Identifying agent behaviors using Markov chain long-term probabilities

Matteo Murari, Alessandro Farinelli, and Riccardo Sartea

University of Verona, Department of Computer Science,
Strada le Grazie 15, 37134 Verona, Italy
{matteo.murari,alessandro.farinelli,riccardo.sartea}@univr.it

Abstract. Markov chains are a powerful tool to represent behaviors of intelligent agents, particularly when such behaviors are gathered by observing how agents interact with the environment or with other agents (e.g., in cyber-security applications). Within these contexts, the extracted Markov chains may contain a significant amount of noise because agents' actions may depend on several elements that may not be directly observed or because the agents themselves may try to hide their behavior (e.g., a malicious software trying to evade an analyzer agent). A recent work proposes the use of long-term probabilities extracted from Markov chains to identify known behaviors of intelligent agents. The key idea is that focusing on the long-term allows to discard noise that may be present in the model. Such work mainly focuses on domains related to cyber-security and particularly malware analysis. In this work we aim to investigate the use of similar techniques in a different application scenario, namely Vickrey auctions. In particular, we focus on the iterated Vickrey auction setting with the aim of identifying the various strategies employed by one of the bidders, modeling its behaviors with Markov chains. Independently of the specific strategy, the bidder tries to guess the real price of an item, hence introducing uncertainty in the bids, and consequently noise in the computed models. Results show that using the long-term transition probability is effective in diminishing the impact of uncertainty due to noise inserted in the behavioral models, converging to a better classification score compared to the classical 1-step transition probability.

Keywords: Markov chain · Learning agent behavior · Vickrey auction

1 Introduction

A fundamental approach employed to study complex real systems consists in abstracting such environments in order to represent the subjects that interact therein as rational agents. It is hence assumed that such agents collect information about the state of the environment and use it to execute suitable actions to achieve a given objective. Examples of such modeled domains include security, agriculture, medicine and so forth. A key task that has to be addressed in these scenarios is represented by modeling the agents' behaviors observed during the execution of the process under analysis. Indeed, the selected formalism affects the quality of the learning processes attempting to extract useful information from the models. Moreover, recently it has been highlighted that several systems where the agents interact are misled by the presence of noise factors. For instance, an agent can divert from its policy, consequently inducing the introduction of noise into the associated model.

A suitable model that is often employed in case of stochastic agent policies or to represent process affected by an uncertain evolution is the Markov chain. There is a significant corpus of works that propose Markov chains to express the behavioral profile shown by an agent and prove their effectiveness in agent learning process [14, 11, 15, 5, 4]. However, a crucial operation to identify the core behaviors embedded in such model consists in the feature extraction. A novel proposal given by [10] is to focus on the long-term transition probability, namely the probability that the current state of the agent moves to another whenever the process represented by the Markov chain reaches a stable constant configuration, i.e. a fixpoint. This work differs from similar previous proposals since it does not require any assumption about the agent's model to hold in order to extract long-term features: using such method, the long-term probability can be extracted from any instance of Markov chain. The main practical field of application that authors take into account is the analysis of Android malware.

The contribution of this paper is to demonstrate that such an approach can be effectively applied to any kind of analysis context that involve Markov chains as model to represent the dynamics of an agent. In particular, we focus on an interesting case of study, that is the identification of strategies played by an agent in a repeated Vickrey auction. We consider iterated Vickrey auctions as a benchmark for our purpose since inferring the participants' policy is typically not a trivial task if compared to other auction designs due to the less amount of information gained by observing the agents [6, 1]. Moreover, despite they have not been widely employed in practical cases, Vickrey auctions represent anyway a well-studied allocation system that is ideal to validate our conjecture. Nonetheless, there are some known important examples of application of this kind of auction in real scenarios, e.g. assignment of oil drilling rights and selling of advertisement spaces on web sites[13]. In our experiments, this problem is posed as a classification task: given a set of strategy labels L and a dataset D of known behavioral models, i.e. a set of Markov chains each labeled with $l \in L$ (the strategy they embed), and a set U of unknown behavioral models, i.e. unlabeled Markov chains, it is required to assign to every element of U the label $k \in L$ that identify the policy it depicts in a Vickrey auction, given the classification of D.

2 Background

In this section, we provide the necessary background notions underneath the method we employed in our experiments.

2.1 Markov chain

Markov chains represent a tool to formally model the evolution of the state of an observable environment affected by a random variable that follows a specific probability distribution. In other words, the state of such system evolves according to a specific probability value. Moreover, a key characteristic of Markov chains is defined as the Markov property: the probability that the current state s of the system transitions to the next state t depends only by s and not by the previous states assumed by the process. The following definitions and theorems, along with their proofs, can be found in [2].

