Linköping University | Department of Computer and Information Science
Master thesis, 30 ECTS | Datateknik
202018 | LIU-IDA/LITH-EX-A--2018/001--SE

A very very long title
– with a subtitle

En himla bra svensk titel

Författaren

Supervisor : Min handledare
Examiner : Min examinator

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Abstract

`Abstract.tex`

Acknowledgments

`Acknowledgments.tex`

Contents

List of Figures

List of Tables

Todo list

1 Introduction

1.1 Motivation

Sport analytics has been traditionally used to understand the relevant features of a sport for a player to succeed as well as to resume the most important information on matches and players using statistical Key Performance Indicators (KPI). Evaluating performance and characteristics of elite athletes [7] or assessing important metrics by sport categories (e.g. net and wall games, invasion games, and striking and fielding games) [9] have tried to give better understanding on sport dynamics.

The usage of video-recording games, the increase in computer power and the rise of machine learning techniques have enabled to evaluate sports in a way impossible before. The evaluation of traditional and new metrics can provide insights to further understanding the game dynamics of a particular sport (e.g. core strategies, effects of particular events on a sequence of events).

In a sport as ice hockey, it is crucial for teams to understand well game dynamics, players' performance and successful scoring strategies to develop and improve their playing performance. This may result, with good management, to more audience in matches, better sponsors, more money to fulfill and increase the objectives of the team.

Markov Decision Processes (MDP), in the view of Reinforcement Learning, provides a general framework to deal with decision making under particular events. With that, one can not only learn which are the best actions to take under a particular state, but also being able to see through historical data which are the most useful/successful strategies to win a match. Also, the creation of more complex valuation metrics for teams and players' valuation (e.g. understanding which players play best/worst together) can be of useful insight for hiring/firing purposes, even for gamblers to bet with meaningful knowledge.

1.2 Aim

The underlying purpose of this project is to analyze the usage of MPDs on players and events' evaluation using the NHL data used in the Routley and Schulte's article [19].

1.3 Research questions

The research questions that this paper addresses are:

1. How a MDP/RL is used to evaluate actions under certain time-series events ?
2. How can I store time series data for the usage of a MDP?
3. What is the MDP valuation of each player of the NHL league per match ?
4. Can I predict player's performance based on their performances in the previous matches ?
5. Is there a way to use a MDP to create a metric for evaluation of players for hiring/maintaining/firing?

1.4 Delimitations

The scope of this project is assessing the usefulness of the Markov Decision Process model to provide new meaningful insights for Ice Hockey for the NHL Dataset. The project does not try to improve or optimize the model or the solving algorithms but rather study the behavior, advantages and limitations of the MDP in Ice Hockey as well as Time Series Modelling for the impacts derived of the MDP. Also, the project will serve myself to boost my skills working on large datasets using SQL.

1.5 Literature Review

Markov Decision Processes (MDP) is a framework classically used for taking sequential decisions under uncertainty in a perfectly observable state. It was developed during the 1950s by Bellman [3], and it is currently used in several fields such as robotics, economics, sports, medicine or manufacturing.

In order to learn which actions are the best ones (i.e. they give higher rewards), the sequence of events with their associated actions need to be explored. Littman [13] explained the theory of a MDP in the view of Q-learning like algorithm from the reinforcement learning theory to find transition probabilities to states and best strategies on the framework of two agents with opposed goals while being in a common environment [8].

There is a lot of research made on player performance for different sports. Some related papers on other sports are fittting a MDP on rugby matches to evaluate which sequences of events are more prone to scores to find the most effective rugby strategies [2] or evaluating the EPV (Expected Point Value) per possession on basketball taking into account players configuration in the field and the ball position [4].

In Ice Hockey, the main metric used across years is the +/- metric: the difference on scoring between your team and the opponent relatively to the current players on the ice. In other words, if you are a player on the ice rink and your team scores, you will be awarded with a +1. On the other hand, if you are on the ice ring and the opponent team scores, you as a player will get -1 point. This non-marginal metric has been tried to be improved in several papers [17, 16] [6] by using regressions (eg. logistic, ridge regression) trying to assess which player has had more impact marginalizing other players' valuation.

Schuckers [20] gave a value to all events performed as a measure to how nearer the possibility of a goal was in the following 20 seconds. Routley and Shulte [19] used the NHL dataset without a window event taking into account the following context: the Zone an event was happening, the Manpower Differential (MD) across teams, the Period in which the match was happening and the Goal Difference (GD) between teams. With this, they created a MDP process that gives players a valuation based on the improved probability of an event to scoring vs receiving a score, extending it lately to a better dataset taking into account more actions and better locations on the field [21], [22]. Also, Shulte [14] used LSTM (Deep Neural Networks) to calculate player's performance on the NHL database using more context variables such as Time Remain of the match or whether and action was a success.

MDPs have also been used to calculate the Probability of winning a match given that an action is performed by a player under certain context variables (Goal and Manpower differential) [10]. Moreover, taking the real impact of a player's score to know which scorers are more valuable [18] or calculating the best pair of players based on performance for each team taking into account positions of players and the time these players has played together each match [15].

2 Theory

This thesis deals with discrete variables and states. Therefore, the theory described in this project is focused on discrete data, even if a large part of it is common with continuous data. The notation as well as the explanation used in the article follows the notation and the structure of the MIT book *Decision Making Under Uncertainty: Theory and Application (2015)* [11].

2.1 Markov Decision Processes (MDP)

Markov Decision Processes (MDP) provide a framework for sequential decision making in situations where the outcome of a given action is non-deterministic. More specifically, an agent chooses action a_t at time t observing a state s_t based on an associated Reward $R(s_t, a_t)$. This state evolves according to a defined state transition function $T(s'|s_t, a_t)$ which depends on the action a_t taken at that particular state. The system modeled is assumed to follow the Markov property, meaning that its state at time t depends only on its state at time $t-1$.

Illustration 2.1 represents graphically a Markov Decision Process with States (S), Actions (A), Rewards (R) for 3 general sequential processes.

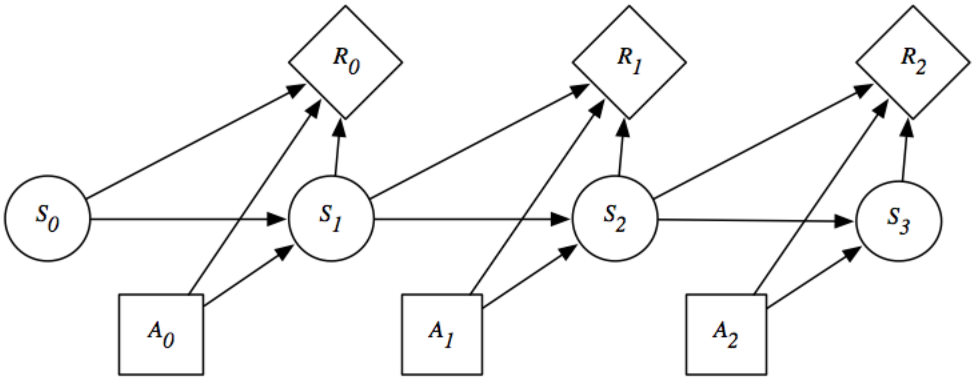


Figure 2.1: Graphical Illustration of a Markov Decision Process with States (S), Actions (A), Rewards (R) for 3 general sequential processes

In other words, a MPD can be explained by the tuple (S,A,T,R):

- S, the set of states
- A, the set of actions the agent can engage in
- T: S $\mapsto \prod$(S), the transition function
- R: S x A $\mapsto \mathbb{R}$, the reward function dependent on the action taken at each state

MDP example

A clear example of a MDP is the modelling of an ice hockey player's decision in control of the puck in the offensive zone of the field: The 4-tuple presented before <(S,A,R,T,)> would be used to describe what each variable would represent:

- States, representing the set of conditions a specific player might be in under which she/he performs an action (e.g.player shots the puck the puck in the offensive zone (Zone), winning 1-0 (score of the match), same number of players per team (manpower differential) and first period (period the match is in)).
- Actions the player can perform, which might be passing, shot, dribble.
- Rewards: The player receives a reward for engaging into an action that improves statistically the probability of scoring against the probability of performing an action that reduces the probability of score or increases the probability of receiving a goal. As an example, the reward might be set to 1 for scoring a goal and -1 for receiving a goal.
- Transition function, representing the set of probabilities of the following state given the action performed in a particular current state (e.g. related to the example set in States: it describes, among others, the probability of scoring given a shot in the light of the context the player is in, as well as the possibility of scoring given that you pass the ball to a player that is alone in the offensive zone)

The Utility and Reward function

The choice of an action at each point in time has an associated reward r. In a finite horizon problem (finite number of states, i.e. n states), the total utility is defined by the sum of associated rewards to the action performed:

$$\sum_{t=0}^{n-1} r_t \tag{2.1}$$

The problem of this simple approach is that for an infinite time horizon (i.e. $t = \infty$) the reward function could reach the value ∞, so that no specific action would be preferred under this case. The simplest way to solve it is to add a discount factor γ^t (which can take values between 0 and 1) that is affected by time t. With that, a nearer reward would be preferable in case of equal rewards.

$$\sum_{t=0}^{n-1} \gamma^t r_t \tag{2.2}$$

That is one way to deal with the infinite horizon problem, but there are other ways such as calculating the average reward:

$$\lim_{x \to \infty} \frac{1}{n} \sum_{t=0}^{n-1} r_t \tag{2.3}$$

Defining policies and utilities

A policy that determines the choice of an action at a point in time t is expressed as $\pi_t(s)$. When transitions and rewards are stationary (i.e. they are equal across time), a policy π is only dependent on the current state s at that point in time (i.e. recall the Markov assumption from the beginning of the section: the current state s defines all previous states). The optimal policy π^* at state s is defined as the policy that maximizes the expected utility:

$$\pi^*(s) = \arg\max_{\pi} U^{\pi}(s) \tag{2.4}$$

The optimal policy of a MDP is the defined by the sequence of actions that maximize the Expected Utility of the whole MDP. That is the sum of the expected reward at each time step, or equivalently, the sum of expected rewards associated to n actions and states. The calculation of the total policy for a certain number of actions is called policy evaluation.

$$U_t^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s'|s, \pi(s)) U_{t-1}^{\pi}(s') \tag{2.5}$$

For an infinite time horizon with discounted rewards (i.e. with discount factor γ) equation 2.5 shows the expected Utility of a sequence of policies is explained by the Reward got at the current given the policy π chosen at that state plus the sum of all future states defined by the Transition function (which defines with probability the exact future state given the current policy and state) and its associated Utility function at the future state multiplied by the discount factor γ.

