Learning Cooperative Solution Concepts From Voting Behavior

A Case Study on the Israeli Knesset

Paper #XXXX

ABSTRACT

Most frameworks for computing solution concepts in hedonic games are theoretical in nature, and require complete knowledge of all agent preferences, an impractical assumption in real-world settings. Recent work lays the theoretical foundation of Probably Approximately Correct (PAC) stability to model stability under uncertainty through sampling. This paper presents the first application of strategic hedonic game models on real-world data. We show that PAC stable solutions can reflect Members of Knesset' political positions at large. Moreover, these comparisons also reveal politicians who are known to deviate from party lines. Finally, we show that PAC hedonic game models compare favorably to both k-means clustering and stochastic block models, which do not account for strategic behavior, for uncovering voting patterns.

1 INTRODUCTION

Hedonic games are a standard model for preference-constrained coalition formation, offering a variety of solution concepts accounting for criteria of justice and strategic considerations. They consider both cooperative aspects as well as individual preferences, combining, in a sense, cooperative and non-cooperative game theoretical settings.

Much of the work on these models focuses on theoretical analysis, rather than empirical or data-driven aspects (with the notable exception of stable matching problems, which, while can be modeled as hedonic games, have developed unto a field of their own). This is due to several factors: first, there are no available large-scale datasets of player preferences over coalitions. Second, most hedonic coalition formation algorithms are meant to be used in full-information domains, and cannot be immediately applied when only observations of partial player preferences are available. Even in domains where preferences are available, there is often no "ground truth" coalition structure to evaluate success.

In this work, we propose a methodology for addressing these issues, via a case-study identifying *probably* core-stable partitions in the Israeli parliament (the *Knesset*). In March 2017, Knesset historical voting data was made publicly available[1]. In the parliamentary setting, Members of Knesset (MKs) can be thought of as players, whose preferences can be induced from their voting patterns. The Israeli parliament votes frequently: more than 7000 bills and proposals within approximately 4 years (March 2015–April 2019). More importantly, the Knesset offers a natural ground truth for comparison: its members' *actual* political party affiliations.

Exploiting this novel data source and recent theoretical frameworks for applying hedonic solution concepts on data, we present a novel application of strategic hedonic games models on a real-world dataset. Our work addresses the following research questions: (1) Can hedonic games model real-world collaborations? (2) Are hedonic game outcomes comparable to the ground truth? (3) How do these outcomes compare to machine learning (ML) models?

1.1 Our Contributions

We use voting records of the Israeli Knesset to construct hedonic game models and corresponding solutions. We compare these against the outputs of standard clustering and community detection techniques, using party affiliations as ground truth. Our hedonic game-based partitions reveal Members of Knesset' political positions at large; moreover, our methodology compares well to standard ML techniques, even identifying 'rogue' MKs who break from party lines.

We utilize the notion of *PAC stability*, introduced in Sliwinski and Zick [17]; this methodology allows us to directly find *probably* stable coalition structures. That is, we are able to create coalition structure that is stable *according to the information we have* (though there is a small probability we have not seen a datapoint that would render our coalition structure unstable). To do so, we design efficient algorithms that compute PAC stable outcomes under various player preference models. We provide an intuitive graphic environment to present our results (https://knesset.s3.amazonaws.com/index.html, which contains additional models and analysis not included in this paper[2]).

1.2 Related Work

Most works on coalition formation games are theoretical; specifically, studies on hedonic games focus on the existence and computational aspects of stability concepts under various player utility models [1–4, 7, 18]. These works provide solid theoretical and algorithmic foundations for diverse classes of hedonic games; however, they all assume knowledge of player preferences, which is often unattainable in real-world settings.

There exist datasets of election data (e.g., PrefLib [14]), but unlike the Knesset dataset, they consist of voters ranking a fixed set of candidates, rather than repeatedly voting on different issues. Thus, the resulting datasets are far less heterogeneous and smaller in size. Some works study coalition formation under uncertainty, taking a Bayesian reinforcement learning approach [8, 9]. In this model, players reason about other players' capabilities and actions, and

[1] https://main.knesset.gov.il/Activity/Info/pages/databases.aspx

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

https://doi.org/doi

[2] We will provide access to the source code and cleaned datasets upon publication.

learn their preferences via repeated interactions, converging to a Bayesian variant of the core.

We take a data-driven approach to cooperative games, where one assumes access to a dataset (rather than a prior) of coalitions and their values, and uses the dataset to generate approximately stable solutions [6, 12, 13, 17]. The models we use rely in part on the theoretical foundations of PAC stability laid out in these works.

