Mining Balanced Sequential Patterns in RTS Games 1

Guillaume Bosc, Mehdi Kaytoue, Chedy Raïssi, Jean-François Boulicaut, Philip Tan

ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-975

Mining Balanced Sequential Patterns in RTS Games[1]

Guillaume Bosc[2] and **Mehdi Kaytoue**[2,3] and **Chedy Raïssi**[4] and **Jean-François Boulicaut**[1] and **Philip Tan**[5]

Abstract. The video game industry has grown enormously over the last twenty years, bringing new challenges to the artificial intelligence and data analysis communities. We tackle here the problem of automatic discovery of strategies in real-time strategy games through pattern mining. Such patterns are the basic units for many tasks such as automated agent design, but also to build tools for the professionally played video games in the electronic sports scene. Our formalization relies on a sequential pattern mining approach and a novel measure, the balance measure, telling how a strategy is likely to win. We experiment our methodology on a real-time strategy game that is professionally played in the electronic sport community.

1 Introduction

Real-time strategy games consist in long and noisy traces of actions made by two opponents in interaction out of which one will win the game. Our goal is to provide the basic pattern mining tools to exhibit characteristics such as weaknesses and strengths of players' strategies. More precisely, the question is how to automatically discover patterns of strategies, and provide for each a metric that gives a notion of balance. The sequential pattern mining framework [7] partially answers the question [2, 4], as it allows to automatically discover frequent sequences of actions from a large corpus of game traces. We propose to introduce the balance measure in this framework. The intuition is the following. Consider a game (s_1, s_2) where s_i is the sequence of actions made by a player i: the game is well-balanced if playing s against s' allows to win as much as it allows to lose. Finding such sequential patterns is a challenging task as current state-of-the-art pattern mining methods fail to handle efficiently this problem [2]. From an application point of view, sequential patterns from zero-sum games allow analysts to derive statistics on the success of a player with different tactics for real-time strategy games. These video games are becoming *de facto* sensations in organized competitions, especially between professional competitors taking part in international tournaments, with prize-pools in millions of US$, surrounded by managers, teams and sponsors [8] and followed by a large world-wide fan base [5]. As such, balancing the games is one of the most important issues of any E-Sport to ensure a fair competitive aspect on which depends a complete economic model. Strategic patterns and their balance measure can also be used for designing AI agents, tutorials for the beginners, as well as tools for professionals in E-Sport for commentating and preparing competitions.

[1] This work was partially supported by the FAPEMIG/INRIA project DMKM. An extended version, data and algorithms can be found in [2, 1].
[2] Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France, email: firstname.name@insa-lyon.fr
[3] Universidade Federal de Minas Gerais, Brazil, email:kaytoue@dcc.ufmg.br
[4] INRIA Nancy Grand Est, France, email: chedy.raissi@inria.fr
[5] MIT Game Lab, Cambridge, MA 02139, USA, email: philip@mit.edu

2 Mining Sequential Patterns

Let $\mathcal{I}$ be a finite set of *items*. Any non-empty subset $X \subseteq \mathcal{I}$ is called an *itemset*. A *sequence* $s = \langle X_1, ..., X_l \rangle$ is an ordered list of $l > 0$ itemsets. Consider $\mathcal{I}$ as a set of events (actions), an itemset denotes simultaneous events while the order between two itemsets a strict preceding relation. A sequence database $\mathcal{D}$ is a set of sequences.

Definition (subsequence, support and frequency). A sequence $s = \langle X_1, ..., X_{l_s} \rangle$ is a *subsequence* of a sequence $s' = \langle X'_1, ..., X'_{l'_s} \rangle$, denoted $s \sqsubseteq s'$, if there exists $1 \leq j_1 < j_2 < ... < j_{l_s} \leq l'_s$ such that $X_1 \subseteq X'_{j_1}, X_2 \subseteq X'_j, ..., X_{l_s} \subseteq X'_{j_{l_s}}$. Let $\mathcal{D}$ be a sequence database. The *support* of a sequence s is given by $support(s, \mathcal{D}) = \{s' \mid s \sqsubseteq s',\ s' \in \mathcal{D}\}$, and the frequency by $freq(s, \mathcal{D}) = |support(s, \mathcal{D})|/|\mathcal{D}|$.

Given a minimal frequency threshold $0 < \sigma \leq 1$, the *frequent sequential pattern mining problem* consists in finding all sequences s such as $freq(s, \mathcal{D}) \geq \sigma$. This can be achieved with several algorithms such as PrefixSpan [7].