Optimization of MDP

If the transition probabilities and the Rewards for each action are known, one can calculate the optimal policy π^* as follows:

- **Policy evaluation:** For any given policy π_l, the expected utility of the policy U^{π_k} is calculated
- **Policy Improvement**: Calculating a new policy using the pseudocode shown below in *Algorithm 1*.

The pseudo-code for the whole algorithm, called policy iteration, can be found below:

Each iteration on the policy iteration algorithm is computationally expensive. Thus, another similar algorithm that is used is the value iteration algorithm. For an associated time horizon n, the optimal utility (only for time horizon equal to n) is carried out following equation 2.6

$$U_n(s) = \max_a (R(s, a) + \sum_{s'} T(s'|s, a) U(s') \tag{2.6}$$

In the case of an infinite time horizon, the optimal utility is calculated using equation 2.7

$$U_n(s) = \max_a (R(s, a) + \gamma \sum_{s'} T(s'|s, a) U(s') \tag{2.7}$$

Then, the optimal policy can be extracted by just taking the actions that optimize the utility at each state.

The value iteration algorithm proceeds as follows:

In order to speed up convergence on algorithm 2, convergence is set to be reached when $|(U_k - U_{k-1})| < \delta$ being δ a parameter of the programmer's choice known as the Bellman residual.

Both algorithms will return the global optimal value function. On a rule of thumb though, policy iteration will converge in fewer iterations but is more computationally complex than value iteration.

Algorithm 1 Policy iteration

```
1.Initialization
$v(s) \in$ and $\pi(s) \in A(s)$ arbitrarily for all s ∈ S
2.Policy Evaluation
repeat $\Delta \leftarrow 0$
    for each s ∈ S do
        temp $\leftarrow v(s)$
        $v(s) \leftarrow \sum_{s'} p(s'|s, \pi(s)) * [r(s, \pi(s), s') + \gamma vs']$ which is $U_t^\pi$ from 2.5
        $\Delta \leftarrow$ max($\Delta$, | temp-v(s) | )
    end for
until $\Delta < \theta$ small positive value
3.Policy Improvement $policy_stable \leftarrow$ True
for each s ∈ S do
    temp $\leftarrow \pi(s)$
    $\pi(s) \leftarrow \arg\max_a \sum_{s'} p(s'|s, a)[R(s, a, s' + \gamma vs']$
    if $temp \neq \pi(s)$ then
        $policy_stable \leftarrow false$
    end if
end for
if $policy_stable$ then
    return stop and return v and $\pi$
else
    go to Policy Evaluation
end if
```

Algorithm 2 Value iteration

```
Initialize array v arbitrarily (e.g. $v(s) = 0$ for all s ∈ $S^+$)
repeat $\Delta \leftarrow 0$
    for each s ∈ S do
        temp $\leftarrow v(s)$
        $\max_a v(s) \leftarrow \sum_{s'} p(s'|s, \pi(s)) * [r(s, \pi(s), s') + \gamma vs']$ which is 2.7
        $\Delta \leftarrow$ arg max($\Delta$, | temp-v(s) | )
    end for
until $\Delta < \theta$ small positive value
Output deterministic policy $\pi$ such that
$\pi(s) \leftarrow \arg\max_a \sum_{s'} p(s'|s, a)[R(s, a, s') + \gamma vs']$
```

Learning a MDP

MDP provides a framework to deal with sequences of states: it enables to learn the transition probabilities between states given actions taken at each point in time. With that, one can decide which events happening are meant to have a reward, and thus evaluating which events are more significant for a specific event to happen.

The transition probabilities are explained following Littman's notation [13]:

- Occ(s) is the number of times state s occurs in the dataset under specific context variables
- Occ(s, s') is the number of times state s is followed by state s'.
- The transition probability $T(s'|s, a)$ then is learned via

$$\frac{Occ(s, s')}{Occ(s)}$$

Parallelism with Reinforcement Learning

This section is merely informative to understand the MDP and the value iteration algorithm from another perspective.

The main objective of reinforcement learning is to make an agent engage in optimal actions in a space. This is learned through performing actions and seeing its results. Those actions which results are understood as being the optimal ones are given positive rewards whereas those with bad results are given a penalty. As an example, we might say that the main objective in hockey is scoring a goal. Thus, whatever action that goes towards the direction of scoring is given a reward, otherwise, it is given a penalty.

However, at the beginning, we do not know which actions in the space gives positive/negative rewards. Thus, the Q-table is used to store and see which actions are indeed positive/negative. In other terms, the Q-table is the expected maximum future reward for each action. In order to do so, the Q-learning algorithm is used, presented as a flowchart below:

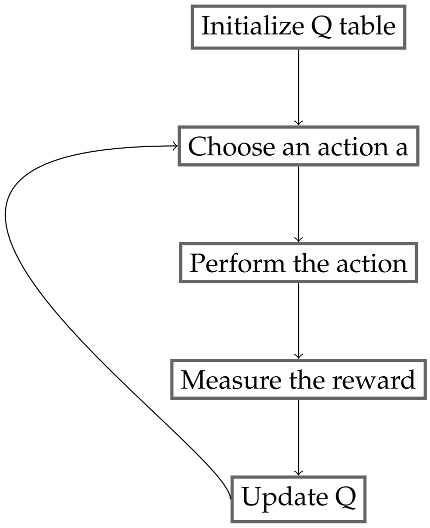


Figure 2.2: Process of the Q learning algorithm

Explained in a little more detail, the functioning is as following:

1. Initiate Q-values from Q(s,a) arbitrarily for all pairs of (s,a).
2. While the number iterations is smaller than the maximum number of iterations stated or while there is no convergence (normally a threshold number is set which accounts for convergence when change is smaller than that number):
   a) Choose an action a in the current state s based on current Q-value estimates (s,·)
   b) take the action chosen and see the next state s′ and the reward r
   c) Update $Q(s,a) = Q(s,a) + \epsilon[r + \gamma \max_a'(Q(s',a') - Q(s,a))]$

The Q-learning process is iterative (see figure 2.2). On the first time, initial values will be set (normally zeros). Then, for as many iterations as the user specifies or until convergence, the following procedure will be carried out: an action will be chosen and performed, a reward will be given to that action based on the Q current estimates and then the Q table is updated. However, at the beginning, all actions are seen equally interesting (a matrix of zeros). Therefore, a value for ϵ must be specified. This value is set at the beginning to 1 because we do not know anything about the Q-value of an action. Progressively, as we know more about the space, this epsilon will decrease, and by drawing a random number between zero and

1, if that random number > ϵ, then the best action (which we have already calculated) will be taken, else, a random action will be chosen. This allows a balance between exploitation (choosing best action with the information we have) and exploration (choosing a random action). On the 'c' part of the explanation, the updated Q(s,a) is the current Q value for that state and action (Q(s,a)), ϵ is the learning rate , R(s,a) is the reward for a specific action in that state, γ is the discounted rate and $\max_a'(Q(s',a'))$ is the maximum new expected future reward given the new state s' and new all possible actions at that new state.

2.2 AD-tree

Motivation for AD-trees on MDP

Finding the best policy at each time step (i.e. performing the best action every time with the current information), is unfeasible for MDPs with a large amount of state variables given its time constraint and its computer complexity.

Therefore, showing data in a compact way such that the transition probabilities of states and the rewards associated to each state must be easily and rapidly used for the usage of the MDP to calculate the evaluation of players.

Introduction to the AD-tree

An AD-tree is a machine learning method developed by B. Anderson and A. Moore, 1998 [1] to summarize relations of events that happen sequentially in a database. Their main purpose was establishing associative rules to extract some general rules based on support and confidence.

Following their article's notation, a literal is an attribute-value pair such as "event = shot". Then, L is all possible combinations of literals in the database. An association rule implies that $S1 \rightarrow S2$, where $S1, S2 \subset L$, and $S1 \cap S2 = \emptyset$. S1 will refer to the first event, being the antecedent, and S2 to the conditional event, called the consequent. Then, we can understand a time series dataset as a compound of rules, each literal associated by time with the following literal. As an example, $S1 \rightarrow S2$, could be on its simplest form as S1 being event = shot and S2 being event = goal.

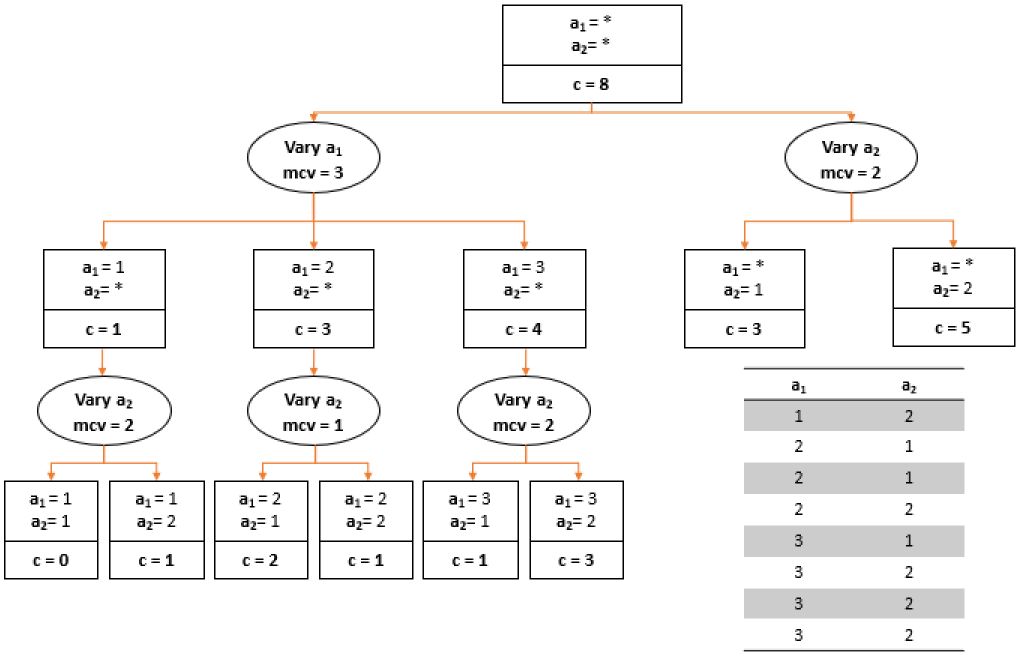
The support measure is a metric defined as the number of records in the database that match all the attribute-value pairs in S. In other words, it is counting how many times an association of events S1, S2 occur on the dataset, also expressed as $supp(S1 \cup S2)$. The more times an association occurs, the higher its support and the higher its significance.

