than past work, there is still a long way to go. Moreover, explaining learned payoff functions to real world decision makers also has many challenging aspects that deserve much more future study.

Acknowledgment

This project is supported by the Office of Naval Research under Grant N00014-16-1-2918, the Israeli Ministry of Defense, the Israeli Ministry of Science and Technology, and the Japan Science and Technology Agency (JST) Strategic International Collaborative Research Program.

References

[1] T. C. Schelling, *The strategy of conflict*. Oxford University Press, 1960.
[2] V. Subrahmanian, A. Mannes, A. Sliva, J. Shakarian, and J. P. Dickerson, *Computational Analysis of Terrorist Groups: Lashkar-e-Taiba: Lashkar-e-Taiba*. Springer Science & Business Media, 2012.
[3] J. P. Dickerson, A. Mannes, and V. Subrahmanian, "Dealing with lashkar-e-taiba: A multi-player game-theoretic perspective," in *2011 European Intelligence and Security Informatics Conference*. IEEE, 2011, pp. 354–359.
[4] D. P. Foster and R. V. Vohra, "Calibrated learning and correlated equilibrium," *Games and Economic Behavior*, vol. 21, no. 1, pp. 40–55, 1997.
[5] F. Xia, B. Jedari, L. T. Yang, J. Ma, and R. Huang, "A signaling game for uncertain data delivery in selfish mobile social networks," *IEEE Transactions on Computational Social Systems*, vol. 3, no. 2, pp. 100–112, 2016.
[6] H. E. Lapan and T. Sandler, "Terrorism and signalling," *European Journal of Political Economy*, vol. 9, no. 3, pp. 383–397, 1993.
[7] W. Enders and T. Sandler, "Terrorism: Theory and applications," *Handbook of defense economics*, vol. 1, pp. 213–249, 1995.
[8] M. Tambe, *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.
[9] R. Lindelauf, P. Borm, and H. Hamers, "The influence of secrecy on the communication structure of covert networks," *Social Networks*, vol. 31, no. 2, pp. 126–137, 2009.
[10] R. Lindelauf, H. Hamers, and B. Husslage, "Cooperative game theoretic centrality analysis of terrorist networks: The cases of jemaah islamiyah and al qaeda," *European Journal of Operational Research*, vol. 229, no. 1, pp. 230–238, 2013.
[11] T. P. Michalak, T. Rahwan, O. Skibski, and M. Wooldridge, "Defeating terrorist networks with game theory," *IEEE Intelligent Systems*, no. 1, pp. 53–61, 2015.
[12] Z. Wang, Y. Yin, and B. An, "Computing optimal monitoring strategy for detecting terrorist plots," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, 2016.
[13] J. P. Dickerson, A. Sawant, M. T. Hajiaghayi, and V. Subrahmanian, "Preve: a policy recommendation engine based on vector equilibria applied to reducing let's attacks," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013, pp. 1084–1091.
[14] A. Petrin, "Quantifying the benefits of new products: The case of the minivan," National Bureau of Economic Research, Tech. Rep., 2001.
[15] A. Nevo, "Measuring market power in the ready-to-eat cereal industry," *Econometrica*, vol. 69, no. 2, pp. 307–342, 2001.
[16] J. Suzuki, "Land use regulation as a barrier to entry: Evidence from the texas lodging industry*," *International Economic Review*, vol. 54, no. 2, pp. 495–523, 2013.
[17] M. Conlin and V. Kadiyali, "Entry-deterring capacity in the texas lodging industry," *Journal of Economics & Management Strategy*, vol. 15, no. 1, pp. 167–185, 2006.
[18] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning." in *ICML*, 2000, pp. 663–670.

[19] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *AAAI*, 2008, pp. 1433–1438.

[20] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," *arXiv preprint arXiv:1206.5264*, 2012.

[21] S. Natarajan, G. Kunapuli, K. Judah, P. Tadepalli, K. Kersting, and J. Shavlik, "Multi-agent inverse reinforcement learning," in *Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on*. IEEE, 2010, pp. 395–400.

[22] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," in *Advances in neural information processing systems*, 2016, pp. 3909–3917.

