Learning to Identify Winning Coalitions in the PAC Model

Ariel D. Procaccia Jeffrey S. Rosenschein

School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel

Abstract

Agents situated in a real-world setting may frequently lack complete knowledge of their environment and of the capabilities of other agents. Researchers have addressed this problem by exploiting tools from learning theory. In particular, advances in reinforcement learning have yielded learning algorithms that converge to various solution concepts for stochastic games. However, scarcely any studies have attempted to tackle learning in coalition formation. Even fewer have applied the widely known *Probably Approximately Correct* (PAC) theory to multiagent systems.

In this paper, we endeavor to remedy these shortcomings by considering PAC learning of *simple* cooperative games, in which the coalitions are partitioned into “winning” and “losing” coalitions. We analyze the sample complexity of a suitable concept class by calculating its Vapnik-Chervonenkis (VC) dimension, and provide an algorithm that learns this class. Furthermore, we study constrained simple games; we demonstrate that the VC dimension can be dramatically reduced when there exists only a single minimum winning coalition (even more so when this coalition has cardinality 1), while other interesting constraints do not significantly lower the dimension.

1 Introduction

A significant portion of recent research in multiagent systems has focused on learning. In particular, researchers have constructed and analyzed algorithms that converge to equilibrium concepts (such as Nash equilibrium) for stochastic games [10, 7]. The latest studies have attempted to introduce algorithms that achieve an optimal payoff in the presence of other agents, but have not considered the convergence to equilibrium as obligatory [4, 15].

On the other hand, very few investigations have been devoted to learning in coalition formation, an area of game theory that is exceedingly relevant to multiagent systems. Classical models of coalition formation assume the values of all coalitions are known *a priori*, but this assumption is unreasonable in many (indeed, almost all) multiagent settings. In [3], a new model for coalition formation was proposed, where agents must learn the values of possible coalitions. Moreover, the (strongest) classical solution concept, called *the core*, is replaced by a modified version called *the Bayesian core*.

All of the aforementioned papers adopt *reinforcement learning*, or some slightly doctored version, as the learning technique; each agent maintains an estimated model of the environment, based on past experiences. The model is updated as the agent repeatedly interacts with other agents and the environment, but at each stage an agent must base its decisions on the current model it possesses.

Surprisingly, only a handful of researchers have attempted to apply PAC[1] (Probably Approximately Correct) learning theory to multiagent settings, although this model has been extensively studied by researchers in computational learning theory. One example of PAC research in a multiagent model can be found in [5], where PAC learning is used to identify agents that can provide specific services. Another investigation [13] introduces a framework used to predict the behavior of

[1] The PAC model is also known as the *formal* model.

a multiagent system with learning agents, and uses PAC learning theory to obtain bounds on the values of the learning parameters.

In this paper, we focus on simple cooperative games, in which each coalition is either "winning" or "losing", the assumption being that if a coalition is winning, any coalition that contains it is also winning. Simple games are a suitable model for numerous n-person conflict situations. An example of such a game is voting on a bill in a parliament (with, say, 120 members), where a majority of votes has to be obtained in order to pass the bill; a winning coalition is any coalition with at least 61 members. Given the set $N = \{1, 2, \ldots, n\}$ of players (agents[2]), we can concisely describe a simple cooperative game by maintaining a list of *minimum* winning coalitions; a coalition is winning if and only if it is a superset (in the weak sense) of some minimum winning coalition, and is losing otherwise. In this way, sets of minimum winning coalitions can be considered as functions from 2^N to $\{0, 1\}$. Let $\mathcal{C}^*$ be the class of such concepts.

In our setting, an agent is learning, in the PAC model, some target concept $c_t \in \mathcal{C}^*$, by training on a set of "sample" coalitions. The samples are labeled either 1 (for winning) or 0 (for losing), and are drawn from some *fixed* distribution on the instance space 2^N. Observing the conventions of PAC learning theory, the hypothesis that is produced by the learning process is expected to be ϵ-close to the target concept, with confidence $1 - \delta$.