The confidence measure is another metric defined as the percentage of records that matches $S1 \cup S2$ out of all records that match S1. That is, the fraction between support of the association rule $S1 \cup S2$ and the support of the antecedent S1. It is expressed as $conf(S1 \cup S2) = \frac{supp(S1 \cup S2)}{supp(S1)}$. The higher the confidence measure, the stronger and the more probable an association is, giving more importance to that specific association rule.

The AD-tree Data Structure

An AD-tree is a summary of a database with support measures for each state. This means that all possible queries have been calculated in advanced to provide basic statistics (support measure) for each query. In other words, the support of a literal can be found next to each literal.

To do that, a basic structure to record all the queries is needed (a tree structure with its support). There are two main components in the AD-tree structure. An ADnode represents a query and stores the number of times that query happens. Each ADnode, (shown as rectangles in figure 2.3) has child-nodes called "vary nodes" (shown as ovals). These "vary nodes" do not store counts but group ADnodes with one only feature. It can also contain the most



a_1	a_2
1	2
2	1
2	1
2	2
3	1
3	2
3	2
3	2

Figure 2.3: Graphical representation of an ADtree based on the dataset shown in the same figure. ADnodes are shown as rectanges. Vary nodes are shown as ovals.

common value (mcv). The "vary nodes" child of an ADnode has one child for each value v_j for feature a_i. These grandchildren ADnodes specialize the grandparent's query by storing the counts of the specific queries.

2.3 Time series

The time series explained in this thesis only serve the purpose of understanding the thesis work. Therefore, theory and models will not be explained in depth. To better understand the models that are going to be presented in this paper, *Time Series Analysis With Applications in R*, [J. Cryer, 2008] [5] is a book that provides a lot of practical examples using R's software. Also, the notation used in this part of the thesis follows [J. Cryer, 2008] [5] book's notation.

Motivation for Time Series modelling

The main objective behind time series modelling is being able to predict certain behaviours of variables through time. Given that the objective of the thesis is trying to predict player's performance valuation based on previous performances, time series modelling can be quite handy due to the ability of modelling different types of trends from previous observations and then forecast the following performances. It is often alleged that forecasting player's performance for match 26 may not depend on matches 1 and 2, but on the most recent previous matches (e.g. the last 5 matches). This could be taken as an evaluation of how good that player is in that specific point in time, therefore used to predict their next match performance.

The concept of Stationarity

In order to evaluate the structure of a stochastic process, it is assumed that the laws in which all stochastic processes occur do not change over time. This assumption is called stationarity.

The 3 main laws that define whether a process is stationary (referring in this thesis for weak stationarity) are:

- The mean function is constant over time: $\mathbf{E}(Y_t) = \mathbf{E}(Y_{t-k})$
- The covariance between Y_t and Y_s being s<t depends only on time difference |t-s| but not on actual times of s and t. Therefore: $Cov(Y_t, Y_s) = Cov(Y_{0,|t-s|})$
- The variance is : $Var(Y_t) < \infty$

The most typical stationary process is the white noise process, which is a sequence of a independent and identically distributed (iid) random variables e_t.

ARMA Models

Inside Stationarity, there are two main processes. An autoregressive process (AR) is a regression process where the current observation is a combination of the p previous observations plus some difference explained as e_t. This model was proposed by Yule in 1926:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t$$

A moving average process (MA) is a another regression process based on q previous white-noises with some specific weights. This model was firstly introduced by Slutsky in 1927:

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q}$$

ARIMA Models

In real life, though, time series data is not often stationary. Nevertheless, there exist techniques which might help us dealing with it.

Differencing is one of the most used methods for making data stationary, which is calculating the difference in Y for lags. As an example, a first order difference would be $\bigtriangledown Y_t = Y_t - Y_{t-1}$. Another method that is used together with differencing is taking logarithms of the data to avoid modelling exponential growth in the data set.

An Integrated autoregressive moving average models (ARIMA), introduced by Box and Jenkins in 1976, is a regression process that takes into account AR, differencing and MA processes to model time series. Considering W_t as $Y_t - Y_{t-1}$, an ARIMA process can be defined as

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + ... + \phi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \tag{2.8}$$

which can be expressed as :

$$\begin{aligned} Y_t - Y_{t-1} = \phi_1(Y_{t-1} - Y_{t-2}) + \phi_2(Y_{t-2} - Y_{t-3}) + ... + \phi_p(Y_{t-p} - Y_{t-p-1}) \\ + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \end{aligned} \tag{2.9}$$

and therefore

$$\begin{aligned} Y_t = (1 + \phi_1)Y_{t-1} + (\phi_2 - \phi_1)Y_{t-2} + ... + (\phi_p - \phi_{p-1})Y_{t-p} + \theta_p Y_{t-p-1} \\ + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \end{aligned} \tag{2.10}$$

which looks like an ARMA process. By using ARIMA processes, AR and MA processes are taken into account on one single process making it really interesting for its application.

3 Domain: Hockey Rules and Hockey Data

3.1 Hockey Rules

Ice hockey, documented for the first time in 1773 in the book *Sports and Pastimes, to Which Are Prefixed, Memoirs of the Author: Including a New Mode of Infant Education*, by Richard Johnson, is a two-team team sport played on an ice rink with 5 players plus a goalkeeper each team. The objective is to score more goals than the opponent in 3 periods of 20 minutes. On regular season, if the 3rd period ends up with a draw, there is an extra period of 5 minutes where the first to score immediately wins. In case there is no score, on a regular season, there will be shootouts for both teams. In playoffs, new periods of 20 minutes will continue until one team scores.

After a penalty occurs, the player who occasioned it will sit in the penalty box for a certain amount of minutes depending on the severity of the fault. The penalized team will be shorthanded, creating a Manpower Differential (MD) between the teams. A team has *powerplay* when their team has more players on the field than their opponent, resulting into manpower advantage. A *powerplay* goal means that a team scored with *Manpower differential*. The opposite is called a *shorthanded goal*.

3.2 Hockey Data

The datasets this project will use are the NHL dataset from the [Routley, 2015] paper [19] scrapped from *www.nhl.com*.

NHL Hockey Dataset

The NHL dataset is a relational database with 2,827,467 play-by-play events recorded by the NHL, containing complete 2007-2014 seasons (i.e. regular season and playoff games) and the first 512 games of the 2014-2015 regular season. General information about the dataset is displayed in Table 3.1a

As shown in table 3.1b, each event can either be an Action Event or an Start/End event. An Action Event represents those events performed by players, whereas Start/End events are events that stop time not performed by players. Each time event is continuous of another time event. That means we have a time series dataset. Additionally, each event performed by

Nº of events	2,827,467
Nº sequences of events	590,924
Nº of games	9,220
Nº of players	1,951
Nº of teams	32

(a) General Information

Action Events	**Start/End Events**
Faceoff	Period Starts
Missed Shot	Period Ends
Shot	Early Intermission Starts
Hit	Early Intermission Ends
Blocked Shot	Stoppage
Giveaway	Penalty
Goal	Game End
Takeaway	Shootout completed
	Game off

(b) Types of events

Figure 3.1: The NHL Dataset

a player (an action) has an associated Zone in which the event occurs (Home, Neutral, Away) and which team performs that event (Home or Away team).

4 Method

4.1 The AD-tree creation

Definition of Context variables

In sports, not only is it important to understand the actions players take, but also under which circumstances that player is deciding and performing that specific action. Inductively, in ice hockey, the probability between scoring a goal given that one team has more players in the field might be higher than when the same team has less. Probably, a reckless pass with fewer players in the rink might imply a higher risk of receiving a takeaway and therefore receiving a score. Also, it might not be equal the probability of scoring an extra goal in period 3 when you are losing by 1 goal than when you are winning by 3.

Thus, taking into consideration the context in which events are performed might be crucial to understand a game itself and evaluate which are the players that play better. With that, the choice of context variables that are important for the game are *Goal Differential*, *Manpower Differential* and *Period* [19]. *Goal Differential* (GD) is defined by how many goals the Home team is winning by. $GD = (HomeGoals - AwayGoals)$. A GD of 1 mean that the Home team wins by 1 goal whereas a GD of -1 stands for the Away team winning by 1. *Manpower Differential* is defined as the difference of players in the rink between both teams following the exact same logic as GD. $MD = HomePlayersinRink - AwayPlayersinRink$. Those context variables are the following ones displayed in the table 4.1

The table is showing the range in which those context variable might happen. As a summary of them, that means our events might happen under 17x7x7 combinations of context variables. That resumes in a total of 833 context states where each action could happen.

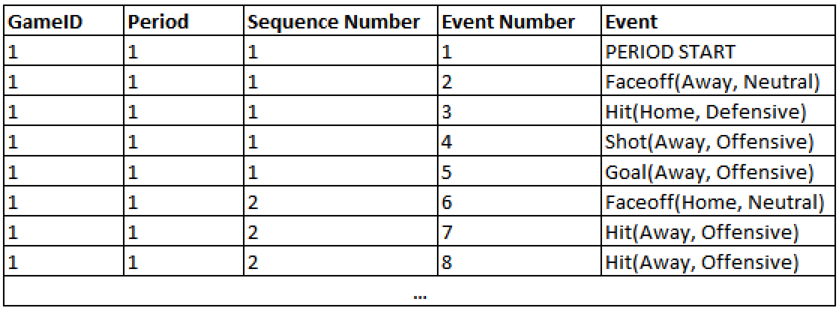
Notation	Variable Name	Range
GD	Goal Differential	[-8,8]
MD	Manpower Differential	[-3,3]
P	Period	[1,7]

Table 4.1: Context Variables associated to each event

Definition of a play sequence

As explained in the dataset information section 3.2, there are 8 main actions a player can perform (see table 3.1b). Those actions have associated a Zone Z in which the action is performed (Offensive, Neutral or Defensive) subjected to the teams T performing the action (Home, Away). A playing sequence is a series of action events. Each play sequence begins with a start marker. If the playing sequence is complete, it will end with an end marker. However, it may be that there are sequences are not complete. Start/End markers are those events in the right column of table 3.1b, adding faceoffs and shots as starting markers and goals as ending markers. To display it clearer here, action events will be explained in the notation form of a(T,Z) from [Levesque et al., 1998] [12], where T is Team and Z is Zone. Goal(Home, Offensive) is an example of the a(T,Z) structure.

