A Markov Game Model for Evaluating National Hockey League Play-By-Play Events

Abstract

A variety of advanced statistics are used to evaluate player actions in the National Hockey League, but they fail to account for the context in which an action occurs or the long-term effects of an action. We present a novel approach to construct a multi-agent Markov Decision Process, called a Markov Game model, from sequences of player actions that accounts for action context and cross-sequence influence, and supports reinforcement learning over a variety of objective functions to evaluate actions and players. A dynamic programming value iteration algorithm is used to learn the values of states in the Markov Game model and compute the impact of player actions. Player actions are found to have both positive and negative effects dependent on the context the action occurs in. Players are ranked according to the aggregate impact of their actions. We examine values of context states with and without event sequences to observe the benefit of including event histories.

1 INTRODUCTION

A fundamental goal of sports statistics is to understand which actions contribute to winning in what situation. As sports have entered the world of big data, there is increasing opportunity for large-scale machine learning to model complex sports dynamics. The research described in this paper applies AI techniques to model the dynamics of ice hockey; specifically the Markov Game model formalism [Littman, 1994], and related computational techniques such as the dynamic programming value iteration algorithm. We make use of an extensive dataset about matches in the National Hockey League (NHL). This dataset comprises all play-by-play events from 2007 to 2014, for a total of over 2.8M events/actions and almost 600K play sequences. The Markov game model comprises over 1.3M states. Whereas most previous work on Markov Game models aim to compute optimal strategies or policies [Littman, 1994] (i.e., minimax or equilibrium strategies), we learn a model of how hockey is actually played, and do not aim to compute optimal strategies. In reinforcement learning (RL) terminology, we use dynamic programming to compute a Q-function in the *policy-on* setting [Sutton and Barto, 1998].

Motivation Motivation for learning a Q-function for the NHL hockey dynamics includes the following.

Knowledge Discovery. The Markov Game model provides information about the likely consequences of actions. The basic model and algorithms can easily be adapted to study different outcomes of interest. For example, with goals as rewards, a Q-function specifies the impact of an action on future goals. With penalties as costs in the same model, the resulting Q-function specifies the impact of an action on future penalties.

Player Evaluation. One of the main tasks for sports statistics is evaluating the performance of players [Schumaker et al., 2010]. A common approach is to assign action values, and add up the corresponding values each time a player takes the respective action. Action value functions used in this way are often referred to as Sabermetrics. An advantage of this additive approach is that it provides highly interpretable player rankings. A simple and widely used example in ice hockey is the +/- score: for each goal scored by (against) a player's team when he is on the ice, add +1 (-1) point. Researchers have developed several extensions of +/- for hockey [Macdonald, 2011; Spagnola, 2013; Schuckers and Curro, 2013].

There are two major problems with the current action count approaches. First, they are unaware of the *context* of actions within a game. For example, a goal is more valuable in a tied-game situation close to the end of the match than earlier in the match, or when the scorer's team is already four goals ahead. Another example is that if a team manages two successive shots on goal, the second attempt typically has a higher chance of success. In the Markov Game model, *context = state.* Formally, the Q function depends

both on the state s and the action a. Richer state spaces therefore capture more of the context of an action. The second problem is previous action scores are based on immediate positive consequences of an action (e.g. goals following a shot). However, an action may have medium-term and/or ripple effects rather than immediate consequences in terms of visible rewards like goals. Therefore evaluating the impact of an action requires *lookahead*. Propagating action effects across sequences utilizes the ordering play sequences in a game, rather than treating sequences as an unordered independent set. Long-term lookahead is especially important in ice hockey because evident rewards like goals occur infrequently [Lock and Schuckers, 2009]. For example, if a player receives a penalty, this leads to a manpower disadvantage for his team, known as a power play for the other team. It is easier to score a goal during a power play, but this does not mean that a goal will be scored immediately after the penalty. For another example, if a team loses the puck in their offensive zone, the resulting counterattack by the other team may lead to a goal eventually but not immediately. The dynamic programming value iteration algorithm of Markov Decision Processes provides a computationally efficient way to perform unbounded lookahead. We leverage this computational capability to propagate the impact of an action on reward events like goals and penalties that may occur after the action was taken. These effects are propagated along a sequence and even across sequences.

Evaluation Our evaluation learns Q-functions for two reward events, scoring the next goal and receiving the next penalty. We observe a wide variance of the impact of actions with respect to states, showing context makes a substantial difference. We provide examples of the context dependence to give a qualitative sense of how the Markov Game model accounts for context. To evaluate player performance, we use the Q-function to quantify the value of a player's action in a context. The action values are then aggregated over games and seasons to get player values. The resulting player rankings correlate with plausible alternative scores, such as a player's total points, and even improve on these measures, as our impact score is based on many more events.

Contributions Our main contributions may be summarized as follows:

1. The first Markov Game model for a large ice hockey state space (over 1.3M), based on entire play data.
2. Learning a Q-function that models play dynamics in the National Hockey League from a massive data set (2.8M events). We introduce a variant of AD-Trees as a data structure to (1) compute and store the large number of sufficient statistics required [Moore and Lee, 1998], and (2) support value iteration updates.
3. Applying the Q-function to define a context-aware measure of the value of an action, over configurable objective functions (rewards).
4. Applying the context-aware action value measure to score player contributions, including how players affect penalties as well as goals. This is a novel AI-based alternative to existing player scoring methods such as the +/- score or sabermetrics.