It is important to note that this setting can be embedded into real-world environments. Consider a group of advisers to the president of some political or financial organization. Different advisers would like to pursue different courses of action. Some of the advisers are closer to the ear of the president; coalitions that include such powerful advisers are more likely to convince the president. Since the group of advisers influences the president's decisions on a daily basis, the "winning" coalitions of advisers can be identified by examining the decisions the president has made over a period of weeks or months. The assumption of a fixed distribution on the coalitions is justified by noting that coalitions are likely to form among advisers with some common agenda; hence the set of plausible coalition structures is not likely to change over a short period of time.

The Vapnik-Chervonenkis (VC) dimension, which is formally defined in Section 2, is a combinatorial measure of the "richness" of a concept class, and is proportional to the difficulty of learning the class. We prove that

$$\text{VC-dim}(\mathcal{C}^*) = \binom{n}{\lfloor n/2 \rfloor}.$$

We use our bounds on the VC dimension to estimate the training set size a consistent[3] algorithm needs in order to be "probably approximately correct", and present a specific algorithm that learns $\mathcal{C}^*$.

We attempt to lower the VC dimension by studying constrained classes of simple games. We demonstrate that some constraints greatly reduce the VC-dimension: requiring that there be a dictator player reduces the dimension to $\lfloor \log n \rfloor$, while a generalization of this constraint to a single winning coalition has a VC-dimension of n. We consider other interesting constraints, and show that they do not substantially reduce the VC dimension.

We wish to stress that although similar upper bounds on the sample complexity can be derived from the size of our (finite) concept classes, we still benefit from computing the VC dimension of such classes, since this also yields lower bounds on the sample complexity.

Interestingly, $\mathcal{C}^*$ is equivalent to the concept class of monotone DNF formulas, which is formally defined below. The latter concept class has not been sufficiently studied in the context of PAC learning and VC dimension. In particular, the constraints we study on simple games stem from game theoretic intuitions that are scarcely relevant when discussing boolean formulas.

The rest of the paper is organized as follows. In Section 2 we discuss known definitions and results. In Section 3 we present our results. In Section 4 we give our conclusions.

2 Preliminaries

We first provide a mathematical model of *simple* games. We then formally define the PAC model and the VC dimension, and present some relevant theorems. The monotone DNF learning problem

[2]Throughout the paper, we use the terms "agent" and "player" interchangeably.

[3]The algorithm is consistent in the sense that it produces a hypothesis that labels the examples in the training set correctly.

is also discussed. We conclude this section by stating Sperner's Theorem [1].

2.1 Simple Cooperative n-Person Games

In this subsection we follow Section 2.5 of [8].

A *cooperative n-person game in characteristic form with side payments* is a pair $(N; v)$, where $N = \{1, 2, \ldots, n\}$ is a set of players, and v is the *characteristic function*, which assigns a real number $v(C)$ to each *coalition* $C \subset N$. $v(C)$ is the *value* of C.

Simple games are games where each coalition has a value of either 1 or 0. A coalition C is said to be *winning* if $v(C) = 1$, and *losing* if $v(C) = 0$. 2^N, the powerset of N, is partitioned into $\mathscr{W}$, the set of winning coalitions, and $\mathscr{L}$, the set of losing coalitions. This partition is assumed to satisfy three properties:

1. $\emptyset \in \mathscr{L}$.
2. $N \in \mathscr{W}$.
3. $C_1 \in \mathscr{W} \wedge C_1 \subset C_2 \rightarrow C_2 \in \mathscr{W}$.

2.2 The PAC model and VC-dimension

In this subsection we give a very short introduction to the PAC model and the VC dimension. A comprehensive overview of the model, and results concerning the VC dimension, can be found in [14], and a more concise summary is available in [12] (the latter is the text we follow in the next few paragraphs).