2 PRELIMINARIES

In this work, we model Members of Knesset (MKs) as players in a *hedonic game*. A hedonic game is given by a set of players $N = \{1, \ldots, n\}$. Let $\mathcal{N}_i = \{S \subseteq N : i \in S\}$ be the set of all subsets (known as *coalitions*) containing player i. Each player $i \in N$ has a complete preference order $\succ_i$ over $\mathcal{N}_i$. Most works assume that players' *ordinal* preferences are encoded via *cardinal* utilities; in other words, players have a utility function $v_i : \mathcal{N}_i \to \mathbb{R}$ such that $S \succ_i T$ iff $v_i(S) > v_i(T)$. A *solution* to a hedonic game is a mapping whose input is a hedonic game, outputting a set of partitions (known as *coalition structures*) satisfying certain fairness/stability criteria. The *core* is one of the more popular solution concepts in the literature. A coalition structure π is in the core if for every coalition $S \subseteq N$, at least one member of S weakly prefers their assigned coalition to S; in other words, if $\pi(i)$ is the coalition that i is assigned to under π, then π is in the core if for every $S \subseteq 2^N$ there exists some $i \in S$ such that $\pi(i) \succeq_i S$. Coalition structures in the core are often referred to as *stable*.

2.1 PAC Stability in Hedonic Games

As mentioned earlier, we assume only partial access to player preferences. More specifically, we assume that we are given a *dataset* $S_1, \ldots, S_m$ of m observations of formed coalitions, where each data entry is a coalition $S_j \subseteq N$, and the (cardinal) valuations of players in S_j $(v_i(S_j))_{i \in S}$. We assume that $S_1, \ldots, S_m$ are sampled i.i.d. from some distribution $\mathcal{D}$, and that future coalitions will be sampled from the same distribution. This is a natural assumption in data analysis, where $S_1, \ldots, S_m$ are the *training data* (used to train a model, or in our case, a solution concept), and future samples are taken from the *test data*. Indeed, in our experimental evaluation, we take i.i.d. samples from the Knesset voting data, which forms our training data. Our algorithms are provably guaranteed to offer *probably stable* solutions, as described below.

Hedonic core stability can be thought of as capturing a *local loss*: given a coalition structure π and a coalition $S \subseteq N$, let

$$\lambda(\pi, S) = \begin{cases} 1 & \text{if } \forall i \in S : v_i(S) > v_i(\pi(i)) \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $\lambda(\pi, S)$ incurs a loss of 1 if π was not able to hedge against the members of S deviating from π. Given a distribution $\mathcal{D}$, one can then naturally define the *expected loss* of π w.r.t. $\mathcal{D}$ as

$$L_{\mathcal{D}}(\pi) = \Pr_{S \sim \mathcal{D}} [\lambda(\pi, S) = 1] \tag{1}$$

Equation (1) captures a probabilistic variant of the core condition: rather than requiring that $\lambda(\pi, S) = 0$ for all $S \subseteq N$ (as is the case for the core), we require that it is low w.r.t. $\mathcal{D}$. Thus, our objective is to find coalition structures that incur low expected loss. More formally, a *PAC stabilizing* algorithm takes as input a set of i.i.d. samples $S_1, \ldots, S_m \sim \mathcal{D}^m$, and outputs a coalition structure π^* (a function of the samples) with the following guarantee:

$$\Pr_{(S_1, \ldots, S_m) \sim \mathcal{D}^m} [L_{\mathcal{D}}(\pi^*) \geq \varepsilon] < \delta \tag{2}$$

Intuitively, δ captures the probability that the i.i.d. observations given to our algorithms are 'badly distributed' (e.g., $\mathcal{D}$ is a uniform distribution, but by sheer coincidence we sampled the same coalition every single time). In other words, in a vast majority of the m samples ($\geq 1 - \delta$) the output of our PAC stabilizing algorithm incurs $< \varepsilon$ expected loss. We require that m, the number of samples needed to offer the guarantee in (2), is polynomial in n, $\frac{1}{\varepsilon}$ and $\log \frac{1}{\delta}$. Note that the above formulation completely sidesteps the need to learn players' preferences, directly learning a stable outcome from samples. Indeed, a series of recent works [12, 13, 17] present efficient algorithms for computing PAC stable outcomes. In fact, Jha and Zick [13] show that only *consistency* with samples is needed to ensure PAC stability, using a number of samples linear in n: an algorithm is a *consistent solver* if given a set of samples $S_1, \ldots, S_m$ evaluated by a hedonic game (N, v), its output π^* satisfies $\lambda(\pi^*, S_j) = 0$ for all $j \in 1, \ldots, m$. In other words, a coalition structure that is stable w.r.t. to the observed samples is likely to be stable w.r.t. future samples, for a sufficiently large m.