Paper Organization. We review related work in measuring player contributions and machine learning in sports in Section 2. We will then give some background information on the ice hockey domain and NHL play-by-play sequences data. Our Markov Game model translates the hockey domain features into the Markov formalism. We then discuss how we implement scalable value iteration for the ice hockey domain. The evaluation section addresses the impact of context and lookahead, the two main advantages of the Markov model. We apply the model to rank the aggregate performance of players and describe the resulting player ranking. We view our work as taking the first, not the last step, in applying AI modelling techniques to ice hockey. Therefore we conclude with a number of potential extensions and open problems for future work.

2 RELATED WORK

Markov Process Models for Ice Hockey A number of Markov process models have been developed for ice hockey [Thomas et al., 2013; Buttrey et al., 2011]. The main difference to our work is these models do not include actions, and hence cannot model the impact of actions. Another difference is previous models focus on goals and use a Poisson model to estimate scoring rates in continuous time. We use a discrete-time model as is common in machine learning, and leave a continuous time hockey Markov model for future work.

Markov Decision Process Models for Other Sports MDP-type models have been applied in a number of sports settings, such as baseball [Sidhu and Caffo, 2014], soccer [Hirotsu et al., 2002], and video games (e.g., [Churchill and Buro, 2013]). The goal of these models is to find an optimal policy for a critical situation in a sport or game. In contrast, we learn in the on-policy setting whose aim is to model hockey dynamics as it is actually played. After we present our results, we discuss how our approach can be extended for purpose of improving a policy, for example to provide advice to coaches. Our work is similar in that our method uses value iteration on a Markovian state space, however, previous Markov models in sports use a much smaller state space. For example, the baseball model of [Sidhu and Caffo, 2014] utilizes only 12 states compared to the $1,325,809$ states in our model.

Evaluating Actions and Players Several papers aim to improve the basic +/- score with statistical techniques [Macdonald, 2011; Gramacy et al., 2013; Spagnola, 2013]. A common approach is to use regression techniques where an indicator variable for each player is used as a regressor for a goal-related quantity (e.g., log-odds of a goal for the player's team vs. the opposing team). The regression weight measures the extent to which the presence of a player contributes to goals for his team or prevents goals for the other team. These approaches look at only goals, no other actions. The only context they take into account is which players are on the ice when a goal is scored.

The closest predecessor to our work is the Total Hockey Rating (THoR) [Schuckers and Curro, 2013]. This assigns a value to all actions, not only goals. Actions were evaluated based on whether or not a goal occurred in the following 20 seconds after an action. This used data from the $2006/2007$ NHL season only. THoR assumes a fixed value for every action and does not account for the context in which an action takes place. Furthermore, the window of 20 seconds restricts the lookahead value of each action. Our Q-learning method is not restricted to any particular time window, but takes into account the event history and looks ahead to the next goal or penalty.

3 DOMAIN DESCRIPTION: HOCKEY RULES AND HOCKEY DATA

We outline the rules of hockey and describe the dataset available from the NHL.

3.1 HOCKEY RULES

We describe a Markov Game Model for ice hockey. To motivate our model, we give a brief overview of rules of play in the NHL. For detailed rules of play in the NHL, refer to [National Hockey League, 2014]. NHL games consist of three periods, each 20 minutes in duration. A team will try to score more goals than their opponent within three periods in order to win the game. If the game is still tied after three periods, the teams will enter a fourth overtime period, where the first team to score a goal wins the game. If the game is still tied after overtime during the regular season, a shootout will commence. During the playoffs, overtime periods are repeated until a team scores a goal to win the game. Teams have five skaters and one goalie on the ice during even strength situations. Penalties result in a player sitting in the penalty box for two, four, or five minutes and the penalized team will be shorthanded, creating a manpower differential between the two teams.

3.2 DATA FORMAT

The NHL provides information about sequences of play-by-play events, which are scraped from `http://www.nhl.com` and stored in a relational database. The real-world dataset is formed from $2,827,467$ play-by-play events recorded by the NHL for the complete 2007-2014 seasons, regular season and playoff games, and the first 512 games of the 2014-2015 regular season. A breakdown of this dataset is shown in Table 1. The type of events recorded by the NHL from the 2007-2008 regular season and onwards are listed in Table 2. There are two types of events: actions performed by players and start and end markers for each play sequence. Every event is marked with a continuous timestamp, and every action is also marked with a zone Z and which team, Home or Away, carries out the action.