Definition 1 (Markov chain). *Let P be a $k \times k$ matrix with elements $\{P_{ij} : i, j = 1, ..., k\}$. A random process $(X_0, X_1, ...)$ with finite space $S = \{s_1, ..., s_k\}$ is a Markov chain with transition matrix P if for all n, all $i, j \in \{1, ..., k\}$ and all $i_0, ..., i_{n-1} \in \{1, ..., k\}$ we have*

$$\begin{aligned} &\mathbb{P}(X_{n+1} = j | X_0 = i_0, ..., X_{n-1} = i_{n-1}, X_n = i) = \\ &\mathbb{P}(X_{n+1} = j | X_n = i) = P_{ij} \end{aligned} \tag{1}$$

Equation 1 in the above definition formally expresses the Markov property. Although such hypothesis is rarely encountered in real world domains, it is often assumed to hold as an acceptable approximation of the dynamics characterizing the studied process. Any instance of Markov chain is well specified as a tuple $M = (S, P, \mu)$, where μ represents the initial probability distribution defined over the state space S. However, we identify a Markov chain using only its transition matrix P and its relative graph representation.

Theorem 1 (Stationary Distribution). *Given a Markov chain P, the vector π such that $\pi P = \pi$ is the stationary distribution of P.*

The stationary distribution π is the main mathematical result that gives information about the transition probabilities between states whenever the chain execution reaches a stable configuration, that is the probability distribution representing the state of the Markov chain that remains constant from a certain point onward. Hence, it is representative of the long-term execution of the Markov chain, since π will be the probability distribution assumed after n transitions made by the chain, where $n \to \infty$. However, for general Markov chain it is not guaranteed the existence of a stationary distribution and neither its uniqueness. Moreover, even if a Markov chain admits such hypothesis, it is necessary to check whether the initial probability distribution of the model converges to the associated stationary distribution. There are some special cases of Markov chain that guarantee the existence of a single stationary distribution that is reached by any initial probability distribution assigned.

Definition 2 (Irreducible Markov Chain). *A set of states is irreducible if it is possible to go from each state to any other in an arbitrary (finite) number of steps. A Markov chain is irreducible if it consists of a single irreducible set. For any finite, irreducible Markov chain, π is unique.*

Definition 3 (Absorbing Markov Chain). *Given a Markov chain P, a state s_i is absorbing if $P_{ii} = 1$, otherwise it is transient. A Markov chain is absorbing if at least one of its states is absorbing and if from every transient state an absorbing one will be eventually reached.*

If the Markov chain we consider is absorbing, it is possible to divide the components of its state space into transient and absorbing. Such partition can also be reflected in the transition matrix, through its transformation in a canonical form. Such representation entails a block decomposition of the states useful to identify the elements of the chain that will be handled by the procedure that extract the embedded long-term features.

Definition 4 (Canonical form of an absorbing Markov chain). *If an absorbing Markov chain P has n transient states and r absorbing states, its transition matrix can be rewritten as*

$$P = \left[\begin{array}{c|c} Q & R \\ \hline \emptyset & I \end{array}\right]$$

where Q is an $n \times n$ matrix of the transition probability between the transient states, R is a $n \times r$ non-null matrix of the transition probability from the transient to the absorbing states, $\emptyset$ is a $r \times n$ null matrix, and I is a $r \times r$ identity matrix.

Lemma 1. *For any absorbing Markov chain in canonical form we have that $Q^k \to 0$ as $k \to \infty$.*

Lemma 1 states that increasing the number of transitions done on the Markov chain leads to the nullification of the Q term of the transition probabilities related to transient states, since the amount of probability weight will gradually flow into the absorbing states. Such lemma is useful to derive Theorems 2 and 3.

Theorem 2 (Fundamental Matrix of an absorbing Markov Chain). *The fundamental matrix N of an absorbing Markov chain P in canonical form is defined as*

$$N = I + Q^1 + ... + Q^k = \sum_{k=0}^{\infty} Q^k = (I - Q)^{-1}$$

where each entry N_{ij} represents the mean of the total number of times that the chain is in a given transient state s_j if starting from the transient state s_i. The inverse of $(I - Q)$ is guaranteed to exist for every absorbing Markov chain.

Theorem 3 (Transient states probability).

$$H = (N - I)N_{dg}^{-1}$$

Each entry H_{ij} represents the probability of reaching transient state s_j starting from transient state s_i before the process is completely absorbed. N_{dg} is the diagonal of N.

It is worth to point out that the transient states probabilities (Theorem 3) has a different meaning than the stationary distribution (Theorem 1). The method we test in this paper involves both notions in distinct phases. In particular, we rely on the latter to transform each input Markov chain of a given model into an absorbing Markov chain.