2.2 The Israeli Knesset Data

The Israeli political system consists of multiple parties, partially due to its proportional voting system, and diverse political landscape. The Knesset is the unicameral legislative branch of the national government. We focus on the twentieth Knesset (2015-2019), which was the longest serving Knesset since the late '80s. The twentieth Knesset had ten parties; however, its political landscape is far more nuanced[3]. This preponderance of political parties (and respective views) has favored a multi-dimensional party system, leading to unexpected coalitions and combinations. However, in the past decades Israeli parties generally align along a single right-left axis, which mainly concerns the parties' approach to the Israeli-Palestinian conflict. This simplifies our considerations when analyzing and comparing the models.

No Israeli party has ever held a majority of Knesset seats; all governments are made of coalitions. In the past years (including this dataset) governments strictly enforced coalition voting compliance via a government committee which issues mandatory voting instructions for all proposed bills. The opposition is under no such control, but due to their relative ideologies, display significant agreement.

The Knesset website provides data access through the Open Data Protocol (OData) on past MKs, bills, and member votes on every bill. The Knesset has 120 seats, but the twentieth Knesset has 147 members due to some MKs resigning or joining mid-term. We retrieve information on all 147 MKs' information including name, party affiliation and their votes for all 7515 bills deliberated[4]. A member's vote can take on one of the following values: 0 (vote canceled), 1 (vote for), 2 (vote against), 3 (abstained), and 4 (did not attend).

[3] See http://bit.ly/2sJUZEi-knesset20 for an overview.

[4] The data contains inconsistencies when cross-checked against another dataset on the Knesset website listing passed / defeated bills. This was likely due to MKs changing votes manually after the voting period, rather than through the electronic system as required. We resolve this by removing the 26 conflicting bills.

3 METHODOLOGY

Previous work [5, 22] involve learning the underlying *complete preference profile* before finding a stable partition. The closer the learned preference profile is to reality, the more likely we are to observe theoretically stable partitions matching the ground truth partition in real-world data.

The problem with this approach lies in verifying the fidelity of the learned preference profile — even simple preference profiles such as *top responsive games* have an exponentially large representation in the number of players (i.e. they have high VC dimension, as noted in Sliwinski and Zick [17]). In the Knesset dataset, a naive description of player preferences will result in a list with approximately $2^{146} \approx 10^{43}$ coalitions. Approximating these rankings is not only computationally intensive, but also somewhat unrealistic: MKs are not likely to have a complete model of their own preferences.

PAC stability inspires an alternative approach: instead of first learning the complete profile, and computing a stable partition with respect to the learned preferences, we directly learn a PAC stable partition from the partial preference relations observed in the Knesset data. This trades off the strong stability guarantees for probabilistic stability guarantees, but makes the problem much more tractable.

Hedonic game models require players to submit a complete ranking of coalitions they belong to. In the Knesset data however, we observe only approval/disapproval of bills; we translate the voting data to preference relations of each parliament member, satisfying the requirements of the given model. If the formulation results in a complete preference profile, we use it to compute a deterministic stable partition. We refer to such models as "full-information" models. These models reflect the aforementioned approach of learning the underlying preference profile first, followed by finding a stable partition. We then attempt to "PAC-ify" the formulation to discover a PAC stable partition directly from sample data, i.e. partial preference relations inferred from voting behaviors observed in a subset of all bills.

For each PAC model, we sample i.i.d. (with replacement) $\frac{3}{4}$ of all bills; we repeat the run 50 times to evaluate the consistency between different runs. We use an information-theoretic measure, *adjusted mutual information (AMI)* [23], to determine if the coalition structure we generate are robust with respect to the sampling. AMI is sensitive in detecting very different partitions; however, AMI is not metric, so it is not meaningful to compare two AMI values. To aggregate these 50 partitions into one single "representative" partition, we select the conceptual "centroid" of the 50 partitions by finding the partition that has the least information distance to the other 49 partitions.

Finally, we compare the partition produced by each model to ground truth party affiliations, both quantitatively and qualitatively. We use AMI to selected the most promising models and qualitatively examine them in greater detail. This lets us find subtleties not captured by quantitative measurements, such as MKs known to hold views that don't match party line.