In the PAC model, the learner is given a set of m instances $x_1, x_2, \ldots, x_m$ which are sampled i.i.d. (independent identically distributed) according to a distribution D over the sample space X. D is unknown, but is fixed throughout the learning process. In this paper, we assume the "realizable" case, where a target concept $c_t(x)$ exists, and the given training examples are $Z = \{x_i, c_t(x_i)\}_{i=1}^m$. Let $\mathcal{C} : X \rightarrow \{0, 1\}$ be the concept class; the error of a concept $h \in C$ is defined as

$$err(h) = \Pr_{x \sim D}[c_t(x) \neq h(x)].$$

$\epsilon > 0$ is a parameter given to the learner that defines the "accuracy" of the learning process: we would like to achieve $err(h) \leq \epsilon$. Notice that $err(c_t) = 0$. The learner is also given a "confidence" parameter $\delta > 0$, that provides an upper bound on the probability that $err(h) > 0$:

$$\Pr[err(h) > \epsilon] < \delta.$$

We now formalize the discussion above:

Definition 1.

1. A *learning algorithm* L is a function from the set of all training examples to $\mathcal{C}$ with the following property: given $\epsilon, \delta \in (0, 1)$ there exists an integer $m_0(\epsilon, \delta)$, such that for any distribution D on X, if Z is a sample of size at least m_0 where the samples are drawn i.i.d. according to D, then with probability at least $1 - \delta$ it holds that $err(L(Z)) \leq \epsilon$.
2. A concept class $\mathcal{C}$ is *PAC-learnable* if there is a learning algorithm for $\mathcal{C}$.

Remark 2. m_0 is known as the *sample complexity* of $\mathcal{C}$.

Definition 3. Let $\mathcal{C} : X \rightarrow \{0, 1\}$ be a concept class, $S = \{x_1, x_2, \ldots, x_m\} \subset X$, and let

$$\Pi_{\mathcal{C}}(S) = \{\langle h(x_1), h(x_2), \ldots, h(x_m)\rangle : h \in \mathcal{C}\}.$$

If $|\Pi_{\mathcal{C}}(S)| = 2^m$, then S is considered *shattered* by $\mathcal{C}$.

In other words, if S is shattered by $\mathcal{C}$, then $\mathcal{C}$ realizes all possible dichotomies on S.

Definition 4. The *Vapnik-Chervonenkis (VC) dimension* of a concept class $\mathcal{C}$, denoted VC-dim($\mathcal{C}$), is the size of the largest set S that is shattered by $\mathcal{C}$. If $\mathcal{C}$ shatters arbitrarily large sets, then VC-dim($\mathcal{C}$) $= \infty$.

Lemma 5. *Let $\mathcal{C}$ be a finite concept class. Then VC-dim($\mathcal{C}$) $\leq \lfloor log_2|\mathcal{C}|\rfloor$.*

Proof. If the VC-dimension is d, a set of cardinality d is shattered by $\mathcal{C}$. Therefore, $\mathcal{C}$ realizes at least 2^d different dichotomies, and thus:

$$|\mathcal{C}| \geq 2^d \;\Rightarrow\; d \leq \lfloor \log_2|\mathcal{C}|\rfloor.$$

□

The following theorem gives an upper bound on the sample complexity of learning a given concept class, i.e., the size of the training set that is required to insure that a consistent hypothesis is ϵ-accurate with confidence $1-\delta$ [2].

Theorem 6 (Double Sampling)**.** *Let $\mathcal{C}$ be any concept class of VC dimension d. Let L be an algorithm such that, when given a set S of m labeled examples $\{(x_i, c(x_i))\}_i$ of some $c \in \mathcal{C}$, sampled i.i.d. according to some fixed but unknown distribution over the instance space X, produces as output a concept $h \in \mathcal{C}$ that is consistent with S. Then L is an (ϵ, δ)-learning algorithm for $\mathcal{C}$ provided that the sample size obeys:*

$$m \geq O\left(\frac{1}{\epsilon}log\frac{1}{\delta} + \frac{d}{\epsilon}log\frac{1}{\epsilon}\right).$$

Interestingly, the VC dimension also gives a lower bound on the sample complexity that almost matches the upper bound [6].