3 Enforcing Absorbency

As previously mentioned, we aim to compute the long-term transition probability referred as the probability of going from each state to every other, given an infinite amount of time. In practice, this translates in executing the process described by a Markov chain until a stable configuration is reached (a fixpoint), i.e., until the probability values of moving between states would not change anymore from that point onward. This allows to analyze connections between states discarding what is not important that may lie in between. Looking at Figure 1, if the policy to learn from a teacher agent consists in *S1* $\rightarrow$ *S3* $\rightarrow$ *S5*, all other actions, i.e., *S2, S4, S6, S7*, are noise not related to the behavior the learner agent should learn, making the task more prone to errors. We use the long-term transition probability instead of the stationary distribution, as the latter is not guaranteed to be meaningful in the behavioral models we are given to analyze. Looking at the Markov chain in Figure 1, the stationary distribution (computed using Theorem 1) is $\pi = [0, 0, 0, 0.37, 0.37, 0.26, 0, 0]$, where π_i corresponds to the probability of being state Si. Notice that π is non-zero only for the set of states forming an irreducible Markov chain ($S3, S4, S5$). Hence, we lose the information that all the states where $\pi_i = 0$ can be reached, even though never visited again thereafter. The long-term transition probability instead would tell us that, considering state S_0 for example, the probability values of reaching the other states are $L_{0i} = [0, 0.6, 0.6, 0.6, 0.6, 0.6, 0.4, 0.4]$. Such information for every couple of states is fundamental for the approach. A problem of the explained operation is that, for example, a Depth First Search (DFS) is not efficient since the presence of cycles might require many steps to reach convergence, moreover, the identification of the fixpoint is not trivial, and as many DFS as the number of states (in the worst case) are required. For this reason, we aim to exploit standard properties of Markov chains, namely the transient states probability (Theorem 3), to efficiently compute the long-term transition probability. Nevertheless, generic Markov chains may not hold the absorbency property (Definition 3), which is fundamental to compute the transient states probability. Thus, we exhibit a procedure, proposed in [10], that can transform any generic Markov chain M (Figure 1) into an absorbing Markov chain M' (Figure 2) from which to derive the long-term transition probability for M.

The core part of the transformation for a generic Markov chain M requires to identify all terminal Strongly Connected Components (SCCs) [12], defined as a SCC T that has no outgoing edges, i.e., once the process enters it can not leave the states of T. The peculiar property that makes terminal SCCs important for the procedure is that in the long-term, as the number of steps increases (toward

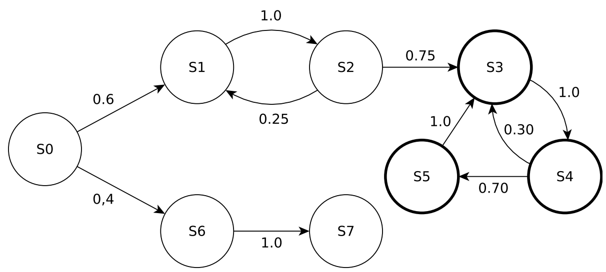

Fig. 1. Markov chain with states in bold (S3, S4, S5) forming a terminal SCC

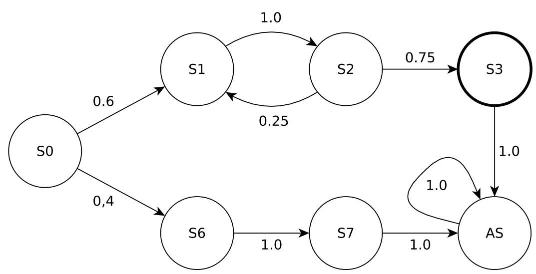

Fig. 2. Absorbing transformation applied to the Markov chain of Figure 1. State S3 is the result of the merge for the terminal SCC (S3, S4, S5)

infinity), the probability of going from each state to every other increases as well, approaching 1. Indeed, since a SCC is an irreducible Markov chain (Definition 2) that can never be left by the process, the more time is given, the more probable is that any state will be reached. Hence, we can merge every terminal SCC into a its corresponding single state s_m without loosing information on the long-term transitions between the states involved since it can be reconstructed by exploiting the knowledge on the long-term behavior for terminal SCCs. Then, the last step to obtain an absorbing Markov chain is to connect with an edge at probability value 1 each state s_m to a newly created absorbing state s_a, obtaining a new absorbing Markov chain M'. In this way, Definition 3 is respected since every state s of M is now transient in M' (possibly merged into a state s_m) and will eventually reach the absorbing state s_a. In fact, either there exist a path $s \rightsquigarrow s_m \to s_a$ or a direct edge $s = s_m \to s_a$. This is true since following any outgoing edge from a state s, a terminal SCC will eventually be reached (and so its s_m merged state) and every s_m is directly connected to the absorbing state s_a. Figure 2 shows an application example of the procedure to the Markov chain in Figure 1. The resulting transition matrix M' in canonical form is visible in Equation 2. Indices for states $S6$ and $S7$ are 4 and 5 respectively in M' as a consequence of merging the terminal SCC. Block matrices Q, R, $\emptyset$ and I are the top-left, top-right, bottom-left, and bottom-right blocks of M' respectively, as of Definition 4.

$$M' = \left[\begin{array}{cccccc|c} 0 & 0.6 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0.75 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \tag{2}$$

Finally, the long-term transition probability between every couple of states (s_i, s_j) for a generic Markov chain M can be computed with Definition 5.