4 HEDONIC GAME MODELS

MKs' political strategies are no doubt varied and complex. However, the need for successful voting blocs stylistically resemble coalition formation games; of these, we focus several more tractable hedonic games models for analysis.

4.1 Appreciation of Friends

In this model, players classify others as friends or enemies, and prefer coalitions with more friends and fewer enemies:

Formally, let G_i be player i's set of friends, and B_i the set of enemies. $G_i \cup B_i \cup i = N$ and $G_i \cap B_i = \emptyset$. A preference profile P^f is based on *appreciation of friends* if for all players $i \in N$, $S \succeq_i T$ if and only if $|S \cap G_i| > |T \cap G_i|$, or $|S \cap G_i| = |T \cap G_i|$ and $|S \cap B_i| \leq |T \cap B_i|$. Dimitrov et al. [10] propose Algorithm 1 for finding core stable partitions for these preference profiles.

Algorithm 1 Appreciation of Friends Core Finding

Input: A preference profile based on appreciation of friends.

$R^1 \leftarrow N$; $\pi \leftarrow \emptyset$.
for $k = 1$ to $|N|$ **do**
  Find S^k, the strongly connected component (SCC) in the graph induced by R^k, with the largest # of vertices.
  $\pi \leftarrow \pi \cup \{S^k\}$ and $R^{k+1} \leftarrow R^k \setminus S^k$
  if $R^{k+1} = \emptyset$ **then**
    return π
  end if
end for
return π

In the Knesset dataset, we define a player's friends as anyone whose votes agreed with the given player's more often than they disagreed. Agreed votes are only counted if the given player's vote is 'effective' — when it is either "for" or "against". We experimented with two different ways of counting the disagreed votes: (1) Broad disagreement (selective friends): the other player's vote is different from mine. (2) Narrow disagreement (general friends): the other player's vote is different from mine and is effective.

Broad disagreement leads to players being more selective of their friends, and friendships can be asymmetric. In both models, those who are not friends are considered enemies.

Example 4.1. Consider a scenario with 3 MKs voting on 3 bills.

players:	$P1$	$P2$	$P3$
bill A	for	for	against
bill B	abstained	for	abstained
bill C	abstained	against	abstained

Under general friends, $P1$ and $P2$ are friends because they agree on bill A. Under selective friends, from $P1$'s perspective, $P2$ is still a friend due to bill A, while $P2$ considers $P1$ an enemy because $P1$ did not vote with her for bills B and C.

Once the sets of friends and enemies are derived, they define a complete preference profile. We then apply Algorithm 1 to obtain a core stable partition for the general friends and selective friends preference profiles.

The appreciation of friends model is an example of a *top responsive* preference profile [10, 18]. Under such preferences, every player derives its utility from a most preferred subset of players within its coalition, breaking ties to favor smaller coalitions. Formally, given player i and coalition $S \in N_i$, the *choice set* of i in S is $Ch(i,S) = \{S' \subseteq S : (i \in S') \wedge (S' \succeq_i S'' \; \forall S'' \subseteq S)\}$. If $|Ch(i,S)| = 1$, the unique choice set is denoted as $ch(i,S)$. A *top responsive* preference profile requires that for any player $i \in N$, and any coalitions $S, T \in \mathcal{N}_i$: (a) $|Ch(i,S)| = 1$. (b) if $ch(i,S) \succ_i ch(i,T)$ then $S \succ_i T$ (c) if $ch(i,S) = ch(i,T)$ and $S \subset T$ then $S \succ_i T$.

Based on the PAC core finding algorithm introduced by Sliwinski and Zick [2017], we "PAC-ify" Algorithm 1 as Algorithm 2. Intuitively, at each iteration k, we sample a new set of coalitions to construct choice sets for each remaining player; Steps-5-13 initializes the choice sets, which are refined through Steps 14-20. Step 21 then selects a coalition S^k to add to the solution, and then removes its members from the remaining data (Step 22). In fact, Algorithm 2 can be applied to samples from *any* top responsive preference profile.