Theorem 7. *Let $\mathcal{C}$ be a concept class of VC dimension d. Then any (ϵ, δ)-learning algorithm for $\mathcal{C}$ must use sample size*

$$m \geq \Omega\left(\frac{1}{\epsilon}log\frac{1}{\delta} + \frac{d}{\epsilon}\right).$$

Remark 8. Since the upper and lower bounds are very close, any algorithm that is consistent with given training sets is nearly optimal asymptotically, in terms of sample complexity.

2.3 Monotone DNF

A concept class that will be shown to be equivalent to the one with which we concern ourselves is the class of monotone DNF formulas. A monotone DNF formula over variables $x_1, \ldots, x_n$ is a disjunction of terms, where each term is a monotone monomial in the variables over which the formula is defined. In other words, each term is a conjunction of literals in which no literal appears negated.

The following result appears as Lemma 6 in [9]:

Lemma 9. *For $1 \leq k \leq n$ and $1 \leq l \leq \binom{n}{k}$ let C be the class of functions expressible as l-term monotone k-DNF formulas, and let m be any integer, $k \leq m \leq n$ such that $\binom{m}{k} \geq l$. Then $VC-dim(C) \geq kl\lfloor log_2\frac{n}{m}\rfloor$.*

Another relevant result is formulated in [11]:

Proposition 10. *The VC dimension for k-term monotone DNF is less than $nk - k \cdot log\frac{k}{e}$.*

2.4 Sperner's Theorem

In this subsection we introduce Sperner's Theorem [1], which will be employed in the proof of Proposition 13. Given the numbers $\{1, 2, \ldots, n\}$, we would like to find a maximal antichain of subsets, i.e., a family of subsets such that for any two subsets, neither one is contained in the other. Finding an antichain of size $\binom{n}{\lfloor n/2\rfloor}$ is easy: we simply choose all the subsets of size $\lfloor n/2\rfloor$ (see Table 1). But can one do better? The theorem gives a negative answer.

Theorem 11 (Sperner's theorem)**.** *Let $\mathcal{F}$ be a family of subsets of $\{1, 2, \ldots, n\}$, such that for all $A, B \in \mathcal{F}$: $A \nsubseteq B$. Then $|\mathcal{F}| \leq \binom{n}{\lfloor n/2\rfloor}$.*

		$\emptyset$			
	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	
{1,2}	**{1,3}**	**{1,4}**	**{2,3}**	**{2,4}**	**{3,4}**
	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	
		$\{1,2,3,4\}$			

Table 1: Subsets of {1,2,3,4}, sorted by size. A maximal antichain is constructed by choosing all subsets of size 2; Sperner's Theorem states that one cannot construct a larger antichain.

3 Results

We now present our results, divided into two subsections. The first focuses on the general scenario of identifying winning coalitions in simple cooperative games, while in the second subsection we place constraints on the games in order to reduce the VC dimension.

3.1 Unconstrained Simple Games

Recall that in a simple game, if $C \subset N$ is winning, any superset of C is also winning. Thus, the game can be concisely represented by the set of minimum winning coalitions: C is a minimum winning coalition if and only if

$$C \in \mathscr{W} \wedge \forall i \in C,\ (C \setminus \{i\}) \in \mathscr{L}.$$

In this way, a set of minimum winning coalitions may be considered as a function from the set of coalitions to $\{0,1\}$: a coalition is winning (labeled by 1) if and only if it is a superset (in the weak sense) of one of the minimum winning coalitions.[4]

Remark 12. Surprisingly, learning to identify minimum winning coalitions is equivalent to learning monotone DNF formulas. Indeed, one can associate players with variables, and coalitions with terms. A hypothesis consisting of a set of minimum winning coalitions is essentially a disjunction of terms. When learning minimum winning coalitions, the sample space X is the space of all coalitions; these can be identified with assignments to variables, where a coalition C induces an assignment of 1 to the variables associated with the players in C, and 0 to all other players. This equivalence will aid us later on.