Table 1: Size of Dataset

Number of Teams	32
Number of Players	1,951
Number of Games	9,220
Number of Sequences	590,924
Number of Events	2,827,467

Table 2: NHL Play-By-Play Events Recorded

Action Event	**Start/End Event**
Faceoff	Period Start
Shot	Period End
Missed Shot	Early Intermission Start
Blocked Shot	Penalty
Takeaway	Stoppage
Giveaway	Shootout Completed
Hit	Game End
Goal	Game Off
	Early Intermission End

4 MARKOV GAMES

In its general form, a Markov Game [Littman, 1994], sometimes called a stochastic game, is defined by a set of states, S, and a collection of action sets, one for each agent in the environment. State transitions are controlled by the current state and one action from each agent. For each agent, there is an associated reward function mapping a state transition to a reward. Our Markov Game model fills in this scheme as follows. There are two players, the Home Team H and the Away Team A. The game is zero-sum, meaning whenever a home team receives a reward, the Away Team receives minus the reward. Therefore we can simply use a single reward value, where positive numbers denote a reward for the home team (the maximizer), and negative number a reward for the Away Team (the minimizer). In each state, only one team performs an action, although not in a turn-based sequence. This reflect the way the

NHL records actions. This is a special case of a Markov Game where at each state exactly one player chooses No-operation. Our Markov Game model for ice-hockey is a semi-episodic model [Sutton and Barto, 1998] where play moves from episode to episode, and information from past episodes is recorded as state features. The past information includes the goal score and manpower. A sequence in the NHL data corresponds to an episode in Markov decision process terminology. *Within* each episode/sequence, our game model is essentially a game tree with perfect information as used in AI game research [Russell and Norvig, 2010]. We introduce the following generic notation for all states. MDP notation follows [Russell and Norvig, 2010], and a modification of the notation used by [Littman, 1994] is used to describe the multi-agent setup specific to NHL games. Notation for value iteration follows [Mitchell, 1997].

- $Occ(s)$ is the number of occurrences of state s as observed in the play-by-play data.
- $Occ(s, s')$ is the number of occurrences of state s being immediately followed by state s' as observed in the play-by-play data. (s, s') forms an edge in the transition graph of the Markov Game model.
- The transition probability function TP is a mapping of $S \times S \rightarrow (0, 1]$. We estimate it using the observed transition frequency $\frac{Occ(s,s')}{Occ(s)}$.

We begin by defining context features, then play sequences.

5 STATE SPACE: CONTEXT FEATURES

Previous work on Markov process models for ice hockey [Thomas et al., 2013] defined states in terms of hand-selected features that are intuitively relevant for the game dynamics, such as the goal differential and penalties. We refer to such features as **context features**. Context features remain the same throughout each play sequence.

5.1 CONTEXT FEATURES

A **context state** lists the values of relevant features at a point in the game. These features are shown in Table 3, together with the range of integer values observed.

Table 3: Context Features

Notation	Name	Range
GD	Goal Differential	[-8,+8]
MD	Manpower Differential	[-3,3]
P	Period	[1,7]

Goal Differential GD is calculated as Number of Home Goals - Number of Away Goals. A positive (negative) goal differential means the home team is leading (trailing). Manpower Differential MD is calculated as Number of Home Skaters on Ice - Number of Away Skaters on Ice. A positive manpower differential typically means the home team is on the powerplay (away team is penalized), and a negative manpower differential typically means the home team is shorthanded (away team is on the powerplay). Period P represents the current period number the play sequence occurs in, typically ranging in value from 1 to 5. Periods 1 to 3 are the regular play of an ice hockey game, and periods 4 and onwards are for overtime and shootout periods as needed.

Potentially, there are $(17 \times 7 \times 7) = 833$ context states. In our NHL dataset, 450 context states occur at least once. The data are for the complete 2007-2014 seasons, as well as the first 512 games of the 2014-2015 season, and includes both regular season and playoff games. Table 4 includes statistics for the top-20 context states over all $590,924$ play sequences, and lists $52,793$ total goals and $89,612$ total penalties. Positive differences are for the home team and negative differences are for the away team. For example, a Goal Difference of 7.1% means the home team is 7.1% more likely to score a goal in that context state than the away team. Similarly, a Penalty Difference of -33.2% means the away team is 33.2% more likely to receive a penalty in that context state than the home team.

A number of previous papers on hockey dynamics have considered the context features of play sequences. The important trends that it is possible to glean from statistics such as those shown in Table 4 have been discussed in several papers. Our data analysis confirms these observations on a larger dataset than previously used. Notable findings include the following.

5.2 DISCUSSION

Some notable findings from Table 4 are outlined below.

1. Home team advantage: the same advantages in terms of context features translate into higher scoring rates.
2. Goals are more frequent than penalties only in the 4th period (cf. [Schuckers and Brozowski, 2012]).
3. Gaining a powerplay substantially increases the probability of scoring a goal [Thomas et al., 2013].
4. Short-handed goals are surprisingly likely, in that a manpower advantage translates only into a goal scoring difference of at most 64.8 %. (Powerplay home team in period 1.)
5. Gaining a powerplay also significantly increases the conditional probability of receiving a penalty [Schuckers and Brozowski, 2012].