[23] T. S. Reddy, V. Gopikrishna, G. Zaruba, and M. Huber, "Inverse reinforcement learning for decentralized non-cooperative multiagent systems," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012, pp. 1930–1935.

[24] B. Wiedenbeck and M. P. Wellman, "Learning payoffs in large symmetric games," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1881–1882.

[25] X. Wang and D. Klabjan, "Competitive multi-agent inverse reinforcement learning with sub-optimal demonstrations," *arXiv preprint arXiv:1801.02124*, 2018.

[26] X. Lin, P. A. Beling, and R. Cogill, "Multiagent inverse reinforcement learning for two-person zero-sum games," *IEEE Transactions on Games*, vol. 10, no. 1, pp. 56–68, 2018.

[27] K. Waugh, D. Bagnell, and B. D. Ziebart, "Computational rationalization: The inverse equilibrium problem," in *ICML)*, 2011, pp. 1169–1176.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[29] A. P. Asal, Victor and J. Wilkenfeld, "Minorities at risk organizational behavior data and codebook version 9/2008 online," Sep. 2008. [Online]. Available: http://www.cidcm.umd.edu/mar/data.asp

[30] W. C. Stirling, "Conditional coordination games on cyclic social influence networks," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 2, pp. 250–267, 2019.

[31] M. Soltanolkottabi, D. Ben-Arieh, and C.-H. Wu, "Modeling behavioral response to vaccination using public goods game," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 2, pp. 268–276, 2019.

[32] I. Nevo and I. Erev, "On surprise, change, and the effect of recent outcomes," *Frontiers in psychology*, vol. 3, 2012.

[33] C. G. Chorus, "A new model of random regret minimization," *EJTIR*, vol. 2, no. 10, 2010.

[34] I. Simonson, "The influence of anticipating regret and responsibility on purchase decisions," *Journal of Consumer Research*, pp. 105–118, 1992.

[35] M. Zeelenberg and R. Pieters, "A theory of regret regulation 1.0," *Journal of Consumer psychology*, vol. 17, no. 1, pp. 3–18, 2007.

[36] L. A. Rademacher, "Approximating the centroid is hard," in *SOCG*, 2007, pp. 302–305.

[37] D. Bertsimas and S. Vempala, "Solving convex programs by random walks," *Journal of the ACM (JACM)*, vol. 51, no. 4, pp. 540–556, 2004.

[38] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[39] V. Subrahmanian, A. Mannes, A. Roul, and R. Raghavan, *Indian Mujahideen: Computational analysis and public policy*. Springer Science & Business Media, 2013.

APPENDIX

The detachable appendix is only intended for the convenience of reviewers and is not part of the overall submission.

Summary of Notations

Symbol	Brief synopsis
$[N]$	The set of players
$\mathcal{A}$	The set of all possible joint actions
A_i	The set of actions of Player i
u_i	The payoff matrix for Player i
$\mathcal{U}$	The set of payoff matrices for the game
$[T]$	The set of past time points
a	The vector representing joint actions of all players
(k, a_{-i})	The vector a with action of player i modified to action k
a^t	The vector of joint actions of all players at time t
H^t	The complete history sequence $< a^1, \ldots, a^t >$ till time t
w_t	The weighted time history till time t
α	Time discount factor for reward
Φ	Family of modification functions for computing regret
$TDR_{i,\Phi}(t)$	Time discounted regret for Player i at time t
$TDER_i(t)$	Time discounted external regret for Player i at time t
$\pi_i(k, w_t)$	Payoff function for Player i, (possibly non-linear).

Proof of Proposition 1

Proof 1. The number of constraints for each player at time $t \in [T]$ is $M - 1$. Each constraint has at most T variables. Thus, the total number of variables that can occur in LC is at most $(M-1)NT$.