Proposition 13. *Let $\mathcal{C}^*$ be a concept class, in which each concept is a set of minimum winning coalitions. Then:*

$$\textit{VC-dim}(\mathcal{C}^*) = \binom{n}{\lfloor n/2 \rfloor}.$$

Proof. We first show that the VC-dimension is at least $\binom{n}{\lfloor n/2 \rfloor}$, by producing a set of size $\binom{n}{\lfloor n/2 \rfloor}$ which is shattered by $\mathcal{C}^*$. Consider the set S of all coalitions of size $\lfloor n/2 \rfloor$. Clearly, any dichotomy on S can be realized by the concept that contains as minimum winning coalitions exactly the coalitions in S that are labeled by 1.

On the other hand, the VC-dimension is at most $\binom{n}{\lfloor n/2 \rfloor}$. Indeed, consider a set S of more than $\binom{n}{\lfloor n/2 \rfloor}$ coalitions. By Theorem 11, there are two coalitions $C_1, C_2 \in S$ such that $C_1 \subset C_2$. The dichotomy in which C_1 is labeled by 1 and C_2 is labeled by 0 cannot be realized. □

Observe Algorithm 1, which receives as input a training set with m coalitions C_i, whose labels are y_i, and returns a set of minimum winning coalitions. We claim that Algorithm 1 is consistent with any training set (assuming the realizable case, where there exists a target concept).

Proposition 14. *Let $\{(C_i, y_i)\}_{i=1}^m$ be a training set given as input to Algorithm 1. Assuming there exists a target concept that is consistent with the training set, then the algorithm returns a set $\mathscr{W}^m$ which is also consistent, i.e., for all $i = 1, \ldots, m$:*

$$y_i = 1 \Leftrightarrow \exists R \in \mathscr{W}^m \text{ s.t. } R \subset C_i.$$

[4] There is an exception to this rule: $\emptyset$ is always a losing coalition.

Algorithm 1 Given a training set $\{(C_i, y_i)\}_{i=1}^{m}$, learns a hypothesis in $\mathcal{C}^*$.

$\mathscr{W}^m \leftarrow \emptyset$
for $i = 1$ to m **do**
  if $y_i = 1 \ \wedge \ \forall R \in \mathscr{W}^m,\ R \nsubseteq C_i$ **then**
    $\mathscr{W}^m \leftarrow \mathscr{W}^m \setminus \{R \subset N : C_i \subset R\}$
    $\mathscr{W}^m \leftarrow \mathscr{W}^m \cup \{C_i\}$
  end if
end for
return $\mathscr{W}^m$

Proof. Assume first that $y_i = 1$. If in iteration i of the for loop there is no minimum coalition which is already contained in C_i, then C_i is added to $\mathscr{W}^m$. In subsequent iterations, a minimum coalition that is a subset of C_i may be replaced by a still smaller coalition, but such a coalition would also be a subset of C_i.

For $y_i = 0$, assume that $\mathscr{W}^m$ labels C_i incorrectly. Then there exists $R \in \mathscr{W}^m$ such that $R \subset C_i$. By the algorithm, R must be some C_j with $y_j = 1$. Hence, there exists no hypothesis that is consistent with the training set — a contradiction. □

Remark 15. By Theorem 6 we have that $\mathcal{C}^*$ is learnable, with sample complexity

$$O\left(\frac{1}{\epsilon}\log\frac{1}{\delta} + \frac{d}{\epsilon}\log\frac{1}{\epsilon}\right),$$

for $d = \binom{n}{\lfloor n/2 \rfloor}$. By Theorem 7, the sample complexity of $\mathcal{C}^*$ *for any consistent algorithm* is $\Omega\left(\frac{1}{\epsilon}\log\frac{1}{\delta} + \frac{d}{\epsilon}\right)$; hence, in terms of sample complexity, we are guaranteed that Algorithm 1 is almost asymptotically optimal.

3.2 Constrained Simple Games

Although C^* is learnable, when the number of agents is large, huge training sets may be needed in order to produce an accurate hypothesis. We wish to show that for some constrained games, the VC dimension of the appropriate concept classes is far smaller than that of $\mathcal{C}^*$. Therefore, by Theorems 6 and 7, identifying the minimum winning coalitions in these games is easier.