Definition 5 (Long-term transition probability). *Given a Markov chain M, the long-term transition probability value L_{ij} of going from state $s_i \in M$ to state $s_j \in M$ can be computed from the transient states probability H (Theorems 2 and 3) as follows*

$$L_{ij}(H) = \begin{cases} 1 & \textit{if } s_i \textit{ and } s_j \textit{ are in the same terminal SCC in } M \\ 0 & \textit{if } s_i \textit{ is in a terminal SCC } T \textit{ in } M \textit{ and } s_j \notin T \\ H_{im} & \textit{if } s_i \textit{ is not in a terminal SCC in } M \textit{ and } s_j \textit{ was} \\ & \textit{merged into a state } s_m \textit{ in } M' \\ H_{ij} & \textit{otherwise} \end{cases}$$

The first case is a direct application of the behavior in the long-term for couple of states contained within the same terminal SCC. The second case is trivial: states within a terminal SCC can only reach other states of the same SCC. The third case is a consequence of merging terminal SCCs into a single node s_m.

4 Feature Extraction

The approach proposed in [10] is a classification technique, hence a feature extraction process is required. The aim is to recognize given behaviors, and for this reason a "blueprint" model D is required in order to know which are the states and transitions to focus on for the feature extraction process. More specifically, the long-term transition probability values computed for an unknown model x are projected over D to obtain a feature vector. Thus, the blueprint D can be regarded as the "shape" for the selection of the features and needs only to specify states and transition edges (no probability values on the edges).

The application of the complete feature extraction procedure to the model x of Figure 1 with the blueprint D of Figure 3 is reported below. The first step is to enforce the absorbency property, obtaining a result visible in Figure 2 and corresponding to Equation 2. Then, by applying Theorems 2 and 3 to the block matrix Q (top-left block of Equation 2) we obtain the transient states probability matrix H of Equation 3.

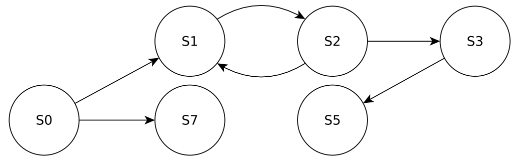

Fig. 3. Example of blueprint D

$$H = \begin{bmatrix} 0 & 0.6 & 0.6 & 0.6 & 0.4 & 0.4 \\ 0 & 0.25 & 1 & 1 & 0 & 0 \\ 0 & 0.25 & 0.25 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3)$$

Finally, the projection of H over the blueprint D making use of Definition 5 results in matrix L of Equation 4. Recall that state $S5$ was merged into $S3$ as effect of the absorbing transformation, hence $L_{i5} = L_{i3}$ for every state s_i, whether states $S4$ and $S6$ are not contained in D (indices 4 and 5 then correspond to states $S5$ and $S7$ respectively in L). Matrix L will then be flattened into the final feature vector.

$$L = \begin{bmatrix} 0 & 0.6 & 0.6 & 0.6 & 0.6 & 0.4 \\ 0 & 0.25 & 1 & 1 & 1 & 0 \\ 0 & 0.25 & 0.25 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4)$$

The exploitation of standard techniques for absorbing Markov chains enable to efficiently extract the long-term transition probability instead of, for example, performing multiple DFSs till a convergence point.

5 Case of study: Vickrey auctions

The intent of our paper is to show that the procedure explained in Sections 3 and 4 can be effectively used to extract the approximated long-term behaviors in any kind of Markov chain model, even on instances that do not represent malicious behavioral profiles. Indeed, the method aim is to give insights about the core behavioral characteristics of a model depicting an agent policy. Such model transformation and the subsequent feature extraction are particularly suitable when we are interested in filtering out from the input model the noise contained therein.

The empirical analysis we performed focuses on iterated Vickrey auction setting[1]. In such scenario, we have a set of bidders, each submitting at each auction stage a bid without knowing the bid value decided by the other agents. Once the auctioneer collects all the players' bids, the agent that made the highest bid wins the item offered in that round and pays a price equivalent to the second best bid [8]. This mechanism is repeated for all the elements offered in the same auction. In our iterated version of Vickrey auction, we assume that we are given the initial state of a second-price sealed-bid auction A that is repeated for a fixed number of times n. Moreover, we provide to shuffle the order of auctioned goods between each iteration of the repeated Vickrey auction. This measure has been chosen to prove the robustness of the classifier trained with long-term probability features. Indeed, if we assume that the items' order is fixed and remain constant through each auction iteration, the models would encode bid patterns easily deductible by any training algorithm; however, if the sequence of offered objects is randomly permuted every time, the identification of such strategic patterns is not trivial. The budgets associated to each player are reset at the start of every iteration. Figures 4 and 5 depict an overview scheme of the experiments we set up. Before explaining how we practically conducted the experiments, it is worth to highlight that, in spite of the presence of the known existence of strategies that constitute equilibria for given players' preferences, we point the attention on the identification of known policies observing the actions performed by an agent under analysis. Hence, we ignore any consideration concerning players' reward in the empirical analysis since it is out of the scope of the technique we test.