Algorithm 2 PAC Appreciation of Friends Core Finding

Input: ε, δ, set $\mathcal{S}$ of $m = (2n^4 + 2n^3)\lceil \frac{1}{\varepsilon} \log \frac{2n^3}{\delta} \rceil$ samples from $\mathcal{D}$

1: $R^1 \leftarrow N$, $\pi \leftarrow \emptyset$
2: $\omega \leftarrow \lceil 2n^2 \frac{1}{\varepsilon} \log \frac{2n^3}{\delta} \rceil$
3: **for** $k = 1$ to $|N|$ **do**
4: $\quad \mathcal{S}' \leftarrow$ take and remove ω samples from $\mathcal{S}$
5: $\quad \mathcal{S}' \leftarrow \{T : T \in \mathcal{S}', T \subseteq R^k\}$
6: $\quad$ **for** $i \in R^k$ **do**
7: $\quad\quad$ **if** $i \notin \bigcup_{X \in \mathcal{S}'} X$ **then**
8: $\quad\quad\quad B_{i,k} \leftarrow \{i\}$
9: $\quad\quad$ **else**
10: $\quad\quad\quad B_{i,k} \in \arg\max_{T \in \mathcal{S}'} v_i(T)$
11: $\quad\quad\quad B_{i,k} \leftarrow \bigcap_{\{T \in \mathcal{S}' : ch(i,T) = ch(i, B_{i,k})\}} T.$
12: $\quad\quad$ **end if**
13: $\quad$ **end for**
14: $\quad$ **for** $j = 1, \ldots, |R^k|$ **do**
15: $\quad\quad \mathcal{S}'' \leftarrow$ take and remove ω samples from $\mathcal{S}$
16: $\quad\quad \mathcal{S}'' \leftarrow \{T : T \in \mathcal{S}'', T \subseteq R^k\}$
17: $\quad\quad$ **for** $i \in R^k$ **do**
18: $\quad\quad\quad B_{i,k} \leftarrow B_{i,k} \cap \bigcap_{T \in \mathcal{S}'' : ch(i,T) = ch(i, B_{i,k})} T.$
19: $\quad\quad$ **end for**
20: $\quad$ **end for**
21: $\quad$ Find S^k, the strongly connected component (SCC) in the graph induced by R^k with the largest # of vertices.
22: $\quad \pi \leftarrow \pi \cup \{S^k\}$; and $R^{k+1} \leftarrow R^k \setminus S^k$
23: $\quad$ **if** $R^{k+1} = \emptyset$ **then**
24: $\quad\quad$ **return** π
25: $\quad$ **end if**
26: **end for**
27: **return** π

THEOREM 4.2. *Algorithm 2 outputs a PAC stable partition for any top responsive game; its running time is a factor of n faster than that of Sliwinski and Zick [17].*

PROOF SKETCH. The correctness of Algorithm 2 is largely derived from the correctness of a similar algorithm proposed by Sliwinski and Zick [17]. Briefly, Steps 4–20 are used to identify (probably correct) proxies for players' choice set amongst the remaining players. Steps 21-22 remove a set of players who likely form a stable coalition, and are the only point of difference between us and Sliwinski and Zick. Whereas Sliwinski and Zick remove the smallest connected component, we remove the *largest* strongly connected component, i.e. the largest subgraph s.t. there is a path between any two vertices. Finding the largest SCC maintains the correctness of the algorithm, and requires a factor of n less time (using Tarjan's algorithm [20]). □

Beyond the theoretical running time improvement, removing larger groups in earlier iterations significantly reduces Algorithm 2's running time compared to the CC procedure on our data. This reduces our runtime from 30 to 10 minutes[5].

We compute PAC stable outcomes based on the general friends and selective friends preference models. Due to random sampling in the PAC version of the experiments, we do not observe all bills every time we approximate players' choice sets; since we take intersections of the best coalitions for players, our PAC approximations are, intuitively, conservative estimates of the true choice sets; this is reflected in the performance of the PAC algorithms versus their full-information counterparts (see § 6.1).

4.2 Boolean Hedonic Games

Under Boolean preferences, each player either likes or dislikes a coalition, i.e. for every i and every $S \in \mathcal{N}_i$, $v_i(S) \in \{0, 1\}$. Aziz et al. [3] show that Algorithm 3 outputs a core-stable outcome for Boolean games.

Algorithm 3 Boolean Hedonic Game Core Finding

1: $N' \leftarrow N$; $\pi \leftarrow \emptyset$.
2: **while** $N' \neq \emptyset$ **do**
3: $\quad$ Find the largest size $S \subset N'$ s.t. for all $i \in S$, $v_i(S) = 1$.
4: $\quad \pi \leftarrow \pi \cup \{S\}$
5: $\quad N' \leftarrow N' \setminus S$
6: **end while**
7: **return** π

The Knesset data offers a natural method of identifying approved coalitions: given a bill, "for" voters and "against" voters form two satisfactory coalitions. This profile is always symmetric — if S is satisfactory for *some* $i \in S$, it is satisfactory for *all* $i' \in S$. Symmetry implies that the bill with the broadest support/disapproval also yields the largest coalition. Any bill with broad inter-party support/disapproval always generates a multi-party coalition in Algorithm 3.