An illustrative example is displayed below in figure 4.1:



GameID	Period	Sequence Number	Event Number	Event
1	1	1	1	PERIOD START
1	1	1	2	Faceoff(Away, Neutral)
1	1	1	3	Hit(Home, Defensive)
1	1	1	4	Shot(Away, Offensive)
1	1	1	5	Goal(Away, Offensive)
1	1	2	6	Faceoff(Home, Neutral)
1	1	2	7	Hit(Away, Offensive)
1	1	2	8	Hit(Away, Offensive)
...				

Figure 4.1: Illustrative example of the variables got in the time series dataset for a hockey match with format a(T,Z) of the events

Definition a state

Following Routley's notation, a state is denoted as s $< x,h >$ where x englobes the combination of context variables (being GD, MD and P) and h the action history (the variables in table 3.1b with Zone, and the Team Performing that action). If the sequence h is empty, the state is purely a context node. An example of this can be found in figure 4.2.

4.2 The AD-tree structure

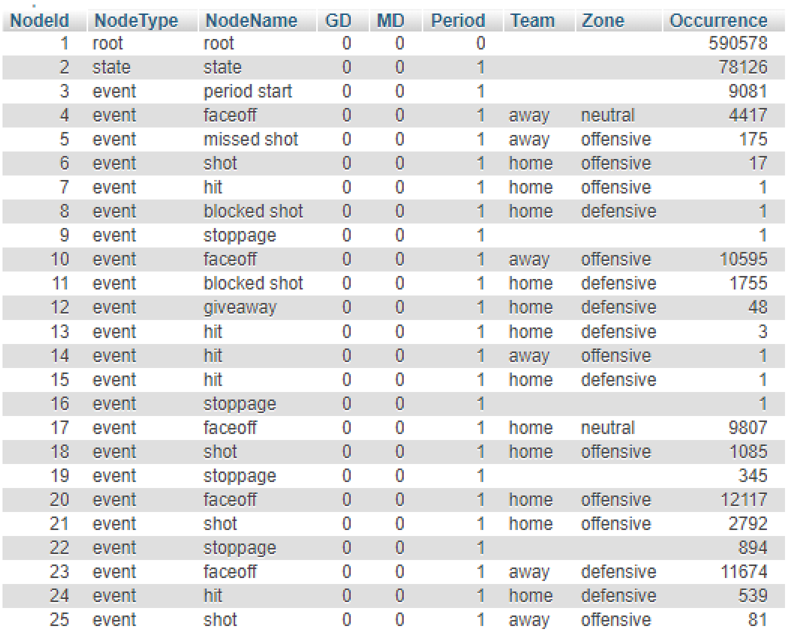
In order to deal with a high dimensional space, an AD-tree has been created using a SQL Database with four tables. Each table tries to store different basic useful information for the posterior calculus of the evaluation of players using MDP.

The Nodes table

The Nodes table is the AD-tree table with all the variables related to s$< x,h >$. Additionally, a variable NodeId is added to each Node working as a unique key for each node and an Occurrence variable, standing for how many times each state event happens in the whole dataset, or in other words, storing the support of each query (see figure 4.2).

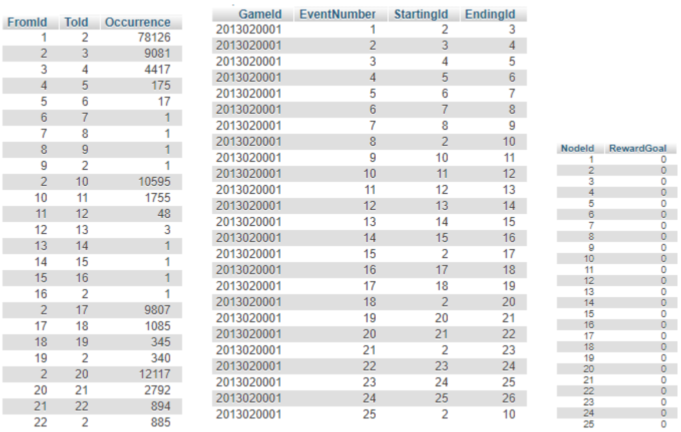
The Edges table

The Edges table stores the links between nodes in the dataset and its occurrence. The Variables FromId and ToId refers to the link created between two nodes (see figure 4.3a). These are the only paths considered later on as possible transitions from one node to another node



NodeId	NodeType	NodeName	GD	MD	Period	Team	Zone	Occurrence
1	root	root	0	0	0			590578
2	state	state	0	0	1			78126
3	event	period start	0	0	1			9081
4	event	faceoff	0	0	1	away	neutral	4417
5	event	missed shot	0	0	1	away	offensive	175
6	event	shot	0	0	1	home	offensive	17
7	event	hit	0	0	1	home	offensive	1
8	event	blocked shot	0	0	1	home	defensive	1
9	event	stoppage	0	0	1			1
10	event	faceoff	0	0	1	away	offensive	10595
11	event	blocked shot	0	0	1	home	defensive	1755
12	event	giveaway	0	0	1	home	defensive	48
13	event	hit	0	0	1	home	defensive	3
14	event	hit	0	0	1	away	offensive	1
15	event	hit	0	0	1	home	defensive	1
16	event	stoppage	0	0	1			1
17	event	faceoff	0	0	1	home	neutral	9807
18	event	shot	0	0	1	home	offensive	1085
19	event	stoppage	0	0	1			345
20	event	faceoff	0	0	1	home	offensive	12117
21	event	shot	0	0	1	home	offensive	2792
22	event	stoppage	0	0	1			894
23	event	faceoff	0	0	1	away	defensive	11674
24	event	hit	0	0	1	home	defensive	539
25	event	shot	0	0	1	away	offensive	81

Figure 4.2: Example of the Nodes table



FromId	ToId	Occurrence
1	2	78126
2	3	9081
3	4	4417
4	5	175
5	6	17
6	7	1
7	8	1
8	9	1
9	2	1
2	10	10595
10	11	1755
11	12	48
12	13	3
13	14	1
14	15	1
15	16	1
16	2	1
2	17	9807
17	18	1085
18	19	345
19	2	340
2	20	12117
20	21	2792
21	22	894
22	2	885

(a) The Edges table

GameId	EventNumber	StartingId	EndingId
2013020001	1	2	3
2013020001	2	3	4
2013020001	3	4	5
2013020001	4	5	6
2013020001	5	6	7
2013020001	6	7	8
2013020001	7	8	9
2013020001	8	2	10
2013020001	9	10	11
2013020001	10	11	12
2013020001	11	12	13
2013020001	12	13	14
2013020001	13	14	15
2013020001	14	15	16
2013020001	15	2	17
2013020001	16	17	18
2013020001	17	18	19
2013020001	18	2	20
2013020001	19	20	21
2013020001	20	21	22
2013020001	21	2	23
2013020001	22	23	24
2013020001	23	24	25
2013020001	24	25	26
2013020001	25	2	10

(b) The Node information table

NodeId	RewardGoal
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0

(c) The Reward table

Figure 4.3: Examples of the AD-tree tables

in the MDP. Non-existent pair of sequences are taken as impossible to occur and therefore with transition probability equal to 0.

The Node Information table

The Node information table stores the original pair of unique keys from the original table (GameId, EventNumber) with the matching NodeId called EndingId (see figure 4.3b). Each NodeId is unique, but different (GameId, EventNumber) tuples can have the same NodeId.

The Reward table

The Reward table stores the reward values associated to each node id. The values the RewardGoal variable takes are 0, 1 or -1 (see figure 4.3c). Since the main objective of Ice Hockey is scoring goals to win a match, the Reward associated for scoring a Goal is determined to be +1. The reward for receiving a goal then is -1, and 0 otherwise. In other papers such as in Routley and Shulte, the rewards given are 0 and 1 for Goals. By giving -1 to receiving a goal, certain actions under which there are a lot of possible events which can finish in other teams scoring will have a lower q-value, which could be seen as well as a measure of risk under certain situations.

4.3 The Markov Decision Process

The reward's value per state

The reward given here differs from Routleys paper and it is defined as following: For any state with a playing sequence starting with Home/Away and finishing with that same team scoring Goal Home/away, they will get a 1 $R_{A}way(GoalAway) = 1$ and $R_{H}ome(GoalHome) = 1$. However the state of receiving a goal by the other team $R_{H}ome(GoalAway) = -1$ and $R_{A}way(GoalHome) = -1$. Alternatively, the reward is set to be 0.

In order to set that in the rewards table, everything is set for the Home Team so Home Goals are +1 and Away Goals -1. Only the sign is necessary to be changed to calculate the q-values for Away team scoring.

Once the AD-tree has been created, a Markov Decision Process has been trained to compute the q-values related to likeness of the next home/away team scoring, being two MDP with 100,000 iterations and a convergence of 0.0001 (using Bellman's convergence). I provide below the pseudocode for the Value Iteration algorithm [23] using the notation used in section 2.1 .

Algorithm 3 Dynamic Programming for Value Iteration

```
Require: Markov Game Model, convergence criterion c, maximum number of iterations M
lastValue = 0
currentValue = 0
converged = False
for $i = 1; i \leqslant M, i \leftarrow i + 1$ do
    for all states s in the Markov Game Model do
        if $converged == False$ then
            $Q_{i+1} = R(s) + \frac{1}{Occ(s)} \sum_{(s,s') \in E} Occ(s,s') x Q_i(s'))$
            $currentValue = currentValue + |Q_{i+1}(s)|$
        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
```

Once it converges, the two variables (probability for Home and Away team to score) are stored in a table with the node id reference.

4.4 Creation of metrics

Once each pair of (GameId, EventNumber) with an event a(T,Z) (as in figure 4.1) has been associated with its q-value, several metrics has been created for posterior evaluation.