In our experiments, we have a set of 10 participant agents to each auction iteration and a passive analyzer[2]. We first select only one player X among the bidders whose behaviors will be tracked by the analyzer in order to construct its model. The analyzer does not account for the actions done by other players. However, while X adopts a different policy in each iterated Vickrey auction, we suppose that the remaining agents follows the same strategy, i.e. try to commit each time a bid that is closest as possible to the real value of the current good[3]. A given initial budget $b = 120$ is associated to every player and we suppose that the analyzer does not know the amount of b, i.e., it is not observable. For this reason, we cannot include such information in the agent behavioral model. Thus, the states of such Markov chain models are labeled with the bids made across the auctions stages (except for the root starting node, always labeled as *Init*), whereas edges are labeled representing object identifiers which drive the bidders to the next auction round. Figure 6 provides an example of the model we adopt.

We design 6 types of bidding strategies for the target X: i) always fair, ii) always aggressive, iii) always prudent, iv) mixed, v) interval, vi) type-based. Strategy i) indicates to make a bid for the current item corresponding to its

[1] Also known as second-price sealed-bid auction

[2] We refer to an agent that does not interact during the auction, i.e. it is limited to observe the bids submitted by each player across every round

[3] Later in this section we call this informally described strategy as "fair"

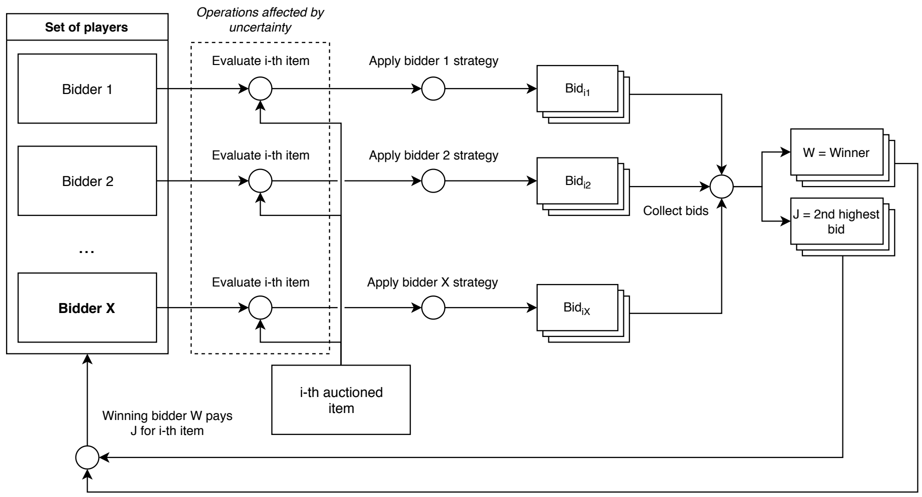

Fig. 4. Description of how it is conducted a single second-price sealed-bid auction round. Repeating the procedure for each element in sale, we obtain the (not iterated) Vickrey auction. Bidder X is observed by the analyzer

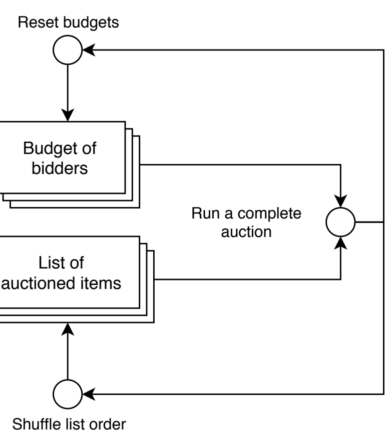

Fig. 5. Overview of the iterated Vickrey auction we designed: once the first Vickrey auction (see figure 4), that is initialized with a fixed budget for players and a given order of the auctioned objects, terminates, the execution proceeds with the next iteration by resetting budgets and randomly changing the items order. Notice that the block *Run a complete auction* implements the process in figure 4

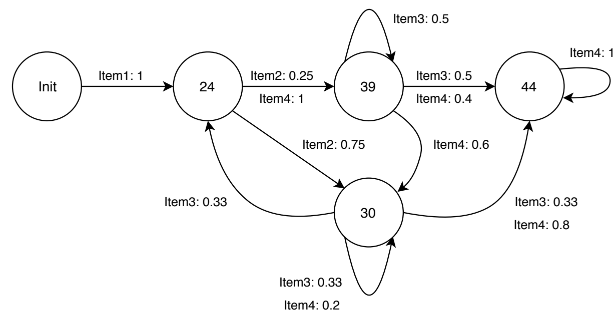

Fig. 6. A simple example model representing the bidding policy of an agent

real evaluation, whereas ii) and iii) respectively play presenting in each stage a bid whose value is always greater and lower than the actual value of the good. To implement such behaviors, ii) and iii) bids are determined by adding and subtracting to v, the real evaluation made for current object, a random value α resulting from a uniform distribution[4]. However, in real auction contexts, it is extremely infrequent for the players to know the exact objective value that auctioned items have: they usually guess with an estimation the objective value related to a good. As a consequence, the bids of the players deviate from the real evaluation due to the uncertainty affecting the players information about the auctioned elements. In order to implement such feature in our Vickrey auction simulations, we provide to add a random bias to the bids resulting from policies i), ii) and iii) that is computed through a gaussian distribution $(0, \sigma)$. Such operation entails the encoding of a degree of noise into the policy bids of players adopting such strategies, which is reflected also in the behavioral models. After a preliminary study, we decided to set $\sigma = 16$ because such value allows to achieve an acceptable tradeoff between making the noise appreciable and preventing the agents to produce unrealistic bids. Moreover, the σ value is reasonable in relation to our empirical set of goods, since we fixed their actual prices in the range $[25, 80]$.