Since players are indifferent among satisfactory coalitions, selecting any coalition (not necessarily the largest) whose members find satisfactory at Step 3 of Algorithm 3 to add to the output partition π maintains core stability. To avoid always including the coalition that corresponds to the bill with the largest multi-party support/disapproval, our implementation selects the median-sized coalition among all satisfactory coalitions in each iteration.

Algorithm 3 can be easily converted to a PAC variant: given a sampled set of coalitions $S_1, \ldots, S_m$, pick some approved S_j from the samples (say, the largest or median size approved S_j) at every iteration, set its players to be in a group together under π, and remove any coalitions in the sample that intersect with S_j; repeat until no coalitions are left. Let L be the set of unassigned players: place the members of L into singleton coalitions in π. The resulting partition π is trivially consistent with the samples: the members of every observed coalition that was added to π are known to approve of their assigned coalition, and would not deviate. In addition, it is not possible that some approved $S \in \{S_1, \ldots, S_m\}$ exists such that $S \subseteq L$: our PAC variant would have added S to π. Thus, the PAC variant of

[5] On a 2018 MacBook Air (i5 CPU, 16GB RAM)

Algorithm 3 is a consistent solver, and outputs a PAC stable outcome given $O(n)$ samples.

Observe that the PAC variant of Algorithm 3 is computationally efficient in the number of samples, whereas Algorithm 3 runs in time exponential in n. This also holds true for the PAC variant of Algorithm 1. That is, PAC learning approaches offer both a theoretically sound method of handling partial information, as well as computationally efficient performance.

5 MACHINE LEARNING MODELS

To study our models' effectiveness, we compare them against 2 machine learning techniques for grouping similar points (players): k-mean clustering (cluster analysis) and stochastic block models (community detection). These techniques are well studied in literature; we briefly overview them below:

k-means clustering [19] is a general-purpose method that divides a given set of samples (vectors of feature values) $\vec{x}_1, \cdots, \vec{x}_n$ into k disjoint sets $T_1, \ldots, T_k$, where each T_j is described by the mean $\vec{\mu}_j$ of its members. This technique produces a partition minimizing the *within-cluster sum-of-squares*: $\sum_{i=1}^{n} \min_{\vec{\mu}_j \in C} (||\vec{x}_i - \vec{\mu}_j||^2)$, where C is the set of k means. To use k-means clustering, we generate a vote-based feature vector $\vec{x}_i$ for each Knesset member i: each coordinate of $\vec{x}_i$ represents a bill, and its value is 1 if i voted for the bill; -1 if i voted against, and 0 otherwise. We use the elbow method [21] and the average silhouette score [16] to determine the "best" cluster sizes to be $k = 2$ and $k = 10$.

The **stochastic block model** (SBM) [11] is a community detection technique that, given a social network, partitions its nodes (players) into communities based on interaction intensity. We adopt the *non-parametric weighted stochastic block model* implementation by Peixoto [15] which incorporates edge weights. The stochastic block model partitions nodes so that nodes belonging to the same group connect to other groups at roughly the same rates.

We construct two SBM instances. In the first model, we use only positive edge weights: each edge weight represents the number of times a pair of politicians voted together, either "for" or "against" a bill, modeled using the geometric distribution. Our second SBM takes into account disagreements between politicians: edge weights are the *difference* between the number of times their effective votes agree and disagree, modeled using a normal distribution.

We use SBM and the MCMC algorithm to perform model averaging to selected the "most representative" model.

6 RESULTS & DISCUSSION

Our main visualization method is the *Sankey diagram*, which is a flow diagram representing a change in a system. On the left of each diagram we have the ground truth state — party affiliation — and on the right is one of our model partitions. Each link from the left party partition to the right model partition represents a parliament member. Richer, more detailed diagrams can be seen in our online demo. We color parties according to their political position: right wing parties are in reddish hues and left wing parties in blueish hues.

6.1 Quantitative Analysis

We use AMI to measure roughly each model's deviation from ground truth party affiliations. High AMI values represent models that capture more aspects of the ground truth. While we cannot compare them directly, we use AMI to select the most promising models for further investigation. Based on the AMI criterion, we focus our qualitative analysis on the PAC models (friends, selective friends and Boolean), and compare them against the SBM geometric community detection model, and the 2- and 10-group k-means partitions.