In order to understand the metrics, a couple of measures are introduced:

- The impact of a player's action stands for how much that specific player has contributed to score with an action performed compared to the previous state (e.g. If we are in GameId, eventNumber with NodeId = n and q-value = 0.3 and an action performed by a player, then previous state is related to the Node's information table where relations between that ids are also stored according to (GameId, EventNumber). If the previous value is 0.4 and also performed by a player, then the impact or contribution of the player with his action is $0.3 - 0.4 = -0.1$).

  $$impact(s, a) = q_T(s, a) - q_T(s)$$

  where T is the Team Performing the current action, s and a refer to Node state, and a to action.

- The time each player has played in each match has also been calculated from the data set.

All metrics are calculated for the regular season, which are 82 matches in the NHL. These metrics are:

- Direct Impact: it measures the individual performance of a player for each match. It will be represented in tables and plots as *Direct*.
- Collective Impact: it measures the collective valuation of the actions the team has performed while a specific player is on the field. It will be represented in tables and plots as *Collective*.
- Direct Impact / time (in hours): it is the same metric as in Direct Impact but taking into account time spent by each player on the field per match. It will be represented in tables and plots as *Directh*.
- Collective Impact Valuation / time (in hours): it is the same metric as in Collective Impact but taking into account time spent by each player on the field per match. It will be represented in tables and plots as *Collectiveh*.

Also, the same metrics will be plotted divided by Games Played per player. In this case, the metric will have the same name with *GP* in front of the metric (e.g.*GPDirect*).

For Regular Seasons 2007-2008 and 2008-2009, these metrics are plotted and evaluated in general terms to understand their behavior in the *results* section . Total Regular season metrics such as Position (Position), Salaries (Salary), Goals (G), Goals Assisted (GA), Points (P) or PlusMinus (PlusMinus) have been scraped from *http://www.dropyourgloves.com*.

Also, for each metric, a data set has been created with player's performance for each regular match. That means there are 82 matches x n players per year with valuations per match. For those cases where there are players that have played in 2 or 3 different teams during the same season, performance has been joined in a sole vector using the *GameId* variable as an order variable. In the weirdest case where a player would have played for 2 teams in a specific number match (eg. match 32), the match with higher performance would have been the one taken into account.

4.5 Time series forecasting of metrics

For each metric and player taken into consideration in each the metric, an ARIMA model has been fitted to evaluate which are the most typical ARIMA models taking into account the number of Games Played by each player in ranges of 10 for a specific season (e.g. the 10 range accounts for players that has played between 5-14 matches, and so on).

In order to evaluate how good these models are, for each metric, players with more than 30 matches Played has been subsetted and selected their first 30 matches. From this, the first 25 has been taken to fit an ARIMA model and predict the next following 5 matches with these ARIMA models. As a matter of a control, in each case, several accuracy measures has been evaluated as well as a control model for each case which is just taken the mean of all observations seen to predict the next one. The distribution of the errors is also shown for each case and metric.

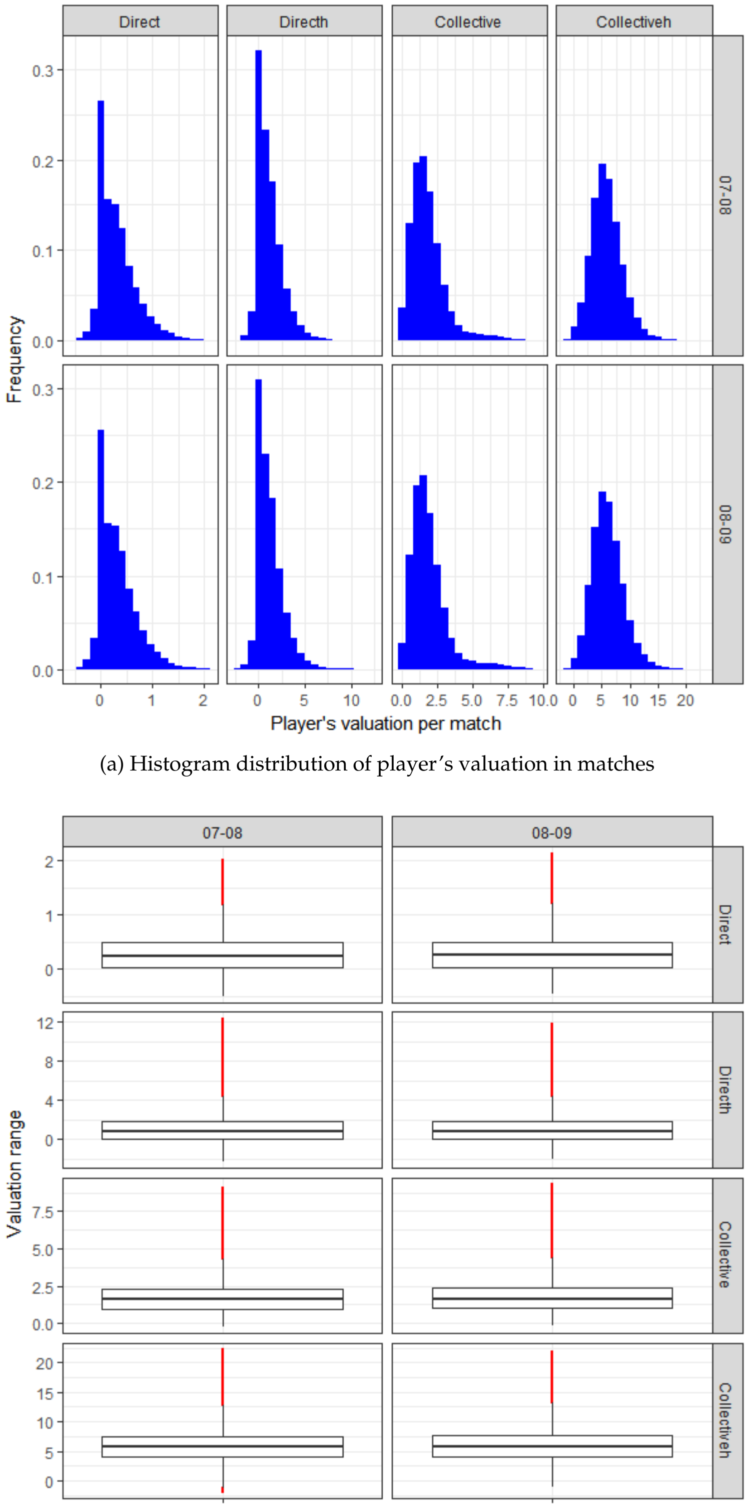
5 Results

5.1 General evaluation of the metrics

In figure 5.1, it can be seen the distribution of the 99.8% player's valuation in matches during Regular Seasons 2007-2008 and 2008-2009 (excluding 0.01% tails) for each metric. The players valuation distribution on matches for both the histogram and the boxplot is really similar across both years, which means that the metrics are stable. In all metrics as well, it can be seen that the valuation distribution is skewed to the right, possibly meaning that players often perform in between some ranges, and that there is some few players who consistently may have some outstanding performances that are larger than the average of the players performance.

In figure 5.2, a quantile regression is shown through the 82 matches for seasons 2007-2008 and 2008-2009, as well as a table showing the general year performances of quantiles with its mean and standard deviation and change compared to the previous year. The quantile plot shows that players range of performance through season is quite stable, meaning that there is no peaks or big changes on valuation for some specific weeks. In all metrics except the Collectiveh (Collective/time (hours)) have a clear separation between the mean and the median (the 50% quantile). This also makes sense since the distribution of matches seen in 5.1 was skewed to the right in all cases but in the Collectiveh metric where the distribution, although it is still skewed, looks much more normal than the other ones.

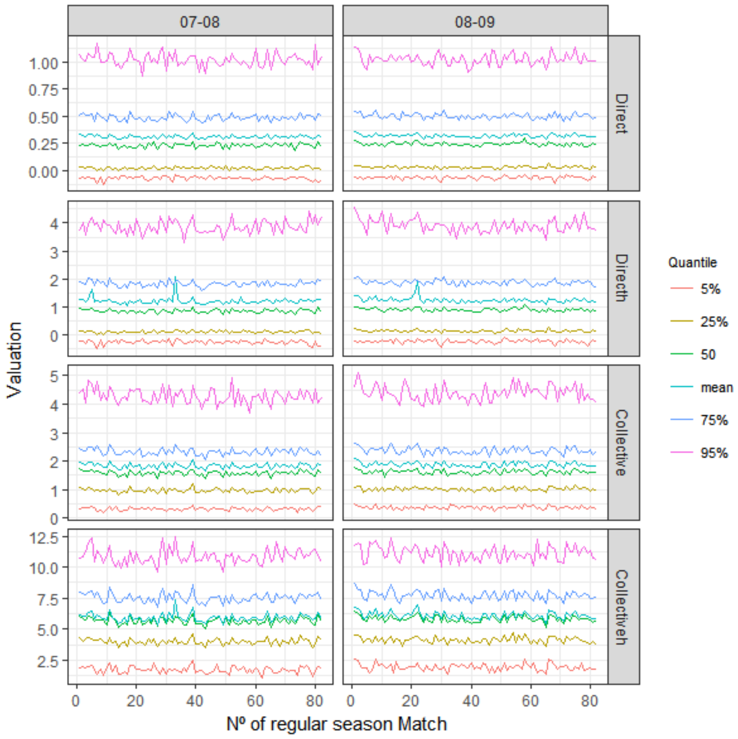
In figure 5.2, a regression of the different player's valuation metrics has been made on Salary (Valuation $\sim$ Salary) for players general Position scrapped from *www.dropyourgloves.com* taking into account also years. The first plot takes into account the sum of players valuation at the end of the regular season, whereas the second takes into account the sum of players valuation at the end of the regular season divided by the number of Games Played each 'player has performed. Additionally, other metrics such as Points (Goals +Assists) and the $+/-$ (PlusMinus) measure are also regressed to Salary. Positions are Central (C), Defender (D), Forward (F), Goalie (G), Left Wing (LW) and Right Wing (RW). In general terms, the PlusMinus measure does not seem to be a good metric to predict players Salary at all. The Direct and the Points measure seems to be good to predict players Salary for Centrals, Defender, Left and Right Wings players. On an even better scale, the Collective measure seems to perform better for the prediction of general Players performance. This might be misleading since the Collective measure takes into account all the impact of actions