Table 4: Statistics for Top-20 Most Frequent Context States

Goal Differential	Manpower Differential	Period	Number of Sequences	Number of Goals	Goal Difference	Number of Penalties	Penalty Difference
0	0	1	78,118	5,524	7.1%	11,398	-2.3%
0	0	2	38,315	2,935	7.6%	5,968	-2.9%
0	0	3	30,142	2,050	5.9%	3,149	-2.2%
1	0	2	29,662	2,329	2.0%	4,749	2.2%
1	0	3	25,780	2,076	4.3%	3,025	3.5%
-1	0	2	25,498	1,970	8.6%	4,044	-8.7%
1	0	1	24,721	1,656	5.3%	4,061	3.4%
-1	0	3	22,535	1,751	0.7%	2,565	-18.3%
-1	0	1	20,813	1,444	4.6%	3,352	-8.1%
2	0	3	17,551	1,459	6.9%	2,286	-0.9%
2	0	2	15,419	1,217	2.7%	2,620	2.9%
-2	0	3	13,834	1,077	-2.3%	1,686	-12.6%
0	1	1	12,435	1,442	64.8%	2,006	65.9%
-2	0	2	11,799	882	3.9%	1,927	-15.7%
0	-1	1	11,717	1,260	-54.8%	2,177	-44.7%
3	0	3	10,819	678	0.3%	1,859	1.2%
-3	0	3	7,569	469	7.0%	1,184	-6.3%
0	1	2	7,480	851	57.0%	1,157	25.7%
0	0	4	7,024	721	5.7%	535	-10.7%
0	-1	2	6,853	791	-52.5%	1,160	-37.4%

While such patterns provide interesting and useful insights into hockey dynamics, they do not consider action events. This means that analysis at the sequence level does not consider the internal dynamics within each sequence, and that it is not suitable for evaluating the impact of hockey actions. We next extend our state space to include actions.

6 STATE SPACE: PLAY SEQUENCES

Previous research on Markov process models of hockey has used only context features such as those described in Section 5. A Markov process model can answer questions such as how goal scoring or penalty rates depend on the game context [Thomas et al., 2013]. However, in this paper our focus is on the impact of a player's actions on a game. We therefore expand our state space with actions and action histories. The basic set of 8 possible actions is listed in Table 2. Each of these actions has two parameters: which team performs the action and zone Z where the action takes place. Zone Z represents the area of the ice rink in which an action takes place. Z can have values Offensive, Neutral, or Defensive, relative to the team peforming an action. For example, $Z =$ Offensive zone relative to the home team is equivalent to $Z =$ Defensive zone relative to the away team. A specification of an action plus parameters is an **action event**. Using action language notation [Levesque et al., 1998], we write action events in the form $a(T, Z)$. For example, $faceoff\,(home, neutral)$ denotes the home team wins a faceoff in the neutral zone. We usually omit the action parameters from generic notation and write a for a generic action event.

A **play sequence** h is a sequence of events starting with exactly one start marker, followed by a list of action events, and ended by at most one end marker. Start and end markers are shown in Table 2, adding shots and faceoffs as start markers, and goals as end markers. We also allow the empty history $\emptyset$ to count as a play sequence. A **complete** play sequence ends with an end marker. A **state** is a pair $s = \langle \mathbf{x}, h \rangle$ where $\mathbf{x}$ denotes a list of context features and h an action history. State s represents a play sequence consisting of action events $a_1, a_2, \ldots, a_n$ and with a particular GD, MD, and P as the context. If the play sequence is empty, then state s is purely a context node. Table 5 shows an example of a NHL play-by-play action sequence in tabular form. Potentially, there are $(7 \times 2 \times 3)^{40} = 42^{40}$ action histories. In our dataset, $1,325,809$ states, that is, combinations of context features and action histories, occur at least once. We store sequence data in SQL tables, so indexed sequence data can be retrieved quickly, and SQL provides native support for the necessary COUNT operations.

Table 5: Sample Play-By-Play Data in Tabular Format

GameId	Period	Sequence Number	Event Number	Event
1	1	1	1	PERIOD START
1	1	1	2	faceoff(Home,Neutral)
1	1	1	3	hit(Away,Neutral)
1	1	1	4	takeaway(Home,Defensive)
1	1	1	5	missed_shot(Away,Offensive)
1	1	1	6	shot(Away,Offensive)
1	1	1	7	giveaway(Away,Defensive)
1	1	1	8	takeaway(Home,Offensive)
1	1	1	9	missed_shot(Away,Offensive)
1	1	1	10	goal(Home,Offensive)
1	1	2	11	faceoff(Away,Neutral)
...				

7 STATE TRANSITIONS

If h is an incomplete play sequence, we write $h \star a$ for the play sequence that results from appending a to h, where a is an action event or an end marker. Similarly if $s = \langle \mathbf{x}, h \rangle$, then $s \star a \equiv \langle \mathbf{x}, h \star a \rangle$ denotes the unique successor state

that results from executing action a in s. This notation utilizes the fact that context features do not change until an end marker is reached. For example, the goal differential does not change unless a goal event occurs. If h is a complete play sequence, then the state $\langle \mathbf{x}, h \rangle$ has a unique successor $\langle \mathbf{x}', \emptyset \rangle$, where the mapping from $\mathbf{x}$ to $\mathbf{x}'$ is determined by the end marker. For instance, if the end marker is $goal(Home)$, then the goal differential increases by 1. A sample of our state transition graph is shown in Figure 1.