While we omit the AMI values due to space constraints, we highlight also a stark difference in AMI of full-information models: selective friends has a high AMI value, while general friends does poorly. This is a major drawback of these full-information models — sensitivity to how "friends" are defined; a slight change from narrow to broad produces two very different, yet each core stable coalition structures.

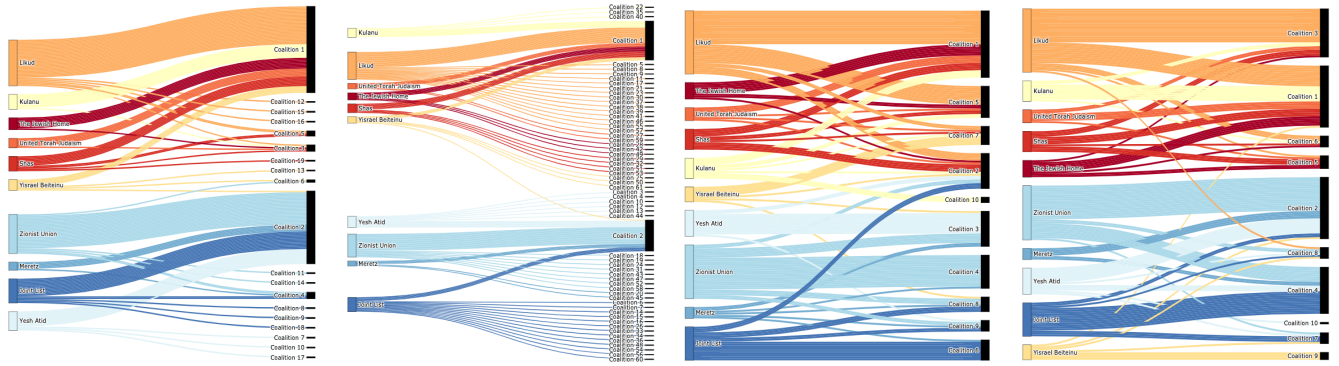
6.2 Qualitative Analysis

Some MKs participated in a small number of votes due to mid-term resignations or external duties. In both cases, low participation may skew cluster assignments. Our interactive demo provides an option to color the flow links representing MKs by their number of effective votes.

Coherence. Successful models should accurately separate government and opposition parties; indeed, our PAC models avoid grouping members from the left and right. Unfortunately, PAC Boolean also produces a large number of singleton coalitions (as it removes median sized groups from the samples it leaves several singletons behind). The 10-group k-means model, and the community based SBM geometric model tend to join members with fewer votes, regardless of their actual voting patterns. Examples of such cross-ideology infrequent-voter groups are Coalition 2 of the 10-group k-means model (see demo diagram k_10_means, color by vote count), and Coalition 8 of the SBM geometric model (demo: diagram smb_discrete-geometric, color by vote count). The 2-group k-means model partition includes five members of the right wing parties in the opposition group (Coalition 2 in demo diagram k_2_means), out of which only one such assignment makes sense — Orly Levy left the right/center-right Yisrael Beiteinu and formed her own party 'Gesher', which ran together with the center-left Labor Party in the 2019 Knesset elections.

Overall, the PAC models are more coherent: PAC selective friends has only two cross-ideological groups (demo: diagram pac_friends_selective) — Coalition 2 involves Levy, a known deviation; Coalition 6 is a small coalition combining two low-attendance members, one on the right edge of the left wing (Daniel Atar with 708 effective votes), and another, who switched between coalition and opposition (Avigdor Lieberman with 725 effective votes). PAC Boolean only has one cross-ideological group involving Orly Levy (demo: diagram pac_boolean). PAC friends exhibits similar behavior, with both Lieberman and Levy crossing party lines.

Overall structure. All PAC models and the 2-group k-means model distinguish between the main government and opposition groups. This stresses that, despite their differences, MKs still tend to vote together due to their relative ideological cohesion, especially within



(a) PAC selective friends **(b) PAC Boolean** **(c) k-means ($k = 10$)** **(d) SBM geometric**

Figure 1: Visualization of Ground Truth vs. Model Generated Partitions

opposition parties who lack binding instructions. The SBM geometric and 10-group k-means models, however, commingled government and opposition MKs (Figures 1d, 1c); most groups formed based on attendance. Both variations of PAC friends present low attendance MKs as singletons, while clustering together ministers, who tend to miss votes. Ministers cluster in two different coalitions in PAC selective friends: Coalitions 3 and 5 (demo: diagram pac_friends_selective). Both the 10-group k-means and the SBM geometric clusters tend to remove more ministers from the overall main government group, and divide them into more separate coalitions (demo: diagram k_10_means and smb_discrete-geometric). One notable exception is a right-wing subgroup formed under PAC friends (Coalition 3 in diagram pac_friends): in addition to some low-attendance members, it consists of right-wing markers of the Likud party (Oren Hazan and Yehuda Glick), as well as Benjamin Netanyahu and Moshe Kahlon (Israel's prime minister and minister of finance, respectively). In addition, it contains several members of Shas, a religious right-wing party. We hypothesize this group's members were particularly consistent in their voting patterns; in other words, these members were likelier to align their vote to government decisions.