Proof of Proposition 2

Proof 2. Let the entropy function be defined as follows

$$- \sum_{i \in [N], a \in \mathcal{A}} u_i(a)\, ln(u_i(a))$$

We obtain the maximum value of this function when $\forall i \in [N]$, $\forall a \in \mathcal{A}$ we can deduce that $u_i(a) = e^{-1}$. Since we know that when $\epsilon \in [0,1]$,

$$\sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}} \epsilon \cdot e^{-1} \leqslant \sum_{\hat{t}=1}^{t-1} \alpha^{t-1-\hat{t}} e^{-1}$$

our rationality constraints in Equation 5 are satisfied. For these reasons it follows that the theorem holds. □

Proof of Proposition 4

Proof 3. We note that for points on one side of decision surface, the value of decision surface is less than 0 and for the other side it is greater than 0. Therefore, if points encoded for the RHS are on the positive side and LHS on the negative side, Equations 6 are satisfied. Otherwise, we can flip sign of the decision function and achieve the same result.

Proof of Proposition 5

Proof 4. Without loss of generality, we assume that points encoding the RHS of Equation 6 are assigned positive labels and points encoding the LHS are assigned negative labels. A misclassified point LHS point can be assigned a decision value higher than the RHS point can lead to at most one violated constraint. Similarly, a misclassified RHS point can lead to at most one violated constraint. Thus in all we can have at most $n_0 + n_1$ violated constraints.

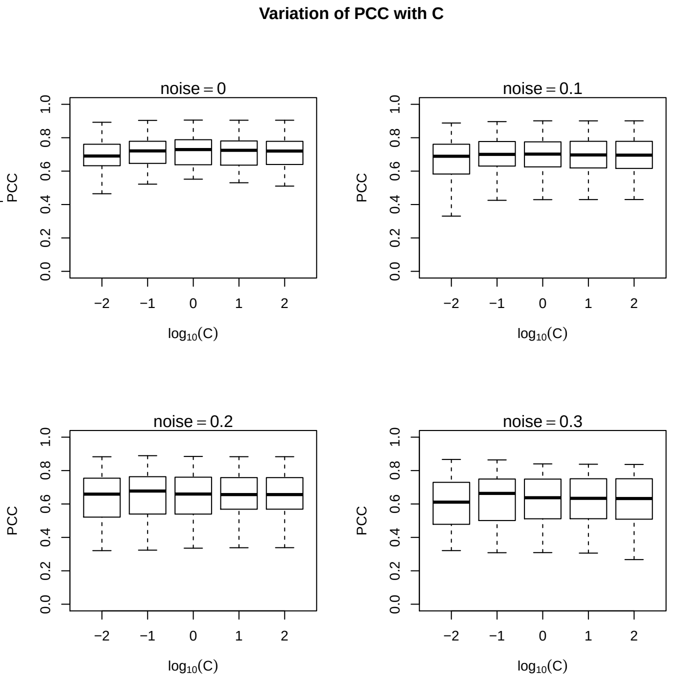

Fig. 19. Effect of cost on performance of SVM method for synthetic data

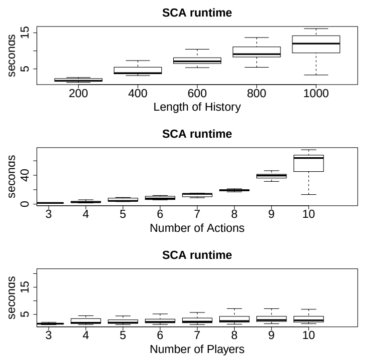

Fig. 20. Runtimes for SCA

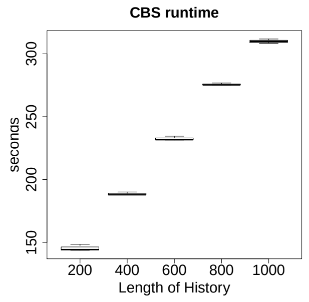

Fig. 21. Runtimes for CBS

Additional experimental results

Effect of Cost (SVM) Figure 19 shows the effect of the hyper parameter C on the performance of the SVMM. We present this figure in order to show that for the synthetic data, the performance is quite stable and not very dependent on the choice of the hyperparameter C. Hence, tuning the SVM is straightforward.

Actual runtime for CBS and SCA methods Figures 20 and 21 show the actual runtimes of CBS and SCA under different settings.