Strategies iv), v) and vi) represent more complex variations that rely on the rationale of the previous. In each auction stage, iv) selects to employ a fair, aggressive or prudent bidding behavior using an arbitrary probability distribution defined over these three policies. Specifically, we tested the strategy iv) with the following probability distribution vectors, each referring to the tuple of behaviors $(fair, prudent, aggressive)$: iv.i) $(0.2, 0.6, 0.2)$; iv.ii) $(0.1, 0.2, 0.7)$; iv.iii) $(0.33, 0.33, 0.34)$. The last mixed case is constructed in order to study whether the agent's behavioral model in case of uniform probability distribution can be distinguished from the other mixed instances.

Similarly, strategy v) decides between fair, aggressive or prudent bids basing on budget thresholds as defined in Equation 5, where parameters $p, q \in \mathbb{R}^+$

[4] In particular, α ranges in interval $[0.05 \cdot v, 0.1 \cdot v]$ in our experiments

and each function $strat_1, strat_2, strat_3$ is chosen among fair, aggressive and prudent strategies. In the empirical analysis, we fixed the following concrete cases for strategy v)[5]: v.i) $aggressive, fair, prudent$ with $p = 80$ and $q = 40$; v.ii) $fair, aggressive, fair$ with $p = 90$ and $q = 30$; v.iii) $prudent, aggressive, prudent$ with $p = 70$ and $q = 50$.

$$bid_i = \begin{cases} strat_1(i, b) & \text{if } b \geq p \\ strat_2(i, b) & \text{if } q \leq b < p \\ strat_3(i, b) & \text{otherwise} \end{cases} \tag{5}$$

Finally, strategy vi) induces the player to formulate bids that are driven by its interest. Specifically, the player has a preference ranking over the object types presented during the auction: whether it is auctioned a good with high preference level the agent uses the aggressive strategy, whilst if such item has medium preference value the agent uses the fair strategy, otherwise the bidder chooses the prudent policy. It is worth to say that we suppose the information about the type of every item as available to the bidders since we deal with rational agents. We designed two opposite type-based strategies in our experiments, defining 4 different kinds of auctioned objects. Hence, we obtained vi.i) with preference vector (t_1, t_2, t_3, t_4) and vi.ii) having reverse ordering (t_4, t_3, t_2, t_1).

Each described strategy has been played 20 times in a repeated Vickrey auction of varying lengths[6], resulting in 220 behavioral models related to the target agent. We remind that the aim of the experiment is to identify the strategies such models embed over the set of the known given 11 strategy, forming the possible labels for the classification. Thus, the classification task is based on a Linear Support Vector Machine (SVM) we trained performing a k-fold cross validation with $k = 5$. Furthermore, the obtained classifier is assessed computing standard learning score measures, i.e. precision, recall and F_1-score. As done in [10], each Markov chain belonging to a model is processed with the algorithm transforming it into an absorbing one and the long-term probability feature vector is then extracted. Once the transforming procedure terminates for every Markov chain of the model, the resulting feature vectors are concatenated to retrieve a single vector accounting for the model. The blueprint D has been created selecting random representatives from each strategy and merging their graphs together. Moreover, in order to highlight the advantages gained using the long-term probabilities as features to train a classifier, we provide to compare the performance achieved by an SVM classifier built through long-term probability features with another SVM trained with the short-term probability features, i.e., employing the 1-step transition likelihoods contained in the transition matrix of original Markov chain models[7]. This last proposal for feature extraction has been taken

[5] For the sake of brevity, the policy functions mentioned in the following are ordered so that i-th strategy specifies $strat_i$

[6] The length parameter ranges from 50 to 600, with an increment equals to 50 in each subsequent experiment w.r.t. the previous

[7] Models obtained from the analysis and not subjected to the absorbing transformation

from [9]. In Table 1 we report the results achieved by the procedure (with an iterated second-price sealed-bid auction of $length = 600$) using long-term probability features, whereas in Table 2 we show learning scores for the short-term probabilities extracted from the native transition matrix.