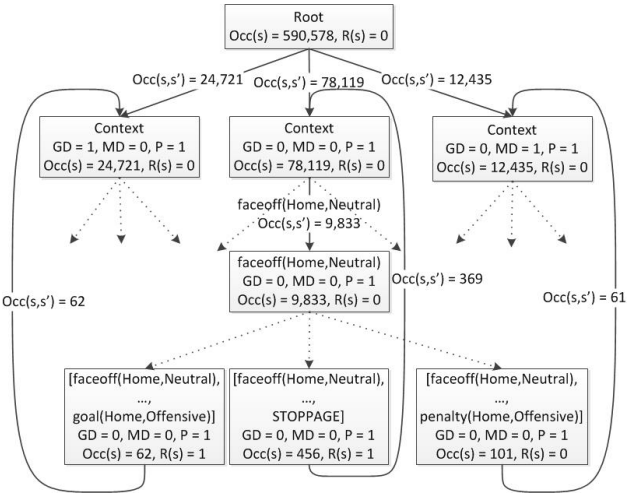


Figure 1: State Transition Graph

Since the complete action history is encoded in the state, action-state pairs are equivalent to state pairs. Therefore we can model transitions from state to state only, rather than transitions from state to state given an action, even though we are mainly interested in the effects of actions. For example, we can write $Q(s \star a)$ to denote the expected reward from taking action a in state s, where Q maps states to real numbers, rather than mapping action-state pairs to real numbers, as is more usual. In reinforcement learning terms, this means the Q-function can be computed by value iteration applied to states.

8 REWARD FUNCTIONS: NEXT GOAL AND NEXT PENALTY

A strength of Markov Game modelling is value iteration can be applied to many reward functions depending on what results are of interest. In this paper we focus on two: scoring the next goal, and receiving the next penalty (a cost rather than a reward). These are important events that change the course of an ice hockey game, and the corresponding Q-functions are easily interpreted. We have computed Q-functions for other reward functions but do not present these due to lack of space. Recall that states encode action histories. So we can define rewards as being associated with states only rather than state-action pairs. The two next-event objectives can be be represented in the Markov Game model as follows.

1. For any state s with a complete play sequence that ends in a Home resp. Away goal, we set $R_H(s) := 1$ resp. $R_A(s) := 1$. For other states the reward is 0.

2. Any state s with a complete play sequence that ends in a Home resp. Away goal is an absorbing state (no transitions from this state).

With these definitions, the expected reward represents the probability that if play starts in state s, a random walk through state space of unbounded length ends with a goal, or penalty, for the Home team resp. the Away team.

9 CONSTRUCTING THE STATE TRANSITION GRAPH

The main computational challenge is to build a data structure for managing the state space. The state space is large because each (sub)sequence of actions defines a new state. Since we are modelling the actual hockey dynamics in the policy-on setting, we need consider only action sequences observed in some NHL match, rather than the much larger space of all possible action sequences. We use the classic AD-tree structure [Moore and Lee, 1998] to compute and store sufficient statistics over observed action sequences. The AD-tree is a tree of play sequences where a node is expanded only with those successors observed in at least one match. The play sequence tree is augmented with additional edges that model further state transitions; for example, a new action sequence is started after a goal. The augmented AD-tree structure compactly manages sufficient statistics, in this case state transition probabilities. It also supports value iteration updates very efficiently.

We outline an algorithm for Context-Aware State Transition Graph. The root is an empty node with no context or event information. For each node, the context information GD, MD, and P are set when the new node is created, and the new action a is added to the sequence along with the zone Z that a occurs in. The reward $R(s)$ is also applied to each node, and the value of $R(s)$ is dependent on the objective function. The node counts $Occ(s)$ and edge counts $Occ(s, s')$ are applied to each node and edge respectively, and are used to generate transition probabilities TP for the value iteration using observed frequencies. The function $incrementCount(s)$ is used to set node count $Occ(s)$, and $incrementCount(s, s')$ is used to set edge count $Occ(s, s')$. The NHL play-by-play event data records goals, but no separate event for the shot leading to the goal exists. Following [Schuckers and Curro, 2013], we record the shot leading to the goal in addition to the goal itself by injecting a shot event into the event sequence prior to the goal.

10 VALUE ITERATION

Recall that since states encode action histories, learning the expected value of states is equivalent to learning a Q-function (Section 7). In reinforcement learning terms, there is no difference between the value function V and the Q-function. We can therefore apply standard value iteration over states [Sutton and Barto, 1998] to learn a Q-function for our ice hockey Markov Game. Algorithm 1 shows pseudo-code. We compute separate Q-functions for the Home team and for the Away team. Since we are in the policy-on setting, we have a fixed policy for the other team. This means we can treat the other team as part of the environment, and reduce the Markov Game to two single-agent Markov decision processes.In our experiments, we use a relative convergence of 0.0001 as our convergence criterion, and 100,000 as the maximum number of steps.

Algorithm 1 Value Iteration Dynamic Programming Algorithm

Require: MDP, convergence criterion c, maximum number of iterations M

$lastValue = 0$
$currentValue = 0$
$converged = false$
for $i = 1; i \leq M; i \leftarrow i + 1$ **do**
  for all states s in the MDP **do**
    if $converged == false$ **then**
      $Q_{i+1}(s) = R(s) + \frac{1}{Occ(s)} \sum_{(s,s') \in E} (Occ(s,s') \times Q_i(s'))$
      $currentValue = currentValue + |Q_{i+1}(s)|$
    end if
  end for
  if $converged == false$ **then**
    if $\frac{currentValue - lastValue}{currentValue} < c$ **then**
      $converged = true$
    end if
  end if
  $lastValue = currentValue$
  $currentValue = 0$
end for

11 EVALUATION AND RESULTS

We discuss the results of action values in Section 11.1 and player values in Section 11.2. Our state transition graph is evaluated in Section 11.3.