Veto games are cooperative games where any coalition with nonzero value contains a distinguished player, called the *veto* player. In some veto games, the inclusion of the veto player in a coalition is also a sufficient condition for the coalition to be winning, not just a necessary one; in this case, we say the veto player is a *dictator*. Some real-world n-person conflict situations are simple games with a dictator. Returning to our example with the group of advisers to a president of an organization, if the president is also considered one of the players, then he is a dictator.

That being the case, in order to identify winning coalitions in a simple game with a dictator, it is enough to pinpoint this distinguished player. The set of minimum coalitions contains exactly one coalition of cardinality 1: the dictator.

Proposition 16. *Let $\mathcal{C}_1$ be a concept class, in which each concept is a single winning coalition of cardinality 1. Then:*

$$\textit{VC-dim}(\mathcal{C}_1) = \lfloor \log n \rfloor.$$

Remark 17. $\mathcal{C}_1$ is equivalent to the class of monotone 1-DNF formulas with one literal. Substituting $k = 1$ and $l = 1$ in Lemma 9 immediately gives a lower bound. Nevertheless, we include our proof since it is far simpler than the one in [9].

Proof of Proposition 16. We begin by proving a lower bound on VC-dim($\mathcal{C}_1$). We will construct a set S of size $\lfloor \log n \rfloor$ which is shattered by $\mathcal{C}_1$. We describe an algorithm that builds the coalitions in S. Initially, all the $\lfloor \log n \rfloor$ coalitions in S are empty. For each subset of coalitions $T \subset S$, we add to every coalition in T a *new* player (an element of N that has not been added to any other subset of coalitions). In other words, we iteratively add players to the coalitions in S; each player is associated with a different subset of coalitions (see Table 2 for an example). In particular, we

C_1	C_2	C_3	$\lvert T\rvert$
1			
	2		1
		3	
4	4		
5		5	2
	6	6	
7	7	7	3

Table 2: Constructing a set of size $\lfloor \log n \rfloor = 3$ which is shattered by $\mathcal{C}_1$, for $n = 8$. The three coalitions in S are C_1, C_2, and C_3, which are initialized as empty. We first add a new player to each subset $T \subset S$ of size 1: player 1 is added to C_1, i.e., to all coalitions in the subset $\{C_1\} \subset S$; 2 is added to C_2, and 3 is added to C_3. Next, the algorithm adds a new player to each subset of coalitions of size 2: 4 is added to C_1 and C_2, i.e., to all coalitions in the subset $\{C_1, C_2\}$; 5 is added to C_1 and C_3, and 6 is added to C_2 and C_3. Finally, 7 is added to C_1, C_2 and C_3. The algorithm yields the coalitions: $C_1 = \{1, 4, 5, 7\}$, $C_2 = \{2, 4, 6, 7\}$, and $C_3 = \{3, 5, 6, 7\}$. Notice that 8 is not in $C_1 \cup C_2 \cup C_3$ — it was intuitively added to all coalitions in the empty subset of coalitions.

Algorithm 2 Given a training set $\{(C_i, y_i)\}_{i=1}^{m}$, returns a hypothesis in $\mathcal{C}_1$.

$W \leftarrow N$
for $i = 1$ to m **do**
  if $y_i = 1$ **then**
    $W \leftarrow W \cap C_i$
  else
    $W \leftarrow W \setminus C_i$
  end if
end for
return some player in W

"add" to the empty subset of coalitions a new player, thus guaranteeing that there is a player that is not a member of any of the coalitions in S. We first notice that S is shattered by $\mathcal{C}_1$. Consider a dichotomy where some subset of coalitions $T \subset S$ is labeled by 1; by the construction of S, all coalitions in T have a common player, which is not a member of any other coalition in $S \setminus T$. Choosing this player as the dictator realizes the dichotomy. Moreover, we used a total of

$$2^{|S|} = 2^{\lfloor \log n \rfloor} \leq n$$

"new" players in the process of constructing S.