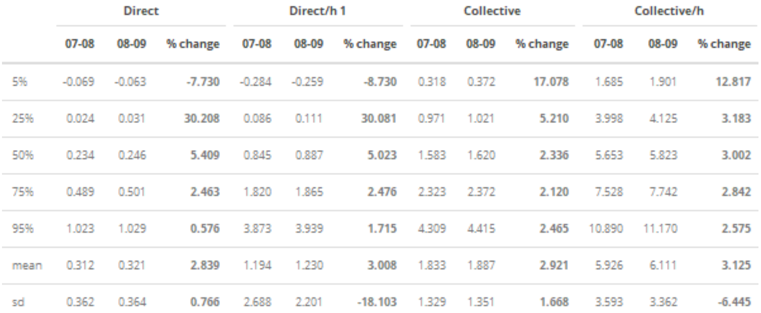
(a) Histogram distribution of player's valuation in matches

(b) Box plot distribution of player's valuation in matches. The box encloses values from $Q1 - 1.5(Q3 - Q1) to Q3 + 1.5(Q3 - Q1)$, where Q1 and Q3 are the 25% and 75% quantiles respectively

Figure 5.1: Distribution of the 99.8% player's valuation in matches during Regular Seasons 2007-2008 and 2008-2009 per metric (excluding 0.01% tails)



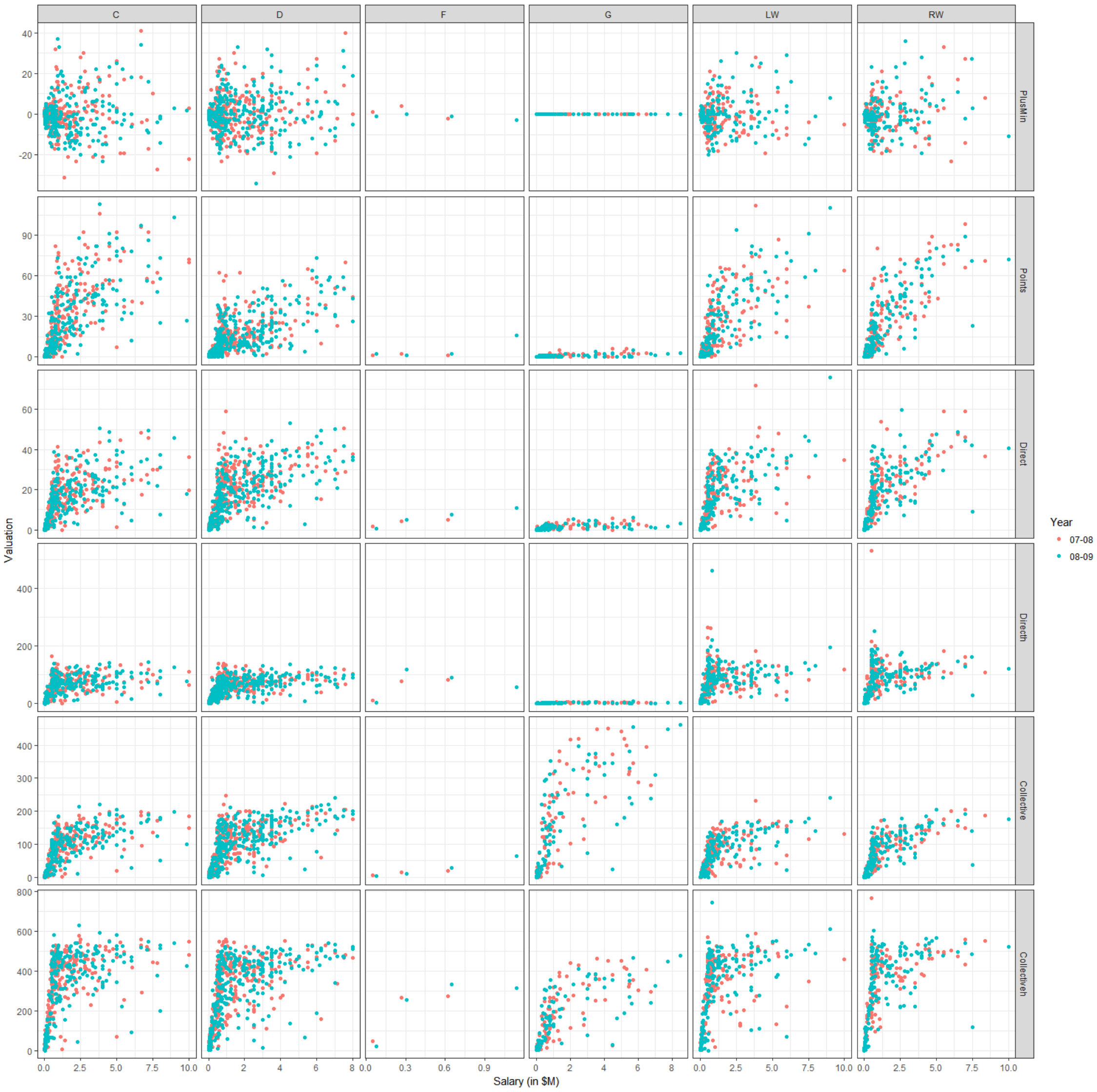
(a) Quantile plot of players valuation per match, Season and Metric



	Direct			Direct/h 1			Collective			Collective/h		
	07-08	08-09	% change	07-08	08-09	% change	07-08	08-09	% change	07-08	08-09	% change
5%	-0.069	-0.063	**-7.730**	-0.284	-0.259	**-8.730**	0.318	0.372	**17.078**	1.685	1.901	**12.817**
25%	0.024	0.031	**30.208**	0.086	0.111	**30.081**	0.971	1.021	**5.210**	3.998	4.125	**3.183**
50%	0.234	0.246	**5.409**	0.845	0.887	**5.023**	1.583	1.620	**2.336**	5.653	5.823	**3.002**
75%	0.489	0.501	**2.463**	1.820	1.865	**2.476**	2.323	2.372	**2.120**	7.528	7.742	**2.842**
95%	1.023	1.029	**0.576**	3.873	3.939	**1.715**	4.309	4.415	**2.465**	10.890	11.170	**2.575**
mean	0.312	0.321	**2.839**	1.194	1.230	**3.008**	1.833	1.887	**2.921**	5.926	6.111	**3.125**
sd	0.362	0.364	**0.766**	2.688	2.201	**-18.103**	1.329	1.351	**1.668**	3.593	3.362	**-6.445**

(b) Quantile Table of players valuation Season and Metric taking into account the standard deviation of the metrics

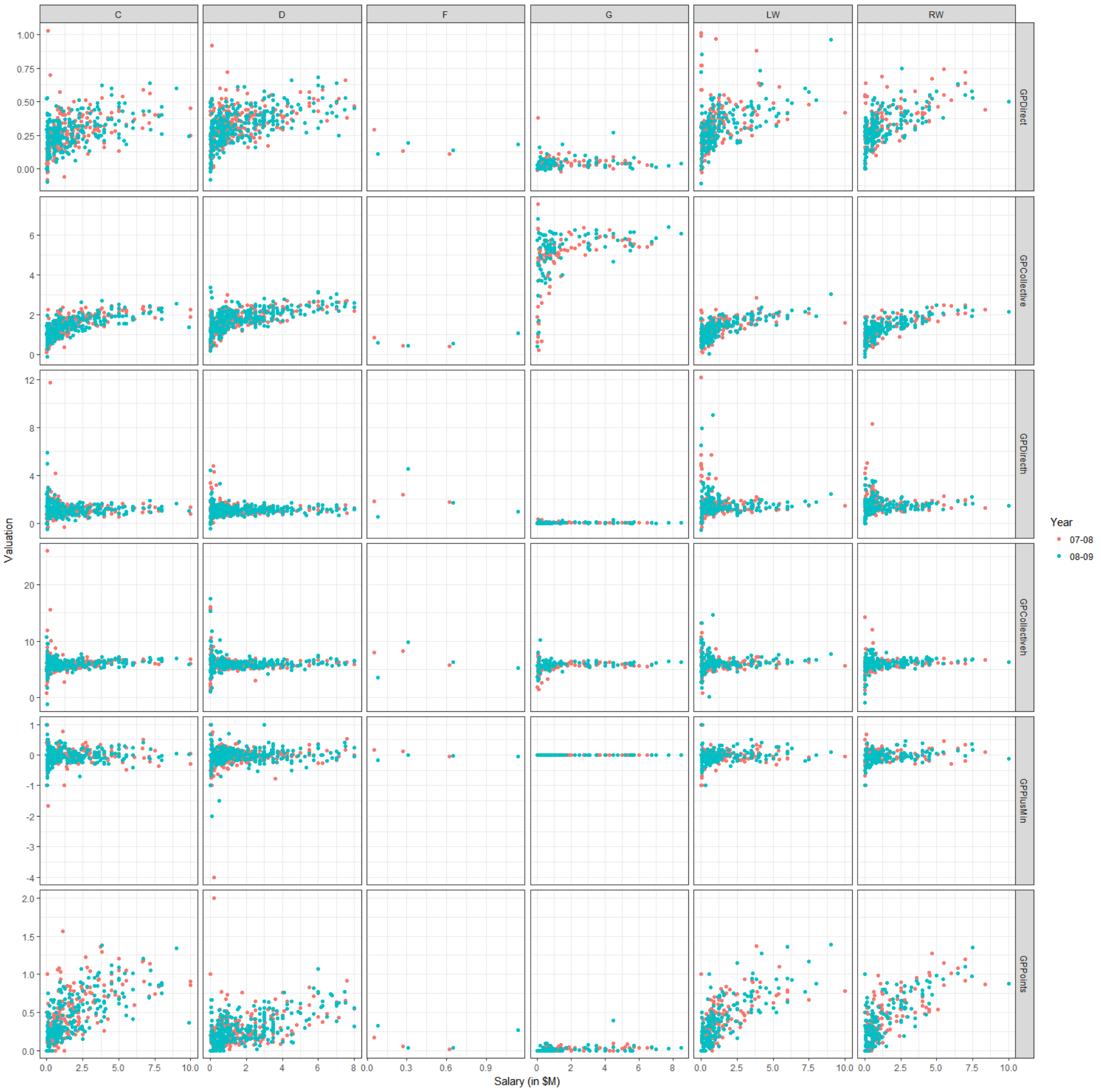
Figure 5.2: Quantile analysis of player's valuation in matches during Regular Seasons 2007-2008 and 2008-2009



(a) Regression of player's Valuation Metrics on Salary at the end of the season

of players while that specific player is in the rink. Of course, if a player has been seen to play good through the season, that player will continue playing even more time since he is proven useful to the team. Also, this measure can be misleading and must be interpreted only between Players playing in similar position. Since Goalies are the players who play most of the time, they are getting all impact actions of their teams and that is why their performances are the highest ones. Both Directh and Collectiveh seem to have some kind of logarithmic modelling, which could be interesting to predict players salary based on performances.