11.1 ACTION IMPACT VALUES

The main quantity we consider is the **impact** of an action as a function of context (= Markov state). This is defined as follows:

$$impact(s,a) \equiv Q_T(s \star a) - Q_T(s)$$

where T is the team executing the action a. In a zero-sum game, the state value is usually defined as the final result following optimal play. Intuitively, the value specifies which player has a better position in a state. Since we are not modelling optimal play, but actual play in a policy-on setting, the expected difference in rewards is the natural counterpart. The impact quantity measures how performing an action in a state affects the expected reward difference. Figure 2 shows a boxplot for the action impact values as they range over different contexts, i.e., states in the Markov Game model. (Boxplots produced with MATLAB R2014a) It is clear from Figure 2 that *depending on the context and event history, the value of an action can vary greatly.* The context-dependence is observed for both scoring goals and receiving penalities.

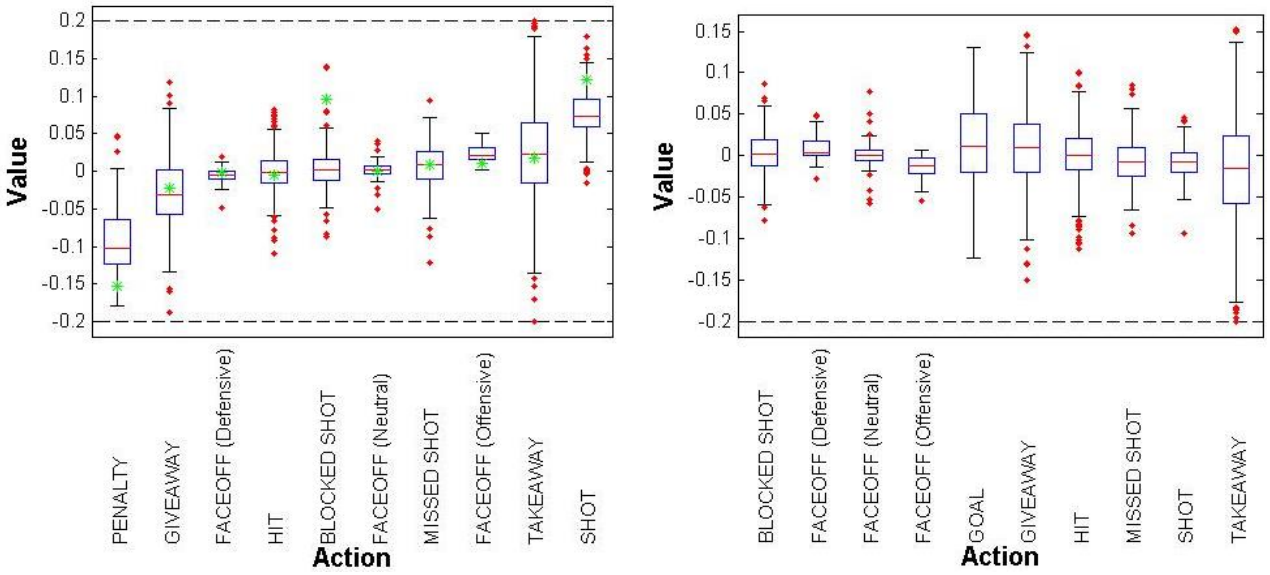
Impact on Scoring the Next Goal. All actions, with the exception of faceoffs won in the offensive zone, have at least one state where the action has a positive impact, and another state with a negative impact. Examples of context-dependence include the following.

(1) Blocking the first shot on net when killing a penalty is bad ($impact = -0.0864$), but blocking the second shot on net is very good ($impact = 0.1399$).

(2) Receiving a penalty when on the powerplay is very bad ($impact = -0.1789$), but if a player on the penalty kill can goad their opponent into an offsetting penalty, it is good ($impact = 0.0474$).

The THoR player ratings compute the impact of actions based on goals that immediately follow the action ([Lock and Schuckers, 2009; Schuckers et al., 2011]; see Section 2). The values given for each action in [Lock and Schuckers, 2009] are displayed as an asterisk in Figure 2(a). The THoR values agree with our median impact values in terms of whether an action generally has positive or negative impact. For example, penalties are known to generally be good for the opposing team, and shots are good for the player's team. THoR values are close to the median Markov model values in 6 out of 10 cases. This comparison suggests that THoR aggregates action values over many contexts that the Markov model makes explicit.

Impact on Receiving Penalties. The range of action values with the probability of the next penalty as the objective function is shown in Figure 2(b). Faceoffs in the Offensive Zone and takeaways cause penalties for the opponent. Goals and giveaways tend to cause a penalty for the player's team. For goals, this finding is consistent with the observation that there are fewer penalties for teams with higher leads [Schuckers and Brozowski, 2012]. A possible explanation is referees are reluctant to penalize the trailing team.



(a) Impact on the probability of Scoring the Next Goal. Higher numbers are *better* for the team that performs the action.

(b) Impact on the probability of Receiving the Next Penalty. Higher numbers are *worse* for the team that performs the action.

Figure 2: Action Impact Values vary with context. The central mark is the median, the edges of the box are the 25th and 75th percentiles. The whiskers are the default value, approximately 2.7 sd.