For a lower bound, we have from Lemma 5 that:

$$|\mathcal{C}_1| = |N| = n \Rightarrow \text{VC-dim}(\mathcal{C}_1) \leq \lfloor \log n \rfloor.$$

□

Consider Algorithm 2. It is clearly a learning algorithm for $\mathcal{C}_1$. However, it is not at all certain that it is optimal, in terms of computational complexity.

It is worthwhile to generalize our requirement of a single dictator player; we now wish to analyze simple games where there is a *junta* coalition, i.e., a coalition W such that for all coalitions C, C is winning if and only if $W \subset C$. Since W is the only minimum winning coalition, the goal of the learning process in such games is to recognize the junta coalition.

Proposition 18. *Let $\mathcal{C}_1^*$ be a concept class, in which each concept is a single minimum coalition. Then:*

$$\textit{VC-dim}(\mathcal{C}_1^*) = n.$$

Proof. We first bound the VC-dimension of $\mathcal{C}_1^*$ from below, by producing a set of size n which is shattered by $\mathcal{C}_1^*$. Observe the set S of coalitions of size $n-1$; there are n such coalitions. Consider a dichotomy on S that labels exactly $k \leq n$ of these coalitions by 1, and denote these k coalitions by $S^+ = \{C_{i_1}, \ldots, C_{i_k}\}$, $C_{i_k} = N \setminus \{i_k\}$. Let $C_0 = \bigcap_{l=1}^{k} C_{i_l} = N \setminus \{i_1, i_2, \ldots, i_k\}$. Notice that any subset $C \subset N$ such that $|C| = n-1$ and $C \notin S^+$ satisfies $C_0 \not\subseteq C$, since $C = N \setminus \{j\}$, and $j \notin \{i_1, i_2, \ldots, i_k\}$, so $j \in C_0$. Hence, the dichotomy is realized by choosing C_0 as the junta coalition.

In order to bound the VC-dimension from above, we invoke Lemma 5:

$$|\mathcal{C}_1^*| = |\{S : \ S \subset N\}| = 2^n \Rightarrow \text{VC-dim}(\mathcal{C}_1^*) \leq n.$$

□

A hypothesis that is consistent with a given training set (assuming there is a junta coalition) can be obtained by simply taking the intersection of all coalitions C_i with $y_i = 1$; this is essentially the well known ELIMINATION algorithm [12].

Another possible generalization is having at most k minimum winning coalitions. An upper bound for this case is given by Proposition 10. In particular, if $k = O(1)$, then VC-dim$(\mathcal{C}_k^*) = O(n)$.

We proceed by examining a different type of constraint:

$$\forall S \subset N, \ S \in \mathscr{W} \rightarrow N \setminus S \in \mathscr{L} \tag{1}$$

Simple games that satisfy equation (1) are known as *proper* simple games. However, this constraint does little to reduce the VC dimension, compared with unconstrained simple games.

Proposition 19. *Let $\mathcal{C}_p^*$ be a concept class, in which each concept is a set of minimum winning coalitions in a proper simple game. Then:*

$$\textit{VC-dim}(\mathcal{C}_p^*) \geq \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}.$$

Proof. We must exhibit a set S of size $\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$ which is shattered by $\mathcal{C}_p^*$. Let

$$S = \left\{ C \subset N : \ 1 \in C \wedge |C| = \left\lfloor \frac{n-1}{2} \right\rfloor + 1 \right\}.$$

It holds that the cardinality of S is as desired. Moreover, for any dichotomy on S, one may choose exactly the subsets labeled by 1 as the minimum winning coalitions: since the intersection of all subsets in S is not empty, the constraint (1) is not violated. □

A popular constraint is the elimination of *dummy* players:

$$\forall i \in N \exists C \subset N \ s.t. \ i \in C \wedge C \in \mathscr{W} \wedge (C \setminus \{i\} \in \mathscr{L}). \tag{2}$$

The elimination of dummy players also does not substantially reduce the VC dimension.