Example. Let $\mathcal{D} = \{s_1, s_2, s_3, s_4\}$ with $\mathcal{I} = \{a,\ b,\ c,\ d,\ e,\ f,\ g\}$ be a sequence database given in the table below. To ease reading, we drop the commas, but also braces for singletons. For a given sequence $s = \langle abc \rangle$, we have $s \sqsubseteq s_1$, $s \sqsubseteq s_4$, $s \not\sqsubseteq s_2$, $s \not\sqsubseteq s_3$. With $\sigma = \frac{3}{4}$, $\langle acc \rangle$ is frequent, $\langle a\{bc\}a \rangle$ is not.

id	$s \in \mathcal{D}$
s_1	$\langle a\{abc\}\{ac\}d\{cf\} \rangle$
s_2	$\langle \{ad\}c\{bc\}\{ae\} \rangle$
s_3	$\langle \{ef\}\{ab\}\{df\}cb \rangle$
s_4	$\langle eg\{af\}cbc \rangle$

3 Mining Balanced Sequential Patterns

We consider a RTS game interaction as a sequence of actions made by two players where exactly one player wins (no ties), typically a zero-sum game. Such a sequential game can be represented as a sequence of sets of actions, each action realized by one of the two players. When both players play in real-time, one can describe these interactions as sequences of itemsets. An itemset will describe game action events that may occur so fast that they seem to be simultaneous over a given time interval.

Definition (Interaction (sequence) database). Given a set of players $Players$ and a set of actions $Actions$, a sequence database $\mathcal{D}_{interaction}$ is called an *interaction database*. Each sequence denotes an interaction between two players and is defined over the set of items $\mathcal{I} = Actions \times Players$. A mapping $class : \mathcal{D}_{interaction} \rightarrow Players$ gives the winner of each interaction. For example, the sequence $s_1 = \langle (a, p_1)\{(b, p_1)(c, p_1)\ (c, p_2)\} \rangle$ with $class(s_1) = p_1$ can be interpreted as: "*Player p_1 did action a, then he did b and c while player p_2 did c, and finally p_1 wins*".

The problem here is to find sequences of actions of both interacting players (supposing that those actions are mutually dependent) as generalizations that appear frequently, and to be able to characterize their discriminating ability for winning/losing through a so-called *balance measure*. In classical sequential pattern mining settings, the goal would be to find frequent sequences (strategies) and a so-called growth-rate like measure (the balance) [6]. We show however in [2] that the existing emerging pattern framework [6] can only be used in very particular cases, and is much less computationally efficient that our new method. Our method relies on the notion of *signed interactions* introduced below. We show how sequential patterns and their balance can be naturally defined with this new representation.

Definition (Signed interaction database). Recall that $Actions$ is a finite set of *actions* shared by both players. We introduce $\mathcal{I}_s = Actions \times \{+,-\}$ denoting actions associated either to a positive or negative class. A signed database $\mathcal{D}_s$ is built from an interaction database $\mathcal{D}_{interaction}$ as follows: each action of an interaction sequence is signed $+$ if it is performed by the winner and signed $-$ if performed by the loser (both players and class labels are dropped).

Definition (Dual of an item, itemset and sequence). Let $\mathcal{I}_s = Actions \times \{+,-\}$ be the set of signed items, or actions. For any $(a,c) \in \mathcal{I}_s$, also written a^c, we define its *dual* as $dual(a,c) = dual(a^c) = (a, \{+,-\} \backslash c) = a^{\{+,-\} \backslash c}$. Informally, it means that the dual of a signed action is the same action where the sign c has changed. This definition is simply propagated for itemsets and sequences of itemsets, for any $X \subseteq \mathcal{I}_s$ and any $s = \langle X_1, X_2, ... \rangle$ a sequence over $\mathcal{I}_s$: $dual(X) = \{dual(x), \forall x \in X\}$ and $dual(s) = \langle dual(X_1), dual(X_2), ... \rangle$. For example, we have $dual(\langle a^+ \{b^+ c^-\} \rangle) = \langle a^- \{b^- c^+\} \rangle$. These definitions allow us to introduce a balance measure gives, for a sequential pattern s the proportion of its support among the support of both itself and its dual.

Definition (Balance). For a pattern s, the balance measure is given by $balance(s) = |support(s, \mathcal{D}_s) \backslash support(dual(s), \mathcal{D}_s)|$ / $|supportD(s, \mathcal{D}_s) \sqcup support(dual(s), \mathcal{D}_s)|$ where $\sqcup$ denotes the exclusive union $A \sqcup B = (A \cup B) \backslash (A \cap B)$. We have that $balance(s) \in [0,1]$ and $balance(s) + balance(dual(s)) = 1$.