11.2 PLAYER VALUATIONS

Table 6: 2013-2014 Top-10 Player Impacts For Goals

Name	Goal Impact	Points	+/-	Salary
Jason Spezza	29.64	66	-26	$5,000,000
Jonathan Toews	28.75	67	25	$6,500,000
Joe Pavelski	27.20	79	23	$4,000,000
Marian Hossa	26.12	57	26	$7,900,000
Patrick Sharp	24.43	77	12	$6,500,000
Sidney Crosby	24.23	104	18	$12,000,000
Claude Giroux	23.89	86	7	$5,000,000
Tyler Seguin	23.89	84	16	$4,500,000
Max Pacioretty	22.54	60	8	$4,000,000
Patrice Bergeron	22.26	62	38	$4,550,000

The value of each action can be applied to players as they perform that action, and a player's action values are aggregated over a season to produce a season impact score. We compare impact on Next Goal Scored with three other player ranking metrics: points earned, salary, and +/-. Player impact with respect to goals is shown in Table 6. Since these players have a high impact on goals, they also tend to have a positive +/- rating. Jason Spezza is an anomaly, as he has the highest impact score but a very negative +/- score. This is due to his team performing poorly overall in the 2013-2014 season, and the team overall had a goal differential of -29, one of the highest goal differentials that season. This example shows the action value approach can distinguish a player who generally performs useful actions but happens to be on a poor team. In Table 7, we see player impact with respect to Next Penalty Received. High impact numbers indicate a tendency to cause penalties for a player's own team, or prevent penalties for the opponent. We compare the Q-function impact numbers to Penalties in Minutes (PIM), salary, and +/-. Players with high Q-function numbers have high penalty minutes as we would expect. They also have low +/-, which shows the importance of penalties for scoring chances. Their salaries tend to be lower. There are however notable exceptions, such as Dion Phaneuf, who draws a high salary although his actions have a strong tendency to incur penalties.

Table 7: 2013-2014 Top-8 Player Impacts For Penalties

Name	Penalty Impact	PIM	+/-	Salary
Chris Neil	62.58	211	-10	$2,100,000
Antoine Roussel	54.26	209	-1	$625,000
Dion Phaneuf	52.52	144	2	$5,500,000
Zac Rinaldo	48.65	153	-13	$750,000
Rich Clune	47.08	166	-7	$525,000
Tom Sestito	46.34	213	-14	$650,000
Zack Smith	44.55	111	-9	$1,500,000
David Perron	42.49	90	-16	$3,500,000

11.3 LESION STUDY WITH DIFFERENT TRANSITION GRAPHS

The transition graph construction algorithm facilitates changing the possible state transitions. We utilize this in our experiments to study how different propagation models

affect the impact of actions on Next Goal Scored. Specifically, we consider three different transitions graphs of increasing density, their sizes shown in Table 8. The number of states/nodes 1,325,809 is the same for all graphs.

Local Transitions Only State transitions occur only within a play sequence, not across play sequences.

Penalty Transitions State transitions occur from penalty leaf nodes to successor context nodes.

Full Transition Graph Includes loopback edges from all leaf nodes to context nodes, as defined in Section 6.

Table 8: Size of State Transition Graphs

	Local	**Penalty**	**Full**
Number of Edges	1,325,808	1,382,780	1,662,504

Action impact changes value depending on the state transition graph. With the local transition graph, value iteration computes the impact of an action on the current play sequence only. Thus the Q-value differential for context states, with the initial empty play sequence, can be obtained from Table 4. The average difference in action values, as well as the standard deviation of the differences, are shown in Table 9. While the aggregate effects provide insight into medium-term hockey dynamics, they do not reflect the considerable context dependence shown by the standard deviations of the impact differentials. The penalty transition graph propagates to the next sequence the effect of penalties only. Propagating the effect of penalties changes most the estimation of the impact of penalties. This change reflects that receiving a penalty lowers the chances of scoring the next goal. Less obviously, winning a faceoff in the offensive zone has a relatively high positive indirect impact on scoring the next goal, via increasing the probability of a penalty against the opposing team. The effect of winning an offensive zone faceoff can also be seen in Figure 2(b). Comparing the full transition graph with penalty propagation only, we still find the strongest average impact change for penalties. This shows that penalties have ripple effects on goals via events other than penalties. This indirect impact is shared by Offensive Zone Faceoff Wins.

12 CONCLUSION

We have constructed a Markov Game Model for a massive set of NHL play-by-play events with a rich state space.Tree-based data structures support efficient parameter estimation and storage. Value iteration computes the values of each action given its context and sequence history—the Q-function of the model. Compared to previous work that assigns a single value to actions, the

Table 9: Difference In Action Impact Values for Next Goal Scored, Across Transition Graphs

	Full vs. Penalty		**Penalty vs. Local**	
Blocked Shot	0.0001	0.0210	-0.0003	0.0126
Faceoff (Defensive)	-0.0030	0.0455	-0.0018	0.0225
Faceoff (Neutral)	0.0013	0.0464	0.0006	0.0203
Faceoff (Offensive)	0.0038	0.0432	0.0024	0.0260
Giveaway	-0.0003	0.0245	-0.0001	0.0142
Hit	0.0000	0.0194	-0.0001	0.0126
Missed Shot	-0.0001	0.0218	0.0003	0.0130
Penalty	**-0.0190**	0.0278	**-0.0235**	0.0337
Shot	0.0002	0.0191	0.0002	0.0103
Takeaway	0.0006	0.0245	0.0003	0.0146