The second plot, which is related to the same metrics but divided by Games Played in each case, shows that GPPoints, GPDirect and GPCollective are be able to capture linearly the Salary Players base on their performance for Centrals, Defender, Left and Right Wings players. Nevertheless, Metrics divided by time seem not to have any impact on players performance. That makes sense, since Games Played measure takes into account to some extent the time a player plays (the more matches a player plays, the more is playing).



(b) Regression of player's Valuation Metrics on Salary at the end of the season divided by Games Played for each player

Figure 5.2: Regression of player's Valuation Metrics on Salary (Valuation∼ Salary) based on players general Position

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	**Direct**	Directh	Collective	Collectiveh
2007												
Alex Ovechkin	LW	22	3.83	82	65	47	28	112	**71.96**	182.65	232.56	588.85
Dion Phaneuf	D	22	0.94	82	17	43	12	60	**59.22**	134.05	246.12	559.67
Rick Nash	RW	23	5.50	80	38	31	3	69	**59.01**	181.80	158.82	485.99
Jarome Iginla	RW	30	7.00	82	50	48	27	98	**58.94**	161.92	204.12	560.88
Dustin Brown	RW	23	1.18	78	33	27	-13	60	**53.78**	156.41	171.40	501.48
Brenden Morrow	LW	28	4.10	82	32	42	23	74	**51.15**	146.62	171.59	504.57
Zdeno Chara	D	30	7.50	77	17	34	14	51	**50.74**	117.69	203.78	468.89
Trent Hunter	RW	27	1.55	82	12	29	-17	41	**50.31**	167.65	153.36	508.27
Mike Green	D	22	0.85	82	18	38	6	56	**48.26**	122.63	219.72	545.08
Pavel Datsyuk	C	29	6.70	82	31	66	41	97	**48.22**	134.68	198.44	559.41
2008												
Alex Ovechkin	LW	23	9.00	79	56	54	8	110	**75.93**	194.34	239.89	612.23
Dustin Brown	RW	24	2.60	80	24	29	-15	53	**59.76**	177.60	178.34	540.84
Shea Weber	D	23	4.50	81	23	30	1	53	**53.14**	136.10	201.19	511.36
Evgeni Malkin	C	22	3.83	82	35	78	17	113	**50.76**	134.92	220.41	591.75
Dion Phaneuf	D	23	7.00	79	11	36	-11	47	**50.34**	122.64	240.57	532.49
Vincent Lecavalier	C	28	7.17	77	29	38	-9	67	**49.46**	143.99	188.17	549.37
Sheldon Souray	D	32	6.25	81	23	30	1	53	**49.38**	125.86	203.08	514.73
Jeff Carter	C	24	4.50	82	46	38	23	84	**48.88**	141.78	189.35	548.30
Rick Nash	RW	24	6.50	78	40	39	11	79	**48.88**	145.11	171.59	498.26
Martin St. Louis	RW	33	5.00	82	30	50	4	80	**47.82**	135.55	204.19	569.06

Table 5.1: Top 10 Players performance for 2007-2008 and 2008-2009 for the Direct Metric.

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	**Directh**	Collective	Collectiveh
2007												
Mike Rupp	RW	27	0.5	64	3	6	-8	9	15.74	**529.76**	49.09	769.03
Riley Cote	LW	25	0.48	70	1	3	2	4	14.55	**263.64**	38.90	570.09
Jeff Cowan	LW	31	0.72	46	0	1	-5	1	10.62	**262.03**	39.03	316.68
DJ King	LW	23	0.5	61	3	3	-4	6	12.06	**227.25**	36.18	485.79
Eric Godard	RW	27	0.47	74	1	1	-8	2	10.80	**214.19**	35.18	505.55
Jared Boll	RW	21	0.74	75	5	5	-4	10	27.41	**201.09**	66.26	472.20
Alex Ovechkin	LW	22	3.83	82	65	47	28	112	71.96	**182.65**	232.56	588.85
Rick Nash	RW	23	5.5	80	38	31	3	69	59.01	**181.80**	158.82	485.99
Raitis Ivanans	LW	28	0.48	73	6	2	-10	8	20.88	**176.36**	57.73	445.11
David Clarkson	RW	23	0.8	81	9	13	1	22	33.91	**168.22**	97.97	476.47
2008												
Derek Boogaard	LW	26	0.8	51	0	3	3	3	10.83	**460.55**	33.48	747.36
Eric Godard	RW	28	0.72	71	2	2	-3	4	14.15	**249.95**	35.40	539.28
Dan Carcillo	LW	23	0.83	73	3	11	-15	14	33.14	**219.86**	94.20	501.42
Jody Shelley	LW	32	0.72	70	2	2	-6	4	20.11	**196.58**	52.38	505.46
Alex Ovechkin	LW	23	9	79	56	54	8	110	75.93	**194.34**	239.89	612.23
Evgeni Artyukhin	RW	25	0.89	73	6	10	1	16	35.72	**189.73**	94.61	504.91
Mike Rupp	RW	28	0.5	72	3	6	-2	9	24.52	**188.26**	73.94	517.70
Cam Janssen	RW	24	0.55	56	1	3	-5	4	13.11	**187.93**	36.51	478.01
Matt Cooke	LW	30	1.17	76	13	18	0	31	36.32	**185.57**	117.48	500.05
Riley Cote	LW	26	0.52	63	0	3	-7	3	12.10	**184.08**	36.36	543.95

Table 5.2: Top 10 Players performance for 2007-2008 and 2008-2009 for the Directh Metric (Direct/time(hours))

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	Directh	**Collective**	Collectiveh
2007												
Henrik Lundqvist	G	25	4.25	72	0	0	0	0	1.31	1.29	**449.62**	453.19
Miikka Kiprusoff	G	31	3.60	76	0	2	0	2	4.13	4.07	**447.26**	464.24
Evgeni Nabokov	G	32	5.00	77	0	2	0	2	3.00	2.98	**440.79**	453.59
Martin Brodeur	G	35	5.20	77	0	4	0	4	4.58	4.55	**419.29**	418.00
Ryan Miller	G	27	2.50	76	0	1	0	1	4.11	4.55	**418.40**	430.52
Cam Ward	G	23	2.00	69	0	1	0	1	1.67	1.70	**417.03**	440.68
Tomas Vokoun	G	31	5.30	69	0	6	0	6	2.87	2.87	**398.65**	413.01
Roberto Luongo	G	28	6.50	73	0	3	0	3	2.09	1.78	**393.72**	407.58
Vesa Toskala	G	30	1.38	66	0	5	0	5	1.73	1.74	**380.84**	391.68
Rick DiPietro	G	26	4.50	63	0	6	0	6	5.75	5.67	**372.26**	382.34
2008												
Miikka Kiprusoff	G	32	8.50	76	0	3	0	3	3.03	3.31	**462.40**	476.85
Marty Turco	G	33	5.70	74	0	5	0	5	6.24	6.48	**454.18**	467.39
Henrik Lundqvist	G	26	7.75	70	0	2	0	2	1.66	1.62	**447.99**	450.00
Cam Ward	G	24	2.50	67	0	1	0	1	2.24	3.09	**396.95**	410.44
Evgeni Nabokov	G	33	5.50	62	0	1	0	1	2.59	2.59	**380.95**	381.38
Marc-Andre Fleury	G	24	3.50	62	0	1	0	1	2.87	2.87	**375.41**	385.18
Niklas Backstrom	G	30	3.10	71	0	0	0	0	4.61	5.22	**373.02**	384.31
Dwayne Roloson	G	39	3.00	63	0	1	0	1	2.62	3.25	**353.05**	374.33
Steve Mason	G	20	0.85	61	0	0	0	0	1.63	1.65	**352.33**	357.26
Ilya Bryzgalov	G	28	4.00	64	0	2	0	2	0.69	0.70	**346.39**	364.53

Table 5.3: Top 10 Players performance for 2007-2008 and 2008-2009 for the Collective Metric

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	Directh	Collective	**Collectiveh**
2007												
Mike Rupp	RW	27	0.50	64	3	6	-8	9	15.74	529.76	49.09	**769.03**
Alex Ovechkin	LW	22	3.83	82	65	47	28	112	71.96	182.65	232.56	**588.85**
Nicklas Backstrom	C	20	2.40	82	14	55	13	69	28.80	86.12	187.96	**577.10**
Riley Cote	LW	25	0.48	70	1	3	2	4	14.55	263.64	38.90	**570.09**
Jarome Iginla	RW	30	7.00	82	50	48	27	98	58.94	161.92	204.12	**560.88**
Viktor Kozlov	C	32	2.50	81	16	38	28	54	32.25	108.51	168.31	**560.82**
Dion Phaneuf	D	22	0.94	82	17	43	12	60	59.22	134.05	246.12	**559.67**
Pavel Datsyuk	C	29	6.70	82	31	66	41	97	48.22	134.68	198.44	**559.41**
Eric Staal	C	23	4.50	82	38	44	-2	82	34.85	96.67	199.04	**552.43**
Tomas Kaberle	D	29	4.25	82	8	45	-8	53	38.32	93.36	221.93	**551.72**
2008												
Derek Boogaard	LW	26	0.80	51	0	3	3	3	10.83	460.55	33.48	**747.36**
Nicklas Backstrom	C	21	2.40	82	22	66	16	88	37.12	111.83	214.19	**630.43**
Alex Ovechkin	LW	23	9.00	79	56	54	8	110	75.93	194.34	239.89	**612.23**
Arron Asham	RW	30	0.64	78	8	12	0	20	18.24	143.65	83.85	**605.67**
Evgeni Malkin	C	22	3.83	82	35	78	17	113	50.76	134.92	220.41	**591.75**
Eric Staal	C	24	5.00	81	40	35	15	75	39.45	110.57	203.62	**581.78**
Brandon Dubinsky	C	22	0.64	82	13	28	-6	41	38.83	137.89	165.14	**580.76**
Nikolai Zherdev	LW	24	3.25	82	23	35	6	58	37.57	133.72	165.89	**579.38**
Chris Thorburn	RW	25	0.54	81	7	8	-10	15	23.15	143.05	93.92	**570.34**
Martin St. Louis	RW	33	5.00	82	30	50	4	80	47.82	135.55	204.19	**569.06**

Table 5.4: Top 10 Players performance for 2007-2008 and 2008-2009 for the Collectiveh Metric (Collective/time(hours))

In this latter part, the top 10 players for each year and metrics have been included. The Top 10 players on the Direct metric seems to perform relatively well in terms of Points (Goals and Assists). This makes sense since both metrics offer quite a similar regression line with Salary. As previously stated, the top 10 Players in the Collective measure are Goalkeepers, since they are the ones gathering all the impact valuation of their teams 5.3. Therefore, those metrics should be taken into account under Position circumstances, and not only under General Total Valuation.