7 CONCLUSION

We examine the Israeli parliament dataset using a data-driven approach that accounts for strategic behavior of parliament members. This study is one of the first to bridge the gap between theoretical aspects of hedonic coalition formation games and its applications to real world scenarios, using a newly available dataset and theoretical tools for handling coalitional stability in data-driven environments. We show that PAC models are not only able to recover the overall structure of governmental vs. opposition groups, but also are more coherent compared to machine learning models such as k-means and Stochastic Block Models. These PAC models also produce more robust partitions than their full-information counterparts. To our knowledge, ours is the first attempt to apply hedonic game models to make predictions on real data. We hope that future works will also take a *descriptive*, rather than *prescriptive*, approach to hedonic games.

REFERENCES

[1] José Alcalde and Pablo Revilla. 2004. Researching with whom? Stability and manipulation. *Journal of Mathematical Economics* 40, 8 (2004), 869–887.

[2] Haris Aziz and Florian Brandl. 2012. Existence of Stability in Hedonic Coalition Formation Games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 763–770.

[3] Haris Aziz, Paul Harrenstein, Jérôme Lang, and Michael Wooldridge. 2016. Boolean Hedonic Games. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. 166–175.

[4] Haris Aziz, Rahul Savani, and Hervé Moulin. 2016. Hedonic Games. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D.Editors Procaccia (Eds.). Cambridge University Press, 356—376.

[5] Maria-Florina Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. 2012. Learning Valuation Functions. In *Proceedings of the 25th Conference on Computational Learning Theory (COLT)*. 4.1–4.24.

[6] Maria-Florina Balcan, Ariel D. Procaccia, and Yair Zick. 2015. Learning Cooperative Games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 475–481.

[7] Anna Bogomolnaia and Matthew O. Jackson. 2002. The Stability of Hedonic Coalition Structures. *Games and Economic Behavior* 38, 2 (2002), 201–230.

[8] G. Chalkiadakis and C. Boutilier. 2004. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 1090–1097.

[9] Georgios Chalkiadakis and Craig Boutilier. 2008. Sequential Decision Making in Repeated Coalition Formation Under Uncertainty. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 347–354.

[10] Dinko Dimitrov, Peter Borm, Ruud Hendrickx, and Shao Chin Sung. 2006. Simple Priorities and Core Stability in Hedonic Games. *Social Choice and Welfare* 26, 2 (2006), 421–433.

[11] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social Networks* 5, 2 (1983), 109–137.

[12] Ayumi Igarashi, Jakub Sliwinski, and Yair Zick. 2019. Forming Probably Stable Communities with Limited Interactions. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*. 2053–2060.

[13] Tushant Jha and Yair Zick. 2019. A Learning Framework for Distribution-Based Game-Theoretic Solution Concepts. (2019). arXiv:1903.08322

[14] Nicholas Mattei and Toby Walsh. 2013. PrefLib: A Library of Preference Data HTTP://PREFLIB.ORG. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*.

[15] Tiago Peixoto. 2017. Nonparametric weighted stochastic block models. *Physical Review E* 97 (2017).

[16] Peter Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational Applied Mathematics* 20, 1 (1987), 53–65.

[17] Jakub Sliwinski and Yair Zick. 2017. Learning Hedonic Games. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*. 2730–2736.

[18] Koji Suzuki and Shao Chin Sung. 2010. Hedonic coalition formation in conservative societies. (2010). Manuscript.

[19] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. *Introduction to Data Mining (2Nd Edition)*. Pearson.
[20] Robert Tarjan. 1972. Depth first search and linear graph algorithms. *SIAM J. Comput.* 1, 2 (1972).
[21] Robert L. Thorndike. 1953. Who belongs in the family. *Psychometrika* (1953), 267–276.
[22] Daniel Vainsencher, Ofer Dekel, and Shie Mannor. 2011. Bundle Selling by Online Estimation of Valuation Functions. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*. 1137–1144.
[23] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research* 11 (2010), 2837–2854.