Q-function incorporates two powerful aspects of valuing hockey actions: (1) It takes into account the context of the action, represented by the Markov Game state. (2) It models the medium-term impact of an action by propagating its effect to future states.Analysis of the computed Q-function shows the impact of an action varies greatly with context, and medium-term ripple effects make a difference. One application of our model is to evaluate the performance of players in terms of the total impact of the actions they perform. Action impact scores are calculated for players with respect to different objective functions. Impact scores for the next goal correlate with points and +/- statistics. The impact of players on the next penalty has not been considered, and show some surprises, as some highly-paid players hurt their team by causing penalties. In sum, the Q-function is a powerful AI concept that captures much information about hockey dynamics as the game is played in the NHL.

Future Work The NHL data provides a rich dataset for real-world event modelling. A number of further AI techniques can be applied to utilize even more of the available information than our Markov Game model does. A promising direction is to extend our Markov Game model, which is discrete with data about continuous quantities. These include (i) the time between events, (ii) the absolute game time of the events, (iii) location of shots (however, reported shot locations are noisy [Krzywicki, 2005]). Our use of reinforcement learning techniques has been mainly for finding patterns in a rich data set, in the spirit of descriptive statistics and data mining. Another goal is to predict a player or team's future performance based on past performance. Machine learning methods aim to provide reliable generalization; the set of techniques for sequence modelling would be able to leverage play sequence information. A promising model class are Piecewise Constant Conditional Intensity Models for continuous time event sequences [Gunawardana et al., 2011; Parikh et al., 2012]. These models are especially well suited for sequences with a large set of possible events, such as our action events.

References

Buttrey, S., Washburn, A., and Price, W. (2011). Estimating nhl scoring rates. *Journal of Quantitative Analysis in Sports*, 7(3).

Churchill, D. and Buro, M. (2013). Portfolio greedy search and simulation for large-scale combat in starcraft. In *IEEE Conference on Computational Intelligence in Games (CIG)*, pages 1–8.

Gramacy, R., Jensen, S., and Taddy, M. (2013). Estimating player contribution in hockey with regularized logistic regression. *Journal of Quantitative Analysis in Sports*, 9:97–111.

Gunawardana, A., Meek, C., and Xu, P. (2011). A model for temporal dependencies in event streams. In *Advances in Neural Information Processing Systems*, pages 1962–1970.

Hirotsu, N., Wright, M., et al. (2002). Using a markov process model of an association football match to determine the optimal timing of substitution and tactical decisions. *Journal of the Operational Research Society*, 53(1):88–96.

Krzywicki, K. (2005). Shot quality model: A logistic regression approach to assessing nhl shots on goal.

Levesque, H., Pirri, F., and Reiter, R. (1998). Foundations for the situation calculus. *Linköping Electronic Articles in Computer and Information Science*, 3(18).

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the eleventh international conference on machine learning*, volume 157, pages 157–163.

Lock, D. and Schuckers, M. (2009). Beyond +/-: A rating system to compare nhl players. Presentation at joint statistical meetings.

Macdonald, B. (2011). A regression-based adjusted plus-minus statistic for nhl players. *Journal of Quantitative Analysis in Sports*, 7(3):29.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

Moore, A. W. and Lee, M. S. (1998). Cached sufficient statistics for efficient machine learning with large datasets. *J. Artif. Intell. Res. (JAIR)*, 8:67–91.

National Hockey League (2014). National hockey league official rules 2014-2015.

Parikh, A. P., Gunawardana, A., and Meek, C. (2012). Conjoint modeling of temporal dependencies in event streams. In *UAI Bayesian Modelling Applications Workshop*.

Russell, S. and Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Schuckers, M. and Brozowski, L. (2012). Referee analytics: An analysis of penalty rates by national hockey league officials. In *MIT Sloan Sports Analytics Conference*.

Schuckers, M. and Curro, J. (2013). Total hockey rating (thor): A comprehensive statistical rating of national hockey league forwards and defensemen based upon all on-ice events. In *7th Annual MIT Sloan Sports Analytics Conference*.

Schuckers, M. E., Lock, D. F., Wells, C., Knickerbocker, C. J., and Lock, R. H. (2011). National hockey league skater ratings based upon all on-ice events: An adjusted minus/plus probability (ampp) approach. Unpublished manuscript.

Schumaker, R. P., Solieman, O. K., and Chen, H. (2010). *Research in Sports Statistics*. Springer US.

Sidhu, G. and Caffo, B. (2014). Moneybarl: Exploiting pitcher decision-making using reinforcement learning. *The Annals of Applied Statistics*, 8(2):926–955.

Spagnola, N. (2013). The complete plus-minus: A case study of the columbus blue jackets. Master's thesis, University of South Carolina.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning : an introduction*. MIT Press, Cambridge, Mass.

Thomas, A., Ventura, S., Jensen, S., and Ma, S. (2013). Competing process hazard function models for player ratings in ice hockey. *The Annals of Applied Statistics*, 7(3):1497–1524.