Proposition 20. *Let $\mathcal{C}_d^*$ be a concept class, in which each concept is a set of minimum winning coalitions in a simple game with no dummy players. Then:*

$$\textit{VC-dim}(\mathcal{C}_d^*) \geq \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}.$$

Proof. We must exhibit a set S of size $\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$ which is shattered by $\mathcal{C}_d^*$. Let

$$S = \left\{ C \subset N : \ 1 \notin C \wedge |C| = \left\lfloor \frac{n-1}{2} \right\rfloor \right\}.$$

It holds that the cardinality of S is as desired. Given a dichotomy on S, let S^+ be the set of coalitions in S that are labeled by 1, and let B be the set of players that are not members of any of the coalitions in S^+. The set of minimum winning coalitions is $S^+ \cup \{B\}$; the purpose of including B is to ensure that constraint (2) is not violated. It remains to show that this is a legitimate set of minimum coalitions in a proper simple game, which realizes the dichotomy. Since $1 \in B$, B is not a subset of any of the coalitions in S, and in particular is not a subset of any of the coalitions in $S \setminus S^+$ — so it is not the case that the addition of B to the set of minimum winning coalitions mistakenly labels a coalition in $S \setminus S^+$ by 1. Clearly, neither is B a superset of any of the coalitions in S^+; thus it is not the case that there are two winning coalitions such that one is a superset of the other. Since the coalitions in S^+ are included in the set of minimum winning coalitions, the dichotomy is realized. Moreover, constraint (2) is satisfied: every $i \in N$ is contained in one of the coalitions in S^+, or in B, and these are all *minimum* winning coalitions. $\square$

4 Conclusions

Simple games have natural interpretations in multiagent systems. Such games can be concisely represented by the set of minimum winning coalitions; this set can be regarded as a function from the set of coalitions to $\{0, 1\}$. We have shown that the VC-dimension of this concept class is $\binom{n}{\lfloor n/2 \rfloor}$, and presented an algorithm that learns the class. We have also discussed constrained simple games, and proved that restricting the set of minimum coalitions to a single coalition, or a dictator player, greatly reduces the VC-dimension. Nevertheless, other popular constrained games are almost as hard to learn as unconstrained games.

We believe there exist many opportunities to apply the PAC learning model in multiagent systems, and intend to investigate such applications in future work.

5 Acknowledgment

This work was partially supported by grant #039-7582 from the Israel Science Foundation.

References

[1] V. K. Balakrishnan. *Theory and Problems of Combinatorics.* Schaum's Outline Series. McGraw-Hill, 1995.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[3] G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1090–1097, 2004.

[4] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 83–90, 2003.

[5] G. Cybenko and G. Jiang. Matching conflicts: Functional validation of agents. In *Proceedings of the AAAI Workshop on Agent Conflicts*, pages 14–19, 1999.

[6] A. Ehrenfeucht, D. Haussler, M.Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

[7] A. Greenwald and K. Hall. Correlated Q-learning. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 242–249, 2003.

[8] J. P. Kahan and A. Rapoport. *Theories of Coalition Formation.* Lawrence Erlbaum Associates, 1984.

[9] N. Littlestone. Learning quickly when irrelevant attributes abound. *Machine Learning*, 2(4):285–318, 1988.

[10] M. L. Littman. Friend-or-Foe: Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 322–328, 2001.

[11] M. Schmitt. On the capabilities of higher-order neurons: A radial basis function approach. *Neural Computation*, 17(3):715–729, 2005.

[12] A. Shashua. Lecture notes on machine learning. Available from http://www.cs.huji.ac.il/course/2004/iml/handouts2004.html, 2004. Lectures 1, 2, 6, 7.

[13] J. M. Vidal and E. H. Durfee. Predicting the expected behavior of agents that learn about agents: The CLRI framework. *Autonomous Agents and Multi-Agent Systems*, 6(1):77–107, 2003.

[14] M. Vidyasagar. *A Theory of Learning and Generalization*. Spinger-Verlag, 1997.

[15] M. Weinberg and J. S. Rosenschein. Best-response multiagent learning in non-stationary environments. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 506–513, 2004.