Metrics accounting for time (Directh and Collectiveh) outputs players with low Goals and Assists on a general basis who might have played little in a lot of matches but who have had a great impact while they were in the rink.

6 Discussion

This chapter contains the following sub-headings.

6.1 Results

Are there anything in the results that stand out and need be analyzed and commented on? How do the results relate to the material covered in the theory chapter? What does the theory imply about the meaning of the results? For example, what does it mean that a certain system got a certain numeric value in a usability evaluation; how good or bad is it? Is there something in the results that is unexpected based on the literature review, or is everything as one would theoretically expect?

6.2 Method

This is where the applied method is discussed and criticized. Taking a self-critical stance to the method used is an important part of the scientific approach.

A study is rarely perfect. There are almost always things one could have done differently if the study could be repeated or with extra resources. Go through the most important limitations with your method and discuss potential consequences for the results. Connect back to the method theory presented in the theory chapter. Refer explicitly to relevant sources.

The discussion shall also demonstrate an awareness of methodological concepts such as replicability, reliability, and validity. The concept of replicability has already been discussed in the Method chapter (4). Reliability is a term for whether one can expect to get the same results if a study is repeated with the same method. A study with a high degree of reliability has a large probability of leading to similar results if repeated. The concept of validity is, somewhat simplified, concerned with whether a performed measurement actually measures what one thinks is being measured. A study with a high degree of validity thus has a high level of credibility. A discussion of these concepts must be transferred to the actual context of the study.

The method discussion shall also contain a paragraph of source criticism. This is where the authors' point of view on the use and selection of sources is described.

In certain contexts it may be the case that the most relevant information for the study is not to be found in scientific literature but rather with individual software developers and open source projects. It must then be clearly stated that efforts have been made to gain access to this information, e.g. by direct communication with developers and/or through discussion forums, etc. Efforts must also be made to indicate the lack of relevant research literature. The precise manner of such investigations must be clearly specified in a method section. The paragraph on source criticism must critically discuss these approaches.

Usually however, there are always relevant related research. If not about the actual research questions, there is certainly important information about the domain under study.

6.3 The work in a wider context

There must be a section discussing ethical and societal aspects related to the work. This is important for the authors to demonstrate a professional maturity and also for achieving the education goals. If the work, for some reason, completely lacks a connection to ethical or societal aspects this must be explicitly stated and justified in the section Delimitations in the introduction chapter.

In the discussion chapter, one must explicitly refer to sources relevant to the discussion.

7 Conclusion

This chapter contains a summarization of the purpose and the research questions. To what extent has the aim been achieved, and what are the answers to the research questions?

The consequences for the target audience (and possibly for researchers and practitioners) must also be described. There should be a section on future work where ideas for continued work are described. If the conclusion chapter contains such a section, the ideas described therein must be concrete and well thought through.

Bibliography

[1] Brigham BS Anderson and Andrew AW Moore. "ADtrees for Fast Counting and for Fast Learning of Association Rules." In: Kdd S 1 (1998), pp. 134–138. URL: http://www.aaai.org/Papers/KDD/1998/KDD98-020.pdf.

[2] James F. Barkell, Alun Pope, Donna O'Connor, and Wayne G. Cotton. "Predictive game patterns in world rugby sevens series games using Markov chain analysis". In: International Journal of Performance Analysis in Sport 17.4 (2017), pp. 630–641. ISSN: 14748185. DOI: 10.1080/24748668.2017.1381459. URL: http://doi.org/10.1080/24748668.2017.1381459.

[3] Richard Bellman. "A Markovian Decision Process". In: Indiana Univ. Math. J. 6 (4 1957), pp. 679–684. ISSN: 0022-2518.

[4] Dan Cervone, Alexander D'Amour, Luke Bornn, and Kirk Goldsberry. "POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data". In: SLOAN Sports Analytics Conference (2014), pp. 1–9. URL: http://dl.frz.ir/FREE/papers-we-love/sports%7B%5C_%7Danalytics/2014-ssac-pointwise-predicting-points-and-valuing-decisions-in-real-time.pdf.

[5] Jonathan D Cryer and Kung-Sik Chan. Time Series Analysis. With Applications to R. January. 2008, p. 487. ISBN: 9780387759586. DOI: 10.1007/978-0-387-75959-3. arXiv: arXiv:1011.1669v3. URL: http://books.google.com/books?hl=en%7B%5C&%7Dlr=%7B%5C&%7Did=MrNY3s2difIC%7B%5C&%7Doi=fnd%7B%5C&%7Dpg=PR2%7B%5C&%7Ddq=Time+series+analysis+with+applications+in+R%7B%5C&%7Dots=UqDwRRMrk5%7B%5C&%7Dsig=Hz1koyO45IysG2YiM0rEO2Maza4.

[6] Robert B. Gramacy, Shane T. Jensen, and Matt Taddy. "Estimating player contribution in hockey with regularized logistic regression". In: Journal of Quantitative Analysis in Sports 9.1 (2013), pp. 97–111. ISSN: 15590410. DOI: 10.1515/jqas-2012-0001. arXiv: 1209.5026.

[7] W G Hopkins, J a Hawley, and L M Burke. "Design and analysis of research on sport performance enhancement." In: Medicine and science in sports and exercise 31.3 (1999), pp. 472–485. ISSN: 0195-9131. DOI: 10.1097/00005768-199905000-00021. URL: http://www.ncbi.nlm.nih.gov/pubmed/10188754.

[8] R. A. Howard. Dynamic Programming and Markov Processes. Cambridge, MA: MIT Press, 1960.

[9] Mike D. Hughes and Roger M. Bartlett. “The use of performance indicators in performance analysis”. In: Journal of Sports Sciences 20.10 (2002), pp. 739–754. ISSN: 02640414. DOI: 10.1080/026404102320675602.

[10] Edward H. Kaplan, Kevin Mongeon, and John T. Ryan. “A Markov Model for Hockey: Manpower Differential and Win Probability Added”. In: INFOR: Information Systems and Operational Research 52.2 (2014), pp. 39–50. DOI: 10.3138/infor.52.2.39. eprint: https://doi.org/10.3138/infor.52.2.39. URL: https://doi.org/10.3138/infor.52.2.39.

[11] Mykel J. Kochenderfer, Christopher Amato, Girish Chowdhary, Jonathan P. How, Hayley J. Davison Reynolds, Jason R. Thornton, Pedro A. Torres-Carrasquillo, N. Kemal Üre, and John Vian. Decision Making Under Uncertainty: Theory and Application. 1st. The MIT Press, 2015. ISBN: 0262029251, 9780262029254.

[12] Hector Levesque, Fiora Pirri, and Ray Reiter. “Foundations for the Situation Calculus”. In: Linköping Electronic Articles in Computer and Information Science 3.18 (1998), p. 18. URL: http://www.ep.liu.se/ea/cis/1998/018/.

[13] Michael L. Littman. “Markov games as a framework for multi-agent reinforcement learning”. In: Machine Learning Proceedings 1994 (1994), pp. 157–163. ISSN: 00493848. DOI: 10.1016/B978-1-55860-335-6.50027-1. URL: http://linkinghub.elsevier.com/retrieve/pii/B9781558603356500271.

[14] Guiliang Liu and Oliver Schulte. “Deep Reinforcement Learning in Ice Hockey for Context-Aware Player Evaluation”. In: (2015), pp. 3442–3448.

[15] Dennis Ljung and Patrick Lambrix. “Player pairs valuation in ice hockey”. In: (2018).

[16] Brian Macdonald. “Adjusted Plus-Minus for NHL Players using Ridge Regression with Goals, Shots, Fenwick, and Corsi”. In: Journal of Quantitative Analysis in Sports 8.3 (2012), pp. 1–24. ISSN: 15590410. DOI: 10.1515/1559-0410.1447. arXiv: 1201.0317.

[17] Brian Macdonald. “An Expected Goals Model for Evaluating NHL Teams and Players”. In: MIT Sloan Sports Analytics Conference 2012 (2012), pp. 1–8. ISSN: 10769757. DOI: 10.1613/jair.301. arXiv: 9605103 [cs].

[18] Stephen Pettigrew. “Assessing the offensive productivity of NHL players using in-game win probabilities 2 A win probability metric for NHL games”. In: (2015), pp. 1–9.

[19] Kurt Routley and Oliver Schulte. “A Markov Game Model for Valuing Player Actions in Ice Hockey”. In: Uncertainty in Artificial Intelligence (UAI). 2015, pp. 782–791.

[20] Michael Schuckers and James Curro. “Total Hockey Rating (THoR): A comprehensive statistical rating of National Hockey League forwards and defensemen based upon all on-ice events”. In: Sports Analytics Conference 7 (2013), pp. 1–10. URL: http://www.sloansportsconference.com/wp-content/uploads/2013/Total%20Hockey%20Rating%20(THoR)%20A%20comprehensive%20statistical%20rating%20of%20National%20Hockey%20League%20forwards%20and%20defensemen%20based%20upon%20all%20on-ice%20events.pdf.

[21] Oliver Schulte, Mahmoud Khademi, Sajjad Gholami, Zeyu Zhao, Mehrsan Javan, and Philippe Desaulniers. “A Markov Game model for valuing actions , locations , and team performance in ice hockey”. In: Data Mining and Knowledge Discovery 31.6 (2017), pp. 1735–1757. ISSN: 1573-756X. DOI: 10.1007/s10618-017-0496-z.

[22] Oliver Schulte, Zeyu Zhao, Computing Science, Mehrsan Javan, and Philippe Desaulniers. “Apples-to-Apples : Clustering and Ranking NHL Players Using Location Information and Scoring Impact”. In: (2017), pp. 1–14.

[23] Richard S. Sutton and Andrew G. Barto. Introduction to Reinforcement Learning. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.