Table 1. Strategy identification in repeated Vickrey auction - Long-term features

Strategy	Precision	Recall	F_1-score
i) Always Fair	1.00	0.92	0.96
ii) Always Aggressive	1.00	1.00	1.00
iii) Always Prudent	1.00	0.75	0.86
iv.i) $Mixed_{(0.2,0.6,0.2)}$	0.75	1.00	0.86
iv.ii) $Mixed_{(0.1,0.2,0.7)}$	0.86	1.00	0.92
iv.iii) $Mixed_{(0.33,0.33,0.34)}$	1.00	0.50	0.67
v.i) $Interval_{p=80,q=40}$	1.00	1.00	1.00
v.ii) $Interval_{p=90,q=30}$	0.92	1.00	0.96
v.iii) $Interval_{p=70,q=50}$	0.80	1.00	0.89
vi.i) $TypeBased_(t_1, t_2, t_3, t_4)$	1.00	1.00	1.00
vi.ii) $TypeBased_(t_4, t_3, t_2, t_1)$	1.00	1.00	1.00
Overall scores	0.94	0.92	0.92

Table 2. Strategy identification in repeated Vickrey auction - Short-term features

Strategy	Precision	Recall	F_1-score
i) Always Fair	1.00	1.00	1.00
ii) Always Aggressive	1.00	1.00	1.00
iii) Always Prudent	0.86	1.00	0.92
iv.i) $Mixed_{(0.2,0.6,0.2)}$	0.50	1.00	0.67
iv.ii) $Mixed_{(0.1,0.2,0.7)}$	0.00	0.00	0.00
iv.iii) $Mixed_{(0.33,0.33,0.34)}$	1.00	0.42	0.59
v.i) $Interval_{p=80,q=40}$	1.00	1.00	1.00
v.ii) $Interval_{p=90,q=30}$	1.00	1.00	1.00
v.iii) $Interval_{p=70,q=50}$	1.00	0.83	0.91
vi.i) $TypeBased_(t_1, t_2, t_3, t_4)$	1.00	1.00	1.00
vi.ii) $TypeBased_(t_4, t_3, t_2, t_1)$	1.00	1.00	1.00
Overall scores	0.85	0.84	0.83

From Table 1, it is apparent that the procedure built is particularly suitable to accomplish our main goal. The first evident remark concerns the vi) family of policies: vi.i) and vi.ii) are perfectly identified. Our conjecture to explain this outcome is that such couple of policies recalls a player's private evaluation: the choice regarding the bid value is heavily affected by the given preference vec-

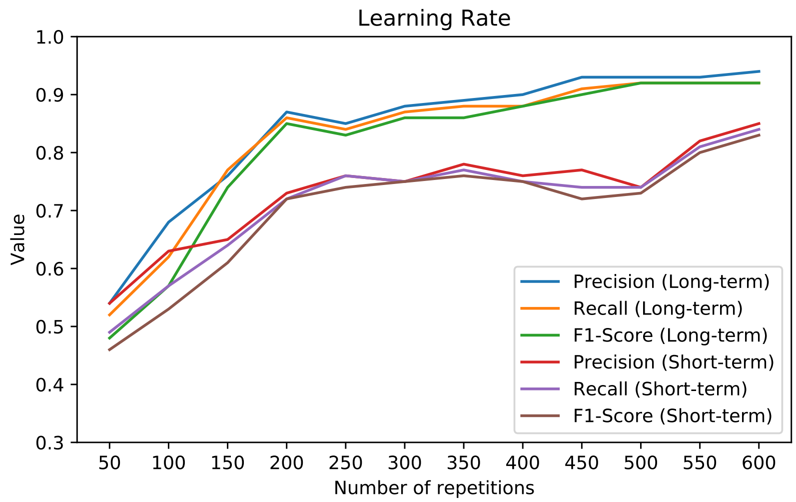

Fig. 7. Learning rate: precision, recall and F_1-score mapped against the number of auction iterations performed

tor, rather than the real price of the considered object. For example, it is not unusual for an agent implementing vi.ii) to overvalue t_4 items and underbid for t_1 elements. This implicit trend provides a behavioral pattern that is fairly detectable by the obtained classifier. Another important observation regards the set of mixed and interval strategies: the overall results show that the use of long-term features allows to construct a classifier resilient to the noise components present in models. Indeed, they represent very sensitive policies to study, whose associated models can become misleading if affected by noisy information. Despite the considerable amount of uncertainty embedded in the inspected player's bids, hence in the states of the Markov chains, due to the approximation performed on the evaluation of goods, the classifier is able to mainly recognize such strategies correctly using the training data. However, it is visible that the SVM still makes some mistakes when it deals with these policies.

The comparison between the two feature selection approaches show a limitation related to the short-term probability as feature: the scores are extremely low for mixed strategies. Indeed, the classifier is unable to actually distinguish them basing on the probability distribution that characterizes them. Such shortcoming burdens significantly on the overall learning values. However, it can be seen that with the other strategies, where the noise affects less the recognition, we obtain measures comparable to long-term ones.

Figure 7 shows the learning rates achieved in our experiments performed with Linear SVM classifier, i.e. how the classification scores (y-axis) change by increasing the number of auction iterations performed per strategy (x-axis). The depicted curves peaks at around 450 auction repetitions and stabilize hereafter for the SVM trained with long-term features. Furthermore, the chart points out that a classifier using the long-term probabilities of Markov chains learns faster than the one employing the short ones. Such evidence confirms our conjecture: whenever we build a Markov chain model affected by noise, long-term features allow to lessen the impact of that such factor exercises in the training phase.