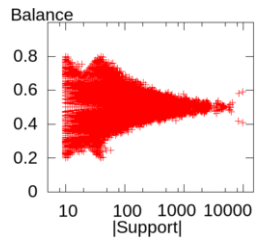
Mining balanced patterns from signed interactions. Let $\mathcal{D}_s$ be a signed interaction database defined over $\mathcal{I}_s$ generated from an interaction database $\mathcal{D}_{interaction}$, and σ a minimum frequency threshold. The problem is to extract the set of so-called *frequent balanced patterns* $\mathcal{F}_s$ such as for $s \in \mathcal{F}_s$, $freq_{\mathcal{D}_s}(s) \geq \sigma$, $freq_{\mathcal{D}_s}(dual(s)) \geq \sigma$ and the balance measure is computed. We propose several algorithms to answer this problem in [2].

4 An Application in Electronic Sport

We study one of the most competitive real-time strategy games (RTS) [5], StarCraft II (Blizzard Entertainment, 2010) which has its own world-wide players ranking system (ELO) and annual world cup competition series (WCS) with a US$1.6 millions prize pool for the year 2014. StarCraft II is a real time strategy game which involves two players each choosing a faction among Zerg (Z), Protoss (P) and Terran (T): there are 6 different possible match-ups with different strategies of game. In a game, two players are battling on a map, controlling buildings and units to gather supply, build an army with the goal of winning by destroying the opponent's forces.

Datasets. We collected $91,503$ pro/high-level games of StarCraft II with a total of 3.19 years of game time. We divided the set of games into six different sequence datasets, one for every match ups (since there are 3 factions). Buildings are one of the key elements of a strategy, since they allow different kinds of units production: from each game, we derive a sequence where the items represent the buildings the players chose to produce in real time, and itemsets denote time windows of 30 seconds. We consider only the 10 first minutes of each game: after that, buildings importance fades away and does not translate anymore into strategy blocks.

Discovered patterns. We consider here the games involving the factions Zerg for the both players (see [2] for more results). It is interesting to visualize the distribution of both the support and the balance of the patterns. The figure on the right gives such distribution for dataset ZvZ that allows very fast computations with low σ (less than 5 seconds for $\sigma = 0.001$). There, both a pattern and its dual are presented, which allows interestingly to observe that $y = 0.5$ gives almost a symmetry axis. Indeed, both a pattern and its dual do not necessarily have the same support. Empirically there are high chances for a pattern with high frequency to have a fair balance around 0.5.



We observed the collection of resulting patterns with a game expert. Firstly, the distribution Support/Balance is an expected result: the game in its current state is globally balanced. Empirically again, the less frequent a pattern the more chances it is a long sequence (limited by space we do not present pattern size distribution: in average the size is 10 with standard deviation of 3.2). This makes also sense, since the goal of any competitive strategy game is that at the initial state of the game balance should be exactly 0.5 and the longer the game the easiest it is to break the symmetry for any of the two players depending of their skills and the choices made.

There are $40,674$ patterns for ZvZ with $\sigma = 0.001$ involving two players. We restricted the set of patterns to those involving two specific items (RoachWarren and Spire) to get only 290 patterns. $\langle SpawPool^+, SpawnPool^-, SpiCrawler^+, RoachWarren^+, Spire^- \rangle$ denotes for example games where one of the players uses exclusively air units in his army while the second relies solely on ground units. These are 2 different known openings with an overall balance of 0.47 and a support of 68. The resulting set of patterns of any extraction is hardly human readable. However, it can be plugged and queried for prediction purposes through an application during a game. This specific application is highly demanded by broadcasters of E-Sport competitions to better interact with their audience [3, 8, 5].

REFERENCES

[1] BalanceSpan. http://liris.cnrs.fr/mehdi.kaytoue/balancespan.html.
[2] Guillaume Bosc, Mehdi Kaytoue, Chedy Rassi, and Jean-Francois Boulicaut. Mining Balanced Sequential Patterns in Real-Time Strategy Games, Research Report RR-LIRIS-2014-007, LIRIS UMR 5205 CNRS, May 2014.
[3] Gifford Cheung and Jeff Huang, 'Starcraft from the stands: understanding the game spectator', in *SIG CHI '11*, pp. 763–772. ACM, (2011).
[4] Milton Garca-Borroto, JosFco. Martnez-Trinidad, and JessAriel Carrasco-Ochoa, 'A survey of emerging patterns for supervised classification', *Artificial Intelligence Review*, 1–17, (2012).
[5] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr., and Chedy Raïssi, 'Watch me playing, i am a professional: a first study on video game live streaming', in *Proc of MSND@WWW'12*. ACM, (2012).
[6] Petra K. Novak, Nada Lavrač, and Geoffrey I. Webb, 'Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining', *J. Mach. Learn. Res.*, **10**, 377–403, (2009).
[7] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu, 'Prefixspan: Mining sequential patterns by prefix-projected growth', in *ICDE*, pp. 215–224. IEEE Computer Society, (2001).
[8] T. L. Taylor, *Raising the Stakes : E-Sports and the Professionalization of Computer Gaming*, MIT Press, 2012.