6 Conclusions and Future Works

With this work, we address the problem of identifying known behaviors of intelligent agents within an uncertain system. In particular, we choose a classification approach to solve such issue that is based on the method shown in [10] which builds behavioral models as Markov chains of the analyzed agent and process them in order to retrieve long-term transition probability as feature. Our aim is to demonstrate the general applicability of this analysis to different domains. We evaluate the methodology in the case of iterated Vickrey auctions. Results suggest that such analysis can be extended to any identification problem that involves an analyzer and a target agent, where the former models behaviors of the latter through Markov chains.

There are many research directions that can be considered to improve the achievements of this work. First of all, it would be interesting to test the technique in other kinds of auction, e.g., dutch, english and first-price sealed-bid auction, or to apply it to other fields. Nevertheless, another direction is represented by the application of long-term features to unveil correlations between models of different bidders, in order to check whether there are players that exploit a vulnerability to hack the auction mechanisms, e.g., collusion, cartels [7, 3]. Furthermore, it would be useful to study whether there are formalisms other than Markov chains that allows to retrieve long-term stable model properties and to compare the performance with our proposal.

References

1. Gerding, E.H., Dash, R.K., Byde, A., Jennings, N.R.: Optimal strategies for bidding agents participating in simultaneous vickrey auctions with perfect substitutes. J. Artif. Int. Res. **32**(1), 939–982 (Aug 2008), http://dl.acm.org/citation.cfm?id=1622673.1622697
2. Kemeny, J., Snell, J.: Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix". Undergraduate Texts in Mathematics, Springer New York (1983), https://books.google.com.sg/books?id=0bTK5uWzbYwC
3. M. Ausubel, L., Milgrom, P.: The lovely but lonely vickrey auction. Comb. Auct. **17** (01 2006). https://doi.org/10.7551/mitpress/9780262033428.003.0002
4. Mariconti, E., Onwuzurike, L., Andriotis, P., Cristofaro, E.D., Ross, G.J., Stringhini, G.: Mamadroid: Detecting android malware by building markov chains of behavioral models. In: NDSS. The Internet Society (2017)
5. Mayil, V.: Web navigation path pattern prediction using first order markov model and depth first evaluation. International Journal of Computer Applications **45**(16), 26–31 (May 2012)

6. Parkes, D.C., Ungar, L.H.: Iterative combinatorial auctions: Theory and practice. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence. pp. 74–81. AAAI Press (2000), http://dl.acm.org/citation.cfm?id=647288.721579
7. Sandholm, T.: Limitations of the vickrey auction in computational multiagent systems. In: In Proceedings of the Second International Conference on Multiagent Systems (ICMAS-96. pp. 299–306. AAAI Press (1996)
8. Sandholm, T.: Issues in computational vickrey auctions. Int. J. Electron. Commerce **4**(3), 107–129 (Mar 2000). https://doi.org/10.1080/10864415.2000.11518374, http://dx.doi.org/10.1080/10864415.2000.11518374
9. Sartea, R., Farinelli, A.: A monte carlo tree search approach to active malware analysis. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 3831–3837 (2017). https://doi.org/10.24963/ijcai.2017/535, https://doi.org/10.24963/ijcai.2017/535
10. Sartea, R., Farinelli, A., Murari, M.: Agent behavioral analysis based on absorbing markov chains. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems. Accepted For Publication, Richland, SC (2019)
11. Sarukkai, R.R.: Link prediction and path analysis using markov chains. In: Proceedings of the 9th International World Wide Web Conference on Computer Networks : The International Journal of Computer and Telecommunications Netowrking. pp. 377–386. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands (2000), http://dl.acm.org/citation.cfm?id=347319.346322
12. Tarjan, R.: Depth-first search and linear graph algorithms. In: 12th Annual Symposium on Switching and Automata Theory (swat 1971). pp. 114–121 (Oct 1971). https://doi.org/10.1109/SWAT.1971.10
13. Varian, H.R.: Position auctions. International Journal of Industrial Organization **25**(6), 1163 – 1178 (2007). https://doi.org/https://doi.org/10.1016/j.ijindorg.2006.10.002, http://www.sciencedirect.com/science/article/pii/S0167718706001317
14. Whittaker, J.A., Thomason, M.G.: A markov chain model for statistical software testing. IEEE Trans. Softw. Eng. **20**(10), 812–824 (Oct 1994). https://doi.org/10.1109/32.328991, http://dx.doi.org/10.1109/32.328991
15. Zhu, J., Hong, J., Hughes, J.G.: Using markov chains for link prediction in adaptive web sites. In: Proceedings of the First International Conference on Computing in an Imperfect World. pp. 60–73. Soft-Ware 2002, Springer-Verlag, Berlin, Heidelberg (2002), http://dl.acm.org/citation.cfm?id=645974.758446